Ganga and DIANE
User Tools for EGI Applications

Dan van der Ster, CERN IT-ES

EGI Technical Forum 2010

16 September 2010

Overview

- Motivation for Application Tools
- Ganga and DIANE: Overview of these tools and recent developments
- Example users: heavy and small user communities
- How to get started…

How to Enable the Grid Application Builders and End-users?

- The Grid fabric can be large and complex
- We have a stable and mature middleware, but taking full advantage of it can require specialized knowledge
- We therefore have a need for tools to enable both end-users and application builders

- This talk is about two such tools from the EGEE RESPECT programme: **Ganga** and **DIANE**

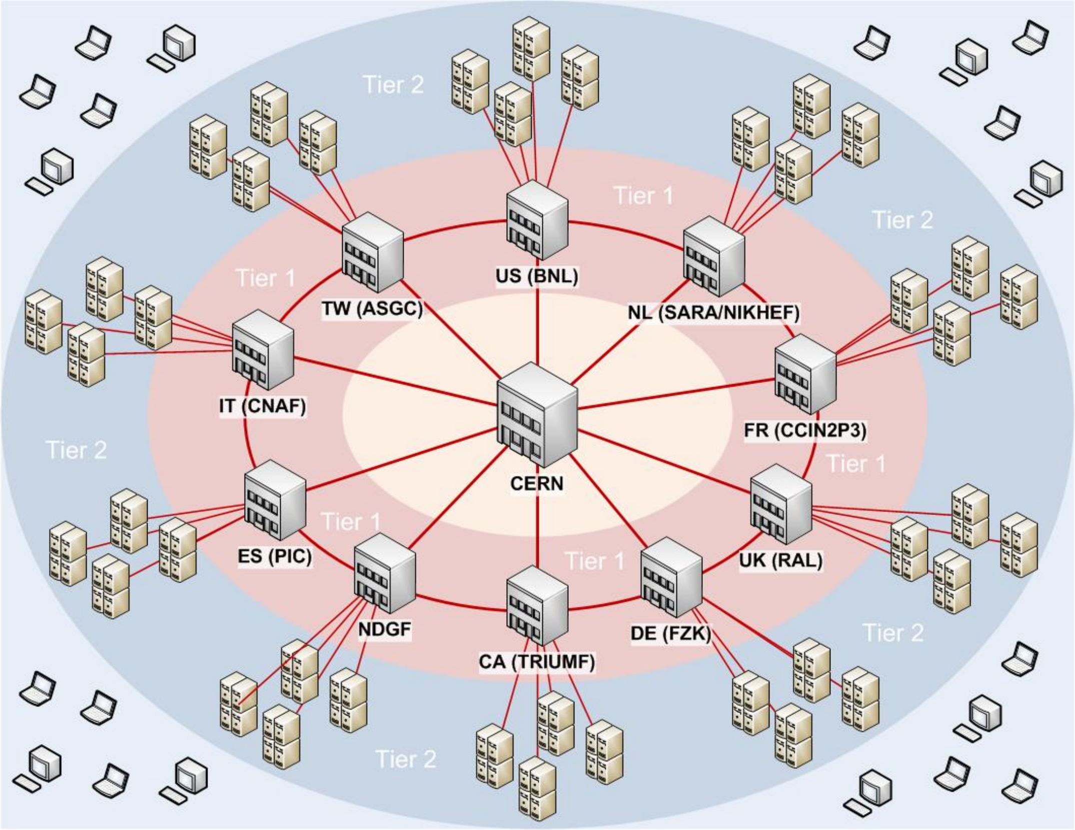
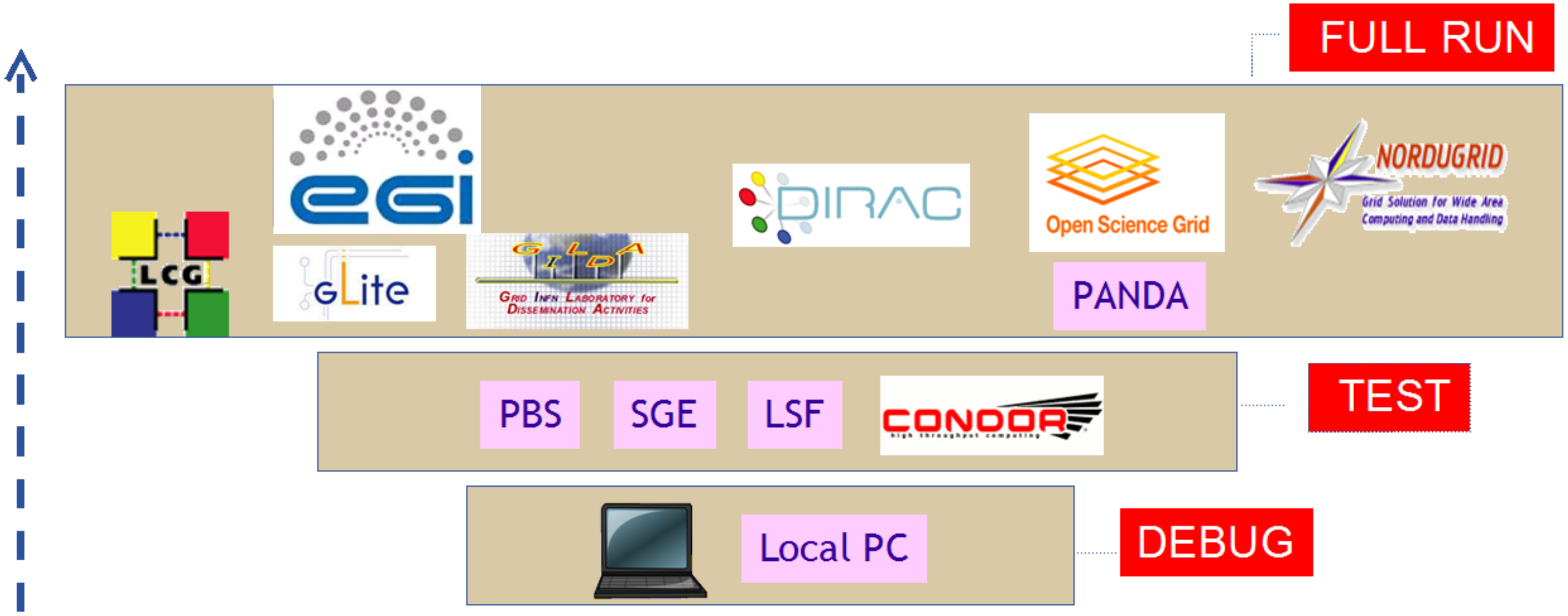


Figure: Structure of the ATLAS Grid Resources

Motivation for a Grid Front-end

- Users want:
  - Development on the laptop; full analysis on “The Grid™”.
  - To get results quickly, utilizing all of the resources available, wherever they are.
  - A familiar and consistent user interface to all of the resources.
- Users don’t want:
  - To know the details of the grids or the resources.
  - To learn yet another tool in order to access some resources
  - To have to reconfigure their application to run on different resources.


FULL RUN
LCG
EGI
gLite
GILDA
Grid INFN Laboratory for Dissemination Activities
DIRAC
Open Science Grid
PANDA
NORDUGRID
Grid Solution for Wide Area Computing and Data Handling
TEST
PBS
SGE
LSF
CONDOR
high throughput computing
DEBUG
Local PC

“configure once, run anywhere”

Introduction to Ganga

- Ganga is a job management tool for end-users
  - Run locally, on batch, and on The Grid.
  - Job mgmt: Repository, Template, Copy, Retry, Kill
- Ganga is a powerful tool for developing Grid Applications
  - Python API exposes all Ganga functionality
  - Plugin architecture to add new applications, backends, or generic features
- Ganga is used actively in by the heavy users LHCb and ATLAS and by other VOs
  - Experiment-specific plugins are included.
- Ganga is an open source community-driven project:
  - Core development is joint between the heavy users
  - Mature and stable, with an organized development process
  - Developer community meets regularly: upcoming developer days in Munich later this month.

Ganga User Interfaces

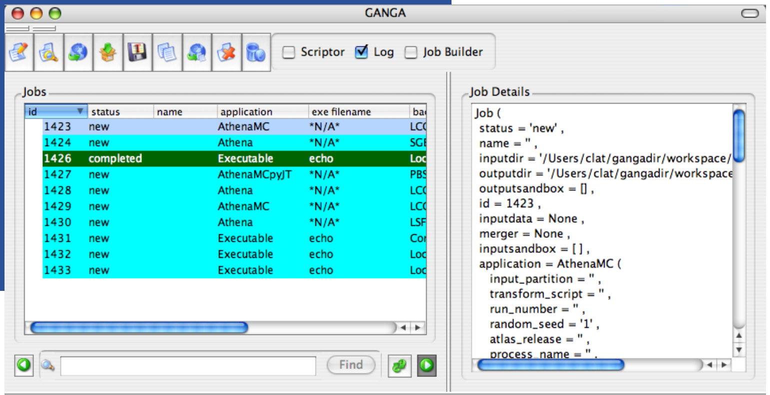
- Choose your own interface: CLI, GUI, or Scripting.

```
*** Welcome to Ganga ***
Version: Ganga-4-2-8
Documentation and support: http://cern.ch/ganga
Type help() or help('index') for online help.

In [1]: jobs
Out[1]: Statistics: 1  jobs
--------------
#   id      status        name   subjobs      application
           backend.actualCE
#    1   completed                             Executable
compute.hpc.unimelb.edu.au:2119/jobmanage
```

To start Ganga:

```
> ganga
```



id	status	name	application	exe filename	ba
1423	new		AthenaMC	*N/A*	LC
1424	new		Athena	*N/A*	SGI
1426	completed		Executable	echo	Loc
1427	new		AthenaMCpyJT	*N/A*	PBS
1428	new		Athena	*N/A*	LC(
1429	new		AthenaMC	*N/A*	LC(
1430	new		Athena	*N/A*	LSF
1431	new		Executable	echo	Cor
1432	new		Executable	echo	Loc
1433	new		Executable	echo	Loc

```
Job (
 status = 'new' ,
 name = '' ,
 inputdir = '/Users/clat/gangadir/workspace/
 outputdir = '/Users/clat/gangadir/workspace
 outputsandbox = [] ,
 id = 1423 ,
 inputdata = None ,
 merger = None ,
 inputsandbox = [ ] ,
 application = AthenaMC (
   input_partition = '' ,
   transform_script = '' ,
   run_number = '' ,
   random_seed = '1' ,
   atlas_release = '' ,
   process_name = '' ,
```

To start the GUI:

```
> ganga --gui
```

```
#!/usr/bin/env ganga
#-*-python-*-
import time
j = Job()
j.backend = LCG()
j.submit()
while not j.status in [‘completed’,’failed’]:
    print(‘job still running’)
    time.sleep(30)
```

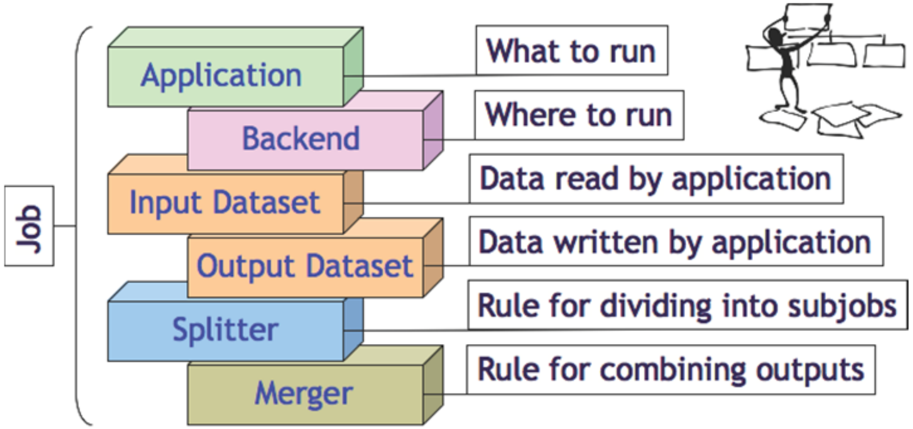
```
./myjob.exec

ganga ./myjob.exec

In [1]:execfile(“myjob.exec”)
```

Ganga Job Management

What's in a Ganga Job?



Run the default job locally:

```
Job().submit()
```

Default job on the EGEE grid:

```
Job(backend=LCG()).submit()
```

Listing of the existing jobs:

```
jobs
```

Get help (e.g. on a job):

```
help(jobs)
```

Display the nth job:

```
jobs(n)
```

Copy and resubmit the nth job:

```
jobs(n).copy().submit()
```

Copy and submit to another grid:

```
j=jobs(n).copy()
j.backend=DIRAC()
j.submit()
```

Kill and remove the nth job:

```
job(n).kill()
job(n).remove()
```

Job splitting

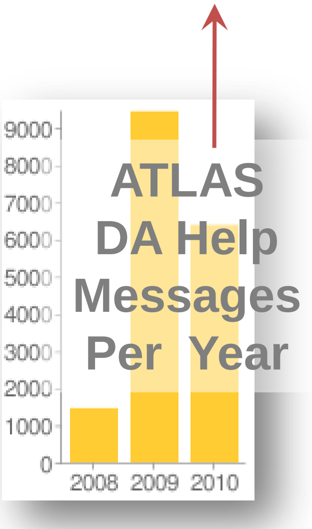
```
j=Job(splitter=ArgSplitter([[str(i)] for i in xrange(10)])).submit()
```

Recent Ganga Developments

- High performance job repository
  - Lazy loading for fast startup time
  - XML files to improved reliability and error recovery

- New Monitoring plugins
  - ActiveMQ-based monitoring plugins for general job monitoring and for experiment plugins (e.g. ATLAS)

- Error Reporting Tool
  - Interactive command to send a job error report to a user support team
  - Bundles job and environment info and uploads to a secure server



DIANE Overview

- DIANE is a framework for efficient control and scheduling of computations on a set of distributed worker nodes. It allows users to:
  - reduce the application execution time by using the resources more efficiently,
  - reduce the user work overhead by providing fully automatic execution and failure management,
  - efficiently integrate local and Grid resources.

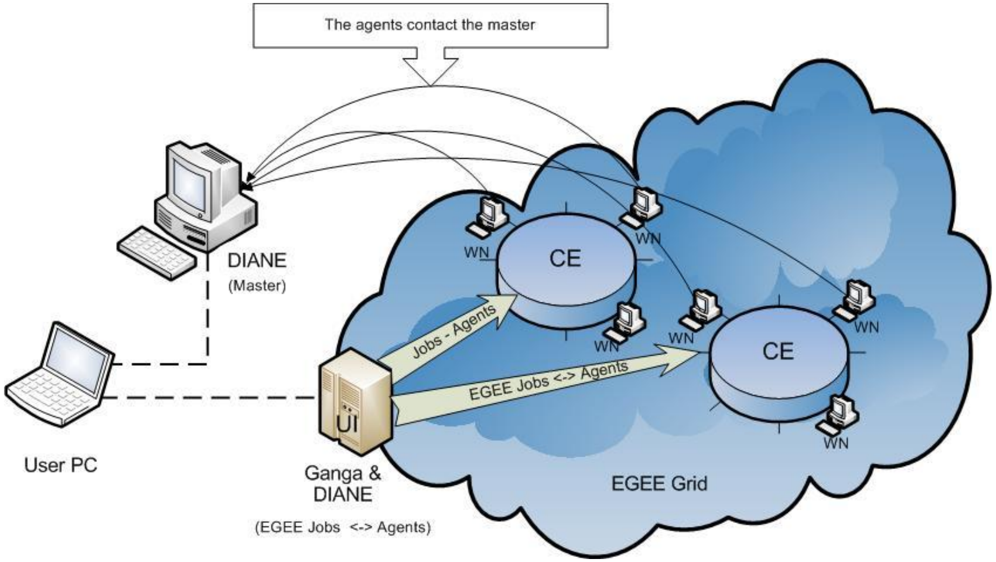
- DIANE improves the reliability and efficiency of job execution by providing automatic **load balancing, fine-grained scheduling** and **failure recovery**.

- Project Leader: Jakub Moscicki Web: *http://cern.ch/DIANE*

DIANE Overview cont'd

- DIANE operates at the user-level via a master and worker agents:
  - Master hosts the work units
  - Agents (pilots) are submitted via Ganga to any backend (Local, LSF, PBS, SGE, Condor, LCG (gLite WMS), etc…)

- Pluggable schedulers:
  - Embarassingly parallel tasks supported out of the box
  - DAG and workflow plugins available (see DAG4DIANE, MOTEUR)

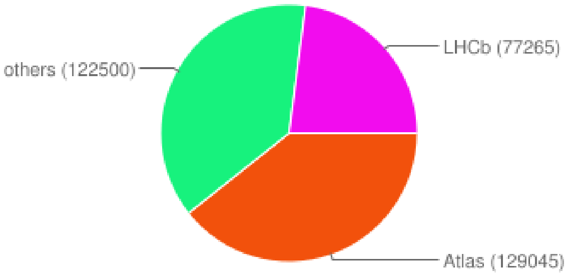


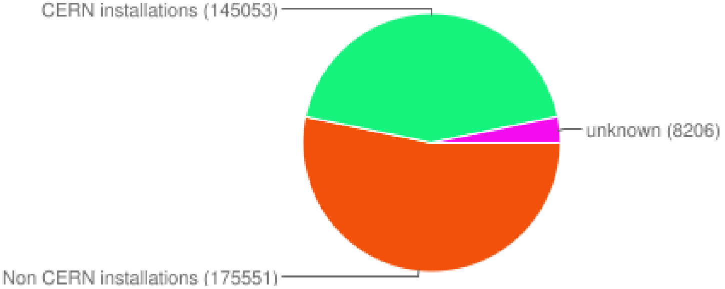
The Ganga/DIANE Dashboard

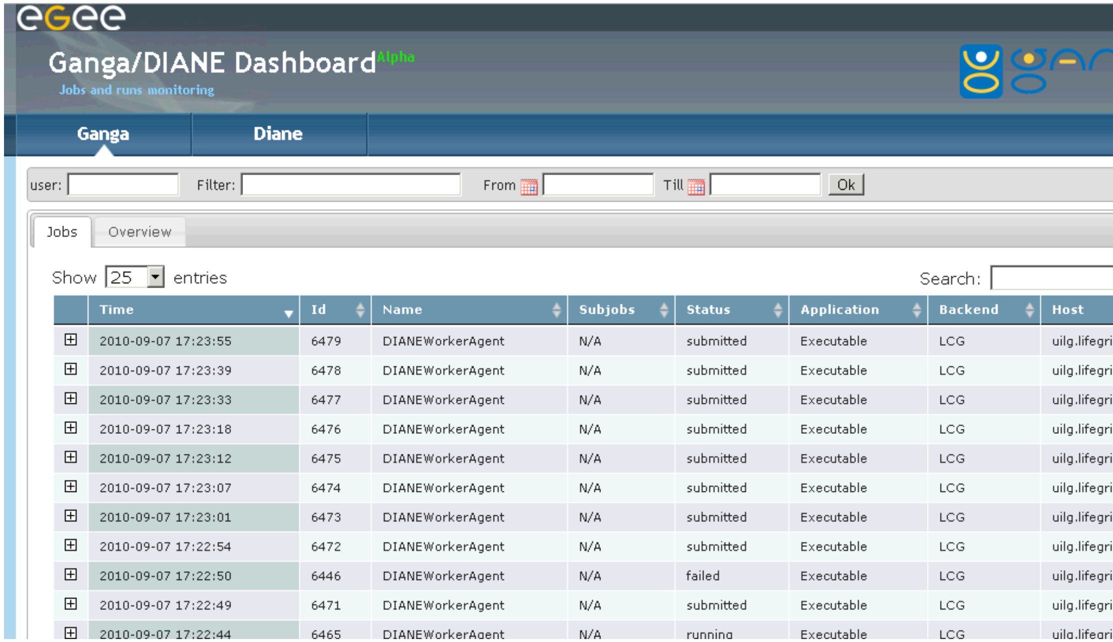
- New system and interface for usage stats and also a new web job dashboard (alpha prototype)
- Available online at http://gangamon.cern.ch/

Ganga Usage Stats in 2010 (until Sept 7)

Total number sessions: 328810
Number unique users: 1242
Number of sites: 121

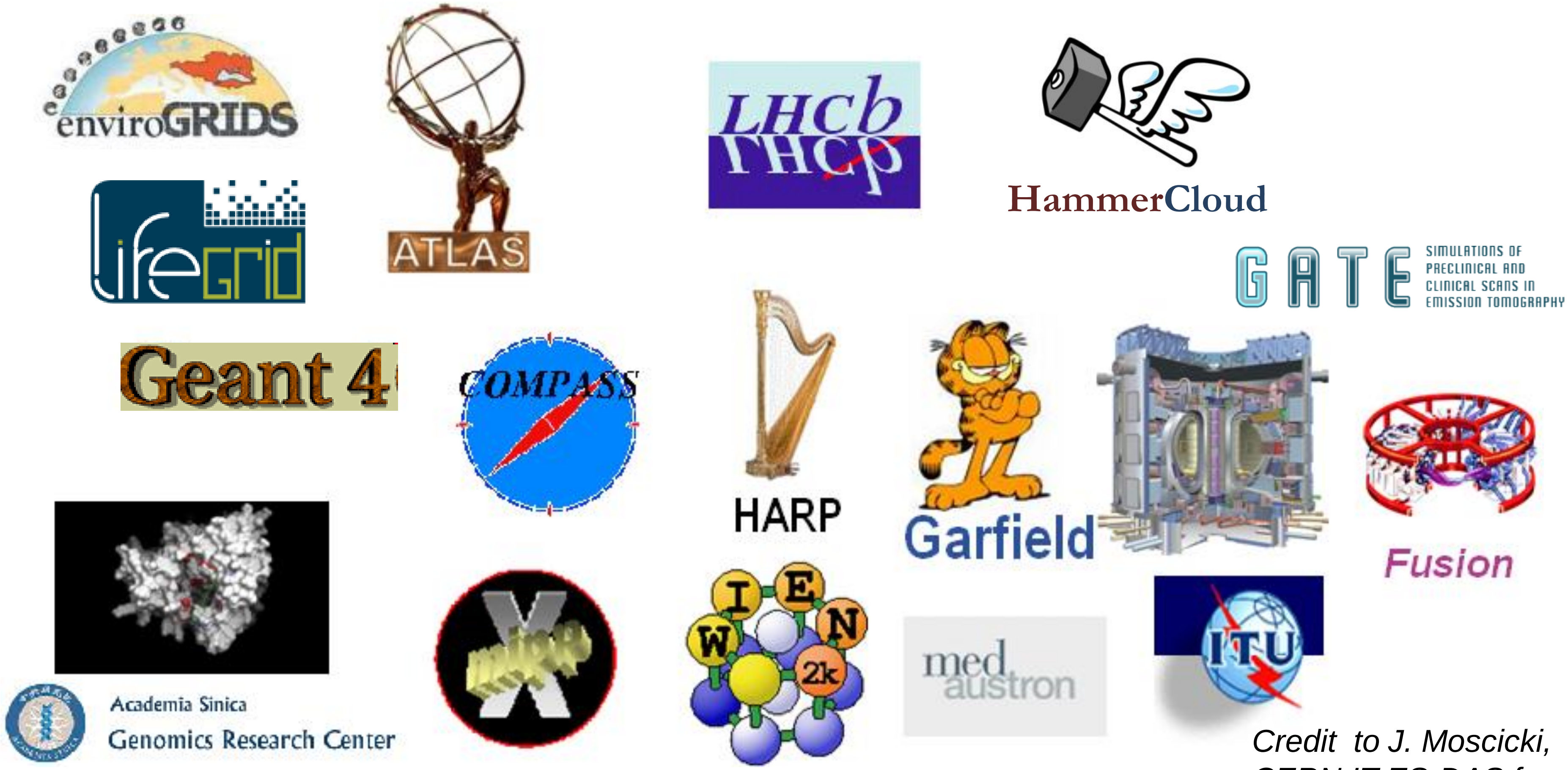






	Time	Id	Name	Subjobs	Status	Application	Backend	Host
⊞	2010-09-07 17:23:55	6479	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:23:39	6478	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:23:33	6477	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:23:18	6476	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:23:12	6475	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:23:07	6474	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:23:01	6473	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:22:54	6472	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:22:50	6446	DIANEWorkerAgent	N/A	failed	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:22:49	6471	DIANEWorkerAgent	N/A	submitted	Executable	LCG	uilg.lifegri
⊞	2010-09-07 17:22:44	6465	DIANEWorkerAgent	N/A	running	Executable	LCG	uilg.lifegri

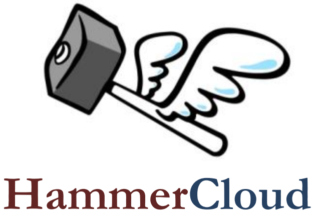
Ganga and DIANE Users

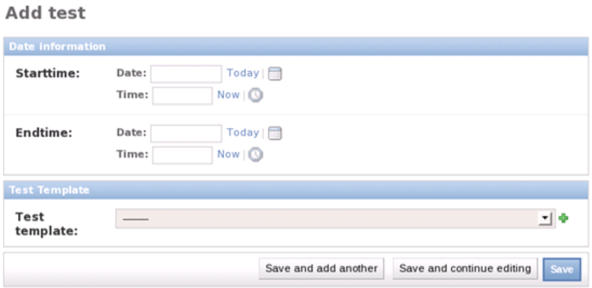


Credit to J. Moscicki, CERN IT-ES-DAS for most of the following slides…

HammerCloud: DA Testing Service

- HammerCloud (HC) is a distributed analysis testing system built around Ganga
  - Developed initially with ATLAS; now being extended to support CMS and LHCb as well
- HC tests:
  - frequent functional tests to validate the services
  - on-demand stress tests to commission new sites or give benchmarks for site comparisons
- Ex: ATLAS has invested more that 200,000 CPU-days of HC testing since late 2008




Add test

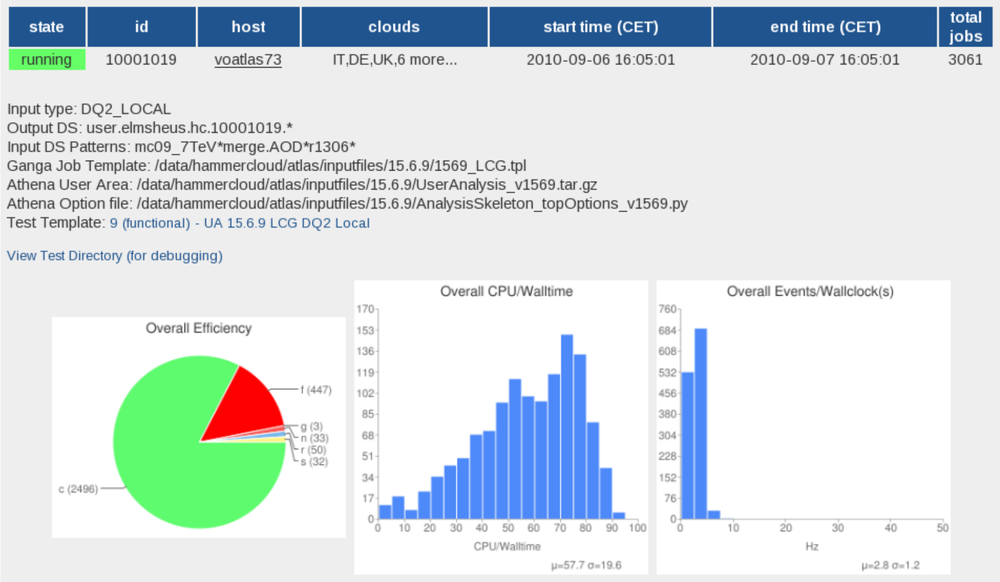
Date information

Starttime: Date: Today | Time: Now |

Endtime: Date: Today | Time: Now |

Test Template

Test template: ——



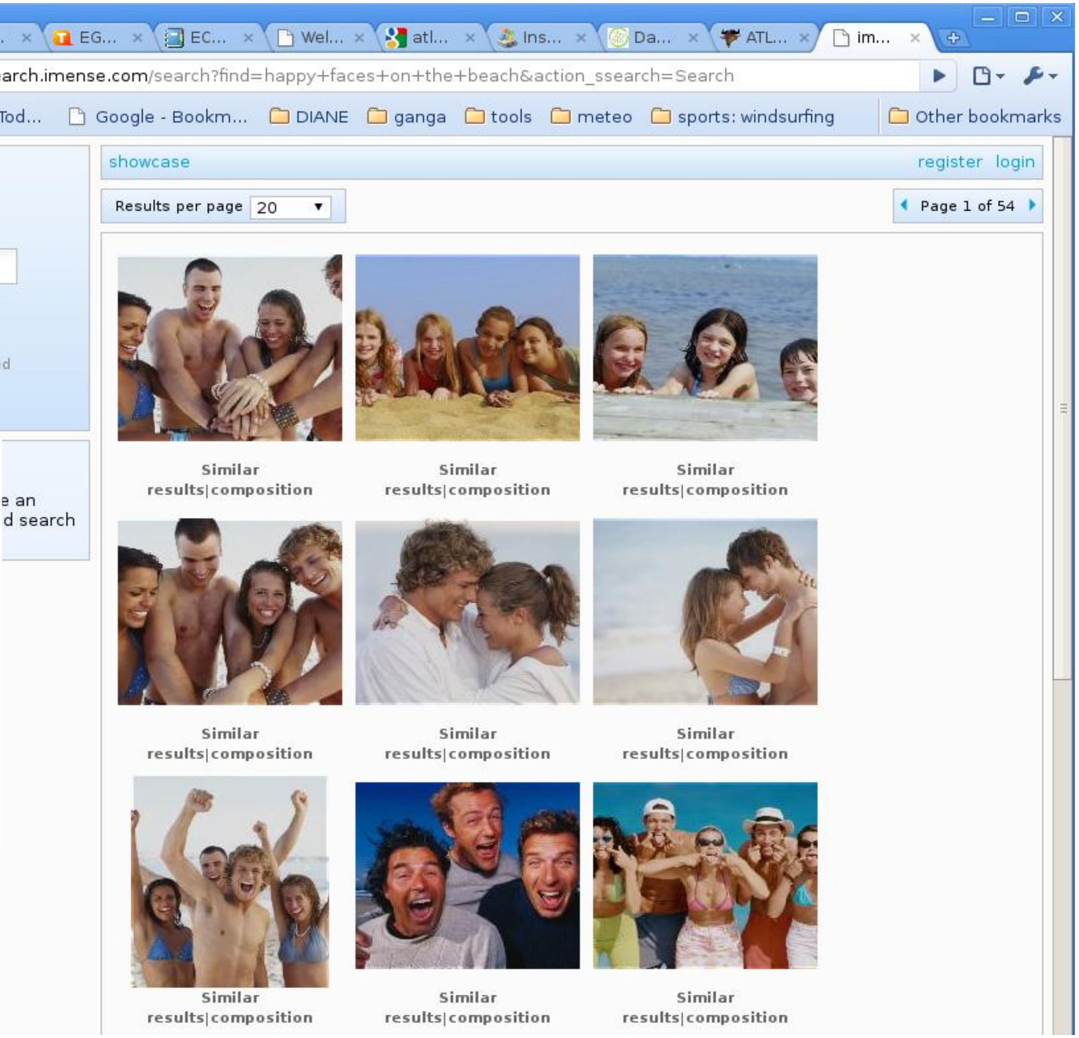
state	id	host	clouds	start time (CET)	end time (CET)	total jobs
running	10001019	voatlas73	IT,DE,UK,6 more...	2010-09-06 16:05:01	2010-09-07 16:05:01	3061

Input type: DQ2_LOCAL
Output DS: user.elmsheus.hc.10001019.*
Input DS Patterns: mc09_7TeV*merge.AOD*r1306*
Ganga Job Template: /data/hammercloud/atlas/inputfiles/15.6.9/1569_LCG.tpl
Athena User Area: /data/hammercloud/atlas/inputfiles/15.6.9/UserAnalysis_v1569.tar.gz
Athena Option file: /data/hammercloud/atlas/inputfiles/15.6.9/AnalysisSkeleton_topOptions_v1569.py
Test Template: 9 (functional) - UA 15.6.9 LCG DQ2 Local

View Test Directory (for debugging)

Poster #23: **The HammerCloud Distributed Analysis Testing Service**

imense.com
Image Classification for Search



Gri... Ani... CD... EG... EC... Wel... atl... Ins... Da... ATL... im...

http://picturesearch.imense.com/search?find=happy+faces+on+the+beach&action_ssearch=Search

Pyhton LibRef | Delegate to Tod... | Google - Bookm... | DIANE | ganga | tools | meteo | sports: windsurfing | Other bookmarks

imense picturesearch

happy faces on the beach

Search | Advanced options

□ Model released □ Rights managed

□ Property released □ Royalty free

e an
d search

showcase | register login

Results per page 20 | Page 1 of 54

Similar results|composition

Similar results|composition

Similar results|composition

Similar results|composition

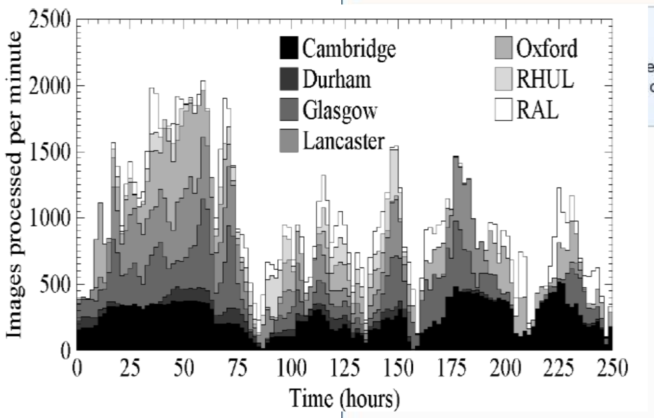
Similar results|composition

Similar results|composition

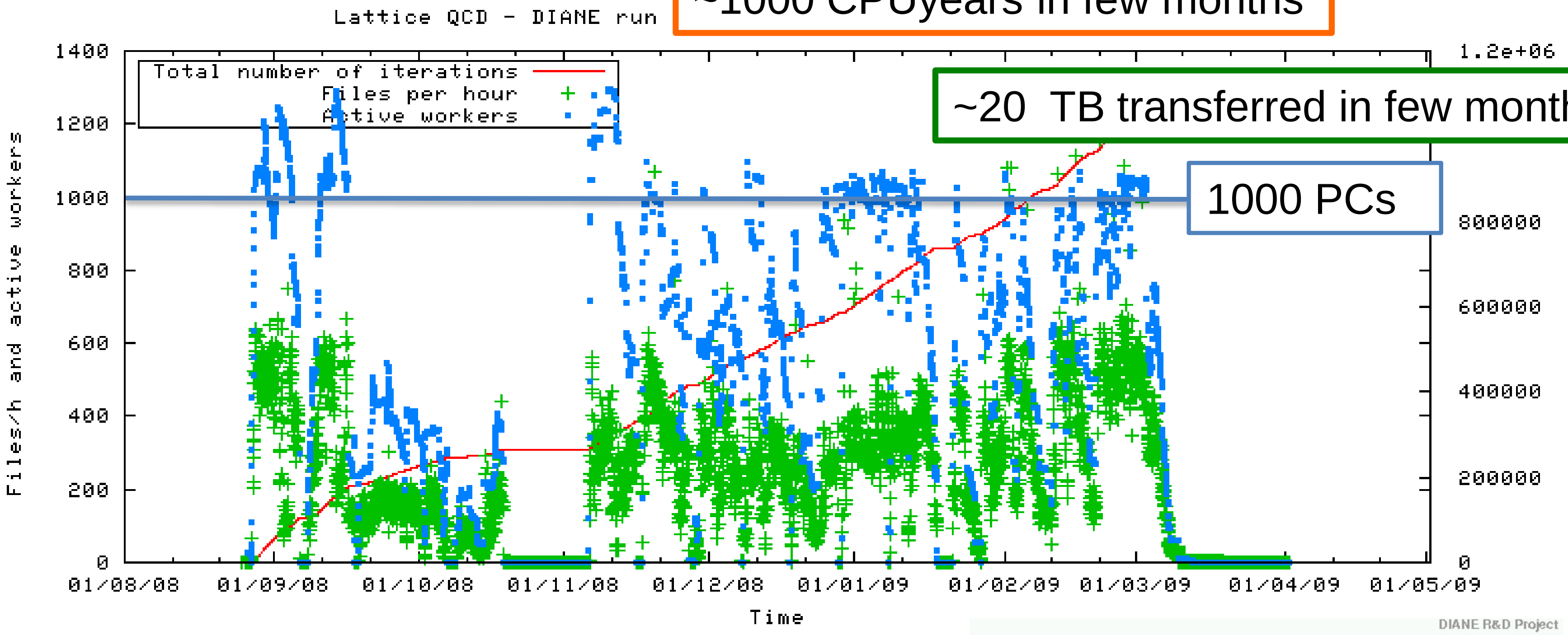
Similar results|composition

Similar results|composition

Similar results|composition



QCD Thermodynamics



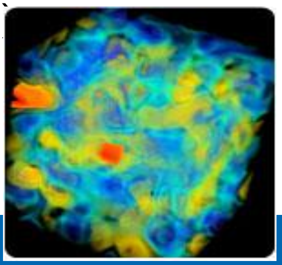
~1000 CPUyears in few months

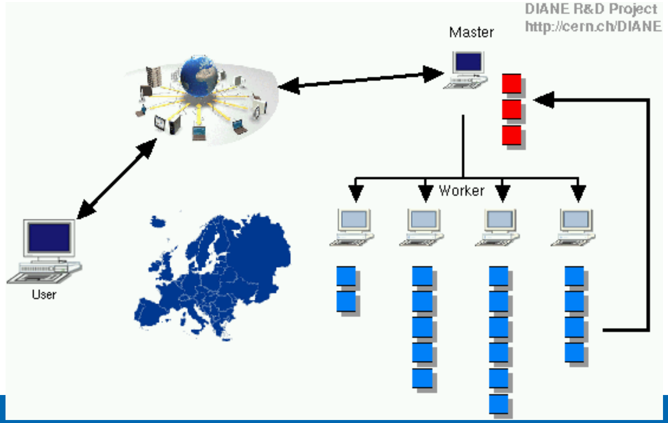
~20 TB transferred in few months

1000 PCs

•Results regularly presented to leading conferences:

•Lattice 2007, 2008 and 2009 by Ph. De Forcrand (ETH and CERN)





Getting Started with Ganga and DIANE

- DIANE and Ganga can be installed via simple installation scripts available online…

diane

- Run these installation steps below to download and install DIANE:

```
mkdir ~/diane
cd ~/diane
wget http://cern.ch/diane/packages/diane-install
python diane-install 2.0-beta20
~/diane/install/2.0-beta20/bin/diane-env -d bash
```

ganga

- Run installation steps below to download and install Ganga:

```
wget http://cern.ch/ganga/download/ganga-install
python ganga-install [OPTIONS] VERSION
```

You can optionally install Ganga modules with:

```
    --extern=GangaGUI,GangaPlotter
```

Visit **cern.ch/DIANE** and **cern.ch/ganga** for more information…

Summary

- In the EGI era, enabling both end-users and application builders is a vital activity
- Ganga and DIANE are effective tools to scale up your scientific computations
  - Together they provide transparent and efficient usage of heterogeneous resources
- Startup overhead is relatively low. You can grid-enable your applications now!
- See poster #19 to learn more...