

Dynamisches Cloning  
als optimierter Mobilitätsmechanismus  
für autonome Agenten<sup>1</sup>  
Analytisches Kostenmodell

Jörn Hartroth

1. September 1997

<sup>1</sup>Erschienen als Interner Bericht 21/97 der Fakultät für Informatik der Universität Karlsruhe

## **Zusammenfassung**

Dieser Bericht gibt die Motivation für die Verwendung des Clonings als Mobilitätsmechanismus in autonomen Agentensystemen. Auf der Grundlage einer vereinfachten Abbildung der Topologie heute existierender Netzwerke wird ein analytisches Kostenmodell für die Ausführung verteilter Anwendungen zur Datensammlung aufgestellt. Innerhalb dieses Modells werden der klassische Client/Server-Ansatz, der herkömmliche Migrationsansatz mobiler Agenten und der neue Cloning-Ansatz gegenübergestellt und bewertet.

Auf Grundlage des analytischen Modells werden für den Cloning-Ansatz Vorteile gegenüber den bisher verwendeten Verfahren erkannt und unter verschiedenen Annahmen über die der Anwendung unterliegende Datenverteilung bestätigt.

# 1 Ziele des mobilen Agentenansatzes

Eine der wesentlichen Motivationen für die Verwendung mobiler Agenten in verteilten Systemen besteht in der erwarteten Reduktion der im Rahmen eines entfernten Zugriffs anfallenden Netzwerkübertragungen sowohl bezüglich ihrer Anzahl als auch bezüglich des bewegten Datenvolumens (siehe [CGH95]). Als Referenz dient üblicherweise das verbreitete Client-Server-Schema. Aufgaben, die hochselektive Operationen auf großen Datenvolumina oder viele Einzelzugriffe enthalten, führen in diesem herkömmlichen Ansatz zu einem erheblichen Übertragungsvolumen im Verhältnis zur effektiven Größe der Ergebnisdaten. Bei Verwendung eines mobilen Agenten ist hingegen nur die einmalige Übertragung der Datenstruktur des Agenten vom Heimatrechner zum Ort des Datenbestands und die Rücksendung des gegenüber den berührten Daten wesentlich kleineren Ergebnisses zu veranschlagen, da die Zugriffs- und Selektionsoperationen vom Agenten lokal auf den Daten ausgeführt werden.

Ein wichtiges Merkmal mobiler Agenten ist die Fähigkeit, durch Migration zwischen verschiedenen Orten auch solche Aufgaben zu lösen, die die Interaktion mit mehreren verschiedenen Datenanbietern erfordern. Das nach aktuellem Forschungsstand verwendete Verfahren sieht dabei vor, daß ein mobiler Agent die relevanten Server sequentiell besucht, in lokaler Interaktion mit den Servern Teilergebnisse gewinnt, die er auf der weiteren Reise mitführt und mit denen er schließlich zu seinem Ausgangspunkt zurückkehrt. Die Mitnahme der akkumulierten Teilergebnisse führt hierbei je nach konkreter Anwendungsfunktion zu einem beständigen Anwachsen des mit dem mobilen Agenten assoziierten Zustands und erzeugt daher während der nachfolgenden Migrationen des Agenten zusätzliche Netzlast.

Das vorgeschlagene neue Verfahren des dynamischen Cloning als Mobilitätsmechanismus für autonome Agenten vermeidet diesen Anteil der Netzlast, indem anstelle des einen sequentiell wandernden Agenten eine Familie von durch Clonierung erzeugten Agenten zu den Servern delegiert wird, die ihre Teilergebnisse unmittelbar an einen Integrationspunkt zurückübertragen. Für Anwendungen, deren Funktionalität diese Art der Parallelverarbeitung zuläßt, indem die einzelnen Teilergebnisse in ihrer Berechnung voneinander unabhängig sind, kann so durch Vermeidung der Akkumulation von Teilergebnissen über mehrere Stationen hinweg eine Reduktion des Netzverkehrs erzielt werden.

Die nachfolgende analytische Betrachtung der anfallenden Kosten für die Ausführung verteilter Funktionen soll die Frage beantworten, unter welchen vom System vorgegebenen Randbedingungen die Verwendung der unterschiedlichen Mobilitätsparadigmen mobiler Agenten für datenakkumulierende verteilte Anwendungen Kostenvorteile erbringt.

## 2 Analytisches Kostenmodell für agentenbasierte verteilte Anwendungen

### 2.1 Grundannahmen

Für die folgende analytische Untersuchung werden die folgenden Grundannahmen getroffen:

1. Die allgemeinen Kosten für die Ausführung einer verteilten Aufgabe teilen sich in Übertragungskosten für den Code der Aufrufe und die Ergebnisse sowie Berechnungskosten für die Ausführung der anwendungsspezifischen Funktionen.
2. Die Kosten für die Datenübertragung werden in Abhängigkeit vom anfallenden Datenvolumen berechnet.
3. Die Berechnungskosten ergeben sich in Abhängigkeit vom Umfang der für die Funktion relevanten Daten. Hierbei wird unterschieden zwischen Datenselektion in einem Server und der Integration mehrerer separat selektierter Teilergebnisse.

- Die auszuführende Anwendungsfunktion hat primär akkumulierenden Charakter. Dies bedeutet, daß die von einem Datenanbieter gewonnenen Daten in ihrer Gesamtheit relevant sind und gesammelt ausgeliefert werden sollen. Funktionen, in denen mehrere Teilergebnisse zu einer kompakten Kennzahl integriert werden können (z.B. Ermittlung von eindimensionalen Extremwerten) erfüllen nicht diese Eigenschaft, können aber durch geeignete Parametrisierung im Rahmen des Kostenmodells erfaßt werden (siehe Abschnitt 2.3.6).

Diese Annahmen befinden sich in Übereinstimmung mit den allgemein verwendeten Szenarien für agentenbasierte Systeme (z.B. [Whi96]). Insbesondere spiegeln sie die Verhältnisse in einem lose gekoppelten weltweiten Rechnerverbund über Weitverkehrsnetze mit teilweise schmalbandig angebundenen mobilen Teilnehmern wider.

Weiterhin werden die folgenden vereinfachenden Annahmen zugrundegelegt, die in der nachfolgenden Abbildung 1 veranschaulicht sind:

Die Kommunikationskosten betragen einen konstanten Faktor  $k_x$  des übertragenen Datenvolumens. Als Abstraktion der unterschiedlichen Kostenbereiche in einem realen Netz wird eine dreifache Untergliederung gewählt. Die erste Stufe umfaßt einen Bereich günstiger Kommunikationsverbindungen, z.B. im Festnetz über leistungsfähige Backbones und wird im folgenden als *lokaler* Bereich bezeichnet. Durch lokale Verbindungen verbundene Server bilden lokale Inseln, zwischen denen teurere *globale* Kommunikationswege verlaufen, etwa in Form von schmalbandigen Weitverkehrsverbindungen. Besonders betrachtet wird nochmals die *Zugangsverbindung* der verteilten Anwendung auf das Gesamtnetz. Diese kann häufig als vom globalen Typ angesehen werden, wenn sich die Anwendung auf einem mit ins Festnetz eingebundenen Rechner befindet, oder zur Modellierung der im Kontext von mobilen Rechnern besonders teuren oder schmalbandigen Zugangswege über Telefonverbindungen und drahtlose Netze verwendet werden.

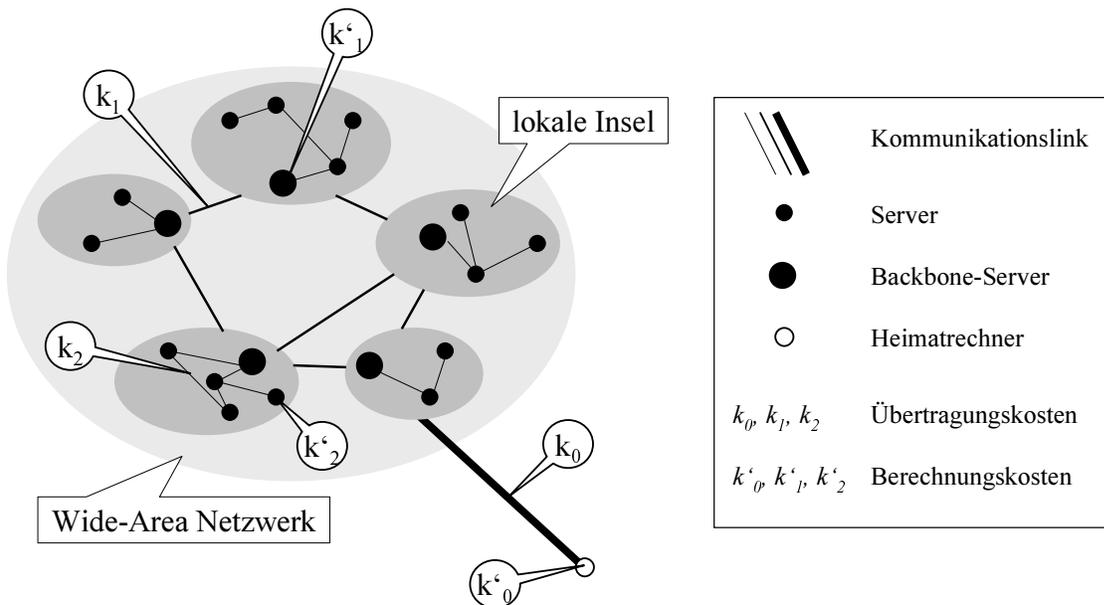


Abbildung 1: Übertragungskosten in der schematischen Topologie

Berechnungskosten werden ebenfalls mit einem konstanten Faktor  $k'_x$  angesetzt, der für den Nutzerrechner die Auslastung der vorhandenen Rechenressourcen und für die Serverrechner den dort anfallenden Berechnungsaufwand bzw. eine explizit vom Benutzer erhobene Gebühr beschreibt. Auch hier wird zwischen drei Stufen unterschieden, die die lokale Berechnung auf dem Rechner des Nutzers, besonders leistungsfähige Backbonerechner im Netz und allgemeine

Server modelliert. Für die nachfolgende Analyse wird angenommen, daß sich in jeder lokalen Insel ein Backbonerechner befindet.

Die zu übertragenden Informationen gliedern sich in die von den Informationsanbietern gelieferten Ergebnisdaten und — im Falle des Agentenansatzes — den Codeanteil der Agentenübertragung. Im Client–Server–Fall wird ein nicht speziell selektiertes Datenvolumen von jedem Server zum Rechner des Nutzers übertragen und dort ausgewertet. Bei Verwendung mobiler Agenten ist das zu übertragende Volumen durch die lokal ausgeführte Selektion um einen konstanten Selektionsfaktor verringert. Zudem wandern die Teilergebnisse nicht unmittelbar vom Server zum Heimatrechner zurück, sondern werden entsprechend dem Mobilitätsschema zunächst zwischen verschiedenen entfernten Orten übertragen. Bei migrierenden Agenten werden Zwischenergebnisse über Teile des Migrationspfads des Agenten mitgeführt, im Cloning–Ansatz werden sie in einer hierarchischen Baumstruktur in mehreren Schritten zum Ursprungsort der Anwendung zurücktransportiert.

## 2.2 Parameter

Die folgende Parameter werden im analytischen Modell verwendet:

- $k_0, k_1, k_2$  Faktoren für die Übertragungskosten,  $k_0$  auf der Zugangsverbindung des Nutzers,  $k_1$  im globalen und  $k_2$  im lokalen Bereich,
- $k'_0, k'_1, k'_2$  Faktoren für die Berechnungskosten,  $k'_0$  auf dem Rechner des Nutzers,  $k'_1$  auf den Backbonerechnern und  $k'_2$  auf allgemeinen Servern,
- $c_s, c_b$  Codegröße des mobilen Agenten, unterschieden nach Selektions- und Berechnungsfunktionalität für die Integration,
- $D$  Größe des relevanten Datenbestands auf einem Server, der von einer nicht durch spezifische Selektion eingegengten Anfrage zurückgeliefert wird,
- $s, b$  Reduktionsfaktor für die Ergebnisgröße, unterschieden nach Selektion und Integration,
- $s', b'$  Berechnungskostenfaktor für Selektion (bezogen auf  $D$ ) und Integration (bezogen auf mehrere Teilergebnisse) von Ergebnisdaten,
- $n_j$  Anzahl der Server innerhalb einer lokalen Insel  $j$ ,
- $m$  Anzahl der lokalen Kommunikationsinseln,
- $r_{hi}, R_{ik}^{jl}$  Anteil der redundanten Daten zwischen je zwei Servern innerhalb einer lokalen Insel (Server  $h$  und  $i$ ) und auf globaler Ebene (Server  $i$  in Insel  $j$  und Server  $k$  in Insel  $l$ ),
- $d_{hi}$  Kurzschreibweise für  $1 - r_{hi}$
- $r_j, R$  Anteil der replizierten Daten innerhalb einer lokalen Insel und auf globaler Ebene.

## 2.3 Abstrahierte Kostenfunktionen

Für die Durchführung der Analyse werden drei verschiedene Szenarien bezüglich der Verteilung der Daten auf den Servern untersucht. Im einfachsten Fall werden die relevanten Anteile der Datenbestände der einzelnen Server als disjunkt angenommen. Im zweiten Fall wird die Existenz von paarweise redundanten Datenanteilen angenommen. Im abschließenden Szenario ist ein Teil der Daten über mehrere Server repliziert. Die Feststellung der Disjunktheit oder

teilweiser Redundanz von Datenelementen kann sowohl auf tatsächlicher Gleichheit als auch auf anwendungsspezifischen Kriterien beruhen und braucht für diese Analyse nicht genauer fixiert zu werden.

Die im folgenden dargestellten Formeln beschreiben zunächst den allgemeinen Fall, in dem für jeden Server unterschiedliche Datenvolumina  $D_i^j$  und unterschiedliche Anteile für die paarweise Redundanz  $r_{hi}^j, R_{ik}^{jl}$  und Replikation  $r_j, R$  erlaubt sind. Um die symbolische Darstellung handhabbar zu halten, wird für die symbolischen Umformungen die vereinfachende Annahme gemacht, daß Datenvolumina mit  $D$  gleichverteilt und die Faktoren für paarweise Redundanz  $r, R$  und Replikation  $r, R$  jeweils für die betrachtete Anwendung konstant sind. In den Gleichungen ist dies durch das modifizierte Gleichheitszeichen  $='$  gekennzeichnet. Diese Einschränkung bedeutet in der Tat eine sehr weitgehende Abstraktion von der Realität. Wie zusätzliche simulative Untersuchungen (siehe [Kau97]) und die weiter unten angestellte Grenzwertbetrachtung zeigen, behalten die qualitativen Erkenntnisse dieser Analyse auch für kompliziertere Modellierungen ihre Gültigkeit.

Ebenfalls aus Handhabbarkeitsgründen wird für die Wegwahl der verwendeten mobilen Agenten ein festes, auf die Topologie abgestimmtes Schema angenommen, das für die angenommene Gleichverteilung der Daten jeweils minimale Transportkosten erzeugt:

- Der migrierende Agent bewegt sich gemäß dieser Vorschrift vom Ausgangsrechner über die Zugangsverbindung mit Kosten  $k_0$  auf einen Server des Festnetzes. Von dort aus besucht er zunächst alle im lokalen ( $k_2$ ) Bereich gelegenen Server und wechselt dann über eine globale Verbindung ( $k_1$ ) in die nächste lokale Insel. In dieser Art fährt er fort, bis er vom letzten Server der letzten besuchten Insel aus die akkumulierten Daten wiederum über die Zugangsverbindung zum Ausgangsort zurücksendet (siehe Abbildung 2). Als typische Optimierung kann im letzten Schritt auf die Rücksendung des Codes zum Ausgangsort verzichtet werden.
- Im Cloning-Ansatz wird zunächst ein einzelner Agent über die Zugangsverbindung mit Kosten  $k_0$  ins Festnetz plaziert. Dieser Agent entsendet jeweils einen Clone über globale Verbindungen ( $k_1$ ) in jede andere lokale Insel und wird damit zur "Wurzel" einer baumförmigen Cloning-Hierarchie. Innerhalb der lokalen Inseln wird je ein Agent zu den benachbarten Servern entsandt ( $k_2$ ), wo er die Datenselektion ausführt und sein Teilergebnis zurücksendet. Je ein Clone pro lokaler Insel führt die Integration der an ihn gesandten Teilergebnisse zurück und schickt das Resultat an die Wurzel ( $k_1$ ). Der Wurzelagent integriert die akkumulierten Teilergebnisse seiner Clones und liefert das erhaltene Gesamtergebnis zum Ausgangsort ( $k_0$ ) zurück (siehe Abbildung 2). Hierin begründet sich die Unterscheidung des Parameters für die Codegröße in einen selektions- und einen integrationsorientierten Teil, denn im Gegensatz zum migrierenden Agenten, bei dem der gesamte Code mitgeführt wird, führen die Clone-Agenten auf der letzten Stufe der Hierarchie nur die Datenselektion aus und benötigen daher nur den Codeanteil  $c_s$ .
- Der Berechnungsaufwand fällt für die Datenselektion auf jedem Server in der Größe  $k_1^l s^l D$  bzw.  $k_2^l s^l D$  an. Die nachfolgende Datenintegration schlägt im migrierenden Agenten mit  $k_{[1,2]}^l b^l (s D_i^j + \text{Größe der bisher akkumulierten Daten})$  zu Buche. Im Cloning-Ansatz wird Ergebnisintegration nur auf den Backbone-Rechnern ausgeführt, so daß die Berechnungskosten hier  $k_1^l b^l (s D_i^j + \text{Selektionsergebnisse der delegierten Clone-Agenten})$  betragen.

Abbildung 2 veranschaulicht die durch diese Mobilitätsschemata vorgegebenen Datenflüsse auf der in Abbildung 1 eingeführten vereinfachten Topologie eines Weitverkehrsnetzes.

Dieses Schema berücksichtigt durch die explizite Modellierung der Zugangsverbindung speziell den Fall des schmalbandig (z.B. drahtlos) angebundenen Nutzerrechners. Der Fall eines

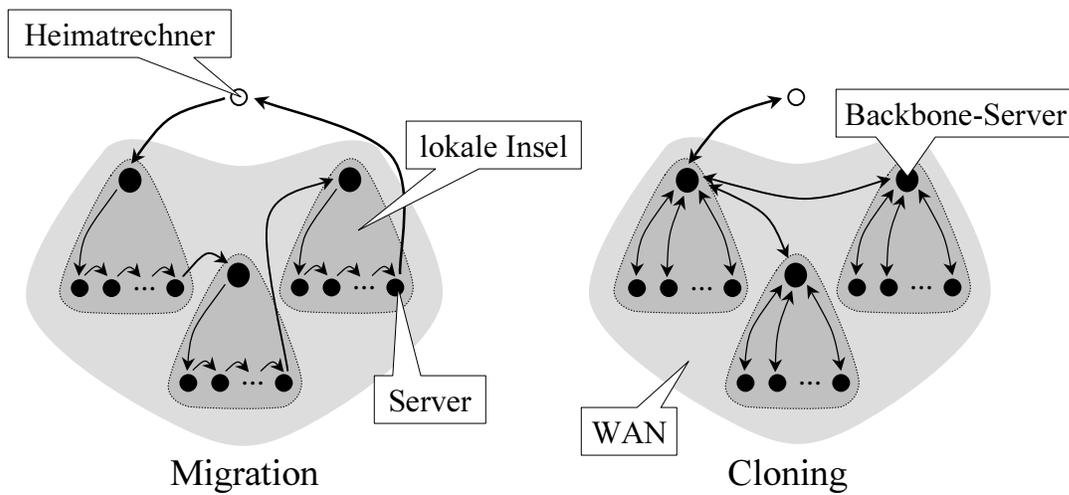


Abbildung 2: Mobilitätsschemata

ins Festnetz eingebetteten Nutzers kann durch die Wahl des Parameters  $k_0 = k_1$  ebenfalls durch das Schema erfaßt werden. In den Funktionen für den Client-Server-Ansatz im reinen Festnetz ist der Faktor  $k_0$  durch  $k_1$  für allgemeine Server und durch  $k_2$  für die Server in der lokalen Insel des Nutzers zu ersetzen, da hier jeweils die direkte Verbindung vom Nutzerrechner zum Server eingeht.

In den folgenden Abschnitten werden jeweils die vereinfachten Formeln für Übertragungs- und Berechnungskosten in den verschiedenen Szenarien unter Annahme einer schmalbandigen Nutzeranbindung gegenübergestellt. Zur Illustration des relativen Verhaltens ist jeweils eine Grafik beigelegt, die die Kosten für eine Serveranzahl zwischen 8 und 128 darstellt, wobei im Sinne einer besseren Darstellbarkeit der Parameter  $n$  für die Inselgröße fest gewählt ist. Alle übrigen Parameter sind in den verschiedenen Szenarien identisch. Entsprechend den topologischen Grundannahmen gelten die Bedingungen  $k_0 > k_1 > k_2$  und  $k'_1 < k'_2$ . Der Faktor  $k'_0$  für die Berechnungen auf dem Nutzerrechner hat keine Auswirkungen auf das relative Verhalten der beiden Formen der Agentenmobilität, bestimmt aber wesentlich den Vergleich zwischen Client-Server- und Agentenansatz allgemein. Für die weitere Analyse wird eine begrenzte Leistung des Nutzerrechners ( $k'_0 \geq k'_2$ ) angenommen, ein alternatives Szenario mit  $k'_0 \ll (k'_1, k'_2)$  begünstigt den Client-Server-Ansatz insgesamt.

### 2.3.1 Disjunkte Daten

In diesem Szenario werden Daten von verschiedenen Servern nachgefragt, die jeweils einmalig sind und daher in ihrer Gesamtheit an den Nutzer ausgeliefert werden müssen. Ein Beispiel hierfür ist eine verteilte Anwendung, die im Rahmen der Beschaffung eines aus mehreren Komponenten bestehenden Rechnersystems von einer Reihe von Anbietern die aktuellen Konditionen einholt. Da es für die Einzelkomponenten jeweils mehrere Konfigurationsmöglichkeiten (z.B. Hersteller des Mainboards und der Festplatte) gibt, genügt es nicht, den Minimalpreis einer Komponente im Agenten mitzuführen, sondern es müssen die vollständigen Daten in Form einer Typbezeichnung und ggf. zusätzliche Informationen wie Garantiedauer und Lieferfristen gesammelt werden.

Der migrierende mobile Agent hat hier keine Möglichkeit, durch Datenintegration eine Reduktion des akkumulierten Volumens zu erzielen. Wohl aber erreicht der mobile Agentenansatz generell eine Reduktion des Datenvolumens gegenüber dem Client-Server-Ansatz, da vor Ort eine Vorselektion ausgeführt wird.

## Übertragungskosten

$$\begin{aligned}
 C_{CS} &= \sum_{j=1}^m \sum_{i=1}^n k_0 D_i^j = k_0 m n D \\
 X_j &= \sum_{l=1}^j \sum_{k=1}^{n_j} s D_k^l = j n s D \\
 C_{mig} &= k_0 (c_s + c_b) + \sum_{j=1}^{m-1} \left[ k_2 \sum_{i=1}^{n_j-1} (c_s + c_b + \sum_{h=1}^i s D_h^j + X_{j-1}) + k_1 (c_s + c_b + X_j) \right] \\
 &\quad + k_2 \sum_{i=1}^{n_m-1} (c_s + c_b + \sum_{h=1}^i s D_h^m + X_{m-1}) + k_0 X_m \\
 &= k_0 (c_s + c_b + b m n s D) + k_1 (m-1) (c_s + c_b + \frac{1}{2} b m n s D) \\
 &\quad + k_2 m (n-1) (c_s + c_b + \frac{1}{2} b m n s D) \\
 C_{clo} &= k_0 (c_s + c_b) + \sum_{j=1}^{m-1} \left[ k_1 (c_s + c_b) + k_2 \sum_{i=1}^{n_j-1} (c_s + s D_i^j) + k_1 \sum_{i=1}^{n_j} s D_i^j \right] \\
 &\quad + k_2 \sum_{i=1}^{n_m-1} (c_s + s D_i^m) + k_0 X_m \\
 &= k_0 (c_s + c_b + b m n s D) + k_1 (m-1) (c_s + c_b + b n s D) + k_2 m (n-1) (c_s + s D)
 \end{aligned}$$

Die Kostenfunktion für den Übertragungsaufwand im Cloning-Ansatz ist in den Termen  $k_1$  und  $k_2$  echt kleiner und im Term  $k_0$  gleich dem Migrationsansatz. Für eine hinreichend große Selektivität der Anwendungsfunktion ( $s \ll 1$ ) sind beide agentenbasierte Ansätze dem Client-Server-Ansatz überlegen. Im Gegensatz zum Cloning-Ansatz tritt jedoch im Migrationsansatz, bedingt durch die Akkumulation der Daten über den Migrationspfad des Agenten, eine quadratische Abhängigkeit von der Anzahl der besuchten Server ( $m$  bzw.  $n$ ) auf, was zu einem ungünstigen Verhalten für große Serverpopulationen führt.

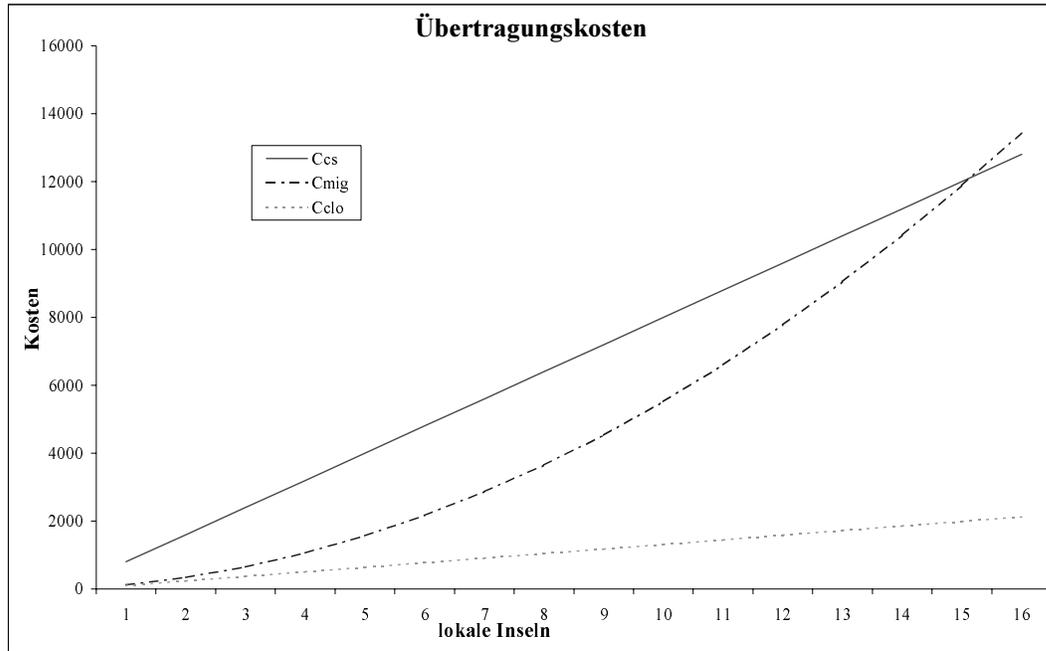


Abbildung 3: Übertragungskosten für disjunkte Daten

## Berechnungskosten

$$\begin{aligned}
 C'_{CS} &= k'_0 \sum_{j=1}^m \sum_{i=1}^n (s' + b') D_i^j = k'_0 m n (s' + b') D \\
 X_j &= \sum_{l=1}^j \sum_{k=1}^{n_j} s D_k^l = j n s D \\
 C'_{mig} &= \sum_{j=1}^m \left[ \sum_{i=1}^{n-1} k'_2 \left( s' D_i^j + b' \left( \sum_{h=1}^i s D_h^j + X_{j-1} \right) \right) + k'_1 (s' D_i^j + b' X_{j-1}) \right] \\
 &= k'_1 (m s' D + \frac{1}{2} b' m (m-1) n s D) + k'_2 m (n-1) s' D (1 + \frac{1}{2} b' n) \\
 C'_{clo} &= \sum_{j=1}^{m-1} \left[ k'_1 s' D_1^j + \sum_{i=1}^{n_j-1} k'_2 s' D_i^j + k'_1 b' \sum_{i=1}^{n_j} s D_i^j \right] + k'_1 s' D_1^j + \sum_{i=1}^{n_j-1} k'_2 s' D_i^j + k'_1 b' X_m \\
 &= k'_1 (m s' D + (2m-1) b' n s D) + k'_2 m (n-1) s' D
 \end{aligned}$$

Auch für die berechnungsbezogenen Kosten ergibt sich ein Vorteil beim Cloning-Ansatz, da die Kosten in beiden Termen  $k'_1$  und  $k'_2$  echt kleiner sind als beim Migrationsansatz. Für die angenommenen begrenzten Rechenressourcen auf dem Nutzerrechner ergibt sich ein großer Faktor  $k'_0$ , so daß der Client-Server-Ansatz wiederum ungünstig abschneidet. Auch hier führt die Akkumulation von Teilergebnissen im Migrationsansatz zu einem quadratischen Term, der schließlich für große Serverpopulationen den Migrationsansatz disqualifiziert.

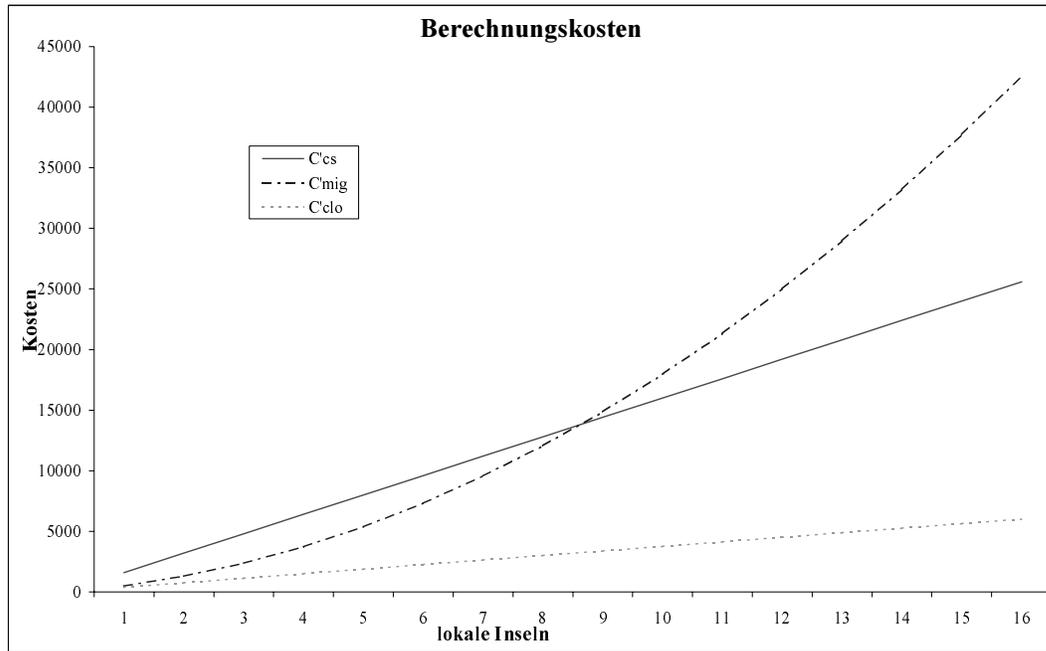


Abbildung 4: Berechnungskosten für disjunkte Daten

### 2.3.2 Redundante Daten

In diesem Szenario besteht zwischen den Datenbeständen je zweier Server ein redundanter Anteil, der im lokalen Bereich  $r$  und im globalen Bereich  $R$  beträgt. Im Beispiel der Kosten- nachfrage für ein Rechnersystem tritt ein solcher Fall dann auf, wenn etwa mehrere Anbieter ein Produkt vom selben Großhändler beziehen und damit identische Konditionen für Preise und Lieferfristen anbieten.

Ein migrierender mobiler Agent kann durch Integration der nacheinander gewonnenen Daten diesen redundanten Anteil schon am Ort des Servers entfernen und dadurch die Gesamtgröße des akkumulierten Datenbestands vermindern. Die parallel arbeitenden Agenten des Cloning-Ansatzes liefern die selektierten Daten zunächst mitsamt dem Redundanzanteil an ihren Erzeuger zurück. Eine dort ausgeführte Integration führt ebenfalls zur Entfernung der redundanten Anteile. Dieses Szenario begünstigt durch die sofortige Integrationsmöglichkeit den migrierenden gegenüber dem Cloning-Ansatz.

## Übertragungskosten

$$\begin{aligned}
C_{CS} &= \sum_{j=1}^m \sum_{i=1}^n k_0 D_i^j = k_0 m n D \\
R_j^{uv} &= \sum_{l=1}^j \sum_{k=1}^{n_l} R_{uk}^{vl} \\
X_j &= \sum_{l=1}^j \sum_{k=1}^{n_l} s D_k^l \left(1 - \sum_{g=1}^{h-1} r_{gh} - R_{l-1}^{kl}\right) = j n s D \left(1 - \frac{1}{2}(n-1)r - \frac{1}{2}(j-1)nR\right) \\
C_{mig} &= k_0(c_s+c_b) + \sum_{j=1}^{m-1} \left[ k_2 \sum_{i=1}^{n_j-1} \left( c_s+c_b + \sum_{h=1}^i s D_h^j \left(1 - \sum_{g=1}^{h-1} r_{gh} - R_{j-1}^{hj}\right) + X_{j-1} \right) + k_1(c_s+c_b + X_j) \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} \left( c_s+c_b + \sum_{h=1}^i s D_h^m \left(1 - \sum_{g=1}^{h-1} r_{gh} - R_{m-1}^{hm}\right) + X_{m-1} \right) + k_0 X_m \\
&= k_0(c_s+c_b + m n s D \left(1 - \frac{1}{2}(n-1)r - \frac{1}{2}(m-1)nR\right)) \\
&\quad + k_1(m-1)(c_s+c_b + \frac{1}{2} m n s D \left(1 - \frac{1}{2}(n-1)r - \frac{1}{3}(m-2)nR\right)) \\
&\quad + k_2 m(n-1)(c_s+c_b + \frac{1}{2} n s D \left(1 - \frac{1}{3}(n-2)r - \frac{1}{2}(m-1)nR\right)) \\
&\quad + \frac{1}{2}(m-1) n s D \left(1 - \frac{1}{2}(n-1)r - \frac{1}{3}(m-2)nR\right) \\
C_{clo} &= k_0(c_s+c_b) + \sum_{j=1}^{m-1} \left[ k_1(c_s+c_b) + k_2 \sum_{i=1}^{n_j-1} (c_s+c_b + s D_i^j) + k_1 \sum_{i=1}^{n_j} s D_i^j \left(1 - \sum_{h=1}^{i-1} r_{hi}\right) \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} (c_s+c_b + s D_i^m) + k_0 X_m \\
&= k_0(c_s+c_b + m n s D \left(1 - \frac{1}{2}(n-1)r - \frac{1}{2}(m-1)nR\right)) \\
&\quad + k_1(m-1)(c_s+c_b + n s D \left(1 - \frac{1}{2}(n-1)r\right)) + k_2 m(n-1)(c_s + s D)
\end{aligned}$$

## Berechnungskosten

$$\begin{aligned}
C'_{CS} &= k'_0 \sum_{j=1}^m \sum_{i=1}^n (s' + b') D_i^j = k'_0 m n (s' + b') D \\
X_j &= \sum_{l=1}^j \sum_{k=1}^{n_j} s D_k^l = j n s D \\
C'_{mig} &= \sum_{j=1}^m \left[ \sum_{i=1}^{n-1} k'_2 \left( s' D_i^j + b' \left( s D_i^j + \sum_{h=1}^{i-1} s D_h^j \left(1 - \sum_{g=1}^{h-1} r_{gh} - R_{j-1}^{hj}\right) + X_{j-1} \right) \right) \right. \\
&\quad \left. + k'_1 \left( s' D_i^j + b' \left( s D_i^j + \sum_{h=1}^{n-1} s D_h^j \left(1 - \sum_{g=1}^{h-1} r_{gh} - R_{j-1}^{hj}\right) + X_{j-1} \right) \right) \right]
\end{aligned}$$

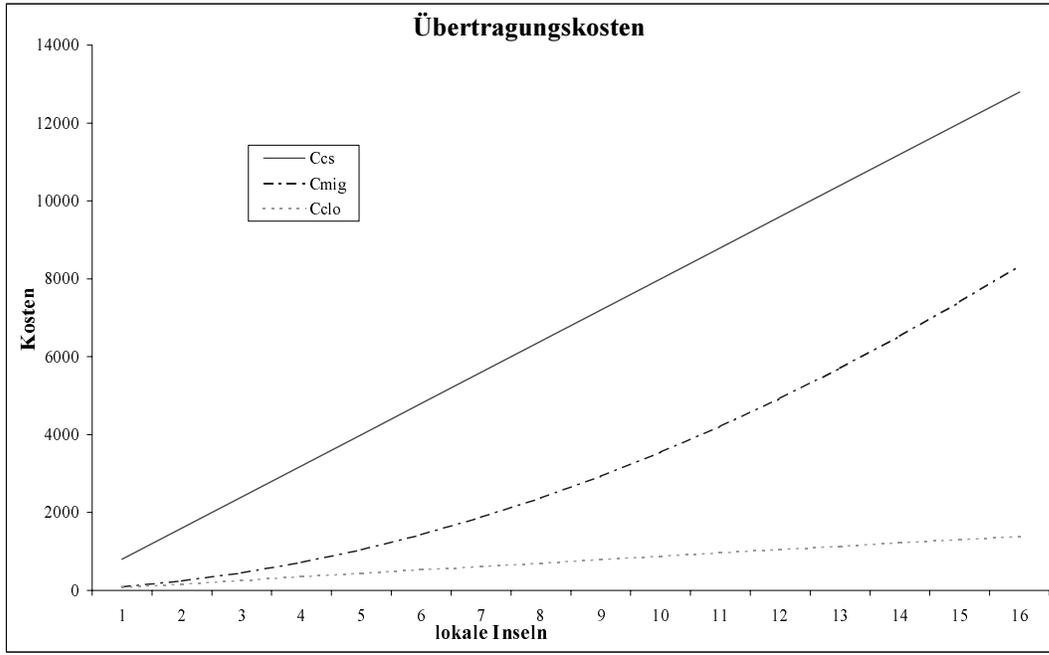


Abbildung 5: Übertragungskosten für redundante Daten

$$\begin{aligned}
&= k'_1 (ms'D + b'nsD (n(n-3)(\frac{1}{6}R - \frac{1}{4}r) + \frac{1}{2}n(1+r) - r) \\
&\quad + \frac{1}{2}b'm^2nsD (1 - \frac{1}{3}(m-3)R - \frac{1}{2}(n-1)r)) \\
&\quad + k'_2m(n-1) (s'D + b'sD (-\frac{1}{6}m^2n^2R + \frac{1}{4}mn((n-1)(R-r) + 3R + 2) \\
&\quad - \frac{1}{12}(n(n-1)(R-r) - 8(n-1)r + 7nR + 4r))) \\
C'_{clo} &= \sum_{j=1}^{m-1} \left[ k'_1s'D_1^j + \sum_{i=1}^{n_j-1} k'_2s'D_i^j + k'_1b' \sum_{i=1}^{n_j} sD_i^j \right] \\
&\quad + k'_1s'D_1^j + \sum_{i=1}^{n_j-1} k'_2s'D_i^j + k'_1b' \left( \sum_{l=1}^m \sum_{k=1}^{n_l} sD_k^l (1 - \sum_{h=1}^{k-1} r_{hk}) \right) \\
&= k'_1 (ms'D + (2m-1)b'nsD - \frac{1}{2}mn(n-1)b'rsD) + k'_2m(n-1)s'D
\end{aligned}$$

### 2.3.3 Replizierte Daten

In diesem Szenario wird ein Anteil  $r$  der Daten über alle Server einer lokalen Insel repliziert. Ein weiterer Teil  $R$  ist global über alle betrachteten Server hinweg repliziert. Konkretes Beispiel bei der Rechnerbeschaffung wären die für einen Wirtschaftsraum (entsprechend einer lokalen Insel) identischen Regelungen für Garantie und Gewährleistung, die vom Agenten nur ein einfacher Ausfertigung zur kontrollierenden Anwendung zurückgemeldet werden müssen. Die Möglichkeiten der Ausfilterung der hier mehrfach vorhandenen Datenanteile durch Integration während der Migration bzw. nach der Rückkehr der delegierten Agentenclones entsprechen den Möglichkeiten, die im vorstehenden Abschnitt für redundante Daten vorgestellt wurden. Der Verlauf der anfallenden Datenvolumina ist jedoch bei redundanten und replizierten Verteilungen unterschiedlich, da im ersteren Fall der auf einem Server vorgefundene neue Anteil beständig schrumpft (wie  $(1 - \sum_{h=1}^{i-1} r)sD_i^j$ ), im letzteren konstant den Wert  $(1-r)sD_i^j$  beibehält. Somit ist eine teilreplizierte Datenverteilung nochmals günstiger für den migrierenden Ansatz, da hier der von den später besuchten Servern stammende und damit über

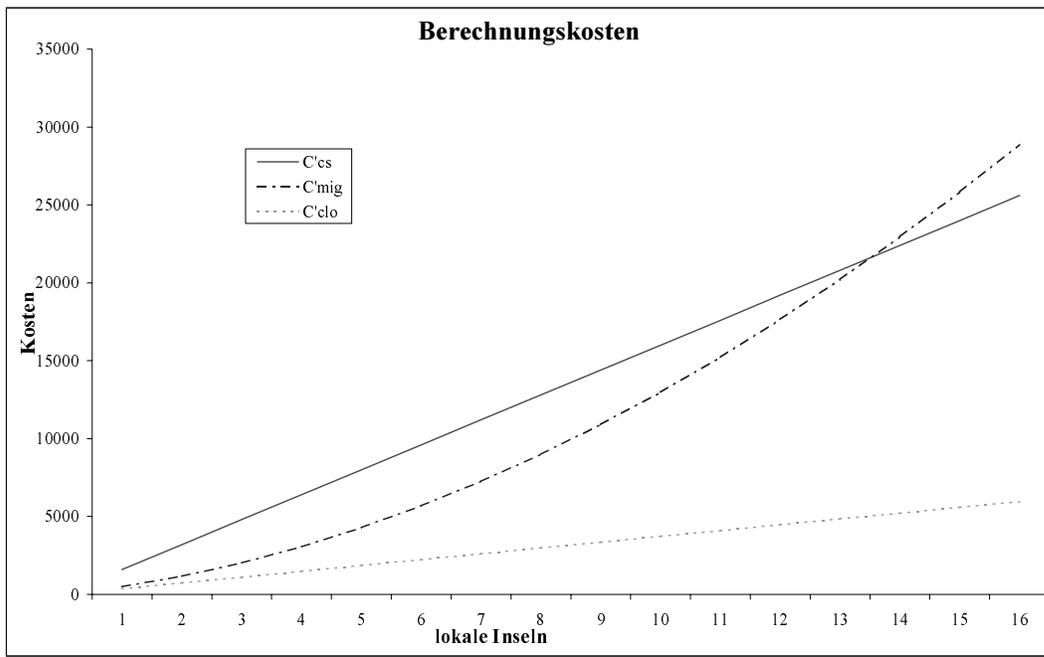


Abbildung 6: Berechnungskosten für redundante Daten

einen kürzeren (und damit günstigeren) Weg mitgeführte Datenanteil größer ist als bei der teilredundanten Verteilung.

### Übertragungskosten

$$\begin{aligned}
C_{CS} &= \sum_{j=1}^m \sum_{i=1}^n k_0 D_i^j = k_0 m n D \\
X_j &= \sum_{l=1}^j \left( \sum_{k=1}^{n_l} s D_k^l (1 - r_l - R) + r_l s D_1^l \right) + R s D_1^j = j (n s D (1 - r - R) + r s D) + R s D \\
C_{mig} &= k_0 (c_s + c_b) + \sum_{j=1}^{m-1} \left[ k_2 \sum_{i=1}^{n_j-1} \left( c_s + c_b + \sum_{h=1}^j s D_h^j (1 - r_j - R) + r_j s D_1^j + X_{j-1} \right) + k_1 (c_s + c_b + X_j) \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} \left( c_s + c_b + \sum_{h=1}^m s D_h^m (1 - r_m - R) + r_m s D_1^m + X_{m-1} \right) + k_0 X_m \\
&= k_0 (c_s + c_b + m n s D (1 - r - R) + m r s D + R s D) \\
&\quad + k_1 (m - 1) (c_s + c_b + \frac{1}{2} m n s D (1 - r - R) + \frac{1}{2} m r s D + R s D) \\
&\quad + k_2 m (n - 1) (c_s + c_b + \frac{1}{2} m n s D (1 - r - R) + \frac{1}{2} (m + 1) r s D + R s D) \\
C_{clo} &= k_0 (c_s + c_b) + \sum_{j=1}^{m-1} \left[ k_1 (c_s + c_b) + k_2 \sum_{i=1}^{n_j-1} (c_s + c_b + s D_i^j) + k_1 \sum_{i=1}^{n_j} s D_i^j (1 - r_j) + r_j s D_1^j \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} (c_s + c_b + s D_i^m) + k_0 X_m \\
&= k_0 (c_s + c_b + m n s D (1 - r - R) + m r s D + R s D) + k_1 (m - 1) (c_s + c_b + n s D (1 - r) + r s D) \\
&\quad + k_2 m (n - 1) (c_s + s D)
\end{aligned}$$

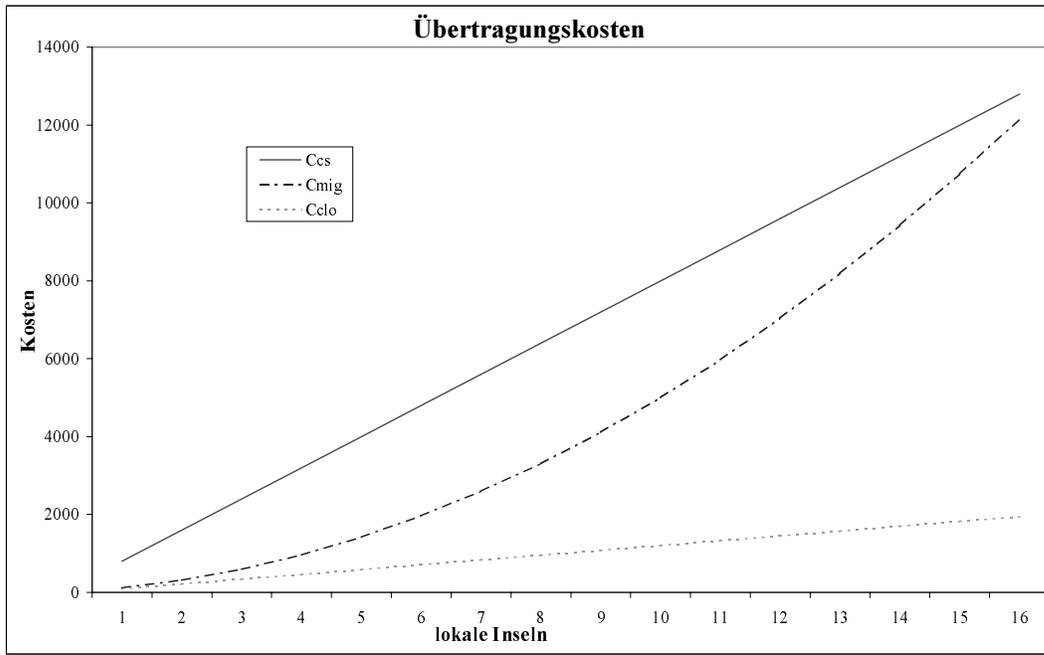


Abbildung 7: Übertragungskosten für replizierte Daten

### Berechnungskosten

$$\begin{aligned}
C'_{CS} &= k'_0 \sum_{j=1}^m \sum_{i=1}^n (s' + b') D_i^j = k'_0 m n (s' + b') D \\
X_j &= \sum_{l=1}^j \sum_{k=1}^{n_j} s D_k^l = j n s D \\
C'_{mig} &= \sum_{j=1}^m \left[ \sum_{i=1}^{n-1} k'_2 \left( s' D_i^j + b' \left( s D_i^j + \sum_{h=1}^{i-1} s D_h^j (1 - r_j - R) + r s D_1^j + X_{j-1} \right) \right) \right. \\
&\quad \left. + k'_1 \left( s' D_i^j + b' \left( s D_i^j + \sum_{h=1}^{n-1} s D_h^j (1 - r_j - R) + r s D_1^j + X_{j-1} \right) \right) \right] \\
&= k'_1 (m s' D + \frac{1}{2} m (m + 1) b' n s D (1 - r - R) + \frac{1}{2} b' m s D ((m + 3)r + 4R)) \\
&\quad + k'_2 m (n - 1) (s' D + \frac{1}{2} b' m n s D (1 - r - R) + \frac{1}{2} b' s D ((m + 3)r + 4R)) \\
C'_{clo} &= \sum_{j=1}^{m-1} \left[ k'_1 s' D_1^j + \sum_{i=1}^{n_j-1} k'_2 s' D_i^j + k'_1 b' \sum_{i=1}^{n_j} s D_i^j \right] \\
&\quad + k'_1 s' D_1^j + \sum_{i=1}^{n_j-1} k'_2 s' D_i^j + k'_1 b' \left( \sum_{l=1}^m \sum_{k=1}^{n_l} s D_k^l (1 - r_l) + r_l s D_1^l \right) \\
&= k'_1 (m s' D + (2m - 1) b' n s D - b' m (n - 1) r s D) + k'_2 m (n - 1) s' D
\end{aligned}$$

### 2.3.4 Verfeinernde Daten

Nachdem in den bisherigen Szenarien die Integrationsfunktion des mobilen Agenten alle von der Selektion gelieferten Datenelemente unter Entfernung von Mehrfachvorkommen, bedingt durch Redundanz oder Replikation, angesammelt hat, wird in diesem Szenario nur derjenige Datenanteil ausgewählt, der auf allen Servern einer lokalen Insel gleichermaßen vorliegt. Dies

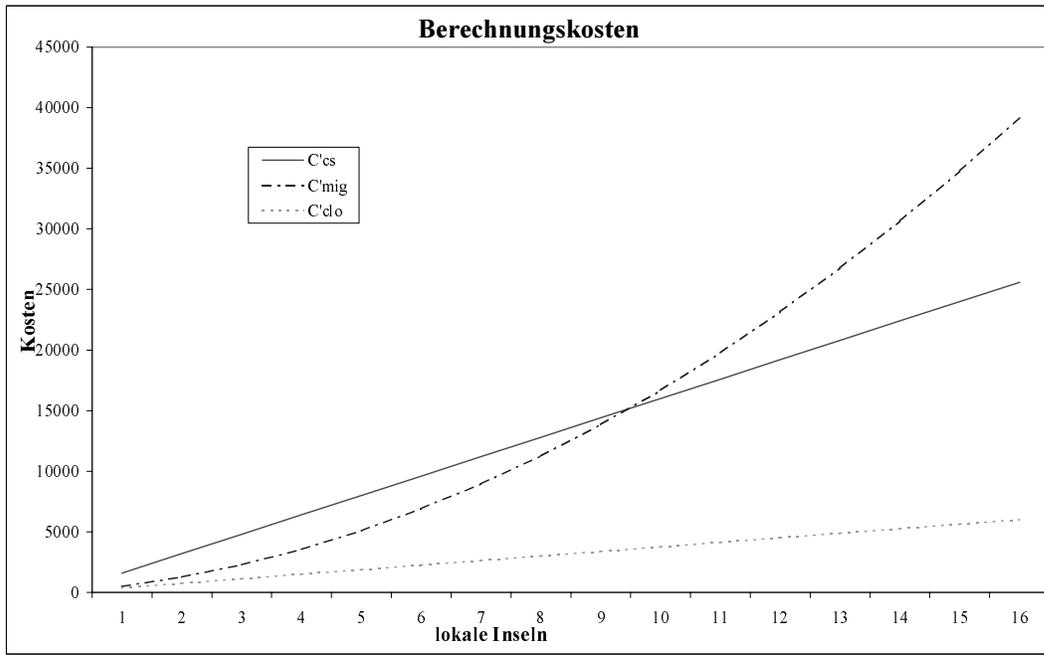


Abbildung 8: Berechnungskosten für replizierte Daten

stellt den Extremfall einer Aufgabe dar, in der ein gewisser Redundanzgrad der ermittelten Daten explizit erwünscht ist. Im Anwendungsbeispiel der Rechnerbeschaffung tritt dies auf, wenn der Auftraggeber etwaige widersprüchliche Angaben zu Komponenten eliminieren will und nur übereinstimmende zurückerhalten möchte.

Dieses Szenario bedeutet eine weitere Begünstigung des Migrationsansatzes gegenüber dem Cloning-Ansatz, da die vom migrierenden Agenten innerhalb einer lokalen Insel transportierten Datenvolumina durch Beseitigung nicht übereinstimmender Datenanteile stetig abnehmen (wie  $(1 - \sum_{h=2}^i sD)$ ), während die ausgesandten Clones jeweils den vollen selektierten Datenbestand  $sD$  des besuchten Servers zurücksenden. In der graphischen Darstellung wird demgemäß auch deutlich, daß für wenige lokale Inseln der Migrationsansatz hier günstiger arbeiten kann als der Cloning-Ansatz, bevor bei zunehmender Anzahl von Inseln der im globalen Netzbereich nach wie vor anfallende quadratische Anteil Überhand nimmt.

Da innerhalb der lokalen Inseln der Migrationsansatz durch die stetig abnehmende mitgeführte Datenmenge besonders günstig arbeitet, kann in diesem Szenario ein hybrider Ansatz, bei dem die im ersten Schritt erzeugten Clones ihre lokale Insel durch Migration bearbeiten, eine weitere Verbesserung erzielen. Das hierbei erreichte Ergebnis ist in den nachfolgenden Formeln und in Abbildung 9 mit  $C_{clo/mig}$  bezeichnet.

$$\begin{aligned}
C_{CS} &= \sum_{j=1}^m \sum_{i=1}^n k_0 D_i^j = k_0 m n D \\
X_j &= \sum_{l=1}^j \left( s D_1^l - \sum_{k=2}^{n_j} d s D_k^l \right) \\
C_{mig} &= k_0 (c_s + c_b) + \sum_{j=1}^{m-1} \left[ k_2 \sum_{i=1}^{n_j-1} (c_s + c_b + s D_1^i - \sum_{h=2}^i d s D_h^i) - X_{j-1} \right] + k_1 (c_s + c_b + X_j) \\
&\quad + k_2 \sum_{i=1}^{n_m-1} (c_s + c_b + s D_1^i - \sum_{h=2}^i d s D_h^i) - X_{m-1} + k_0 (X_m)
\end{aligned}$$

$$\begin{aligned}
&= k_0(c_s+c_b + msD(1 - (n-1)d)) + k_1(m-1)(c_s+c_b + \frac{1}{2}msD(1 - (n-1)d)) \\
&\quad + k_2m(n-1)(c_s+c_b + \frac{1}{2}(d+1)sD + \frac{1}{2}msD(1 - (n-1)r)) \\
C_{clo} &= k_0(c_s+c_b) + \sum_{j=1}^{m-1} \left[ k_1(c_s+c_b) + k_2 \sum_{i=1}^{n_j-1} (c_s + sD_i^j) + k_1X_j \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} (c_s + sD_i^m) + k_0(X_m) \\
&= k_0(c_s+c_b + msD(1 - (n-1)d)) + k_1(m-1)(c_s+c_b + sD(1 - (n-1)d)) \\
&\quad + k_2m(n-1)(c_s + sD) \\
C_{clo/mig} &= k_0(c_s+c_b) + \sum_{j=1}^{m-1} \left[ k_1(c_s+c_b) + k_2 \sum_{i=1}^{n_j-1} (c_s+c_b + sD_1^j - \sum_{h=2}^i dsD_h^j) + k_1X_j \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} (c_s+c_b + sD_1^m - \sum_{h=2}^i dsD_h^m) + k_0(X_m) \\
&= k_0(c_s+c_b + msD(1 - (n-1)d)) + k_1(m-1)(c_s+c_b + sD(1 - (n-1)d)) \\
&\quad + k_2m(n-1)(c_s+c_b + sD(1 - (n-1)d))
\end{aligned}$$

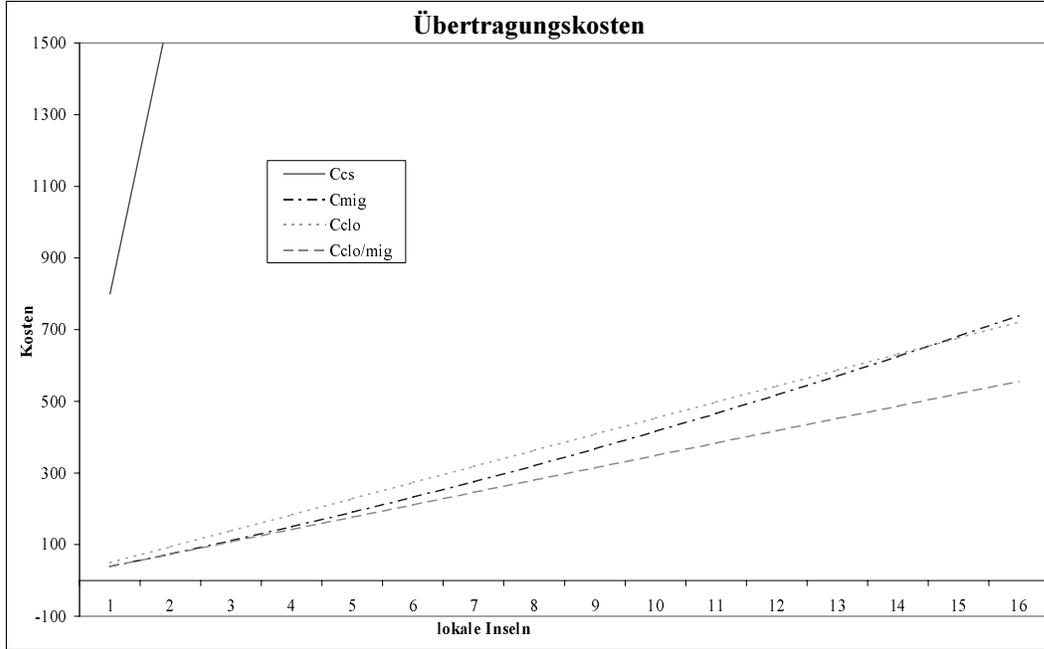


Abbildung 9: Übertragungskosten für verfeinernde Daten

### 2.3.5 Generisches Modell

Die drei vorstehend aufgeführten Szenarien für das Verhalten mobiler Agenten unter unterschiedlichen Annahmen über die Datenverteilung bilden spezielle Ausprägungen eines generischen Kostenmodells. Die allgemeine Formulierung der Kostenfunktionen für Datenübertragung und Berechnung akkumulierender Anwendungen wird im folgenden aufgeführt. Diese Darstellung ist nur noch von den in Kapiteln 2.1 und 2.3 gemachten Annahmen über das Topologie- und Bewegungsschema abhängig und erlaubt es, über die vier ausführlich diskutierten Beispiele hinaus je nach Phantasie des Lesers mit weiteren unterschiedlichen Datenverteilungen parametrisiert zu werden.

## Übertragungskosten

$$\begin{aligned}
C_{CS} &= \sum_{j=1}^m \sum_{i=1}^n k_0 D_i^j \\
b_{mig}(i, j) &= \begin{cases} f(sD_i^j, b_{mig}(i-1, j)) & i, j > 1 \\ f(sD_i^j, b_{mig}(n, j-1)) & i = 1, j > 1 \\ f(sD_i^j) & i, j = 1 \end{cases} \\
C_{mig} &= k_0(c_s + c_b) + \sum_{j=1}^{m-1} \left[ k_2 \sum_{i=1}^{n_j-1} (c_s + c_b + b_{mig}(i, j)) + k_1(c_s + c_b + b_{mig}(n, j)) \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} (c_s + c_b + b_{mig}(i, m)) + k_0 b_{mig}(n_m, m) \\
b_{clo} &= f(b_{clo}(1), b_{clo}(2), \dots, b_{clo}(m)) \\
b_{clo}(j) &= f(sD_1^j, sD_2^j, \dots, sD_{n_j}^j) \\
C_{clo} &= k_0(c_s + c_b) + \sum_{j=1}^{m-1} \left[ k_1(c_s + c_b) + k_2 \sum_{i=1}^{n_j-1} (c_s + sD_i^j) + k_1 b_{clo}(j) \right] \\
&\quad + k_2 \sum_{i=1}^{n_m-1} (c_s + sD_i^m) + k_0 b_{clo}
\end{aligned}$$

## Berechnungskosten

$$\begin{aligned}
C'_{CS} &= k'_0 \sum_{j=1}^m \sum_{i=1}^n (s' + b') D_i^j \\
b'_{mig}(i, j) &= \begin{cases} f'(sD_i^j + b_{mig}(i-1, j)) & i, j > 1 \\ f'(sD_i^j + b_{mig}(n, j-1)) & i = 1, j > 1 \\ f'(sD_i^j) & i, j = 1 \end{cases} \\
C'_{mig} &= \sum_{j=1}^m \left[ \sum_{i=1}^{n-1} k'_2 (s' D_i^j + b'_{mig}(i, j)) + k_1 (s' D_i^j + b'_{mig}(n, j)) \right] \\
b'_{clo} &= f'(b_{clo}(1), b_{clo}(2), \dots, b_{clo}(m)) \\
b'_{clo}(j) &= f'(sD_1^j, sD_2^j, \dots, sD_{n_j}^j) \\
C'_{clo} &= \sum_{j=1}^{m-1} \left[ k'_1 s' D_1^j + \sum_{i=1}^{n_j-1} k'_2 s' D_i^j + k'_1 b'_{clo}(j) \right] + k'_1 s' D_1^m + \sum_{i=1}^{n_j-1} k'_2 s' D_i^m + k'_1 (b'_{clo}(m) + b'_{clo})
\end{aligned}$$

### 2.3.6 Spezialfälle

Neben den vorgestellten unterschiedlichen Szenarien für die Datenverteilung und Auswertungsmodalitäten können durch geeignete Parameterwahl noch weitere Fälle im Rahmen des Kostenmodells erfaßt werden. Exemplarisch wird dies für die in den einleitenden Bemerkungen zunächst ausgenommenen nicht-akkumulierenden Auswertungsfunktionen und die Auswahl einmaliger Datenelemente gezeigt.

**Nicht-akkumulierende Auswertung** Eine häufig betrachtete Anwendungsaufgabe für mobile Agenten im Kontext des elektronischen Handels besteht in der Ermittlung bestimmter

Kennzahlen, etwa eines Minimalpreises oder einer maximalen Qualität, für ein Wirtschaftsgut bei verschiedenen Anbietern. Der hierfür im Agenten benötigte Datenbestand beschränkt sich auf den Wert der Kennzahl und einen Verweis auf den Ort, an dem der aktuelle Extremwert vorgefunden wurde, ist also in seinem Volumen im wesentlichen konstant.

Dieser Fall kann im vorliegenden Modell durch das Szenario der replizierten Daten mit einem globalen Replikationsgrad von  $R = 1$  und einem lokalen Replikationsgrad von  $r = 0$  nachgebildet werden. Dadurch ergibt sich in jedem Integrationsschritt ein konstantes Datenvolumen von  $D$ . Da das Modell nur das jeweilige Volumen und nicht die konkreten Dateninhalte betrachtet, wird das Verhalten dadurch korrekt wiedergegeben.

Im Ergebnis sind in diesem Fall die Übertragungskosten für Migrations- und Cloning-Ansatz im Nutzdatenanteil exakt gleich, der Cloning-Ansatz besitzt nur den geringen Vorteil, daß der integrationsorientierte Codeanteil  $c_b$  innerhalb der lokalen Inseln nicht mit übertragen wird.

**Auslöschende Daten** Eine weitere mögliche Anforderung der Anwendung kann darin bestehen, nur einmalig auftretende Daten zurückzusenden und Verdopplungen auszusondern. Dieser Fall wird durch das Szenario der paarweise redundanten Daten abgedeckt, wobei hier als Redundanzfaktor mit  $r = 2r'$  bzw.  $R = 2R'$  der doppelte Wert der tatsächlich vorhandenen Redundanz  $r', R'$  angesetzt wird. In der Konsequenz führt dies zur Entfernung beider redundanter Datenelemente in der Integrationsfunktion und erzielt so das gewünschte Ergebnis.

Der Vergleich der Ansätze in diesem Szenario entspricht im Ergebnis den oben angestellten Betrachtungen für redundante Daten.

### 2.3.7 Gemeinsame Eigenschaften

Bei Betrachtung der vereinfachten Funktionen ist zu erkennen, daß die Kosten für den Cloning-Ansatz in allen drei Modellen für die Datenverteilung in den Termen  $k_1$  und  $k_2$  echt kleiner sind als für den Migrationansatz und auf der Zugangsverbindung (im Term  $k_0$ ) gleich sind. Die aufgeführten Grafiken bestätigen diese Erkenntnis. Ein wichtiges Ergebnis besteht darin, daß die Migrationskosten mit hinreichend großer Serverpopulation auch bei erheblichem Selektionspotential die Kosten für den Client-Server-Ansatz übersteigen. Dies schränkt die Verwendbarkeit migrierender Agenten für akkumulierende Anwendungsfunktionen auf wenige zu besuchende Server ein.

Bei den Berechnungskosten ist prinzipiell derselbe Trend zu beobachten, was angesichts der von der Datengröße abhängigen Wahl der Kostenfunktion nicht überrascht.

## 2.4 Grenzwertbetrachtung

Eine wesentliche Vereinfachung der vorstehenden analytischen Betrachtung gegenüber der Realität besteht in der Annahme einer uniformen Verteilung der Datenvolumina auf die besuchten Server. Während die Betrachtung beliebiger Verteilungen in der analytischen Auswertung nicht möglich ist, sollen doch die Grenzfälle inhomogener Datenverteilung kurz diskutiert werden.

Der Extremfall bezüglich der Datenanordnung tritt offensichtlich auf, wenn sämtliche relevante Daten auf genau einem Server (o.B.d.A. in Insel  $x$  auf Server  $y$ ) vorgefunden werden und die Selektion auf allen übrigen Servern keine anwendungsrelevanten Daten entdeckt. Betrachtungen über Redundanz und Replikation und die integrationsbezogenen Berechnungskosten sind in diesem Fall gegenstandslos.

Eine naheliegende Optimierung für dieses Szenario besteht darin, nach Erhalt der Daten von

einem Server auf die Nachfrage bei weiteren Servern zu verzichten. Hier soll jedoch vorausgesetzt werden, daß die Anwendung a priori keine Kenntnis über den Gesamtumfang und die topologische Verteilung der Daten hat, so daß sie grundsätzlich alle Server kontaktieren muß. Für den Client–Server–Ansatz ergeben sich, wie bisher stets, Kosten in Höhe von

$$C_{CS} = \sum_{j=1}^m \sum_{i=1}^n k_0 D_i^j = k_0 mnD$$

Die in diesem Szenario anfallenden Kosten für agentenbasierte Verarbeitung setzen sich jeweils aus einem Anteil für die Codeübertragung, der von der Plazierung des datenträgenden Servers unabhängig bleibt, und einem Nutzdaten–bezogenen Anteil zusammen, der je nach Plazierung sehr stark variieren kann. Für den Cloning–Ansatz ergeben sich nach den weiter oben eingeführten Formeln Übertragungskosten von

$$C_{clo} = k_0(c_s+c_b) + \sum_{j=1}^{m-1} \left[ k_1(c_s+c_b) + k_2 \sum_{i=1}^{n_j-1} c_s \right] \\ + \begin{cases} k_0 s D_x^y & \text{falls Daten auf Zugangsserver} \\ (k_1 + k_0) s D_x^y & \text{falls Daten auf Backbone–Server} \\ (k_2 + k_1 + k_0) s D_x^y & \text{sonst} \end{cases}$$

mit einer geringen Variationsbreite für den datenbezogenen Teil zwischen  $k_0 s D$  und  $(k_0 + k_1 + k_2) s D$ . Für den Migrationsansatz mit Kosten

$$C_{mig} = k_0(c_s+c_b) + \sum_{j=1}^{m-1} \left[ k_1(c_s+c_b) + k_2 \sum_{i=1}^{n_j-1} (c_s+c_b) \right] \\ + k_2 \sum_{k=x}^{n_l-1} s D_x^y + \sum_{l=y+1}^m \left[ k_1 s D_x^y + k_2 \sum_{k=1}^{n_l-1} s D_x^y \right] + k_0 s D_x^y$$

ergibt sich hingegen in den datenbezogenen hinteren drei Termen eine erhebliche Abhängigkeit von der zufälligen Plazierung  $(x, y)$  des ausgezeichneten Servers innerhalb der topologischen Struktur und der dadurch bestimmten Position, an der er im Migrationspfad erreicht wird. Der günstigste Fall für den Cloning–Ansatz tritt auf, wenn der erste vom Heimatrechner kontaktierte Backbone–Rechner die gewünschten Daten enthält, und für den Migrationsansatz, wenn diese erst beim letzten besuchten Server gefunden werden. Es fallen dann für die Migration dieselben Nutzdaten–bezogenen Kosten  $k_0 s D$  an wie beim Cloning, und das Cloning hat nur leichte Vorteile bei der Codeübertragung, da wie oben schon erwähnt der Anteil  $c_b$  nicht an die Blätter der Clone–Hierarchie übertragen wird. Im für das Cloning ungünstigsten Fall, daß die Daten auf einem Blatt der Clone–Hierarchie vorliegen, wachsen die vom Datenanteil verursachten Kosten moderat auf  $(k_0 + k_1 + k_2) s D$ . Bei der Migration wächst dieser Kostenanteil jedoch im Mittel auf  $(k_0 + k_1 \frac{1}{2}(m-1) + k_2 \frac{1}{2}m(n-1)) s D$  und im ungünstigsten Fall, wenn die Daten bereits auf dem ersten besuchten Server gefunden und über den gesamten Migrationspfad mitgeführt werden müssen, bis auf  $(k_0 + k_1(m-1) + k_2 m(n-1)) s D$ . Somit ist der Cloning–Ansatz dem Migrationsansatz auch für den Fall extrem inhomogener Datenverteilung unter Kostenaspekten überlegen.

Abbildung 10 veranschaulicht den deutlichen Unterschied im Verlauf der Kostenfunktion für auf einem Server konzentrierte Daten abhängig von der Position  $x$  innerhalb der lokalen Insel  $y$ , die der Server in der willkürlichen Nummerierung einnimmt. Der Cloning–Ansatz zeigt sich hier sehr robust gegenüber der zufälligen Plazierung des ausgezeichneten Servers, im Migrationsansatz herrscht hingegen eine starke Abhängigkeit von der Position des Servers auf dem Migrationspfad.

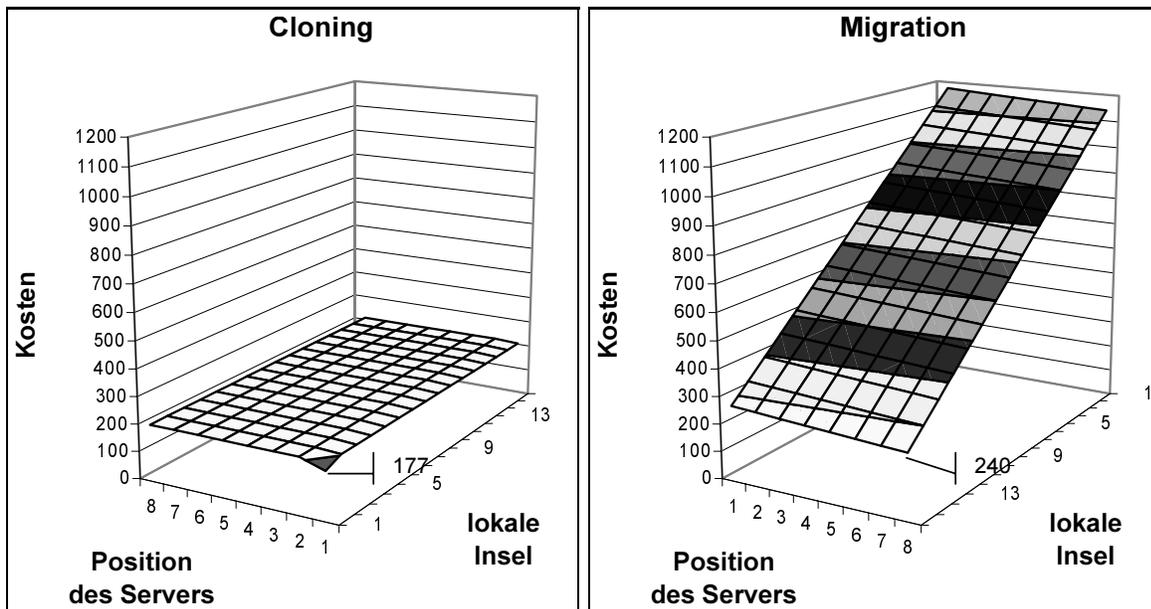


Abbildung 10: Übertragungskosten für Konzentration auf einem Server

### 3 Ergebnisse der Analyse

Als Resultat der analytischen Betrachtung der Kommunikations- und Berechnungskosten ergibt sich, daß für die betrachtete Anwendungscharakteristik einer Daten akkumulierenden Funktion die parallele Auswertung durch Clone-Agenten der herkömmlichen Vorgehensweise des einzelnen migrierenden Agenten weit überlegen ist. Sowohl für die summierten Kosten der Datenübertragung als auch beim berechnungsbezogenen Aufwand ergibt sich ein Vorteil für den Cloning-Ansatz. Mehr noch, während der herkömmliche Migrationsansatz bei hinreichender Anzahl von Zielserversn durch die Mitnahme akkumulierter Zwischenergebnisse schließlich die Kosten des Client-Server-Ansatzes übersteigt, kann sich das dynamische Cloning im betrachteten Kontext gegenüber Client-Server grundsätzlich durchsetzen. Dies bedeutet einen erheblichen Gewinn für die Anwendbarkeit des mobilen Agenten-Paradigmas insgesamt.

Ein interessantes Teilergebnis besteht in der Erkenntnis, daß auch die Möglichkeit des migrierenden Agenten, inhärente Redundanz in den Datenbeständen der verschiedenen Datenanbieter zur Reduktion des mitgeführten Zustands zu nutzen, den Nachteil der Mitführung über lange Strecken nicht aufwiegen kann.

Die beobachteten Tendenzen verstärken sich noch bei Verwendung eines auf Multicast basierenden Transportmechanismus. In diesem Fall reduzieren sich die konstanten Kostenfaktoren  $k_0, k_1, k_2$  für die Datenübertragung im Cloning-Ansatz weiter, da die Versendung einer Multicast-Nachricht an  $n$  Empfänger deutlich günstiger ist als die Versendung von  $n$  Unicast-Nachrichten. Die im Cloning-Ansatz der autonomen Agenten anfallenden Kommunikationsmuster eignen sich grundsätzlich für eine Multicast-Unterstützung, wie in [HH97] näher erläutert wird.

Als Fazit ergibt sich, daß dynamisches Cloning eine sinnvolle Erweiterung der Mechanismen für mobile Agentensysteme darstellt. Ein vollständiger Ersatz der Migration durch Cloning ist jedoch nicht möglich, da nicht alle Anwendungen hinreichend parallelisierbar sind und z.B. im Falle enger Datenabhängigkeiten das Modell der parallel delegierten Clones nicht angewandt werden kann.

## Literatur

- [CGH95] Colin G. Harrison and David M. Chess and Aaron Kershenbaum: *Mobile Agents: Are they a good idea?*, Technical Report RC19987, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598, März 1995.
- [HH97] Jörn Hartroth und Markus Hofmann: *Using IP multicast to improve communication in large scale mobile agent systems*, Angenommen bei der 31st Hawaiian International Conference on System Sciences (HICSS'31), Hawaii, Januar 1998.
- [Kau97] Reinhold Kautzleben: *Analyse alternativer Mobilitätskonzepte für agentenbasierte verteilte Anwendungen*, Diplomarbeit, Institut für Telematik, Universität Karlsruhe, August 1997.
- [Whi96] James E. White: *Mobile Agents*, In: Jeffrey Bradshaw (ed.): *Software Agents*, AAAI Press, Menlo Park, California, 1996.