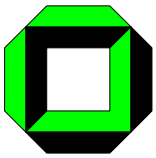


Universität Karlsruhe (TH)
Fakultät für Informatik
76128 Karlsruhe

IT-Management in der Praxis

Seminar – WS 2004/05

Herausgeber:
Prof. Dr. Wilfried Juling
Prof. Dr. Hannes Hartenstein
Jochen Dinger

Universität Karlsruhe (TH)
Institut für Telematik
Lehrstuhl Rechnersysteme und Infrastruktur der Informationsverarbeitung
Lehr- u. Forschungsbereich Dezentrale Systeme und Netzdienste
in Zusammenarbeit mit dem Rechenzentrum der Universität Karlsruhe (TH)

Interner Bericht 2005-1
ISSN 1432–7864

Zusammenfassung

Der vorliegende Interne Bericht enthält die Beiträge zum Seminar „IT-Management in der Praxis“, das im Wintersemester 2004/05 stattgefunden hat. Das Seminar wurde von den Lehrstühlen Prof. Wilfried Juling und Prof. Hannes Hartenstein in Zusammenarbeit mit dem Rechenzentrum der Universität Karlsruhe (TH) veranstaltet.

Das Seminar hat Themen aus folgenden Bereichen sowohl in technischer wie auch in betrieblicher Sicht behandelt:

- Cluster Computing
- Roaming
- IT-Sicherheit
- Softwareverteilung

Abstract

This Technical Report contains student papers of the seminar „IT-Management in der Praxis“ held in the winter semester 2004/05. The seminar was organized by the research groups of Prof. Wilfried Juling and Prof. Hannes Hartenstein in cooperation with the Computing Center of the University Karlsruhe (TH).

The seminar covers the following topics from a technical as well as from an operations point of view:

- Cluster Computing
- Roaming
- IT-Security
- Software Distribution

Inhaltsverzeichnis

Vorwort

In dem Seminar „IT-Management in der Praxis“ wurde Studierenden der Fakultät für Informatik die Möglichkeit gegeben, sich mit aktuellen Themen aus dem Umfeld des Rechenzentrums zu beschäftigen, wobei sowohl wissenschaftliche als auch betriebliche Aspekte zu betrachten waren.

Wir danken den Studierenden für ihr Engagement und die verfassten Beiträge sowie den Mitarbeitern des Rechenzentrums, welche die Seminarteilnehmer tatkräftig unterstützt haben: Willi Fries, Nikolaus Geers, Horst Gernert, Sabine Glas, Jörg Kramer, Roland Laifer, apl. Prof. Dr. Rudolf Lohner, Martin Rode, Reinhard Strebler und Ralf Wigand.

Die Beiträge stammen aus den vier Themenbereichen Cluster Computing, Roaming, IT-Sicherheit und Softwareverteilung. Hierbei zeigen die Beiträge auch die Überschneidung bzw. Interaktion der Themen untereinander auf.

0.1 Cluster Computing

Rechner-Cluster verschiedenster Ausprägung stehen heutzutage als Infrastruktur für die Berechnung wissenschaftlicher Probleme zur Verfügung. In Kombination mit dem kostenfreien Betriebssystem Linux ergeben sich neue Möglichkeiten und Herausforderungen, welche im Beitrag „Linuxcluster im RZ-Betrieb“ behandelt werden. Das Thema „High Performance Interconnects“ zeigt zudem verschiedene Techniken zur Vernetzung solcher Cluster auf. Eine Betrachtung verschiedener verteilter Dateisysteme findet in dem Beitrag „Distributed File Systems“ statt.

0.2 Roaming

Funknetze werden für Netzbetreiber immer bedeutungsvoller, da sie ständige Netzkonnektivität ermöglichen und daher von den Nutzern „anywhere, anytime“ gefordert werden. Diese allgegenwärtige Funktechnik verbunden mit einem sicheren Netzbetrieb stellt neue Herausforderungen an einen Netzbetreiber. Der Beitrag „Roaming in und zwischen Funknetzwerken“ behandelt sowohl Aspekte der Mobilität in Funknetzen als auch das Roaming zwischen verschiedenen Netzen.

0.3 IT-Sicherheit

Aus dem großen Spektrum der IT-Sicherheitsthemen wurden in diesem Seminar drei ausgewählte aktuelle Themen behandelt. Die starke Zunahme unerwünschter Werbe-Emails am gesamten Emailverkehr und die damit einhergehenden Einschränkungen des Dienstes erfordern Vorkehrungen, welche im Beitrag „Anti-Spam Techniken“ aufgezeigt werden.

Präventive Sicherheitsmaßnahmen wie etwa der Einsatz von Firewalls müssen durch reaktive Maßnahmen ergänzt werden. Hierzu stehen ergänzende Techniken zur Verfügung, die im Beitrag „Intrusion Detection und Prevention Systeme“ besprochen werden.

Zertifikate dienen als Basis vertraulicher Kommunikation. Der Beitrag „Zertifizieren und Signieren mittels UNIKA-CA“ befasst sich mit der Ausgabe sowie Verwaltung der Zertifikate innerhalb der Universität Karlsruhe.

0.4 Softwareverteilung

Der Betrieb eines Rechenzentrums erfordert den Einsatz von effizienten Softwareverteilungsmechanismen, um den Ansprüchen der Kunden nach aktuellen Systemen gerecht zu werden. Zudem sind solche Mechanismen auch zum Patch-Management und daher zum sicheren Betrieb von Rechnern notwendig. Der Beitrag „Softwareverteilung“ befasst sich mit den existierenden Mechanismen für die Betriebssysteme Windows und Linux.

Linuxcluster im RZ-Betrieb

Claus Wonnemann

Kurzfassung

Es wird auf die Merkmale eines Clustersystems eingegangen und es werden die Einsatzmöglichkeiten dieses Rechnertyps vorgestellt, wobei der grundsätzliche Unterschied zwischen den Bereichen *High Performance Computing* und *High Throughput Computing* aufgezeigt wird. In einer Beschreibung der hard- und softwareseitigen Komponenten eines Linuxclusters wird auf verschiedene Aufbaumöglichkeiten und häufig gebrauchte Programme für die Konfiguration und den Betrieb eines Linuxclusters eingegangen, wobei ein Schwerpunkt auf der *Condor*-Software liegt. Ein Überblick stellt die am Rechenzentrum der Universität Karlsruhe eingesetzten Linuxcluster vor und beschreibt deren Anwendungsgebiete sowie die besonderen Herausforderungen beim Aufbau und Betrieb eines solchen Systems. Ein Ausblick bewertet Linux als Clusterbetriebssystem aus Nutzer- und Herstellersicht.

1 Motivation

1.1 Anwendungsmöglichkeiten von Clustern

Cluster finden heute in vielfältigen Gebieten, sowohl der Wissenschaft und Forschung als auch der Industrie breite Anwendung. In fast allen Bereichen, in denen große Rechenleistungen zur Bewältigung der jeweiligen Aufgaben benötigt werden, sind mittlerweile Clustersysteme zu finden. So werden diese Computer beispielsweise in der Autoindustrie und im Rennsport zur Simulation von Crashversuchen oder zur Verbesserung der Aerodynamik eines Fahrzeugs eingesetzt. In den Geowissenschaften und der Meteorologie werden Cluster verwendet, um Aussagen zur Wahrscheinlichkeit von tektonischen Veränderungen treffen und zukünftige Klima- und Wetterverhältnisse vorhersagen zu können. Ein recht neuer Anwendungszweig findet sich in den Biowissenschaften, in denen Linuxcluster zur Sequenzierung genetischer Daten eingesetzt werden [RaVo05]. Durch ihre verteilte Struktur eignet sich diese Architektur vor allem für Probleme, die gut zu parallelisieren sind. Dieser potentielle Nachteil lässt sich durch den Einsatz moderner Verbindungstechnologien allerdings weitgehend kompensieren, so dass die Verwendung von Clustersystemen in praktisch allen Bereichen, in denen große Rechenkapazitäten gebraucht werden, zunimmt (siehe auch Abbildung 1).

Ein wesentlicher Grund für die zunehmende Verbreitung dieses Rechnertyps besteht sicherlich darin, dass Nutzergruppen, die bis dahin nicht über die Ressourcen zur Anschaffung eines Großrechners verfügten, nun in die Lage kommen, viel Rechenleistung zu vergleichsweise geringen Kosten erwerben zu können. So lassen sich aus handelsüblichen Arbeitsplatzrechnern und Verbindungsnetzwerken bereits sehr leistungsstarke Systeme aufbauen. Das in den letzten Jahren stark gewachsene Angebot an quelloffener Software hat zur freien Verfügbarkeit von Werkzeugen und Programmierbibliotheken zur Administration und Nutzung von Rechnerbündeln geführt, so dass Anwender nicht mehr auf kommerzielle und herstellerabhängige Programme angewiesen sind. An der US-amerikanischen Hochschule *Virginia Tech* wurde auf

diese Weise im Eigenbau ein Cluster aus 1100 *Apple Power Mac G5*-Rechnern aufgebaut, der im Jahr 2003 als drittschnellster Rechner der Welt geführt wurde [Virg04].

Einige der ersten Linuxcluster sind Anfang der 1990er Jahre unter dem Namen *Beowulf* bei der US-Raumfahrtbehörde *NASA* installiert worden [Beow05], wo sie zur Berechnung astronomischer Modelle verwendet wurden [CT P05]. Bei diesen Projekten wurde von Anfang an Wert darauf gelegt, ein leistungsfähiges System ausschließlich aus günstigen Standardkomponenten aufzubauen, was sicherlich auch die Wahl von Linux als Betriebssystem begünstigt hat.

1.2 Merkmale eines Clusters

Als (Computer-)Cluster wird ein durch ein Netzwerk gekoppelter Verbund von Rechnerknoten bezeichnet, die jeweils über eigenen Speicher und über einen oder auch mehrere Prozessoren verfügen. Cluster werden zunehmend beliebter, da sie sehr leistungsfähig sind, sich oft aus vergleichsweise günstigen Standardkomponenten aufbauen lassen und je nach verfügbarem Budget einfach vergrößer- und modernisierbar sind. Zudem stellen sie oft die einzige Möglichkeit zur Bewältigung immer anspruchsvoller werdender Computeraufgaben dar, da die Leistungssteigerung einzelner Prozessoren durch Erhöhungen der Transistor-Packungsdichten immer problematischer und damit auch teurer wird. Bei eng gekoppelten Parallelrechnern stellt sich mit zunehmender Größe zudem das Problem einer ausreichenden Kühlung des Systems, während dem Ausbau von Clustern im Prinzip keine Grenzen gesetzt sind.

Ein besonderes Merkmal von Clustern, das oft als der Flaschenhals dieser Systeme bezeichnet wird, ist die im Vergleich zur Prozessorgeschwindigkeit um mehrere Größenordnungen langsamere Kommunikation zwischen den Rechnerknoten. Obwohl in den letzten Jahren große Fortschritte in der Entwicklung besserer Verbindungstechniken mit höheren Durchsätzen und kürzeren Latenzzeiten gemacht worden sind, hat sich der Unterschied durch eine überproportionale Leistungssteigerung der Prozessoren weiter vergrößert. Entwurfsziel jeder Software, die auf Clustersystemen laufen soll, muss daher eine möglichst hohe Lokalität der Daten und eine Minimierung der Kommunikation im Netzwerk sein.

1.3 *High Performance Computing* vs. *High Throughput Computing*

Trotz des gemeinsamen Oberbegriffs verbergen sich hinter der Bezeichnung „Cluster“ teilweise sehr verschiedene Rechnerarchitekturen. So gibt es große Unterschiede hinsichtlich der verwendeten Knoten und Kommunikationshardware, die einen erheblichen Einfluss auf die Anwendbarkeit eines Systems in verschiedenen Problemklassen haben. Für bestimmte zeitkritische Aufgaben, wie beispielsweise die Erstellung einer aktuellen Wettervorhersage, ist es wichtig, dass kurzfristig eine hohe Rechenleistung zur Verfügung steht, um das Problem rechtzeitig lösen zu können. Der Ausbau eines Clustersystems zur Leistungssteigerung wird in diesem Bereich sinnvoller Weise dazu genutzt, die Antwortzeit zu verkürzen, also die gestellte Aufgabe schneller bearbeiten zu können. Diese Problemklasse, in der der parallele Einsatz vieler Rechner hauptsächlich zur Beschleunigung der Programmausführung genutzt wird, stellt die höchsten Anforderungen an die verwendete Hardware und benötigt effiziente parallele Algorithmen um den Rechnerverbund gut ausnutzen zu können. Da alle Rechnerknoten an verschiedenen Teilbereichen desselben Problems arbeiten, ist der Synchronisations- und Kommunikationsbedarf in der Regel sehr hoch. Dedizierte Clustersysteme und speziell für die Hochgeschwindigkeitskommunikation entwickelte Netzwerke kommen vor allem in diesem als *High Performance Computing* (HPC) oder als Höchstleistungsrechnen bezeichneten Gebiet zum Einsatz. Der Kostenvorteil, den Cluster oft im Vergleich zu anderen Rechnerarchitekturen besitzen, fällt hier am wenigsten ins Gewicht, da die Systeme meist für den jeweiligen Einsatz zugeschnitten und mit teurer Spezialhardware ausgestattet sind.

Einen anderen Ansatz verfolgt das *High Throughput Computing* (HTC), dessen Ziel nicht die Beschleunigung der Programmausführung, sondern eine Erhöhung des Durchsatzes mittels paralleler Verarbeitung ist. Anstatt eine Aufgabe schneller zu berechnen, werden viele Aufgaben gleichzeitig behandelt. Die Anforderungen an die Kommunikationsleistung im Netzwerk sind hier niedrig, da alle Knoten mit lokalen Daten arbeiten können und lediglich Aufträge verteilt und Ergebnisse eingesammelt werden müssen. Es ist außerdem einfach möglich, verschiedene Plattformen zu einem heterogenen Verbund zusammenzufügen und Rechner verschiedener Leistungsklassen gemeinsam arbeiten zu lassen, da jeder Computer eigene Aufgaben bekommt und Knoten nicht gegenseitig aufeinander warten müssen. Die Rechnerknoten brauchen nicht ausschließlich zur Verwendung im Cluster zur Verfügung zu stehen, sondern können beispielsweise als normale Arbeitsplatzrechner verwendet werden, die bei Nichtbenutzung in den Verbund einbezogen werden. Ein Anwendungsbeispiel ist das *Rendering* eines Films, bei dem jeweils kleine Abschnitte den Knoten zur Verarbeitung übergeben und anschließend die bearbeiteten Stücke abgeholt und zusammengefügt werden. In einem größeren Maßstab arbeitet auch das *SETI@home*-Projekt der *University of California at Berkeley* nach diesem Prinzip [SETI04]. Hier werden Daten von Radioteleskopen den weltweit verteilten Rechnern der Teilnehmer übergeben, die diese analysieren solange der Computer nicht anderweitig benutzt wird.

Sicherheitsüberlegungen (im Sinne von *Security*) spielen bei der Entwicklung von Clustersystemen eine zunehmend größer werdende Rolle. Zum einen müssen Systeme, auf die auch durch nicht autorisierte Personen zugegriffen werden kann (zum Beispiel über das Internet), gegen Einbruchs- und Angriffsversuche geschützt werden. Zu diesem Zweck werden Verfahren zur *intrusion detection and prevention*, die auch in einem folgenden Beitrag detailliert beschrieben werden, genutzt. Vor allem im HTC-Bereich müssen darüber hinaus die gleichzeitig laufenden Anwendungen und dazugehörenden Daten, die von jeweils unterschiedlichen Auftraggebern stammen können, voneinander abgeschirmt und gegen Einsichtnahme geschützt werden. Gerade kommerzielle Nutzer, die Rechenleistung auf fremden Systemen nutzen, erwarten ein hohes Maß an Sicherheit, insbesondere wenn sensible Informationen und Firmeninterna auf diesen Rechnern verarbeitet werden. Bei vielen für den Clustereinsatz entwickelten Softwaresystemen, wie beispielsweise der *Condor*-Software (siehe Abschnitt 4), gibt es daher Bestrebungen, diese um leistungsfähige Sicherheitsinfrastrukturen zu erweitern.

1.4 Leistungsfähigkeit

Durch ihre einfache Erweiterbarkeit sind der Rechenleistung von Clustersystemen im Prinzip keine Grenzen gesetzt. Im Gegensatz zu anderen Rechnertypen ist es problemlos möglich, ein zu Anfang kleines und günstiges System wachsenden Anforderungen schrittweise anzupassen. Im Bereich des Höchstleistungsrechnens besteht die größere Herausforderung sicherlich darin, diese enorme Leistung durch die Entwicklung geeigneter paralleler Anwendungen auch ausnutzen zu können.

Die von der Universität Mannheim, der *University of Tennessee* und dem *National Energy Research Scientific Computing Center* zweimal jährlich herausgegebene Liste der fünfhundert schnellsten Rechner zeigt anschaulich die wachsende Bedeutung von Clustersystemen im Segment des *High Performance Computing* [TOP504]. Die Basis der Leistungsbewertung in dieser Rangliste bildet der *Linpack*-Benchmark, der aus der Geschwindigkeit, mit der ein lineares Gleichungssystem gelöst werden kann, auf die Anzahl der Gleitkommaoperationen, die der Testrechner pro Sekunde durchführen kann, schließt. In Abbildung 1 ist der Anteil, den die verschiedenen Rechnerarchitekturen an der Liste seit ihrer ersten Veröffentlichung im Jahr 1993 gehabt haben, dargestellt. Inzwischen stellen Cluster mehr als die Hälfte der auf dieser Liste verzeichneten Rechner, wobei die genaue Abgrenzung der Architekturen voneinander

ander allerdings manchmal schwierig ist. Dennoch spiegelt die Liste den ausgeprägten Trend zu nachrichtengekoppelten Parallelrechnern im Supercomputer-Segment gut wider.

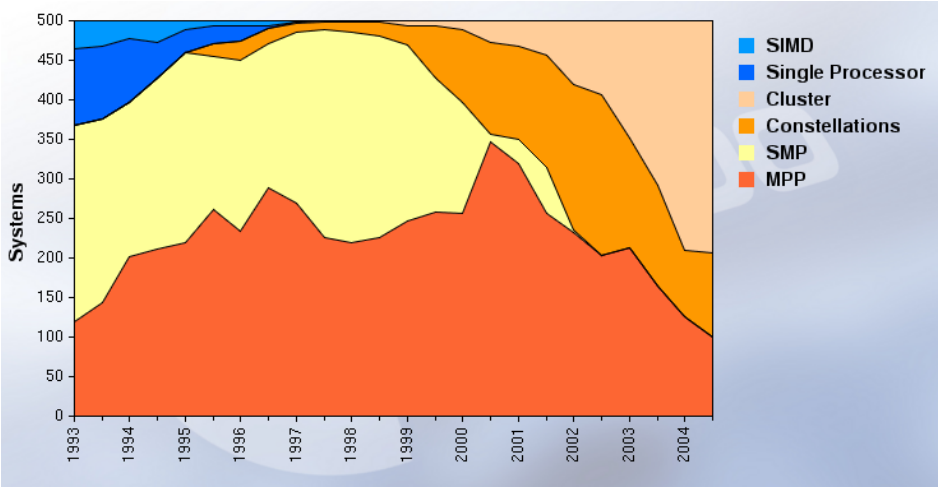


Abbildung 1: Anteile verschiedener Rechnerarchitekturen an der 24. *Top500*-Liste (November 2004) [TOP504]

2 Aufbau eines Clusters

2.1 Rechnerknoten

Als Knoten eines Clusters kommen im Prinzip alle Arten von Rechnern, die über eigenen Speicher verfügen, in Betracht. Diese können sowohl mit einem Prozessor bestückt wie auch als symmetrische Multiprozessoren ausgeführt sein. Da nicht-dedizierte Rechnerbündel auch für andere Aufgaben genutzt werden, sind die Knoten hier in der Regel gewöhnliche Arbeitsplatzrechner mit angeschlossenen Peripheriegeräten, die beispielsweise in einem Büro oder Computerpool stehen und jeweils mit dem gemeinsamen Netzwerk verbunden sind. In ausschließlich für die parallele Verarbeitung bestimmten Rechnerbündeln (dedizierten Clustern) kann auf den Anschluss externer Hardware an die Knoten und an die Erweiterung durch Steckkarten normalerweise verzichtet werden. Diese Rechner sind daher oft Platz sparend in für den Einbau in so genannten *Racks* vorgesehenen Einschubgehäusen untergebracht und können einfach über die rückseitig angebrachten Anschlüsse der Netzwerkadapter miteinander verkabelt werden. Auf diese Weise lässt sich mit einigen *Racks*, in denen jeweils mehrere Rechnerknoten montiert sind, bei relativ geringem Platzbedarf ein großes Clustersystem aufbauen. Abbildung 2 zeigt den „Tungsten“-Linuxcluster, der im November 2004 als zehntschnellster Rechner der Welt geführt wurde.

2.2 Verbindungen

Als Verbindungsnetzwerk nicht-dedizierter Clustersysteme wird in der Regel *Fast Ethernet* oder auch *Gigabit Ethernet* eingesetzt. Diese Verbindungstechnologie ist sehr kostengünstig und für die Kommunikation zwischen Arbeitplatzrechnern im Normalfall völlig ausreichend. Soll auf einem solchen Cluster aber eine verteilte Anwendung (die gleichzeitig mehrere Maschinen beansprucht) betrieben werden, kann sich das Netzwerk sehr schnell als ein Flaschenhals erweisen, der den Vorteil einer höheren Rechenleistung des verteilten Systems zunichte macht.

Abbildung 2: „Tungsten“-Linuxcluster am *National Center for Supercomputer Applications* [NCSA04]

Die im Vergleich zu Hauptspeicherzugriffen niedrigen Bandbreiten und vor allem hohen Latenzzeiten sind unbrauchbar für verteilte Anwendungen mit mittlerem oder hohen Kommunikationsaufwand. Neben der Hardware eines Verbindungsnetzwerkes hat auch das eingesetzte Kommunikationsprotokoll einen entscheidenden Einfluss auf die Anwendbarkeit eines Clustersystems. So kann beispielsweise das von einigen *Ethernets* verwendete *CSMA/CD*-Protokoll (*Carrier Sense Multiple Access/Collision Detection*), das unkoordiniertes Senden erlaubt und auf Kollisionen mit erneutem Senden reagiert, im Prinzip keine Garantien bezüglich der Latenzzeit geben und ist somit in HPC-Systemen unbrauchbar.

Aus diesen Gründen werden in dedizierten Clustern leistungsfähigere, aber auch sehr viel teurere Netzwerktechnologien eingesetzt, die Bandbreiten von mehreren Gigabit pro Sekunde und Latenzzeiten im Bereich einiger Mikrosekunden erreichen. Die wichtigsten Vertreter sind das von der Firma *Myricom* entwickelte *Myrinet* [Myri04], die Standards *SCI* (*Scalable Coherent Interface*) [SCI 04] und *InfiniBand* [Infi04], das *QsNet* von *Quadrics* [Quad04] sowie *Switched Gigabit Ethernet*. Diese Netzwerke übertreffen teilweise die Leistungsfähigkeit der in Computersystemen eingesetzten Bussysteme wie *PCI* oder auch *PCI-X*, die den Prozessor mit externen Geräten wie eben auch dem Netzwerkadapter verbinden. Dieser Verbindungsengpass soll durch neue Technologien wie *PCI Express* oder auch eine direkte Anbindung an den Speicherbus behoben werden. Eine detaillierte Beschreibung von Hochleistungs-Verbindungsnetzwerken wird in dem folgenden Seminarbeitrag gegeben.

2.3 Topologien

Die Topologie eines Netzwerks beschreibt die Organisation der Verbindungen zwischen den Knoten und ist somit entscheidend für die Erreichbarkeit von Rechnern im Netz sowie die Ausfallsicherheit des Systems. Kommt es beispielsweise zu einer Unterbrechung in der Verbindung zweier Rechner, so können diese trotzdem weiterhin miteinander kommunizieren, falls die Möglichkeit einer alternativen Wegewahl besteht.

Eine weit verbreitete Topologie, die auch in *Ethernets* Verwendung findet, sofern die Rechner nicht über *Switches* verbunden sind, ist der Bus, bei der alle Geräte an einem Hauptkabel angeschlossen sind. Sie bildet die einfachste und preiswerteste Form zum Aufbau eines dynamischen Netzes, das problemlos um weitere Rechner erweitert werden kann. Fällt ein Knoten aus, so können die übrigen weiterhin miteinander kommunizieren, ein Fehler in der Verbindung jedoch kann nicht ausgeglichen werden. Zudem ist die Belastung des Netzes recht hoch, da sämtliche Rechner alle Nachrichten mithören und die für sie bestimmten herausfiltern müssen.

Die in dedizierten Clustern gebrauchten Netzwerktopologien sollen möglichst kurze und redundante Verbindungen zwischen den Rechnerknoten ermöglichen und Engpässe vermeiden helfen. Häufig verwendete Typen sind zwei- oder mehrdimensionale Tori, über Kreuzschienenverteiler gekoppelte Knoten und so genannte *Fat Trees* (siehe Abbildung 3). Bei letztgenannter Topologie ist das Netzwerk wie ein Baum strukturiert, dessen Kapazität mit der Nähe zur Wurzel zunimmt und dessen Blätter von den Rechnerknoten gebildet werden.

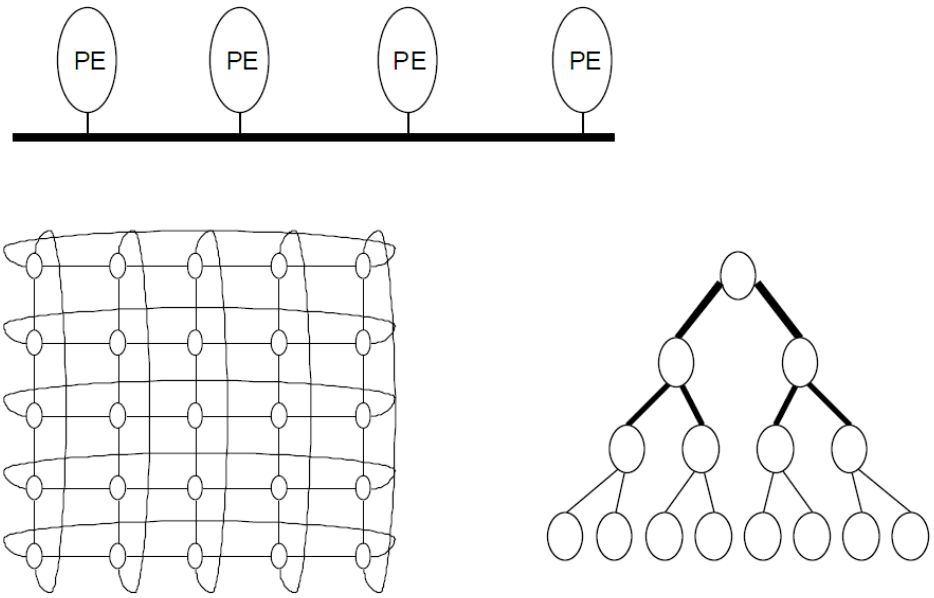


Abbildung 3: Schematische Darstellung der Topologien Bus, Torus und *Fat Tree*

3 Software

3.1 Linux als Cluster-Betriebssystem

Mittlerweile wird von fast allen Herstellern dedizierter Clustersysteme, die auf einer *Intel*-Prozessorarchitektur basieren, Linux als Betriebssystem angeboten. In der Regel handelt es sich dabei um eine angepasste Variante weit verbreiteter Linux-Distributionen für die entsprechende Plattform, wie *Red Hat Enterprise Linux* [Red 04] oder *SuSE Linux Enterprise Server* [SUSE04], die um Werkzeuge und Treiber, zum Beispiel für den Netzwerkadapter oder ein verteiltes Dateisystem, ergänzt worden sind. Aus Sicht eines Unternehmens gibt es sicherlich einige Gründe, Linux in sein Angebot aufzunehmen und aktiv zu fördern, auch wenn dies zu Lasten eines firmeneigenen, kommerziellen Betriebssystems gehen sollte. So entfällt durch die freie Verfügbarkeit und ständige Verbesserung der Aufwand für die Entwicklung und Pflege eines eigenen Systems für die entsprechende (meist *Intel*-)Plattform, während durch die Offenheit der Quelltexte trotzdem spezifische Anpassungen an das eigene Produkt möglich sind.

Die (früher) oft geäußerten Bedenken hinsichtlich einer mangelnden Sicherheit oder Zuverlässigkeit von Linux beziehungsweise quelloffener Software im Allgemeinen, werden inzwischen in weiten Teilen der Industrie als nicht mehr zutreffend angesehen, so dass ihr Einsatz auch in kommerziellen Produkten zunimmt. Im Clustereinsatz verwendete Software, wie zum Beispiel Programmierbibliotheken, *Middleware* und Datenbanken, ist in einem für den Produktiveinsatz geeigneten Entwicklungsstand für Linux verfügbar, sowohl unter freien als auch unter kommerziellen Lizenzen. Zwar gibt es auch weitere frei erhältliche Betriebssysteme, die für den Clustereinsatz in Frage kommen, wie zum Beispiel *FreeBSD* [The 05], jedoch hat sich Linux bei den Herstellern von Clustersystemen klar durchgesetzt, was sicherlich vor allem auf die große Anwendergemeinde und die Vertrautheit vieler Benutzer mit diesem System zurückzuführen ist [RaVo05].

Aus den bereits genannten Gründen bietet sich Linux natürlich auch für nichtdedizierte Cluster an, zumal, falls es auf den Knoten bereits als Arbeitsplatzbetriebssystem genutzt wird. Die lange Erfahrung, die in der Wissenschaft sowie der industriellen und staatlichen Forschung (die insbesondere in den USA ein Hauptabnehmer von HPC-Systemen ist) mit Linux gemacht worden ist, hat diesem Betriebssystem sicherlich auch zu einer verstärkten Nachfrage im Supercomputer-Segment verholfen.

3.2 Konfigurationswerkzeuge

Da ein Cluster aus vielen hundert oder tausend Rechnerknoten bestehen kann, wird es mit zunehmender Größe schwieriger, systemweit Veränderungen an der Konfiguration vorzunehmen. Manuelle Wartungsarbeiten an jedem Knoten sind sehr zeitaufwendig und auch fehlerträchtig, daher liegt es nahe, diese Arbeiten weitestgehend zu automatisieren.

Zu den häufig durchgeführten Aufgaben im Clusterbetrieb gehören beispielsweise die Anpassung von Konfigurationsdateien, das Feststellen und Beheben von Fehleinstellungen und das Einspielen aktualisierter Softwareversionen. Das Prinzip vieler Konfigurationswerkzeuge für den Clusterbetrieb besteht nun darin, eine vorgegebene Richtlinie systemweit auf den Knoten umzusetzen. Da bei vielen Clustern alle Rechner für dieselben Aufgaben bestimmt sind, ist eine einheitliche Konfiguration oft ausreichend, jedoch kommt es auch vor, dass manchen Rechnern spezielle Aufgaben übertragen werden, für die besondere Einstellungen nötig sind, so zum Beispiel für Dateiserver, Überwachungsknoten oder *Frontends*.

Das an der Universität Oslo entwickelte *Cfengine* besteht aus einem System autonomer Agenten, das über den gesamten Cluster verteilt ist [Cfen04]. In einer Skriptsprache lässt sich eine Konfiguration erstellen, wobei von vordefinierten Routinen zur Erledigung häufig anfallender Aufgaben Gebrauch gemacht werden kann. Dazu gehören beispielsweise das Editieren von Dateien, die Konfiguration der Netzwerkschnittstelle, das Ändern von Datei- und Ausführungsrechten oder das Einbinden von Dateisystemen. Durch eine klassenbasierte Entscheidungsstruktur entfallen die üblichen Fallunterscheidungen, zum Beispiel bezüglich der zugrundeliegenden Rechnerarchitektur, indem für jeden Fall eine spezifische Klasse definiert wird und die Entscheidung über die Anwendung dem Agenten überlassen wird. Ausgehend von einer solchen Richtlinie überprüft *Cfengine* die vorliegenden Konfiguration und passt sie gegebenenfalls an.

Einen anderen Ansatz verfolgt die ebenfalls nichtkommerzielle *Rocks Cluster Distribution*, die eine komplette, auf *Red Hat* basierende, Linux-Distribution umfasst [Rock04]. Neben Anwendungen für den Clusterbetrieb, wie zum Beispiel *Condor* (siehe Abschnitt 4), sind speziell auf verschiedene Plattformen angepasste Versionen von Betriebssystemkomponenten enthalten, so beispielsweise für die Prozessorarchitekturen von *AMD* und *Intel* optimierte Laufzeitbibliotheken.

Auf den Rechnerknoten benötigte Anwendungen und Einstellungen werden in Konfigurationsdateien festgelegt, die wiederum in einem Graphen hierarchisch angeordnet werden können. Auf diese Weise können, ausgehend von einer Basisversion, für verschiedene Rechnerknoten alternative Konfigurationen festgelegt werden.

Anstatt Modifikationen an einer bestehenden Betriebssysteminstallation vorzunehmen, ist die komplette Neuinstallation die grundsätzliche Vorgehensweise der *Rocks Cluster Distribution*. Sollen Veränderungen an der Konfiguration vorgenommen oder zusätzliche Software eingespielt werden, wird die auf einem Rechnerknoten vorhandene Installation durch eine neue, entsprechend angepasste Variante ersetzt. Insbesondere auf großen Clustern fällt der Zusatzaufwand nicht mehr stark ins Gewicht, da die Installation auf vielen Knoten parallel vorgenommen werden kann und eine Analyse der vorhandenen Konfiguration überflüssig wird. Dementsprechend viel Wert ist bei *Rocks Cluster* darauf gelegt worden, die Betriebssysteminstallation automatisch ablaufen lassen zu können.

3.3 Batchsysteme

Mit Hilfe von Batchsystemen können einem Computersystem viele Aufträge auf einmal übergeben werden, die dann nacheinander ohne weitere Benutzereingriffe abgearbeitet werden. Im Falle eines Clusters sorgt das Batchsystem für eine möglichst effiziente Aufteilung der Aufgaben auf die Rechnerknoten und stellt ein Rechnerbündel auf diese Weise gegenüber dem Benutzer als eine einzige Rechenressource dar, der ohne Berücksichtigung der verteilten Struktur Aufträge übertragen werden können. Dabei kann es sich sowohl um sequentielle Programme handeln, die einem Rechnerknoten zugeordnet werden, als auch um parallele Anwendungen, die zur Bearbeitung mehrere Computer gleichzeitig beanspruchen. Dem Batchsystem fällt die Aufgabe zu, nach freien, für die Bearbeitung des Programms geeigneten Ressourcen zu suchen, und dieses unter Bereitstellung eventuell benötigter Bibliotheken oder sonstiger Dateien dort auszuführen. Beispielsweise benötigen bestimmte Anwendungen eine spezielle Plattform oder ein gewisses Mindestmaß an Rechenleistung, so dass insbesondere in heterogenen Umgebungen eine sinnvolle Zuordnung zu den Rechnerknoten wichtig ist. Als Rückgabe erhält der Anwender die Resultate des Programmlaufs, die bei parallelen Anwendungen von mehreren Knoten eingesammelt und zusammengeführt werden müssen.

Die Reihenfolge der Abarbeitung hängt von der gewählten *Scheduling*-Strategie ab und kann zum Beispiel einer Priorisierung, einem Warteschlangen- (*first-in, first-out*) oder Durchsatzmodell (weniger umfangreiche Aufträge werden bevorzugt) folgen. Zudem gibt es die Möglichkeit, für eine Aufgabe zu einem bestimmten Zeitpunkt Rechenleistung vorab zu reservieren. Besteht zwischen mehreren abzuarbeitenden Aufträgen eine Abhängigkeitsbeziehung, so kann diese in einem Graphen ausgedrückt werden, der das Batchsystem veranlasst, Programme, deren vollständige Abarbeitung die Ausführung einer anderen Anwendung bedingt, entsprechend früher laufen zu lassen.

Des Weiteren können einige Batchsysteme zu bestimmten Zeitpunkten im Lauf einer Anwendung so genannte Checkpoints setzen, die es erlauben, im Falle einer Unterbrechung oder eines Fehlers, zu dem Zustand der Anwendung am letzten Checkpoint zurückzukehren und von hier aus erneut mit der Bearbeitung fortzufahren, so dass die Ergebnisse eines unter Umständen bis dahin sehr langen Programmlaufs nicht verloren gehen.

Beispiele für solche Batchsysteme sind das kostenlos erhältliche *Open Portable Batch System* (*OpenPBS*) [Port04], sowie dessen kommerzielle Variante *PBS Pro* [PBS 04], der *Load Leveler* von *IBM* [IBM 04], die *Load Sharing Facility* (*LSF*)-Produktreihe von *Platform Computing* [Plat04] und das frei verfügbare *Condor*, das im folgenden Abschnitt im Detail vorgestellt wird.

4 Workload-Management mit *Condor*

4.1 Überblick und Aufbau

Die *Condor*-Software wird seit 1985 an der *University of Wisconsin Madison* entwickelt und bietet neben den im vorigen Abschnitt beschriebenen Funktionen eines Batchsystems zusätzliche Möglichkeiten zur Lastverteilung und einer effizienten Verwendung der Ressourcen eines Clusters [Cond04][TiMI04]. Ein primäres Entwicklungsziel ist die Ausnutzung der brachliegenden Leistung von Arbeitsplatzrechnern zu Zeiten in denen diese nicht vom Anwender benutzt werden. Diese machen in der Regel einen erheblichen Anteil an der Lebensdauer eines Rechnersystems aus. So können auf diese Weise nach Aussagen der Entwickler an ihrem Institut mit 1000 Computern im täglichen Durchschnitt 650 CPU-Tage anderen Aufgaben zur Verfügung gestellt werden (siehe auch Abbildung 4). Bei einem bestehenden Rechnerpool kann auf diese Weise, vom zusätzlichen Administrationsaufwand und einem höheren Stromverbrauch abgesehen, ohne weitere Mehrkosten eine unter Umständen erhebliche Rechenkapazität erschlossen werden.

In einem *Condor*-System existieren vier verschiedene Typen von Rechnerknoten. Den größten Anteil machen in der Regel die für die Abarbeitung von Aufträgen zuständigen Rechenknoten aus. Darüber hinaus gibt es genau einen zentralen Management-Computer, der über alle notwendigen Informationen hinsichtlich der Struktur des Clusters verfügt und für die Aufgabenzuteilung zuständig ist, sowie beliebig viele Zugangsknoten, über die Benutzer Aufträge in das System eingeben können. Falls die in *Condor* bestehende Möglichkeit, zu verschiedenen Zeitpunkten während der Laufzeit einer Anwendung Sicherungspunkte (Checkpoints) anlegen zu lassen, genutzt werden soll, muss es außerdem einen Checkpoint-Server zur Verwaltung dieser Sicherungspunkte geben. Einem Computer des Clusters können auch mehrere dieser Aufgaben zugleich übertragen werden, es kann also beispielsweise eine Maschine sowohl Rechen- als auch Zugangsknoten und zentraler Managementrechner sein.

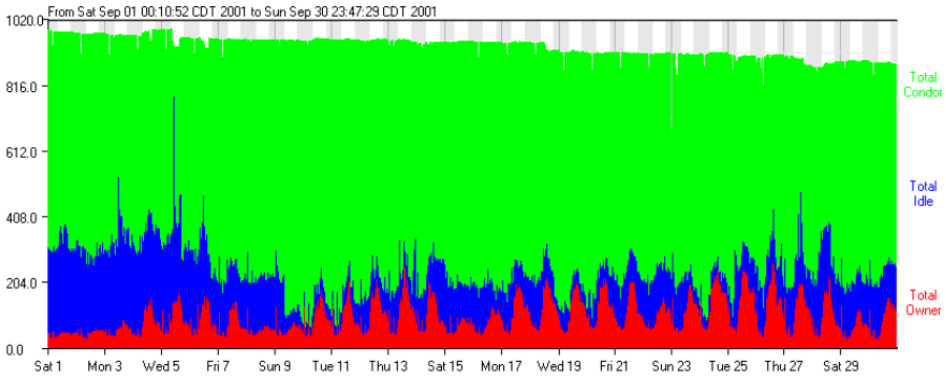


Abbildung 4: Anteile der von *Condor* genutzten Kapazität eines Rechnerpools über einen Monat [Cond04]

4.2 Die *Condor*-Universen

Für die Abarbeitung eines Programms im *Condor*-System kann eine Umgebung für die Ausführung gewählt werden, die als *Universe* bezeichnet wird und abhängig von der Art der Anwendung eine unterschiedlich große beziehungsweise angepasste Funktionalität bietet. Sequentielle Programme, die nicht im Quelltext oder als Objektcode vorhanden sind, werden im

Vanilla Universe ausgeführt, für das *Condor* die grundlegende Funktionalität eines Batchsystems bereitstellt.

Weitergehende Möglichkeiten bietet das *Standard Universe*, das allerdings erfordert, eine auszuführende Anwendung mit einer *Condor*-Bibliothek zu binden. Hier wird es nun möglich, den Checkpoint-Mechanismus zum Anlegen von Sicherungspunkten einzusetzen, was insbesondere bei sehr großen Programmlaufzeiten von großem Vorteil sein kann. Zudem kann auf diese Weise einem Benutzer, der an seinen Arbeitsplatzrechner, auf dem gerade eine *Condor*-Anwendung läuft, zurückkehrt, unmittelbar wieder die volle Computerleistung zur Verfügung gestellt werden, ohne dass der *Condor*-Auftrag dauerhaft unterbrochen werden müsste. Wird festgestellt, dass ein Anwender den Rechner wieder benutzt, so wird ein Sicherungspunkt angelegt und das Programm auf einen anderen freien Rechner migriert, auf dem es vom aktuellen Zustand aus weiterlaufen kann. Mit der *Condor*-Bibliothek ist es außerdem möglich, Systemaufrufe der Anwendung auf eine andere Maschine umzuleiten, falls diese auf dem Rechenknoten nicht erwünscht sind (zum Beispiel aus Sicherheitsüberlegungen). Da ein Rechner im *Standard Universe* weiterhin uneingeschränkt zur Verfügung steht und gut von fremden Programmen abgeschirmt werden kann, fällt es Benutzern unter Umständen leichter, den eigenen Computer einem *Condor*-System zur Verfügung zu stellen. Da die *Condor*-Bibliothek im unprivilegierten Modus arbeitet, gibt es allerdings ein paar Einschränkungen für Anwendungen im *Standard Universe*, falls der Checkpoint-Mechanismus verwendet werden soll. So dürfen einige Betriebssystemaufrufe wie *fork()* sowie *Kernel Threads* nicht verwendet werden und die Interprozesskommunikation ist beschränkt (die Benutzung von *pipes* und gemeinsamem Speicher ist nicht erlaubt).

Daneben gibt es noch einige weitere Universen für die Verwendung paralleler Programme, die mit Hilfe der Programmierbibliotheken *MPI* oder *PVM* geschrieben worden sind. *Java*-Anwendungen können im *Java Universe* gestartet werden, das über Informationen bezüglich der *Java Virtual Machines* im Cluster verfügt und genauere Angaben bei eventuell auftretenden Programmfehlern liefern kann, indem *Java*-Ausnahmen (*Exceptions*) protokolliert und dem Benutzer mitgeteilt werden.

4.3 Batchsystem und Auftragszuordnung

Die Auftragsbearbeitung ist in *Condor* als reine Stapelverarbeitung organisiert, also nicht interaktiv, dennoch können laufende Aufträge manuell abgebrochen und der Wiederanlauf veranlasst werden. Über den Stand der Abarbeitung kann sich der Benutzer per Email oder grafischer Oberfläche informieren lassen, darüber hinaus werden vom System Protokolldateien geschrieben. Aufträge können mit Prioritäten versehen und entsprechend eventuell bestehender Abhängigkeiten als Liste oder als Graph angeordnet eingegeben werden.

Bei der Auftragszuordnung fungiert das *Condor*-System als ein Makler, der die Anforderungen zu bearbeitender Aufträge mit zur Verfügung stehenden Ressourcen abgleicht und bestmöglich verknüpft. Der Besitzer (beziehungsweise Administrator) eines Rechners kann sehr weit reichend bestimmen, in welchem Maße und für welche Aufgaben seine Maschine vom *Condor*-System eingesetzt wird. Er charakterisiert den Computer einerseits hinsichtlich der Hardwareausstattung, des Betriebssystems und der installierten Software und legt andererseits Benutzbarkeitsregeln und -zeiträume fest. So kann er beispielsweise Aufträgen, die aus seiner eigenen Abteilung kommen, Vorrang gewähren oder die Benutzung durch *Condor* an bestimmten Wochentagen untersagen. Ein Benutzer legt für seinen Auftrag ein Anforderungsprofil an, das unter anderem den Prozessortyp, benötigte Dateien und erforderlichen Speicher umfassen kann, wobei es möglich ist, sowohl eine gewünschte als auch eine zwingend benötigte Ressourcenanforderung zu definieren. Da das *Condor*-System vor der Programmausführung benötigte Dateien auf den Rechenknoten kopieren kann, muss der Rechnerpool

weder über ein verteiltes Dateisystem verfügen, noch muss auf den verwendeten Computern eine Benutzerkennung des Auftraggebers vorhanden sein.

4.4 Grid-Computing mit *Condor*

Mehrere Computerpools, auf denen *Condor* installiert ist, können ihre Ressourcen vereinigen und so den Benutzern eine erweiterte Rechenkapazität zur Verfügung stellen. Mit dieser als *Flocking* bezeichneten Funktion *Condor*s bleiben alle erweiterten Möglichkeiten des *Standard Universe*, wie zum Beispiel der Checkpoint-Mechanismus oder entfernte Systemaufrufe, erhalten und können über das gesamte Rechnerangebot hinweg verwendet werden.

Darüber hinaus können auch externe Ressourcen, die nicht mit *Condor* betrieben werden, von einer *Condor*-Maschine aus zugänglich gemacht werden, sofern sie eine *Globus*-Schnittstelle anbieten. *Globus* ist eine quelloffene Implementierung Grid-bezogener Protokolle und *Middleware*-Anwendungen, die unter anderem verschiedene Funktionen zur entfernten Programmausführung und eine Sicherheitsinfrastruktur bieten. Mit der Erweiterung *Condor-G* können auf diese Weise über eine einheitliche Oberfläche Aufträge an externe Cluster weitergeleitet werden, die mit anderen Batchsystemen wie *PBS* oder *LSF* betrieben werden. Falls auf die erweiterten Funktionen des *Standard Universe* nicht verzichtet werden soll, können über den *Glidein*-Mechanismus externe Ressourcen auch zeitweilig dem lokalen *Condor*-Pool hinzugefügt werden, in dem dann die volle Funktionalität genutzt werden kann. Neu in den Pool aufgenommene externe Ressourcen sind unmittelbar für alle Nutzer sichtbar, können aber nur von dem Anwender gebraucht werden, der sie hinzugefügt hat. Um einen fremden Rechnerverbund einzubinden, werden die erforderlichen Dateien vom *Condor*-System, falls nicht vorhanden, übertragen, und ein *Condor*-Dienst gestartet, der alle Funktionen des *Standard Universe* bereitstellt. Abbildung 5 zeigt einen *Condor*-Pool, der durch die *Flocking*-Funktion mit einem weiteren *Condor*-Verbund (*„Friendly Condor Pool“*) und über die *Globus*-Schnittstelle mit anderen Clustern gekoppelt ist. Diese können, wie dargestellt, sowohl mit *Condor* als auch mit anderen Batchsystemen wie *PBS* oder *LSF* betrieben werden.

5 Linuxcluster am Rechenzentrum

5.1 Überblick über Linuxcluster am RZ

Um Erfahrungen mit dem Aufbau und Betrieb von Linuxclustern zu sammeln, betreibt das Rechenzentrum der Universität Karlsruhe einen experimentellen Linuxcluster, der ausschließlich zu Testzwecken und nicht für den Benutzerbetrieb zur Verfügung steht [Rech04c]. Dabei wurde bewusst auf handelsübliche Standardkomponenten zurückgegriffen und der Einbau teurer Spezialhardware vermieden (*„Commodity off the shelf“*). Insgesamt besteht der experimentelle Linuxcluster des RZ aus zehn Rechnerknoten, von denen einer als Kontrollrechner und Dateiserver dient. Acht Maschinen sind mit *AMD Athlon MP 1900+*-Prozessoren ausgestattet, wobei vier Computer als Doppelprozessorknoten ausgeführt sind. Ein weiterer Rechnerknoten verfügt über zwei *AMD Opteron*-Prozessoren, so dass auch Anwendungen auf dieser 64-Bit-Plattform getestet werden können. Eine solche Konfiguration ermöglicht einen großen Spielraum für Tests, da Programme sowohl auf Einzel- als auch auf Doppelprozessorknoten und im gemischten Betrieb ausgeführt werden können. Zusätzlich zu einer bestehenden *Red Hat*-Installation ist die Einrichtung weiterer Linux-Distributionen geplant.

Zur Zeit im Aufbau befindet sich ein neuer Höchstleistungsrechner, der in seiner letzten Ausbaustufe mit 1200 *Intel Itanium 2*-Prozessoren eine Spitzenleistung von $11*10^{12}$ Gleitkommaoperationen in der Sekunde (*TFlop/s*) bewältigen können soll und über einen Hauptspeicher

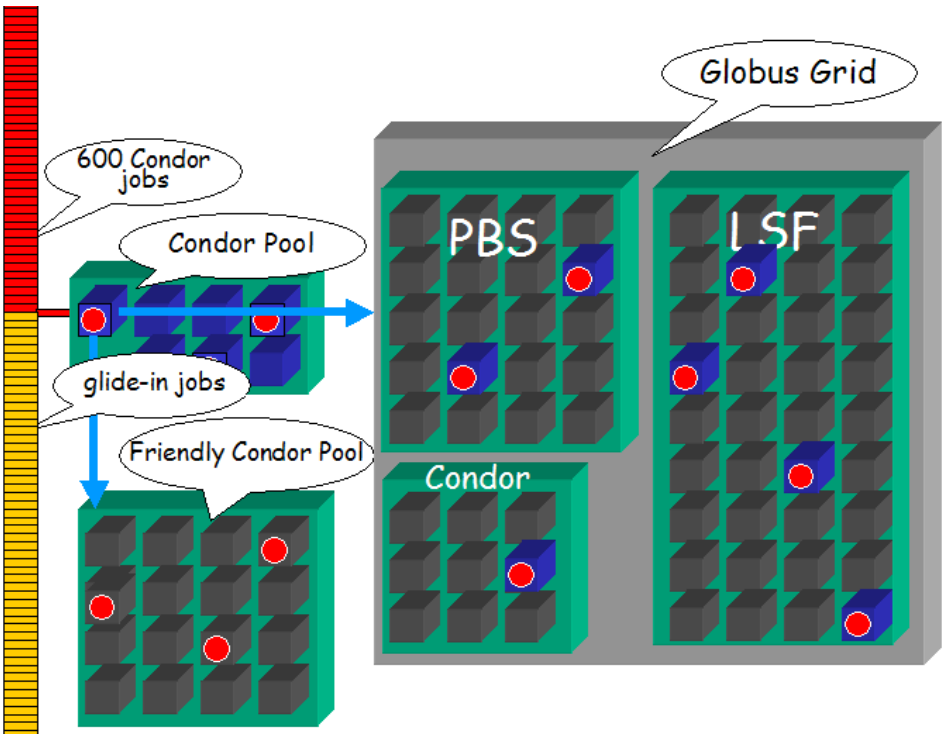


Abbildung 5: Schematische Darstellung verschiedener Möglichkeiten des *Grid Computing* mit *Condor* [Cond04]

von insgesamt sieben TByte verfügen wird [Rech04b]. Dieser von *Hewlett-Packard* gelieferte *HP XC6000*-Cluster wird verschiedene Knotentypen mit jeweils 2, 4 oder 16 Prozessoren bereitstellen, so dass auch Anwendungen, die nach dem Prinzip des gemeinsamen Speicherzugriffs parallelisiert sind, auf einzelnen Knoten ausgeführt werden können. Das skalierbare, parallele *Lustre*-Dateisystem ist für den Einsatz mit sehr vielen Clients und für hohe Bandbreiten konzipiert und wird auf diesem Rechner für die Verwaltung von 40 TByte globalem Plattenplatz verwendet [Clus04]. Die Kommunikation zwischen den Rechnerknoten erfolgt über ein *Quadrics QSNet II*-Netzwerk, das auf der Programmierebene der nachrichtenbasierten *MPI*-Schnittstelle eine Bandbreite von 800 MByte in der Sekunde bei einer Latenz von drei Mikrosekunden zur Verfügung stellt. Als Betriebsumgebung wird *HP XC Linux* eingesetzt, das, basierend auf der *Red Hat Advanced Server*-Distribution, um einige Komponenten, wie zum Beispiel *Quadrics*-Treiber, erweitert wurde.

5.2 Anwendungsgebiete und Nutzergruppen

Das zur Zeit im Aufbau befindliche *HP XC6000*-Clustersystem am Rechenzentrum wird Forschern und kommerziellen Nutzern landes- und bundesweit zur Verfügung stehen und soll auch als Teil von zusammengeschalteten Rechnersystemen (*Grid Computing*) zum Einsatz kommen. Die Nutzung des Systems durch die Industrie erfolgt über eine institutionelle Kooperation industrieller und universitärer Partner, zu deren Zweck im Jahr 1995 die hww GmbH (Höchstleistungsrechner für Wissenschaft und Wirtschaft Betriebsgesellschaft mbH) gegründet wurde, in der die Industrie und der öffentliche Bereich (die Universitäten Heidelberg, Karlsruhe und Stuttgart sowie das Land Baden-Württemberg) zu gleichen Teilen vertreten sind. Durch den gemeinsamen Betrieb der verschiedenen Computersysteme der Anteilseigner können Anwendern verschiedene Rechnerarchitekturen zur Verfügung gestellt und Synergieeffekte bei der Beschaffung und Nutzung erreicht werden. Der Vertrieb bei Indus-

triekunden erfolgt über die organisatorischen Strukturen der Firma *T-Systems*, die ebenfalls an der hww GmbH beteiligt ist [T-Sy04]. Die verwendete Rechenleistung wird nach CPU-Stunden, evtl. unter Einbeziehung des Speicherverbrauchs, abgerechnet. Das am Rechenzentrum der Universität Karlsruhe entstehende System eignet sich dabei besonders für Anwendungen des Höchstleistungsrechnens, die vom vergleichsweise großen Datencache der *Itanium 2*-Prozessoren profitieren können.

Auch auf Institutsebene werden an der Universität Karlsruhe verstärkt Linuxcluster als kostengünstige Parallelrechner eingesetzt. Insgesamt werden zur Zeit etwa zehn dieser Computersysteme mit bis zu 120 Prozessoren an verschiedenen Einrichtungen der Universität betrieben und beispielsweise für Anwendungen in den Bereichen der numerischen Strömungssimulation, der Teilchen- und Molekülphysik, der Quantenchemie oder auch der Weiterentwicklung des Cluster-Computing verwendet [Rech04a]. Darüber hinaus gibt es Überlegungen von Instituten, die Clustersysteme anschaffen möchten, ihre Ressourcen gemeinsam zum Kauf eines leistungsstärkeren Rechners einzusetzen, der dann anteilig benutzt werden kann. Obwohl sich aus Sicht eines Instituts auf diese Weise größere Problemklassen bearbeiten lassen, ergeben sich durch die gemeinsame Nutzung natürlich Einschränkungen hinsichtlich der Verfügbarkeit und Verwaltung eines solchen Systems. Oft besitzen einzelne Institute allerdings auch nicht die Infrastruktur, beispielsweise im Hinblick auf die Stromversorgung und die Klimaregulierung, zur Aufstellung eines größeren Clustersystems, so dass hier eventuell auf die Ressourcen des Rechenzentrums zurückgegriffen werden müsste. Das Rechenzentrum berät die Einrichtungen zudem bei der Beschaffung eines Rechnersystems und gibt Hilfestellung bei der späteren Installation und Administration.

5.3 Herausforderungen bei Planung, Aufbau und Betrieb eines Clustersystems

Der Installation eines größeren Clustersystems geht in der Regel eine längere Planungs- und Testphase voraus, in der die Anforderungen festgelegt und die Bedingungen mit den Herstellern ausgehandelt werden müssen. Die späteren Anwendungsgebiete sind entscheidend für die Wahl einer geeigneten Hardware, da beispielsweise numerische Simulationen eine hohe Leistung im Bereich der Gleitkommaoperationen benötigen, während Datenbankanwendungen vor allem von schnellen Ganzzahlberechnungen profitieren und der verwendete Prozessortyp entsprechend ausgesucht werden sollte. Wird ein Cluster, wie beim Höchstleistungsrechnen üblich, als Komplettsystem von einem Hersteller bezogen und nicht im Eigenbau aus Komponenten gefertigt, so werden Anforderungen des zu liefernden Systems vertraglich festgelegt, die nach der Installation durch *Benchmark*-Tests, zum Beispiel hinsichtlich der Latenz des Netzwerks, überprüft werden können. Durch die oft langen Vorlaufzeiten ist die genaue Spezifikation eines Rechners oft schwierig, da bis zum Aufbau schon neue Generationen verschiedener Bauteile verfügbar sein können. Bei einer öffentlichen Anschaffung in einer Größenordnung wie der des *HP XC6000*-Clusters, die einen Umfang von circa elf Millionen Euro hat, ist zudem eine europaweite Ausschreibung erforderlich. Dieses System wird in mehreren Stufen am Rechenzentrum installiert und schließlich im ersten Quartal 2006 seinen vollen Ausbau erreicht haben.

Die während des Betriebs notwendige technische Unterstützung ist bei einer solchen Computeranlage in der Regel im Kaufvertrag vereinbart und wird vom Hersteller übernommen, sofern nach dessen Vorgaben regelmäßige Aktualisierungen des Systems vorgenommen werden. Die Anwendungsportierung auf ein neues System stellt in der Regel kein größeres Hindernis dar, da vor allem bei Linux-Systemen normalerweise offene Standards verwendet werden.

5.4 Poolrechner im Clusterverbund

An einigen Einrichtungen der Universität Karlsruhe besteht ein großer Bedarf an Rechenleistung für trivial parallelisierbare Probleme, die keine oder wenig Kommunikation zwischen den Rechnerknoten eines Clusters erfordern. So ist zum Beispiel das Institut für experimentelle Kernphysik (IEKP) der Fakultät für Physik [Inst04] an verschiedenen Experimenten der Hochenergiephysik beteiligt, die eine Auswertung großer Datenmengen erfordern. Beispielsweise werden für Experimente mit dem *Large Hadron Collider* (*LHC*) am europäischen Forschungszentrum *CERN* jährlich einige PetaByte an Daten zu bewältigen sein, die an mehreren regionalen Rechenzentren in den beteiligten Nationen ausgewertet werden sollen [Grid04]. Standort des deutschen Regionalrechenzentrums ist das Forschungszentrum Karlsruhe (FZK), an dessen Computer-Ressourcen auch das IEKP angeschlossen ist. Eine nahe liegende Möglichkeit zur Steigerung der für solche oder ähnliche Aufgaben zur Verfügung stehenden Rechenleistung ist die Nutzbarmachung von nicht benötigten Zyklen der mehreren hundert Arbeitplatzcomputer in den Pools des Rechenzentrums. Aufgrund des langjährigen erfolgreichen Einsatzes der *Condor*-Software an zahlreichen Institutionen weltweit [Cond05] soll in den kommenden Monaten aus den verfügbaren Maschinen ein *Condor*-Cluster aufgebaut werden. Dabei werden wegen der in Abschnitt 4 beschriebenen Möglichkeiten dieses Systems, das dem Anwender der *Condor*-Software auch eine direkte Verwendung seiner Programme ohne Anpassungen erlaubt und den Benutzer vor dem Computer kaum einschränkt, nur wenig Probleme auf beiden Seiten erwartet.

5.5 Grid Computing

Das Beispiel des im vorigen Abschnitt erwähnten *LHC*-Experiments zeigt, dass für viele Problemstellungen auch die Leistungsfähigkeit vieler zu einem Cluster verbundener Rechner nicht ausreicht. Eine nächste Stufe zur Erhöhung der Rechenleistung besteht nun darin, verschiedene Hochleistungsrechner, die unter Umständen sehr weit voneinander entfernt installiert sein können, wiederum zu einer gemeinsamen Rechenressource zusammenzuschalten. Auch das Rechenzentrum der Universität Karlsruhe ist mit seinen Computersystemen an verschiedenen Projekten in diesem als *Grid Computing* bezeichneten Bereich beteiligt. So wurde, um die personellen und finanziellen Kräfte im Bereich des Höchstleistungsrechnens in Baden-Württemberg zu bündeln, im Juni 2004 das Höchstleistungsrechner-Kompetenzzentrum Baden-Württemberg (hkz-bw) [Höch04] gegründet, das im Kern vom Höchstleistungsrechenzentrum der Universität Stuttgart (HLRS) [HLRS04] und dem Scientific Supercomputing Center der Universität Karlsruhe (SSCK) [Rech04d] gebildet wird und außerdem ein Bestandteil der vom Bundesministerium für Bildung und Forschung (BMBF) geförderten D-Grid-Initiative [D-Gr04] ist. Zur Infrastruktur des Kompetenzzentrums gehört neben dem *HP XC6000*-Linuxcluster ein Vektorrechner der obersten Leistungsklasse des HLRS. Einerseits kann durch die unterschiedliche Architektur der beiden Systeme ein breites Anwendungsspektrum abgedeckt werden, andererseits können sie auch als ein verteiltes System betrieben werden. Die Kopplung erfolgt über das BelWü-Landesforschungsnetz, das auf eine Bandbreite von 40 GBit pro Sekunde ausgebaut werden soll, während ein gemeinsames Dateisystem aus Benutzersicht für einen einheitlichen Datenraum sorgt.

Darüber hinaus wurde bereits 1996 von der Universität und dem Forschungszentrum Karlsruhe das Virtuelle Rechenzentrum Karlsruhe gegründet, dessen Ressourcen über eine Datenleitung mit einer Bandbreite von zehn GBit in der Sekunde gekoppelt werden können [Rech04e].

6 Zusammenfassung und Ausblick

Clustersysteme haben sich in vielen Computeranwendungen durchgesetzt, da sie gegenüber anderen Rechnerarchitekturen einige wesentliche Vorteile besitzen. Im Bereich des *High Throughput Computing* überzeugen sie vor allem durch die Möglichkeit eines einfachen und sehr flexiblen Aufbaus aus Standardkomponenten, die auch heterogene Umgebungen ohne weiteres zulässt. Im Höchstleistungsrechnen entwickeln sich Clustersysteme zur dominierenden Architekur, da Leistungssteigerungen einzelner Maschinen schwer zu erreichen sind und neue Supercomputer am einfachsten durch die Kopplung mehrerer Rechner gebaut werden können. Die zukünftigen Herausforderungen liegen vor allem in einer schnelleren Verbindung der einzelnen Maschinen untereinander, da auch moderne Verbindungstechnik mit hinreichenden Leistungsdaten durch eine zu langsame Anbindung des Netzwerkadapters an den Prozessor nicht voll ausgenutzt werden kann. Neue Standards für eine schnellere Anbindung peripherer Geräte oder auch die direkte Kopplung von Netzwerkanschlüssen an den Speicherbus eines Computers befinden sich zwar in der Entwicklung, sind zum gegenwärtigen Zeitpunkt aber noch nicht verfügbar.

Bereits auf den ersten Clustern des *Beowulf*-Projekts (siehe auch Abschnitt 1.1) wurde Linux als Betriebssystem eingesetzt und hat sich bis heute erfolgreich auf diesen Systemen behauptet. Die häufig genannten Argumente, die für den Einsatz von Linux angeführt werden, gelten im Wesentlichen auch für den Clusterbetrieb. So liegen für den Softwareentwickler und Systemadministrator die größten Vorteile sicherlich in der freien Verfügbarkeit eines modernen und stets weiterentwickelten Betriebssystems für die gängigsten Plattformen. Es gibt ein großes Angebot an zuverlässigen und robusten Werkzeugen und Programmierbibliotheken, sowie ein umfangreiches Reservoir verschiedener Möglichkeiten der technischen Unterstützung durch andere Entwickler und Benutzer. Auch die Konfiguration und Modifikation von Komponenten des Betriebssystems ist bei Linux für versierte Anwender einfacher oder zumindest flexibler möglich als bei proprietären Systemen, deren Quelltexte und Programmierschnittstellen in der Regel nicht vollständig einsehbar sind. Computerhersteller können die von Dritten geleistete Entwicklungsarbeit ausnutzen und ihre Ressourcen darauf konzentrieren, auf dem bestehenden Linux basierende Varianten herauszugeben, die optimal mit der eigenen Hardware funktionieren.

Andererseits können die in der *Open Source*-Gemeinde üblichen Softwarelizenzen den Hersteller unter Umständen dazu zwingen, auch selbstentwickelte Erweiterungen unter einer solchen Lizenz zu veröffentlichen und den Quelltext verfügbar zu machen, was eventuell, zum Beispiel unter Marketing- und Konkurrenzgesichtspunkten, nicht erwünscht ist. In diesem Fall liegt es nahe, auf eine andere Software auszuweichen, so dass es auch in Zukunft sicherlich mehrere Alternativen bei der Wahl eines geeigneten Clusterbetriebssystems geben wird.

Die momentan wachsende Verbreitung von Linux als Betriebssystem wird sich sicherlich auch im Bereich des Cluster-Computing am Rechenzentrum der Universität Karlsruhe fortsetzen. Schon jetzt sind die Rechnerarbeitsplätze für Studenten weitgehend mit diesem Betriebssystem ausgestattet, das oft wahlweise als Alternative zu *Microsoft Windows* gestartet werden kann. In nächster Zukunft sollen diese Ressourcen auch im Verbund durch den Einsatz der *Condor*-Software besser ausgenutzt werden können. Die Wahl des neuen Höchstleistungsrechners zeigt, dass auch in diesem Segment Linux der Vorzug über die verschiedenen herstellerabhängigen *UNIX*-Derivate gegeben wird, obwohl die Anschaffungskosten für die Betriebssoftware hier im Vergleich zu alternativen Systemen sicherlich nicht derart stark ins Gewicht fallen. Auch wenn die Wartungsarbeiten überwiegend vom Hersteller wahrgenommen werden, profitieren Benutzer und Administratoren oftmals von einer einheitlichen Betriebssystemumgebung auf Arbeitsplatzrechnern und dem Hochleistungscluster sowie von einer einfachen Übertragbarkeit ihrer Software auf das neue System.

Literatur

[Beow05] Beowulf.org: Overview – History. http://www.beowulf.org/overview/history.html. Website, Jan 2005.

[Cfen04] Cfengine. http://www.cfengine.org/. Projekt-Website, Dez 2004.

[Clus04] Cluster File Systems, Inc. - Lustre. http://www.lustre.org/. Website, Dez 2004.

[Cond04] Condor Project Homepage. http://www.cs.wisc.edu/condor/. Website, Dez 2004.

[Cond05] Condor World Map. http://www.cs.wisc.edu/condor/map/. Website, Jan 2005.

[CT P05] CT Project: Fact Sheet. http://ct.gsfc.nasa.gov/essfactsheet.html. Website, Jan 2005.

[D-Gr04] D-Grid. http://www.d-grid.de/. Website, Dez 2004.

[Grid04] GridKa - Grid Computing Center Karlsruhe. http://www.gridka.de/. Website, Dez 2004.

[Höch04] Höchstleistungsrechner-Kompetenzzentrum Baden-Württemberg. http://www.hkz-bw.de/. Website, Dez 2004.

[HLRS04] HLRS - Höchstleistungsrechenzentrum Stuttgart. http://www.hlrs.de/. Website, Dez 2004.

[IBM 04] IBM LoadLeveler. http://publib.boulder.ibm.com/clresctr/windows/public/llbooks.html. Website, Dez 2004.

[Infi04] InfiniBand Trade Association. http://www.infinibandta.org/. Website, Dez 2004.

[Inst04] Institut für Experimentelle Kernphysik. http://www-ekp.physik.uni-karlsruhe.de/. Website, Dez 2004.

[Myri04] Myricom, Inc. http://www.myri.com/. Firmenwebsite, Dez 2004.

[NCSA04] NCSA Tungsten Cluster Downloads. http://www.ncsa.uiuc.edu/Divisions/PublicAffairs/LinuxCluster/tungsten.html. Website, Dez 2004.

[PBS 04] PBS Pro. http://www.pbspro.com/. Website, Dez 2004.

[Plat04] Platform LSF Family. http://www.platform.com/products/LSFfamily/. Website, Dez 2004.

[Port04] Portable Batch System. http://www.openpbs.org/. Website, Dez 2004.

[Quad04] Quadrics Ltd. http://www.quadrics.com/. Firmenwebsite, Dez 2004.

[RaVo05] R. Rademacher und M. Vogel. Linux-Cluster vergleicht Genomsequenzen. *Computer Zeitung* (1/2), Jan 2005, S. 22.

[Rech04a] Rechenzentrum Uni KA - Arbeitskreis Linux Cluster. http://www.rz.uni-karlsruhe.de/dienste/ak-linux-cluster.php. Website, Dez 2004.

[Rech04b] Rechenzentrum Uni KA - HP XC6000. http://www.rz.uni-karlsruhe.de/ssc/4805.php. Website, Dez 2004.

[Rech04c] Rechenzentrum Uni KA - Linux Cluster. http://www.rz.uni-karlsruhe.de/dienste/2346.php. Website, Dez 2004.

[Rech04d] Rechenzentrum Uni KA - Scientific Supercomputing. http://www.rz.uni-karlsruhe.de/ssc/ssc.php. Website, Dez 2004.

[Rech04e] Rechenzentrum Uni KA - Virtuelles Rechenzentrum Karlsruhe. http://www.rz.uni-karlsruhe.de/projekte/3370.php. Website, Dez 2004.

[Red 04] Red Hat Enterprise Linux. http://www.redhat.com/software/rhel/. Website, Dez 2004.

[Rock04] Rocks Cluster Distribution. http://www.rocksclusters.org/Rocks/. Projekt-Website, Dez 2004.

[SCI 04] SCI - Scalable Coherent Interface. http://hsi.web.cern.ch/HSI/sci/sci.html. Website, Dez 2004.

[SETI04] SETI@home. http://setiathome.ssl.berkeley.edu/. Website, Dez 2004.

[SUSE04] SUSE LINUX Enterprise Server. http://www.novell.com/products/linuxenterpriseserver/. Website, Dez 2004.

[T-Sy04] T-Systems hpcPortal. http://www.hpcportal.de/. Website, Dez 2004.

[The 05] The FreeBSD Project. http://www.freebsd.org/. Website, Jan 2005.

[TiMI04] W. F. Tichy, Th. Moschny und F. Isaila. Rechnerbündel WS 2004/05. Folien zur Vorlesung, Okt 2004.

[TOP504] TOP500 List for November 2004. http://www.top500.org/lists/2004/11/. Website, Dez 2004.

[Virg04] Virginia Tech Terascale Computing Facility. http://www.tcf.vt.edu/systemX.html. Website, Dez 2004.

Abbildungsverzeichnis

Distributed File Systems

Matthias Schmitt

Kurzfassung

Verteilte Dateisysteme (Distributed File Systems, DFS) erlauben es, mit Dateien zu arbeiten, die auf anderen, über ein Netzwerk erreichbaren Rechnern gespeichert sind. Da mitunter mehrere Anwendungen gleichzeitig auf die Dateien zugreifen, Betriebssysteme mit unterschiedlichen Datei-Spezifikationen im Einsatz sind, und die Verwendung für den Nutzer so transparent wie möglich erscheinen soll, müssen sich DFS diversen Anforderungen stellen. Locking, Caching und ein vielfältig einsetzbares Dateimodell sind der Schlüssel zu oben genannten Problemen, und in gebräuchlichen DFS wie dem Network File System (NFS) und dem Common Internet File System (CIFS) implementiert. Hersteller haben DFS-Software und spezielle Hardware zu so genannten Network-Attached-Storage-Produkten (NAS) gebündelt, um Unternehmen den Einsatz zu erleichtern. Neuere DFS-Entwicklungen wie Coda warten mit zahlreichen Ideen auf, z.B. für die Verwendung im Wireless-Lan.

1 Einleitung

Verteilte Dateisysteme befinden sich seit Anfang der 80er Jahre im Einsatz. Sie kombinieren die Vorteile von Dateisystemen mit den Möglichkeiten moderner, leistungsfähiger Netzwerk-Infrastruktur. Sie sollen eine einfache, zentralisierte Administration gewährleisten, und sind das Rückgrat vieler Rechenzentren, ob in der privaten Wirtschaft oder in öffentlichen Einrichtungen. Das Augenmerk dieser Arbeit richtet sich hierbei auf die zwei im täglichen Einsatz bewährten verteilten Dateisysteme NFS und CIFS. Klassische Netzwerktechnologien sind Ethernet und Token Ring. Der Bedarf an immer schnelleren Verbindungen, größeren Datenvolumen und weiteren Entfernungen hat zur Entwicklung neuerer Netzwerktechnologien geführt, wie z.B. Gigabit-Ethernet. Internet-Protokolle haben dazu beigetragen, Kommunikation zwischen unterschiedlichsten Teilnehmern zu ermöglichen, und die Möglichkeit geschaffen, kostengünstig kleinere Netzwerke an weltweit verteilten Standorten miteinander zu verbinden. Gleichzeitig zur Entwicklung neuer und schnellerer Netzwerktechnologien steigt die nachgefragte, und auch verfügbare, Kapazität von Speichermedien. DFS kombinieren Netzwerktechnologien mit Speichermedien.

2 Dateisysteme

Damit die Daten, die auf den Plattenlaufwerken landen, wieder aufgefunden werden können, bedarf es einer weiteren Abstraktionsebene. Die meisten Anwendungssysteme nutzen aus diesem Grund nicht direkt die Blöcke, die von den Festplatten-Laufwerken als Speichereinheit zur Verfügung gestellt werden. Vielmehr arbeiten sie mit Dateien, die in Verzeichnishierarchien geordnet sind. Die meisten Betriebssysteme schalten eine Dateisystem-Schicht zwischen Anwendungs-Programm und Block-Geräte-Ebene.

2.1 Speichermedien

Die gebräuchlichsten Speichermedien sind zur Zeit Festplatten-Laufwerke, die Daten zuverlässig speichern und auf die schnell zugegriffen werden kann. Die Daten werden in Form von Blöcken zwischen Rechner und Laufwerk ausgetauscht, deren Größe in der Dimension von 4kB liegen. Neben den Festplatten spielen Bandlaufwerke eine bedeutende Rolle was die Archivierung von Daten anbelangt. Zur Anbindung der Festplatten an die einzelnen Server hat sich im professionellen Umfeld SCSI durchgesetzt. SCSI wurde im Laufe der Zeit mehrfach erweitert und an gestiegene Ansprüche in den Bereichen Geschwindigkeit und Erweiterbarkeit angepasst. Aktuell maximale Kabellängen belaufen sich auf 12 (LVD) bzw. 25 (HVD) Meter, Gerätezahl auf 16 und mit Ultra320 SCSI lässt sich eine Übertragungsrate von 320 MByte/s erreichen.

2.2 SAN

Für die in Rechenzentren nachgefragten Datenmengen wird eine Vielzahl an Festplatten benötigt. Werden die Festplatten direkt in die Servern verbaut, so lassen sie sich nur umständlich verwalten. Ist der freie Speicherplatz einer Platte erschöpft, müssen neue Platten an diesem Server in Betrieb genommen werden, obwohl in anderen Servern evtl. noch genügend ungenutzter Platz zur Verfügung steht. Erschwerend kommt hinzu, das schon aus Platzgründen nur eine begrenzte Anzahl von Platten eingebaut und mit den gewohnten Protokollen wie SCSI an einen Controller nicht beliebig viele Laufwerke angeschlossen werden können, meist 8 oder 16, je nach eingesetzter Technologie. Hier kommen Speichernetze (Storage Area Networks, SAN) ins Spiel. Die Festplatten werden über ein Speichernetz mit den einzelnen Servern verbunden (Abbildung 1). Sie lassen sich dann zentral verwalten und erweitern, und beliebig an die einzelnen Server verteilen. Die größeren Entfernungen zwischen Platte und Server sowie die hinzukommende Netzwerkfunktionalität bedingt neue Übertragungstechniken, wie z.B. FibreChannel, iSCSI und Infiniband. Eine solche, reine Zentralisierung von Platten wird JBOD (Just a Bunch Of Disks) genannt. Der Contoller eines JBODs kann die Platten den Rechnern zuordnen, und leere Platten vorrätig halten. Es gibt aber auch SAN-Systeme, die mehr Aufgaben übernehmen. Sie können RAID-Funktionalität anbieten, um Performanz und Datensicherheit zu erhöhen. Die angeschlossenen Server arbeiten dann mit virtuellen Platten, die eigentliche Speicherung der Daten erfolgt transparent. Andere Aufgaben, die von intelligenten Controllern durchgeführt werden können, sind das schnelle Kopieren ganzer virtueller Platten und Datenabgleich mit weiteren Systemen um eine höhere Sicherheit zu gewährleisten.

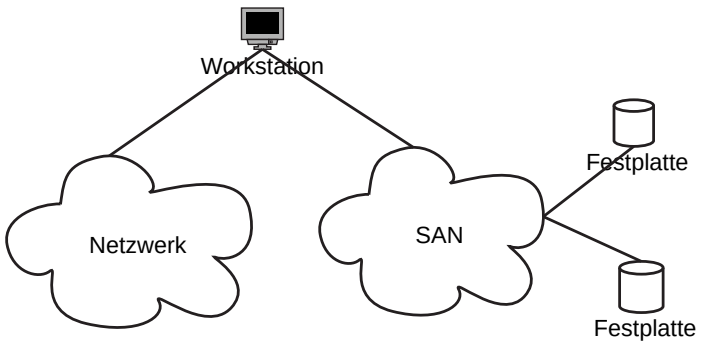


Abbildung 1: SAN

2.3 Aufgaben von Dateisystemen

Dateisysteme ermöglichen es, zusammengehörige Daten in Dateien zu speichern, und in Verzeichnishierarchien zu ordnen. Dateisysteme sollen einen schellen Zugriff auf die Daten erlauben. Sie verwenden einen Index, um nicht bei jedem Zugriff den gesamten Laufwerks- bzw. Partitionsinhalt durchsuchen zu müssen. Dateisysteme verknüpfen Metadaten mit den eigentlichen, in Dateien abgelegten Nutzdaten.

2.4 Rechtemanagement

In Mehrbenutzersystemen stellen Dateisysteme Möglichkeiten zur Verfügung, den Zugriff auf Dateien zu beschränken. Klassische Unix-Dateisystem kontrollieren die Erlaubnis zu lesen, zu schreiben, ein Programm auszuführen oder den Inhalt eines Verzeichnisses anzuzeigen, und lassen eine Unterscheidung in Besitzer, Gruppen, und übrige Benutzer zu. Neuer sind Access Controll Lists (ACLs), die eine detailliertere Vergabe von Rechten ermöglichen. Das Überwachen dieser Einschränkungen ist Aufgabe des Betriebssystems. Es muss die Benutzer und bei Zugriff auf das Dateisystem überprüfen, ob die notwendigen Rechte vorliegen.

2.5 Weitere Features

Damit sich im Multiuser-Betrieb nicht mehrere Nutzer beim Arbeiten mit der selben Datei in die Quere kommen, besteht in modernen Betriebssystemen die Möglichkeit, Dateien zu locken. Die Anwendung, die den Lock erhält, hat exklusiven Zugriff auf die Datei, die übrigen müssen auf die Aufhebung des Locks warten. Dateisysteme sollen im Falle einer Störung, z.B. eines Stromausfalls, möglichst schnell wieder Betriebsbereit sein und dabei die konsistente Speicherung von Daten gewährleisten. Damit angefangene, und noch nicht beendete Schreibaktionen rückgängig gemacht oder vervollständigt werden können, führen Journaling-Dateisysteme wie Ext3, JFS, oder XFS ein Log. Verschlüsseln und automatischen Komprimieren sind weitere Fähigkeiten, die von einzelnen Dateisystemen angeboten werden.

2.6 Die Virtual-File-System-Schicht

Die Anwendungsprogramme arbeiten mit Dateien, indem sie Routinen des Betriebssystems aufrufen. Diese Routinen sind in der Virtual-File-System-Schicht (VFS-Schicht) angesiedelt und umfassen Funktionen wie das Öffnen und Schließen einer Datei, das Springen an bestimmte Stellen, das Lesen und Schreiben. Entsprechende Routinen für das Betrachten von Verzeichnisinhalten und setzen bestimmter Datei-Attribute sind ebenso vorhanden. Analoge Mechanismen, wenn auch unter verschiedenen Namen (hier wird die Unix-Terminologie verwendet) und mit teilweise variierendem Funktionsumfang existieren in den meisten gebräuchlichen Betriebssystemen. Die VFS-Schicht delegiert die Aufgaben an die jeweiligen Dateisystemtreiber, die über entsprechende Gegenstücke zu diesen Routinen verfügen, bzw. sie emulieren. Diese Dateisystemtreiber kommunizieren nun direkt mit der Blockgeräteebene (Abbildung 2). Der Zwischenschritt einer VFS-Schicht erlaubt es, den darunter liegenden Dateisystemtreiber auszutauschen und somit verschiedene, dem Verwendungszweck angebrachte Dateisysteme zu verwenden.

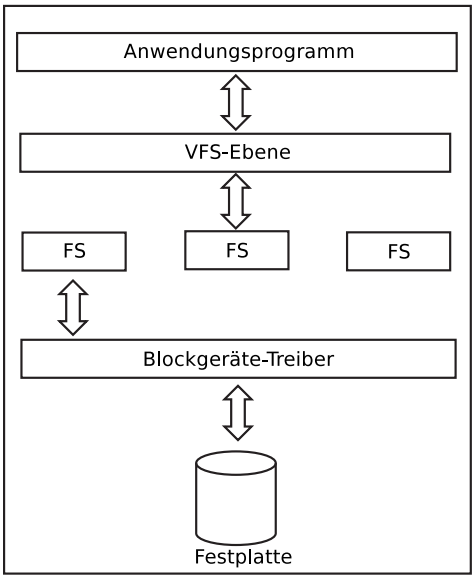


Abbildung 2: VFS-Architektur

3 Verteilte Dateisysteme (DFS)

3.1 Einordnung

Im Gegensatz zum SAN, bei dem an das Netzwerk angeschlossene Datenträger auf Blockeben angesprochen werden, und die gleichen Dateisysteme wie im nicht verteilten Fall zum Einsatz kommen, sind DFS direkt für eine verteilte Datenhaltung vorgesehen. DFS delegieren die eigentliche Speicherung der Daten an konventionelle Dateisysteme, und stellen verschieden Dienste bereit, über die Benutzer authentifiziert, angebotenen FS-Hierarchien aufgefunden und der gleichzeitige Zugriff durch viele Benutzer ermöglicht werden.

3.2 Architektur

Die Clients werden über das Netzwerk mit einem oder mehreren Fileservern verbunden. Diese exportieren dann die Dateisysteme der an ihnen angeschlossenen Laufwerke. Die Laufwerke können wiederum über ein SAN mit den Fileservern verbunden sein. Ein Anwendungsprogramm merkt keinen Unterschied zwischen einem lokalen Laufwerk und einem virtuellen, über den Fileserver zur Verfügung gestellten. Der NFS-Client übernimmt die Rolle eines weiteren Dateisystem-Treibers, leitet die Anfragen aber an den NFS-Server weiter, anstatt direkt auf ein Blockgerät zuzugreifen (Abbildung 3).

3.3 Sicherheit

Da die Daten nun über ein Netzwerk verschoben werden, müssen entsprechende Vorkehrungen getroffen werden, die einen unberechtigten Zugriff auf diese Unterbinden. Handelt es sich um ein unsicheres Netzwerk, wenn zum Beispiel einzelne Clients über das Internet verbunden sind, dann muss die Kommunikation mit dem Fileserver verschlüsselt ablaufen. Diese Funktionalität fällt jedoch meist nicht in den Aufgabenbereich eines DFS, sondern wir mit herkömmlichen Mitteln wie Tunneln, bzw. Virtual Private Networks (VPNs) gelöst. Aber auch wenn das zugrunde liegende Netzwerk als sicher angesehen werden kann, weil zum Beispiel

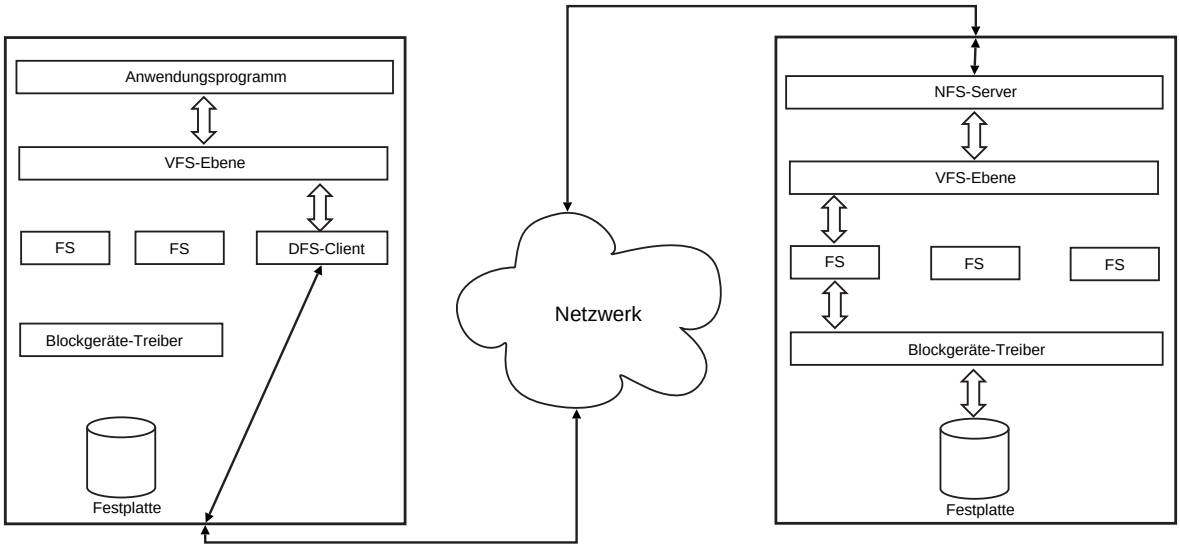


Abbildung 3: VFS-Architektur

die Rechner über feste Anschlüsse verbunden werden, bleibt das Problem der Authentifizierung. Authentifizierung kann bedeuten, dass die Identität jedes Benutzers festgestellt wird, der mit Daten auf dem Fileserver arbeiten will. Die einfachere Variante ist, nur die Identität des Clients zu überprüfen, und diesem das Rechtemanagement der dort arbeitenden Benutzer zu überlassen. Weist einer der Clients jedoch Sicherheitslücken auf, so könnten Unbefugte auf die dort gemounteten Dateisystem-Ausschnitte zugreifen. Diese Problematik ist zwar nicht DFS-spezifisch, da dies auch der Fall bei lokaler Speicherung der Daten wäre. Jedoch wird es bei einer großen Anzahl von mount-berechtigten Clients zunehmend schwierig, die Sicherheit aller Clients zu gewährleisten. Methoden der Authentifizierung können der Vergleich von Netzwerkadressen sein, falls nur die Identität des Clients festgestellt werden soll, oder die Übermittlung von Passwörtern, unter Zuhilfenahme der üblichen kryptographischen Verfahren. Falls für Clients die Dateisysteme anderer Clients sichtbar sind, muss schließlich auch den Nutzern bewusst sein, dass sie die Rechte ihrer eigenen Dateien entsprechend setzen. Sieht jeder Client nur seine eigenen Dateien, so entfällt diese Problematik, jedoch ist damit auch ein gemeinsames Nutzen von Dateien nicht mehr möglich.

3.4 Caching

Es gibt ein großes Gefälle zwischen der Zugriffszeit zu auf lokalen Festplatten gespeicherten Dateien, und zu solchen, die den Weg über das Netzwerk nehmen. Dies trifft auch auf die Übertragungsraten zu. Ferner belastet das steigende Datenaufkommen das Netzwerk stark, und die übrigen Nutzer werden ausgebremst. Aus diesem Grund greift man in modernen DFS auf Caching zurück. Bereits vom Fileserver übertragene Daten werden beim Client in einem Cache gehalten, und können beim nächsten Zugriff dann aus diesem bedient werden; das nochmalige Versenden über das Netzwerk entfällt. Zu beachten ist, dass die Konsistenz der Daten gewahrt werden muss - ein Problem, wenn mehrere Clients die selben Daten in ihrem Cache halten. Ändert ein Client eine Datei, führt dies dazu, dass die übrigen mit veralteten Daten arbeiten. Deswegen spielen Locking-Verfahren im DFS-Umfeld eine große Rolle.

4 Cluster-Dateisysteme

Falls die Festplatten über ein SAN mit den Client-Rechnern verbunden sind, können Cluster-Dateisysteme eingesetzt werden. Sie setzen an einer anderen Stelle an als verteilte Dateisysteme. Die einzelnen Rechner greifen direkt auf die Blockgeräte zu, ohne den Umweg über einen Fileserver zu gehen. Die Aufgabe von CFS ist es nun, diesen Zugriff zu koordinieren. Der Datenverkehr findet also über das schnellere, und sowieso vorhandenen Speichernetz statt, das normale LAN wird nicht belastet, und Flaschenhälse in der Form von zentralen Fileservern werden vermieden.

5 NFS

5.1 Geschichte

NFS steht für Network File System und wurde von Sun Microsystems entwickelt. Es stammt aus dem Jahr 1984 und hat sich seither zum De-Facto-Standard für verteilten Dateizugriff unter Unix entwickelt. Seine Spezifikation liegt in der Version 4 vor, aber die Versionen 2 und 3 sind immer noch weit verbreitet.

5.2 Architektur

NFS fußt auf den Virtual-File-System-Layer (VFS) von Unix. Auf dem Client-Rechner, der auf die entfernt liegenden Daten zugreifen will, erscheint es als eines der normalen Dateisysteme. Die entsprechenden Anforderungen, wie das Öffnen einer Datei oder das Schreiben eines Datenblockes werden vom NFS-Client entsprechend dem NFS-Protokoll an den Server weitergeleitet. Der NFS-Server empfängt diese Anforderungen und führt sie bei sich aus. Die Überwachung der Zugriffsrechte ist Aufgabe des Clients. Der NFS-Server kann transparent vom Betriebssystem implementiert werden, da er als Anwendungsprogramm nur die normalen Dateioperationen durchführen muss. Oft wird jedoch eine Integration in das Betriebssystem aus Effizienzgründen gewählt.

5.3 Netzwerkprotokoll

Zur Kommunikation verwenden Client und Server Remote Procedure Calls (RPC), wobei wahlweise (je nach verwendeter Software) UDP oder TCP als Transportprotokoll genutzt werden. TCP steht dabei offiziell erst ab NFSv3 zur Verfügung. NFS ist stark an das Unix-Modell von Dateien bzw. Dateisystemen angelehnt. Es unterstützt die üblichen Dateiattribute, Dateitypen, Links, etc.. NFS arbeitet zustandslos, was die Konzeption von NFS-Servern vereinfacht. Viele Anforderungen an ein DFS, wie z.B. Locking von Dateien, lassen sich jedoch nur mit Hilfe verbindungsorientierter Protokolle realisieren, was zur Folge hat, dass diese separat vom NFS-Server als Zusätze implementiert werden müssen.

5.4 Unterschiede zu lokalen Dateisystemen

Ein Unterschied zwischen einem echten, lokalen Dateisystem, und NFS ist, dass beim echten Dateisystem Anforderungen wie das Schreiben sofortige Änderungen an der betroffenen Datei zur Folge hat, wohingegen bei NFS solche Änderungen erst bei schließen der Datei wirksam werden. Ein direktes Schreiben führt zu hoher Netzwerkbelastung und erschwert das Cachen von Dateien, weshalb viele NFS-Implementationen diese so genannten 'Session-Semantics' (im Gegensatz zu Unix-Semantics) wählen.

5.5 Caching

Wie schon Locking ist auch Caching im NFS-Standard nicht näher spezifiziert. Die meisten NFS-Produkte verwenden aber ihre eigenen Caching-Mechanismen, wodurch die Leistung erheblich gesteigert werden kann.

5.6 NFSv4

In NFSv4 kommen viele Neuerungen hinzu, die eine bessere Anbindung an Windows-Betriebssysteme zu ermöglichen, und auch Erweiterungen vieler Unixe wie ACLs Rechnung zu tragen. Locking-Fähigkeiten sind nun in rudimentärer vorhanden. Ein Lock wird dem NFS-Client immer für eine bestimmte Zeitspanne (Lease) zugestanden, wonach er ihn erneuern muss. In NFSv4 ist auch die Zustandslosigkeit des Protokolls aufgegeben worden.

6 NFS in der Praxis

6.1 Usermapping

Damit ein Benutzer von einem Client-Rechner aus auf seine NFS-verwaltete Dateien zugreifen kann, muss unter Umständen seine User-Id angepasst werden. Außer der UID 0, die in allen Unixen dem Superuser vorbehalten ist, können die übrigen nämlich je nach System verschieden sein, und müssen folglich auf Client und Server nicht zwangsläufig übereinstimmen. Das Gleiche trifft auf Group-Ids zu. Dieses Mapping kann statisch stattfinden, indem serverseitig in einer Tabelle die entsprechenden Kombinationen abgelegt werden. Das Mapping kann mit Hilfe von NIS (Network Information Service) erfolgen, welches von zentraler Stelle aus Client und Server mit den Ids versorgt. Eine dezentrale Möglichkeit besteht in der Nutzung eines UGID-Daemons (für User Group ID), der, auf dem Client laufend, die IDs anhand der Namen mit dem Server abstimmt.

6.2 Firewalls

Um vor Zugriff Unbefugter zu schützen, können Firewalls die Nutzung auf bestimmte Quelladressen einschränken. Das können dann sowohl ganze Netze oder einzelne Rechner sein. Authentifizierte Tunnel können gegen Angriffe, bei denen der Angreifer eine Falsche IP-Adresse vorspiegelt helfen. Diese Mechanismen sind jedoch nicht Teil von NFS selbst. Die einzelnen Zugriffe können auch protokolliert und in Log-Dateien abgespeichert werden, um solche Angriffe erkennen zu können.

6.3 Hilfsprogramme

Neben dem NFS-Daemon, der die Hauptarbeit erledigt, kommen auf NFS-Servern weitere Daemons zum Einsatz, die z.B. für die Bearbeitung initialer Mount-Anfragen oder das Locken von Dateien zuständig sind. Ein Portmapper leitet eingehende Anfragen an die entsprechenden Ports weiter. NFS erlaubt es, Dateisysteme als ganzes oder auch nur bestimmte Ausschnitte an die Clients zu exportieren. Falls die Clients sehr viele NFS-Shares mounten müssten, z.B. weil sie Zugriff auf die Home-Verzeichnisse aller Benutzer einer Organisation erlauben sollen, kann es sinnvoll sein, das Mounten bis zu dem Zeitpunkt zu verschieben , zu dem der Zugriff wirklich erfolgt. Dieses On-Demand-Mounten erledigen Automounter, die dem Benutzer und dem NFS-Client zwischengeschaltet sind, und bei Bedarf die notwendigen NFS-Shares mounten.

7 CIFS/SMB

7.1 Geschichte

CIFS hat seine Wurzeln im Distributed Computing Environment (DCE bzw. DCE/RPC). Dieses Protokoll wurde aus lizenzrechtlichen Gründen von Microsoft als MS RPC reimplementiert und in Server Message Block (SMB) weiterentwickelt. SMB beschreibt, wie Daten über ein Netzwerk verschoben werden können, in der mittlerweile relevantesten Version NBT wird TCP/IP verwendet. Der Bestandteil von SMB, der den verteilten Zugriff auf Dateien ermöglicht, wurde schließlich in Common Internet File System (CIFS) umbenannt. Neben den Microsoft-Produkten existiert mit Samba eine freie Implementierung, die die Nutzung von CIFS unter Unix möglich macht. Mittlerweile existieren auch eine Vielzahl von NAS-Lösungen (siehe Abschnitt 10.1), die das Verwenden von CIFS vereinfachen.

7.2 Netzwerkprotokoll

Das in der Windows-Welt gebräuchliche Netzwerk-Protokoll war lange Zeit NetBIOS, welches sich grundsätzlich von den Internet-Protokollen unterscheidet. NBT regelt die Interaktion zwischen den beiden Protokoll-Welten. Ein NetBios-Rechner hat einen Namen, unter dem auf ihn zugegriffen werden kann, ein TCP/IP-Rechner in erster Linie nur eine IP-Adresse. Um die angemeldeten Rechner zu verwalten und mit einer IP-Adresse zu verknüpfen, kann ein NetBIOS-Name-Server (NBNS) verwendet werden. Die NetBIOS Namen zusammen mit den angesprochenen Objekten, die sie anbieten, sind unter der Bezeichnung Universal Naming Convention (UNC) bekannt und stellen das Gegenstück zu den URLs unter TCP/IP dar. NBT-Rechner können verschieden Dienste anbieten, dazu gehört unter anderem der Fileserver-Dienst. NBT und NetBIOS erlauben es, das Netzwerk in Arbeitsgruppen (Workgroups) zu partitionieren. Im Gegensatz zu NFS (bis v3) ist CIFS verbindungsorientiert. Bei CIFS ist es die Aufgabe des Servers, die Zugriffsberechtigung auf Dateien zu überprüfen. Die Clients müssen sich mit Benutzername und Identitätsnachweis, z.B. Passwort, beim Server für jede Session anmelden.

7.3 Caching

CIFS unterstützt Read-Ahead- und Write-Caching. Beim Read-Ahead-Caching fordert der CIFS-Client Daten nach der aktuellen Leseposition an, um bei fortgesetztem sequenziellen Zugriff die Anfragen aus dem Cache bedienen zu können. Beim Write-Caching werden Änderungen vorläufig nur im Cache sichtbar, und erst zu einem späteren Zeitpunkt zum Server überragen. Der CIFS-Server informiert einen Client über Zugriffe anderer Clients auf gleiche Dateien. Somit kann jeder Client die optimale Caching-Strategie wählen.

7.4 Locking

CIFS unterstützt Datei- und Record-Locking. Record-Locking bedeutet, dass Teile einer Datei gelockt werden können. Eine Datei muss nicht gelockt werden - dies geschieht auf Anforderung des Clients. Es kommen Oplocks (Oportunistic Locks) zum Einsatz: Der Server kann, z.B. wenn ein anderer Client auf eine Datei zugreifen will, den gewährten Lock zurücknehmen. Neben Oplocks können noch andere Locking-Mechanismen, wie Byte-Range- und Sharing-Locks verwendet werden. Diese werden aber in der Regel nicht von den CIFS-Clients, sondern von den Anwendungen auf dem Client-Rechner direkt angefordert.

7.5 Oplocks

Oplocks stellen ein einfaches Mittel dar, ein DFS um Mehrbenutzer-Eigenschaften zu erweitern, ohne Geschwindigkeitseinbußen in Kauf zu nehmen, falls die Benutzer keine Dateien gleichzeitig verwenden. Öffnet ein Client A eine Datei, so bekommt er einen Oplock, falls er bis dato der einzige ist, der die Datei geöffnet hat. Kein anderer Client kann die Datei öffnen, solange A den Oplock hält. Allerdings ist es der Server, der entscheidet, wann A den Oplock wieder verliert, er teilt dies A mit und wartet eine gewisse Zeitspanne auf eine Bestätigung. A kann vorher den Inhalt der Datei auf den Server zurückschreiben, falls er diesen geändert hat. Arbeitet mehr als ein Client mit der selben Datei, so bekommt keiner einen Oplock zugebilligt, und Änderungen müssen direkt an den Server zurückgemeldet werden. Arbeitet hingegen nur ein einziger Client mit einer Datei, was im normalen Einsatz den Regelfall darstellt, so kann er deren Inhalt cachen und somit den Zugriff erheblich beschleunigen.

7.6 CIFS mit Windows

Windowsversionen ab Windows für Workgroups erlauben die Nutzung von CIFS. CIFS-Client und Server sind kombiniert, das heißt, ein Benutzer kann sowohl eigene Laufwerke und Verzeichnisse exportieren und auch auf andere zugreifen. Daneben erlaubt die Software das Verfügbarmachen und Nutzen anderer Ressourcen wie Drucker im Netzwerk.

7.7 CIFS mit Unix

Unter Unix steht mit Samba ein freie Software-Lösung zur Verfügung, die Client und Serverfunktionalität beinhaltet. Samba verwendet die gleiche Benutzer-Datenbank wie das Unix-System, auf dem es läuft, dass heißt es wird versucht, CIFS so nahtlos wie möglich in eine heterogene (Windows/Unix) Umgebung einzugliedern.

8 Vergleich NFS - CIFS

Bei NFS wurde historisch mit Blick auf die Performanz entwickelt, der Schwerpunkt bei CIFS lag auf Mehrbenutzer-Fähigkeit. Sichtbar wird dies z.B. anhand des Lockings, das fester Bestandteil von CIFS ist, und bei NFS als externe Komponente realisiert wird. Bei CIFS wiederum war ursprünglich keinerlei Caching vorgesehen. Beides trifft aber nicht mehr auf aktuelle Versionen der Protokolle zu. NFS hat sich, was die Auswahl der verfügbaren Dateioperationen angeht, historisch an Unix angelehnt, ist aber mit NFSv4 in Richtung Windows erweitert worden. CIFS, das seinen Ursprung im Windows-Umfeld hat, kann aufgrund dessen mächtigeren Dateisystems auch einfacher mit Unix zusammenarbeiten.

9 AFS und Coda

AFS wurde an der Carnegy Mellon University entwickelt. Es ist umfangreicher als NFS, benutzt z.B. Kerberos zur Authentifizierung und eine erweiterte Caching-Strategie. Eine Variante von AFS namens DFS der Open Software Foundation wird im RZ der Uni Karlsruhe zum Zugriff auf den Parallelrechner angeboten. Als Nachfolger von AFS ist Coda mittlerweile etabliert. Coda geht eine Spur weiter als die bisher genannten verteilten Dateisysteme. NFS und CIFS setzen voraus, das Client und Server über ein Netzwerk verbunden sind. Es werden höchstens Vorkehrungen getroffen, um die Performanz-Einbußen bei Unterbrechung einer

Verbindung gering zu halten. Coda hingegen unterstützt es, auch bei fehlender Verbindung zum Server mit den Dateien weiterzuarbeiten, die auf dem Client liegen. Dies setzt natürlich komplizierte Synchronisationsmechanismen bei Wiederaufbau der Verbindung voraus (wenn einer der Clients eine Datei ändert). Coda hat allgemein einen höheren Funktionsumfang, wie auch sein Vorgänger AFS, es unterstützt z.B. Server-Replikation und verfügt über ein Sicherheitsmodell zur Verschlüsselung und Authentifizierung.

10 NAS und SAN

10.1 NAS

Als Network Attached Storage (kurz NAS) werden fertige Fileserver bezeichnet, bestehend aus DFS-Server-Software und optimierter Hardware. Sie zeichnen sich durch einfache Handhabung aus, lassen sich schnell in Betrieb nehmen und einfacher administrieren, als viele in Betriebssystemen eingebaute DFS-Software. Die Fileserver werden an das vorhandene Netzwerk angeschlossen und erlauben danach den übrigen Rechnern den Zugriff auf die von ihnen verwalteten Dateien (Abbildung 4).

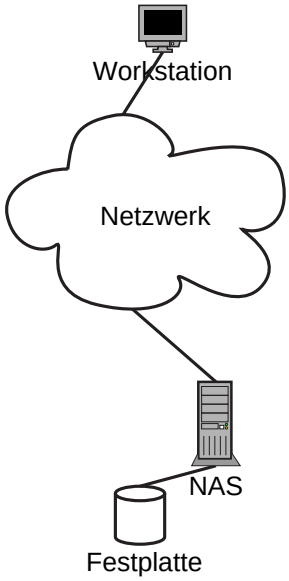


Abbildung 4: NAS

10.2 NAS vs. SAN

Oft werden NAS und SAN als konkurrierende Produkte dargestellt. Sie sind es insofern, als beide dazu eingesetzt werden, die Speicherkapazität der eingesetzten IT-Infrastruktur zu erweitern. Da bei SAN speziell hierfür entwickelte Netzwerk-Architekturen zum Einsatz kommen, die außerdem parallel zum herkömmlichen Netzwerk installiert werden, ergeben sich natürlich gewisse Geschwindigkeitsvorteile. Andererseits erlaubt NAS ganz andere Möglichkeiten der Interaktion, da die einzelnen Rechner mit den gleichen Dateien arbeiten können. Durch Caching kann hier einiges des genannten Geschwindigkeitsnachteils wett gemacht werden. Im Grunde lassen sich SAN und NAS auch als komplementäre Technologien betrachten, denn schließlich muss auch der Fileserver irgendwo seine Dateien speichern. Eine Kombination

von NAS und SAN bietet sich also an (Abbildung 5), um die Vorteile beider Speichertechnologien zu vereinen.

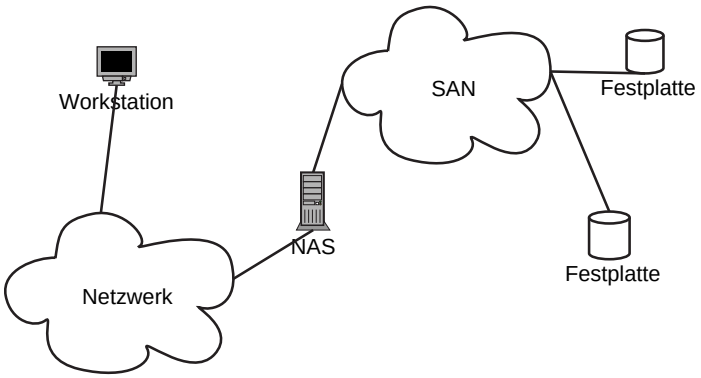


Abbildung 5: NAS und SAN kombiniert

11 NFS und CIFS in der RZ-Praxis

Das Rechenzentrum der Universität Karlsruhe ermöglicht den Instituten und Studenten, ihre Dateien auf einem Server abzulegen. Intern werden hierfür augenblicklich ein NAS-Element namens Celerra und dahinter ein SAN-Element Namens Symmetrix der Firma EMC verwendet. Das Celerra System besteht aus drei PC, von denen einer im normalen Betrieb als Ersatz zur Verfügung steht. Einer der PC fungiert als Kontroll-Station. Auf der Kontrollstation läuft ein Linux, auf den übrigen ein DART genanntes OS. Die PCs sind über mehrere Netzwerkschnittstellen mit der Außenwelt verbunden. Alle Dateisysteme der Celerra sind auf dem rzfs-Rechner gemounted, von dem aus Aufgaben wie erzeugen von Benutzern und deren Homedirectories, etc., erledigt werden. Die auf den Platten abgelegten Dateisysteme (vom Typ UxFS) werden auf Slices angelegt, es können im Betrieb zusätzliche Slices zu den einzelnen FS hinzugefügt werden.

12 Ausblick

Mit NFS und CIFS wurden exemplarisch zwei verteilte Dateisysteme herausgegriffen, da diese sich in der Praxis bewährt haben. Es existieren noch eine Vielzahl weiterer verteilter Dateisysteme, an Labors und Universitäten entwickelt und Teilweise wohl nie das Versuchsstadium verlassen werden. Vielmehr erscheint es wahrscheinlich, dass die beiden genannten auf absehbare Zeit ihre dominante Rolle behalten werden, da viele Firmen bereits in entsprechende Lösungen investiert haben. Bei CIFS hat Microsoft den Weg einer Standardisierung eingeschlagen. Die entsprechende Spezifikation ist frei verfügbar, was sich positiv in Verfügbarkeit und Investitionssicherheit auswirkt. NFS ist ein fester Bestandteil aller modernen Unixe. Mit der vierten Version wurden Schritte unternommen, Altlasten zu beseitigen und das Protokoll auch mit von Unix verschiedenen Betriebssystemen nutzbarer zu machen. Viele der in alternativen DFS-Projekten erarbeiteten Ansätze werden nach und nach integriert, sofern dies technisch machbar und aus betriebswirtschaftlicher Sicht wünschenswert ist. Als Beispiel sei hier Caching genannt, welches in ursprünglichen Versionen von CIFS nicht vorgesehen war. Wichtige Aspekte, die nicht angesprochen wurden, sind der Sicherheit und Interoperabilität. Diese fallen jedoch nicht in den Kernbereich von verteilten Dateisystemen, sondern sind vielmehr allgemeine Aspekte.

Literatur

[Pres02] W. Curtis Preston. *Using SANs and NAS*. O'Reilly. 2002.

[t0302] Common Internet File System (CIFS) Technical Reference. SNIA Technical Proposal, März 2002.

[TavS02] Andrew S. Tanenbaum und Maarten van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall. 2002.

[TrEr03] Ulf Troppens und Rainer Erkens. *Speichernetze: Grundlagen und Einsatz von Fibre Channel SAN, NAS, iSCSI und InfiniBand*. dpunkt.verlag. 2003.

Abbildungsverzeichnis

High Performance Interconnects

Christian Gärtner

Kurzfassung

Für den Betrieb von Clustern (Rechnerbündeln) ist die Wahl des geeigneten Interconnects von fundamentaler Bedeutung. Die Kommunikation zweier Knoten ist um Größenordnungen langsamer als der Zugriff auf den Hauptspeicher des lokalen Rechners. Deshalb sind die Latenzzeit und die Bandbreite des Interconnects die entscheidenden Faktoren, welche die Fähigkeit eines Clusters bestimmen, Aufgaben effizient verteilt zu verarbeiten. Dies gilt insbesondere bei Clustern mit vielen Knoten und bei Anwendungen mit häufiger Kommunikation. In dieser Seminararbeit werden ausgehend von Gigabit Ethernet unterschiedliche Interconnect-Techniken vorgestellt und hinsichtlich ihrer Leistungsfähigkeit und Kosten charakterisiert.

1 Einleitung

Möchte man eine Aufgabe auf mehrere Rechner verteilt ausführen, so besteht in gewissen zeitlichen Abständen Kommunikationsbedarf zwischen den Knoten. Der Zugriff auf ein Datum im lokalen Hauptspeicher erfolgt innerhalb von etwa $t_H = 10ns$ (PC400, DDR-Ram, Stand 2004). Soll hingegen auf ein Datum im Hauptspeicher eines anderen Knoten über ein Netzwerk zugegriffen werden, so benötigt der Zugriff wesentlich länger. Gigabit Ethernet (ohne Verbesserungen) benötigt hierfür beispielsweise etwa $t_{NW} = 100\mu s$. Dieser große Unterschied führt bei häufiger Kommunikation zu einer deutlichen Verlangsamung.

Bei zunehmender Anzahl an Rechenknoten steigt auch das Kommunikationsaufkommen. Dies ist beispielsweise auf eine häufiger notwendige Synchronisation zurückzuführen. Wegen der sehr unterschiedlichen Zugriffszeiten t_H und t_{NW} nimmt der zeitliche Anteil für die Kommunikation an der Gesamtzeit spürbar zu. Dies kann dazu führen, dass ab einer bestimmten Anzahl an Knoten kein Geschwindigkeitsgewinn durch Erhöhung der Knotenzahl erreichbar ist. Diese maximale Knotenzahl wird durch die Häufigkeit der Kommunikation sowie durch die Leistungsfähigkeit des Interconnects bestimmt. Aus diesem Grund ist es von großer Wichtigkeit, einen Interconnect mit ausreichender Leistung zu verwenden.

Anwendungsbeispiel — Dieser Zusammenhang soll an einem Beispiel verdeutlicht werden. Wir setzen eine Hauptspeicher-Zugriffszeit von $t_H = 10ns$ vorraus. Auf einen anderen Knoten soll über das Netzwerk in $t_{NW} = 100\mu s$ zugegriffen werden können, was in etwa der Leistungsfähigkeit von Gigabit Ethernet entspricht. Es wird angenommen, dass sich der Kommunikationsbedarf quadratisch mit der Zahl der Knoten n_{node}im Cluster erhöht. Bei zwei Knoten sei einer unter 10^9Befehlen ein Kommunikationsbefehl. Der Anteil von Rechenzeit und Kommunikationszeit an der Gesamtzeit ist in Abbildung 1 in Abhängigkeit der Knotenzahl aufgetragen. Wird nur ein Knoten verwendet, so setzt sich die Gesamtzeit zur Ausführung erwartungsgemäß nur aus Rechenzeit zusammen. Werden nun mehr als ein Knoten verwendet, so sinkt die Rechenzeit, jedoch kommt ein steigender Anteil an Kommunikationszeit hinzu. Man erkennt ein Minimum der Gesamt-Ausführungszeit bei 10 Knoten von 181,0 Zeiteinheiten. In diesem

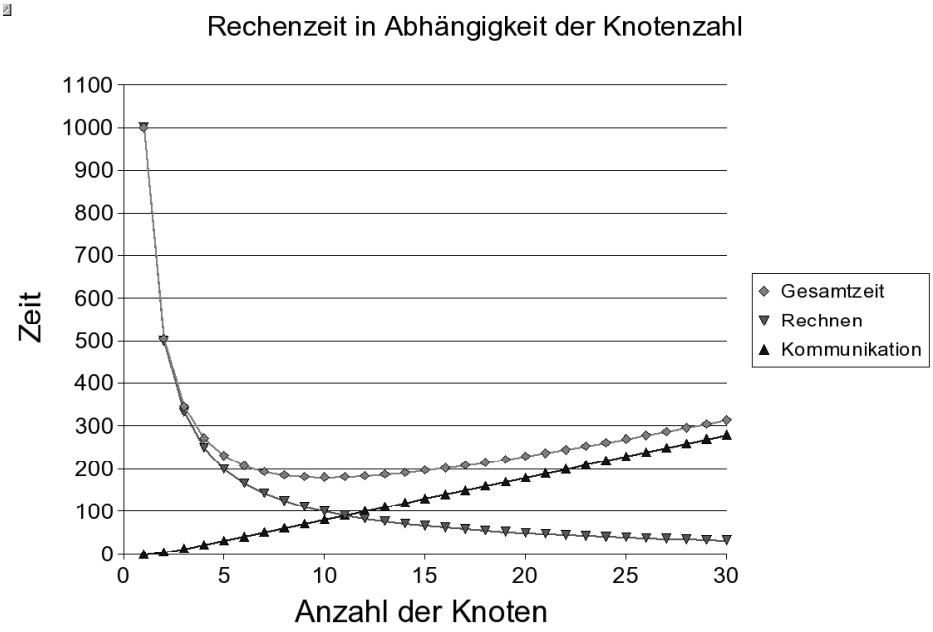


Abbildung 1: Beispiel der Zeitanteile zum Rechnen und zur Kommunikation in Abhängigkeit der Knotenanzahl für $t_{HS} = 10ns$ und $t_{NW} = 100\mu s$

Beispiel ist es somit nicht sinnvoll, mehr als 10 Knoten zu verwenden, da sonst die insgesamt benötigte Zeit wieder ansteigt.

Es soll nun die Leistungsfähigkeit des Interconnects verbessert werden. Es wird eine Zugriffszeit über das Netzwerk von $t_{NW} = 10\mu s$ angenommen. Dieser Leistungswert wird von fast allen, für des Clusterbetrieb entwickelten Interconnects erreicht. Abbildung 2 zeigt die benötigte Rechenzeit, welche über der Anzahl der Knoten aufgetragen wurde. Der schnellere Interconnect hat zur Folge, dass viel mehr Knoten effizient verwendet werden können, um eine Geschwindigkeitssteigerung zu erzielen. Die optimale Knotenzahl liegt in diesem Fall bei 32, die lediglich 61,3 Zeiteinheiten für die Berechnung der Aufgabe benötigen, was eine Geschwindigkeitssteigerung um den Faktor 2,95 zum ersten Fall bedeutet. Durch einen schnellen Interconnect kann also die Anzahl der effizient verwendbaren Knoten gesteigert werden und die gesamte Bearbeitungszeit verringert werden.

In Abhängigkeit von der Netzwerk-Zugriffszeit t_{NW} auf einen anderen Knoten soll nun die Zahl der Knoten bestimmt werden, bei welcher die Ausführungszeit minimal ist. Abbildung 3 zeigt die Anzahl der Knoten für ein optimales Bearbeitungsergebnis über der logarithmisch aufgetragenen Zugriffszeit. Man erkennt, dass durch eine kleinere Netzwerk-Zugriffszeit die Anzahl der effizient verwendbaren Knoten und damit auch die Leistungsfähigkeit des Clusters sehr stark ansteigt.

Weitere wichtige Merkmale sind die Datenübertragungsrate und die Topologie des Netzwerks, sowie dessen Skalierbarkeit und die Kosten pro Knoten für den Interconnect. Das für die spezifische Anwendung geeignetste Netz muss nach Berücksichtigung dieser Parameter ausgewählt werden [Tess03, PaHa92, FPeaCSD01a].

2 Topologie des Netzes

Die verfügbaren Techniken können nach der Art der Vernetzung klassifiziert werden. In diesem Kapitel wird die Vernetzung in Form fetter Bäume und in Form eines 2D- bzw. eines 3D-Torus

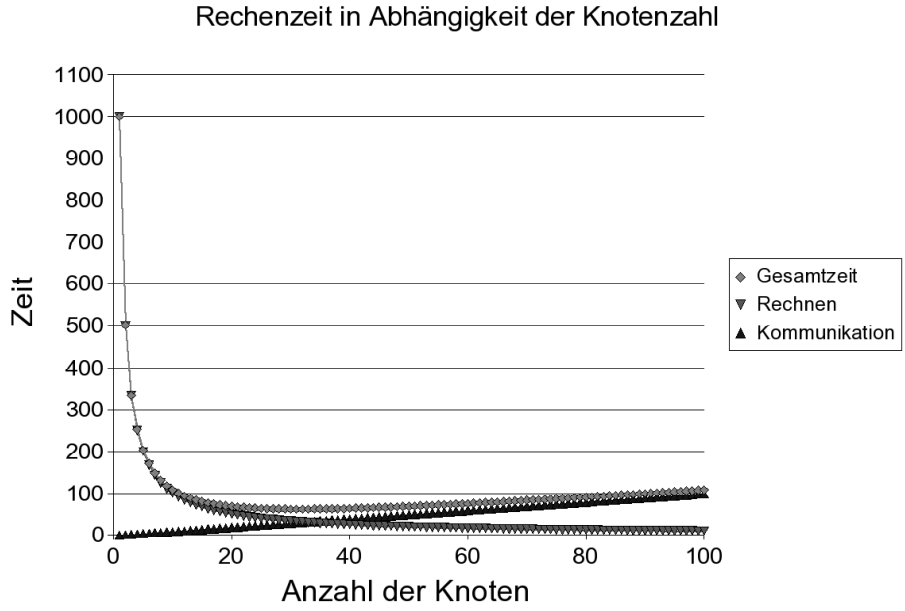


Abbildung 2: Beispiel der Zeitanteile zum Rechnen und zur Kommunikation in Abhängigkeit der Knotenanzahl für $t_{HS} = 10ns$ und $t_{NW} = 10\mu s$

beschrieben. Dieser Unterschied ist in Hinblick auf die maximal erreichbare Performance, das Kommunikationsverhalten der Anwendung sowie aus Kostengesichtspunkten interessant.

2.1 Fette Bäume (k, n)

Um eine optimale Performance zu erreichen und die Skalierbarkeit des Clusters sicherzustellen möchte man, dass gleichzeitig jeder Knoten mit jedem anderen Knoten kommunizieren kann. Dies wird erreicht durch so genannte fette Bäume (fat trees). Ein fetter Baum wird beschrieben durch die beiden Parameter Dimension n und die Verbindungszahl k. Ein fetter Baum fB(k, 1) (d.h. mit Verbindungszahl k und der Dimension 1) besteht aus einem Switch, an welchem k Knoten angeschlossen sind. Ein fetter Baum der Dimension n wird nun dadurch gebildet, dass k fette Bäume der Dimension n-1 mit k weiteren Switches verbunden werden. Abbildung 4zeigt fette Bäume der Konnektivität $k = 2$ für die Dimensionen $n = 1, 2, 3$. Für einen fetten Baum fB(k, n) werden $n * k^{n-1}$Switches benötigt. Ein solcher Baum kann bis zu k^nKnoten aufnehmen. Die vielen verwendeten Switches tragen maßgeblich zu den Kosten des Interconnects bei. Diese Topologie wird bei Myrinet, Infiniband und Quadrics verwendet [Meye04, BaPa93, Zhao03, PaHa92].

2.2 Torus mit 2 oder 3 Dimensionen

Ein anderer topologischer Ansatz besteht darin, die Knoten des Clusters in Ringen zusammenzuschalten. Jeder Knoten verfügt über 2 bzw. 3 Ports. Über jeden Port wird dieser Knoten mit einer gewissen Anzahl von Knoten verbunden. Abbildung 5 zeigt 16 Knoten, die durch einen 2D-Torus verbunden sind.

Der große Vorteil des Verwendens der Torus-Vernetzung besteht darin, dass die hohen Kosten für Switches eingespart werden können. Neue Knoten können einfach hinzugefügt werden, indem sie in eine der bestehenden Ringstrukturen hinzugefügt werden. Ein wesentlicher Nachteil besteht darin, dass die Ringe zum Flaschenhals werden können. Bei einer zu großen Zahl an

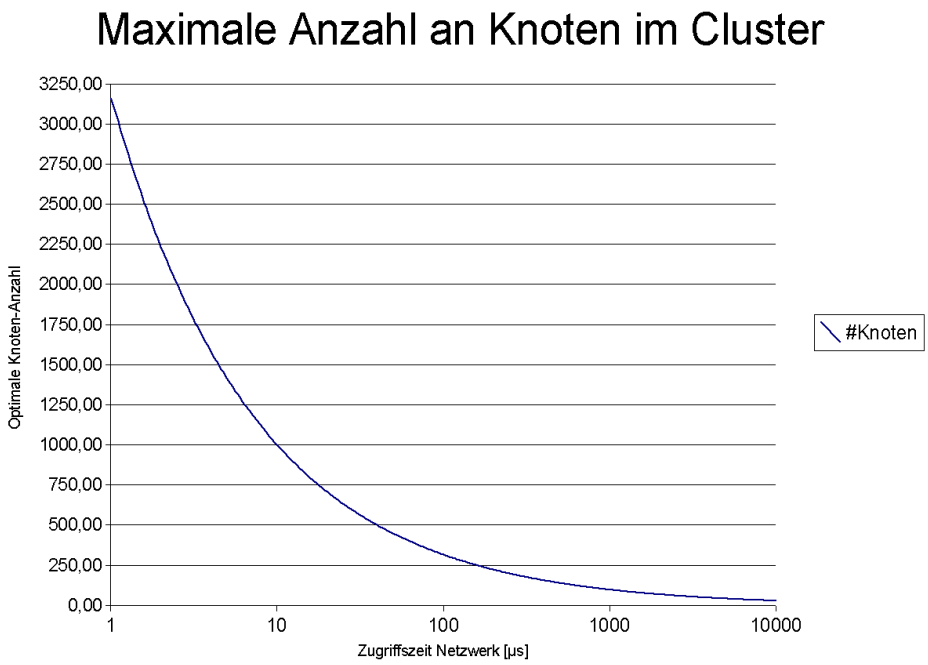


Abbildung 3: Beispiel für die optimale Anzahl an Knoten in Abhängigkeit der Latenzzeit

Knoten oder bei Anwendungen mit ungünstigem Kommunikationsverhalten reicht die Bandbreite des Netzes möglicherweise nicht mehr aus [OmPa97].

Torusförmig aufgebaute Netze werden beim Scalable Coherent Interface (SCI) der Firma Dolphin verwendet. Die maximale Zahl der Knoten wird durch die Übertragungsrate des Netzes begrenzt. In [Meye04] werden bei Verwendung von SCI je nach Anwendung 6 bis 8 Knoten pro Ring empfohlen.

3 Anbindung des Netzwerk-Adapters

Eine weitere wichtige Größe für die Leistungsfähigkeit des Interconnects ist die Anbindung des Netzwerk-Adapters an die CPU des PC-Systems. Netzwerkkarten liegen meist in Form von Steckkarten für eine bestimmte Art von PCI-Bus vor. Ist die Bandbreite des PCI-Busses

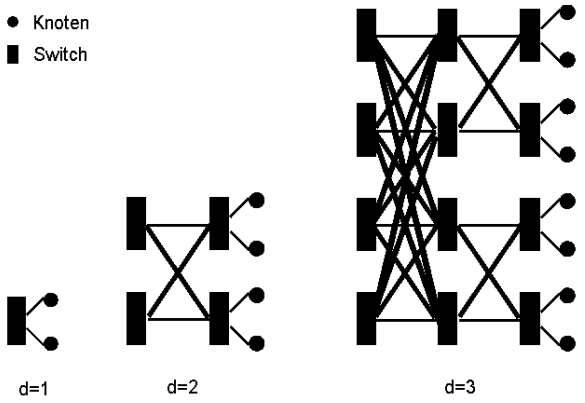


Abbildung 4: Fette Bäume der Konnektivität $k = 2$ für die Dimensionen $n = 1, 2, 3$

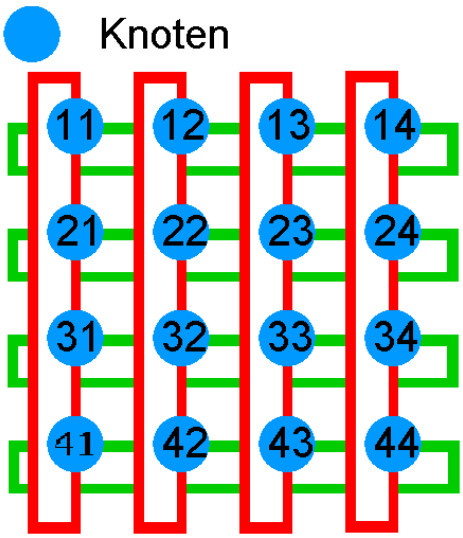


Abbildung 5: 2D-Torus mit 4 Knoten pro Ring

PCI-Version	PCI	PCI	PCI	PCI	PCI X	PCI X	PCI X
	2.0	2.1	2.2	2.3	1.0	2.0	3.0
Busbreite (Bit)	32	64	64	64	64	64	64
Taktrate (MHz)	33	66	66	66	133	533	1066
Bandbreite (GB/s)	0,13	0,5	0,5	0,5	1	4	8

Tabelle 1: Übersicht PCI-Busse (entnommen aus [GmbH04])

zu klein, so gerät dieser zum Flaschenhals, was dazu führt, dass nicht die volle Bandbreite des Interconnects genutzt werden kann.

In Tabelle 1 sind die verschiedenen Versionen von PCI-Bussen aufgelistet. In normalen PCs trifft man normalerweise den PCI 2.0-Bus mit einer Breite von 32 Bit und einer Frequenz von 33 Mhz. Dies ergibt eine maximale Bandbreite von 133 MB/s. In Server- und Cluster-Rechnern wird hingegen der 64 Bit breite PCI-X-Bus verwendet. Dieser ist mit bis zu 133 Mhz (PCI-X-1.0) verfügbar und wird beispielsweise von AMD Opteron-Plattformen unterstützt. Dieser hat somit eine theoretische Bandbreite von 1 GB/s.

Seit Sommer 2004 sind auch Systeme mit PCI Express Bus verfügbar. Dieser serielle Bus liefert pro Kanal eine Bandbreite von 256 MB/s. Es existieren Steckplätze, auf denen 8 bzw. 16 dieser Kanäle verwendet werden, womit eine Bandbreite von 2 GB/s bzw. 4GB/s erreicht werden kann.

Eine andere Möglichkeit besteht darin, den PCI-Bus komplett zu umgehen, indem der Netzwerk-Chip auf der Northbridge des Systems integriert wird. Dies ermöglicht eine potentiell bessere Latenzzeit und Bandbreite, die durch ein geringeres Maß an Flexibilität erkauft wird [Meye04, GmbH04].

4 Messung und Bewertung der Leistung des Interconnects

Die Bewertung der Leistungsfähigkeit eines Interconnects geschieht in der Regel durch die beiden Maßzahlen „Latenzzeit“ sowie „Bandbreite“. Unter der Bandbreite versteht man die

Anzahl an Daten, die pro Zeiteinheit zwischen zwei Knoten übertragen werden können. Die Bandbreite ist stark von der Größe der Pakete abhängig. Bei kleineren Paketen ist die bandbreite wesentlich niedriger als die theoretische Bandbreite, da kleine Pakete zu einem großen Overhead führen. Bei großen Paketen kommen die Interconnects der theoretischen Bandbreite recht nahe.

Die Zeitspanne, die vom Anfordern eines Datums bis zum Eintreffen vergeht, wird als Antwortzeit bezeichnet. Die Antwortzeit setzt sich zusammen aus der Datenübertragungszeit und der Latenzzeit. Die Datenübertragungszeit ist abhängig von der Paketgröße und bezeichnet den Zeitanteil, welcher zum Übertragen der eigentlichen Daten notwendig ist. Die Latenzzeit entspricht der Antwortzeit, bei Übertragung eines Pakets der Größe 0 Byte.

Diese Messungen werden mit unterschiedlichen APIs durchgeführt. Der schnellste Zugriff erfolgt dabei durch Verwendung des Message Passing Interface (MPI). Das Anwendungsprogramm verwendet dabei die vom Interconnect-Hersteller zur Verfügung gestellten Treiber, welche die Netzwerkkarte direkt ansprechen. Dabei wird das Betriebssystem umgangen, was den Overhead verringert und eine größere Bandbreite erlaubt. Die meisten Interconnects erlauben einen Zugriff auf den Hauptspeichers des Zielknotens, wodurch die CPU des Zielknotens umgangen wird. Ein Nachteil dieses Verfahrens besteht darin, dass die in der Anwendungssoftware verwendete API den verwendeten Interconnect unterstützten muss. Mittlerweile existieren jedoch zahlreiche kommerzielle und freie Implementierungen. Zu erwähnen ist das freie MPICH, für das Implementierungen für Myrinet, Quadrics und mittlerweile auch für Infiniband existieren.

Eine andere Möglichkeit besteht darin, das TCP-Protokoll zu verwenden. Dies führt jedoch meist zu einer signifikanten Verringerung der Leistung, da zum einen die CPU-Belastung steigt, als auch die Latenzzeit erhöht wird, da Betriebssystemfunktionen verwendet werden [BMBBe, RoCh99, GFDe00].

5 Aktuelle Techniken

In diesem Kapitel werden aktuelle Interconnect-Lösungen hinsichtlich der verwendeten Technik, ihrer Leistungsfähigkeit, ihres Einsatzgebiets sowie nach Kostengesichtspunkten betrachtet. Als Bezugssystem wird Gigabit Ethernet betrachtet. Dieses ist aufgrund seines niedrigen Preises eine im LAN-Bereich weit verbreitete Technik. Diese wird auf ihre Tauglichkeit als Cluster Interconnect bewertet.

Als weitere, speziell für die Verwendung in Clustern entwickelte Techniken, werden Myrinet, SCI, IBM HPS und Quadrics betrachtet. Am Ende des Kapitels wird noch auf InfiniBand eingegangen, das aufgrund seiner Skalierbarkeit bezüglich der Bandbreite sowie unter Kostengesichtspunkten eine interessante Alternative für zukünftige Anwendungen darstellt [Wilb03, Andj03].

5.1 Gigabit Ethernet

Ethernet eines Typs wurde bereits in der Vergangenheit als Cluster-Interconnect verwendet. Durch die recht niedrigen Leistungswerte beschränkt sich der Einsatz von Gigabit-Ethernet heutzutage jedoch auf Anwendungen mit geringem Kommunikationsanteil. Zudem wird Gigabit Ethernet meist als Netzwerk zur Anbindung des Clusters an die Außenwelt verwendet. Der Hauptvorteil von Gigabit Ethernet liegt in den sehr geringen Kosten pro Knoten. Deswegen wird es in diesem Kapitel als „Einstiegslösung“ vorgestellt. Weitere Informationen findet man in [eal95, GCSc02, FaOn00, uLew01, Mach99, BMBBe, RoCh99, Neal02].

5.1.1 Merkmale

Der Kern des Ethernet-Netzwerks wird aus Switches gebildet, an welche die Knoten mit Netzwerkkarten (NIC = network interface controller) angebunden werden. Die Größe des Clusters wird durch die Anzahl der Ports des Switches begrenzt. Ethernet-Switches sind sehr komplex aufgebaut, da sie für das gesamte Routing verantwortlich sind. Deswegen müssen vollständige Routing-Tabellen bereitgehalten werden, um ein möglichst schnelles Routing zu ermöglichen. Ethernet-Switches sind mit bis zu 480 Ports zu wirtschaftlichen Preisen erhältlich. Ein Kaskadieren von Switches würde zu einer Erhöhung der Latenzzeit und zu Flaschenhälsen hinsichtlich der Bandbreite führen und wird deshalb nicht verwendet.

5.1.2 Leistung

Der Aufbau des Ethernet-Protokolls führt zu einem recht großen Overhead bei der Datenübertragung und bei der CPU-Last, da der Datenstrom in Pakete unterteilt wird, die maximal 1500 Bytes Nutzdaten enthalten. Das Verwenden der CPU für die Datenübertragung ist ein Grund, weshalb man eine Latenzzeit von 90μs erreicht. Durch den Overhead, der durch die Paketbildung hervorgerufen wird, beträgt die üblicherweise erreichbare Bandbreite etwa 300-500 Mbps.

5.1.3 Verbesserungen

Um die Leistung von Gigabit Ethernet zu verbessern wurden zum einen Jumbo- Frames eingeführt. Diese erlauben maximal 9000 Bytes an Nutzdaten. Dadurch wird der effektive Overhead verringert und somit die Bandbreite erhöht. Dies setzt jedoch vorraus, dass sämtliche Komponenten des Netzwerks diese Technik unterstützen. Damit werden Bandbreiten von bis zu 940 Mbps erreicht (Alteon ACENIC Chipsatz).

Die hohe Latenzzeit von 90μs kommt dadurch zustande, dass für die Abarbeitung des Protokolls ein Mikroprozessor verwendet wird. Dies führt zwar zu mehr Flexibilität, jedoch wird dadurch die Latenzzeit erhöht. Abhilfe schafft die Verwendung von ASICs (Application specific integrated circuit). Dadurch können Latenzzeiten von bis zu 31μs erzielt werden (Broadcom Chipsatz).

5.1.4 Kosten

Gigabit Ethernet stellt die günstigste Lösung für Cluster Interconnects dar. Die Kosten werden in erster Linie bestimmt durch die Kosten für den Switch, welche von der Anzahl der Ports abhängen. Bei sehr kleinen Clustern (<64 Knoten) entstehen Kosten von \$150 pro Knoten. Bei Clustern mit 480 Ports fallen insgesamt Kosten von \$750 pro Knoten an.

5.1.5 10 Gigabit Ethernet

Eine weitere Alternative, um mit Ethernet bessere Performance zu erreichen, ist 10 Gigabit Ethernet. Der größte Vorteil besteht in der Kompatibilität mit früheren Ethernet-Generationen. Die größten Nachteile bestehen in der weiterhin hohen CPU-Belastung und die sehr hohen Kosten für Switches. Die Preise pro Knoten betragen etwa \$10000. Außerdem sind nur Switches mit bis zu 48 Ports erhältlich, weshalb die Skalierbarkeit nicht geben ist.

5.2 Myrinet

5.2.1 Merkmale

Myrinet ist der erste Interconnect, der speziell für Cluster entwickelt wurde. Bei diesem Interconnect wurde deshalb auf eine kleine Latenzzeit und eine hohe Datenübertragungsrate Wert gelegt. Deswegen wurde bei Myrinet im Vergleich zu Ethernet auf Merkmale verzichtet, welche im Clusterbetrieb nicht unbedingt notwendig sind. Beim Betrieb von Clustern kann das Netzwerk als statisch angenommen werden. Somit kann jedem Knoten der Aufbau des gesamten Netzes statisch bekannt sein. Dadurch kann das Routing von den Switches in die Knoten verlegt werden. Dies hat einen wesentlich einfacheren Aufbau der Switches und somit auch potentiell geringere Latenzzeiten und Kosten zur Folge. Die Knoten des Clusters werden bei Myrinet durch eine „Fetter Baum“-Struktur verbunden. Die Switches bieten 16 Ports.

Die Knoten des Netzwerks verfügen über eine PCI-X Netzwerkkarte, welche eine Übertragungsrate von 2 Gbps liefert. Die Netto-Übertragungsrate beträgt 1,88 Gbps. Es ist auch möglich, zwei Netze parallel zu betreiben, um die Übertragungsrate zu erhöhen. Dadurch verdoppeln sich jedoch auch die Kosten.

Myrinet weist eine MPI-Latenzzeit von 6-7 μs auf und eine TCP-Latenzzeit von 27-30 μs. Die Software wurde so ausgelegt, dass die CPU-Belastung und die Routing-Berechnungen auf ein Minimum reduziert wurden. Für Myrinet existiert, verglichen mit anderen Interconnects, die breiteste Unterstützung von verschiedenen Plattformen und Betriebssystemen. Es sind Software-Implementierungen für die Betriebssysteme Windows, Linux, Solaris, Mac OS, FreeBSD, AIX, Tru64, VxWorks und die Plattformen x64, Alpha, PowerPC, Sparc verfügbar. Der Hersteller Myricom bietet zahlreiche, Open-Source-basierte Software an[eal95, Myri04, CDea98].

5.2.2 Kosten

Myrinet ist unter den für Cluster entwickelten Interconnects die günstigste und verbreiteste Alternative. Die Kosten pro Knoten betragen bei einem Cluster $1295 bei bis zu 8 Knoten [Meye04], $1070 bei bis zu 128 Knoten [Meye04] und $1737 bis bis zu 1024 Knoten [BMBBe].

5.3 Scalable Coherent Interface (SCI)

5.3.1 Merkmale

Der SCI-Interconnect von Dolphin ist der einzige hier vorgestellte Interconnect, der keine Switches verwendet. Jeder Knoten verfügt über einen Netzwerkadapter (NIC) mit zwei oder drei Ports. Die Knoten werden nun in Form eines 2D- bzw. 3D-Torus zusammengeschalten (siehe Abschnitt 2.2). Dies hat den Vorteil, dass sich die Kosten linear mit der Anzahl der Knoten entwickeln.

Auf jedem Netzwerkadapter befindet sich ein ASIC (Application Specific Integrated Circuit), welcher die Routing-Aufgaben und das Weiterreichen von Daten übernimmt. Dadurch wird die CPU des Hostknotens entlastet. Der Nachteil besteht darin, dass beim Ausfall eines Knotens ein gesamter Ring ausfällt. Cluster, die einen 2D-Torus verwenden, können 64 bis zu 100 Knoten haben. Bei Verwendung eines 3D-Torus beträgt die Zahl der verwendbaren Knoten 640 bis zu 1000. Die maximale Kabellänge ist auf 3-5m begrenzt. Da jeder Knoten mit 4 bzw. 6 anderen Knoten verbunden werden muss, kann dies zu Poblemen bei der räumlichen Anordnung der Knoten führen [OmPa97].

5.3.2 Kosten

Die Kosten pro Knoten für den Interconnect sind unabhängig von der Anzahl der Knoten. Bei Clustern, die einen 2D-Torus-Interconnect verwenden, betragen die Kosten $1095 pro Knoten. Wird ein 3D-Torus-Interconnect verwendet, so haben die Netzwerkadapter 3 Ports und sind deshalb teurer. Die Kosten betragen dann $1595 pro Interconnect [Meye04].

5.4 InfiniBand

5.4.1 Merkmale

InfiniBand ist ein industrieller Standard, der für eine Vielzahl von Anwendungsmöglichkeiten vorgesehen ist. Diese reichen von der Aufgabe als einfacher System-Bus bis hin zur Anwendung als Interconnect in Cluster. Durch das breite Anwendungsfeld wird von einer hohen Stückzahl ausgegangen, was sehr geringe Kosten zur Folge haben könnte. Bei einer genügend großen Verbreitung ließe sich die InfiniBand-Hardware auf der Northbridge des Knotens unterbringen, was zu weitaus geringeren Kosten und einer geringeren Latenzzeit führen würde.

InfiniBand bietet serielle Verbindungen mit einer Geschwindigkeit von 250 MB/s pro Kanal. Mehrere Kanäle lassen sich zusammenschalten. 8 Kanäle (abgekürtzt „8x“) würden dann eine Bandbreite von 2GB/s liefern, bei 16 Kanälen (16x) wären es 4 GB/s. Dadurch wird eine Skalierbarkeit der Performance erreicht. Die Latenzzeiten liegen bei 6-7μs [Chri02, Rene04].

5.4.2 Kosten

Infiniband für Cluster wird beispielsweise von der Firma Mellanox vertrieben. Aktuelle Cluster mit einem Infiniband-Interconnect verwenden PCI-X Netzwerkadapter, die mit einem Switch verbunden werden. Diese Switches haben 8 bzw. 24 „4x“-Ports. Die Kosten betragen etwa $1200-$1600 pro Port. Bei sehr großen Netzen sind die Kosten sehr viel höher.

5.5 Quadrics

In diesem Kapitel werden die Interconnects QsNet und QsNetII der Firma Quadrics beschrieben. QsNet ist der teuerste und leistungsfähigste Interconnect. Er zeichnet sich gegenüber anderen Interconnects durch einen großen Funktionsumfang aus wie beispielsweise eine integrierte globale Speicherverwaltung. Einführende Beschreibungen zu QsNet und QsNetII findet man in [Petr02, Quad04, Quad03]. Die Verwendung des Interconnects in bestehenden Clustern wird in [eAl.93, FPeaCSD01a, FPeaCSD01b] beschrieben. Informationen zur Verwendung von QsNet zusammen mit anderen Interconnects findet man in [eal04].

5.5.1 Aufbau

Der Quadrics-Interconnect QsNet besteht aus zwei Hardware-Komponenten, nämlich den Netzwerk-Adaptern „Elan“ und Switches „Elite“. Diese können in einer fetten Baum-Topoligie angeordnet werden.

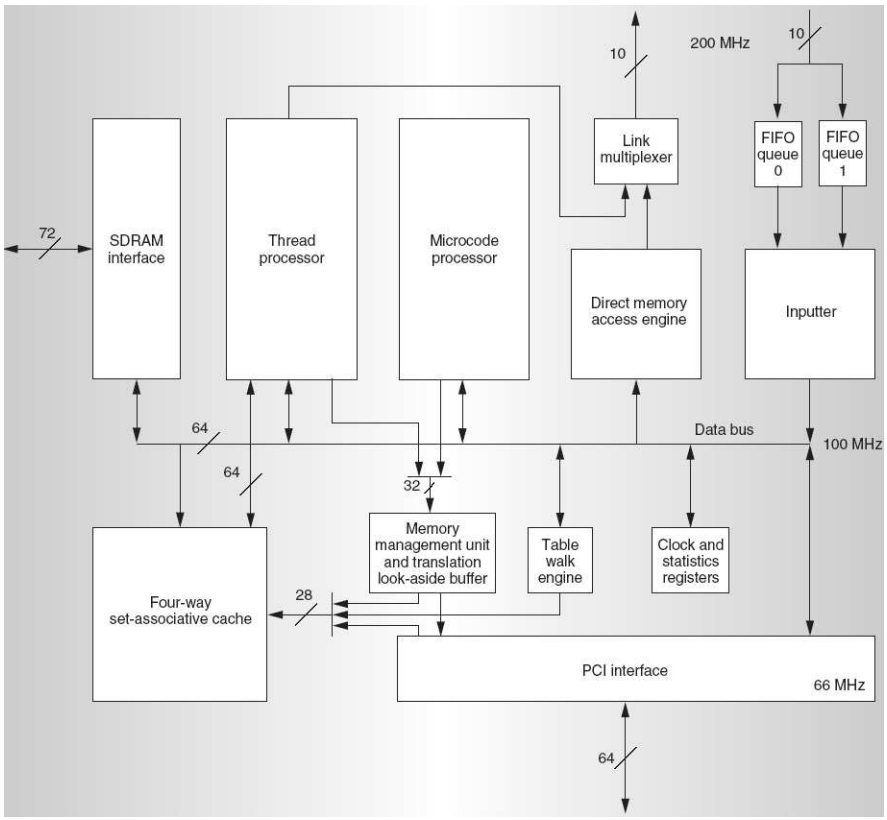


Abbildung 6: Aufbau des ELAN-Adapters, entnomen aus [Petr02]

ELAN-Netzwerk-Adapter — Der Netzwerk-Adapter Elan stellt die Verbindung eines Rechnerknotens zum Netzwerk her. Er besteht aus einem programmierbaren Nerzwerk-Interface mit eigenem Prozessor, der das Kommunikationsprotokoll implementiert. Dieser Prozessor ermöglicht das Implementieren von höheren Protokollen wie MPI. In Abbildung 6 wird der prinzipielle Aufbau des ELAN-Adapters gezeigt.

Der 32-Bit Mikroprozessor erlaubt das Ausführen von bis zu 4 Hardware-Threads, welche die ausstehenden Speicheranforderungen bearbeiten. Der erste Thread behandelt die Dateneingabe aus dem Netzwerk. Der zweite Thread funktioniert als DMA-Einheit, welche den lokalen Speicherzugriff per DMA übernimmt und die Daten zurück ins Netz schreibt. Der dritte Thread ist für das Scheduling der Threads zuständig. Ein vierter Thread übernimmt die Verarbeitung der Daten-Anforderungen auf Benutzerebene. Es können bis zu 8 Speicheranforderungen im Anforderungsfenster vorgehalten werden.

ELITE-Switch — Die „Elite"-Switches haben 8 Ports und zeichnen sich durch eine sehr niedrige Durchleite-Latenzzeit von 35ns aus. Zudem existiert eine CRC- Fehlererkennung für die Daten- und Routing-Pakete. Die Switches werden mit Kabeln einer maximalen Länge von 10m in Form eines fetten Baums verbunden. Damit ist das Verkabeln der Komponenten wesentlich einfacher als bei SCI, wo die Kabellängen auf 3-5m beschränkt sind. Bei QsNet können maximal 4096 Knoten verwendet werden.

Routing der Datenpakete — Das Routing wird vom Sender übernommen. Jeder Knoten kennt das gesamte Netzwerk, welches statisch ausgelegt ist. Diese Annahme ist bei Clustern möglich, da während des Betriebs keine Änderungen an der Konfiguration vorgenommen werden. Beim versenden eines Paketes werden den eigentlichen Daten Routing-Informationen

	QsNet	QsNetII
Bus interface	PCI 2.1	PCI-X 1.0
Peak bus bandwidth	528 MBytes/s	1064MBytes/s
QsNet link width	10 bits	10 bits
QsNet line rate	400 Mbaud	1.333Gbaud
Sustainable transfer rate	350MBytes/s	900MBytes/s
Onchip cache	4KBytes unified	32KBytes D + 16KBytes I
Local Memory	64MBytes ECC SDRAM	64MBytes ECC DDR SDRAM
Peak Memory Bandwidth	800MBytes/s	2.67 GByte/s
IO processor	100 MHz 32 bit	200 MHz 64 bit
Physical Addressing	48 bits	52 bits
Virtual Address	32 bit VA, 4K contexts	64 bit VA, 4K/16K contexts
MMU	16 entry TLB + table walk	2 x 128 entry TLB + hash table

Tabelle 2: Vergleich QsNet mit QsNet-2, entnommen aus [Quad04]

vorangestellt. Diese bestehen aus einer Abfolge von Elite-Verbindungsanweisungen. Jeder Elite-Switch wertet die erste Verbindungsanweisung aus, entfernt sie anschließend und leitet das restliche Paket zum nächsten Switch oder zum Empfängerknoten weiter. Dieses einfache Verfahren minimiert den Hardware-Aufwand der Switches und ermöglicht ein schnelles Weiterleiten. Es können einzelne Knoten als auch Gruppen von Knoten Adressiert werden.

Globaler virtueller Speicher — QsNet ermöglicht es, einen globalen virtuellen Adressraum anzusprechen. Hierbei können Daten direkt zwischen den Adressräumen kooperierender Prozesse übertragen werden. Hierfür hat die ELAN-Einheit eine MMU, welche die virtuelle Adresse unter Angabe von Thread-Prozessor, DMA- Einheit etc. in eine physikalische Adresse auf dem jeweiligen Knoten umsetzen kann. Darüber hinaus werden die Adressen auch korrekt zwischen Plattformen mit unterschiedlichem Adressformat umgesetzt (32/64 Bit, big/little endian). Der Schutz der virtuellen Thread-Adressräume bleibt bestehen.

QsNet-2 — Die schon sehr gute Leistung von QsNet wird durch QsNet-2 nochmals übertroffen. Im August 2003 wurde QsNet-2 vorgestellt. In Während QsNet PCI-2.1- Netzwerkadapter verwendet, benutzt QsNet-2 PCI-X-1.0- Adapter. In Abbildung 7 sind die beiden Interconnects gegenübergestellt. Neben der Erhöhung der Bandbreite (siehe Abbildung 8) wurde auch die Latenzzeit verringert (siehe Abbildung 7). In Tabelle 2 wurde die Antwortzeit über der Paketgröße aufgetragen, in welcher ein angefordertes Datum erhalten wird. Da die Anforderung als auch die Übertragung der Daten das Netz durchquert, entspricht dies der doppelten Latenzzeit.

5.5.2 Leistungsmerkmale

Die Latenzzeit und die Bandbreite von QsNet und QsNet-2 sind beeindruckend. Die MPI-Latenzzeit liegt bei lediglich 2.4μs bzw. 1.7 μs. Pro Kanal wird eine Bandbreite von 350 MB/s bei QsNet bzw. 900MB/s bei QsNet-2 erreicht. Es können auch mehrere Kanäle parallel betrieben werden. Dies wird angewendet beispielsweise bei sehr große Knoten, bei denen die Bandbreite weiter erhöht werden soll. Außerdem kann so auch der Einfluß des Flaschenhalses PCI-Bus verringert werden.

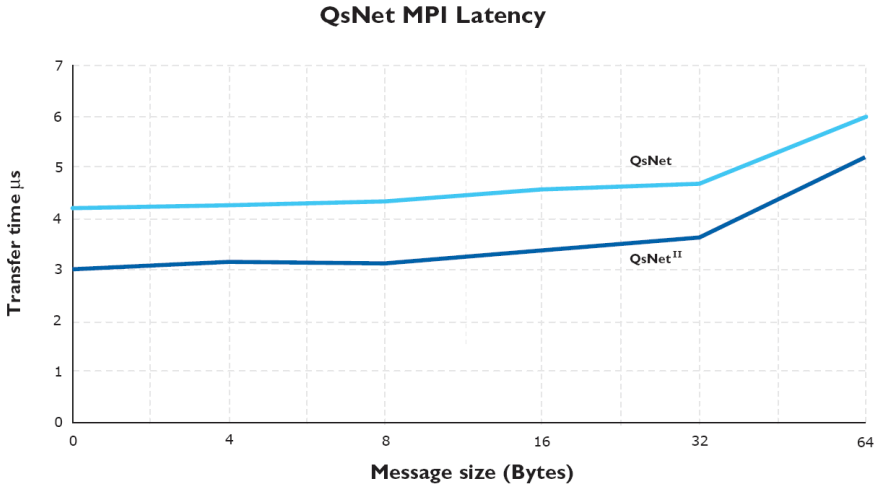


Abbildung 7: Antwortzeit in Abhängigkeit der Paketgröße von QsNet und QsNet-2, entnommen aus [Quad04].

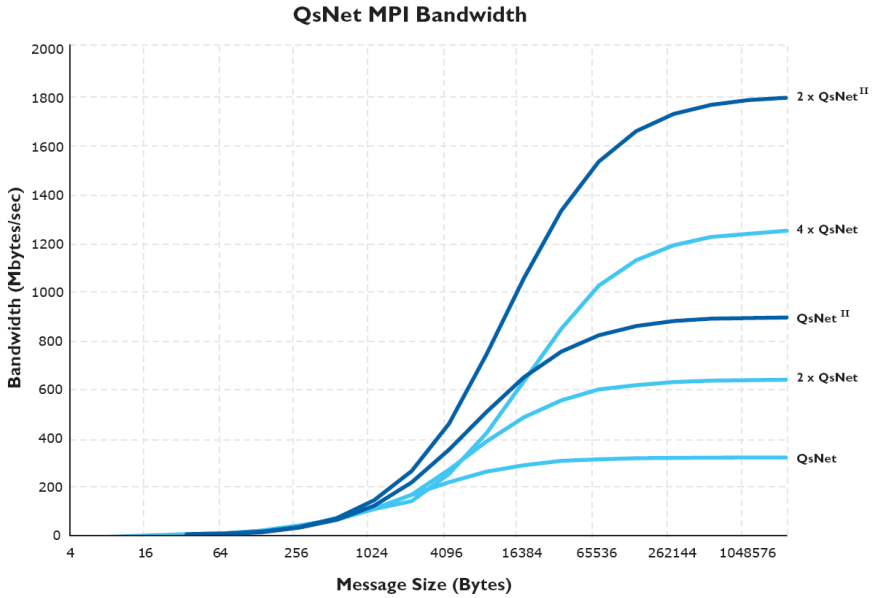


Abbildung 8: Bandbreite von QsNet und QsNet-2, entnommen aus [Quad04]

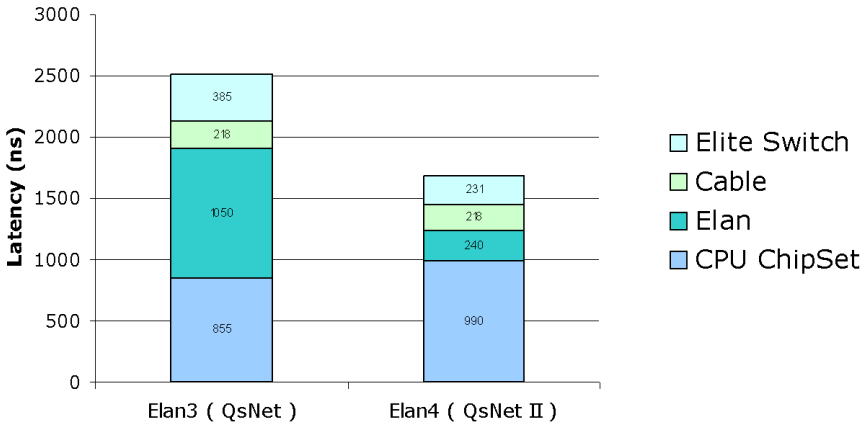


Abbildung 9: Übersicht der Anteile an der Latenzzeit bei QsNet und QsNetII

In Abbildung 9 werden die Anteile der Latenzzeiten bei QsNet und QsNetII gegenübergestellt. Laut Quadrics werden bei QsNet-2 60% der Latenzzeit durch den Transfer von CPU zur PCI-Karte verursacht. Deshalb sollen zukünftige Versionen durch Verwendung von PCI-X-2.0 oder PCI Express verbessert werden.

5.5.3 Kosten

QsNet ist der leistungsfähigste und, wenn man die Kosten pro Knoten betrachtet, auch der teuerste Interconnect. QsNet-2-Netzwerkadapter kosten pro Knoten $1385, ein 16-Port-Switch kostet etwa $8000, ein 128-Port Switch kostet $82900 (Preise entnommen aus [Meye04]). Bei einem Cluster mit 8 Knoten betragen die Kosten pro Knoten demnach $2385, bei einem Cluster mit 128 Knoten betragen sie $2033.

5.6 Vergleich der verschiedenen Techniken

Abschließend werden nun die Kenngrößen wie Latenzzeit, Bandbreite und Kosten der verschiedenen Interconnects verglichen. In Tabelle 3 sind die Bandbreiten und Latenzzeiten der vorgestellten Interconnects aufgelistet. Diese sind aufgeschlüsselt nach dem verwendeten Protokoll. Die MPI-Leistungsdaten basieren auf der freien Implementierung MPICH. Zum Vergleich mit Ethernet sind auch TCP- und Sockets-Leistungswerte der übrigen Interconnects aufgeführt. Durch den großen Overhead des TCP-Protokolls kommt es zu wesentlichen Leistungsverlusten.

In Tabelle 4 sind die Kosten für die Interconnects gegenübergestellt. Gigabit Ethernet bildet das untere Ende der Preisspanne. Die Kosten für Ethernet-Cluster werden bei großer Knotenzahl hauptsächlich durch die Kosten für die komplexen Switches bestimmt. Interessant ist die lineare Entwicklung der Kosten mit der Knotenzahl von SCI, welches ohne Switches auskommt. Das obere Ende der Preisspanne bilden QsNetII und InfiniBand. InfiniBand wird jedoch in Zukunft das potential haben, durch hohe Stückzahlen kostengünstiger zu werden.

6 Höchstleistungs-Rechner HPXC6000 am SSC Karlsruhe

Am Scientific Supercomputing Center der Universität Karlsruhe wird zurzeit ein Höchstleistungsrechner aufgebaut. Dieser besteht momentan aus 116 2-Wege Knoten mit jeweils 2 Intel

		Ethernet 1Gb	Myrinet	SCI	QsNetII	Infiniband
Treiber API	Bandbreite	-	250MB/s	350MB/s	950MB/s	950MB/s
	Latenzzeit	-	6,3μs	1,4μs	1,7μs	4,5μs
MPI	Bandbreite	-	238MB/s	223MB/s	905MB/s	763MB/s
	Latenzzeit	-	8,1μs	3,8μs	2,6μs	5,4μs
TCP/IP	Bandbreite	117,5MB/s	210MB/s	210MB/s	-	180MB/s
	Latenzzeit	31μs	46μs	18,0μs	-	100μs
Sockets	Bandbreite	-	227MB/s	255MB/s	-	471MB/s
	Latenzzeit	-	12μs	2,3μs	-	28,0μs

Tabelle 3: Latenzzeiten und Bandbreiten der Interconnects, entnommen aus [Meye04]

	Ethernet 1Gb	Myrinet	SCI	QsNetII	Infiniband
Adapter	\$75	\$595	\$1095	\$1200	\$827
Kabel 3m	\$10	\$75	-	\$185	\$117
Switch 8	\$115	\$5000	-	\$8000	\$8063 (24x)
Switch 128	\$32000	\$51200	-	\$82900	\$80000 (144x)
Cluster 8	\$979	\$10360	\$8760	\$19080	\$15615
Cluster 128	\$42880	\$136960	\$140160	\$260180	\$200832
Preis/Knoten 8	\$99	\$1295	\$1095	\$2385	\$1952
Preis/Knoten 128	\$335	\$1070	\$1095	\$2033	\$1569

Tabelle 4: Kosten der Interconnects, entnommen aus [Meye04]

Itanium2 Prozessoren und 12 GB Hauptspeicher. Diese sind mit einem Quadrics QSNet II Interconnect verbunden (ein Kanal). Bis zum Jahr 2006 soll dieser Cluster auf 218 4-Wege Knoten mit jeweils 2 ”dual core” Intel Itanium2 Prozessoren erweitert werden. Diese Knoten werden zur weiteren Steigerung der Leistungsfähigkeit dann mit einem Dual-Rail Quadrics QSNet II Interconnect verbunden.

Der Grund für die Wahl von QsNetII als Interconnect war die erreichbare geringe Latenzzeit, welche die Bearbeitung von kommunikationsintensiven Anwendungen bei großen Prozessorzahlen erlaubt. Auf dem System wurden eine Latenzzeit von 3μs und eine Bandbreite von ca. 800MB/s auf MPI-Ebene erreicht.

7 Zusammenfassung

Zu Beginn wurde die Wichtigkeit von leistungsfähigen Interconnects an einem Beispiel veranschaulicht. Ausgehend von Gigabit Ethernet, einem aus dem Serverbereich stammenden Netzwerk, wurden aktuelle Interconnects vorgestellt. Diese Interconnects unterscheiden sich wesentlich hinsichtlich Leistungsfähigkeit, Skalierbarkeit, Kosten, der verwendeten Netzwerk-Topologie und der Verfügbarkeit von Software. Deshalb ist es sehr wichtig, das für die jeweilige Anwendung geeignete Produkt auszuwählen. Bei Anwendungen mit geringer Kommunikation ist möglicherweise die Bandbreite und Latenzzeit von Gigabit Ethernet ausreichend und es können Kosten gespart werden. Anwendungen mit hoher Kommunikation oder Cluster mit sehr vielen Knoten können hingegen nur mit einem leistungsfähigen Interconnect effizient arbeiten.

Literatur

[Andj03] Mario Andjelic. DMD: Network Device Driver Architecture for High-Performance Systems. In *7th International Conference on Telecommunications: ConTEL 2003*. ConTEL, 2003, S. 6.

[BaPa93] Debashis Basak und Dhabaleswar K. Panda. Scalable Architecture with K-Ary N-Cube Cluster-C Organisation. *IEEE*, April 1993, S. 8. The Ohio State University, Columbus, OH.

[BMBBe] Jason J. Hill Brett M. Bode und Troy R. Benjegerdes. Cluster Interconnect Overview. *IEEE*, S. 7.

[CDea98] Princeton University Cezary Dubnicki et al. MYRINET COMMUNICATION. *IEEE Micro* Band 1998, 1998, S. 3.

[Chri02] Mellanox Technologies Chris Eddington. INFINIBRIDGE: AN INFINIBAND CHANNEL ADAPTER WITH INTEGRATED SWITCH. *IEEE Micro* Band 2002, 2002, S. 9.

[eAl.93] L.M. Mackenzie et Al. CORBA: A High Performance Interconnection for Large Multicomputers. *IEEE* Band 1993, 1993, S. 3.

[eal95] Nanette J. Baden et al. Myrinet: A Gigabit-Per-Second Local Area Network. *IEEE Micro* Band 1995-2, 2 1995, S. 8. Myricom Inc.

[eal04] Weikuan Yu et al. Efficient and Scalable Barrier over Quadrics and Myrinet with a New NIC-Based Collective Message Passing Protocol. *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS04)* Band 2004, 2004, S. 8.

[EFFe02] Salvador Coll Eitan Frachtenberg, Fabrizio Petrini und Wu-Chun Feng. Gang Scheduling with Lightweight User-Level Communication. *U.S. Government Work* Band 2002, 2002, S. 7.

[FaOn00] Paul A. Farrell und Hong Ong. Communication Performance over a Gigabit Ethernet Network. *IEEE* Band 2000, 2000, S. 9. Department of Mathematics and Computer Science, Kent State University, Kent, OH 44242.

[FPeaCSD01a] Computer Fabrizio Petrini et al. und Los Alamos National Laboratory Computational Sciences Division. Performance Evaluation of the Quadrics Interconnection Network. *IEEE* Band 2001, 2001, S. 9.

[FPeaCSD01b] Computer Fabrizio Petrini et al. und Los Alamos National Laboratory Computational Sciences Division. The Quadrics Network (QsNet): High-Performance Clustering Technology. *IEEE* Band 2001, 2001, S. 6.

[GCSc02] Marco Ehlert Giuseppe Ciaccio und Bettina Schnor. Exploiting Gigabit Ethernet Capacity for Cluster Applications. *IEEE* Band 2002, 2002, S. 10. Universita di Genova, Italy und Institut für Informatik, Universität Potsdam, Germany.

[GFDe00] Michel Lavoie Ghassan Fadlallah und Louis-A. Dessaint. Parallel Computing Environments and Methods. *IEEE* Band 2000, 2000, S. 6. Groupe de Recherche en Electrique de Puissance et Commande Industrielle (GREPCI), Montreal, Quebec, Canada.

[GmbH04] Swyx Solutions GmbH. Überblick über Die Verschiedenen PCI Karten- und Slot-Typen und Deren Kombinationsmöglichkeiten (Knowledge Base 2471). Technischer Bericht, Swyx Solutions GmbH, Joseph-von-Fraunhofer-Str. 13a, 44227 Dortmund, 2004.

[Mach99] Jens Mache. An Assessment of Gigabit Ethernet as Cluster Interconnect. *IEEE*, 1999, S. 7. Lewis and Clark College Portland, OR 97219, USA.

[Meye04] Stefan Meyer. Cluster Interconnects: Verfuegbarkeit, Leistung und Verbreitung. *Seminararbeit*, 2004.

[Myri04] Myricom. Myrinet Overview. Webpage, Miricom Ltd., http://www.myricom.com/myrinet/overview/, Juli 2004.

[NaIy96] Chitra Natarajan und Ravishankar K. Iyer. Measurement and Simulation Based Performance Analysis of Parallel I/O in a High-Performance Cluster System. *IEEE* Band 1996, 1996, S. 8. University of Illinois, USA.

[Neal02] CA Neal Bierbaum Sandia National Laboratories, Livermore. MPI and Embedded TCP/IP Gigabit Ethernet Cluster Computing. *IEEE* Band 2002, 2002, S. 2.

[OmPa97] Knut Omang und Bodo Parady. Scalability of SCI Workstation Clusters, a Preliminary Study. *IEEE* Band 1997, 1997, S. 6. Department of Informatics, University Oslo and Sun Microsystems Inc.

[PaHa92] Manish Parashar und Salim Hariri. A Requirement Analysis For High Performance Distributed Computing over LANs. *IEEE*, 1992, S. 10. Electrical and Computer Engineering Syracuse University.

[Petr02] Fabrizio Petrini. THE QUADRICS NETWORK: HIGH-PERFORMANCE CLUSTERING TECHNOLOGY. *IEEE* Band 01-2002, JANUARY 2002, S. 5.

[Quad03] Quadrics. *QsNetII Installation and Diagnostics Manual*. Quadrics Ltd., http://www.quadrics.com/, Document Version 0. Auflage, Dezember 2003.

[Quad04] Quadrics. QsNet High Performance Interconnect. Technischer Bericht, http://doc.quadrics.com/Quadrics/QuadricsHome.nsf/DisplayPages/3A912204F260613680256DD9005122C7, November 2004.

[Rene04] Wolfgang Rehm Rene Grabner, Frank Mietke. An MPICH2 Channel Device Implementation over VAPI on InfiniBand. *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS04)* Band 04, 2004, S. 8. Chemnitz University of Technology, Faculty of Computer Science.

[RoCh99] Sumit Roy und Vipin Chaudhary. Evalution Of Cluster Interconnects for a Distributed Shared Memory. *IEEE*, 1999, S. 7. Parallel and Distributed Computing Laboratory, Wayne State University, Detroit, Michigan.

[Tess03] Daniele Tessera. Performance Analysis of an IBM Supercluster. *Computer Society, Proceedings of the Eleventh Euromicro Conference on Parallel, Distributed and Network-Based Processing (Euro-PDP03)* Band 03, 2003, S. 8. Dipartimento di Informatica e Sistemistica Universit adegli Studi di Pavia, Italy.

[uLew01] Jens Mache und Lewis Clark College Portland USA. An Assessment of Gigabit Ethernet as Cluster Interconnect. *IEEE* Band 2001, 2001, S. 7.

[Wilb03] Amy Apon Amd Larry Wilbur. AmpNet - A Highly Available Cluster Interconnection Network. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS03)*, Band 2003. University of Arkansas and Belobox Networks, IEEE, 2003, S. 10.

[Zhao03] Yu Chen Zhaoyang Li, Yi Zhang. Design and Implementation of a High-Performance Interconnection Network: TH-Net. *IEEE* Band 2003, 2003, S. 4. Department of Computer Science and Technology, Tsinghua University, Beijing, China.

Abbildungsverzeichnis

Tabellenverzeichnis

Roaming in und zwischen Funknetzwerken

Eric Stiegeler

Kurzfassung

Diese Seminararbeit behandelt Roaming in Funknetzwerken. Es wird zwischen Roaming in verschiedenen Schichtebenen und Funktechniken unterschieden. Die Umsetzung des Roaming im RZ Karlsruhe, innerhalb von Baden-Württemberg und im DFN und die Probleme und Lösungen des Roaming auf Layer 2 und Layer 3. Es wird zwischen den Begriffen „Roaming“ und „Handover“ bei Funknetzen mit integriertem Roaming/Handover und Funknetzen ohne diese Leistung unterschieden. Anschließend werden Funknetze, die benötigten Standards und Lösungsansätze betrachtet.

1 Einleitung

Roaming in Funknetzwerken soll die Kommunikation zwischen verschiedenen Technologien und verschiedenen Schichten ermöglichen. Die Zielsetzung des Roamings ist, dem Benutzer eine ständige Erreichbarkeit anbieten zu können. Diese ständige Erreichbarkeit bietet dem Benutzer mehr Mobilität, Komfort und Kompatibilität. Die Erreichbarkeit wird über das Roaming zwischen den verschiedenen Funktechniken und Layern gewährleistet. Einige Funknetze, wie z.B. GSM und UMTS bieten bereits eine solche Funktionalität. Bei WLAN, WiMAX und zwischen WLAN/WiMAX und GSM/UMTS ist Roaming nur durch zusätzliche Standards und Authentifikation, die einen sicheren Wechsel in unsicheren Netzen bieten, möglich.

2 Begriffsbestimmung

2.1 Roaming vs. Handover

- Unter **„Roaming im GSM“** [t07b] versteht man, dass die Benutzer an beliebigen Orten anrufen und angerufen werden können. Der Benutzer kann unter derselben Nummer auch Netze nutzen, mit denen er keinen Vertrag hat. Mit **„Handover im GSM“** ist die Bewegung des Benutzers von einer Zelle zur nächsten Zelle gemeint, während ohne Unterbrechung der Verbindung, die Verantwortung an die jeweils nächste Zelle übergeben wird. Die Reichweite einer solchen Zelle beträgt bis zu 37 km. Es gibt zwei verschiedene Handover:

  1. **Intracell Handover**:
     - aufgrund der Signalqualität wird innerhalb einer Zelle auf eine andere Frequenz umgeschaltet
  2. **Intercell Handover**:
     - Frequenz und Zelle werden gewechselt
     - Internes Handover: Wechsel zwischen Zellen eines Base Station Controller

  - Externes Handover: Wechsel zwischen Zellen verschiedener Base Station Controller

- **WLANs** (Wireless LAN) basieren auf dem 1997 vom Institute of Electrical and Electronics Engineers (IEEE) definierten Standard IEEE 802.11. **„Roaming in 802.11“** bezeichnet die Situation, dass der mobile Client erkennt, die Sendeleistung eines Accesspoints ist zu niedrig und deshalb nach einem neuen Accesspoint entweder Aktiv (durch Versenden spezieller Frames) oder Passiv (durch Mithören) gesucht wird. Der mobile Client registriert sich, sobald er einen geeigneten Accesspoint gefunden hat. Dieser neue Accesspoint muss Zellenwechselinformationen im Netz bekannt geben. Dieser Roaming-Mechanismus entspricht im Grunde dem Handover in Mobilfunknetzen. Das Roaming in Mobilfunknetzen entspricht am ehesten „Roaming zwischen WLAN's von unterschiedlichen Betreibern“.

3 Funknetze

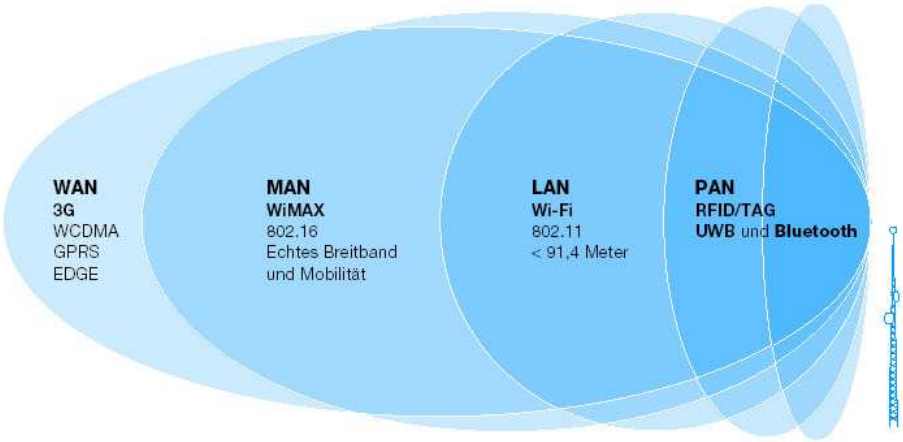


Abbildung 1: Einteilung der verschiedenen Netze

3.1 UMTS (Universal Mobile Telecommunications System)

UMTS ist der Mobilfunkstandard der 3. Generation. Es soll neben Sprachübertragung auch schnelle Datenübertragung (z.B. für Multimedia-Anwendungen) bieten. Um den zusätzlichen Ausbau von Antennen in Ballungsgebieten einzuschränken und dadurch Kosten einzusparen, aber trotzdem genügend Kapazitäten für den Benutzer anbieten zu können, kann Roaming zwischen UMTS und WLAN (siehe 6) eingesetzt werden [vSeb02] .

3.2 WiMAX (Worldwide Interoperability for Microwave Access)

WiMAX ist eine Technik für drahtlose Breitband-Internetzugänge nach dem 802.16-Standard. Als Standards sind „802.16a“ zwischen 2 und 11 Ghz, „802.16b“ zwischen 5 und 6 Ghz und „802.16e“ als Mobile Wireless (Handy) Zugang zu 802.16 geplant.

Theoretisch sind Übertragungsraten von bis zu 70Mbit/s und Reichweiten bis zu 50 km möglich. WiMAX könnte somit eine Brücke zwischen DSL und UMTS bilden. Es unterstützt Point-to-Multipoint (eine Verbindung zu mehreren Stationen wird aufgebaut).

Die Markteinführung von WiMAX wird unterstützt durch das WiMAX Forum. (z.B. geplant ist die Markteinführung in Korea 2006) Intel ist Mitglied dieses Forums und entwickelt gerade eine WiMAX Schnittstelle die Teil von Intel Centrino wird.

Geplant ist, dass WiMAX in drei Phasen eingeführt:

1. **Phase:** WiMAX-Technologie (IEEE 802.16-2004) wird über Außenantennen bereitgestellt
2. **Phase:** umfasst Innenantennen, die den Serviceanbietern die Installation beim Benutzer erleichtern
3. **Phase:** geplant für das Jahr 2006 WiMAX Hardware, die auf IEEE 802.16e Spezifikationen basiert

Vorteile von WiMAX sind:

- Non Line of Sight (NLOS) für drahtlose Zugänge
- Zellengröße ähnlich wie bei UMTS
- 4 Service Klassen einschließlich Voice over IP und MPEG Video
- Hohe Bandbreite
- Vorbereitet für High Quality Voice Service (Sprachdienste)

WiMAX ist sehr gut für Breitband-Internet geeignet. Der größte Unterschied liegt in der Struktur des PLMN (Public Land Mobile Network). Im Moment wird ein großer Hype um WiMAX gemacht, allerdings beginnt der Massenmarkt frühestens 2006. Es hat aber Potential für einen neuen großen Telekommunikationsmarkt. Die Lizenzvergabe für den Frequenzbereich durch die Regulierungsbehörde ist noch nicht erfolgt und somit noch völlig ungeklärt.

4 Standards

4.1 Mobile IP

Mobile IP [Perk98] soll ermöglichen, dass der Host immer unter derselben IP Adresse erreichbar ist und die Anbindung an das Internet unterbrechungsfrei ist. Dieser unterbrechungsfreie Ortswechsel innerhalb des Internets, wird **Seamless Roaming** genannt. Das bedeutet, dass keine Applikation beendet werden muss oder dass keine Neukonfiguration notwendig ist.

Mobile IP kann folgendermaßen realisiert werden:

1. Im Home Network:
   - MN (Mobile Node) erhält ein Advertisment vom HA (Home Agent) mit IP Adresse des HA
   - MN hat die IP Adresse des HN (Home Network)
   - HA ist das Gateway ins Internet
2. Im Foreign Network:

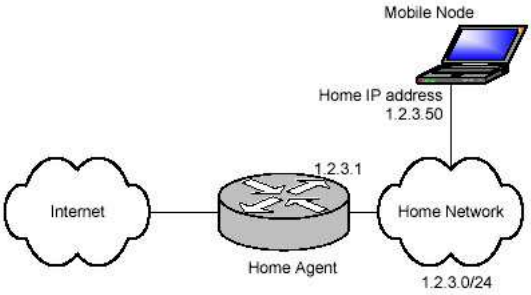


Abbildung 2: Situation im Home Network [vSeb02]

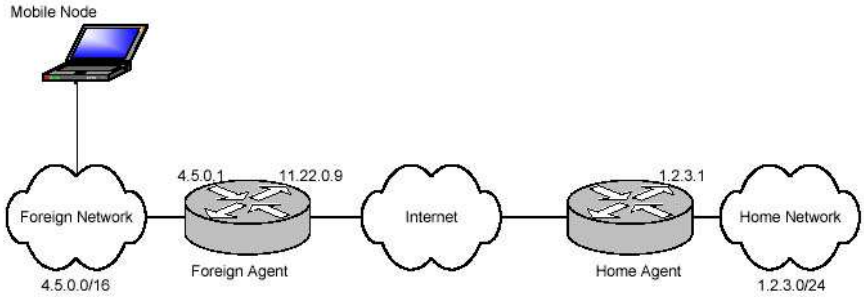


Abbildung 3: Situation im Foreign network [vSeb02]

- MN erkennt anhand der abgelaufenen Lebenszeit des Advertisment die Bewegung
- MN erkennt den FA (Foreign Agent) anhand der Advertisment Message, über die die MN nun den FA und den COA (Care-of Adress) kennt

- MN sendet einen Registration Request an den HA über den FA
- HA sendet den Registration Reply an den FA, den der FA an die MN weiter gibt
- HA baut einen Tunnel zu COA auf, um die IP Pakete für die MN umzuleiten
- MN sendet Pakete zum Correspondent Note (CN), die CN sendet Pakete zum HA
- im HN tunnelt der HA die Pakete zur COA und damit zum FA
- FA gibt die Pakete an die MN weiter
- diese Art des Routings wird **Triangular Routing** genannt

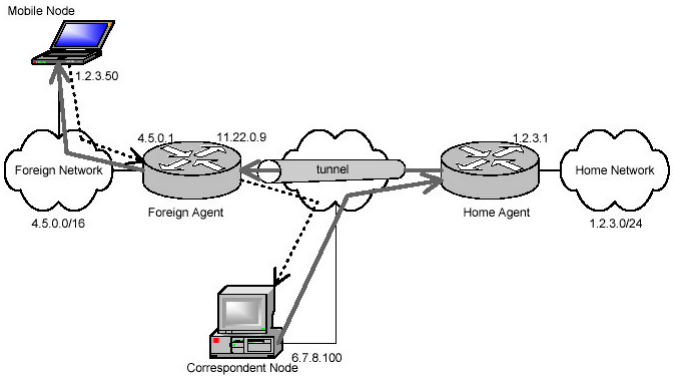


Abbildung 4: Situation im Foreign network [vSeb02]

4.2 IPv6

Die Entwicklung von IPv6 war notwendig, um den kleinen Adressraum von IPv4 zu erweitern. Da immer mehr Geräte in der Zukunft eine eigene IP benötigen wie zum Beispiel Mobile Endgeräte und Haushaltsgeräte. Die Adressgröße bei IPv6 beträgt 128-Bit mit neuem Header-Format aus einem zwingenden Basis-Header und ein oder mehrere Zusatz-Header. Unicast, Multicast und › Header sind kleiner und an

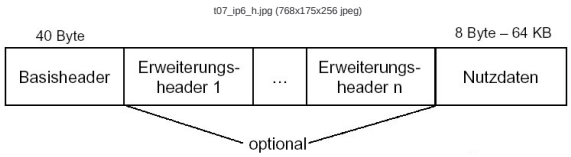


Abbildung 5: IPv6 Header

Das IPv6-Protokoll ist erweiterbar. Außerdem bietet IPv6 QOS-Mechanismen. Bei IPv6 sind „sanfte Wechsel“ von einem Foreign Agent zum nächsten möglich, indem die ankommende Pakete des letzten Foreign Agent zum nächsten geroutet werden. Das ist möglich, da jeder IPv6 Knoten eine automatische Konfiguration beherrscht. Durch diesen Mechanismus ist die Erlangung einer CoA bereits in IPv6 integriert.

4.2.1 IPsec

Um die Schwächen des Internetprotokolls (IP) zu beheben wurde IPsec 1998 entwickelt. IPsec stellt eine Sicherheitsarchitektur zu Verfügung. Es soll vor Replay-Angriffen (ein Angreifer kann nicht durch Abspielen eines vorher mitgeschnittenen Dialogs die Gegenstelle zu einer wiederholten Aktion verleiten) schützen. Zur automatischen Schlüsselverwaltung für IPsec dient das Internet Key Exchange (IKE) Protokoll. Für den Austausch von Schlüsseln über ein unsicheres Netzwerk wird der Diffie-Hellman-Schlüsselaustausch verwendet.

4.3 AAA-System (Authentication, Authorization und Accounting)

AAA steht für die Zusammenfassung eines Sicherheitskonzepts. Die Verwaltung und Speicherung von Benutzerdaten findet an einer zentralen Stelle statt. Mit diesen Daten kann sich der Benutzer authentifizieren. Der Zugriff von außen zum Netzwerk wird über eine VPN Verbindung hergestellt, über die sich der Benutzer vor dem Zugang zum Netzwerk authentifizieren muss. Das RADIUS-Protokoll übernimmt die Authentifizierung und Verschlüsselung, sowie das Accounting. Der Anfang und das Ende der Benutzung einer Leistung wird vom RADIUS-Server protokolliert und zu Abrechnungszwecken herangezogen.

4.3.1 RADIUS (Remote Authentication Dial-In User Service)

Der RADIUS-Server ist ein Dienst der zur Authentifizierung von Clientgeräten oder Diensten gegenüber Datenbanken genutzt wird. Er dient zur zentralen Authentifizierung von Einwahlverbindungen.

4.4 VPN Konzentrator

VPN-Gateways werden auch VPN (Virtual Private Network) Konzentrator genannt. Der Gast oder Mitarbeiter baut eine Verbindung (verschlüsselter Tunnel) zwischen seinem Endgerät und dem VPN-Gateway auf.

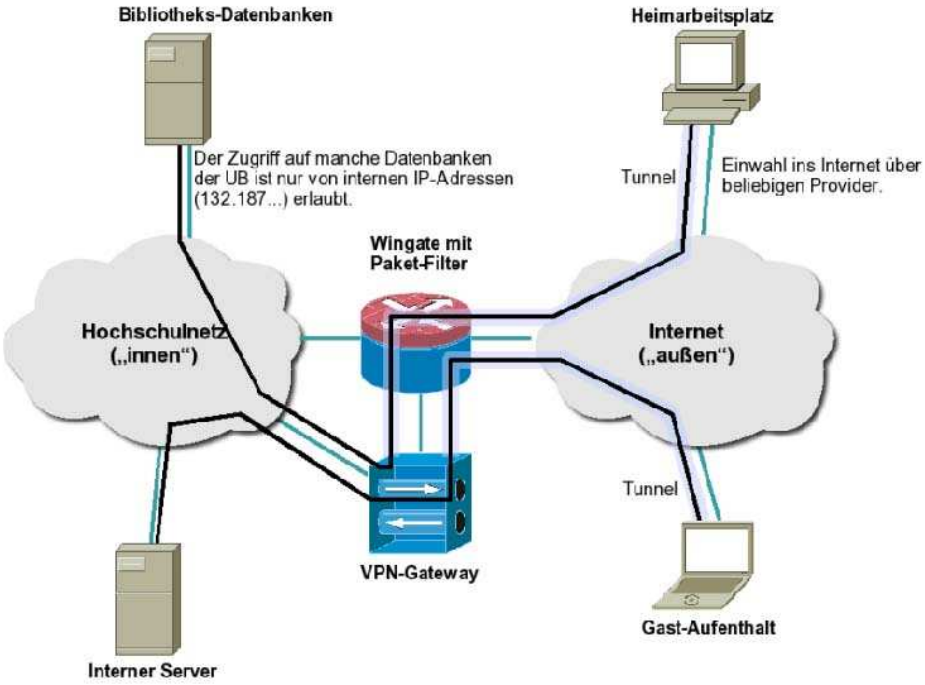


Abbildung 6: Funktionsweise eines VPN-Gateways [Würz]

Durch dieses VPN wird gewährleistet, dass Daten verschlüsselt übertragen werden und berechtigte Benutzer über den Umweg zum VPN Konzentrator interne Dienste verwenden können.

4.5 802.1x-Standard

Der 802.1x-Standard [t07c] wurde für den gesicherten Netzzugang entwickelt. Die Zugangskontrolle erfolgt über Ports. Es gibt kontrollierte und unkontrollierte Ports. Die angeschlossenen Geräte werden bei unkontrollierten Ports auf bestimmte Netzwerkadressen beschränkt. Hingegen erlauben kontrollierte Ports den Geräten mit allen Netzwerkgeräten zu kommunizieren. Ein Client kann einen solchen kontrollierten Port verwenden, wenn er sich authentifiziert hat. Die Authentifizierung des 802.1x-Protokolls sieht zwei Rollen vor. Der Authentifizierer kontrolliert den Client und entscheidet ob der Zugriff erteilt wird. Der Bittsteller möchte Zugriff erhalten. Ein Accesspoint könnte zum Beispiel ein Authentifizierer sein. Allerdings eignen sich RADIUS-Server (Remote Authentication Dial-In User Service) besser zur Authentifizierung. Zur Authentifizierung wird das EAP (Extensible Authentication Protocol) oder den PEAP (Protected-EAP) verwendet. Das EAP/PEAP bietet eine ganze Auswahl von Authentifizierungsmethoden. Normalerweise wird das EAP-TLS (EAP-Transport Layer Security) eingesetzt. Dabei wird der EAP-gesicherte Austausch mit TLS (verwandt zu SSL) verschlüsselt.

Der Ablauf der Authentifizierung ist:

1. Über den unkontrollierten Port beginnt der Verbindungsaufbau vom Client mit dem Accesspoint.

2. Der Client antwortet mit einer Identifikation.
3. Die Identifikation des Clients wird vom Accesspoint über RADIUS an den Authentifizierer weitergeleitet.
4. Anhand der Identifikation kontrolliert der RADIUS-Server das spezifizierte Konto und ermittelt die notwendigen Anmeldeinformationen. Mit dieser Information wird eine Anmeldeinformations-Anfrage an den Client geschickt.
5. Der Client sendet über den unkontrollierten Port die Anmeldeinformation an den Accesspoint.
6. Die Anmeldeinformationen werden von dem RADIUS-Server ausgewertet. Sind die Anmeldeinformationen richtig, so wird ein verschlüsselter Authentifizierungsschlüssel an den Accesspoint gesendet, den nur der Accesspoint entschlüsseln kann.
7. Der Schlüssel wird vom Accesspoint entschlüsselt und verwendet, um einen neuen Schlüssel für den Client zu erzeugen. Anschließend wird der neue Schlüssel an den Client gesendet und dort zur Verschlüsselung des globalen Authentifizierungsschlüssels benutzt.

In regelmäßigen Abständen wird ein neuer globaler Authentifizierungsschlüssel vom Accesspoint generiert und an den Client gesendet, um das Problem der langlebigen festen Schlüssel vom 802.11-Standard zu beseitigen.

4.6 802.11i

Bei WEP (Wired Equivalent Privacy) war der Secret Key mit 40 Bit zu kurz, die Keys mussten manuell an jeder Station eingetragen werden, boten keine Schlüsselverwaltung und es gab keine Benutzerauthentifizierung. Zur Verbesserung der WLAN Sicherheit wurde der 802.11i-Standard entwickelt. Beim Datentransport findet eine Erkennung von Wiederholungen der Pakete, Authentisierung der Nachrichtenquelle, Vermeidung von Schlüsselwiederholung und starke Verschlüsselungstechnik statt. Mit dem Robust Security Network (RSN) sollen die bisherigen Funktionen verbessert werden. Vorraussetzungen für ein solches RSN sind eine bessere Datenverschlüsselung durch TKIP (Temporal Key Integrity Protocol) um mit RC4 basierter Hardware höhere Sicherheitsanforderungen zu erfüllen und WRAP (Wireless Robust Authenticated Protocol) das auf AES und CCMP (CounterMode with CBC-MAC Protocol) basiert. Ein besseres Management von gesicherten Verbindungen wird mit 802.1x (siehe 4.5) basierte Authentisierung, Schlüsselverwaltung und durch RSN bei der Verhandlungen zur Errichtung des Sicherheits-Kontexts erreicht. TKIP maskiert die Schwächen von WEP und ist als „Hülle“ für den WEP entwickelt worden. CCMP wurde extra für 802.11i entworfen und benötigt einen 128 Bit Schlüssel. Für die Schlüsselkonfiguration ist 802.1x zuständig.

Die Kommunikation wird in 3. Phasen unterteilt:

1. **Phase: Discovery**
   - Erkennen von Kommunikationspartnern
   - Accesspoint informiert die Station
2. **Phase: Authentification basierend auf 802.1x**
   - Zugangskontrolle und zentrale Verwaltung im AS
   - Entscheidung der Station über den Verbindungsaufbau

- gegenseitige Authentifizierung zwischen AS und Station
- Erzeugung des Master Keys
- Erzeugung des Authorization Token aus dem Master Key
- AS übergibt den Session Key (PMK) an den Accesspoint

3. **Phase: Key Management mittels 802.1x**

- Bindung des Session Keys zur Station und zum Accesspoint
- Bestätigung des Erhalts des PMK vom Accesspoint und der Station
- Erzeugung der neuen PTKs

5 Layer 2 Roaming

Ein LAN (Local Area Network) besteht aus zwei oder mehr in einem Netzwerk angeordneten Rechnern, die sich innerhalb einer physikalischen Broadcastdomain befinden. Diese Rechner sind so miteinander verbunden, dass ein Broadcast jeden dieser miteinander verbundenen Rechner erreichen kann. Bei VLAN's (Virtual Local Area Network) werden die Endgeräte der Benutzer zu logischen Gruppen zusammengefasst, unabhängig von ihrem physischen Standort.

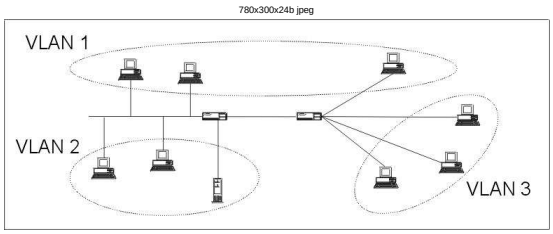


Abbildung 7: VLAN

Durch VLAN's werden Accesspoints aus verschiedenen LAN's (Local Area Network) zusammengefasst. Das VLAN ist die logische Sichtweise des physikalischen LANs, das heißt die verschiedenen LANs werden in ein einheitliches VLAN abgebildet. Mehrere Accesspoints eines bestimmten Gebietes bilden ein VLAN. Das IAPP (Inter Access Point Protocol) ermöglicht die Kommunikation zwischen Accesspoints. Um ein Roaming von Daten zwischen zwei Zellen, die über eine WLAN-Bridge verbunden sind, bietet das IAPP Protokoll aus 802.11f die Möglichkeit. Die automatische Konfiguration der Accesspoints wird vom Protokoll unterstützt. Es sendet die Daten des Clients der den Accesspoints verlässt an den nächsten Accesspoints. Mit dem Einsatz des IAPP wird auch die Interoperabilität der Accesspoints von verschiedenen Herstellern erhöht. Somit ist das Roaming innerhalb des VLAN zwischen den Accesspoints gewährleistet. Allerdings kann der Benutzer nicht zwischen den verschiedenen VLAN's roamen. Die Authentifizierung erfolgt über RADIUS-Server (siehe 4.5). Er erkennt, ob eine „externe" (von zu hause) VPN-Verbindung (Virtual Private Network) oder eine „interne" (beim RZ Karlsruhe über WLAN) VPN-Verbindung vorliegt und routet entsprechend die Pakete. Dies stellt im Moment allerdings kein großes Problem dar, da der normale Benutzer mit einem Laptop einen relativ geringen Bewegungsradius ausweist und somit der Fall, dass ein Roaming zwischen zwei VLAN's nötig wird, selten auftritt. Sobald aber genügend Mobiltelefone oder PDA's (Personal Digital Assistant) mit WLAN auf dem Markt sind und diese auch am Campus genutzt werden, wird das nicht vorhandene Roaming zwischen den verschiedenen VLAN's für den Benutzer auffällig, sobald er durch den großen Bewegungsradius mit dem Handy ständig verbunden sein will. Um die Roamingfunktionalität anbieten zu können, müsste Mobile IP (siehe 4.1) oder IPv6 (siehe 4.2) auf Layer 3 realisiert werden.

6 Layer 3 Roaming

VPN's werden üblicherweise mit kabelgebundenen Technologien verwendet. Dieses Konzept wird durch MVPN (Mobile Virtual Private Network) [P.M.03] auf kabellose Technologien erweitert.

Mit MVPN's kann einfacher und schneller Zugang zu dem Netzwerk gewährleistet werden. UMTS und WLAN haben unterschiedliche Stärken und Schwächen, die sich sehr gut ergänzen würden. Nachteile von WLAN's sind die begrenzte Reichweite und das Fehlen eines Mobility Management. Allerdings können WLAN's die Gesprächskapazität erhöhen, indem sie an Orten mit hoher Konzentration von Nutzern eingesetzt werden. Dadurch lässt sich das UMTS Netz viel effizienter planen. Das WLAN verbirgt die Details zum 802.11-Netzwerk und implementiert die benötigten UMTS Protokolle.

Es bestehen zwei Lösungsvorschläge für die Architektur:

1. **Tightly-Coupled Interworking** Bei diesem Entwurf wird das WLAN direkt in das 3G-Netz integriert und gleichzeitig ist das WLAN auch mit UTRAN verbunden. Das WLAN emuliert Zugangsfunktionen des UMTS Netzes.

   Nachteile von „Tightly-Coupled“:

   - Entwurf benötigt, bedingt durch die komplexe Lösung, eine lange Entwicklungszeit.
   - Außerdem erfordert der Entwurf auch zusätzliche Standardisierungen.
   - Das UMTS Interface wird offen gelegt, dies kann zu Sicherheitsproblemen führen.
   - Spezielle UMTS-Bauteile müssen neu entwickelt werden.

2. **Loosely-Coupled Interworking**

   Das WLAN ergänzt als paketbasiertes Netzwerk das 3G-Netz. Dadurch ist es möglich die AAA-Möglichkeiten (siehe 4.3) weiterverwenden zu können. Die Implementierung kann auf vorhandener Technologie erfolgen. Dabei werden die Daten komplett zwischen dem 802.11 und dem UMTS Netz aufgeteilt, wobei die Daten nie über das UMTS Core Netzwerk gehen.

   Vorteile von „Loosely-Coupled“:

   - Entwurf ist von 802.11 und UMTS unabhängig
   - es sind keine großen finanziellen Investitionen notwendig

Der Entwurf „Loosely-Coupled Interworking“ besitzt wesentlich mehr Vorteile als „Tightly-Coupled“. „Loosley-Coupled“ erlaubt eine unabhängige Entwicklung von WLAN und UMTS. Dabei kann der UMTS Betreiber von der Entwicklung des WLAN ohne großes eigenes finanzielles Engagement profitieren. Durch Abschluss von Roamingabkommen mit anderen Partnern wird ein einheitlicher Zugang geschaffen. Im Gegensatz zu „Tightly-Coupled“ erlaubt „Loosely-Coupled“ dem Betreiber Leistungen des eigenen Accesspoints anzubieten, während gleichzeitig eine Zusammenarbeit über Roamingabkommen mit einem UMTS Betreiber bestehen kann. Außerdem unterstützt der Entwurf eine integrierte Authentifizierung und eine „unified“ Abrechnung, so dass der UMTS Anbieter Service Bundel aus 802.11 und UMTS anbieten kann. Durch die Integration von GPRS, PLMN (Public Land Mobile Network), IPv4 und RADIUS in den UMTS Core, ist es relativ einfach WLAN Technologie zu integrieren. Die „Seamless Mobility“ zwischen WLAN und UMTS wird durch die Integration von Mobile IP Funktionen in den GGSN ermöglicht. Mobile IP (siehe 4.1) bietet IP Layer 3 Mobility Management an, macht die Bewegung des Benutzers für Anwendungen über die Subnetze

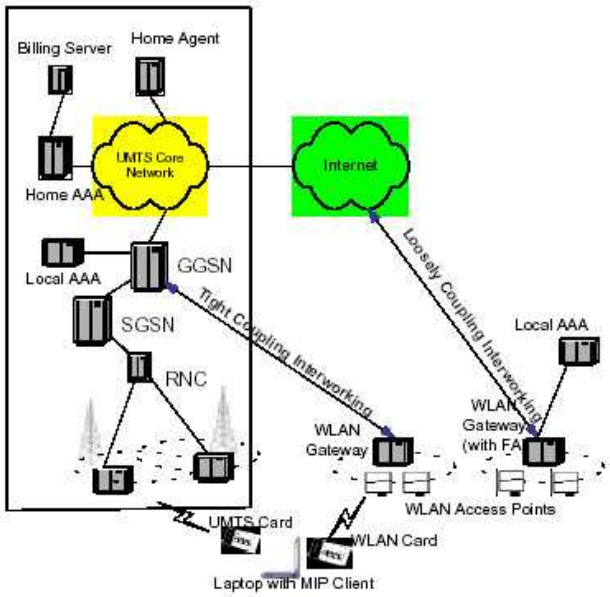


Abbildung 8: Tightly-Coupled und Loosley-Coupled [vSeb02]

sichtbar und bietet eine gute Authentifikation. Die MVPN Verbindung ist unabhängig von der darunter liegenden Zugangstechnologie. Der FA (Foreign Agent) gehört zum GGSN des UMTS Netzes und in den Edge-Router oder in das WLAN Gateway. IPsec 4.2.1 bietet über Layer 3 Sicherheit bei der Authentifizierung des Benutzers und der Verschlüsselung der Daten. Im Protokoll Stack ist IPsec (siehe 4.2.1) über Mobile IP angesiedelt, um Sicherheit und „Seamless MVPN“ anbieten zu können.

End-to-End Tunnel: Der Betreiber kann bei diesem Modell keine zusätzlichen Leistung wie zum Beispiel „quality of service“ anbieten.

Network-based MVPN's bestehen aus zwei Teilen. Zum einen aus einem benutzerinitalisierten Tunnel der beim Core Network des Betreibers endet und einem statisch konfiguriertem Tunnel vom Core Network zum Benutzer Enterprise Network. Dieses Modell ist sehr gut skalierbar und ermöglicht dem Betreiber das Anbieten von weiteren Leistungen wie zum Beispiel Web Caching.

7 Roaming im BelWü

Die Realisierung des Roamings zwischen wissenschaftlichen Einrichtungen in Baden-Württemberg erfolgt über einen anderen Ansatz als das DFNRoaming (siehe 8). Der Benutzer baut im fremden Netz einen Tunnel auf zur eigenen wissenschaftlichen Einrichtung. Dieser Aufbau erfolgt über den eigenen VPN Konzentrator (siehe 4.4) am eigenen RADIUS (siehe 4.3.1). Nach erfolgreicher Authentifizierung erhält der Benutzer direkten Zugang zum Netz der eigenen Einrichtung. Das bedeutet für die eigene Einrichtung ist der Benutzer ein „interner“ Benutzer.

Die Vorteile sind seine Einfachheit und die nicht zu unterschätzende Möglichkeit die „Home-Policy“ im fremden Netz anwenden zu können. Layer-2-Forwarding ermöglicht den VPN-Aufbau von multiplen Verbindungen (Multitunnels). Allerdings hat dieser Ansatz auch seine Nachteile. Durch den Multitunnel ergibt sich eine schlechte Skalierbarkeit und eine schlechte Wartbarkeit. Das Nutzerverzeichnis muss an jeder Einrichtung seperat aktualisiert werden.

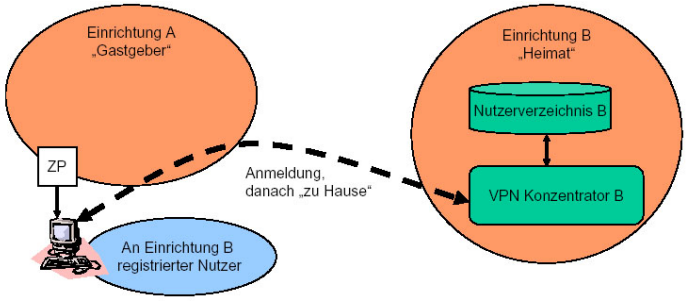


Abbildung 9: Roaming im BelWü [t07e]

8 Roaming im DFN

Ziel des DFNRoaming [t07d] ist es, eine Nutzung der Netzzugangs-Infrastruktur des Gastgebers ohne den erheblichen Verwaltungsaufwand für Nutzer und Netzverwalter. Dies soll einen Zugang zum Wissenschaftsnetz nicht nur in der eigenen sondern auch in anderen wissenschaftlichen Einrichtungen ermöglichen. Die Kosten für die Nutzung des DFN-Roaming sind bereits in den Kosten für den DFNInternet Dienst enthalten. Zur Nutzung des Dienstes ist eine einmalige Registrierung notwendig, um eine Kennung zu erhalten die bei allen anderen Einrichtungen gültig ist.

8.1 Leistung des DFN

Das DFN betreibt und pflegt das DFN-Toplevel-Verzeichnis und zertifiziert und registriert Nutzerverzeichnisse und Webserver. Außerdem bietet es eine Migrationslösung zu 802.1x an, da oft die bereits vorhandenen Accesspoints den IEEE 802.1x Standard nicht unterstützen. Diese Accesspoint können parallel zum Betrieb durch 802.1x-fähige Accesspoints ausgetauscht werden. Die vorhandene Radiusinfrastruktur kann weiter genutzt werden und VPN-basierte Lösungen können parallel verwendet werden. Die Koordination des Dienstes in internationale Infrastrukturen übernimmt ebenfalls das DFN. Die Aufgabe der jeweiligen wissenschaftlichen Einrichtung ist es, ein dienstkonformes WLAN zu betreiben. Diese registriert seine Nutzer und betreibt ein Nutzerverzeichnis, das beim DFN-Toplevel Verzeichnis angemeldet werden muss.

8.2 Betriebskonzept des DFN

Über ein dienstkonformes WLAN authentifiziert sich der registrierter Nutzer mit seiner Kennung und bekommt Zugang zum GWin (nationaler Teil des Deutschen Forschungsnetzes). Um die Kennung des Nutzers zu überprüfen, wird im Nutzerverzeichnis nach der Zugangsberechtigung des Nutzers gesucht. Um dieses Nutzerverzeichnis zu finden, wird das vom DFN betriebene DFN-Toplevel Verzeichnis verwendet.

Der Pilotbetrieb des DFNRoaming wurde am 30.06.2004 erfolgreich abgeschlossen. Allerdings haben sich bisher wenige wissenschaftliche Einrichtungen dem Programm angeschlossen. Bisher angeschlossene Standorte:

- ZIB Berlin
- DFN-Geschäftsstelle Berlin
- Hochschule Mittweida

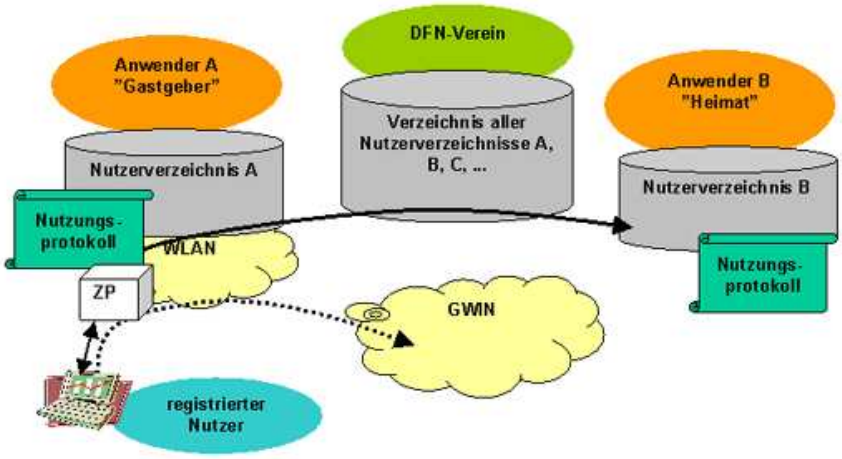


Abbildung 10: Roaming im G-WIN des DFN [t07e]

- Hochschule für Graphik und Buchkunst Leipzig
- Universität Leipzig
- Technische Universität Chemnitz
- Universität des Saarlandes

9 EduRoam (Education Roaming)

Einen ähnlichen Ansatz wie das DFN, allerdings auf europäischer Ebene, verfolgt das **EduRoam** [t07a]. Es bietet teilnehmenden wissenschaftlichen Einrichtungen, den Zugang zu allen anderen teilnehmenden Einrichtungen. Bisher sind die Niederlande, England, Griechenland, Tschechien, Spanien, Portugal, Kroatien, Slowenien, Dänemark, Polen, Litauen, Finnland und Norwegen an das EduRoam angeschlossen. Die Authentifikation erfolgt an der heimischen Einrichtung und die Genehmigung an der besuchten Einrichtung. Im Rahmen des Géant2 Projekts soll eine Infrastruktur für die Authentifizierung und Genehmigung für ganz Europa aufgebaut werden.

10 Zusammenfassung

Roaming auf Layer 2 Ebene funktioniert gut allerdings reicht das nicht, wenn bei großen Funknetzen mehrere IP-Subnetze benötigt werden. In diesem Fall muss entweder Mobile IP (siehe 4.1) oder IPv6 (siehe 4.2) für Roaming auf Layer 3 eingeführt werden.

Die Vorteile des DFN-Modells sind die gute Skalierbarkeit und die gute Wartbarkeit. Allerdings gibt es auch Nachteile. Da die Authentifizierung über das Nutzerverzeichnis des Heimnetzes des Nutzers erfolgt, wird dabei auch die Policy des Fremdnetzes angewandt. Das kann zu Problemen führen, da die Policy des Heimnetzes unterschiedlich zu der Policy des jetzt verwendeten Netzes sein kann. Dieser Nachteil kann durch Aufbau eines Tunnels ins Heimatnetz kompensiert werden.

Der BelWü-Ansatz hat seine Vorteile in seiner Einfachheit und der Möglichkeit die „Home-Policy“ im fremden Netz anwenden zu können. Die Nachteile sind die schlechte Skalierbarkeit und die schlechte Wartbarkeit, da das Nutzerverzeichnis jeder Einrichtung nicht zentral aktualisiert werden kann.

In Zukunft werden wohl DFNRoaming und BelWü-Roaming in das EduRoaming überführt.

Roaming zwischen GPRS/UMTS und WLAN/WiMAX lässt sich sehr gut realisieren und wird wohl in Zukunft in Ballungsgebieten eingesetzt werden, um günstig große Kapazitäten anbieten zu können.

Literatur

[Perk98] Charles E. Perkins. *MOBILE IP: desgin principles and practices.* Addison-Wesley. 1998.

[P.M.03] S.Martin-Leon P.M.Feder, N.Y.Lee. A Seamless Mobile VPN Data Solution for UMTS and WLAN Users. 2003.

[t07a] Education Roaming. http://www.eduroam.org/.

[t07b] Elektronik Kompendium. http://www.elektronik-kompendium.de/sites/kom/0910191.htm.

[t07c] IEEE. http://grouper.ieee.org/groups/802/1/pages/802.1X-rev.html.

[t07d] Roaming im DFN. http://www.dfn.de/content/dienstleistungen/dfnroaming/.

[t07e] RZ Karlsruhe. http://www.rz.uni-karlsruhe.de/download/.

[t07f] Wikipedia WLAN. http://de.wikipedia.org/wiki/Wlan.

[vSeb02] T.C. van Sebille. WLAN-GPRS Roaming based on Mobile IP (v4). Diplomarbeit, Eindhoven University of Technology, 2002.

[Würz] Universität Würzburg. VPN-Server. http://www.rz.uni-wuerzburg.de/dienste/kommunikation/.

Abbildungsverzeichnis

Intrusion Detection und Prevention Systeme

Nikolay Orozov

Kurzfassung

Im vorliegenden Dokument werden grundlegende Informationen zu Intrusion Detection und Prevention Systemen bereitgestellt. Zunächst werden im Kapitel 2 Architekturen und Funktionsweise von Intrusion Detection Systemen erläutert. Dabei wird auf die Methoden der Angriffserkennung näher eingegangen. Als Beispiel von einem Network Intrusion Detection System wird Snort genommen. Kapitel 3 beschäftigt sich mit dem Begriff Intrusion Prevention.

1 Motivation

Die Zahl der Angriffe in den letzten 10 Jahren steigt exponentiell, laut Statistiken von CERT[1](Abbildung 1). Dabei handelt es sich um Angriffe, die bei CERT registriert worden sind. In der Wirklichkeit ist die Anzahl viel höher. Deshalb haben sich Intrusion Detection Systeme (IDS) zu einem festen Bestandteil von Sicherheitsstrukturen entwickelt. Inzwischen geht man noch einen Schritt weiter zu Intrusion Prevention: es werden Methoden entwickelt, die Angriffe nicht nur entdecken, sondern auch verhindern sollen.

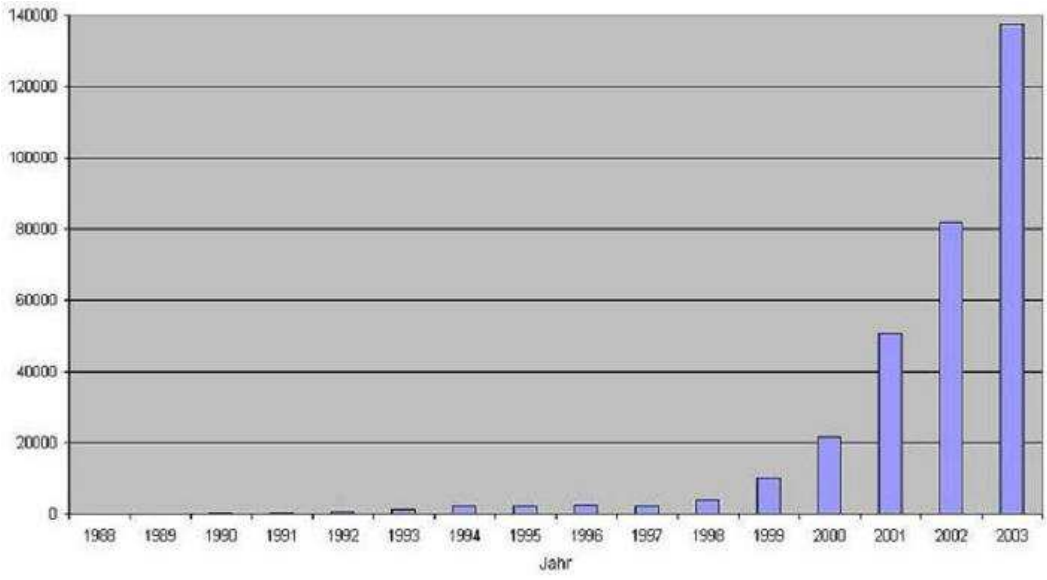


Abbildung 1: Registrierte Sicherheitsvorfälle nach CERT [CERT04].

[1]CERT: Computer Emergency Response Team der Carnegie Mellon University

2 Was ist ein Intrusion Detection System?

Der Begriff Intrusion ist von Heberlein, Levitt und Mukherjee von der University of California, Davis [HeML91] wie folgt definiert: “Eine Menge von Handlungen, deren Ziel es ist, die Integrität, die Verfügbarkeit oder die Vertraulichkeit eines Betriebsmittels zu kompromittieren“. Im Allgemeinen ist eine Intrusion als Verletzung der Sicherheit eines Computersystems oder Computernetzes zu vestehen. “Als Intursion Detection wird die aktive Überwachung von Computersystemen und/oder -netzen mit dem Ziel Erkennung von Angriffen und Missbrauch bezeichnet“ [fSid02]. Aus allen stattfindenden Ereignissen sollen diejenigen herausgefunden werden, die auf Angriffe oder Sicherheitsverletzungen hindeuten, damit sie später vertieft untersucht werden können. Diese Ereignisse sollen dabei in Echtzeit erkannt und gemeldet werden. Der Intrusion-Detection-Prozess wird durch verschiedene Werkzeuge unterstützt, z. B. Werkzeuge zum Vergleich von Checksummen und Zeichenketten, für Analyse von Log-Dateien, zur Auswertung und Alarmierung sowie zur Archivierung der Ergebnisse. Eine Zusammenstellung von geeigneten Werkzeugen wird als Intrusion Detection System (IDS) bezeichnet.

Der Grundprinzip eines IDS (Abbildung 2) ist mit der Funktionsweise einer Alarmanlage vergleichbar. Nach einem Einbruch wird sofort Alarm ausgelöst und werden Gegenmassnahmen getroffen, wie z.B. Anruf bei den zuständigen Sicherheitsbehörden. Ein IDS benachrichtigt sofort nach einem erkannten Angriff den Sicherheitsadministrator (über e-mail, Pager, etc.), protokolliert den Angriff und entscheidet je nach Konfiguration, was als Nächstes zu tun ist. Im Extremfall wird versucht, die Verbindung zu trennen oder das ganze System wird heruntergefahren.

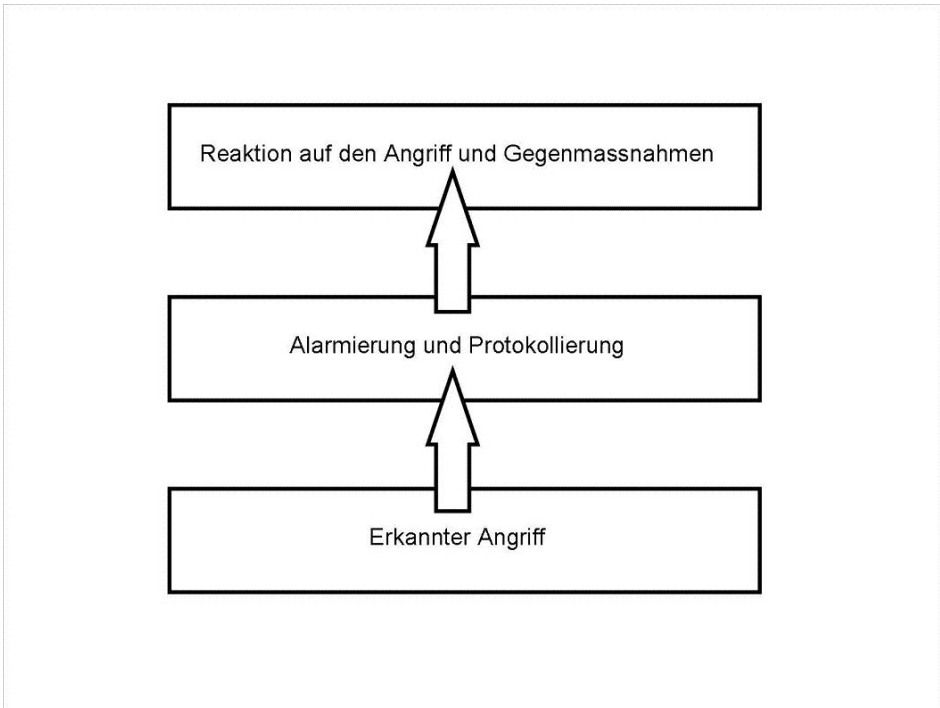


Abbildung 2: Grundprinzip eines IDS

2.1 Komponenten eines Intrusion Detection Systems

Ein IDS besteht normalerweise aus folgenden Hauptkomponenten:

- Sensoren[2] (Host- und Netzsensoren) zur Überwachung des Betriebssystems, von Applikationen und des Netzverkehrs.
- Datenbank zur Speicherung und Verwaltung der Ereignisdaten.
- Komponeneten zur Anzeige, Konfiguration, Erstellung und Anpassung von Regeln, Sortierung und Klassifikation der Ergebnisse, Generierung von Reports etc.

2.2 Architekturen und Funktionsweise von Intrusion Detection Systeme

In der Regel werden IDS-Systeme danach unterschieden, was sie überwachen: ein gesamtes Netzwerk, einen Host/Server oder nur eine einzelne Anwendung. Um die Effektivität zu steigern, wird meistens eine Kombination aus Netzwerk- und Host- basierter Intrusion Detection verwendet (Abbildung 3).

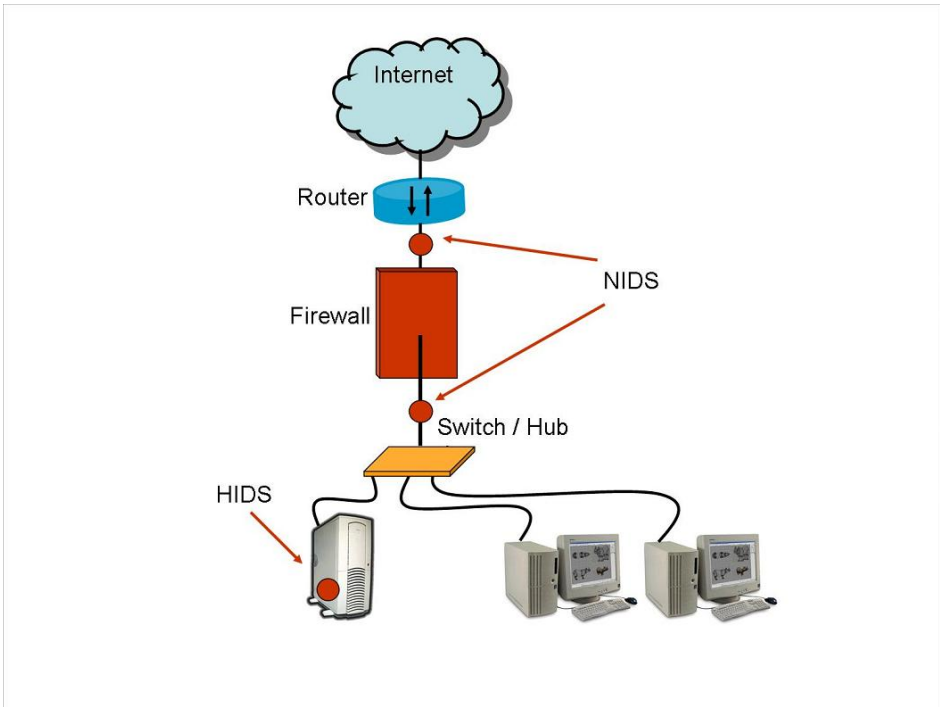


Abbildung 3: Mögliche Kombination von NIDS (2.2.1) und HIDS (2.2.2)

[2]Sensor: Ein System, das die drei Komponenten Datensammlung, Datenanalyse und Darstellung der Ergebnisse kombiniert. Ein Sensor (zu lateinisch sensus - "Gefühl") oder (Mess)Fühler ist in der Technik ein Bauteil, das neben bestimmten physikalischen oder chemischen Eigenschaften (z.B.: Temperatur, Feuchtigkeit, Druck, Helligkeit, Beschleunigung) auch die stoffliche Beschaffenheit seiner Umgebung qualitativ oder als Messgröße quantitativ erfassen kann. Die Abgrenzung der Begriffe Sensor und Messgerät ist fließend. Hier wird der Begriff Sensor im Sinne von Modul zur Erfassung und Analyse von Daten verwendet.

2.2.1 Netzwerk Intrusion Detection Systeme (NIDS)

Die NIDS belauschen (sniffen) den Verkehr in Echtzeit an mehreren wichtigen Punkten im Netzwerk. Im Normalfall werden die Sensoren ausserhalb der Firewall platziert. Auf diese Weise werden aber nicht alle Attacken entdeckt. Damit Angriffe sowohl von innen als auch von aussen erkannt werden können, müssen Sensoren auch hinter der Firewall platziert werden (Abbildung 3). So "kann man entdecken, ob die Firewall falsch konfiguriert ist (wenn Angriffe durchgehen, die eigentlich aufgehalten werden sollten)" [NoNo04].

In einem geswitchten Netzwerk muss der Datenverkehr auf einen Mirror-Port des Switches gespiegelt und an den IDS-Sensor geleitet werden. Das ist erforderlich, damit der gesamte Verkehr, der durch den Switch geht, gefiltert werden kann. Das Netzwerk-Interface des IDS muss auch alle Pakete akzeptieren. Bildlich gesprochen ist das NIDS wie ein Sicherheitssystem von Überwachungskamera einer Bank. Es werden nicht nur Personen und Kunden, die von aussen hereinkommen, sondern auch die Mitarbeiter überwacht.

2.2.2 Host Intrusion Detection Systeme (HIDS)

Das sind Systeme, die auf Software basieren und die auf jeden Host, den überwacht werden soll, installiert werden müssen. Der Software-Agent beeinflusst jedoch die Systemperformance, ist aber abhängig von dem Host-Betriebssystem, da er seine Informationen aus Log- und Kernel-Dateien erhält. Es wird der Zugriff auf wichtige Systemdateien sowohl Änderungen an Benutzerberechtigungen verfolgt.

2.2.3 Application-Layer IDS

Die Application-Layer Intrusion Detection Systeme sind ein Spezialfall von Host-IDS. Sie werden eingesetzt, um Ereignisse auf Applikationsebene zu überwachen. Die Daten einer bestimmten Anwendung bzw. eines Netzwerkprotokolls (TCP/IP) werden herausgefiltert. Dann wird versucht, die Anfrage zu verstehen und zu unterdrücken, falls es sich um Aufrufe mit schädlichem Inhalt handelt.

2.3 Methoden der Angriffserkennung

Um Angriffsversuche erkennen zu können, müssen gesammelte Daten (Auditdaten) mit geeigneten Methoden analysiert werden. Hierfür haben sich in den letzten Jahren zwei verschiedene Methoden entwickelt: die Missbrauchserkennung und die Anomalieerkennung. Die Missbrauchserkennung versucht, bekannte Angriffe anhand vordefinierter Muster zu identifizieren. Die Anomalieerkennung dagegen hängt vom Benutzerverhalten ab und versucht Abweichungen von dem normalen Verhalten festzustellen.

Um zu bestimmen, was ein "normales Verhalten" bedeutet, muss das IDS einige Informationen sammeln, wie z.B. welche Systemparameter "beobachtet" werden, für welchen Zeitraum und welche Werte von diesen Parametern als normal bezeichent werden. Dafür ist eine Einarbeitungsphase oder Lernphase erforderlich. Einige Systemparameter, die berücksichtigt werden:

- Seitenwechselrate
- CPU-Auslasung
- Speicher-Auslastung

- Anzahl der Telnet-Anforderungen während eines Zeitraums
- Anzahl der aktiven Ports

In den folgenden Abschnitten werden verschiedene Verfahren zur Anomalieerkennung kurz erläutert.

- Anomalieerkennung durch Protokollanalyse
  Eine Weiterentwicklung der signatur-basierte Erkennungstechnologie stellt die "Protokollanalyse" dar. Hierbei werden die Daten aufgrund der vorliegenden Protokollinformationen in den Datenpaketen analysiert. Es ist leicht durch die Protokollspezifikation zu definieren, was als "normal" betrachtet werden muss. Gegenüber der Signaturerkennung ist diese Methode effizienter, da keine Signatur-Datenbank durchsucht und getestet werden muss. Nachteilig ist, dass die fehlerhaften Protokollspezifikationen sowohl zu Fehlalarmen als auch zum Übersehen von Angriffen führen können.

- Anomalieerkennung auf Basis statistischer Daten
  Hier wird das normale Verhalten durch statistische Kenngrössen festgelegt. Es werden eine Menge von statistischen Profilen verwaltet, die das Normalverhalten der Benutzer und der Systemkomponenten beinhalten. Diese Profile werden in periodischen Abständen aktualisiert. Anhand dieser Werte ermittelt das IDS, ob das aktuelle Verhalten vom Normalverhalten abweicht. Mit statistischen Verfahren können Angreifer erkannt werden, die als andere Benutzer agieren, indem sie deren Account benutzen. Da sich der Angreifer anders verhält als der "normale" Benutzer, wird das IDS darauf aufmerksam. Auf diese Weise können auch völlig neue Angriffstechniken erkannt werden. Der große Nachteil vieler statistischer Verfahren ist die Unfähigkeit im Echtzeit-Betrieb zu arbeiten, da eine sehr hohe Performanz dafür nötig ist. Ausserdem ist eine automatische Anomalieerkennung auf Basis statistischer Daten sehr komliziert und fehlerbehaftet.

- Anomalieerkennung auf Basis von Künstlicher Intelligenz
  Die Methoden der KI werden vor allem in der Lernphase eingesetzt, um festzulegen, was ein normales Verhalten ist. Eine große Bedeutung in der Künstlichen Intelligenz haben die Neuronalen Netze. Sie sind lernfähige Systeme, die aus mehreren Knoten (Neuronen) bestehen. Ihre Funktionsweise entspricht dem Verhalten der Nervenzellen im Gehirn: über die zwischen den Neuronen bestehenden Verbindungen werden Informationen in Form von Signale ausgetauscht. Den Verbindungen zwischen den Knoten in einem Neuronalen Netz werden Gewichte zugeordnet, die durch Algorithmen (z.B. Backpropagation Algorithmus) verändert werden können. Dieser Vorgang wird "Trainieren" oder "Lernen" genannt. Das "Wissen" des Neuronalen Netzes ist genau in diesen Gewichten kodiert.

- Anomalieerkennung auf Basis von Honeypots
  Honeypots sind dedizierte IT-Systeme (Server, Netze, Programme, Prozesse), die keine produktive Funktion erfüllen, sondern ausschließlich "Fallen" für Angreifer darstellen, in dem sie produktive oder auch besonders sicherheitskritische Systeme vortäuschen [fSid02]. Honeypots eignen sich sehr gut für eine Anomalieanalyse, da im Wesentlichen sämtliche Zugriffe und Aktivitäten als normal einzustufen und damit beobachtenswert sind. Die Honeypots werden meistens als Web-, Mail- oder DNS-Server eingerichtet, da sie am häufigsten angegriffen werden.

2.4 Vergleich der Analysemethoden

Mittels Signaturanalyse können die meist bekannten Angriffe erkannt werden, vorausgesetzt zu jedem Angriff ein Angriffsmuster in der Signaturdatenbank zu finden ist. Sonst wird ein

entsprechender Angriff nicht erkannt. Eine Signatur kann aber auch leicht modifiziert werden. Das führt zu einem hohen Wartungsaufwand, weil die Signaturdatenbank ständig aktualliesiert werden muss. Durch die Anomalieerkennung dagegen ist es möglich, auch neue völlig unbekannte Attacken zu entdecken. Ein Angreifer, der unter falschem Benutzeraccount aggiert, kann auch erkannt werden, ohne dabei Mißbrauch zu verursachen. Andererseits ist es sehr schwierig der Ansatzt des "normalen Verhaltens" in der Praxis umzusetzten. Das Verhalten eines Benutzers hängt mit Sicherheit von seinen aktuellen Aufgaben. Jede Abweichung von den am häufigsten durchgefürten Aufgaben würde dann zu Fehlalarmen führen. Mit den verschiedenen Methoden der Anomalieerkennung wird natürlich versucht, ein normales Verhalten relativ exakt zu definieren. Ein Angreifer kann aber auch das ausnutzen, in dem er schon vor der Lernphase einbricht. Falls in dieser Phase ein Eindringling im System oder im Netz ist, wird später dieses Verhalten als normal betrachtet und nicht als Angriff erkannt. Eine Attacke kann dann problemlos durchgeführt werden, ohne dabei einen Verdacht zu erregen.

Ein Unterschied zwischen den beiden Methoden besteht auch bei der Erkennungsaufwand: "gibt es zu einem Angriff eine Signatur, so wächst der zu betreibende Aufwand zur Erkennung des Angriffs exponentiell in Abhängigkeit vom Effektivitätsgrad des Angriffs. Bei der Anomalieerkennung verhält es sich umgekehrt: der Aufwand bei der Erkennung wächst logarithmisch mit der Effektivität des Angriffs" (Abbildung 4) [fSid04]. Daraus folgt, dass die Anomalieerkennung effektivere Angriffe mit einem geringeren Aufwand als die Missbrauchserkennung erkennen kann.

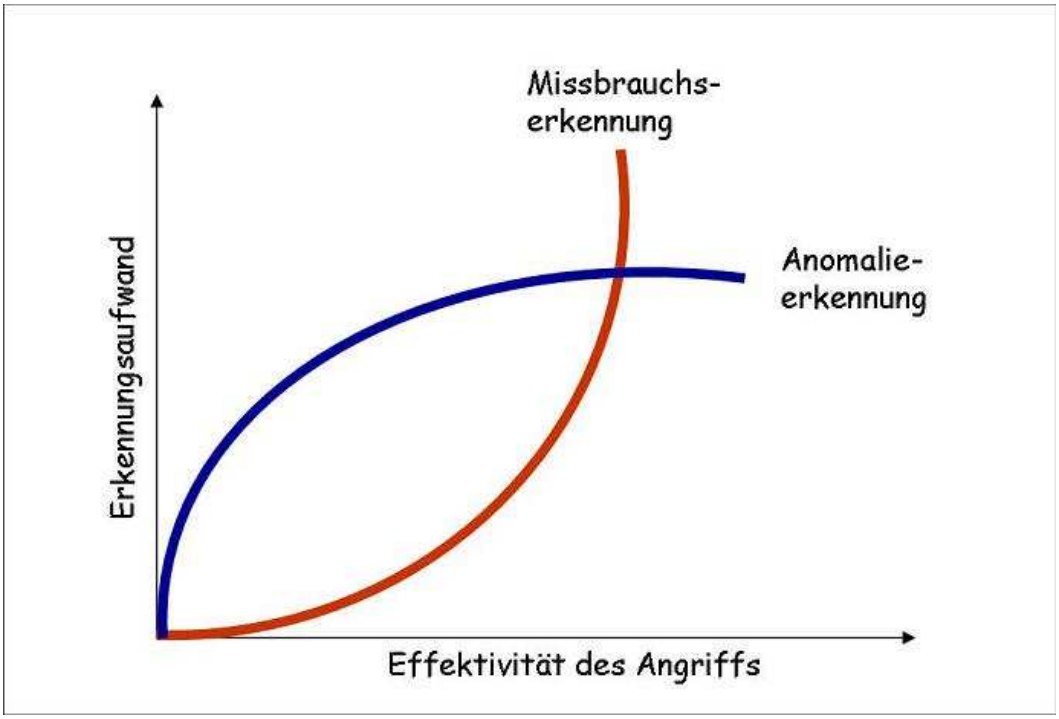


Abbildung 4: Aufwandsunterschiede bei der Angriffserkennung [fSid04].

Zusammenfassend kann man sagen, dass die beiden Methoden ihre Vor- und Nachteile haben. Durch eine geeignete Kombination der Verfahren können die Schwächen gemindert und gleichzeitig die Effektivität gesteigert werden.

2.5 Reaktion eines IDS im Angriffsfall

Wird ein Angriff festgestellt, bestehen verschiedene Reaktionsmöglichkeiten:

- Der Angriff wird nur protokolliert und zur wietere Bearbeitung in der Datenbank gespeichert. Es erfolgen aber keine Gegenmassnahmen.
- Der Administrator wird sofort benachrichtigt (per e-mail, sms). Das Problem dabei ist, dass mehrere Angriffe leicht dazu führen können, dass der Administrator mit E-Mails überschwemmt wird. Es werden zu viele Informationen geliefert, die nicht verarbeitet werden können.
- Es besteht die Möglichkeit, automatisch auf Angriffe zu reagieren. Dies kann wie oben beschrieben eine einfache Benachrichtigung sein, aber auch aktivere Gegenmaßnahmen, um einen Angriff zu blockieren. Für diese Blockade kommen vor allem folgende Techniken zum Einsatz:
  - Beenden der Verbindung durch Reset-Pakete.
  - Dauerhafte Sperrung der IP-Adresse des Angreifers .
  - Isolation eines Netzes vom Internet: wenn mehrere Angriffe über einem bestimmten Zeitraum auftretten, kann das IDS mit einem Befehl dem Router den Strom abstellen.
  - SYN/ACK: das IDS kennt alle Ports, die von der Firewall gesperrt sind. Jeds Mal, wenn ein *TCP-SYN-Paket* an einen gesperrten Port gesendet wird, antwortet das IDS mit einem *SYN-ACK*. Der Angreifer führt den Drei-Wege-Handschlag durch und denkt, dass er eine Menge möglicher Ziele gefunden hat. Auf diese Weise wird er in die Falle geführt.

  Das direkte Eingreifen zur Unterbrechung der Kommunikation zwischen einem Angreifer und dem Opfer wird oft als *Session Sniping* oder *Knockdown* bezeichnet und erfolgt durch das Einfügen von Reset-Paketen, um die Verbindung zu unterbrechen [Snel03]. In manchen Fällen kann es besser sein, die Kommunikation zu dem angegriffenen System zu unterbrechen, anstatt eine Beschädigung zu riskieren. Ein Angreifer kann aber eine zwiete Verbindung initialisieren und seinen TCP-Stack so ändern, dass er Reset-Pakete einfach ignoriert. Diese Technik wird dann unbrauchbar. Das Isolieren eines gesamten Netzes ist auch keine dauerhafte Lösung, denn das kann die Arbeit einer ganzen Firma blockieren und das IDS kann zum Ziel von Denial-of-Service-Attacken werden. Das Aussperren ist eine der wichtigsten automatischen Reaktionen, die zur Zeit zur Verfühgung steht. Man braucht aber eine Liste mit IP-Adressen (von Partnern und Kunden), die niemals gesperrt werden sollen. Das führt zu dem Problem mit *IP-Spoofing*[3], was wieder von einem Angreifer ausgenutzt werden kann, um die Kommunikation mit den Geschäftspartnern zu unterbrechen.

  Andere Interventionsmethoden umfassen die Neukonfigurierung der Firewall, aktiv Informationen über den Host oder die Website des Angreifers herauszufinden oder sogar eine Gegenattacke zu starten. Bevor man solche Funktionen aktiviert, sollte man aber auf jeden Fall juristischen Rat einholen.

2.6 Einige Nachteile von IDS

- Das IDS kann den Angriff erst erkennen, wenn das Angriffsmuster schon auf dem Weg zum Opfer ist. Ein Reset-Paket, das vom IDS verschickt wird, kommt daher erst dann beim Angreifer an, wenn der Angriff bereits beendet ist. Deswegen sinkt die Effektivität einer Reaktion über das Versenden von Reset-Paketen.

[3]IP-Spoofing bezeichnet in Computernetzen das Versenden von IP-Paketen mit gefälschter Quell-IP-Adresse.

- Das Einfügen von Regeln und die Neukonfiguration einer Firewall benötigt auch Zeit. Inzwischen kann ein automatischer Angriff bereits abgeschlossen sein.

- Ein weiteres Problem ist die Anfälligkeit für Denial-of-Service-Angriffe. Dabei kann das IDS selbst zum Opfer eines Angriffs werden.

- Die Blockier-Reaktion eines IDS auf einen Fehlalarm kann schwerwiegende Folgen haben.

2.7 NIDS Beispiel: Snort

Snort ist eine Idee von Martin Roesch und wird von ihm als Open Source NIDS unter der GPL weiterentwickelt. Snort ist fähig, den Traffic an bestimmten Ports im Netzwerk in Echtzeit zu überwachen und zu analysieren. Snort läuft auf mehr als 20 verschiedenen Platformen, darunter Linux, FreeBSD, OpenBSD, NetBSD, HP-UX, Solaris, MacOS X und Windows. Snort verwendet zur Paketüberwachung die Berkeley-Paketfilterbibliothek "libpcap" für Linux und Unix-Systeme und "winpcap" für Windows-Systeme [Möhl04]. Den Netzwerkverkehr wird anhand von Signaturen gefiltert. D.h. Snort arbeitet bei der Analyse der gesammelten Daten nach dem Prinzip der Missbrauchserkennung. Die Konfiguration von Snort ist einfach und flexibel. Der Bentzer hat die Möglichkeit, eigene Signaturen zu erstellen, sowie das Einfügen neuer Funktionen durch Plugins. Die zahlreichen Optionen von Snort erlauben Zugriff auf alle wichtigen Bestandteile der zu untersuchenden Pakete. Den Datenteil eines Paketes kann Snort anhand von Binärmustern oder mit Pattern-Matching analysieren.

Im Normalfall wird Snort über die Kommandozeile gestartet, es wurden jedoch inzwischen einige grafische Werkzeuge entwickelt, wie z.B. ACID[4], Snort IDScenter[5], das eine GUI für Windows zur Verfügung stellt. (Abbildung 5).

Snort kann auch als reiner Packet-Sniffer oder als Packet-Logger agieren. Die vier aufeinander aufbauenden Hauptkomponenten sind: Sniffer, Präprozessor, Detection Engine und einem für die Ausgabe zuständigen Plugin. Die Präprozessoren nehmen die Paketdaten auf und verarbeiten sie noch bevor sie analysiert werden. Sie können aussrdem Portscans entdecken und IP-Defragmentierung durchführen.

Die gesammelten Daten werden auf ein bestimmtes Verhalten untersucht. Snort bietet über seine Plugin-Schnittstelle das Plugin SPADE[6], welches von Silicon Defense entwickelt wird. Dadurch ist eine statistische Anomalieerkennung möglich.

Snort bietet noch Möglichkeiten, Paket-Informationen zu sichern. Dies kann mit Hilfe von XML geschehen. Am CERT wurde die sogenannte SNML[7] entwickelt, mit deren Hilfe Snort die Daten formatiert in eine Datei oder eine Datenbank schreiben kann.

Snort sammelt seine Daten in einem Verzeichnis und Unterverzeichnissen, die nach der IP-Adresse benannt werden, von der der Angriff ausging. Daten über Portscans werden in einer gesonderten Datei gesichert.

Über seine Regeln kann Snort Reaktionen auf Angriffe auslösen. Die Möglichkeiten erlauben eine explizite Beendigung der TCP-Verbindung über ein RST-Paket oder über ICMP.

[4]ACID: Analysis Console for Intrusion Databases, http://www.cert.org/kb/acid/
[5]Snort IDScenter: http://www.engagesecurity.com/
[6]SPADE: Statistical Packet Anomaly Detection Engine
[7]SNML: Simple Network Markup Language

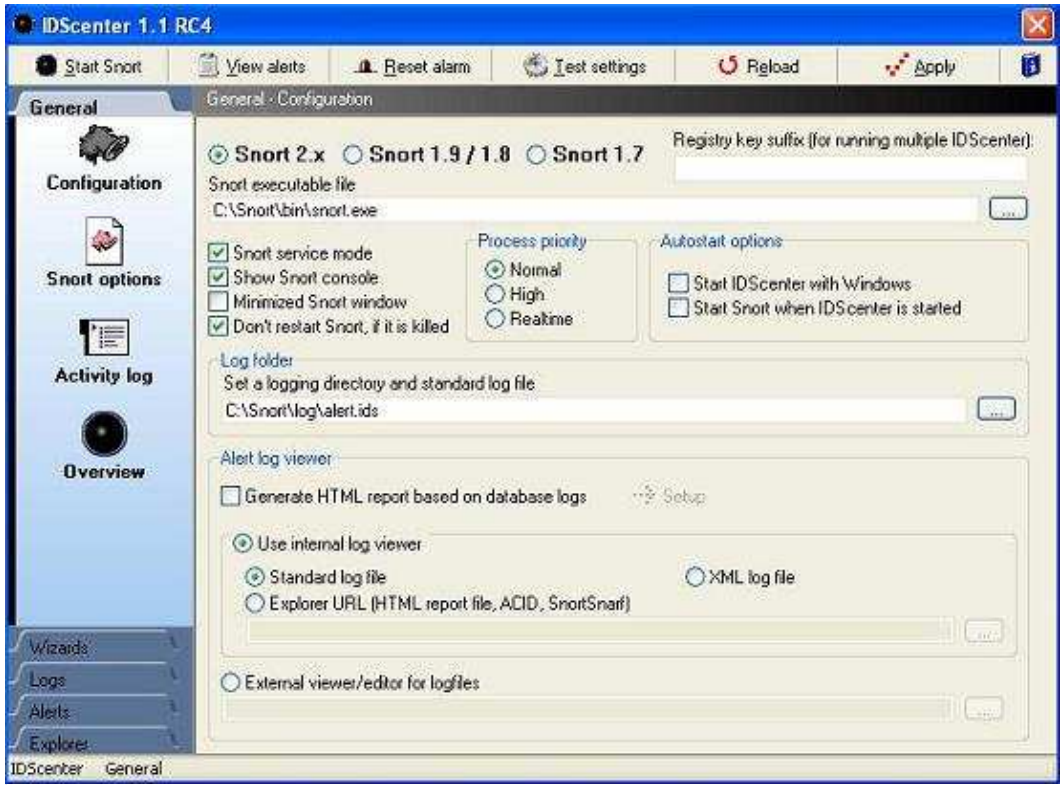


Abbildung 5: Snort IDScenter Screenshot

3 Intrusion Prevention Systeme

3.1 Wo liegt der Unterschied zwischen Intrusion Detection und Intrusion Prevention?

Intrusion Detection Systeme bieten keinen Schutz. Sie sind fähig, einen Einbruch zu erkennen und zu registrieren, aber die Reaktionen finden immer erst nach dem durchgeführten Angriff statt, die feindlichen Pakete haben unter Umständen bereits ihr Ziel erreicht. Wenn man ein IDS mit einem Alarmsystem vergleicht, bedeutet das, dass ein Einbruch in einem Haus vom Alarmsystem erkannt und gemeldet wird, aber die verursachten Schäden sind nicht mehr rückgängig zu machen. Diesen Mangel versuchen Intrusion Prevention Systeme zu beheben. Als Intrusion Prevention System kann man ein System bezeichnen, das Anfriffe nicht nur erkennen, sonder auch aktiv verhindern kann. In diesem Sinn stellt das IPS eine Erweiterung des IDS dar: ein IPS muss als erstes ein hoch akurates IDS sein. Also, ein IDS erfüllt die Aufgabe, einen Angriff zu entdecken, möglichst viel Informationen über den Angriff zu sammeln und Alarm auszulösen. Die Haputanfordernung eines IPS ist Attacken und Endringlinge zweifelslos zu identifizieren und rechtzeitig geeignete Gegenmassnahemen zu treffen, so dass mögliche Schäden verhindert werden.

3.2 Architekturen von Intrusion Prevention Systeme

Sowohl Intrusion Detection Systeme als auch Intrusion Prevention Systeme können netzwerk-, host- oder applikationsbasiert sein:

- Netzwerk Intrusion Prevention System (NIPS)

Ein NIPS bietet Schütz vor Angriffen auf Netzwerkebene. Es ist in der Lage vollautomatisch Angriffe zu erkennen und danach Gegenmassnahmen zu starten, um die Folgen eines Angriffs zu verhindern.

- Host Intrusion Prevention System (HIPS)

  HIPS sind Software-Agenten, die auf jedem zu überwachenden System installiert werden. Sie bilden eine Hülle um den Systemkern und kontrollieren den Zugriff auf System-Ressourcen.

- Application-Layer IPS

  Fast täglich werden neue Schwachstellen und Einbruchsmöglichkeiten in Applikationen oder Betriebssystemen entdeckt. Besonders gefährdet sind die Web-Applikationen. Die Systeme, die sich auf Web-Applikations-IPS spezialisiert haben, arbeiten jedoch auf einer anderen Ebene. Sie lernen teilweise automatisch die benötigten URLs jeder Applikation, die erlaubten Wertebereiche, Zeichen und Längen von Eingabewerten und bauen daraus eine Policy auf, gegen die jede http-Anfrage geprüft wird. Zusätzlich überwachen sie den Status der Benutzersessions. Einer der bekanntesten Web-Angriffe ist der Buffer-Overflow.

  Was passiert eigentlich bei einem Buffer-Overflow?

  Bei modernen Betriebssystemen hat jede Anwendung einen eigenen, virtuellen Adressraum. Beim Start der Anwendung wird der Adressraum in drei Teilen aufgeteilt: Code-Speicher, Daten-Speicher (Heap) und Stack. Der Stack ist ein Zwischenspeicher für lokale Variablen, Rücksprungadressen und Übergabeparameter. Er funktioniert nach dem LIFO-Prinzip (Last-In-First-Out). Wenn also mehr Daten in den Puffer kopiert werden, als er behalten kann, kommt es zu einem Buffer-Overflow (Speicherüberlauf) und dadurch wird die Rücksprungadresse überschrieben. An dieser Stelle wird das Programm abstürzen.

  Wie kann ein Angreifer einen Buffer-Overflow ausnutzen?

  Ein Angreifer kann aber genau diese Schwäche besser ausnutzen, in dem er die Anwendung mit sinnvollem Code (Befehle) überflutet. Die Daten, die dabei außerhalb der Speichergrenze geraten, werden von der CPU als Programmcode interpretiert und ausgeführt. Das heisst, ein Angreifer kann in nur wenigen Schritten die Kontrolle über den Rechner übernehmen: In einem ersten Schritt wird der Buffer-Overflow ausgelöst, dann wird die Rücksprungadresse manipuliert. Das Programm wird dann an einer anderen Stellen weitermachen, als sonst geplant. Danach wird der schädliche Code ausgeführt.

  Den Versuch, einen Buffer-Overflow auf einer Web-Anwendung zu erzeugen, kann vom Application-IPS verhindert werden. Das IPS überprüft alle Systemaufrufe und stellt dabei fest, ob der Code von einer Anwendung oder aus einem Speicherüberlauf stammt. Ist dies der Fall, kann das IPS Pakete aus dem Stack entfernen noch bevor die Befehle vom Betriebssytem ausgeführt werden. Ist der Code Teil einer Anwendung, wird er ganz normal ausgeführt [McAf04b].

3.3 Ein anderes Verfahren zur Angriffserkennung - Intentionserkennung

Intrusion Prevention Systeme, die nicht auf der Erkennungstechnologie klassischer IDS basieren, haben ganz andere Eigenschaften. Dazu gehören Systeme, die mit dem Angreifer interagieren, um seine Intention herauszufinden. Man versucht erst gar nicht tausende Siganturen von potentiellen Angriffen zu erkennen. Die Idee der “Identifikation von Angreifern durch Nachwies ihrer Intention geht davon aus, dass alle tatsächlich durchgeführten Angriffe ein gemeinsames Vorgehen haben“ [Ciro04]. Dieses Vorgehen ist in mehreren Schritten gegliedert:

der erste Schritt ist das Sammeln von Informtionen. In einem zweiten Schritt wird der Angriff durchgeführt und dann folgen weitere Schritte zum Verwischen der Spuren. Zunächst wird ein Angreifer also nach Schwachstellen suchen und erst wenn er weiß, wo er angreifen kann, wird er den ersten Angriffsversuch starten. Im Gegensatz zu über 4000 verschiedenen Mustern, die im Normalfall ein klassisches IDS kennt und die täglich aktuallisiert werden müssen, ist es in einem ersten Schritt ausreichend, 10 bis 20 verschiedene Methoden der Informationssammlung zu erkennen. Das Gewinnen von Informationen reicht aber nicht aus, um auf einen tatsächlichen Angriff zu schließen. Durch vorgetäuschten Schwachstellen wird der Angreifer in die Falle geführt. Sobald er versucht, eine vorgetäuschte Schwachstelle anzugreifen, ist er identifiziert. Durch den Zusammenhang zwischen dem gezielten Suchen nach Informationen und dem darauf folgenden Angriffsversuch ist die Intention des Angreifers bekannt und beweisbar und er kann blockiert werden. Dieses Prinzip erinnert an bereits existierende Honey-Pots. Im Gegensatz zu diesen wird ein Einbruch noch rechtzeitig verhindert.

Die Vorteile dieser Methode liegen auf der Hand:

- Durch die Erkennung der Intention kann der Angreifer blokiert werden, noch bevor er Schaden eingerichtet hat. Die üblichen Methoden, wie das Einfügen temporärer Firewall Regeln, können bei rechtzeitiger Reaktion noch funktionieren.
- Das IPS kann sich auf die Überwachung von Zugriffen auf vorgetäuschte Schwachstellen beschränken. Auf diese Weise wird die Systemperformance weniger beeinflusst.
- Es müss nicht ständig die Signatur-Datenbank aktualisiert werden.
- Die Intentionserkennung liefert viel weniger Fehlalarme, da tatsächlich eine Interaktion zwischen dem Angreifer und dem IPS stattfindet und bewiesen werden kann.
- Da es keine falschen Alarme gibt, können sinnvolle statistische Auswertungen aus den erkannten Angriffen erzeugt werden.

Dieser Ansatz löst selbstverständlich auch nicht alle Sicherheitsprobleme, die bei ein klassisches IDS vorkommen. Ein fiktiver Angreifer, der genau weiß, wohin er will und auf jegliche Informationen verzichtet, würde auch von dieser Erkennungsmethode nicht erkannt.

4 Zusammenfassung

Intrusion Detection Systeme haben sich zu einem wichtigen Teil der Sicherheitsstruktur vieler Unternehmen entwickelt. Welche IDS-Architektur die beste Informationsquelle ist, kann man schwer sagen. Doch alle drei haben ihre Vor- und Nachteile. Ein Hybrid-IDS-System, das Netzwerk-, Host- und Application-IDS kombiniert, ist der beste Weg, die Stärken dieser Architekturen auszunutzen. Zusätzlich zu einem IDS sollte man von andren Abwehrmechanismen, wie Firewall und Antivirensoftware, nicht verzichten. Ein NIDS kann zu einem Flaschenhals für den Netzwerkverkehr werden, wenn es nicht über die entsprechende Bandbreite verfügt. Als Folge davon wird nicht nur die Übertragungsgeschwindigkeit “gebremst“, sondern es werden nicht alle Pakete analysiert. Ein HIDS wird eingesetzt, um die Sicherheit von bestimmten Rechner (Server) zu gewährleisten. Das HIDS belastet jedoch das Host-Betriebssystem. Die Tatsache, dass ein IDS ein Alarmsystem darstellt, aber keinen Schutz vor Angriffen bietet, ist das grösste Problem von IDS überhaupt. Auch die automatischen Reaktionen eines IDS finden statt, erst nachdem die Attacke bereits Schäden eingerichtet hat.

IPS bieten einen viel versprechenden Ansatz, um die Netzwerk- und Systemsicherheit im Vergleich zu IDS weiter zu erhöhen. Durch neue Erkennungstechnologien wird versucht, einen Angriff genauer und frühzeitig zu erkennen und zu blockieren, noch bevor er Schaden eingerichtet hat.

5 Ausblick

Die immer mehr zunehmenden Gefahren sind der Grund, warum die Interesse an Intrusion Detection und Intrusion Prevention ständig wächst. Die Angreifer entwickeln sich immer besser und in der nahen Zukunft werden sie nicht aufhören, Angriffscode zu schreiben. Es wird immer mehr notwendig, dass neue Sicherheitsstrategien entwickelt werden müssen.

Meiner Meinung nach kann eine Kombination aus den verschiedenen Technologien und Methoden, die hier beschrieben wurden, einen zuverlässigen Schutz leisten. In Anbetracht der Schwierigkeiten, die ein derartiges Hybridsystem bereiten kann, ist allerdings ein fundiertes Wissen über Technik und Prozesse notwendig. Ein Sicherheitsadministrator muss sein IDS/IPS-System und die Firewall richtig konfigurieren können. Die von diesen Systemen gesammelten Informationen müssen auch von jemand analysiert werden.

Die Verantwortung über die Sicherheit eines Firmennetzes muss man nicht dem Sicherheitsteam alleine überlassen. Alle Computernutzer müssen über die Gefahren informiert sein. Jeder, der ins Internet will, braucht eine Art “Internetführerschein“. Wenn alle Benutzer ihre E-Mail-Anhänge auf ihren Rechner speichern oder nicht vertrauenswürdige Web-Seiten aufsuchen, ohne dabei nachzudenken, sind auch als potenzielle “Angreifer“ zu betrachten. Ein hochinteligentes IPS kann nicht viel dagegen machen.

Auf der Applikationsebene könnte man durch die Software-Entwickler auch Hilfe schaffen. Es ist bekannt, dass die meisten Sicherheitsporbleme der letzten Jahren durch Browser hervorgebracht worden sind. Der beste Schutz gegen Buffer-Overflows ist sicherheitsbewusste Programmierung. Die Programmiersprache Java bietet eine Lösung dieses Problems, da die Java-Plattform die Grenzen der Speicherbereiche zur Laufzeit überwacht.

Literatur

[CERT04] CERT-Coordination-Center. CERT/CC Statistics 1988-2004, 2004.

[Ciro04] Cirosec. Der Wandel von Intrusion Detection zu Intrusion Prevention, 2004.

[fSid02] Bundesamt für Sicherheit in der Informationstechnik. Einführung von Intrusion Detection Systemen, 2002.

[fSid04] Bundesamt für Sicherheit in der Informationstechnik. Der praktische Einsatz von ID-Systemen, 2004.

[Götz03] Christian Götz. Intrusion Detection Systeme im Vergleich, 2003.

[HeML91] Heberlein, Mukherjee und Levitt. A Method to Detect Intrusive Activity in a Networked Environment. Proc. of the 14th National Computer Security Conference, 10 1991.

[HeML94] Heberlein, Mukherjee und Levitt. Network Intrusion Detection, 1994.

[HiMe01] Alexis Hildebrandt und Mathias Meyer. Intrusion Detection am Beispiel von Snort, 2001.

[Inte04] Integralis. Systems Security, 2004.

[McAf04a] McAfee-Security. IPS und interne Firewall, 2004.

[McAf04b] McAffe-Security. Mehrfache Buffer Overflow Sicherheitslücken in IIS und RAS, 2004.

[Möhl04] Tanja Möhler. Sicherheit muss nicht teuer sein, 2004.

[Neum04] Andreas Neumeier. Intrusion Detection und Intrusion Prevention, 2004.

[NoNo04] Stephen Northcutt und Judy Novak. *Network Intrusion Detection.* Hüthing. 2004.

[Radw04] Radware. Intrusion Prevention, 2004.

[Roes04] Marty Roesch. Snort, Freies, signaturbasiertes NIDS, 2004.

[Rogg03] Marko Rogge. BSI-Studie: Intrusion Detection Systeme, 2003.

[Seil03] Martin Seiler. Intrusion Detection ist kompliziert und ungenau, 2003.

[Snel03] Mark Snell. Intrusion Detection-Systeme: eine Einführung, 2003.

[Spen04] Ralf Spenneberg. *Intrusion Detection und Prevention mit Snort2 and Co.* Addison-Wesley. 2004.

[Tett04] Matt Tett. Detektive fürs LAN: Intrusion Detection-Systeme im Test, 2004.

[Wiki04] Wikipedia. Intrusion Detection System, 2004.

[Wölf04] Thomas Wölfer. Basiswissen Buffer Overflow, 2004.

Abbildungsverzeichnis

Anti-Spam Techniken

Nils L. Roßmann

Kurzfassung

Neben Viren und Würmern sind unerwünschte Werbemails (Spam), weiterhin eines der größten Probleme im Internet. Kaum eine Firma oder ein Provider kommt daher heutzutage ohne einen Spam-Filter aus. Ein Standard zur Vermeidung von Spam wurde trotz intensiver Bemühung noch nicht verabschiedet und bis auf ein paar Insellösungen werden hauptsächlich Mailfilter zum Aussortieren von Spam eingesetzt. Diese Arbeit stellt die verschiedenen Ansätze zur Vermeidung von Spam und zur Filterung von E-mails vor und beschreibt anhand des Beispiels SpamAssassin die Umsetzung eines zentralen Mailfilters im Rechenzentrum der Universität Karlsruhe. Aufgrund der guten Ergebnisse wird dabei vor allem der statistische Ansatz mittels Bayes-Filtern genauer erläutert. Außerdem wird auf die rechtliche Lage beim Einsatz von Mailfiltern eingegangen.

1 Einleitung

Mit zunehmender Nutzung von E-mails zur privaten und geschäftlichen Kommunikation wurden E-mails auch für Werbevermarkter immer interessanter. Das Medium E-mail ermöglicht es kostengünstig und automatisiert eine große Gruppe zu erreichen. Die E-mail-Adressen können automatisch auf Webseiten und in Newsgroups gesammelt oder von anderen Anbietern eingekauft werden. Im Usenet bürgerte sich für diese unerwünschten Werbemails der Begriff SPAM ein. Im Gegensatz dazu steht der Begriff HAM für erwünschte E-mails. Zurzeit liegt der Spam-Anteil bei eingehenden E-mails in der Universität Karlsruhe bei ca. 70%. Selbst wenn nur die E-mails berücksichtigt werden, deren Zieladresse im Bereich der Universität auch tatsächlich existiert, sind immer noch ca. 50% (entspricht derzeit ca. 60.000/Tag) aller eingehenden E-mails Spam. Nach einer Studie von Forrester Data im Auftrag der Business Software Alliance vom Dezember 2004 [For04] ist Spam in Deutschland dennoch recht erfolgreich. So gaben 32% der befragten Personen an, Spam-Mails für Software-Produkte gelesen zu haben und 29% haben bereits ein per Spam beworbenes Software-Produkt gekauft.

Spam kostet nicht nur wertvolle Zeit der Benutzer, sondern belastet auch die Router und Server. Darüberhinaus wird unnötig Speicherplatz belegt. Bei teilweise über 100 Spam-Mails pro Person ist eine produktive Nutzung des Mediums E-mail fast nicht mehr möglich. Die Gefahr, wichtige E-mails zu übersehen, verlangt nach einer Software, die den Nutzern die leidige Aufgabe des Aussortierens unerwünschter E-mails abnimmt. Daher setzen immer mehr Provider und Firmen zentrale Spam-Filter ein. Zum besseren Verständnis der einzelnen Verfahren ist in Abbildung 1 der Ablauf einer Mailübertragung dargestellt.

2 Techniken zur Spam-Bekämpfung

Die verschiedenen Techniken zur Bekämpfung von Spam kann man grob in zwei Kategorien einteilen: Verhindern von Spam und Filtern von Spam. Der Schwerpunkt dieser Arbeit liegt dabei auf dem Filtern. Daher werden die anderen Techniken nur kurz angesprochen.

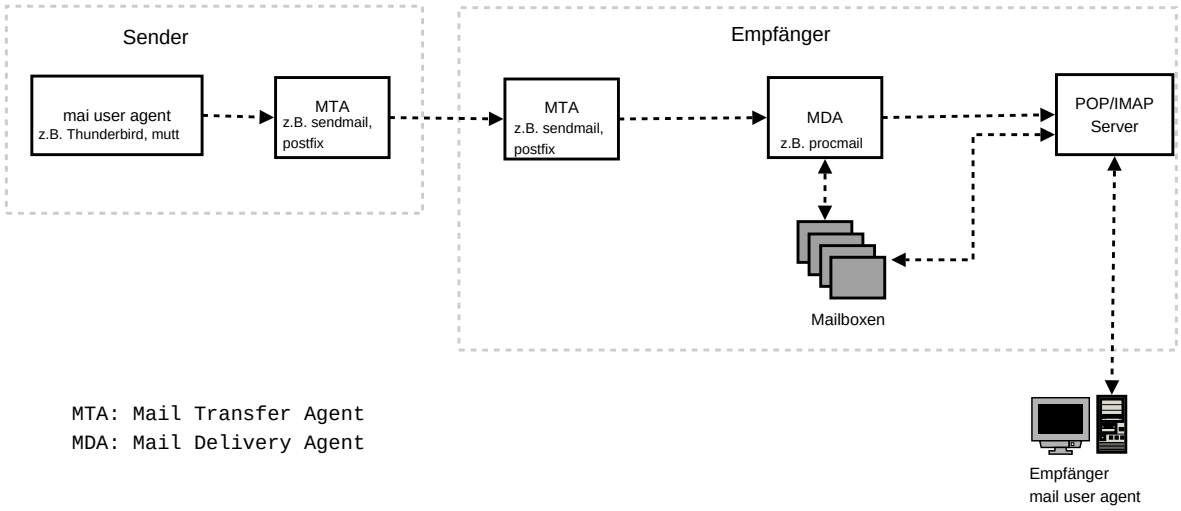


Abbildung 1: Typischer Ablauf einer Mailübertragung

2.1 Spam-Versand

Neben eigenen Systemen der Spammer oder Accounts bei großen Freemailern, werden zum Großteil unzureichend gesicherte fremde Rechner zum Spam-Versand mißbraucht. Die beiden am häufigsten verwendeten Wege sind dabei sogenannte Open-Relays sowie Backdoors in Viren und Würmern. Das Absichern der eigenen Systeme ist daher der erste Schritt zur Vermeidung von Spam.

2.1.1 Open Relays

Viele Spam-Versender versuchen ihre Identität zu verbergen und senden daher ihre E-mails über unzureichend geschützte Mailserver (Open Relays). Diese MTA's nehmen jede E-mail an, ohne zu überprüfen ob der Empfänger zum eigenen Netz gehört bzw. die E-mail aus dem eigenen Netz versendet wurde. Um es den Spammern nicht zu einfach zu machen, muss daher darauf geachtet werden, dass die eigenen Systeme entsprechend sicher konfiguriert werden. Außerdem nehmen viele Blacklist-Anbieter (siehe Abschnitt 2.3.2) open-relays in ihre Listen auf und so werden von den meisten größeren E-mail-Providern (wie z.B. web.de, gmx etc.) E-mails, die über open relays verschickt werden entweder gar nicht erst angenommen oder werden als Spam markiert.

2.1.2 Backdoors in Viren und Würmern

Ein weiteres Problem entsteht durch Rechner, die von Viren oder Würmern befallen wurden. In den letzten Monaten wurden vermehrt Backdoors in die Würmer integriert um Spam über befallene Rechner verteilen zu können. Außerdem gelangen die Spammer dadurch auch an eine Vielzahl gültiger E-mail-Adressen und verringern somit ihre bounce-Quote (nichtzustellbare E-mails).

2.2 Verhindern von Spam

2.2.1 Challenge-Response-Verfahren

Beim Challenge-Response-Verfahren muss sich der Absender beim Empfänger registrieren bevor die E-mail zugestellt wird. Dies geschieht z.B. indem er die Ziffer eines generierten Bildes zurückschickt oder in ein Web-Formular eingibt. Dieses Verfahren ist allerdings sehr umstritten und kommt nur sehr selten zum Einsatz. Würde es sowohl beim Absender als auch beim Empfänger eingesetzt, würden beide Systeme auf die Antwort des anderen warten (deadlock).

2.2.2 Greylisting

Eine weitere Möglichkeit ist das Greylisting-Verfahren. Bei diesem Verfahren wird eine E-mail von einem bisher unbekannten Absender erst einmal mit der temporären Fehlermeldung „450 you are greylisted - try again later“ abgewiesen. Das empfangende Gateway speichert die Envelop-Adresse des Absenders und des Empfängers sowie die IP-Adresse der Gegenstelle. Das SMTP-Protokoll [Klen] sieht vor, dass in solchen Fällen der Zustellversuch wiederholt wird. Das Gateway schaltet daher das Tripel (IP-Adresse, Absender, Empfänger) nach ca. 10 Minuten frei und die E-mail wird bei einem erneuten Zustellversuch angenommen. Das Tripel bleibt mehrere Tage gültig, so dass nur bei der ersten E-mail eine Verzögerung auftritt. Spammer versuchen im Allgemeinen nur einmal eine E-mail zuzustellen und werden somit abgewiesen.

Dieses Verfahren wurde von der Universität Würzburg erfolgreich eingesetzt und ist in [Völk04] ausführlich beschrieben. Dadurch ist es gelungen den Spam-Anteil von über 90% auf 35% zu reduzieren. Um Probleme zu vermeiden wird das Greylisting nur auf potentiell verdächtige E-mails angewendet. Also solche von Dial-up-Rechnern, E-mails mit fehlender From-Adresse und bei Unstimmigkeiten zwischen der Domain des Gateways und der Absenderadresse.

Sollte dieses Verfahren vermehrt zum Einsatz kommen, werden sich die Spammer aber vermutlich darauf einstellen. Auch wenn es darauf ankommt, dass E-mails sehr schnell zugestellt werden, ist Greylisting eher ungeeignet, da in dem SMTP-Protokoll [Klen] nur eine Mindestwartezeit von 30 Minuten festgelegt ist. Wie viel Zeit bis zu einem erneuten Zustellversuch vergeht, ist abhängig von der jeweiligen Implementierung in dem MTA.

2.2.3 Standardisierung

Zurzeit werden weitere Vorschläge diskutiert, wie in Zukunft Spam verhindert oder zumindest vermindert werden kann. Ansatzpunkte sind zum einen die Kosten für die Spammer und zum anderen die Vertrauenswürdigkeit der Mail-Gateways bzw. der Versender von E-mails.

Um die Kosten für Massenmailer zu erhöhen, könnte man z.B. für das Versenden von E-mails Geld verlangen (Micropayment) oder von jedem Absender verlangen, dass er auf seinem Rechner eine Rechenoperation ausführt so dass der Versand länger dauert. Diese beiden Konzepte sind allerdings praktisch nicht realisierbar. Für ein Micropayment-System müsste zuerst eine Abrechnungs-Infrastruktur aufgebaut werden und es müsste ein Kostenmodell gefunden werden, dass für alle akzeptabel ist und gleichzeitig den Versand von Massenmails unrentabel macht. Das Ausführen von Rechenoperationen scheitert ebenso an der praktischen Durchführbarkeit. Weder E-mail-Provider noch Firmen oder größere Institutionen werden die Rechenzeit ihrer Server mit sinnlosen Berechnungen vergeuden.

Vielversprechender sind die Vorschläge zur Vertrauenswürdigkeit. Das Domain-Key-Verfahren [Yaho04] von Yahoo, sieht vor, dass jede E-mail von dem Mail-Gateway der Absender-Domain mit einem privaten Schlüssel signiert wird. Der öffentliche Schlüssel ist frei zugänglich und ermöglicht so dem Empfangssystem die Echtheit überprüfen.

Die Verfahren Sender-ID [Micr04] und Sender Policy Framework (SPF) [IC G04] sehen vor, dass im Domain Name Service (DNS) nicht nur das Mail-Gateway einer Domain gespeichert wird, sondern auch welche Rechner dieser Domain E-mails versenden dürfen. Bei diesen Verfahren ist es also möglich, die Identität des Absenders sehr viel genauer zu bestimmen. Allerdings hindert dieses Verfahren die Spammer nicht daran, ihre Rechner als berechtigt einzutragen.

Die Organisation IETF hat die Arbeitsgruppe MARID [IETF04] mit dem Ziel gegründet, einen Standard zur Authentifizierung von Mailabsendern im DNS zu verabschieden. Favorisiert wurde zuletzt vor allem das SenderID-Verfahren von Microsoft. Aufgrund von Patentstreitigkeiten um das von Microsoft beführwortete Verfahren ist die Standardisierung jedoch gescheitert. Die Arbeitsgruppe hat sich daraufhin im September 2004 aufgelöst [Erme04].

2.3 Spam-Filter

Da es bisher kein standardisiertes Verfahren zur Vermeidung von Spam gibt, sind Spam-Filter weiterhin das Mittel der Wahl. Der Schwerpunkt dieser Arbeit beschäftigt sich mit dem Filtern von E-mails.

Aus Sicht des Endbenutzers lassen sich gefilterte E-mails in vier Kategorien einteilen:

- true negatives (Ham): Sowohl der Nutzer als auch der Mailfilter deklarieren die E-mail als erwünscht.
- true positives (Spam): Sowohl der Nutzer als auch der Mailfilter deklarieren die E-mail als unerwünscht.
- false positives: Der Mailfilter hat die E-mail fälschlicherweise als Spam deklariert, obwohl der Nutzer die E-mail empfangen möchte.
- false negatives: Der Mailfilter hat die E-mail nicht als Spam erkannt, obwohl es sich um Spam handelt.

Das Ziel jedes Spam-Filters besteht darin, Spam und Ham so sauber wie möglich zu trennen und false negatives sowie vor allem false positives zu vermeiden. Gerade wenn Spam-Filter gut funktionieren und die Benutzer sich darauf verlassen sind false positives sehr gefährlich, da sich der Benutzer in falscher Sicherheit wiegt.

Je nach Vorgehensweise kann man Spam-Filter wiederum in drei Kategorien einteilen: Regelbasierte Systeme, Black-/Whitelists, Lernende Systeme (Bayes).

Alle drei Verfahren werden im folgenden vorgestellt.

2.3.1 Regelbasiert

Regelbasierte Systeme analysieren die E-mail nachdem sie von dem MTA entgegengenommen wurde. Sie haben ein fest vorgegebenes Regelwerk anhand dessen sie den Header und den Body einer E-mail analysieren und bewerten. Jede Regel durchsucht die E-mail nach bestimmten Worten (z.B. porn, viagra, etc.) oder auf ein bestimmtes Muster (z.B. Großschreibung des

Betreffs, sehr viele Ziffern in der Absenderadresse etc.). Dieses Verfahren wird noch einmal ausführlich in Abschnitt 3.2 am Beispiel von SpamAsssin behandelt.

Regelbasierte Systeme haben den Nachteil, dass die Regeln fest vorgegeben sind und eine Anpassung an neue Spam-Formen aufwendig ist. Oft reicht es auch schon aus, wenn Suchbegriffe leicht verändert werden, z.B. p0rn anstelle von porn.

2.3.2 Black-/Whitelists

Bei Blacklists wird eine Liste aller Rechner bzw. Netze gepflegt, von der keine E-mails entgegengenommen werden (z.B. alle open relays). Häufig sind diese schon in den MTA integriert, sodass die E-mails von solchen Systemen gar nicht erst auf dem Server verarbeitet werden müssen und so die Ressourcen schonen. Meist wird dazu die Liste von externen Dienstleistern wie z.B. SpamCop verwendet.

Die Verwendung von Blacklists ist allerdings umstritten. Die Gefahr dabei ist, dass oft auch größere Provider fälschlicherweise auf solchen Listen landen und somit dessen Kunden kaum noch E-mails verschicken können.

Bei Whitelists wird eine Liste mit allen erlaubten Absendern/Rechnern gepflegt und alle anderen E-mails abgewiesen. Dies ist sehr aufwendig und verhindert die Kontaktaufnahme von bisher unbekannten Personen (z.B. neuen Kunden). Whitelists werden daher meist verwendet um false positives zu verringern (z.B. wenn bestimmte Absender immer wieder fälschlicherweise als Spam markiert werden).

2.3.3 Lernende Systeme (Bayes)

Im Gegensatz zu regelbasierten Filtern, müssen lernende Systeme vor dem Einsatz mit bereits klassifizierten Spam- bzw. Ham-Mails trainiert werden. Ansätze für Lernende Systeme sind zum einen neuronale Netze und zum anderen Bayes-Filter. Da neuronale Netze, im Gegensatz zu Bayes-Filtern, in Spam-Filtern fast keine Verwendung finden, werden diese hier nicht weiter behandelt.

Bayes-Filter basieren auf der Formel für bedingte Wahrscheinlichkeiten des Mathematiker Thomas Bayes:

$$P(A_i|B) = \frac{P(B|A_i) \cdot P(A_i)}{\sum_{j=1}^{n} P(B|A_j) \cdot P(A_j)}$$

Wobei $\{A_1, A_2, ..., A_n\}$ paarweise disjunkte Ereignisse sind und als „Ursache“ bzw. „Hypothesen“ für das Eintreten des Ereignisses B aufgefasst werden (siehe [HeKa00]).

Ein Bayes-Filter muss mit zuvor klassifizierten Ham- und Spam-Mails trainiert werden. Dabei wird für jedes Wort w die Wahrscheinlichkeit ermittelt, dass es in einer Spam-Mail (C=S) bzw. in einer Ham-Mail (C=H) vorkommt:

$$P(W = w|C = S) = \frac{S(w)/N_S}{S(w)/N_S + H(w)/N_H}$$

Wobei S(w) die Anzahl der Vorkommen des Wortes w in der Menge der Spam-Mails und H(w) die Anzahl der Vorkommen des Wortes w in der Menge der Ham-Mails bezeichnet.

N_S steht für die Anzahl der Spam-Mails und N_H für die Anzahl der Ham-Mails aus der Trainingsmenge.

Beim Analysieren einer neuen E-mail wird mit Hilfe der Bayes Formel ermittelt, wie wahrscheinlich es ist, dass es sich um eine Spam-Mail handelt. Dazu wählt der Filter aus jeder E-mail $M = \{w_1, w_2, ..., w_N\}$ automatisch die n aussagekräftigsten Wörter $\{\dot{w}_1, \dot{w}_2, ..., \dot{w}_n\} \subseteq M$ aus. Wörter sind aussagekräftig, wenn sie häufig in Spam-, bzw. Ham-Mails vorkommen. Also $P(\dot{w}_i|C = S) \approx 1$ bzw. $P(\dot{w}_i|C = S) \approx 0$ ist. n sollte dabei nicht zu groß gewählt werden (ca. 15-20 Wörter), da ansonsten die Gefahr besteht, dass größere Spam-mails, die größtenteils unverfänglichen Text enthalten, übersehen werden (siehe: [Grah03]).

$$P(C = S|\vec{W} = M) = \frac{P(C = S) \prod_{i=1}^{n} P(W = \dot{w}_i|C = S)}{P(C = S) \prod_{i=1}^{n} P(W = \dot{w}_i|C = S) \; + \; P(C = H) \prod_{i=1}^{n} P(W = \dot{w}_i|C = H)}$$

Bayes basierte Filter haben den großen Vorteil, dass sie sich an neue Spam-Typen schnell anpassen können und dass im Vorfeld keine Regeln erstellt werden müssen.

Diesen Vorteil erkauft man sich allerdings mit dem Nachteil, den Filter erst einmal mit einer großen Anzahl (mehrere hundert) möglichst aktueller Ham- und Spam-Mails trainieren zu müssen. Außerdem muss die Wortdatenbank ständig aktualisiert werden. Es sollten also möglichst alle falsch klassifizierten E-mails als neue Trainingsdaten verwendet werden.

3 Beispiel SpamAssassin

3.1 Warum SpamAssassin?

SpamAssassin ist zurzeit der populärste Spam-Filter für den Einsatz auf einem E-mail-Server. Er ist modular aufgebaut und kann vom Benutzer an die eigenen Wünsche angepasst werden. Neben regelbasierten Filtern unterstützt es außerdem Blacklisting und verfügt über ein Bayes-Modul. SpamAssassin kann sowohl an den MTA als auch an den MDA gekoppelt werden. Außerdem besteht die Möglichkeit ihn als POP3-Proxy zu verwenden.

SpamAssassin steht unter der GNU Public License und ist somit für jeden frei verfügbar. Eine ausführlichere Beschreibung findet man auf der Homepage von SpamAssassin [Spa] bzw. in dem Buch „SpamAssassin“ aus dem O'Reilly Verlag [Schw04].

3.2 Prinzip

SpamAssassin überprüft jede E-mail anhand einer Vielzahl von Regeln. Erfüllt diese die Regelbedingung, wird die Gewichtung (positiv oder negativ) der Regel zu dem Gesamtwert (Score) der E-mail addiert. Je höher dieser Wert ist, desto eher handelt es sich um Spam.

Diese Bewertung wird im Header der E-mail eingetragen. Außerdem kann noch eine Schwelle festgelegt werden, ab der SpamAssassin die E-mail als Spam klassifiziert:

```
X-Spam-Status: Yes, hits=13.0 tag1=3.0 tag2=5.1 kill=5.1 tests=BAYES_99,
  DATE_IN_FUTURE_24_48, HTML_70_80, HTML_FONTCOLOR_BLUE, HTML_FONTCOLOR_RED,
  HTML_FONT_BIG, HTML_MESSAGE, HTML_TITLE_UNTITLED, IMPOTENCE,
  MIME_BASE64_TEXT, MIME_HTML_NO_CHARSET, MIME_HTML_ONLY, PENIS_ENLARGE
 X-Spam-Level: ************
```

3.3 Regeln

SpamAssassin verfügt über einen fest eingebauten Satz von Regeln, kann aber auch um weitere Regeln ergänzt werden. Die Regeln überprüfen dabei sowohl den Header als auch den Body der E-mail. So wird der Header auf Einhaltung der Internet-Standards überprüft, der Body auf Phrasen, die häufig in Spam vorkommen (z.B. Make Money fast) aber auch auffällige HTML-Mails (JavaScript, fehlende End-Tags um Wörter zu zerhacken, etc.) führen zu einer hohen Bewertung. Weitere Regeln prüfen auf Vorhandensein in einer Black- bzw. Whitelist. Diese Überprüfungen werden meist online bei einem externen Dienstleister (wie z.B. SpamCop) durchgeführt und werden daher in SpamAssassin als Netzwerktests bezeichnet.

Für jede Regel lassen sich 4 unterschiedliche Scores festlegen:

- ohne Bayes, ohne Netzwerktests
- ohne Bayes, mit Netzwerktests
- mit Bayes, ohne Netzwerktests
- mit Bayes, mit Netzwerktests

Die Scores lassen sich zentral für das ganze System oder aber per Benutzer festlegen (in der Datei: .spamassassin/user_prefs im Homeverzeichnis des Benutzers). Falls keine Homeverzeichnisse existieren ist es auch möglich die Scores in einer SQL-Datenbank (MySQL, PostgreSQL, ODBC) zu speichern.

Ein Test, der beispielsweise überprüft, ob das Subject einer E-mail vollständig Großgeschrieben wurde sieht folgendermaßen aus:

```
header SUBJ_ALL_CAPS eval:subject_is_all_caps()
describe SUBJ_ALL_CAPS Subject is all capitals
Score SUB_ALL_CAPS 0.550 0.567 0 0
```

Und als weiteres Beispiel ein Test, der überprüft ob die Absenderadresse (From:) mit Ziffern beginnt:

```
header FROM_STARTS_WITH_NUMS From =~ /^\d\d/
describe FROM_STARTS_WITH_NUMS From: starts with nums
Score FROM_STARTS_WITH_NUMS 0.390 1.574 1.044 0.579
```

3.4 Bayes'sches Lernen

Der Bayes-Filter basiert auf den Arbeiten von Paul Graham [Grah02, Grah03] sowie den Verbesserungen (chi-quadrat Verteilung) von Gary Robinson [Robi03].

Das Bayes-Modul in SpamAssassin ist relativ unabhängig von dem bisherigen regelbasierten System. Es liefert aber genauso wie alle anderen Tests einen Wert zurück, der zu dem Gesamtwert addiert wird. Dies erfolgt über die BAYES_*-Tests. BAYES_00 ist erfüllt, wenn die Spamwahrscheinlichkeit zwischen 0% und 1% liegt. Dies führt bei der Bewertung (Score) zu einem Abzug von 1,665 oder 2,599 (je nach verwendetem Score). BAYES_99 ist erfüllt, wenn die Spamwahrscheinlichkeit zwischen 99% und 100% liegt. Dies führt zu einer Erhöhung des Scores um 1,886 oder 4,07.

In der Standardkonfiguration ist SpamAssassin so eingestellt, dass der Bayes-Filter erst nach dem Training von 200 Spam- und 200 Ham-Mails verwendet wird. In der Praxis hat sich allerdings gezeigt, dass mindestens 1000 Spam- und 1000 Ham-Mails notwendig sind (supervised training).

SpamAssassin unterstützt die beiden Strategien 'train everything' und 'train-on-error'. Bei der ersten Strategie werden alle zur Verfügung stehenden E-mails als Trainingsdaten verwendet. Wobei darauf zu achten ist, dass Ham- und Spam-Mails aus dem gleichen Zeitraum stammen. Ansonsten würde das Datum als Klasssifizierungskriterium verwendet. Nach ca. 10.000 trainierten Spam- und Ham-Mails ist es sinnvoll auf die train-on-error Strategie zu wechseln. Hierbei werden nur noch falsch klassifizierte E-mails trainiert.

Neben dem supervised-training unterstützt SpamAssassin auch noch das unsupervised-training. Beim eigenständigen Lernen geht SpamAssassin allerdings sehr konservativ vor. Es werden nur E-mails berücksichtigt, die einen sehr hohen Wert anhand der statischen Regeln erreichen (ohne die Werte des Bayes-Filters). Außerdem müssen mindestens drei Body- und drei Header-Regeln zutreffen und die E-mail darf nicht bereits durch den Bayes-Filter richtig eingeordnet worden sein (siehe [Gord04]).

Jede E-mail (header und body!) wird wie in Abschnitt 2.3.3 beschrieben in Tokens (Zeichenketten mit einer Länge von 3-15 Zeichen) zerlegt und in jeweils einer Datenbank für Wörter aus Spam-Mails und einer für Wörter aus Ham-Mails gespeichert. Tokens, die lange nicht mehr in einer E-mail vorkommen, werden zur Verbesserung der Performance entfernt.

Beim Überprüfen einer neu eintreffenden E-mail, wird auch diese in Tokens zerlegt und bis zu 150 der aussagekräftigsten Tokens werden zur Berechnung der Wahrscheinlichkeit (wie in Abschnitt 2.3.3 beschrieben) verwendet.

4 Einsatz im Rechenzentrum der Universität Karlsruhe

Im Rechenzentrum wird derzeit der ISP-Ansatz verfolgt. Das heißt, alle eingehenden und ausgehenden E-mails werden zentral über ein Mail-Server geleitet. Auf diesem Server wird SpamAssassin 2.6 inklusive Bayes-Filter eingesetzt. Es wird also die Spalte 3 der Scores verwendet (mit Bayes, ohne Netzwerktests). Im Juni 2005 ist ein Releasewechsel auf SpamAssassin 3.0 angedacht. Blacklisting wurde aus Performancegründen wieder deaktiviert. In Zukunft ist aber angedacht, die Blacklist lokal vorzuhalten. Auch der Einsatz von Greylisting ist aufgrund von Problemen mit der Geschwindigkeit nicht vorgesehen.

Die Ausbeute liegt inklusive des Bayes-Filters bei ca. 90%, wobei ohne Bayes nur ca. 40% des Spams ausgefiltert würde. Dies liegt unter anderem daran, dass E-mails, die Bayes_99 erfüllen, direkt als Spam markiert werden.

Als weitere Maßnahme werden nur noch E-mails angenommen, wenn der adressierte Empfänger im Bereich der Universität existiert (siehe [Preu04]). Dies reduzierte das Mailaufkommen um bis zu 250.000 E-mails pro Tag und der Spam-Anteil (SpamAssassin Score > 7) ging von 70% auf 50% zurück (siehe Abbildung 2).

Der Bayes-Filter wurde mit ca. 4000 Ham-Mails und 2300 Spam-Mails trainiert. Danach wurden nur noch falsch klassifizierte Mails zum Lernen verwendet. Das automatische Lernen wurde abgeschaltet, da mehr Spam- als Ham-Mails ankommen und somit einfache Worte durch die Masse an Spam eine falsche Gewichtung bekommen würden. Um die Erkennung zu verbessern sollen in Zukunft wieder vermehrt neue Ham-Mails einsortiert werden. Dies geht allerdings nur, wenn Benutzer ihre E-mails freiwillig zur Verfügung stellen. Auch ein vermehrtes Einsortierung von falsch klassifizierten E-mails Zurzeit melden allerdings nur sehr

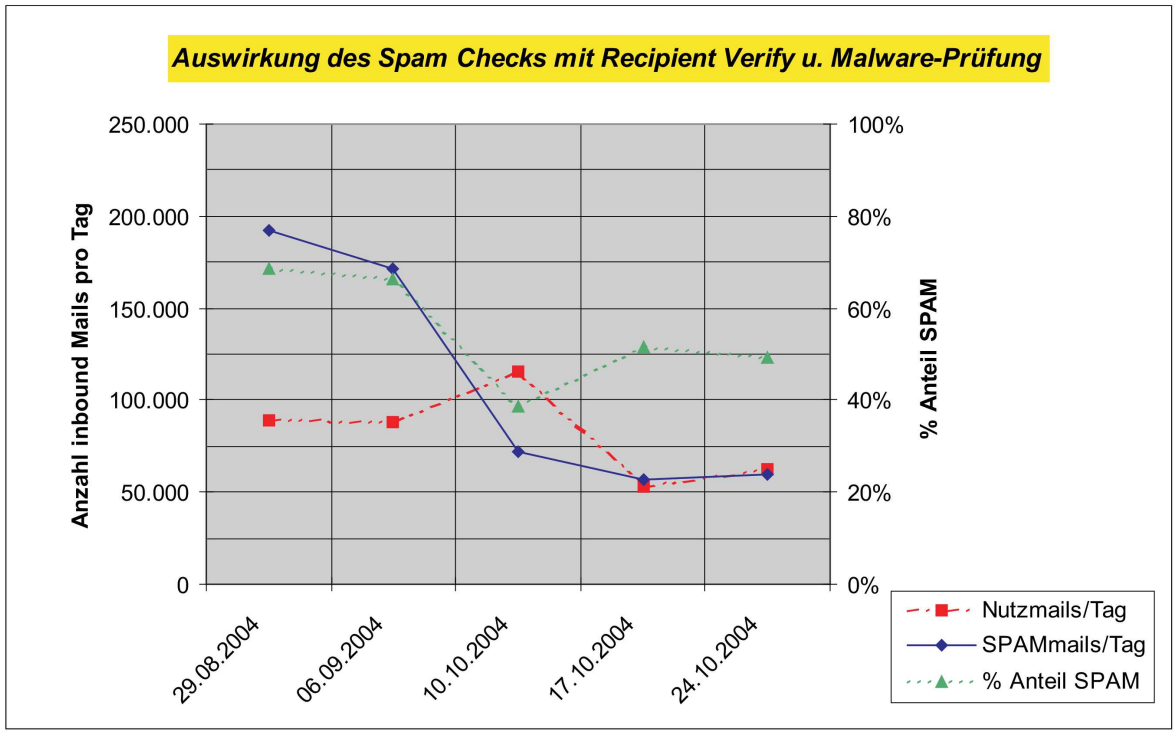


Abbildung 2: Mail Statistik RZ Universität Karlsruhe, aus [Preu04]

wenige Nutzer falsch sortierte E-mails dem Rechenzentrum. Nur wenn häufig false positives auftreten kommt es zu Beschwerden. Ein uniweites automatisiertes Einlernen durch die Nutzer ist trotzdem nicht vorgesehen, da durch falsches Einsortieren leicht alles verfälscht werden kann.

Die Standard Spam-Schwelle liegt derzeit bei 5 Punkten. Ab diesem Level werden alle E-mails in einen Spamverdachts-Ordner verschoben. Die Schwelle kann aber vom Benutzer über ein Web-Interface angepasst werden. Außerdem kann jeder Benutzer einstellen, nach wieviel Tagen die E-mails im Verdachts-Ordner gelöscht bzw. ab welchem Spam-Level die E-mail sofort gelöscht werden soll.

Trotz der guten Erkennungsrate gibt es auch noch einige Probleme. Viele false positives treten zurzeit vor allem bei englischen E-mails auf. Der Grund dafür ist noch nicht bekannt, es könnte allerdings an den zunehmend höflichen Formulierungen („Dear Sirs“) in Spam-Mails liegen. Diese werden durch die große Masse zunehmend ein Indikator für Spam und verfälschen somit das Ergebnis des Bayes-Klassifikators.

Des weiteren gibt es Probleme mit Cron-Jobs, die von dem Bayes-Filter meist als Spam deklariert werden. Um die Ergebnisse des Bayes-Filters nicht zu verfälschen, werden diese in einer Whitelist erfasst. In die zentrale Whitelist können auch sonstige problematische Tupel (To, From) eingetragen werden. Weiterhin sind auch HTML-Mails nicht unproblematisch, da HTML von SpamAssassin bestraft (positive Scores) wird. Abgesehen von den Cron-Job-Mails gab es mit kurzen E-mails im Gegensatz zu den Ergebnissen von Flavio Garcia [Flav04] keine Probleme.

Optimal wäre es, den Filter weiter zu personalisieren, wenn z.B. auch verschiedene Sprachen oder Länder geblacklisted werden könnten. Zentral ist es nicht möglich bestimmte Sprachen zu blacklisten, da vor allem mit den Ländern mit hohem Spam-Aufkommen (USA, China,...) ein sehr reger E-mail-Austausch besteht.

5 Analyse

5.1 Effektivität

5.1.1 Allgemein (Techniken)

Flavio Garcia [Flav04] kommt zu dem Ergebnis, dass für den Einsatz auf ISP-Ebene ein regelbasierter Algorithmus am besten geeignet ist. Auf Benutzerseite favorisiert er den Bayes-Ansatz mit den Verbesserungen von Gary Robinson [Robi03]. In diesem Artikel wird allerdings nicht untersucht, wie eine Kombination aus regelbasiertem- und Bayes-Filter abschneiden würde. Gerade diese Kombination hat sich im Rechenzentrum der Universität Karlsruhe sehr bewährt.

Um eine zentrale Filterung ohne größere Verzögerung bei der Mailzustellung zu gewährleisten, darf die Gesamtzahl der E-mails pro Tag nicht zu groß werden. Umso mehr E-mails direkt am MTA abgelehnt werden können, desto mehr Ressourcen stehen für eine anschließende Klassifizierung zur Verfügung. Am vielversprechendsten ist daher eine dreistufige Sortierung der E-Mails:

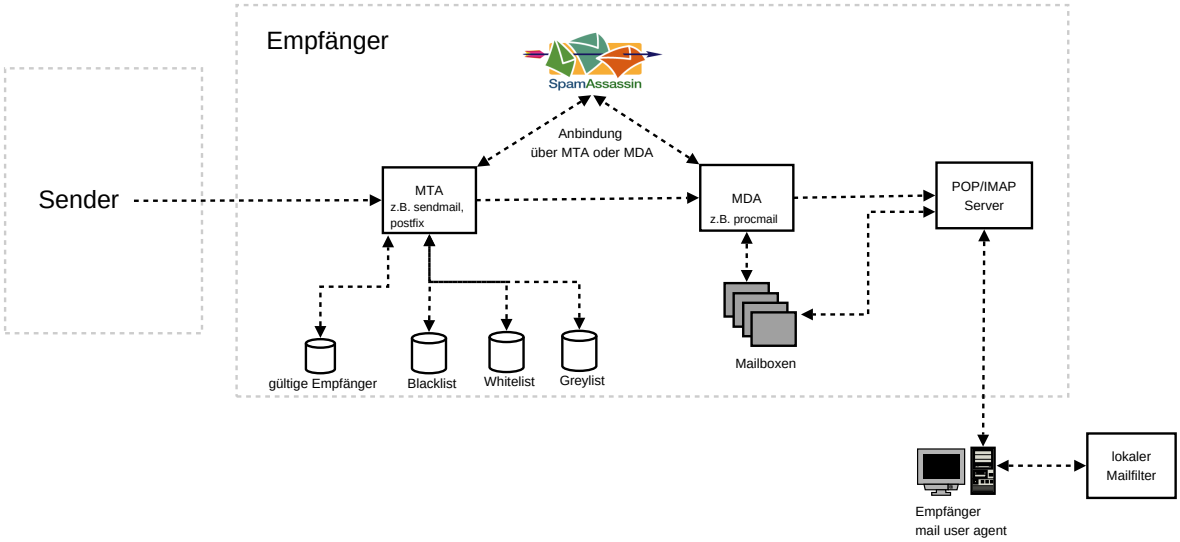


Abbildung 3: Dreistufiges Spamfilter-Konzept

1. Abweisen von E-mails, die nicht für das Zielsystem bestimmt sind, direkt am MTA. Bei Bedarf Abweisen aller E-mails von Gateways auf Blacklists.

2. Zentrales Filtern mittels eines regelbasierten-Filter sowie eine statistische Analyse mittels Bayes und einer zentralen Whitelist für bekannte Problemfälle. Gute bis sehr gute Ergebnisse liefert dabei SpamAssassin.

3. Feinsortierung im Mailprogramm des Benutzers.

Falls eine Verzögerung der Zustellung tolerierbar ist, könnte in Stufe 1 auch das Greylisting-Verfahren (siehe Abschnitt 2.2.2) zum Einsatz kommen. Am meisten profitieren werden davon aber nur die ISP's, die das solch einen Filter schnell genug implementieren, so dass sich die Spammer noch nicht daran angepasst haben.

5.1.2 SpamAssassin

SpamAssassin hat sich als zentraler Spam-Filter bewährt. Vor allem der Bayes-Filter trägt zu dem guten Ergebnis bei. Allerdings ist die Pflege sehr aufwendig, da ständig neue E-mails eingelernt werden müssen. Die Schwierigkeit bei SpamAssassin besteht vor allem darin, eine gute Schwelle zu finden, wann eine E-mail als Spam klassifiziert werden soll. In der Praxis hat sich bisher ein Wert zwischen 5 und 7 bewährt.

Die große Verbreitung von SpamAssassin führt außerdem dazu, dass Spam-Mails vor dem Versenden einer Bewertung mit SpamAssassin oder anderen gängigen Filterprogrammen unterzogen werden und bei Bedarf angepasst werden. Die Analyse der Header kann dadurch zwar nicht beeinflusst werden, aber dies führt zu einer deutlich niedrigeren Bewertung und ist je nach eingestellter Schwelle ausreichend, um eine Klassifizierung als Spam zu vermeiden.

5.2 Probleme

Auch die Spam-Versender verbessern ihre Methoden ständig. Früher wurden Spam-Mails fast nur in englisch verfasst und die Ausdrucksweise war sehr primitiv. Heute jedoch werden Spam-Mails immer mehr personalisiert. Die Sprache wird dem Empfängerland angepasst und die Texte werden höflicher formuliert, um möglichst unauffällig zu wirken.

Häufig kommt es auch vor, dass die Werbebotschaften in einem Bild untergebracht sind, um eine semantische Analyse zu verhindern. Auch lange gewöhnliche nichtsaussagende Texte dienen dazu die Wahrscheinlichkeitsrechnung des Spam-Filter zu überlisten. Daher ist es umso wichtiger bei Bayes-Filtern nur die aussagekräftigsten Wörter in die Berechnung mit einzubeziehen (siehe Abschnitt 2.3.3).

Die aber wohl wichtigste Frage bei der Klassifizierung ist die nach der Priorität auf false positives oder false negatives. Eine allgemeingültige Antwort gibt es darauf nicht. Es ist daher am sinnvollsten dies dem Benutzer zu überlassen.

Nachdem eine E-mail als Spam klassifiziert wurde, stellt sich die Frage was damit nun geschehen soll. Löschen, Markieren oder in einen extra Ordner verschieben? Die endgültige Entscheidung darüber sollte allein schon aus rechtlichen Gründen in jedem Fall dem Benutzer überlassen werden (siehe Abschnitt 6.2).

Falls die E-mails auf einem IMAP-Server verwaltet werden, ist es am sinnvollsten, wenn alle verdächtigen E-mails in einen Spamverdacht-Ordner verschoben werden. Werden die E-mails lokal auf dem Rechner des Benutzers verwaltet, sollte es dem Mailclient überlassen werden, wie er die klassifizierten E-mails behandelt. Oftmals ist es außerdem sinnvoll, wenn ein im Mailclient eingebauter Filter eine Feinsortierung vornimmt.

6 Rechtliche Fragen beim Einsatz von Anti-Spam Techniken

Soll in einer Firma ein zentraler Spam-Filter eingerichtet werden, müssen einige rechtliche Bestimmungen beachtet werden. Im Gegensatz zum Scannen von E-mails nach Viren und gegebenenfalls dem Löschen (wobei der Absender/Empfänger davon in Kenntnis gesetzt werden muß) von infizierten E-mails, kann man bei dem Filtern von Spam das Einverständnis des Benutzers nicht ohne weiteres voraus setzen.

6.1 Spam filtern

Bei der rechtlichen Bewertung muss unterschieden werden, ob die private Nutzung von E-mails erlaubt bzw. geduldet wird oder nicht.

Ist die private Nutzung erlaubt bzw. wird sie geduldet, sind Spam-Filter mit semantischer Auswertung (Stichwortsuche oder Bayes) unzulässig, da dies einen unerlaubten Eingriff in das Fernmeldegeheimnis darstellt (Quelle: [Fox04]). Aber auch wenn die private Nutzung nicht erlaubt ist, ist strittig, ob eine Betriebsvereinbarung ausreicht: „Ob auch bei ausschließlich dienstlicher Nutzung mit einer solchen Filterung ein Verstoß gegen das Fernmeldegeheimnis vorliegt, ist strittig: Der “Hinweis Nr. 37“ des Innenministeriums Baden-Württemberg verneint dies, ein Standpunkt, der aber nicht ohne heftigen Widerspruch geblieben ist und dem auch die amtliche Begründung zum § 206 StGB (im Begleitgesetz zum TKG) entgegensteht.“ [Fox04] Jeder Nutzer sollte daher der Filterung explizit und freiwillig zustimmen.

Findet hingegen keine semantische Analyse statt, ist eine Filterung rechtlich unbedenklich. Allerdings sind die Ergebnisse solcher Filter sehr viel schlechter.

6.2 Spam-Behandlung

Verdächtige E-mails dürfen nicht einfach gelöscht werden. Rechtlich unbedenklich ist das Markieren (durch zusätzlichen Headereintrag wie z.B. bei SpamAssassin) bzw. das Verschieben in einen Quarantäne-Ordner (nur mit Betriebsvereinbarung). Das Löschen von E-mails ist nur erlaubt, wenn der Benutzer dies explizit wünscht und die Konfiguration des Filters selbst vornehmen kann.

Ausführlich wird diese Fragestellung in dem Artikel „Leiden oder Löschen“ [Fox04] behandelt, dessen Einschätzung sich immer mehr durchsetzt.

7 Fazit

Eine vollständige Lösung des Spam-Problems ist noch nicht in Sicht. Die vorgestellten Filtertechniken sind zwar mit Erkennungsraten von ca. 90% recht effektiv, lösen aber nicht das grundlegende Problem. Dazu ist weiterhin ein einheitlicher Standard zur Verhinderung von Spam notwendig, der auch von allen umgesetzt wird. Aber selbst dann ist noch lange nicht gewährleistet, dass damit das Problem endgültig gelöst wird. Die bisherigen Vorschläge für einen Standard reichen allenfalls aus, um den Spammern den Versand zu erschweren. Filtertechniken werden daher weiterhin benötigt, um das Medium E-mail produktiv einsetzen zu können, wobei eine dreistufige Sortierung der E-mails zurzeit am erfolgversprechendsten ist. Technisch ist das Wettrüsten gegen die Spammer aber wohl nur schwer zu gewinnen. Neben einer ernsthaften gerichtlichen Verfolgung von Spammern, ist es außerdem notwendig die Empfänger von Spam dazu zu bringen, keine darüber beworbenen Produkte zu kaufen.

Literatur

[Erme04] Monika Ermert. Anti-Spam Arbeitsgruppe MARID der IETF steicht die Segel. http://www.heise.de/newsticker/meldung/51379, 2004.

[Flav04] Jeroen van Nieuwenhuizen Flavio D. Garcia, Jaap-Henk Hoepman. Spam Filter Analysis. In *Proceedings of 19th IFIP International Information Security Conference, WCC2004-SEC*. IFIP, 2004.

[For04] Verbraucher-Einstellung zu Spam in Deutschland. http://www.bsa.org/germany/kampagnen/loader.cfm?url=/commonspot/security/getfile.cfm&PageID=20692, 12 2004.

[Fox04] Dirk Fox. Leiden oder Löschen. *<kes> online* 2004(3), 2004. http://www.kes.info/archiv/online/04-3-006.htm.

[Gord04] Thomas Lynam Gordon Cormack. A Study of Supervised Spam Detection Applied to Eight Months of Personal Email. http://plg.uwaterloo.ca/ gvcormac/spamcormack.html, jun 2004.

[Grah02] Paul Graham. A Plan for Spam. http://www.paulgraham.com/spam.html, August 2002.

[Grah03] Paul Graham. Better Bayesian Filtering. http://paulgraham.com/better.html, Januar 2003.

[HeKa00] Prof. Dr. N. Henze und Priv.-Doz. Dr. D. Kadelka. *Skript zur Vorlesung Wahrscheinlichkeitstheorie und Statistik für Studierende der Informatik*. aug 2000.

[IC G04] Inc. IC Group. Sender Policy Framework. http://spf.pobox.com/, 2004.

[IETF04] IETF. MTA Authorization Records in DNS (marid). http://www.ietf.org/html.charters/OLD/marid-charter.html, 2004.

[Klen] John C. Klensin. *RFC 2821: Simple Mail Transfer Protocol*. http://www.faqs.org/rfcs/rfc2821.html.

[Micr04] Microsoft. Sender ID Framework Overview. http://www.microsoft.com/mscorp/twc/privacy/spam/senderid/overview.mspx, 2004.

[Preu04] Wolfgang Preuß. Spam- und Virencheck über zentrale Mailserver erfolgreich. *RZ news* Band 10/11, 2004, S. 5–5. http://www.rz.uni-karlsruhe.de/rd/4704.php.

[Robi03] Gary Robinson. A statistical approach to the spam problem. *Linux Journal* 2003(107), März 2003.

[Schw04] Alan Schwartz. *SpamAssassin*. O'Reilly. 2004.

[Spa] SpamAssassin: Documentation. http://spamassassin.apache.org/doc.html.

[Völk04] Dr. Roland Völker. Mit Greylisting gegen Spam vorgehen. *ix* 2004(12), 12 2004, S. 94–96.

[Yaho04] Inc Yahoo! Yahoo! Anti-Spam Resource Center. http://antispam.yahoo.com/domainkeys, 2004.

Abbildungsverzeichnis

Zertifizieren und Signieren mittels UNIKA-CA

Peter Leidinger

Kurzfassung

Dieses Dokument gibt einen Einblick in das Gebiet der Zertifizierung. In Zeiten der Globalisierung und des exponentiellen Anstieges digitaler Nachrichten steigt das Verlangen nach Authentizitätsprüfung. Immer größer werdende Prozentsätze heutiger Emails Fallen in die Kategorie „Spam“. Diesem Problem kann durch Zertifizierungen Abhilfe geschaffen werden. Zertifizierte Nachrichten können eindeutig zu einzelnen (juristischen) Personen zugeordnet werden. Leider wird noch sehr wenig Gebrauch von Zertifizierungs-Dienstanbietern gemacht. Im Folgenden wird die Sicherheit heutiger Zertifizierungen untermauert; die gesetzlichen Rahmenbedingungen kritisch diskutiert. Weiterhin wird ein Einblick in die Zertifizierungsinstanz UNIKA-CA der Universität Karlsruhe (TH), welche sich in die Zertifizierungskette der DFN-PCA einordnet, gegeben.

1 Motivation

Das Volumen des digitalen Nachrichten- und Informationsaustausches nimmt immer größere Maße an. Alleine in Deutschland gibt es zur zeit über acht Millionen .de-Domains. Die Kardinalität der Domains scheint, von 1994 angesehen, exponentiell anzusteigen (Abbildung 1).

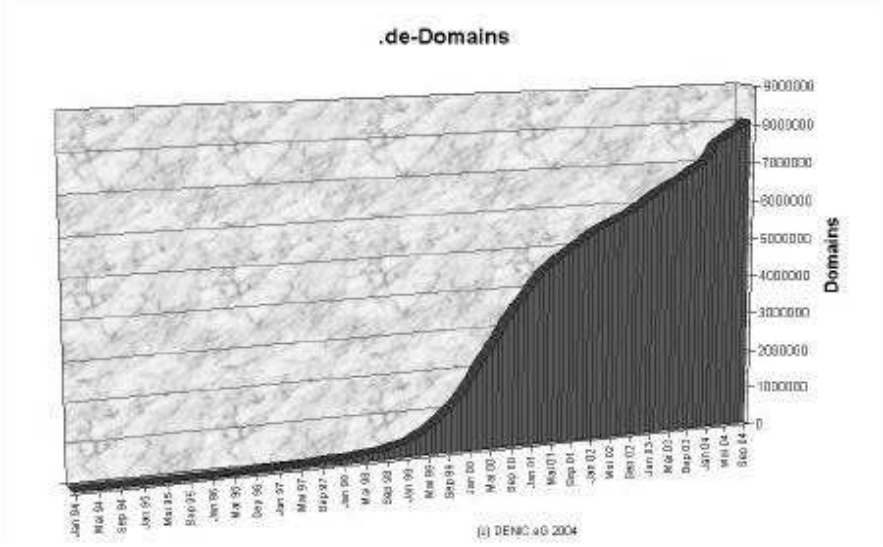


Abbildung 1: lineare Grafik der .de Domain-Entwicklung seit 1994 [t1105c]

Suchmaschinen decken zur zeit einen Suchbereich ab, der im Bereich von vielen Milliarden Web-Seiten liegt. Diese Vielzahl, die keiner Behörde mehr erschließbar ist, öffnet ein riesiges Portal für kriminelle Machenschaften jeglicher Art. Niemand ist als einzelnes Individuum in der Lage mehrmals am Tage die Authentizität einer Vielzahl digitaler Nachrichten persönlich zu prüfen. Dies mag bei der einen oder anderen Recherche nicht ins Gewicht fallen. Geht es

jedoch um offizielle Tätigkeiten oder um Transaktionen pekuniärer Art, so sieht dies anders aus. Würden Sie eine flüchtige Bekanntschaft damit beauftragen für Sie einen unbeträchtlichen baren Betrag zu Ihrer Hausbank zu bringen? Wohl nicht. Sicherlich würden Sie einer Internet-Seite, welche die exakt gleiche GUI Ihrer Hausbank nutzt schon eher trauen. Im Jahr 2003 haben in dieser Weise 1,8 Millionen Menschen vertrauliche Daten durch „Phishing“ an Dritte weitergegeben. Es entstand ein Schaden von 1,2 Milliarden Dollar [t1105b]. Wäre es nicht interessant für Sie, wenn Sie jemanden kennen würden, der für Sie die Authentizität solcher Seiten prüfen würde? Genau hier setzt die Idee der Zertifizierung der DFN-PCA (Deutsches-Forschungs-Netz Policy-Certification-Agency) ein. Hier ist eine regulierte Zertifizierungsinstanz gegeben, der Sie vertrauen können. Diese stellt Zertifizierungen aus, die einer Unterschrift mindestens gleichzusetzen sind.

2 Theoretische Grundlagen

Zu den klassischen Anforderungen der IT-Sicherheit zählen Vertraulichkeit, Integrität, Verfügbarkeit, Verbindlichkeit und Authentizität. Eine Zertifizierungsstelle (CA) soll hierbei alle Anforderungen, außer der Verfügbarkeit, abdecken. Hierzu prüft die CA die Identität des Senders und stellt ihm mittels digitaler Signatur ein Zertifikat aus. Dieses Zertifikat ist, durch entsprechende Verschlüsselung, fälschungssicher. Es liefert den eindeutigen Nachweis über die Zugehörigkeit eines öffentlichen Schlüssels zu einer Person. Ein Zertifikat enthält Informationen über den Namen des Inhabers, dessen öffentlichen Schlüssel, eine Seriennummer, eine Gültigkeitsdauer, den Namen der Zertifizierungsstelle [t1105a] und weitere CA-spezifische Eintragungen. Diese Daten sind in der Regel mit dem privaten Schlüssel der Zertifizierungsstelle signiert und können somit mit dem öffentlichen Schlüssel der Zertifizierungsstelle überprüft werden. Zertifikate für Schlüssel, die nicht mehr sicher sind, können über so genannte Sperrlisten (Certificate Revocation List, CRL) gesperrt werden. Die Integrität der, vom Sender verschickten, Nachrichten wird durch einen Hash-Code realisiert. Dieser wird der Signatur hinzugefügt. Die durch die CA zur Verfügung gestellte Infrastruktur wird, da mit public-key-Verfahren 2.1 signiert und verschlüsselt wird, als Public-Key-Infrastruktur bezeichnet. Im Folgenden werden nun die theoretischen Grundlagen, welche zum fundamentalen Verständnis einer CA notwendig sind, erläutert.

2.1 Public-Key-Verschlüsselung

Das Public-Key-Verfahren ist ein sicheres und gängiges Verfahren zur Ver- und Entschlüsselung digitaler Daten. Die Sicherheit begründet sich damit, dass die Faktorisierung sehr großer Zahlen (2048 bit) faktisch nicht in endlicher Zeit zu berechnen ist. Das Verfahren ermöglicht es ein Schlüsselpaar zu erzeugen. Hiervon wird einer, der „public key“, veröffentlicht. Der andere, der „private key“, wird, wie der Name schon sagt, geheimgehalten. Jede Nachricht, die von einem dieser Schlüssel verschlüsselt wird kann nur von dem zweiten Schlüssel des Schlüsselpaares entschlüsselt werden. Die hauptsächliche Schwachstelle dieses Verfahrens ist die tatsächliche Geheimhaltung des „private key“.

2.1.1 Der RSA-Algorithmus

RSA ist ein von Ron Rivest, Adi Shamir, Leonard Adleman entwickeltes asymmetrisches Public-Key-Verfahren. Es überstand eine jahrelange Kryptoanalyse und wird heute als Standardverfahren von fast allen gängigen Verschlüsselungssystemen eingesetzt. Zur Erzeugung des Schlüsselpaares generiert man zwei sehr große Primzahlen p und q aus welchen man das

Produkt $n = p * g$ bildet. Nun ermittelt man eine Zahl e, welche zu $(p-1)(q-1)$ teilerfrei ist. Die Zahlen n und e definiert man als öffentlichen Schlüssel (public key). Der geheime Schlüssel (private key) sei jene Zahl d, welche die Bedingung $(e * d)\ mod\ (p-1)(q-1) = 1$ erfüllt. Sei K die zu verschlüsselnde Nachricht. Dann ergibt sich die verschlüsselte Nachricht K' aus

$$K' = K^e\ mod\ n.$$

Um aus K' wieder K zu erhalten berechnet man

$$K = K'^(e)\ mod\ n.$$

2.2 Public-Key-Infrastruktur

Eine PKI liefer die Infrastruktur zum Verwalten und Nutzen von signierten öffentlichen Schlüsseln (Zertifikaten) asymmetrischer Schlüsselpaare [Rieg04]. Eine PKI unterliegt hierbei einer hierarchischen Struktur. Die Wurzel dieser Struktur liegt bei der CA-root. Diese zertifiziert sich selbst und all ihre Sub-CAs. Sub-CAs dienen den unterschiedlichen Anwendungsgebieten der Zertifizierungsstelle. Eine CA, die Emails und Web-Server signieren möchte, muss notwendigerweise zwei Sub-CAs gründen und zertifizieren. Eine „Email"-Sub-CA und eine „Server"-Sub-CA. Hierbei sind die unteren Glieder nur dann gültig, wenn die oberen Glieder der Kette gültig sind (Zertifizierungskette). Sub-CAs können weitere Sub-Sub-CAs nach gleichem Prinzip gründen. Die PKI kümmert sich hierbei um die Beantragung, Ausstellung, Sperrung und Verwaltung der Zertifikate. Um die Gültigkeit der Zertifikate zu überprüfen werden durch die PKI so genannte Sperrlisten (CRLs) eingerichtet. Verliert ein Zertifikat seine Gültigkeit (z. B. durch Missachtung des Geheimhaltung des privat-keys des Zertifikatnehmers), so wird dies durch die CRL bekannt gegeben. Die Richtlinien bezüglich Beantragung, Ausstellung, Verwaltung und Sperrung von Zertifikaten sind in den Policies der CA definiert 4.1. Dies betrifft insbesondere die Definition der Sicherheitsvorkehrungen, die, seitens der CA so wie auf Zertifikatnehmerseite, einzuhalten sind.

2.3 Standard Zertifikat-Formate

In der Anwendung spielen hauptsächlich zwei Zertifikat-Formate für öffentliche Schlüssel eine Rolle: Das in der ITU-Empfehlung X.509 in der Version 3 von 1997 [ITU97] standardisierte Zertifikat-Format und das Nachrichten- und Zertifikat-Format des Programms PGP („Pretty Good Privacy") [Zimm95] und der dazu kompatiblen Software.

2.3.1 X.509v3

Der Standard X.509 basiert auf dem 1991 herausgegebenen Verzeichnisdienst der X.500-Standard-Serie [Camp98]. Diese Serie enthielt ein Austauschformat für Public-Key-Zertifikate. X.509v3 ist die überarbeitete Fassung des Standards von 1997 [ITU97]. Er ermöglicht das Auftreten abweichender Internet-Namen für Rechner, Mailadressen oder WWW-Seiten anstelle nur eines einzigen Namens für den Inhaber oder Aussteller. Weiterhin sind generische Zertifikat-Erweiterungen vorgesehen. Diese ermöglichen das Einlagern weiterer Informationen in das Zertifikat. Dies ermöglicht die Nutzbarkeit des Zertifikats für noch unbestimmte Anwendungsbereiche. Eine schematische Darstellung eines X.509v3-Zertifikats ist in (Abbildung 2) gegeben. Wichtig ist hierbei, dass X.509v3-Zertifikate eine Zertifizierung von ungleichen Partnern vorsieht. Dies bedeutet, dass eine ausgezeichnete Instanz, also die CA, die Zertifizierungen vornimmt. Der umgekehrte Ansatz ist lediglich durch Cross-Zertifikate zu Realisieren. Solche Zertifizierungen ermöglichen die Zertifizierung einer CA von einer anderen und umgekehrt.

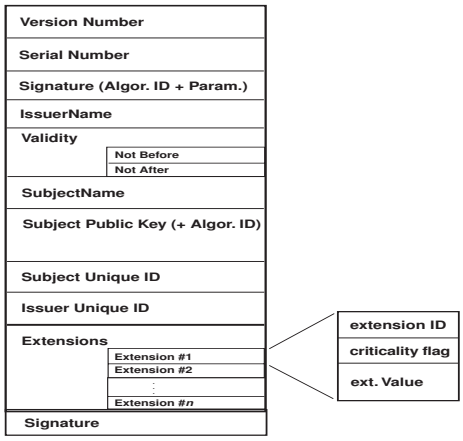


Abbildung 2: Struktur eines X.509v3-Zertifikates

2.3.2 PGP

PGP ist ein weiterer Standard, der vor allem von der Firma PGP in kommerziellen Software-Paketen verwendet wird. PGP wird hauptsächlich kommerziell genutzt und beinhaltet inkombatibilitäten in seinen Versionen. Der größte Unterschied zu X.509 besteht darin, dass der PGP-Standard eine Zertifizierung von gleich zu gleich vorsieht. Somit müssen also auch die Zertifizierungsstellen auf die normale PGP-Anwendersoftware zugreifen oder eine vorhandene API entsprechend erweitern [Camp98]. Ein PGP-Zertifikat ist in (Abbildung 3) zu sehen.

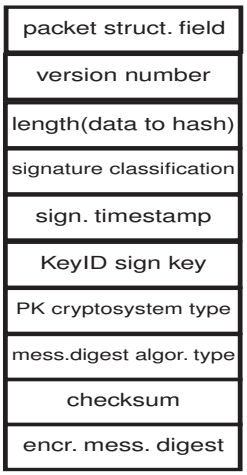


Abbildung 3: Struktur eines PGP-Key-Zertifikats [Camp98]

3 Rechtliche Grundlagen, Signatur-Gesetz 2001

Die rechtliche Regelung bezüglich „elektronischer Signierungen“ definiert sich im Signatur-Gesetz [Bund01]. Die Instanzierung einer, diesem Gesetz entsprechenden, Zertifizierungsstelle entspricht in keinem Fall den finanziellen und organisatorischen Möglichkeiten einer staatlichen Hochschule. Demnach ist die DFN-PCA, sowie alle unterhalb dieser liegenden CAs, zu denen auch die UNIKA-CA zählt, keine CA nach dem Signatur-Gesetz. Aus diesem Grunde wird das zugrundeliegende Gesetz an dieser Stelle diskutiert.

3.1 Allgemeine Bestimmungen

Das „Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz - SigG)“ definiert die Rahmenbedingungen für elektronische Signaturen. Es ist nur für diejenigen CAs verbindlich, die Zertifizierungen vornehmen, die nach Rechtsvorschrift dazu in der Lage sein müssen nach dem Signaturgesetz zu zertifizieren. Diese können durch weitere Rechtsvorschriften weiteren Bedingungen unterworfen werden. Das Gesetz unterscheidet zwischen „elektronischen Signaturen“, „fortgeschrittenen elektronischen Signaturen“ und „qualifizierten elektronischen Signaturen“. Erstere beschreiben Signaturen jeglicher Art. Die Folgende beschreibt diejenigen Signaturen, die den Inhaber authentifizieren, kontrolliert durch den Inhaber erzeugt worden sind und nachträgliche Manipulation(en) der zu signierenden Nachrichten erkennbar machen können. „qualifizierte elektronische Signaturen“ sind „fortgeschrittene Signaturen“, die auf einem qualifizierten Zertifikat beruhen und sicher erzeugt wurden. Des weiteren werden durch das Gesetz Begriffe der Informatik juristischer Nomenklatur unterworfen. Die Benennung jeglicher Zertifizierungs-Software sowie Hardware als „Signaturanwendungskomponenten“ (§2 Abs.11, SigG) mag zwar einen Juristen, aber nicht den Leser dieses Dokumentes, erfreuen. Aus diesem Grunde werden im folgenden üblichen Begriffe verwendet.

Der Betrieb einer CA nach SigG ist „genehmigungsfrei“. Allerdings müssen zur Betreibung eines Zertifizierungsdienstes (CA) folgende Bedingungen seitens der CA erfüllt sein:

- Bereitstellung einer Deckungsvorsorge in Höhe von 500.000 DM (250.000 EUR)
- Zuverlässigkeit und Fachkunde der CA.

Die zuerst aufgeführte Bedingung ist klar und deutlich. Um eine eventuell auftretende Haftungsklage begleichen zu können sind 250.000 EUR (der einfachheithalber wurden DM-Beträge bei allen Behörden mit dem Faktor 0.5 in den EUR-Betrag umgerechnet). Beim zweiten Punkt steckt sprichwörtlich der Teufel im Detail. Die Prüfung der Zuverlässigkeit und Fachkunde erfordert, ähnlich einer Wirtschaftsprüfung, Gutachten. Diese Gutachten sind nicht nur aufwendig und zeitintensiv. Sie sind von einem enormen organisatorischen sowie finanziellen Ausmaß. Dieses Ausmaß übersteigt das eines Rechenzentrums gewaltig.

3.1.1 Haftung

Grundsätzlich haftet die CA für den Schaden eines Dritten, sofern der Schaden durch Missachtung des SigG von Seiten der CA entstanden ist. Dies sind solche Schäden, die einem Dritten durch das Vertrauen in eine (oder mehrere) Zertifikat(e) entstehen, wenn das Zertifikat nicht den Bedingungen des SigG genügt. Die Haftung der CA tritt allerdings weder dann ein, wenn der Dritte über die fehlerhaften Angaben des Zertifikates informiert war, noch wenn die CA nicht schuldhaft gehandelt hat. Dies hat zur Folge, dass ein Geschädigter die Schuldfähigkeit der CA nachzuweisen hat. Des weiteren hat er glaubhaft nachzuweisen, dass er die Fehlerhaftigkeit des Zertifikates nicht erkannte. Es ist abzusehen, dass hier nur durch ein gerichtliches Urteil die Haftbarkeit der CA nachgewiesen werden kann. Ein solches Verfahren wurde allerdings bisher nicht geführt.

3.2 Freiwillige Akkreditierung

Die Akkreditierung eines Zertifizierungsdienstanbieters ist freiwillig. Entscheidet sich eine CA dafür eine nach SigG akkreditierte CA zu sein, so hat sie dies bei der zuständigen Behörde zu beantragen. Diese Behörde prüft dann die technischen und administrative Sicherheit der CA.

Ist die Sicherheit aus Sicht der Behörde gewährleistet, so kann die CA sich als „akkreditierter Zertifizierungsanbieter“ bezeichnen. Als solche kann Sie Zertifikate nach dem SigG erstellen. Die Sicherheit der nun akkreditierten CA wird allerdings in periodischen Abständen immer wieder von der zuständigen Behörde geprüft.

4 DFN-PCA

Die „Deutsche Forschungsnetz - Policy Certification Authority (DFN-PCA) existiert seit dem 01. Januar 2001 unter dem Dach der „DFN-CERT Services GmbH“ (bis 31. Dezember 2003: DFN-CERT: „Zentrum für Netzsicherheit“) [DFN-05]. Aufgabe der DFN-PCA ist der Aufbau einer DFN-weiten PKI 2.2. Hierzu hat sie eine eine CA für CAs sowie User aufgebaut. Im folgenden wird die Policy der DFN-PCA, welche für alle CAs innerhalb des DFN, sowie für CAs innerhalb der Zertifizierungskette der DFN-PCA liegen, definiert.

4.1 Policies

Grundsätzlich lassen sich CA-Policies (Richtlinien) in „Low-Level“-, „Medium-Level“- und „High-Level“-Security-Policy. Da eine „High-Level“-Security-Policy, oder gar eine nach SigG „akkreditierte“ Policy, für die Lebensdauer der DFN-PCA nicht erforderlich ist, verwendet man lediglich die „Low-Level“- und „Medium-Level“ Variante. Es sei angemerkt, dass die DFN-PCA sowohl PGP als auch X.509 unterstützt. Dieses Dokument befasst sich lediglich mit X.509-Zertifikaten.

4.1.1 Low-Level-CA Policy Version 1.6

Die DFN-PCA Low-Level-Policy ist für das DFN und seine Mitglieder maßgebend [DFN-05]. Sie stellt grundsätzlich nur Zertifikate für CAs aus; bietet jedoch Sub-CAs für Nutzer-Zertifikate an. Die Zertifikate genügen dem X.509v3-Standard. Weitere gebräuchliche Standards werden nach Ermessen unterstützt. Die dieser Policy zugrundeliegenden Anforderungen an technische Komponenten und Verfahren zur Zertifizierung genügen derzeit nur den Kriterien der ëinfachen digitalen Signaturnnach § 2 Nr. 1 SigG 2001 [DFN-05]. Die Policy erlaubt die Zertifizierung von CAs, Registrierungsinstanzen (RAs) und Benutzern. Die Root-CA für Low-Level-CAs ist die DFN-PCA. Diese ist (zur Zeit) in keine anderen Hierarchien eingebunden. Sub-CAs können sich allerdings mittels Cross-Zertifizierung (gegenseitige Zertifizierung mehrerer CAs) an andere CAs anbinden. Die CAs können eigene RAs zertifizieren. Diese können die Identitätsprüfungen der Antragssteller übernehmen. Es ist weder CAs noch RAs erlaubt Benutzer-Schlüsselpaare zu erzeugen. Hierdurch wird verhindert, dass die CA (oder RA) in Besitz des geheimen Nutzer-Schlüssels gelangt. Wird eine Zertifizierung an einer RA beantragt, so übermittelt diese den Antrag elektronisch an die CA weiter. Diese führt nun die Zertifizierung durch. Die Policy beinhaltet weiterhin Maßgaben zu den Sicherheitsanforderungen, denen eine CA genügen muss.

1. Anforderungen an eine DFN-PCA-Low-Level-CA

   - Zertifizierungsrechner off-line ohne Netzwerkanschluss
   - jeglicher Datenaustausch vom CA-Rechner zu anderen ausschließlich per Diskette oder Magnetband; automatische Datenbearbeitung ist untersagt; schlüsseltragende Datenträger sind unter sicherem Verschluss zu halten

- Signierschlüssel werden ausschließlich durch dezidierte Mitarbeiter am CA-Rechner generiert; Speicherung der Signierschlüssel nur auf externen Datenträgern (PIN-geschützt), die ausschließlich durch den CA-Rechner verwendet werden dürfen
- Signaturschlüssel dient lediglich für CA-Schlüssel, CRLs und Cross-Zertifizierungen
- Schlüsselpaare haben Mindestlänge von 2048 Bit
- Integrität der Daten und Programme der DFN-PCA wird ständig geprüft; Mitarbeiter gehen vertraulich mit Daten um
- Regelmäßige Datensicherung aller DFN-PCA-Daten.

2. Anforderungen an die von der DFN-PCA zertifizierten CAs

   - CA-Rechner muss vor unbefugtem Zugang geschützt sein
   - Schutz der geheimen Schlüssel wie bei DFN-PCA selbst
   - Signaturschlüssel dient für Generierung von Benutzer-Schlüsseln (!), CA-Schlüssel, CRLs und Cross-Zertifizierungen; CAs müssen ihre Schlüssel jedoch autark generieren
   - Empfohlene Schlüssellänge: 2048 Bit (1024 Bit in Ausnahmefällen auch möglich)
   - Erzeugt die CA Nutzer-Schlüssel, so ist sicherzustellen, dass der geheime Nutzer-Schlüssel nach Versenden an den Nutzer innerhalb der CA nicht mehr verfügbar ist
   - Getrennte Schlüssel zum Signieren und Verschlüsseln empfohlen
   - Mitarbeiter gehen vertraulich mit Daten um

3. Anforderungen an die von einer CA eingesetzten RAs

   - Sicherer RA-Rechner
   - Schlüssel der RA sind vor Mißbrauch zu schützen
   - Mindestlänge der Schlüsselpaaren von 2048 Bit (1024 Bit in Ausnahmefällen auch möglich) RSA bzw. 1024 Bit DSA/DSS
   - CA kann weitere Sicherheitsanforderungen an die RA stellen.

4. Anforderungen an Benutzer

   Der potentielle Nutzer muss folgenden Anforderungen genügen: Zum einen hat er die Geheimhaltung seines geheimen Schlüssels sicherzustellen; zum anderen muss er Schlüsselpaare mit einer Mindestlänge von 2048 Bit (1024 Bit in Ausnahmefällen auch möglich) RSA bzw. 1024 Bit DSA/DSS generieren.

Beantragt eine CA ein Zertifikat, so hat sie die hier genannte Richtlinien einzuhalten. Die Identitätsprüfung erfolgt durch persönlichen Kontakt eines ausgewählten, für den Betrieb der CA zuständigen, Mitarbeiters. Die maximale Gültigkeitsdauer solcher Zertifikate liegt bei drei Jahren. Die Zertifizierungsregeln für RAs sowie für Benutzer unterscheiden sich nicht. Die maximal Gültigkeit solcher Zertifikate erstreckt sich über zwei Jahre. Die Namensgebung muss strikt unterscheidbar sein (distinguish names, DN). DNs von CAs müssen dem Schema (C=DE, [L=DFN,] O=<Organisation> [,OU=<Abteilung>]) folgen. Für Benutzer gilt das Schema (C=DE, [L=DFN,] O=<Organisation>, [OU=<Abteilung>,] CN=<Vorname Name>).

4.1.2 Medium-Level-CA Policy

Die Medium-Level-CA stellt die eigentliche CA dar. Sie unterliegt, im Gegensatz zur Low-Level CA, höheren Sicherheitsansprüchen die im Folgenden nahegelegt werden. Bei der Medium-Level-CA findet der komplette Zertifizierungsvorgang innerhalb des RZ statt. Ein Zertifikats-Antragssteller muss folgenden Ansprüchen genügen, um ein Zertifikat zu erhalten: Zuerst muss er persönlichen Kontakt mit dem zuständigen CA-Mitarbeiter herstellen und sich mittels Personalausweis ausweisen. Des weiteren muss er angehöriger oder Projektpartner der Universität sein. Dies ist seperat mittels Studentenausweis oder entsprechenden Verträgen nachzuweisen. Des weiteren muss der Antragssteller über eine Universitäts-interne Mailadresse verfügen. Diese ist anzugeben. Der Nachrichtenaustausch wird über selbige Adresse stattfinden. Die Authentizität der Mailadresse wird von Seiten der CA geprüft. Nun kann der Antragssteller der CA seinen public-key via Email übermitteln. Hierbei hat er nachzuweisen, dass er über den passenden private-key verfügt. Dies kann durch die Verschlüsselung einer von der CA initialisierten Nachricht geschehen. Ist die Verschlüsselte Nachricht durch den bei der CA vorliegenden public-key zu entschlüsseln, so ist dieser Nachweis erbracht. Nun kann die CA den public-key zertifizieren und an die genannte Uni-interne Mail-Adresse dem Antragssteller übermitteln. Die CA ist grundsätzlich nicht verpflichtet eine Zertifikation durchzuführen. Sie kann ihren Dienst ohne Begründung verwähren. Die Medium-Level-CA macht weiterhin Vorgaben über den Vorgang der Zertifikatserstellung. Diese verlangt die Verschlusshaltung des geheimen Signierschlüssels auf einem Wechselmedium (Diskette, ZIP-Disk oder ähnliches). Das Medium ist nur während des Verschlüsselungsvorganges aus seinem Verschluss zu entnehmen. Weder die Software, noch der Rechner, auf dem die Signierung vorgenommen wird, darf das Rechenzentrum verlassen. Für den Zugriff auf den Signierschlüssel gilt das „Vier-Augen-Prinzip“. Der Zugriff darf nur von zwei befugten CA-Mitarbeitern zusammen erfolgen. Somit wird Fehlern und Betrug durch einen Mitarbeiter vorgebeugt. Die DFN-PCA sieht ausdrücklich keine Generierung von Nutzer-Schlüsselpaaren, durch die CA selbst, vor.

5 Praktische Umsetzung UNIKA-CA

Die UNIKA-CA stellt ausdrücklich keine akkreditierte CA nach SigG dar. Sie genügt lediglich den Kriterien einer ëinfachen digitalen Signaturnnach § 2 Nr. 1 SigG 2001 [DFN-05]. Dies beruht nicht auf Sicherheitsgründen. Vielmehr ist eine Akkreditierung aus finanziellen Gründen nicht durchführbar. Ein Zertifikat der UNIKA-CA genügt allerdings höchsten Sicherheitsanforderungen und ist einer Unterschrift mindestens Gleichwertig. Unterschriebene Dokumente können im Inhalt verändert werden; zertifizierte Dokumente sind in ihrem Inhalt nicht unbemerkt veränderbar. Die UNIKA-CA verwendet den X.509v3-Standard für Zertifikate. Sie befindet sich innerhalb der Zertifizierungskette der DFN-PCA. Dort verlieren die unteren Glieder der Kette (UNIKA-CA sowie deren Sub-CAs) ihre Gültigkeit, wenn ein oberes Glied der Kette (DFN-PCA-Root) seine Gültigkeit verliert. Hierbei ist zu beachten, dass aus diesem Grunde jede CA nur innerhalb der ersten Hälfte seiner Gültigkeitsdauer Zertifikate ausstellt. Ist diese Zeit abgelaufen, so wird eine neue Sub-CA instanziiert. Diese Sub-CA ist, bis auf den Namen, identisch mit der vorherigen CA. Sie unterliegt den gleichen Sicherheitsbedingungen. In unserem Falle ist die DFN-PCA die Root-CA. Sie ist insgesammt je drei Jahre gültig. Jede Sub-CA der DFN-PCA, wie zum Beipiel die UNIKA-CA, erhalten die halbe Gültigkeitsdauer der Parent-CA. Die UNIKA-CA ist somit eineinhalb Jahre lang gültig und kann ein viertel Jahr lang Zertifikate ausstellen. Es ist also prinzipiell gewährleistet, dass während der Gültigkeitsdauer einer Sub-CA in jedem Falle die Root-CA ihre Gültigkeit nicht verliert, selbst wenn die Root-CA keine Zertifikate mehr ausstellen kann. Dies ist als Prinzip der „vollständigen Einbettung“ bekannt.

5.1 UNIKA-CA Policy

Die Policy einer CA enthält bindende Vorschriften, die innerhalb der CA eingehalten werden um Zertifikate auszustellen. Sie unterliegen einer strengen Prüfung durch die PCA der CA. Für die UNIKA-CA ist dies die PCA des DFN. Die aktuelle Policy in Version 2.2 der UNIKA-CA wird im folgenden erläutert.
Die Policy definiert des weiteren die Sicherheitsanforderungen denen die UNIKA-CA genügen muss. Hierbei ist vorgeschrieben, dass der Zertifizierungsrechner „sicher“ sein muss. Die Sicherheit ist dann gewährleistet, wenn der Rechner vor unbefugtem Zugriff gesichert ist. Er darf sich ausschließlich innerhalb des RZ befinden und keineswegs über Netzwerkanschlüsse verfügen. Somit wird gewährleistet, dass nur derjenige Zertifikate ausstellen kann, der auch Zugriff zum Zertifizierungsrechner besitzt. Diese administrative Person ist einziger Inhaber des „private-keys“, welcher zur Erstellung eines Zertifikates notwendig ist. Dieser „private-key“ hat sich auf einem Passwortgeschützten Wechseldatenträger zu befinden (Smartcard, Disk, CD-Rom). Dieser Schlüssel darf auf keinen Fall das RZ verlassen. Hierbei sei angemerkt, dass sich der „private-key“ der UNIKA-CA in einem Tonnenschweren und mindestens 48-Stunden-Brandschutzsicheren Tresor befindet. Der Tresorschlüssel befindet sich in Besitz des Administrators der UNIKA-CA. Weiterhin hat der Signatur-Schlüssel ausschließlich der Signierung von CA-Schlüsseln zu dienen. Für Benutzerschlüssel und Widerrufslisten werden Sub-CA-Schlüssel verwendet. Die UNIKA-CA stellt grundsätzlich keine Benutzerschlüssel aus. Dies ist alleinige Aufgabe der Antragssteller. Der Signatur-Schlüssel besitzt eine Mindestlänge von 2048 Bit. Somit genügt er bei weitem den Sicherheitsanforderungen von heute und kommender Jahre. Des weiteren unterliegt jeder Mitarbeiter sowie Angestellte der UNIKA-CA der Einhaltung des Datenschutzes. Vertrauliche Daten der Zertifizierungsnehmer sind vertraulich zu behandeln. Diese Sicherheitsanforderungen gelten zum großen Teil auch für RAs.
Weiter schreibt die Policy vor, dass ein Zertifizierungsnehmer seine Schlüssel autark zu generieren hat. Hierbei ist ein asymmetrisches Verschlüsselungsverfahren zu wählen. Eine mindestlänge der Schlüssel von 1024 Bit ist hierbei penibel einzuhalten. Des weiteren ist ein Zerftifizierungsnehmer dazu verpflichtet, seinen „private-key“ geheim zu halten. Keineswegs darf er seinen „private-key“ an Dritte weitergeben. Bei Missachtung erlicht die Gültigkeit des Zertifikats. Der Zertifikats-Antragssteller muss sich einer Identitätsprüfung unterziehen. Hierzu genügt in den meisten Fällen der persönliche Personalausweiß. An dieser Stelle sei wiederum erwähnt, dass die CA nicht verpflichtet ist Zertifikate auszustellen. Sie kann ohne Angaben von Gründen die Vergabe von Zertifikaten verwähren. Die Namenswahl ist nach dem X.500-Standard zu generieren. Somit wird die unterscheidbarkeit der Namen sichergestellt. Zertifikate haben eine maximale Gültigkeitsdauer von einem Jahr und dürfen nicht verlängert werden. Der Antragssteller hat weiterhin ein schriftliche Teilnahmeerklärung zu unterzeichnen.

Die User-CA-Policy verwendet, wie alle anderen Policies der UNIKA-CA, die X.509v3-Policy. User-Zertifikate richten sich an Mitglieder der TH Karlsruhe und Fallen in den Zuständigkeitsbereich der UNIKA-U-CA. Dies ist die zweite Sub-Ca der UNIKA-CA. Um ein solches Zertifikat zu erhalten muss ein Zertifikats-Antragssteller über eine auf uni-karlsruhe.de oder uka.de endende email-Adresse verfügen (name@rz.uni-karlsruhe.de, name@stud.uni-karlsruhe.de, ...). Dies hat er durch die Beantwortung einer vom Administator der UNIKA-U-CA initialisierten Testnachricht nachzuweisen. Die Zertifizierung findet ausschließlich mittels der am RZ eingerichteten Online-Schnittstelle statt (Abbildung 5).

5.2 Vergabe von UNIKA-CA-Server-Zertifikaten

Die Server-CA (UNIKA-S-CA 2005-2006) ist eine von zwei Sub-CA der UNIKA-CA. Sie stellt seit dem 13.12.2004 neue SSL-Zertifikate nach der Server-CA-Policy aus. Um ein solches Zertifikat zu erhalten muss eine Request erstellt werden. Dieser Request ist durch Apache für Linux/Unix oder durch Microsoft IIS für Windows zu generieren. Hierbei werden nur solche Server zertifiziert, die im Umfeld der Universität Karlsruhe (TH) agieren. Der primäre DNS-Eintrag muss vom Rechenzentrum der Universität Karlsruhe (TH) gehostet sein. Das in Abbildung 4 ersichtliche Web-Frontend dient zur Übermittlung des erstellten Requestes an den Administrator der UNIKA-CA. Nach Bearbeitung des Requests seitens der CA ist am RZ mittel Ausweis eine Identitätsprüfung durchzuführen um das finale Zertifikat zu erhalten. Nach Bearbeitung des Requests werden die in 5.1 zitierten Prüfungen vorgenommen. Die

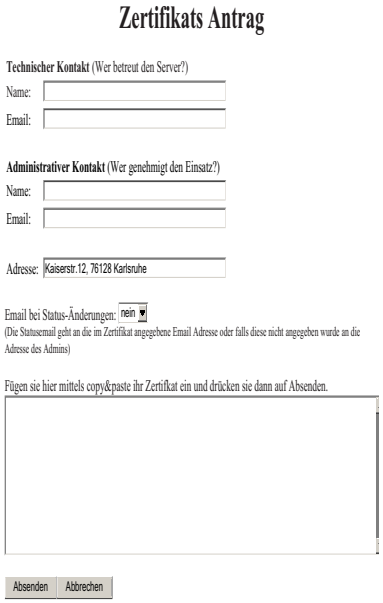


Abbildung 4: Teilnehmer-Erklärung zwischen UNIKA-CA und Benutzer (SSL)

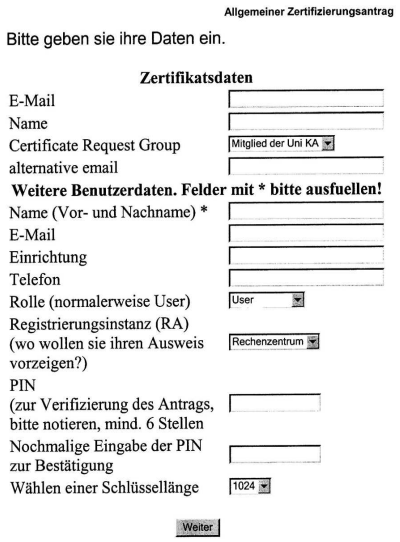
Gültigkeitsdauer eines so erhaltenen Zertifikats erstreckt sich über ein Jahr. Das Zertifikat wird nicht automatisch verlängert. Vor Ablauf der Gültigkeitsdauer sollte ein neues Zertifikat beantragt werden.

5.3 Erstellung von UNIKA-CA-User-Zertifikates

Der Antrag von User-Zertifikaten erfolgt online nach der User-CA-Policy 5.1. Das Antragsformular ist in Abbildung 5 ersichtlich. Die Bearbeitung der Requests wird durch zwei RAs durchgeführt. Dies ist zum einen die BIT8000 im Foyer des RZ, zum anderen die ATIS für ATIS-Benutzer. Die Gültigkeitsdauer der Zertifikate erstreckt sich, wie bei Server-Zertifikaten, über ein Jahr. Auch hier finden keinen automatischen Verlängerungen statt. Vor Ablauf der Gültigkeitsdauer muss sich der User um ein neues Zertifikat bemühen.

5.4 Kosten der UNIKA-CA

Der laufende Betrieb der UNIKA-CA wird durch einem wissenschaftlichen Mitarbeiter des Rechenzentrums und einer wissenschaftlichen Hilfskraft der Universität Karlsruhe (TH) getragen. Hierbei dient der wissenschaftliche Mitarbeiter zur Erstellung von Server-Zertifikaten.


Allgemeiner Zertifizierungsantrag

Bitte geben sie ihre Daten ein.

Zertifikatsdaten

E-Mail
Name
Certificate Request Group: Mitglied der Uni KA
alternative email

Weitere Benutzerdaten. Felder mit * bitte ausfuellen!

Name (Vor- und Nachname) *
E-Mail
Einrichtung
Telefon
Rolle (normalerweise User): User
Registrierungsinstanz (RA) (wo wollen sie ihren Ausweis vorzeigen?): Rechenzentrum
PIN (zur Verifizierung des Antrags, bitte notieren, mind. 6 Stellen
Nochmalige Eingabe der PIN zur Bestätigung
Wählen einer Schlüssellänge: 1024

Weiter

Abbildung 5: Teilnehmer-Erklärung zwischen UNIKA-CA und Benutzer

Hierzu veranschlagt er fünf Prozent seines Gesamtvolumens an Arbeitszeit. Die wissenschaftliche Hilfskraft arbeitet vierzig Stunden pro Monat an User-Zertifizierungen. Der Zertifizierungsrechner fällt finanziell nicht ins Gewicht. Der Tresor zur Aufbewahrung der Signatur-Schlüssel befand sich bereits vor Inbetriebnahme der UNIKA-CA ungenutzt im Besitz des Rechenzentrums. Somit beziffern sich die Kosten für den laufenden Betrieb der UNIKA-CA auf einige hundert Euro pro Monat.

5.5 Aktuelles Zertifikat - UNIKA-CA 2005-2008

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 94134675 (0x59c6193)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=DE, O=Deutsches Forschungsnetz, OU=DFN-CERT GmbH,
                OU=DFN-PCA, CN=DFN Toplevel Certification Authority
                /emailAddress=certify@pca.dfn.de
        Validity
            Not Before: Oct 25 11:31:17 2004 GMT
            Not After : Oct 25 11:31:17 2008 GMT
        Subject: C=DE, O=Universitaet Karlsruhe, OU=Rechenzentrum,
            CN=UNIKA-CA 2005-2008/emailAddress=ca@uni-karlsruhe.de
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:a0:90:29:5e:5e:5b:c4:d1:ca:bc:c7:75:3e:43:
                    ab:2c:ba:15:3c:88:96:33:a0:29:5d:2f:50:d9:b9:
                    3d:9d:b6:b2:9e:79:02:8b:33:c1:af:91:71:ac:13:
                    8c:c1:8e:0b:00:c7:71:8e:74:c8:d5:c5:7a:4e:68:
```

```
                    23:3e:54:9b:bb:86:21:d1:7f:98:ce:c3:1b:24:da:
                    90:d8:82:6a:da:c2:f0:d2:27:0c:e4:59:dd:b1:96:
                    4b:d5:9a:fa:31:53:7c:29:6c:02:ae:7f:53:bb:c1:
                    11:9b:01:02:b7:6d:ac:ce:62:cf:35:5a:fa:49:48:
                    61:9c:44:5c:27:eb:73:7a:48:fc:2d:4b:90:99:23:
                    7b:29:6a:ac:1e:69:33:56:b8:d8:84:ac:be:78:40:
                    56:b2:ea:52:77:28:d8:4c:52:b2:81:b4:d6:a9:73:
                    e7:a5:43:b3:02:43:f5:74:15:05:30:75:bf:0e:46:
                    fe:56:06:b3:11:1a:3a:69:73:cc:a9:7c:9c:cf:a5:
                    5d:ae:72:16:27:f8:ae:57:48:4b:9e:4e:60:01:b3:
                    be:9d:81:e5:c6:c9:fc:3c:69:96:86:91:f0:1a:64:
                    a1:2c:73:21:2f:44:85:f5:8c:8f:82:39:ed:46:ee:
                    11:b5:c1:4d:0c:86:9a:c6:25:4e:94:ce:c6:0a:b4:
                    22:7f
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage:
                Certificate Sign, CRL Sign
            X509v3 Subject Key Identifier:
                E2:F5:BA:97:7B:03:68:23:84:53:1F:93:ED:AE:AB:
                      36:9D:FB:A0:9A
            X509v3 Authority Key Identifier:
                keyid:06:0B:FA:B5:F8:48:78:A3:20:B1:0B:3E:CF:
                      A0:D0:C4:D1:7F:7D:D0
                DirName:/C=DE/O=Deutsches Forschungsnetz/OU=DFN-CERT GmbH
                        /OU=DFN-PCA/CN=DFN Toplevel Certification Authority
                        /emailAddress=certify@pca.dfn.de
                serial:15:CF:FD

            X509v3 CRL Distribution Points:
                URI:http://www.dfn-pca.de/certification/x509/g1/data
                      /crls/root-ca-crl.crx
                URI:http://www.dfn-pca.de/certification/x509/g1/data
                      /crls/root-ca-crl.crl

            Netscape Cert Type:
                SSL CA, S/MIME CA, Object Signing CA
            Netscape CA Policy Url:
                http://www.dfn-pca.de/certification/policies
                  /x509policy.html
            Netscape Comment:
                This certificate was issued by the DFN-PCA, the Top
Level Certification Authority of the German Research
Network (Deutsches Forschungsnetz, DFN).
The key owner's identity was authenticated in
accordance with the DFN-PCA x509 Policy.
            Netscape Revocation Url:
                https://www.dfn-pca.de/cgi/check-rev.cgi
            X509v3 Certificate Policies:
                Policy: 1.3.6.1.4.1.11418.300.1.1
```

```
                CPS: http://www.dfn-pca.de/certification/policies
                       /x509policy.html

    Signature Algorithm: sha1WithRSAEncryption
        52:86:f5:98:d2:31:e2:90:8b:19:b0:e0:ed:13:4d:87:01:81:
        ef:12:7c:72:1e:9b:6b:f7:55:fa:d0:2b:fb:25:68:7f:b6:6d:
        5c:6e:e6:c4:1b:a9:08:62:11:ce:e3:75:ae:bc:7d:b1:47:e5:
        3d:d9:02:c6:e2:15:7a:92:f1:d0:22:62:c7:71:8a:10:34:af:
        74:06:40:11:28:ea:76:72:12:b2:a4:cb:5d:90:dc:24:4f:bb:
        de:ae:8b:dc:d1:86:78:74:f4:8b:06:75:51:84:a7:ed:0d:47:
        b7:47:03:32:44:02:65:37:c1:62:5d:d0:6d:60:04:fd:bc:b2:
        ea:90:86:40:c4:c5:90:c9:a9:0b:d8:09:89:f9:6f:cd:31:18:
        0b:24:83:f5:3c:71:f4:8c:a0:89:36:98:63:ec:a3:24:b0:fa:
        55:53:1c:a8:b3:3a:38:db:28:47:2e:bb:0f:d6:7a:2b:6c:8e:
        29:39:93:0a:8b:ec:50:69:42:58:96:e0:c1:4d:13:ae:2f:09:
        37:4f:a6:dd:f2:01:66:12:b0:1a:91:07:fa:bc:e7:0e:be:0b:
        ed:f3:e5:95:f8:ee:97:ba:65:da:07:b7:92:43:ec:85:3f:f2:
        40:31:0b:3c:fe:aa:dd:c7:66:52:f3:80:11:9a:63:3c:b8:68:
        d4:4b:ed:b0
```

6 Ausblick

Durch Zertifikate lassen sich viele Probleme des täglichen Nachrichtenaustausches lösen. Würde die Versendung von unzertifizierten Emails untersagt werden, so wäre auf einen Schlag das „Spam“-Problem gelöst. Verteiler von Werbung sowie anderer ungewünschter digitaler Post könnten lokalisiert und bestraft werden. Dies würde zu erheblichen Einsparungen innerhalb der weltweiten Ökonomie bedeuten. Die Sender von Viren und Würmern könnten ebenfalls leichter lokalisiert werden. Am RZ der Universität Karlsruhe (TH) plant man derzeit Zugangspasswörter der Mitglieder durch signierte Schlüssel zu ersetzten. Solche Zertifikate könnten auf handlichen USB-Sticks statisch gespeichert werden. Somit könnte man unter anderem den Missbrauch der RZ-Rechner durch Unbefugte vorbeugen.

Literatur

[Bund01] Bundesregierung. *Signatur Gesetz*. Bundesdruckerei. 2001.

[Camp98] Igmar Camphausen. *DFN-PCA*. Springer. 1998.

[DFN-05] DFN-PCA. *http://www.dfn-pca.de - Policy Certification Authority (PCA) Die Zertifizierungsinstanz für das Deutsche Forschungsnetz*. DFN-PCA. 2005.

[ITU97] ITU. *ITU-T Recommendation X.509 (1997:E) The Directory: Authentication Framework*. 1997.

[Rieg04] Sebastian Rieger. *Public KEy Infrastrukturen nach X.509*. PKI-Workshop der GWDG. 2004.

[t1105a] *http://de.wikipedia.org/wiki/Zertifikat 2005*. wiki. 2005.

[t1105b] *http://www.3sat.de/3sat.php?http://www.3sat.de/boerse/magazin/71006/*. 2005.

[t1105c] *http://www.denic.de/de/domains/statistiken/domainentwicklung/*. 2005.

[Zimm95] Philip Zimmermann. *The Official PGP User's Guide*. MIT Press. 1995.

Abbildungsverzeichnis

Softwareverteilung

Danna Feng

Kurzfassung

In der Ausarbeitung wird zuerst die Notwendigkeit der Softwareverteilung gezeigt, dass sie erstens viel Zeit und Kraft spart und zweitens Software besser zentral verwaltet. Im zweiten Teil, der Hauptteil der Ausarbeitung werden die Remoteinstallation von Betriebssystem und die Verteilung von Anwendungen erklärt. Vor allem werden die technischen Möglichkeiten wie Protokolle vorgestellt. Dann wird das Verfahren von Remoteinstallation mit den Technologien RIS, ADS aus der Windows-Welt und FAI aus der Linux-Welt gezeigt. Nach der Bereitstellung von Betriebssystem können Anwendungen auch verteilt werden. Daher wird die Verteilung von Anwendungen anschließend vorgestellt. Zum Schluss des Teils wird der Verteilungsmechanismus des Rechenzentrums der Universität Karlsruhe erklärt. System und Software sollen vor Angriffen sowohl wegen Softwarelücken als auch wegen Viren immer auf den aktuellsten Stand gebracht werden. Daher wird im dritten Teil der Ausarbeitung Updatemanagement sowie Updatemechanismen des Rechenzentrums gezeigt. Bei der Verteilung von Software gibt es einige Sicherheitsprobleme, die im letzten Teil der Ausarbeitung gezeigt und die Lösungen entsprechend gegeben werden.

1 Motivation

Heutzutage ist es üblich, Software zu installieren und zu konfigurieren. Aber viele Benutzer verfügen nicht über genügende Berechtigungen und Kenntnisse, um Software zu installieren. Es ist schwer vorstellbar, dass jeder Student in den Poolraum des Rechnerzentrums geht und vor der Benutzung selbst ein Betriebsystem installiert. In den Unternehmern werden alle Installationen und Konfigurationen von einem oder mehreren qualifizierten Mitarbeitern ausgeführt. Falls ein Mitarbeiter oder eine kleine Gruppe von Mitarbeitern zu jedem Computer geht und einen nach anderem installiert und konfiguriert, ist der Aufwand wohl groß. Daher ist notwendig, diesen Prozess zu automatisieren.

Softwareverteilung ist eine Disziplin, die eine vollautomatische Installation, Konfiguration und Wartung von Betriebssystem und Anwendungssoftware für eine große Anzahl von Computern ermöglicht. Damit wird Software standardmäßig installiert und konfiguriert, Updates planmäßig unter Kontrolle durchgeführt und viele Arbeitskräfte und Zeit eingespart.

2 Remoteinstallation von Betriebssystem und anderen Software

2.1 Unterstützende Protokolle

Die Computer einer Organisation sind heute mit dem Netzwerk verbunden. Dies ermöglicht Installationspakete und Konfigurationsdateien über das Netzwerk zu übertragen. Viele Protokolle und Standards stehen zur Verfügung, damit Datenaustausch zwischen Computern möglich ist. Die für die Softwareverteilung wichtigen Protokolle sind BootP, PXE, DHCP, TFTP, die auf der Anwendungsebene im ISO/OSI-Basisreferenzmodell stehen, wie in der Abbildung 1 gezeigt wird. Im Folgenden werden die Protokolle jeweils erklärt.

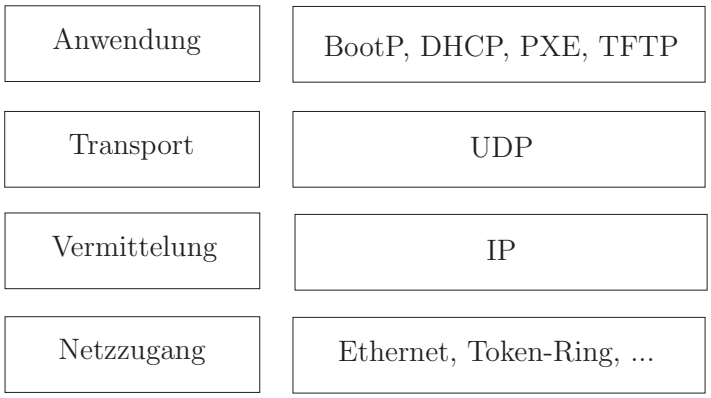


Abbildung 1: ISO/OSI-Basisreferenzmodell

2.1.1 BootP (Bootstrap-Protocol)

Das BootP kann die IP-Adresse einem plattenlosen Computer in einem Netzwerk zuweisen. Es verwendet UDP-Port 67 und 68, wobei UDP 67 von BootP-Server und UDP 68 von BootP-Client festgelegt wird.

Beim Einschalten eines Computers kennt er weder seine IP-Adresse noch die des BootP-Servers. Er sendet eine Bootanforderung mit seiner MAC-Adresse. Die IP-Adresse des Absenders wird als 0.0.0.0 eingesetzt und die Empfängeradresse lautet 255.255.255.255, was im eigenen Netz als Broadcast interpretiert wird. Die Bootanforderung von einem BootP-Client wird immer auf dem Zielport 67 gesendet und der Client lauscht auf dem Port 68, auf dem die Boot-Antwort gesendet wird. Auf der Serverseite steht eine Datenbank zur Verfügung, in der die MAC-Adressen aller vorher registrierten Computer und die Zuordnung der IP-Adresse gespeichert werden. Nachdem der BootP-Server die Anforderung auf dem entsprechenden Port bekommen hat, überprüft er die MAC-Adresse und vergleicht die mit den Einträgen in der Datenbank. Falls er den richtigen Eintrag findet, sendet er eine Bootantwort wieder als Broadcast zurück. Die Antwort enthält die MAC-Adresse des Clients, die IP-Adresse des Clients, die IP-Adresse und Hostname des Boot-Servers, der Name und Pfadangabe der Bootdatei und der Name des Verzeichnisses, das vom Kernel über NFS als Boot-Partition eingebunden werden soll. Der Client lauscht auf dem vorgegebenen Port und wartet auf die Antwort. Wenn das Paket ankommt, überprüft er die MAC-Adresse, die das enthält, mit seiner MAC-Adresse. Falls die beiden übereinstimmen, erkennt der Client, dass die Antwort für ihn ist. Mittels der IP-Adresse des Boot-Servers kann der Client eine Anforderung an ihn senden und fordert die Boot-Datei an, die den Kernel, der zentraler Bestandteil eines Betriebssystem ist, enthält.[compa]

2.1.2 DHCP-Protokoll (Dynamic Host Configuration Protocol)

Das DHCP-Protokoll (Dynamic Host Configuration Protocol) ist die Fortsetzung vom BootP. Es ermöglicht eine dynamische Zuweisung der IP-Adresse und anderer Parameter an Computer in einem Netzwerk, ohne jedem Computer manuell Adressen zuzuweisen. Durch DHCP ist die Einbindung eines Computers mit dem Netzwerk ohne weitere Konfiguration möglich. Ohne DHCP muss man außer IP-Adresse noch die Parameter wie Netzmaske, Gateway angeben. Beim Einschalten eines Computers wird eine Nachricht, die so genannte DHCPDiscover-Nachricht als Broadcast gesendet. Die DHCP-Server im Netzwerk reagieren darauf und senden eine DHCPOffer-Nachricht mit dem Vorschlag für eine IP-Adresse zurück. Sofern der Client einen Vorschlag der IP-Adresse bekommen hat, sendet er eine DHCP-Anforderung an den Server zurück, auf die der Server ihm, in einer DHCPACK-Nachricht, die IP-Adresse mit den

weiteren relevanten Daten übermittelt. Der Client erhält noch eine Lease-Zeit in der Nachricht, die zeigt wie lange die zugeteilte IP-Adresse für ihn reserviert ist. Vor dem Ablauf der Lease-Zeit sendet der Client eine Anforderung an eine IP-Adresse und der Server wird merken, ob dem Client schon eine IP-Adresse zugewiesen wurde. Falls ja, wird eine DHCPACK-Nachricht mit derselben IP-Adresse und Parametern und eine verlängerte Lease-Zeit zurück gesendet. Die Zuweisung von einer IP-Adresse wird nach dem Ablauf zurückgesetzt und damit wird realisert, dass nicht mehr gebrauchte IP-Adressen wieder nutzbar sind.[compb]

2.1.3 PXE (Preboot Execution Environment)

PXE (Preboot Execution Environment) ermöglicht Booten eines festplattenlosen Computer über ein Netzwerk. Die Vorrausetzungen sind eine PXE-fähige Netzwerkkarte und ein Server, der DHCP und TFTP unterstützt. Wenn ein Computer mit einer PXE-fähigen Netzwerkkarte eingeschaltet wird, startet das BIOS eine Routine auf dem Flash-Chip der Netzwerkkarte und der Computer versucht eine IP-Adresse mittels DHCP zu bekommen. Dann wartet der Client auf den Dateinamen und den Pfad der Datei, die er von einem TFTP-Server herunterladen und ausführen kann. Da der Flash-Chip normalerweise klein ist, wird zuerst ein Bootloader-Datei heruntergeladen, der weitere Dateien von dem TFTP-Server laden kann, damit der Bootvorgang weiter ausgeführt werden kann.[compc]

2.1.4 TFTP (Trivial File Transfer Protocol)

Das TFTP (Trivial File Transfer Protocol) gehört auch zur Anwendungsebene und ist ein Datenübertragungsprotokoll. Im Gegensatz zu FTP unterstützt das TFTP nur das Lesen oder Schreiben von Dateien. Es können keine Zugriffsrechte vergeben, vorhandene Dateien angezeigt oder Benutzer authentifizieren. Beim Starten des Betriebssystems spielt es aber eine wichtige Rolle. Via DHCP bekommt der Computer die IP-Adresse des Bootservers und Pfad bzw. Name der Bootdatei. Er kann dann mittels TFTP die Datei, die den Kernel eines Betriebssystems enthält, vom Server herunterladen. Dann wird der Kernel gestartet. [compe]

2.2 Remote Installation des Betriebssystems

Zum Installieren eines Betriebssystem kann man entweder CD/DVD oder Protokolle zum Netzwerkbooten von Computer verwenden. Im Rahmen der Seminararbeit wird nur die Methode mittels Netzwerkbooten betrachtet, weil es sich um remote Installation handelt.

Bei der remote Installation zieht das BIOS des Computers zuerst vom Netzwerk die notwendige Programme mittels TFTP und führt diese aus. Dann wird das Installationsprogramm (der Kernel, ein Image vom Betriebssystem oder nur Dateien) von Betriebssystem von einem Server im Netzwerk geladen. Die Anforderung an den remote Installation-Dienst ist, dass man ein Betriebssystem über das Netzwerk automatisch und angepasst auf einem neuen Computer installieren kann. In der Windowswelt werden die Remote-Installations-Technik RIS und ADS und in der Unixwelt FAI (Fully Automatic Installation) im folgenden vorgestellt.

2.2.1 RIS (Remote Installation Service)

RIS-Server wird nur von den Plattformen Windows 2000 Server und Windows 2003 Server unterstützt und braucht Unterstützung von DNS (Domain Name System), DHCP-Server und Active Directory. Der RIS-Server verwendet DNS zur Auffindung der Active-Directory-Verzeichnis-dienste und zum Abschließen von Domänenoperationen. Mittels DHCP-Server

kann ein zu installierender Computer vor dem Kontakt mit dem RIS-Server seine IP-Adresse und die IP-Adresse vom RIS-Server bekommen. Active Directory ist der Verzeichnisdienst in Windows und speichert Informationen zu Netzwerkobjekten und implementiert den Dienst, diese Informationen für Benutzer verfügbar zu machen. Alle Computer und Benutzer in einem Standort können in dem Active Directory gefunden werden und werden in Domänen und Organisationseinheiten weiter kategorisiert.

RIS besteht aus verschiedenen Diensten:

- BINL (Boot Information Negotiation Layer) - Dieser Dienst empfängt und beantwortet DHCP-Anforderungen. Die Anforderungen des Clientinstallations-Assistenten werden damit auch bearbeitet. BNL verweist den Client auf die Dateien, die zum Starten einer OS-Installation benötigt sind.
- TFTPD (TFTP Daemon): Der RIS-Server verwendet TFTP zum Herunterladen der für das Starten der Installation benötigten Dateien über Netzwerk. Die Dateien sind Clientinstallations-Assistenten und alle zum Starten von Setup benötigten Dateien.
- SIS (Single Instance Store): SIS-Dienste bestehen aus einem NTFS-Dateisystemfilter und einem Dienst, der als der Datenträger mit dem RIS-Abbildern bezeichnet wird. Damit werden die doppelten Dateien zusammengelegt und Speicherplatz eingespart.[1]

Vor der Installation müssen die Betriebssystemabbilder (Images) bereitgestellt werden. Es existieren zwei Abbildtypen: CD-basierte Abbilder und 'komplette' Abbilder (RIPrep). Ein CD-basiertes Abbild ist eine Kopie des Inhalts der Betriebssystem-CD-ROM auf dem RIS-Server. Ein RIPrep-Abbild ist das Abbild einer kompletten Standarddesktop-Konfiguration. Ein Abbild enthält die Betriebssystemkonfiguration, Desktopanpassungen und lokal installierte Anwendungen. Das CD-basierte Abbild ist einfach zu installieren und kann mit Script angepasst werden. Aber bei der Installation auf vergleichbarer Hardware dauert es länger als das RIPrep-Abbild. Nur wenn die Hardwareausstattungen von Computern gleich sind, kann man das RIPrep-Abbild benutzen.

Eine unbeaufsichtigte Installation von Windows 2000 mit CD-basierte Abbild wird mit Windows 2000 Setup und einem angepassten Script ausgeführt. Diese Textdatei enthält die Informationen, die Windows 2000 vom Benutzer während der Installation oder des Upgrades abfragt. Trotz des Scripts kann eine oder mehrere Benutzereingaben erforderlich sein. Zuerst muss man das Script erstellen, dann wird das Script mit einem Betriebssystemabbild verknüpft. Das Tool Setup Manager dient zur Erstellung von solchen Scripten. Man kann auch mit einem Texteditor die Scripten den Anforderungen anpassen.

Der Installationsvorgang mit RIS wird in der Abbildung 2 gezeigt:[2] Alle Clientcomputer müssen vor der Installation im Active Directory vorbereitet sein, das heißt, die Computerkontoobjekte von den Computern müssen bereits im Active Directory erstellt sein. Bevor ein RIS-Server die Anforderungen von Clients annehmen kann, muss der Server vom Active Directory autorisiert sein. Erst nach der Autorisierung kann man RIS-Server konfigurieren. In einem lokalen Netzwerk können mehrere RIS-Server vorhanden sein. Dabei ist ein RIS-Referenzserver notwendig, der die Anforderungen eines Clients an einen RIS-Server weiterleitet, sofern er überprüft hat, dass es ein Computerkontoobjekt des Clients im Active Directory gibt. Im Active Directory findet er die Information, welcher RIS-Server den Client bedienen soll.

[1] Zitiert aus dem Buch [GoWe00b]. Leicht verkürzt.
[2] Diese Abbildung stammt aus der Präsentation von Microsoft [ScSc]. Modifiziert.

Beim Einschalten von PXE-fähigen Computer werden die Anforderungen von einer IP-Adresse und der IP-Adresse des RIS-Servers mit die MAC-Adresse des Computers mittels DHCP gesendet. Die MAC-Adresse, die auch während der Vorbereitung des Clients mit dem erstellten Computerkontoobjekt gespeichert wurde, kann den Client im Active Directory identifizieren. Der identifizierte Client empfängt vom DHCP-Server eine IP-Adresse und die IP-Adresse des RIS-Servers, der den Client bedient. Der Client kann nun den ersten Kontakt mit RIS-Server annehmen. Der Client sendet eine Anforderung zum Herunterladen einer Startimagedatei Startrom.com. Dann wird mittels TFTP OSChooser heruntergeladen und der Clientinstallations-Assistent steht zur Verfügung.

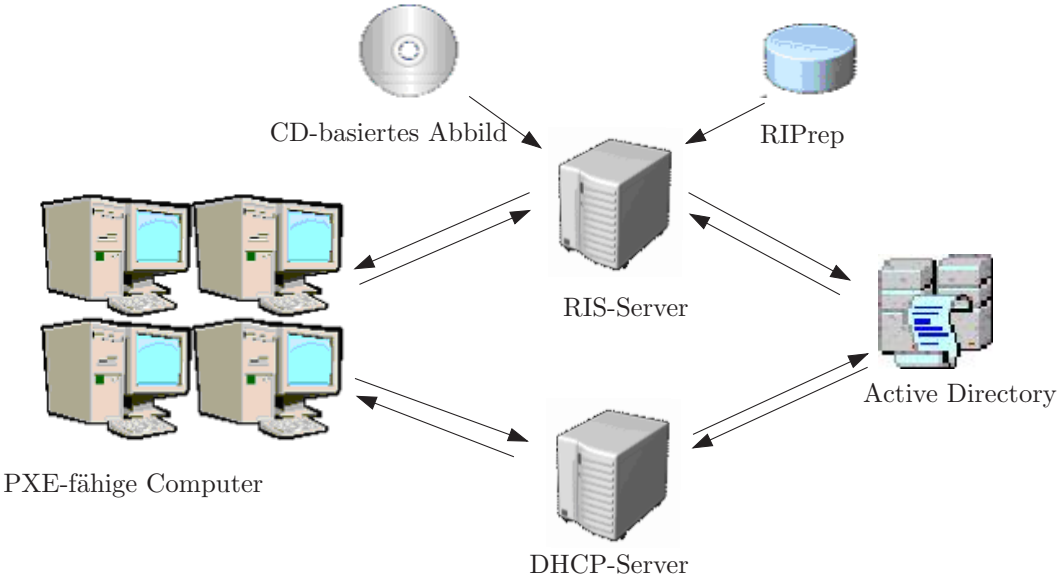


Abbildung 2: Installationsvorgang von RIS

2.2.2 ADS (Automated Deployment Services)

Im Vergleich zu RIS ist ADS eher für Deployment von Windows Server gedacht. Er beinhaltet ein Tool zur Verwaltung von Images und eine noch sichere und remote-fähige Infrastruktur für eine schnelle Verteilung von Windows Server. ADS bietet auch ein sicheres und zuverlässiges Script Execution Framework, damit die scriptbasierte Verwaltung auf 1000 Server genauso einfach ist, wie vorher auf einem einzigen Server. Der unterstützt auch nur Windows 2000 Server und Windows 2003 Server.

In der Abbildung 3 sieht man einen Überblick von ADS-Struktur.[3] Die Komponenten von ADS sind eine Menge von Diensten: Controller Service, Network Boot Service und Image Distribution Service, Volume Imaging Tools und eine Menge von Agenten.

Der Controller Service ist das Herz von ADS und stellt die Informationen für Konfiguration anderen ADS-Diensten bereit. Er koordiniert die administrative Tätigkeit während des Deployments und der Administration, bietet sichere Kommunikation zwischen Agenten, wenn sie Task Sequence ausführen. Er verwendet einen SQL-Server, um alle Daten von Computern und Konfiguration und die Log-Information für alle Tasks, die auf den Computern laufen, zu speichern.

Task Sequence ist eine wichtige Funktionalität von ADS. Task Sequences werden mit Task Sequence Editor erstellt, der eine Vorlage bereitstellt. Alle Tätigkeiten, die bei Deployment

[3]Die Abbildung ist aus der ADS-Seite von Microsoft [Micr03]

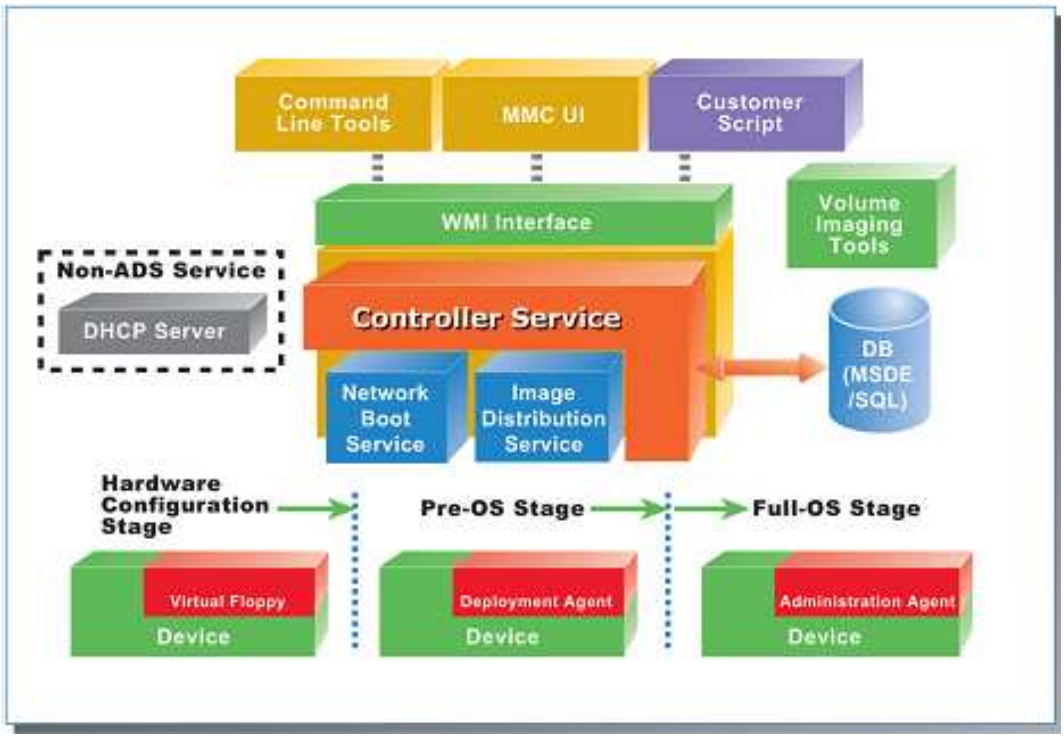


Abbildung 3: Struktur von ADS

und Administration ausgeführt werden, werden mit Task Sequences in XML-Format beschrieben, die dann auf einem oder mehreren Clientcomputer laufen. Der ADS Controller notiert alle ausgeführten Befehle und deren Ausgaben und speichert diese in der SQL-Datenbank.

Network Boot Service verwendet DHCP, um den ADS-Bootbefehl an den Computern zu geben. ADS arbeitet nicht mehr mit Active Directory. Falls ein PXE-fähiges Computer eingeschaltet wird, kann er wegen dieses Dienstes sofort mit ADS kommunizieren. Der Dienst bietet einen ADS Deployment Agent Builder an, der die Hardwareausstattung des Computers dem Controller Service mitteilt, und einen der Hardwareausstattung angepassten Deployment Agent erzeugt und den Agenten dem Computer weitergibt, auf dem der Agent im lokalen Speicher ausgeführt wird. Um den Agenten auf dem Computer herunterzuladen, verfügt der Network Boot Service über die Möglichkeit, ein MS-DOS-basiertes Virtual Floppy Image herunterzuladen. Virtual Floppy kann Hardwarekonfiguration wie Konfiguration von RAID Controllern und Updates durchführen.

Der Image Distribution Service verwaltet Images des Betriebssystems und ermöglicht dem Administrator, die Images zu verteilen, die von Volume Imaging Tools erzeugt werden. Bei der Verteilung wird von dem Controller Service eine sichere Verbindung, die Protokolle wie SSL (Secure Sockets Layer) verwendet, zwischen den Computern hergestellt. Image Distribution Service kann Images von mehreren Computern über unicast-fähigen und multicastfähigen Netzwerken verteilen. Dies ermöglicht eine schnellere Verteilung im Vergleich zu RIS.

Mittels ADS Imaging Tools kann man Images erstellen, verschlüsselen, editieren und entfernen. Images sind hier die Abbilder eines konfigurierten Betriebssystems, ähnlich wie RIPrep-Abbilder von RIS, wobei RIS nur das NTFS Format unterstützt, aber ADS NTFS und FAT unterstützt. Mittels dem Tool Imgdeploy werden Images erstellt, komprimiert und verschlüsselt. Die erstellten Images können mit dem Tool Imgmount editiert werden. Mit dem Tool Adsimage kann man alle verfügbaren Images auflisten, Images der Liste hinzufügen sowie entfernen.

ADS verfügen über zwei Agenten: der Deployment Agent und der Administration Agent. Der Deployment Agent ist eine Miniversion von Windows Server, die aus dem RAM von Clientcomputer geladen werden kann und die Operationen wie Diskpartitionierung und Herunterladen von Images ausführen. Der Administration Agent wird nach dem Deployment eines Betriebssystems auf dem Clientcomputer verwendet, um Post-Deployment-Operationen auszuführen. Beispielsweise führt er die lokalen Anwendungen oder die Anwendungen, die vom Netzwerk geladen werden können, aus.

Zur remote Installation muss man zuerst ein Master-PC mit dem Betriebssystem installieren und konfigurieren, auf dem die Images erstellt werden. Ein Image kann je nach Bedarf mittels ADS Imaging Tools editiert werden. Dann werden Task Sequences erstellt. Die PXE-fähigen Computer können vom Network Boot Service den Deployment-Agent herunterladen, der die Imageverteilung ausführt. Mittels des Administration-Agents werden die Tätigkeiten nach dem Deployment ausgeführt.

In der Tabelle 1 findet man einen Vergleich zwischen RIS und ADS:

	RIS	ADS
Verteilung von Betriebssystem	Windows 2000 (Professional und Server) Windows Server 2003 (alle 32-Bit Versionen) Windows Server 2003 (64-Bit Version) Windows XP (ausser Home Edition)	Windows 2000 Server Windows Server 2003 (alle 32-Bit Versionen)
Plattform	RIS läuft auf Windows 2000 Server und Windows Server 2003	ADS läuft nur auf Windows Server 2003
Methode von Deployment	Script-basiert und Image-basiert	Image-basiert
Hardwareanforderung des Clients	Mittels CD-basierter Abbilder ist die Gleichheit der Hardware nicht erforderlich.	Hardware von Clientcomputern müssen gleich sein
Multicast	Nicht unterstützt	unterstützt
Format von Image	NTFS	NTFS, FAT
Mehrere Partitionen	Nicht unterstützt	unterstützt
Active Directory	erforderlich	Nicht nötig

Tabelle 1: Vergleich von RIS und ADS

2.2.3 FAI (Fully Automatic Installation)

FAI ist ein sehr verbreitetes Tool zur automatisierten Installation von Debian GNU/Linux auf einem oder mehreren PCs. Die automatische Installation von Solaris wird auch unterstützt. Mittels FAI wird ein Betriebssystem automatisch installiert und konfiguriert ohne zu beaufsichtigen. FAI verwendet Debian-Distribution und eine Kollektion von den Shell- und Perlscripts für den Installationsprozess. Konfigurationsdateien können mittels cfengine, Shell- oder Perlscript modifiziert werden.

Normalerweise befinden sich drei Debian-Distributionen im dists Verzeichnis. Dies sind die stable, die testing und die unstable Distribution. Manchmal gibt es auch eine frozen Distribution. Jede Distribution ist ein symbolischer Link in das entsprechende Verzeichnis mit einem Kodenamen im dists Verzeichnis.

Vor der Installation wird ein Debian-Abbild mittels des Scripts mkdebmirror erstellt. Ein partiales Abbild für Debian 3.0 ohne Ressourcenpaketen benötigt etwa 9.0 GB Speicherplatz, daher wird normalerweise mittels NFS auf dieses Abbild zugegriffen. Man kann auch mittels HTTP darauf zugreifen, dazu muss man eine Web-Server Software installieren und einen Symlink zum lokalen Verzeichnis, wo das Debian-Abbild liegt, erstellen.

Ein FAI-Server muss auch verfügbar sein, auf dem FAI installiert wurde. FAI kann mit dpkg installiert werden. Bei der Remote Installation werden Clientcomputer über Netzwerk gebootet. Dann muss man alle Dateisysteme über das lokale Netzwerk per NFS mounten. Beim Setup von FAI wird ein mount-fähiges Verzeichnis nfsroot erstellt, das sich im FAI-Server befindet und ein komplettes Filesystem für alle Clients während der Installation ist. Für ein Netzwerkbooten mit PXE wird ein PXE Linux-Bootloader und eine spezielle Version von TFTP-Dämon benötigt, die man im Debian-Paket tftpd-hpa finden kann. Man kann auch mittels BootP Clientcomputer booten. Hier muss man einen symbolischen Link zum Image von Kernel mit tlink erstellen.

Beim Einschalten werden Clientcomputer gebootet. FAI führt mehrere Aktionen beim Booten von Clients, die in FAI_ACTION definiert wurden, aus. Man kann mittels hooks weitere Funktionen zu der Installation addieren oder einige default Tasks von FAI entfernen. Damit kann man den ganzen Installationsprozess nach eigenen Wünschen anpassen. Zuerst wird das Script rcs_fai automatisch ausgeführt. Damit wird Setup von FAI gestartet. Nach dem Setup wird das Script fai-class zu Klassendefinition für alle zu installierenden Clients ausgeführt. Klassen werden bei der Auswahl von Konfigurationsdateien benutzt. Eine Konfigurationsdatei legt fest, wie die Festplatte partitioniert werden und wo die Dateisysteme erstellt werden können. Danach werden Debian und andere benötigte Softwarepakete auf den neuen Dateisysteme mit dem Befehl install-packeges installiert. Nach der Installation kann man auch die default Konfiguration unterbrechen und selbst definierte Konfiguration mit Script ausführen. Am Ende des Installationsprozess werden Log-Dateien automatisch ins Verzeichnis /var/log/fai/$HOSTNAME/install/ auf jedem installierten Client und zum Konto auf FAI-Server geschrieben. Während der Remote Installation kann man alle Clients mit faimond überwachen.

Man kann auch Sun Solaris Betriebssystem unter der Verwendung von FAI mit der Kombination von jumpstart installieren.

2.3 Verteilung von Software

Viele Anwendungen wie MS Office und Netscape sind allgemein benutzt, daher ist die Verteilung von Anwendungen notwendig. Die grundlegende Funktion von Softwareverteilung ist die automatische Installation von Software. Falls möglich ist, kann man noch festlegen, welche Software welchem Benutzer zur Verfügung gestellt werden und wann die Installation ausgeführt werden kann. Außerdem können Updates auch rechtzeitig verteilt installiert werden.

Aber selbst für einen erfahrenen Benutzer der Aufwand zu groß ist, Programme aus den Quelltexten zu installieren, zu verteilen und zu warten. Viele Benutzer verfügen auch nicht über genügende Zeit und Kenntnisse davon. Daher werden heutzutage die Programme in kompilierter und leicht installierbarer Form - Paket zusammengefasst. Paket ist die Sammlung von Dateien, die für die automatisierte Installation einer Software notwendig sind, z.B. Programme, Bibliotheken und zusätzliche Steuerinformationen für das Paketsystem. Normalerweise sind mehrere Dateien komprimiert zusammengepackt. Die meisten Installationspakete lassen sich vom Herstellern auf CDs oder vom Internet bekommen.

Im Folgenden werden die Anforderungen an Paketmanagement genannt:

- Ein Tool für das Paketmanagement soll garantieren, dass man ein neues Paket installiert, ohne die funktionierenden Programme zu beschädigen. Falls bei der Installation ein Paket ein anderes Paket benötigt, soll das benötigte Programm automatisch mitinstalliert werden. Ansonsten soll das Löschen eines vorhandenen Pakets verhindert werden, falls das Paket ein bereits installiertes Paket unbedingt benötigt.

- Tools für das Paketmanagement sollen update-fähig und upgrade-fähig sein, das heißt, bei der Installation eines Updates oder Upgrades soll erkannt werden, welche Dateien von dem Programm aktualisiert werden sollen und welche nicht. Ein Update oder Upgrade soll möglichst einfach sein, ohne das System neu zu starten.

- Die Tools sollen auch rücksetzbar sein. Bei einer erfolglosen Installation soll ermöglicht werden, dass bei der Installation gelöschte oder überschriebene Dateien auf den Anfangzustand (vor der Installation) zurückgesetzt werden können.

- Bei der Deinstallation sollen alle nicht mehr benötigten Dateien und Ressourcen entfernt werden.

Bei der Softwareverteilung werden Softwarepakete in einem Verteilungspunkt im Netzwerk (ein File-System) gelegt und wird die Verfügbarkeit der zu installierenden Softwarepakete sichergestellt. Verteilungstools werden dabei helfen, Software automatisch auf mehreren Computern zu installieren.

Im Folgenden werden das Paketmanagement und die Softwareverteilungstechnologien sowohl von der Windows-Welt als auch von der Linux-Welt vorgestellt.

2.3.1 Softwareverteilung von Windows

Für Softwareverteilung ist vor allem wichtig, Software bereitzustellen. Der Windows Installer Service ist eine Komponente des Windowssystems und eine clientseitige Technologie der Softwarebereitstellung, mit der man die bereitgestellte Software installieren kann. Es ist im Windows 2000 und höhreren Versionen integriert und mittels Servicepack auch für Windows 95, Windows 98 und Windows NT 4.0 verfügbar.

Der Windows Installer Service besteht aus drei Teilen: Komponenten, Features und Produkte. Im Gegensatz zur anderen Installationstechnologien verwaltet der Windows Installer Service nicht direkt die Dateien und Ressourcen, sondern Komponenten. Eine Komponente (Windows Installer component) ist der kleinste Bestandteil von den drei Teilen und ist eine Sammlung von Registry Keys und allen für die Installation und zum Laufen des Programms benötigten Ressourcen, die nicht trennbar sind. Falls eine Komponente entfernt wird, werden alle Ressourcen und Registry Key auch entfernt. Keine zwei Komponenten haben dieselbe Ressource. Eine Komponente ist global eindeutig, das heißt, dass eine Komponente immer dieselbe Menge von Ressourcen besitzt, egal von welcher Software sie verwendet wird. Jeder Komponente wird eine GUID (Globally Unique Identifier) zugeordnet. Die Verwaltung auf der Komponentenebene ist einfacher und übersichtlicher als die Verwaltung direkt auf der Ressourcenebene.

Das Feature (Windows Installer Feature) ist der einzelne Teil von einer Anwendung, den ein Benutzer nach eigenem Wunsch installieren kann. Ein Feature ist eine Menge von Komponenten. Ein Feature kann mehrere Features beinhalten, wodurch die Anwendung hierarchisch organisiert werden kann. Beispielsweise enthält Microsoft Office das Feature Microsoft Word, das weitere Subfeatures, wie Proofing Tools enthält. Falls ein Feature zur Installieren gewählt wird, werden alle Komponenten des Features installiert. Weil Windows Installer Service auf

der Komponentenebene Pakete verwaltet, können zwei Features dieselbe Komponente haben, ohne das Laufen des Programms zu beschädigen.

Das Produkt (Windows Installer Product) präsentiert ein Produkt von Microsoft, wie MS Office und enthält mehrere Features. Jedes Produkt wird im Windows Installer Service als ein einziges Paket (*.msi) dargestellt. Jedes Produkt wird mittels einer ID (Product Code) identifiziert. Dies vereinfacht die Erkennung, welches laufende Programm welche Datei noch benutzt. Früher wurde für jede geteilte Datei ein Shared Reference Count in dem Systemregistry erstellt, damit das System erkennen kann, ob diese Datei noch von einem Programm benutzt wird. Falls die Datei gleichzeitig von drei Programmen benutzt wird, ist dieser Count für die Datei im Systemregistry gleich 3. Nun kann der Windows Installer Service das Product ID benutzen und eine Liste erstellen, in der die Abbildung von einer benutzten Komponente zu dem Produkt steht, das nun die Komponente benutzt, wobei die Komponente und das Produkt jeweils mittels GUID der Komponente und product ID des Produkts kenngezeichnet werden. Dies ist viel besser als rein eine Integerzahl, damit das System noch erkennt, auf welchem Produkt die Komponente läuft. Beispielsweise erkennt beim Update einer Software das System sofort, dass die Komponente, die aktualisiert werden soll, noch von einem anderen Programm verwendet wird, und kann das System genau angeben, von welchem Produkt die Komponente verwendet wird.

Das Windows Installer-Paket (*.msi) enthält expliziete Anweisungen zum Installieren und Entfernen spezieller Anwendungen. Das Format von einem Paket ist eine relationale Datenbank mit allen notwendigen Informationen zur Beschreibung der Installation einer Anwendung, z.B. Anweisungen zum Installieren einer Anwendung, wenn eine frühere Version bereits installiert wurde. In einem MSI-Paket sind Standardzusammenfassungen enthalten: Tabellen mit Beschreibung der Funktionen, die eine Anwendung ausmachen, Zeiger auf die zu installierenden Quelledateien und Zeiger, wohin die Anwendung auf dem Computer installiert werden soll. Bei der Installation öffnet der Windows Installer Service das Paket und bestimmt gemäß der Informationen in dem Paket alle auszuführenden Installationsoperationen. Es existieren einige Programme für das systemeigene Verfassen eines Windows Installer-Pakets: das Tool VERITAS WinINSTALL LE (limited Edition) und InstallShield. [4]

Wegen der hierarchischen Struktur eines Paketes wird ein einwandfreies Entfernen einer Anwendung garantiert. Beispielsweise wird eine Komponente, die von anderen vorhandenen Anwendungen benutzt wird, nicht entfernt. Außerdem kann der Windows Installer Service beim Starten einer Anwendung schnell die Existenz von Schlüsseldateien überprüfen, die für das Ausführen der Anwendung benötigt werden. Er kann auch die Anwendung reparieren, falls eine Komponente entfernt oder geschädigt wurde.

Softwareinstallation und -wartung ist eine Funktion von IntelliMirror, eine Technologie von Windows 2000 Server, und ermöglicht Benutzern, die Installation, Konfiguration, Reparatur und Deinstallation von Software zentral zu verwalten. IntelliMirror hilft den Benutzern bei der Verwaltung von Benutzer- und Computerinformationen, Einstellungen und Anwendungen. Es verwendet Active Directory und Gruppenrichtlinien. Gruppenrichtlinien können auf Standorte, Domänen sowie Organisationseinheiten angewendet werden, um die Fähigkeit von Gruppen, Computern und Benutzern zu definieren, die sich innerhalb des Standorts, der Domäne oder der Organisationseinheit befinden. Mit der Funktion Softwareinstallation und -wartung kann man Gruppenrichtlinieneinstellung definieren, die festlegen, welche Anwendungen die Benutzer verwenden können, unabhängig davon, auf welchem Computer sie arbeiten.

Nach der Erstellung eines Softwarepaketes und der Anpassung des Paketes werden die Software und deren Installer-Pakete auf den Softwareverteilungspunkt kopiert, der unter der Verwendung von Distributed File System (DFS) sowie anderen Windows 2000 Serverfunktionen

[4] Weitere Informationen stehen unter http://windows.microsoft.com/windows2000/reskit/webresources.

erstellt wird. Die Berechtigungen für das Verzeichnis im Verteilungspunkt muss so festgelegt werden, dass nur Administratoren Lese- und Schreibberechtigungen haben und Benutzer das Verzeichnis nur lesen können. Danach kann man die Gruppenrichtlinien und die Softwareinstallationskonsole verwenden, um Software zu verteilen. Man kann einem Computer oder einem Benutzer Software zuweisen oder einem Benutzer Software veröffentlichen. An einen Computer zugewiesene Software wird beim nächsten Start des Computers automatisch installiert. Bei der an einen Benutzer zugewiesenen Software wird bei der Anmeldung des Benutzers die Verknüpfung für die Anwendung auf dem Desktop des Benutzers oder im Menü Start angezeigt und die Registrierung wird mit den Informationen zu der Anwendung, einschließlich dem Pfad des Anwendungspaketes und dem Pfad der Quelldateien für die Installation aktualisiert. Wenn der Benutzer die Verknüpfung zum ersten Mal auswählt, wird mit der Hilfe dieser Informationen die Anfrage für dieses Softwareprogramm an den Verteilungspunkt gesendet. Der Windows Installer wird gestartet und installiert das angefragte Installer-Paket. Damit wird die Software erst installiert. Die zugewiesene Software kann nicht von Benutzern gelöscht werden. Die veröffentlichte Software wird nur in der Systemsteuerung unter Software angezeigt. Wenn eine Anwendung veröffentlicht wird, werden auf dem Desktop des Benutzers keine Shortcuts angezeigt, und es werden keine lokalen Registrierungseinträge vorgenommen. Das heißt, die Anwendung ist auf dem Computer des Benutzers nicht vorhanden. Die für veröffentlichte Anwendungen erforderlichen Informationen werden im Gruppenrichtlinienobjekt gespeichert. Benutzer können die Software installieren, falls sie die wirklich brauchen. Um eine veröffentliche Anwendung zu installieren, können Benutzer die Option Software der Systemsteuerung verwenden, die eine Liste aller veröffentlichten und verfügbaren Anwendungen enthält.

2.3.2 Softwareverteilung von Debian GNU/Linux

Debian GNU/Linux Paketmanagement ist modular aufgebaut. Es besteht aus den drei Programmen: dpkg-deb, dpkg und dselect, die jeweils Operationen verschiedener Abstraktions-Ebenen und unterschiedlicher Komplexität bereitstellen. Die von den Programmen benötigten Informationen über den Zustand installierter und verfügbarer Pakete werden in einer Statusdatenbank gespeichert. dpkg-deb steht auf der untersten Ebene und stellt den Zugriff auf das Paketmanagementtool bereit. Es kann Informationen über das Paket anzeigen und Daten aus dem Paket extrahieren. Mittels dpkg-deb kann man Debian-Pakete erstellen. dselect ist die Benutzerschnittstelle, durch die man die zu löschenden und zu installierenden Pakete auswählen kann. dpkg führt dann die Installation, das Entfernen und das Update aus. dselect und dpkg können auf die Statusdatenbank zugreifen, in der der aktuelle Zustand (z.B. installed, Not Installed) und die Aktion (z.B. Install, Remove) für jedes Paketes gespeichert wird. Mittels dselect kann man die Aktion eines ausgewählten Pakets modifizieren. Falls ein Paket mittels dselect zum Installieren ausgewählt wird, setzt dselect die Aktion des Paketes in der Datenbank auf 'Install'. Dann wird dpkg aufgerufen und die Aktion wird durchgeführt. Nach der Installation wird der Zustand des Paketes auf Installed gesetzt.

Die interne Struktur eines Pakets kennt nur dpkg-deb. Daher werden alle Manipulationen an Paketen von dpkg-deb vorgenommen, einschließlich des Aufbaus eines Paketes. Der Träger der verschiedenen Komponenten eines Debian-Paketes ist ein Archiv im ar(1)-Format. Diese Debian-Archivdatei enthält ausführbare Dateien, Bibliotheken, Abhängigkeiten von Paketen und Dokumentationen.

Damit eine Software korrekt funktionieren kann, muss man vorher alle von dieser Software abhängigen Softwarepakete richtig installiert haben. Man soll auch vermeiden, die benötigten Pakete unabsichtlich zu entfernen. Die Installation und Deinstallation der Software durch das Paketsystem dpkg verwendet 'Abhängigkeiten', welche sorgfältig vom Paketbetreuer bestimmt wurden. Diese Abhängigkeiten legen fest, wie Programm A abhängig von der Existenz

von Programm B läuft. Es sind unterschiedliche Abhängigkeiten z.B. A hängt von B ab, A schlägt B vor, A ersetzt B u.s.w. vorhanden. Diese Abhängigkeiten sind in der control-Datei, die jedem Paket zugeordnet ist, enthalten.

Bei dem Erstellen eines Paketes wird der Quellcode geholt und unter einem Verzeichnis mit Name (Programmname_Versionnummer-DebianRevisionnummer.deb) kopiert. Mittels des Befehls dh_make wird ein Unterverzeichinis erzeugt, das die zum Erstellen eines Paketes benötigten Vorlagen der Dateien wie rules, copyright, changelog, control, dirs und menu enthält. Nach der Angabe von dem Befehl dpkg-buildpackage in der Shell bekommet man das Binary-Paket '*.deb', die Änderung '*.diff.gz', das Original '*.orig.tar.gz' und die Beschreibung der Quelle '*.dsc.'.

Debian wird auf der ganzen Welt mittels Spiegel-Server [5] verteilt, um die Benutzer mit besserem Zugriff auf das Archiv von Debian zu versorgen und die Server von Debian zu entlasten. Die Software von Debian ist in einem der vielen Debian-Verzeichnisbäume auf jedem Spiegel-Server durch FTP oder HTTP verfügbar. Eines der wichtigsten Verzeichnisse ist dists/. Dieses Verzeichnis enthält die 'Distributions', wo sich alle aktuell verfügbaren Debianpakete befinden. Es gibt drei Distributions: stable, testing, unstable. Die Pakete unter dem Verzeichnis stable/main/ bilden die aktuelle Ausgabe des Debiansystems und sind frei verfügbar und verteilbar. Die Pakete, die zur unstable-Distribution gehören, werden solange im Verzeichnis unstable aufbewahrt, bis sie getestet werden und ins Verzeichnis testing/ verschoben werden. Sie müssen weniger Fehler haben als die Version in unstable-Distribution.

Der Administrator kann gewünschte Pakete von der Distribution auf einem Master-Computer herunterladen und installieren. Die Verteilung der Software auf weitere Computer erfolgt durch Script. Man kann z.B. ein Script schreiben, das zu einem vordefinierten Zeitpunkt automatisch ausgeführt wird, um die Änderungen, die auf dem Master-Computer geschehen sind, auf alle anderen Computer zu übertragen.

2.4 Verteilungsmechanismen des Rechenzentrums

Das Rechenzentrum der Universität Karlsruhe verfügt über einen eigenen Verteilungsmechanismus, der die Unterstützungen von DHCP, PXE, TFTP benötigt. In den Poolräumen sind normalerweise zwei Betriebssysteme auf jedem PC vorhanden: ein Windows Betriebssystem und ein Linux-Betriebssystem. Die Remote-Installation von den Betriebsystemen ist mittels UNIX-Server realisiert.

Der Verteilungsmechanismus des Rechenzentrums basiert auf der Vorraussetzung, dass die Hardwareausstattung jedes Rechners fast gleich ist. Zur Bereitstellung von neuen Computern werden die MAC-Adresse aller Computer in einer Datenbank eingetragen, wenn die Computer an einem lokalen Netzwerk angeschlossen sind. Beim Einschalten eines Computers sendet er eine PXE-Anforderung mit seiner MAC-Adresse, die von einem DHCP-Server behandelt. Der Server überprüft, ob die MAC-Adresse schon in der Datenbank vorhanden ist. Falls ja, wird eine DHCP-Antwort, sogenante FOUND-Frame mit der IP-Adresse eines TFTP-Servers, verteilter IP-Adresse des Computers und dem Pfad bzw. dem Namen der zum Booten notwendigen Bootdatei zurückgesendet. In diesem FOUND-Frame findet der Computer die IP-Adresse des TFTP-Servers, sendet eine Anforderung mit dem Namen und dem Pfad der gewünschten Datei und holt die Bootdatei vom TFTP-Server, die mitteilt, wo die Konfigurationsdatei 'bootcfg.py' liegt. Dann holt der Computer die Konfigurationsdatei von dem Server, dessen IP-Adresse in der Boot-Datei steht. Die Konfigurationsdatei wird dann auf dem Client ausgeführt und auf dem Bildschirm des Computers sieht man die verschiedenen Startmodi: Windows, Linux, Admin. Der Benutzer kann nun auswählen, in welchem Modus der Computer starten soll.

[5] Spiegelserver sind solche Computer, die dieselben Daten enthalten, um Datenverlust zu vermeiden.

Falls man ein Betriebssystem auswählt, erkennt die Datenbank, in der die Informationen aller Poolrechner gespeichert wurden, ob auf dem gebooteten Computer schon ein Betriebssystem installiert wurde. Falls das ausgewählte Betriebssystem auf dem Computer installiert wurde, wird der Computer wie immer weiter gebootet. Falls noch kein Betriebsystem vorhanden ist, wird automatisch ein minimales Linux-System vom Netzwerk auf dem Computer installiert, das die Partitionierung vom Festplatten und Anpassungen an Hardware ausführt. Danach wird das Image vom ausgewählten Betriebssystem mittels TFTP auf dem lokalen Computer übertragen.

Falls man das Linux-System auswählt, stehen zwei Methoden zur Verfügung. Die erste Methode ist Image-basiert. Eine Image-Datei zur Installation wird vom Server geholt, entpackt sich und führt sich aus. Aber Installationen mit Image benötigen genügend Cache-Speicherplatz. Falls der Computer nicht ausreichend Speicher hat, werden alle zur Installation benötigten Dateien vom Netz gezogen und lokal kopiert.

Falls man das Windows-System auswählt, stehen auch zwei Methoden zur Verfügung. Man kann ein Windowssystem mit CD-basierter Image oder RIPrep installieren. Gemäß vordefinierter Konfiguration werden auf unterschiedlichen Clients verschiedene Images benutzt.

Der Vorteil des Verteilungsmechanismus des Rechenzentrums ist, dass man Netzwerkbooten benutzt, ohne auf die Festplatte eines Computers zuzugreifen. Damit kann man überprüfen, ob die Festplatte des Computers in Ordnung ist.

Aber falls viele Computer gleichzeitig vom Netzwerk Image ziehen, kann dies einen großen Datenfluss und eine Serverüberlastung verursachen. Daher werden die Clients bei der ersten Installation hierarchisch verwaltet, dass nur eine Gruppe von Computern direkt vom Server Image ziehen kann und das Image dann an ihre Submenge von Computern weiterleitet.

Um die Verzeichnisse leicht zu verwalten, bietet das Rechenzentrum die kleine Baumschule - der Softwareverteilungsmechanismus des Rechenzentrums an. Die Baumschule bietet vorkonfigurierte, an die Uni-Umgebung angepasste und geupdatete Programme an. Das Rechenzentrum kann durch die gleiche Umgebung der Baumschule bei Problemen einen gewissen Support anbieten. Die kleine Baumschule sieht so aus, dass die zum Betrieb des Rechners unbedingt notwendigen Dateien auf der lokalen Platte liegen müssen (...\machine), die oft gebrauchten Dateien, die im lokalen Netz-Segment vorhanden sind (...\segment) und die wenig notwendigen großen Dateien auf dem zentralen Fileserver gelegt werden können (...\rzserv). Die übergeordneten Verzeichnisse unterscheiden sich noch zwischen betriebssystemspezifischen (\usr\...) und -unabhängigen Dateien (\usr\common\...), so dass auch in gemischten Pools doppelte Datenhaltung vermieden wird.

3 Updatemechanismen

Um die schwachen Punkten einer Software zu beseitigen, gegen neu vorkommende Viren das System bzw. Software zu sichern, oder neue Funktionalitäten einer Software zu ergänzen, stehen immer Updatepakete zur Verfügung. Aber die Probleme sind:

- Wie kann man rechtzeitig erfahren, dass ein Update vorhanden ist.
- Wie kann man feststellen, ob man dieses Update wirklich braucht.
- Wie kann man festlegen, ob nach dem Update das System bzw. die benutzte Software richtig läuft und alle Konfigurationen noch richtig bleiben.

Um solche Probleme zu lösen, braucht man Unterstützung von Tools, die einem System beim Update helfen können.

3.1 Updatemechanismen bei Windowssystem

3.1.1 MBSA (Microsoft Baseline Security Analyzer)

MBSA ist ein standalone Tool und läuft auf den Plattformen Windows XP, Windows 2000 und Windows Server 2003. Der kann über eine Auflistung von allen installierten Updates im System verfügen und die fehlenden Sicherheitsupdates und sicherheitsgefährende Konfigurationsfehler in dem Betriebssystem, IIS-Server (Internet Information Services), MS SQL-Server, MS Internet Explore und MS Office entdecken und berichten. Außerdem kann ein Administrator eine Liste von Updates erstellen, die ihm wichtig erscheinen, wobei MBSA nur die fehlenden Updates von dieser Liste scannt.

MBSA kann auch benutzt werden, um die Computer von einer Domäne oder eine Range von Computern während eines niedrigen Betriebszeitraums zu scannen. Dies ist mittels eines Schedulingtools realisierbar, damit man die Schedules für die MBSA-Operationen vordefinieren kann.

3.1.2 SUS (Software Update Services)

Software Update Services bietet die Möglichkeit, einfach und effektiv Updates, Patches und Hotfixes für MS Betriebssysteme ab Windows 2000 zu verwalten und zu verteilen. Der SUS-Server braucht die Unterstützung von Windows 2000 Server, Windows Server 2003, IIS und Port 80, um mit Clients zu kommunizieren. Ein SUS-Server kann entweder mit dem Windows Update Server oder mit einem hierarchisch höher gelegten SUS-Server synchronisiert werden. Vorteile von SUS-Server sind, dass erstens diese manuell oder nach einem vorher definierten Zeitplan mit dem MS Windows Update Server synchronisiert werden können, zweitens der Server für die Clients im eigenen Netzwerk so konfiguriert werden kann, dass sich diese Clients ständig über den SUS-Server aktualisieren können und drittens man mit der Hilfe von Gruppenrichtlinien bestimmen kann, welcher Client oder Server zu welcher Uhrzeit die Updates bekommen kann.

Manch wichtige heruntergeladene Updates werden nicht sofort verfügbar gemacht, um die Kompabilität zwischen dem neuen Update und dem laufenden System in einer kleinen Umgebung zu testen. Nachdem das geupdatete System den Test erfolgreich bestanden hat, wird das Update erst im Netzwerk verteilt.

SUS-Server können sich untereinander synchronisieren, dadurch wird das Netzwerk weniger belastet. Um dies zu realisieren, wird eine hierarchische Struktur aufgebaut. Beispielsweise kann jedes Institut einen eigenen SUS-Server haben, die Clients innerhalb dieses Instituts werden nur mit diesem SUS-Server sysnchronisiert. Der SUS-Server gleicht sich entweder direkt mit dem Windows Update Server oder mit einem SUS-Server, der hierarchisch höher gelegt ist.

Die nächste Version von SUS ist WUS (Windows Update Services), die am Ende des Jahres 2004 verfügbar ist. WUS verfügt über die Patches für SQL-Server und Exchange-Server. Unter der Benutzung von Active Directory kann der Administrator mittels WUS Updates an eine vordefinierte Gruppe verteilen. Außerdem stellt WUS Bandbreit-Management (bandwidthmanagement) zur Verfügung, damit bei Updates die Last vom Netzwerk berücksichtigt werden kann.

3.1.3 SMS 2003 (System Management Server 2003)

SMS wird für Softwareverteilung und Updatemanagement von einer großen Organisation benutzt. SMS enthält die Tools Security Update Inventory Installer, Distribute Software Updates Wizard Installer und Web Reports Add-In for Software Updates.

Mittels Security Update Inventory Installer, integriert mit der MBSA-Technologie wird ein Scan auf allen Clients ausgeführt und ein Inventar von allen anwendbaren und installierten Updates erstellt. Der neueste Katalog 'Security Update Bulletin Catalog' kann auch mit dem Tool heruntergeladen und über einen Verteilungspunkt, der in der Infrastruktur von SMS vorhanden ist, an alle Clients verteilt werden. Das Tool Distribution Software Updates Wizard überprüft den Updatezustand von Clients gemäß der in dem Inventar gespeicherten Informationen. Das Tool verfügt über noch eine Methode, um einen Review aller empfohlenen Updates auszuführen und die Updates zu autorisieren, damit nur die von dem Administrator erlaubten Updates verteilt werden können. Dann werden die autorisierten Updates und dazugehörenden Informationen heruntergeladen. Der Security Update Inventory Installer baut auf einem einzelnen Updatepaket oder einer Menge von Update-Paketen auf und SMS wird die Pakete auf den Clients verteilen. Dazu wird ein Software Update Installation Agent für Clients aufgestellt, der die vorher installierten Updates auswertet und nur die notwendigen Updates ausführt. Damit wird das Systemoverhead durch Elimination von Redundanz und unnötigen Updates stark reduziert.

Außerdem enthält SMS einen Netzwerkmonitor. Mit seiner Hilfe kann man das Netzwerk analysieren, Netzwerkpakete erfassen und die Informationen von Paketen lesen, welche Protokolle die Pakete jeweils benutzen und welche Computer die Dienste anbieten.

Das Tool Web Reports Add-In for Software Updates ist ein Webbericht-Tool, um den Benutzern über die Informationen, die von Software Update Inventory Tools gesammelt wurden, zu berichten.

Im Vergleich zu SUS ist SMS für große Unternehmer geeignet und SMS verfügt über eine gute Kontrolle darüber, welche Updates zu welcher Zeit an bestimmte Clients verteilt werden, ob die Verteilung einphasig oder mehrphasig ausgeführt wird, ob die Updates die richtige Signatur tragen und wie die Installation verläuft.

3.2 Updatemechanismen bei Unixsystem

Es ist immer aufwendig, das gewünschte Paket aus den vielen neu hochgeladenen Paketen zu identifizieren. Um dies besser handhaben zu können, sind einige spezialisierte Paketverwaltungstools verwendbar, wobei dpkg zur Manipulation der Paketdatei und APT (Advanced Packaging Tool) bzw. dselect zum Herunterladen der Paketdateien aus dem Debian-Archiv dienen.

Vor allem soll man wissen, wo man Pakete finden kann. APT benutzt eine Datei, die die Quellen, von denen man die Pakete beziehen kann, auflistet. Diese Datei heißt /etc/apt/sources.list. Die Einträge in dieser Datei sind von folgendem Format:
deb http://site.http.org/debian distribution sektion1 sektion2 sektion3
deb-src http://site.http.org/debian distribution sektion1 sektion2 sektion3
Wobei deb und deb-src den Typ des Archivs angeben. deb zeigt die Quelle der Binary-Dateien und deb-scr die Quelle der originalen Programm.

Das Paketsystem von Debian verfügt über eine Datenbank mit Informationen über installierte, nicht installierte und für eine Installation verfügbare Pakete. Um die Liste aller verfügbaren Pakete zu aktualisieren, benutzt man den Befehl apt-get update. Man sucht dann die neu

verfügbaren Pakete von der Paketliste /etc/apt/sources.list. Es ist empfehlenswert, diesen Befehl regelmäßig auszuführen, um das System auf dem neusten Stand zu halten.

Das Aktualisieren eines einzelnen Paketes von Debian ist einfach durch den Befehl apt-get upgrade realisiert. Falls man den Befehl mit der Option -u ausführt, sieht man die komplette Liste der Pakete, die aktualisiert werden sollen. Es ist wichtig, vor der Aktualisierung eines einzelnen Paketes die Liste aller verfügbaren Pakete zu aktualisieren. Mit der Hilfe von dselect kann man noch die zu installierenden Pakete auswählen. Dies ist besonders nützlich, falls bei der Installation eines Paketes die Installation anderer Pakete empfehlenswert ist.

Für die Updates bzw. Upgrades ist weiterhin wichtig, die Konfigurationsdatei während der Aktualisierung aufzubewahren, weil es möglich ist, nach der Aktualisierung das System ohne Modifizierungen von Konfigurationsdateien anzupassen. Dafür gibt es ein Mechanismus mit dem Namen conffiles. Die Konfigurationsdateien werden in den conffiles innerhalb von Debian-Paketsystem angegeben. Diese Dateien werden dann bei einer Paketaktualisierung nicht überschrieben werden. Um zu bestimmen, welche Pakete während der Aktualisierung bewahrt werden, führt man dpkg -status Paket aus.

3.3 Updatemechanismen des Rechenzentrums

Zum Update des Windows-Systems benutzt der Administrator des Rechenzentrums momentan den SUS. Zur Zeit werden zwei SUS-Server betrieben: einer (stud-sus-01.stud.uni-karlsruhe.de) für die Windows-Rechner in den Poolräumen und der zweite (rzms-sus1.rz.uni-karlsruhe.de) für die Windows-Rechner der Mitarbeiter und für die RZ-Server. Beide SUS-Server laufen parallel und holen sich ihre Updates von einem Microsoft Update Server. Damit wird die Last auf zwei Servern verteilt. Die Konfiguration der SUS-Server des Rechenzentrums wird in der Abbildung 4 gezeigt. In diesem Jahr wird die nächste Version von SUS - WUS (Windows Update Services) im Rechenzentrum eingesetzt und die zwei SUS-Server werden durch zwei WUS-Server ersetzt.

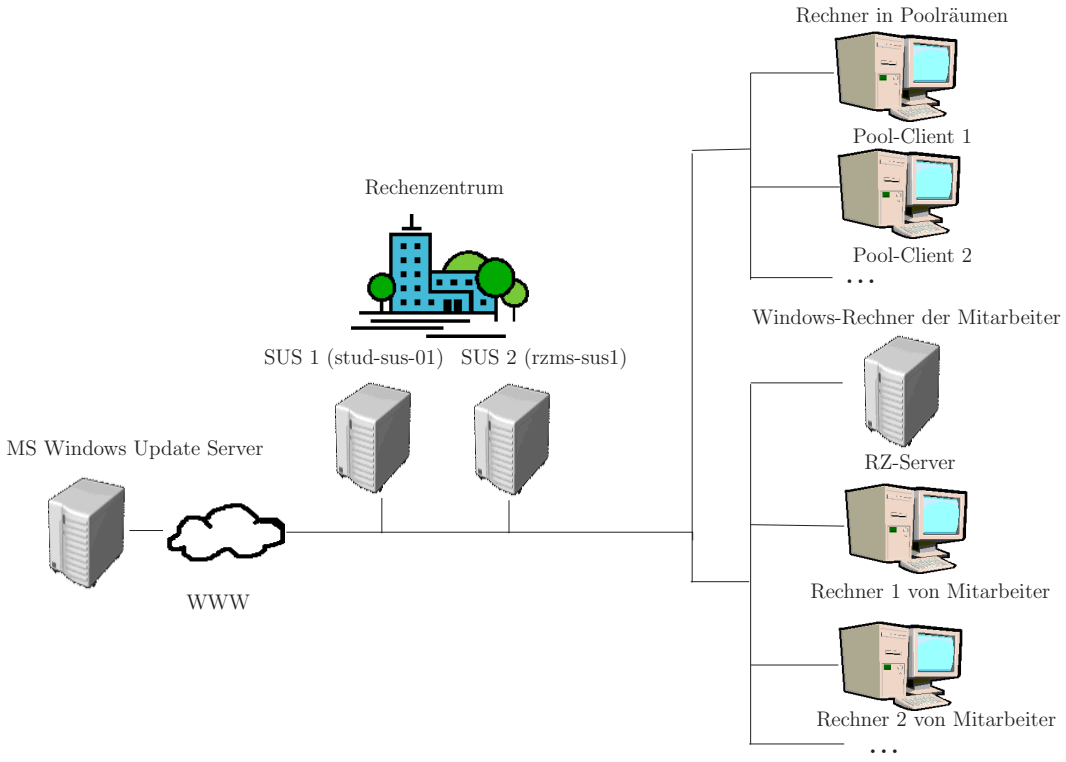


Abbildung 4: Konfiguration der SUS-Server des Rechenzentrums

Zum Update des Linix-Systems benutzt man das Programm autoop (Automatic Operator). Weil das Rechenzentrum den Verteilungsmechanismus 'die kleine Baumschule' hat, besitzt jeder Computer im Poolräumen dieselbe Verzeichnisstruktur und dieselben Standardprogramme, das heißt, alle Computer laufen in der gleichen Umgebung. Daher können die Unterschiede zwischen dem Master-Computer und den Poolraumrechnern erkannt werden. Autoop ruft jede Nacht ein Script auf, das alle Veränderungen an den Verzeichnissen von einem 'Master Server' in die auf dem Rechner lokal angelegten Verzeichnisse überträgt. In dem Installationsscript des Rechenzentrums sind folgende Programme vorhanden: autoop (Automatic Operator), rztrans (RZ-Remote-Printing), ssh (Secure Shell) und mailcap (Multimedia Steuerdatei).

4 Sicherheitsprobleme und die Lösungen

4.1 Sicherheitsanforderungen bei Softwareverteilung

Die Sicherheit bei der Softwareverteilung hat zwei Aspekte. Erstens muss garantiert werden, dass Sicherheitsupdates regelmäßig und rechtzeitig ausgeführt werden sollen, damit die Software auf dem aktuellen Zustand bleibt, um den möglichen Virusangriff oder den Missbrauch wegen der Lückenhaftigkeit von Software zu vermeiden. Wie man Updates rechtzeitig ausführt, wurde im 3. Kapitel behandelt. Worauf sich dieser Teil der Ausarbeitung bezieht, ist der zweite Aspekt: Die Softwareverteilung muss auch in einem sicheren Verfahren ausgeführt werden. In diesem Aspekt werden drei Punkte betrachtet:

- Bei der Betrachtung des Servers: Dem Server muss klar sein, welchem Client er die Software zuweist. Dabei muss der Server den Client authentifizieren. Die entsprechende Information wird normalerweise in der entsprechenden Datenbank eines Servers gespeichert.
- Bei der Betrachtung des Clients: Im Gegensatz dazu muss dem Client auch klar sein, dass die erhaltenen Pakete wirklich von dem angesprochenen Server sind und während der Übertragung nicht modifiziert wurden.
- Während der Übertragung über Netzwerk: Die übergetragenen Pakete sollen nicht von Dritten modifiziert werden.

Im Folgenden werden die Lösungen der Windows-Welt (Beispiel Windows 2000 Server) und Lösungen der Linux-Welt (Beispiel Bebian) jeweils gegeben.

4.2 Lösungen für die Sicherheitsprobleme

4.2.1 Lösungen von Windows

Die Informationen über alle Rechner in einem lokalen Netzwerk werden als Objekt in Active Directory gespeichert. Bei der Remote Installation wird der Computer, der die Dienstanforderung an den Server gestellt hat, vor allem authentifiziert dadurch, dass dieser eine Anforderung mit seiner MAC-Adresse sendet, die den Computer im Active Directory identifiziert. Im Gegensatz muss der Server für Active Directory autorisiert sein, bevor ein RIS-Server Anforderungen annehmen kann. Nur das Mitglied der Gruppe 'Organisations-Admins' kann einen RIS-Server oder einen DHCP-Server im Active Directory autorisieren. Dies garantiert dem Client, dass er mit einem richtigen Server kommuniziert.

Gruppenrichtlinien spielen bei einer sicheren Softwareverteilung eine wichtige Rolle. Man kann mit der in Gruppenrichtlinien integrierten Technologie der Softwareinstallation und -wartung

festlegen, welchem Computer oder Benutzer welche Software zugewiesen oder veröffentlicht werden sollen, wie bereits im vorherigen Kapitel erwähnt wurde.

Um die Übertragung von Softwarepaketen zu sichern, wird die Software bei Bedarf auch digital signiert. Signierte Software gewährleistet, dass Benutzer verifizieren können, woher die Software stammt und dass kein Dritter unerlaubte Änderungen an der Software vorgenommen hat. Vor der Freigabe der Software wird eine digitale Signatur erstellt. Es gibt momentan viele Technologien, um die Programmcode zu signieren, z.B. Microsoft Authenticode - Technologie. Anbei ist wichtig, den privaten Schlüssel, der zum Signieren von Code verwendet wird, besonders zu schützen.

Bei der Softwareverteilung innerhalb von einer Organisation kann man zum Schutz des Netzwerks vor Schaden verursachenden Programmen und Viren in der Internet Explorer - Konfiguration Sicherheitseinstellungen für Sicherheitszonen angeben. Zu diesen Zonen gehören die Internetzone, die lokale Intranetzone, die Zone für vertrauenswürdige Sites und die Zone für eingeschränkte Sites. Man kann dann Sicherheitseinstellungen angeben, die Benutzer daran hindern, nicht signierte Software von einer Sicherheitszone herunterzuladen und auszuführen. Man kann den Internet Explorer so konfigurieren, dass bestimmte Softwareanbieter als vertrauenswürdig betrachtet werden und von diesen Anbietern signierte Software automatisch ohne Benachrichtigung der Beutzer heruntergeladen werden können. Statt dessen kann man auch eine Gruppenrichtlinie für öffentliche Schlüssel konfigurieren, um die Codesignatur-Zertifizierungsstellen anzugeben, die innerhalb von seiner Organisation als vertrauenswürdig betrachtet werden.

4.2.2 Lösungen von Linux (Beispiel Debian)

Ähnlich wie bei Windows werden die Informationen aller Computer in einem lokalen Netzwerk in einer Datenbank gespeichert. Die zu installierenden Computer werden über Netzwerk gebootet und bekommen ein minimales Betriebssystem mittels TFTP, sofern der Server merkt, dass die MAC-Adresse des Computera mit der, die in der Datenbank liegt, übereinstimmt. Zugriffsrechte von Benutzern auf Dateien und Verzeichnisse sind vom Administrator vordefiniert. Daher wird der von einem normalen Benutzer verursachte Schaden sich nur auf einem eingeschränkten Bereich auswirken.

Debian verwaltet die hochgeladenen Pakete so, dass die sich zunächst unter http://incoming.debian.org befinden. Nachdem sie überprüft wurden, um sicher zu stellen, dass sie wirklich von einem Debian-Entwickler stammen, werden sie nach unstable/ verschoben. Ansonst werden sie in das DELAYED Unterverzeichnis verschoben.

Das Problem freier Software wie Debian ist, dass der Benutzer sicher sein muss, dass das gerade heruntergeladene Paket wirklich von den Debian-Entwicklern entwickelt wurde und zum aktuellen Release gehört. Debian verfügt momentan nicht über Standardwerkzeug für das Signieren von Paketen. Man kann das Script von Anthony Towns benutzen. Mit dem Script kann der Benutzer sicher sein, dass das heruntergeladene Paket genau das von Debian hergestellte Paket ist. Das Script verhindert, dass ein Mirror fast genau abbildet, das aber nicht ganz wie Debian ist, oder dass veraltete Versionen von instabilen Paketen mit bekannten Sicherheitslücken zur Verfügung gestellt werden. Den Quellcode findet man unter: http://www.debian.org/doc/manuals/securing-debian-howto/ch6.de.html.

Debian bietet noch Werkzeuge zur Fern-Überprüfung der Verwundbarkeit. Das meist benutzte Werkzeug heißt Nessus. Nessus besteht aus einem Client (Benutzerschnittstelle) und einem Server, der die programmierten Attacken startet. Nessus erkennt die Verwundbarkeit für die Systeme wie FTP-Server, Netzwerkanwendungen und WWW-Server.

Wegen Ressourcenmangel kann das Sicherheitsteam von Debian nicht alle Pakete aus Debian auf potentielle Sicherheitslücken analysieren. In der Tat kann ein Debian-Entwickler in einem Paket einen Trojaner verbreiten und es gibt keine Möglichkeit, dies nachzuprüfen. Jedoch können Bedian-Benutzer Vertrauen fassen, dass der stable Quellcode eine breite Prüfung hinter sich hat. Daher ist nicht zu empfehlen, ungetestete Software auf wichtigen Systemen zu installieren.

5 Zusammenfassung und Ausblick

In dieser Ausarbeitung wurden Remoteinstallation des Betriebssystems, Softwareverteilung, Updatemanagement sowie die Sicherheitsprobleme, die mit der Softwareverteilung entstehen, sowohl in der Windows-Welt als auch in der Linux-Welt gezeigt. Zur Remoteinstallation stehen folgende Software zur Verfügung: RIS, ADS und FAI. Während eine Standardfunktion Software-Installation und -Wartung zur Softwareverteilung im Windows-System existiert, kann man mit den von Benutzern geschriebenen Scripten Software im Linux-System verteilen. Dies zeigt einerseits die Flexibilität der Linux-Software, andererseits verliert man den Vorteil des Standards, dass man die Software leichter bedienen kann und es zu Problemen immer Standardlösungen gibt. Der SUS als Updatedienst ist mächtiger als der Updatedienst des Linux-Systems, wobei beispielsweise die fehlenden Updates des Windows-Systems automatisch gezeigt werden, aber man die benötigten Updates für Debian-Pakete noch von einer Liste /etc/apt/sources.list selbst herausfinden soll.

Die Softwareverteilung vereinfacht den Installationsprozess von mehreren Computern und die Administrationsaufgaben bzw. ermöglicht die zentrale Verwaltung von Software-Wartungen und -Updates. Sie zeigt einen grossen Entwicklungsraum in der Zukunft: Erstens wird die Softwareverteilung immer leichter ausführbar, sowohl für Administratoren als auch für Endbenutzer. Zweitens werden die Tools zur Softwareverteilung und zum Updatemanagement immer flexibler und auf mehreren Plattformen anwendbar. Drittens wird die Robustheit sich weiter verbessern, so dass der Zustand vom System trotz fehlergeschlagener Installation immer rücksetzbar sein kann. Der letzte aber auch der wichtigste Punkt ist die Sicherheit bei der Softwareverteilung. Beispielsweise muss man die Datenbank, die die Informationen von allen Computern im lokalen Netzwerk enthält, gut aufbewahren und beim Ausfall eine Ersatzmöglichkeit zur Hand haben. Der Server im Linux-System muss auch für Benutzer authentifiziert werden, damit der Benutzer sicher sein kann, dass er mit einem sicheren Server kommuniziert. Viele Hacker verwendeen TFTP-Server zur Verbreitung von Viren, weil TFTP-Server den Benutzer beim Zugriff nicht authentifiziert. Eine Gegenmaßnahme ist, dass nur Herunterladen vom TFTP-Server erlaubt wird. Dateien können nicht auf den TFTP-Server hochgeladen werden.

Literatur

[AHOP+a] C. Aslan, M. Hammer, T. Osterwald, L. Petrean und F. Stolz. SUS - Szenarien. Technischer Bericht, University of Karlsruhe, http://www.escde.net/downloads/public/docs/de/SUS-Szenarien.zip.

[AHOP+b] C. Aslan, M. Hammer, T. Osterwald, L. Petrean und F. Stolz. *SUS (Software Update Services) - Grundlagen.* University of Karlsruhe, http://www.escde.net/downloads/public/docs/de/SUS%20Grundlagen.zip.

[BaWe97] D. Barnes und T. Wenzel. Linux RPM Howto. http://www.linuxhaven.de/dlhp/HOWTO/DE-RPM-HOWTO.html#toc3, September 1997.

[compa] computerbase. BootP. http://www.computerbase.de/lexikon/BOOTP.

[compb] computerbase. DHCP. http://www.computerbase.de/lexikon/DHCP.

[compc] computerbase. PXE. http://www.computerbase.de/lexikon/PXE.

[compd] computerbase. Softwareverteilung. http://www.computerbase.de/lexikon/Softwareverteilung#.C3.9Cbersicht.

[compe] computerbase. TFTP. http://www.computerbase.de/lexikon/TFTP.

[Debia] Debian. Aktualisierung eines Debian-Systems. http://www.debian.org/doc/manuals/reference/ch-system.de.html#s-uptodate.

[Debib] Debian. APT How To. http://www.debian.org/doc/manuals/apt-howto/.

[Debic] Debian. Debian-Referenz - Das Debianpaketverwaltungssystem. http://www.debian.org/doc/manuals/reference/ch-system.de.html#s-pkg-basics.

[Debid] Debian. Sicherheit im Debian Betriebssystem. http://www.debian.org/doc/manuals/securing-debian-howto/ch10.de.html#s10.1.

[dUni] Rechenzentrum der Universität Karlsruhe. Die kleine Baumschule. http://www.rz.uni-karlsruhe.de/dienste/3405.php.

[EiSW00] Kerstin Eisenkolb, Jochen Sommer und Helge Weickardt. *Microsoft Active Directory - Der Verzeichnisdienst für Wondows 2000.* ADDISON-WESLEY Verlag. 2000.

[GoWe00a] Paul Goode und Ken Western. *Microsoft Windows 2000 Server - Die technische Referenz: Einsatzplanung.* Microsoft Press. 2000.

[GoWe00b] Paul Goode und Ken Western. *Microsoft Windows 2000 Server - Die technische Referenz: Verteilte Systeme.* Microsoft Press. 2000.

[Köln] Universität Köln. FAI (Fully Automatic Installation) for Debian/GNU Linux. http://www.informatik.uni-koeln.de/fai/.

[labm] labmice.techtarget.com. Unattended, Automated, and Remote Installations of Windows 2000. http://labmice.techtarget.com/windows2000/install/unattend_install.htm.

[Micra] Microsoft. Resource Kits - Web Resources. http://windows.microsoft.com/windows2000/reskit/webresources/.

[Micrb] Microsoft. Understanding Patch and Update Management: Microsoft Software Update Strategy. Technischer Bericht, http://www.microsoft.com/technet/security/topics /patch/patchmanagement.mspx.

[Micr99a] Microsoft. Remote Operating System Installation. http://www.microsoft.com/technet/prodtechnol/windows2000serv/deploy/ depopt/remoteos.mspx, September 1999.

[Micr99b] The ATM Forum (Hrsg.). Windows Installer Service Overview. White Paper, http://www.microsoft.com/windows2000/techinfo/howitworks/management/ installer.asp, April 1999.

[Micr00] Microsoft. Software Installation and Maintenance. White Paper, http://www.microsoft.com/windows2000/techinfo/ administration/management/ siamwp.asp, Februar 2000.

[Micr03] Microsoft. Automated Deployment Services Technical Overview. http://www.microsoft.com/windowsserver2003/techinfo/overview/ads.mspx, März 2003.

[RPM] RPM. RPM Package Manager. http://www.rpm.org/.

[RuSc97] Sven Rudolph und Heiko Schlittermann. Gut verpackt ist halb gewonnen. http://www.schlittermann.de/deb-intern/dpkg/, Juli 1997.

[ScSc] Björn Schneider und Holger Schweitzberger. Deploymenttechnologien und effizientes Client Rollout. *Microsoft*.

[Stös] Jan Stöss. *Die Baumschule*. Microbit vom Rechenzentrum der Universität Karlsruhe.

Abbildungsverzeichnis

Tabellenverzeichnis