

S.T.A.R.T.

Simple
Tool for
Application
Runs and
Transfer of Data

Marcus Hardt, Ahmad Hammad
SCC @ KIT

Old Name

- **Simple Helper for Installation and Transfer of data**

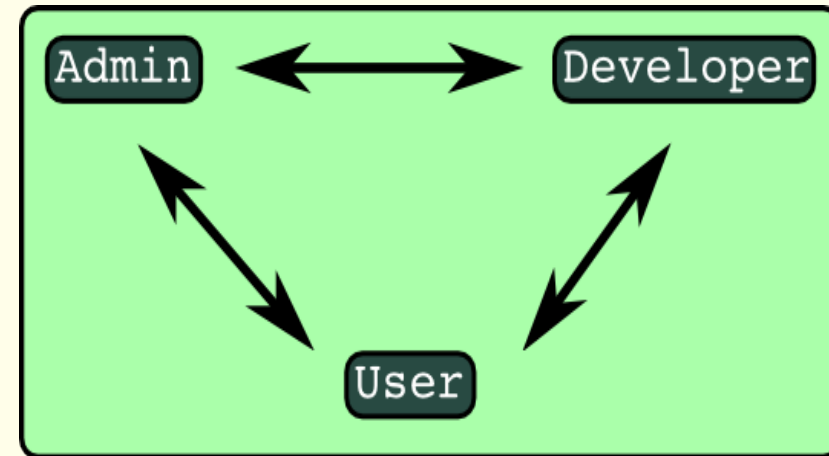
From Workstation to the Grid

- Roles:
 - **Admin:** Install computers (Apps, Libs, Network, ...)
 - **Developer:** Edit source code, compile, versioning
 - **User:** Compile, run, production, scientific results
- How do developer admin and user relate?

From Workstation to the Grid

- Workstation

- User = Developer = Admin
=> No conflicts
=> Problem: Too slow





- Cluster / Supercomputer

- Developer provides sourcecode

- User requests Admin to install

- Libraries **Iterative procedure**
~ 1 week
- Compiler **Code updates one request**
~ few hours
- Sourcecode

From Workstation to the Grid

- Grid: **One** Admin => **hundreds** of Admins
 - User cannot ask **hi**  for an update
 - Developer does not  which libs are installed
 - Nobody knows **which version** runs **on which site**

Don't worry!

SA1 is there to help

The use case

- Developer or User modifies code
- For testing he needs the grid
 - Local testing has no MPI
 - Local testing is too slow
 - Developer has to use Windows
 - ...
- Code changes about 10-50 times per day
 - => Productivity increases with deployment speed



How can we help him?

- Simple jdl Int

- Scriptable

- RAS/MD inte

General form:

```
<proto>://<host>/<file>
```

```
with: proto = [gsiftp|http|ftp|sftp|lfn|file]
```

```
Executable      = "s.t.a.r.t.sh";
Arguments       = "--software http://cvs.fzk.de/autobuild/eirene.tar.gz
                 --executable eirene
                 --input      http://cvs.fzk.de/~marcus/my_input.txt
                 --input      ftp://euforia.eu/iter/geometry.dat
                 --output     ftp://ftp.euforia.eu/output.tar.gz
                 --libs       http://cvs.fzk.de/euforia/libs/
                 --prepare    lfn://prepare.sh
                 --postpare   file:post.sh
                 -- -parameter=value --key=value2";
StdOutput       = "std.out";
StdError        = "std.err";
InputSandbox    = {"start.sh", "post.sh"};
OutputSandbox   = {"output.tar.gz"};
```

Some word to the admins

- I know you think it's evil stuff! ;-)
- But:
 - It's nothing users couldn't do themselves anyways
 - The idea is to use s.th. like an `INSTALL_PATH`
 - `s.t.a.r.t.sh` checks all places in that path for...
 - ...existing software.tar.gz => No download, if up-to-date
 - ...permission to write => Only write somewhere if allowed
 - **`INSTALL_PATH=$VO_EUFORIA_SW_DIR:HOMES:/tmp`**
 - Libraries can be handled separately, if need be

Autobuild

- What?
 - Retrieve code from repository (CVS|SVN|File)
 - Run autobuild.sh (to be provided by developer)
 - Publish output on webpage
- Why?
 - Standardised build environment
 - **Only one** admin
 - Clear versioning



EUFORIA -- Continuous Automatic Builder (continuous)



Modules by name *Modules by group* Modules by repository ISO images

Builder status
Overall status: Good
Date: Wed Dec 17 2008
Start Time: 14:55:02 UTC 15:55:02 CET
Cycle Duration: 00h 00m 32s
Build Counter: 1229525702
Build Timestamp: 1229525702

Module groups

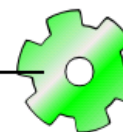
Software

EIRENE-Grid-parallel	success
EIRENE-Grid-sequential	success
EIRENE-Grid-trilinhex	success

Powered by Test-AutoBuild, managed by Build Administrator

[1.0](#) feed for all modules, or failed modules

EUFORIA -- Continuous Automatic Builder (continuous)



Modules by name Modules by group Modules by repository ISO

images Builder status

Overall status: success
Date: Wed Dec 17 2008
Start Time: 14:57:02 UTC 15:57:02 CET
Cycle Duration: 00h 00m 33s
Build Counter: 1229525822
Build Timestamp: 1229525822

Module: EIRENE-Grid-trilinhex

SCM checkout status: success
Build status: cached
Logs: SCM checkout (670 b) Build output (35.00 KB)
SCM checkout time: 00h 00m 09s
Build time: 00h 01m 13s
Links: EIRENE Bugtracker EIRENE SVN
Artifacts: README

Generated packages

Packages: Scientific Linux SL release 4.5 (Beryllium)

Filename	Size	MD5 Sum
eirene-trilinhex-ver-144.tar.gz	6.12 MB	2b3cbb1ff18bce27ec40d225fcad6bbf
eirene-trilinhex-ver-144.zip	6.12 MB	f6562be040228452b253bc6d2b260d45

Changes since last build

Changelist	User	Date
144	root	Tue Dec 9 2008 17:57:08 UTC

Description: Just taken from grid-parallel branch and adjusted the name of the branch

Files: A /branches/eirene-trilinhex/autobuild.sh

Summary

- Using the grid is complicated
- SA1 is there to help
- We develop tools to help users
- Just talk to us **AND** use the infrastructure

Additional services

- Savannah
 - Couple of users
 - Codes in SVN
 - Working fine
- Autobuild
 - Only used for Eirene at the moment
 - No great community pick-up
 - To be integrated for grid jobs

Grid – How to use -> Courses

- Participate in courses
 - e.g. GridKa School 31.8.-4.9.2009.
- Read Material of passed courses
 - Available online => Adrian knows where
- Read the **User Induction** page in Euforia WIKI
<http://wiki.eu-euforia.eu>

Grid – How to use -> Overview

1. Get a certificate

- At your country (of work) Certification Authority
- **Contact Marcus Hardt or Isabel Campos** if you need help

2. Get access to a **U**ser **I**nterface (UI)

- UI provided by a grid site in your country
- If not, **contact Isabel Campos or Marcus Hardt** for an account at one of the two fallback Uis

3. Contact us for supporting your Application on grid

- We have local experts that can really help

Grid – How to use -> Authentication

- X.509 Certificate
 - To be obtained at the country of the workplace
- Join a VO
 - Virtual Organisation EUFORIA
 - => <https://i2g-voms.lip.pt:8080/vomses>
- Account on a User Interface computer (UI)
 - Either at a grid-site in your country or at one of the fallback UIs either in Santander or Karlsruhe.
- Read the **User Induction** page in Euforia WIKI
<http://wiki.eu-euforia.eu>

Grid – How to use -> Low level job

- Basic job execution

- Write a .jdl-file

- `i2g-job-submit <.jdl-file>`
=> Returns job-service URL

- `i2g-job-submit <service URL>`

- `i2g-job-submit <service URL>`

- Yes, this is a bit painful

- => We suggest this for **development**
and **early integration**

Contacts: Isabel Campos, Marcus Hardt

```
Type = "job";
JobType = "parallel";
SubJobType = "openmpi";
NodeNumber = 12;
Executable = "test.sh";
Arguments = "-t 2h";
StdOutput = "std.out";
StdError = "std.out";
InputSandbox = {"sleeper.sh"};
OutputSandbox = {"stdout", "stderr"};
```

Summing up important stuff

- Contacts:
 - In any case of trouble or question:
 - Isabel Campos <isabel@campos-it.es>
 - Marcus Hardt <marcus@hardt-it.de>
 - If related to a GUI, additionally you can ask:
 - Marcin Plociennik <marcinp@man.poznan.pl>
- Links:
 - **<http://wiki.eu-euforia.eu>**
=> User Induction + User Guide (extensive)