Open Source Tools for Grid Application Development

Hurng-Chun Lee, Academia Sinica and CERN
Hurng-Chun.Lee@cern.ch

Academia Sinica Grid Computing

Outline

- Grid computing in a nutshell
- Grid application development
- Open source tools for grid application

It's a computing infrastructure …

"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

- C. Kesselman and I. Foster (1998)

It's about resource sharing and collaboration ...

Grid computing is concerned with "coordinated resource sharing and problem solving in dynamic, multi-institutional **virtual organization**."

- The Anatomy of the Grid (2000)

It should be “open” and “high-level”

- Grid is a system …
  - … coordinating resources that are not subject to centralized control
  - … using standard, open, general-purpose protocols and interfaces
  - … delivering nontrivial qualities of services

- I. Forster (2002)

So, Grid is ...

- a service for sharing computer power and data storage capacity over the Internet

- Grid Café (2003)

So, Grid is ...

- a service for sharing computer power and data storage capacity over the Internet

- Grid Café (2003)

World-Wide Web ⇒ Share of information

World-Wide Grid ⇒ Share of computing power

The dream - a global computer

- a system goes well beyond simple communication between computers, and aims ultimately to turn the global network of computers into one vast computational resource

The dream - a global computer

- a system goes well beyond simple communication between computers, and aims ultimately to turn the global network of computers into one vast computational resource

Here is your input data coming from

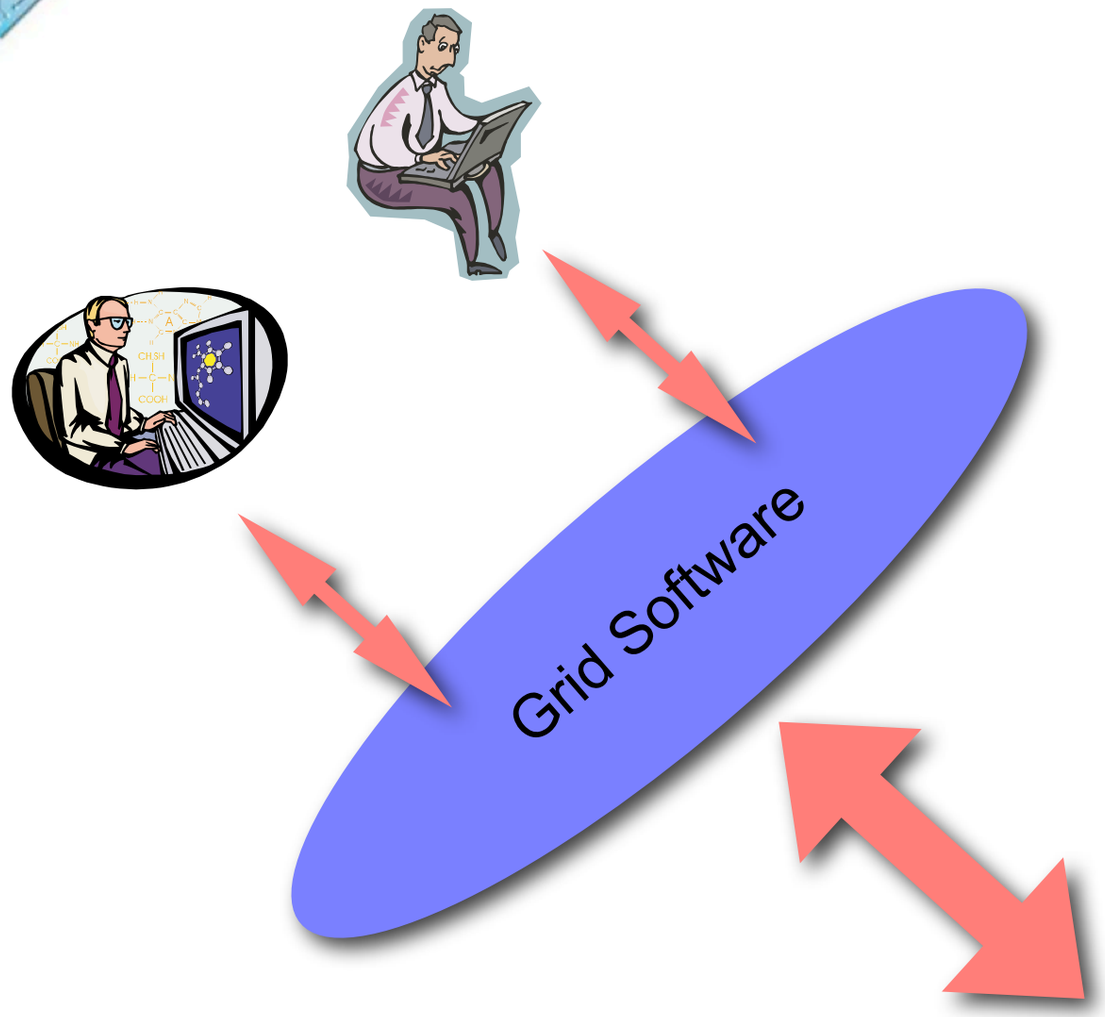
Here is your process is run

Here is your output data going to

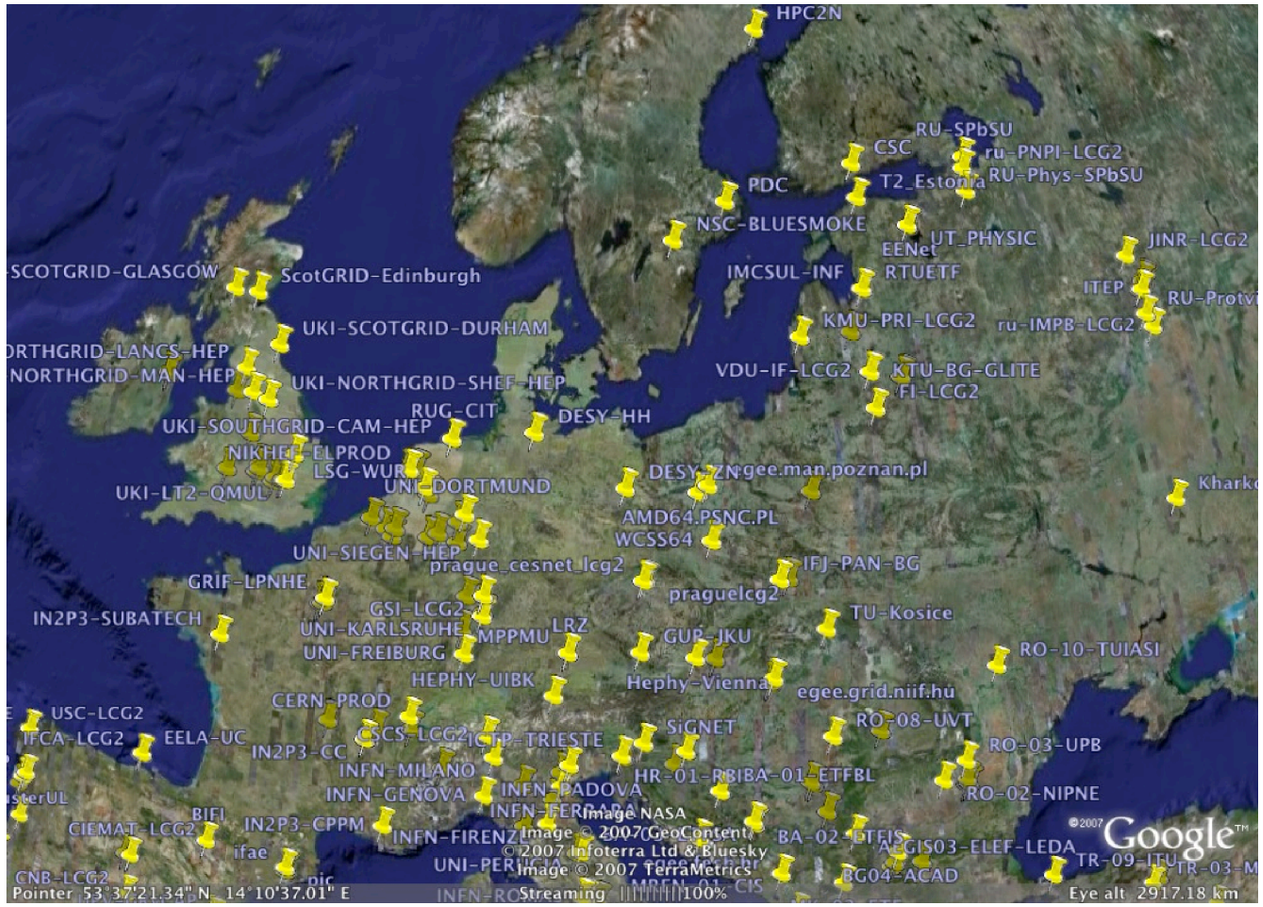
You are here

The reality ...

Building the Grid = software development



- build on top of the existing computing infrastructure (fabric)
  - internet
  - computing/data centers
  - etc.



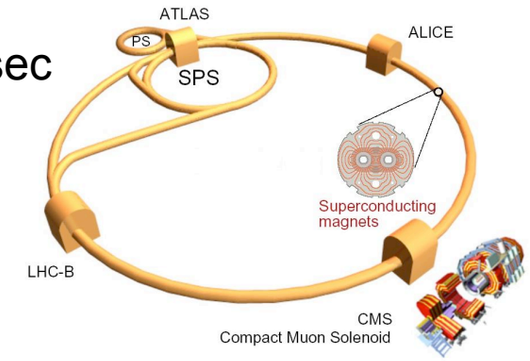
- Driven by scientific challenges
  - High-energy physics
  - Biomedical applications
  - etc.

Use case: High-energy physics

central production

~ 200 - 1500 MByte/sec

~ PByte/sec



CERN Geneva

10 - 40 Gbps

TW CA FR DE IT NL SP UK US

~ 10 Gbps

NTU AS AU JP KR

1 - 10 Gbps

Institute Institute Institute

thousands of users world-wide

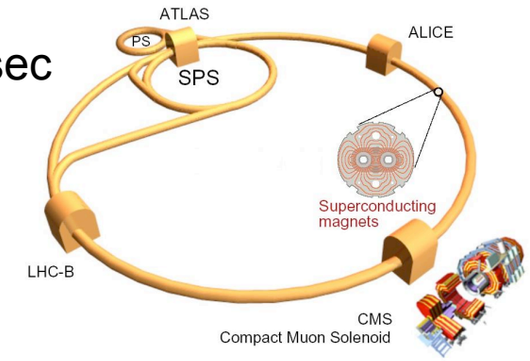
distributed data analysis

Use case: High-energy physics

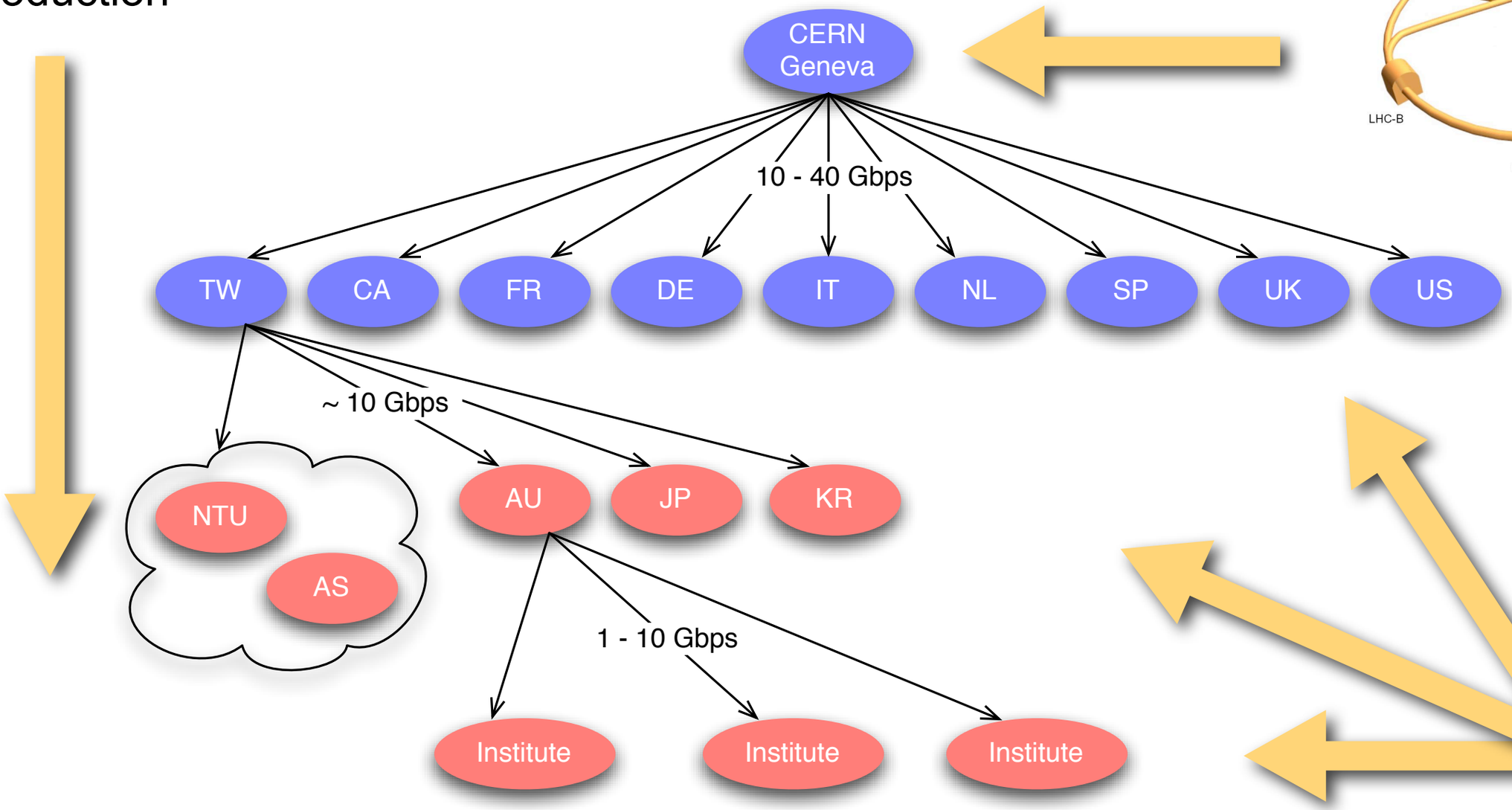
data production rate: ~ 10 PByte/year
24 x 7 operation

~ 200 - 1500 MByte/sec

~ PByte/sec



central production



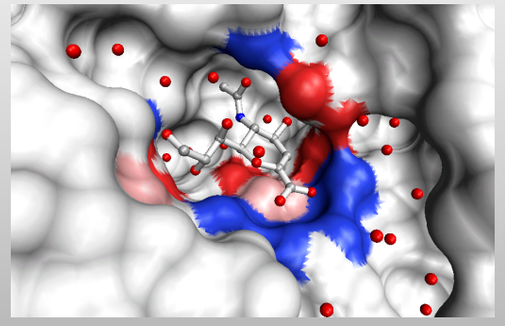
Use case: drug discovery

Millions of chemical compounds available in laboratories

High Throughput Screening
$2/compound, nearly impossible

300,000 Chemical compounds:
ZINC
Chemical combinatorial library

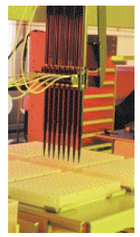
Molecular docking (Autodock)
~137 CPU years, 600 GB data



Data challenge on EGEE, Auvergrid, TWGrid
~6 weeks on ~2000 computers

Target (PDB) :
Neuraminidase (8 structures)

Hits sorting and refining



In vitro screening of 100 hits

Use case: drug discovery

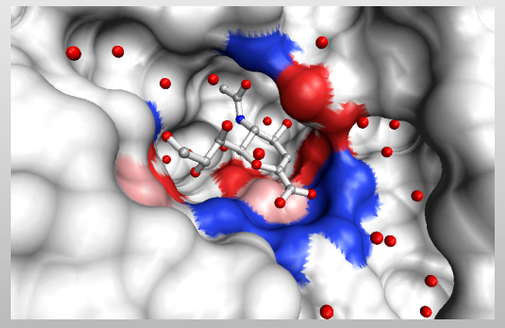
on-demand, short deadline, cost effective

Millions of chemical compounds available in laboratories

High Throughput Screening
$2/compound, nearly impossible

300,000 Chemical compounds:
ZINC
Chemical combinatorial library

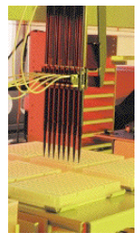
Molecular docking (Autodock)
~137 CPU years, 600 GB data



Data challenge on EGEE, Auvergrid, TWGrid
~6 weeks on ~2000 computers

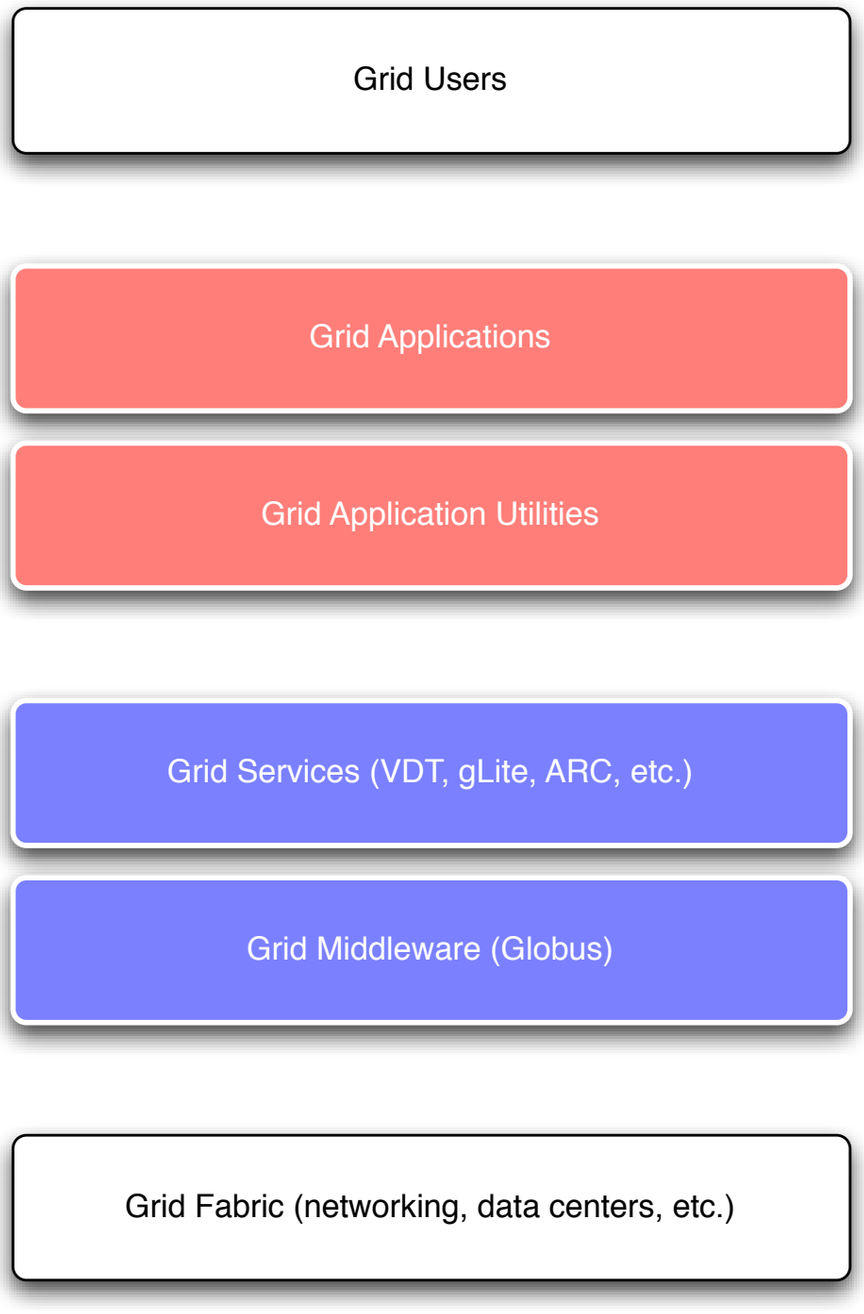
Target (PDB) :
Neuraminidase (8 structures)

Hits sorting and refining



In vitro screening of 100 hits

The grid software stack



domain experts; non-IT engineers

domain specific functionalities

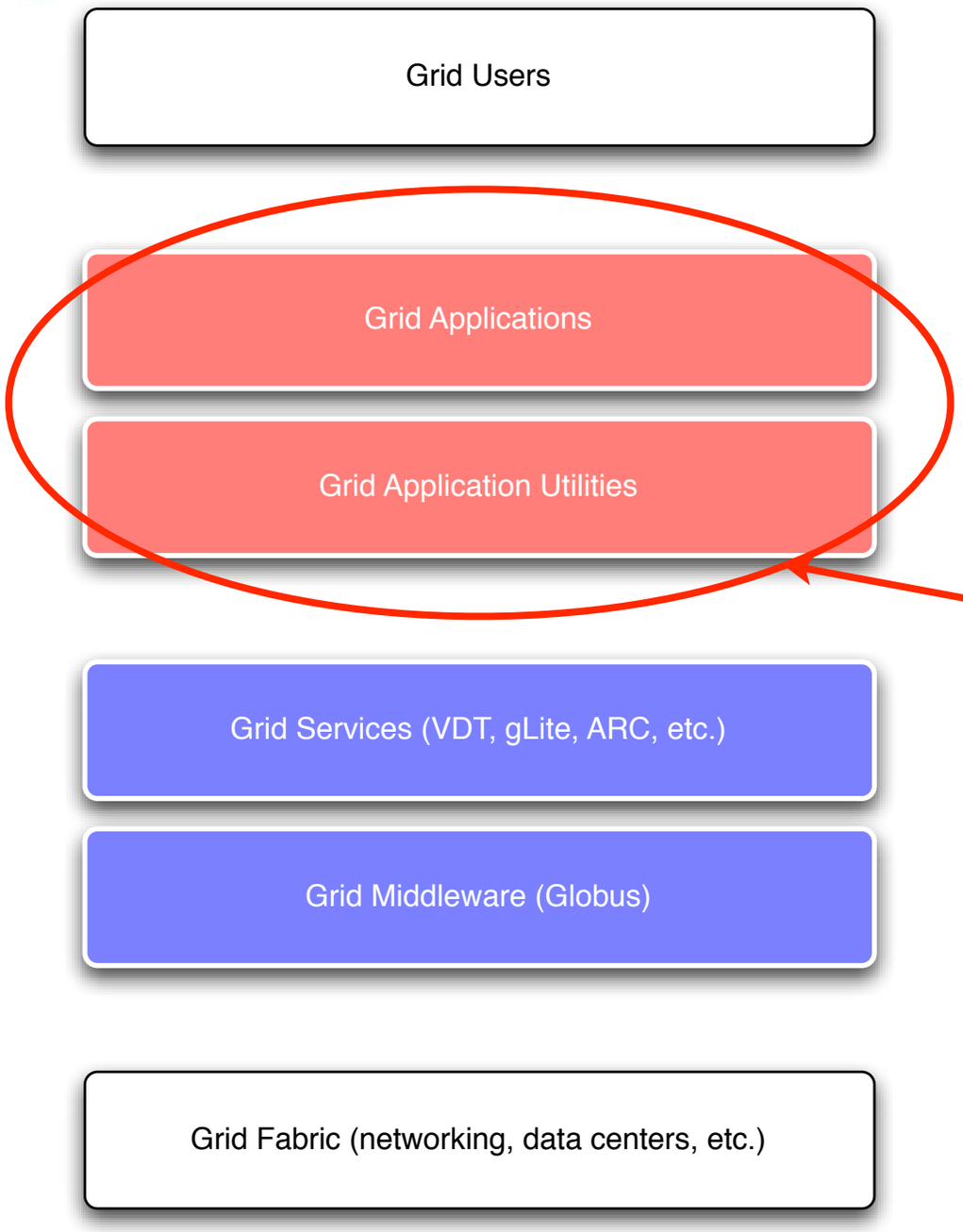
usability, reliability, interoperability

monitoring, accounting, brokering

basic protocols for resource sharing/binding

heterogeneous and evolving technologies

The grid software stack



domain experts; non-IT engineers

domain specific functionalities

usability, reliability, interoperability

our focus in this talk

monitoring, accounting, brokering

basic protocols for resource sharing/binding

heterogeneous and evolving technologies

Application development issues

- Grid application is a large scale application
  - using thousands of distributed computers to produce millions of files
- Grid is dynamic
  - we have to live with failures
- Grid is generic
  - not well customized for a specific application
- Grid is complex
  - hard to be understood by a normal user

Grid application utilities

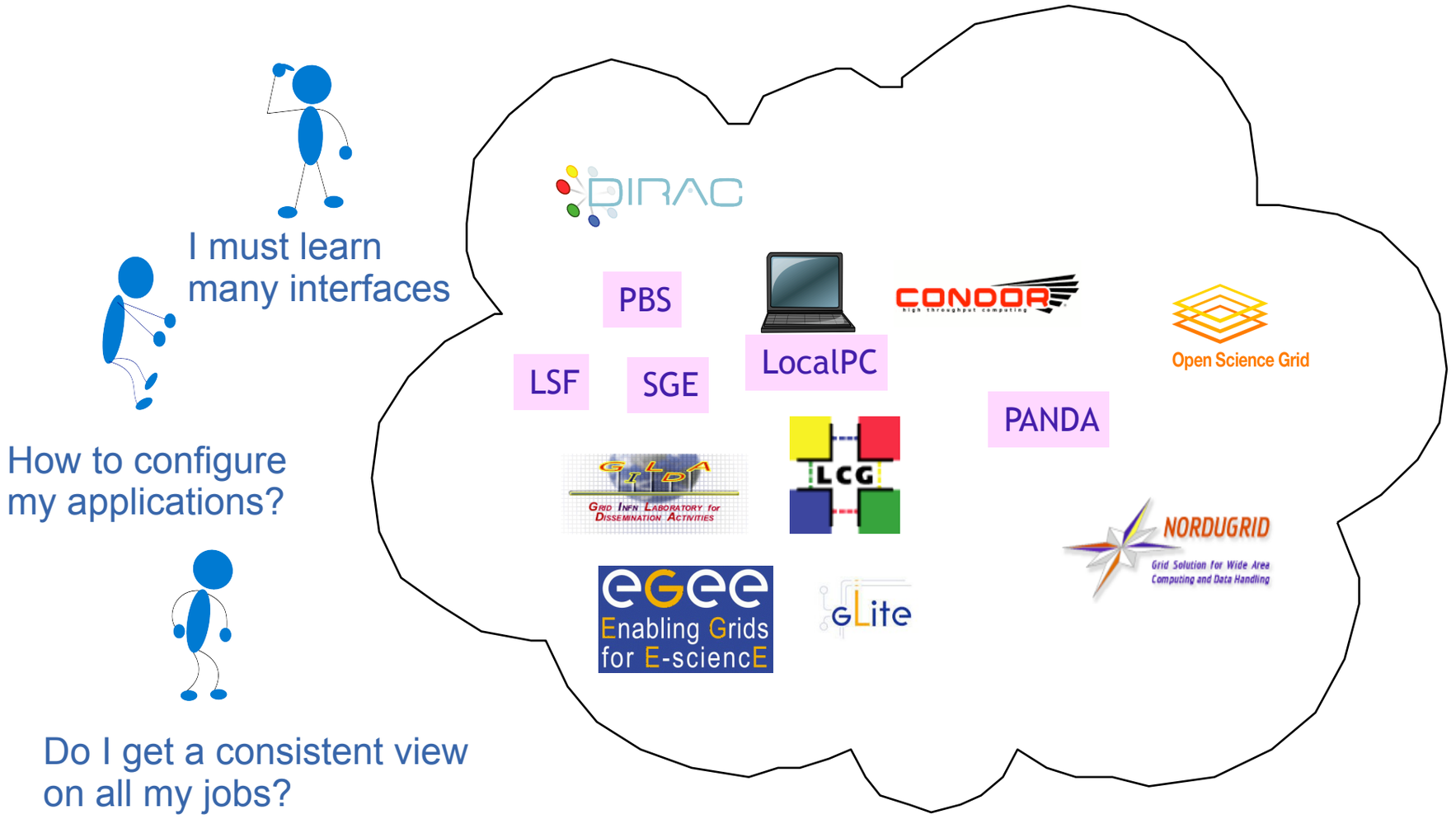
- High-level utilities are needed in building grid applications
  - Also generic; but can be easily customized to meet different application requirements
  - simple, reliable, scalable
  - extensible (i.e. plug-in approach)

- Open source examples
  - GANGA, DIANE, AMGA

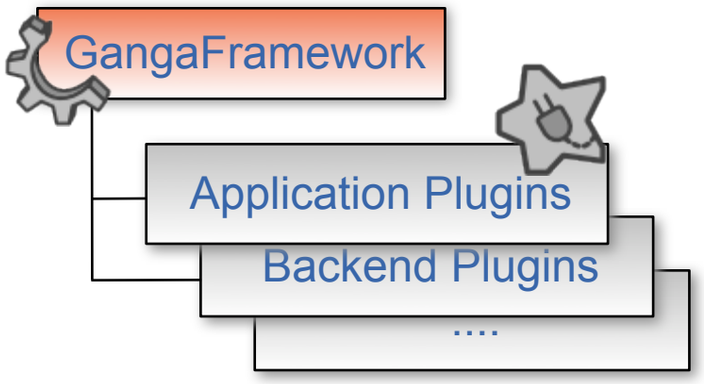
GANGA: motivation

http://cern.ch/ganga

- FAQ: running applications on multiple computing backends
  - grid is a complementary computing environment
  - technology is evolving



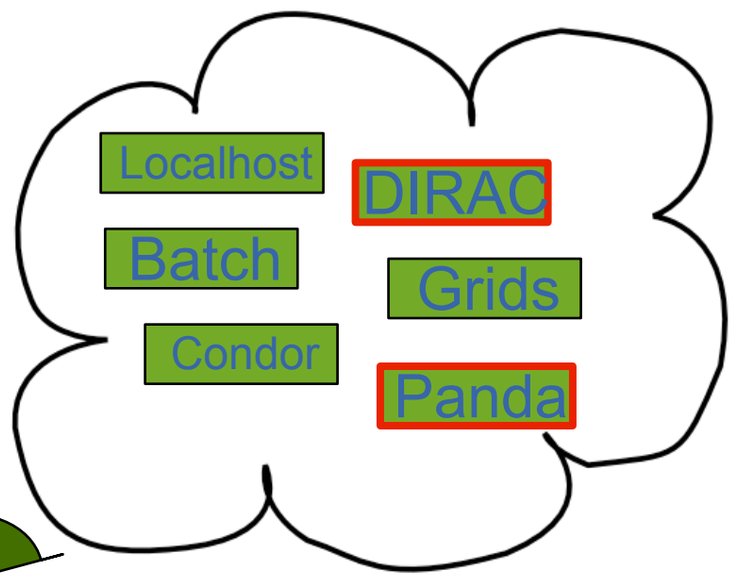
GANGA: a high-level job management tool



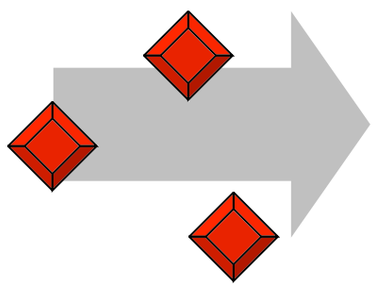
"configure once, run anywhere"

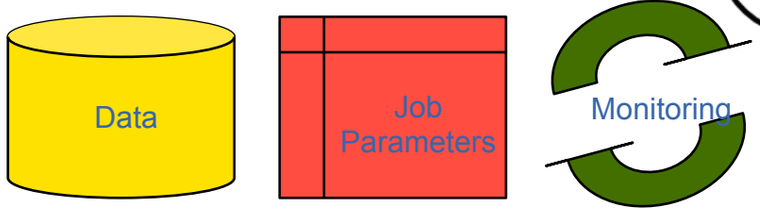
applications

resources



jobs

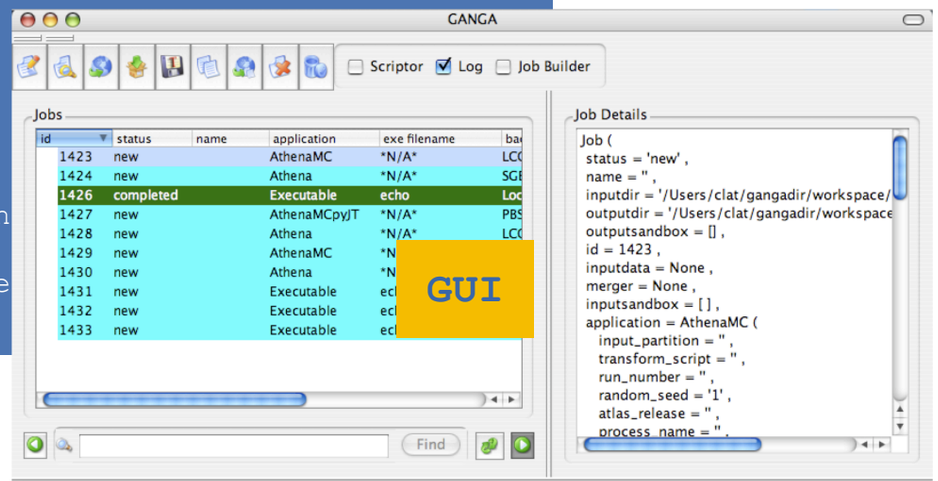




GANGA: multiple UI flavors

```
*** 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
```

CLIP



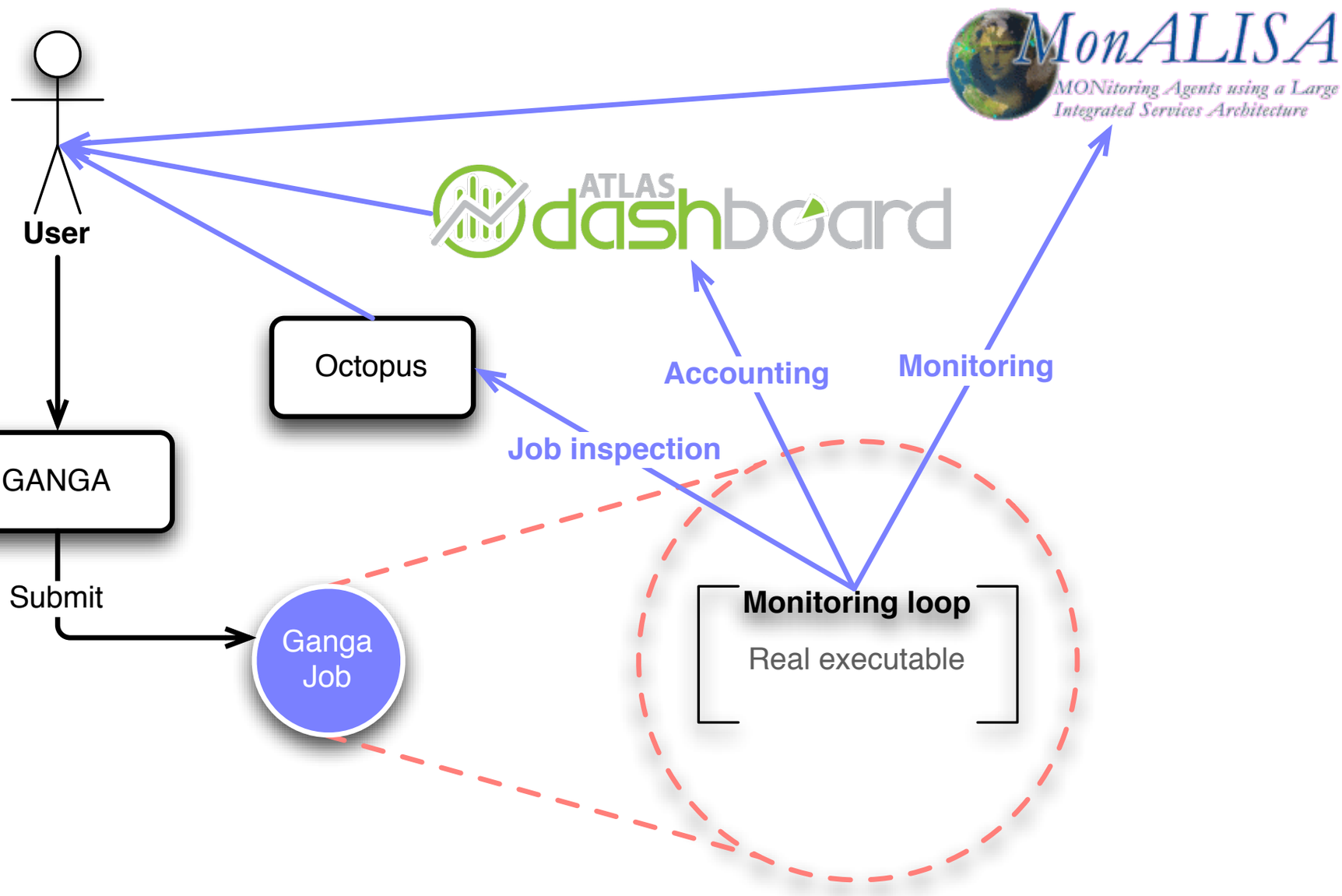
GUI

```
#!/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)
```

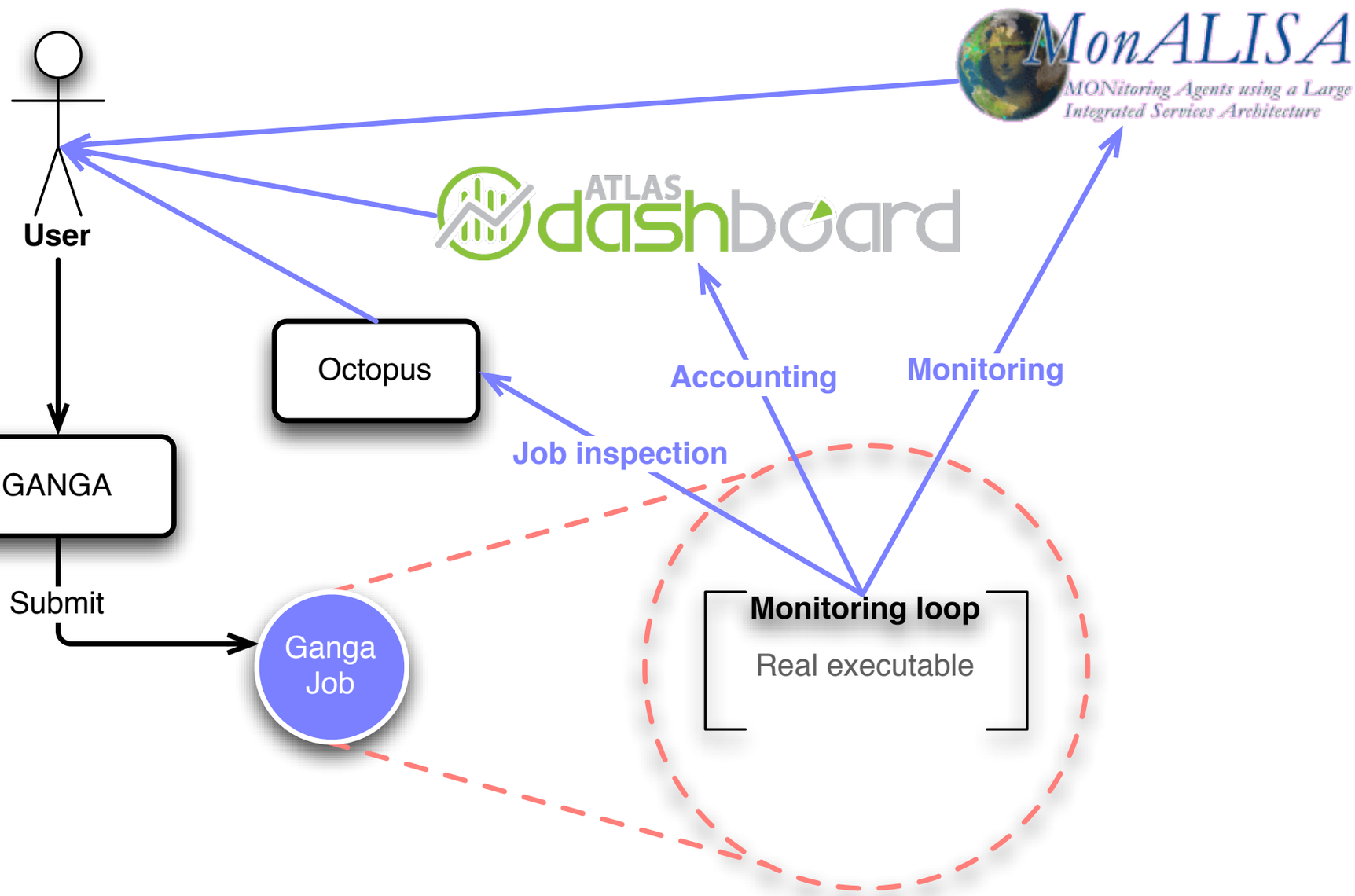
```
./myjob.exec

ganga ./myjob.exec

In [1]:execfile("myjob.exec")
```

GPI & Scripting

GANGA: monitoring, accounting and debugging



GANGA: monitoring, accounting and debugging



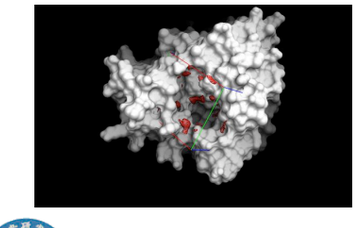
modular approach for application specific information

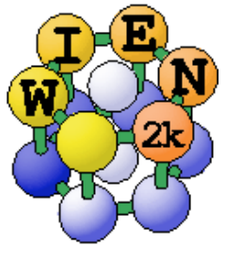
GANGA: users

Geant 4

HARP

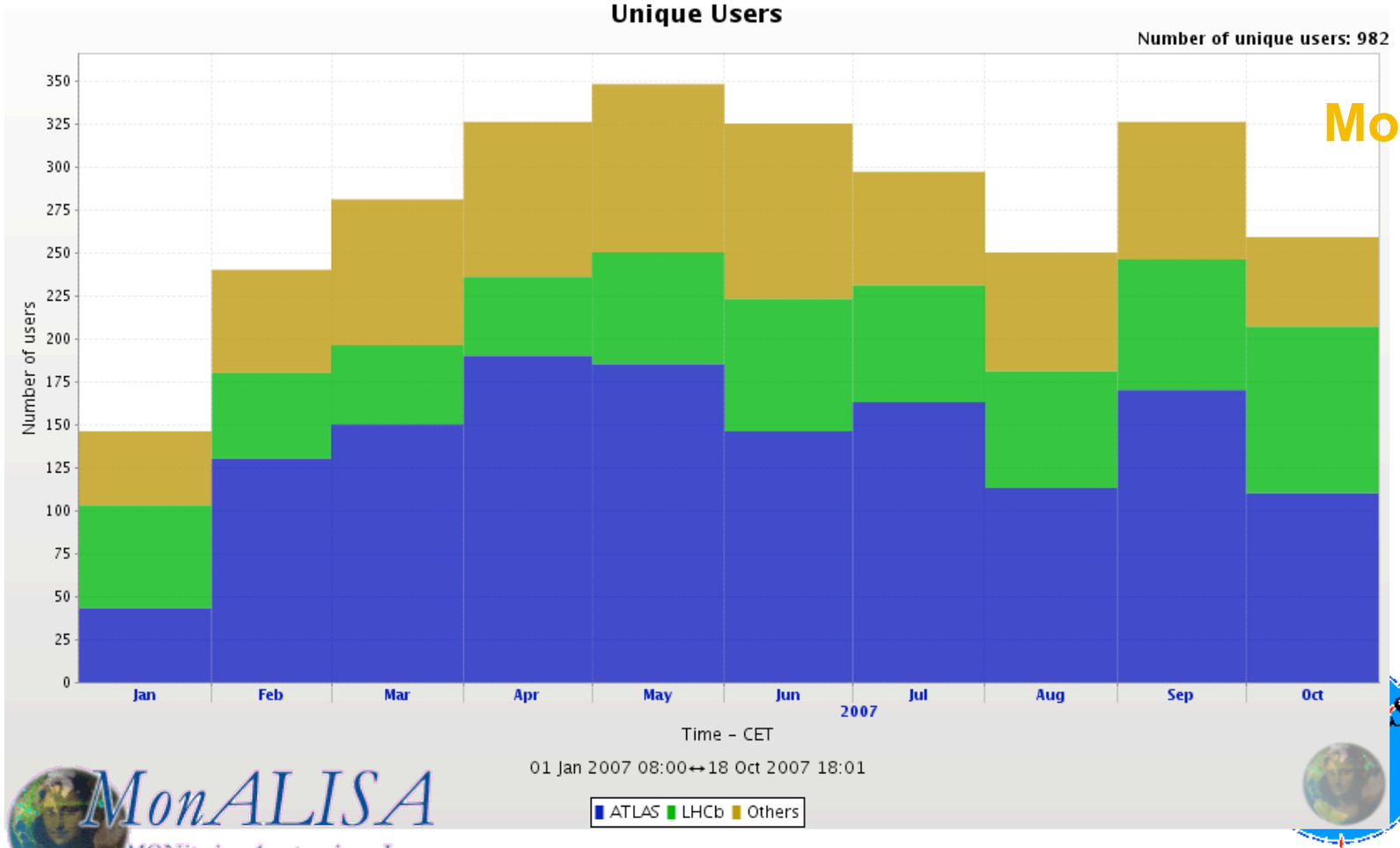
Garfield





Academia Sinica
Genomics Research Center

GANGA: users



MonALISA
MONitoring Agents using a Large Integrated Services Architecture

More than 900 users use Ganga

HARP

Garfield

Academia Sinica
Genomics Research Center

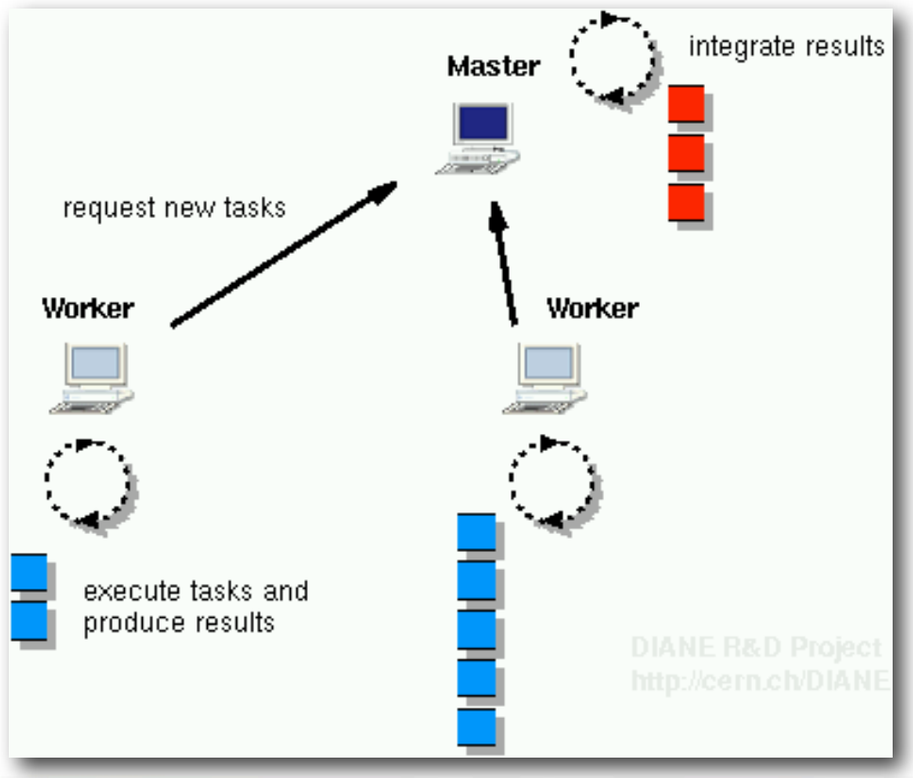
DIANE: motivation

http://cern.ch/diane

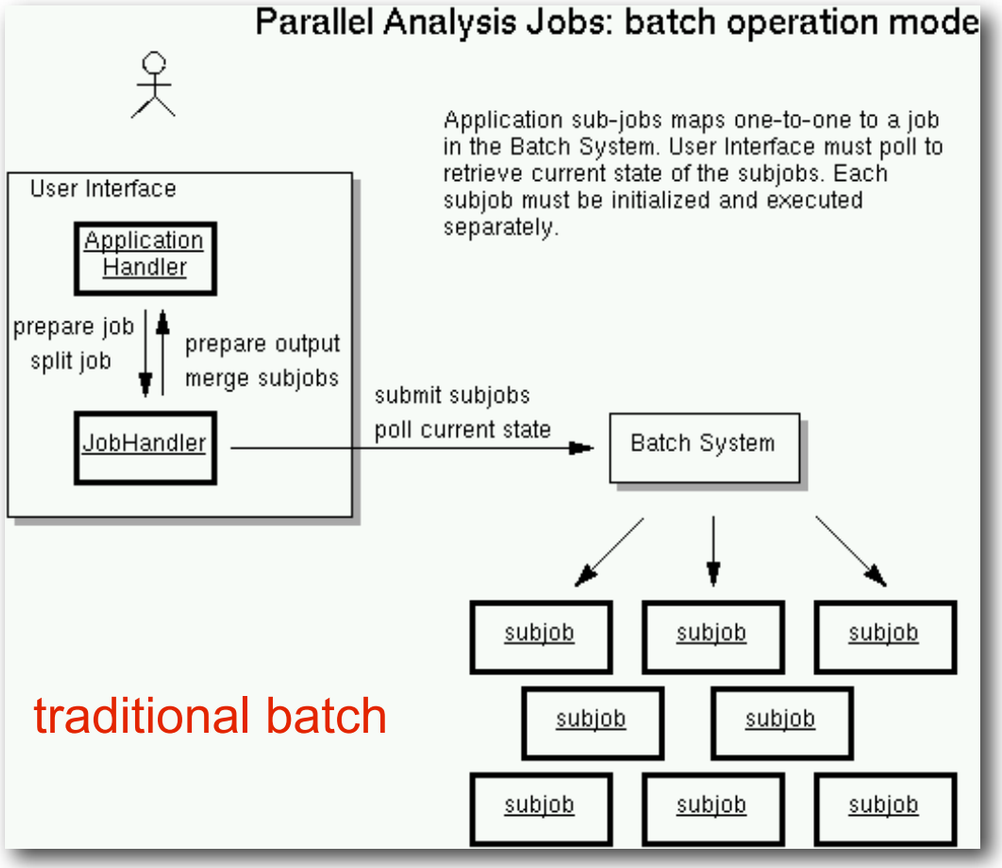
- user(application)-level scheduling
  - advanced resource control by application
  - save the process scheduling overhead
  - efficient computing resource usage
- failure recovery
  - resource error (i.e. network or service unavailable)
  - application error (i.e. library incompatible)
- An use case of GANGA

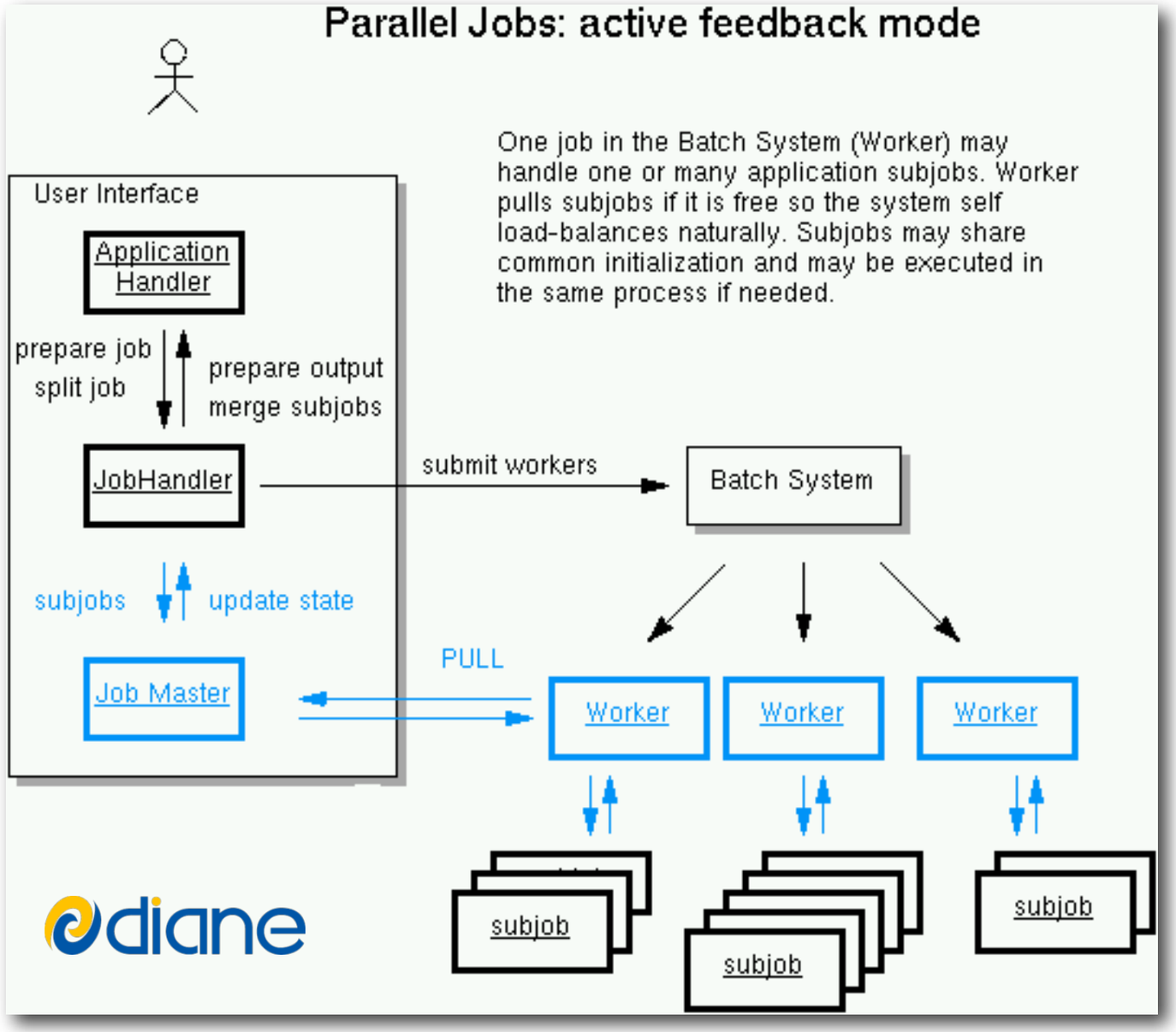
DIANE: high-level task scheduler

- A lightweight framework for parallel scientific applications in master-worker model
  - assuming that job can be split into a number of independent tasks
- It takes care of all synchronization, communication and workflow management details on behalf of application
  - it's a framework
- Heartbeat worker health check
- Plug-ins for specific applications
- Application defined fail-over mechanisms



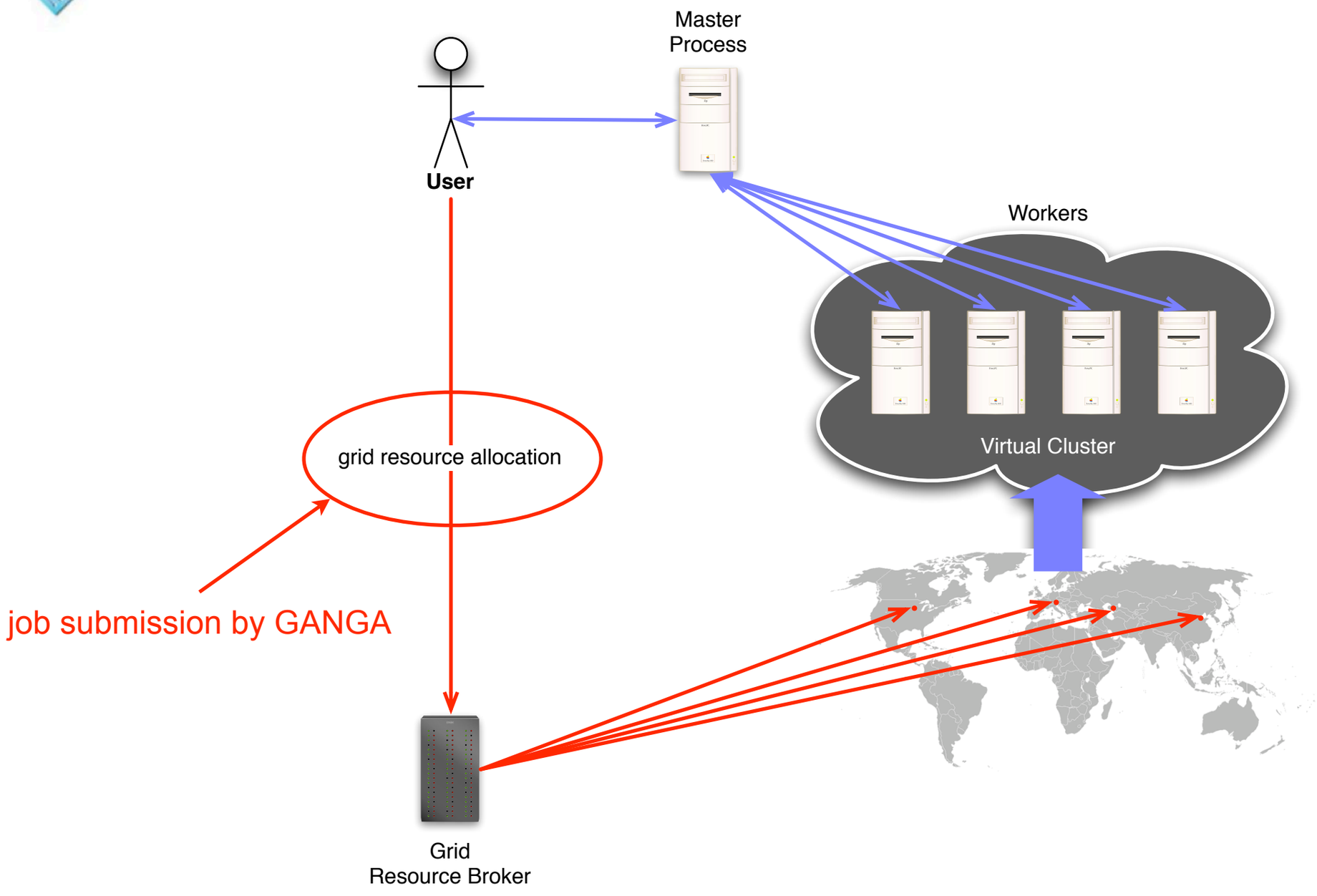
DIANE: job pull model



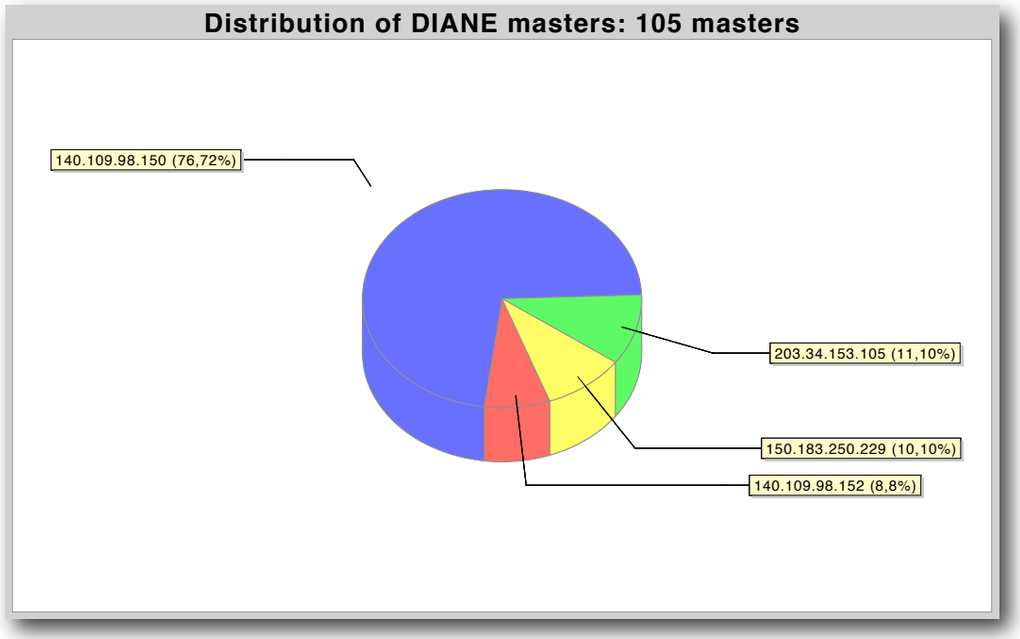


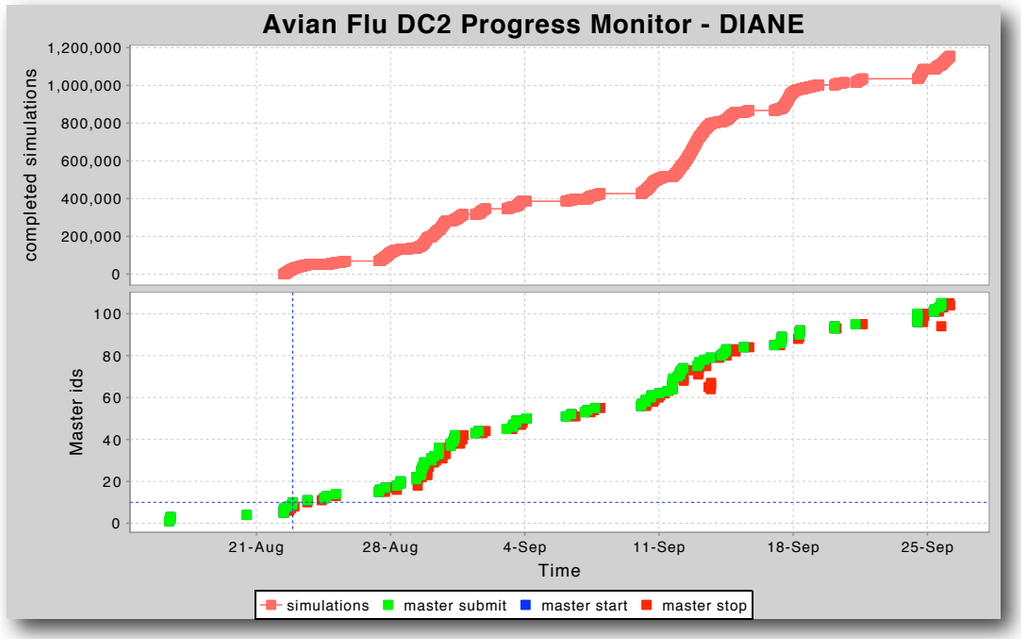
- less physical jobs
- saving scheduling time
- load-balancing
- interactions with working processes

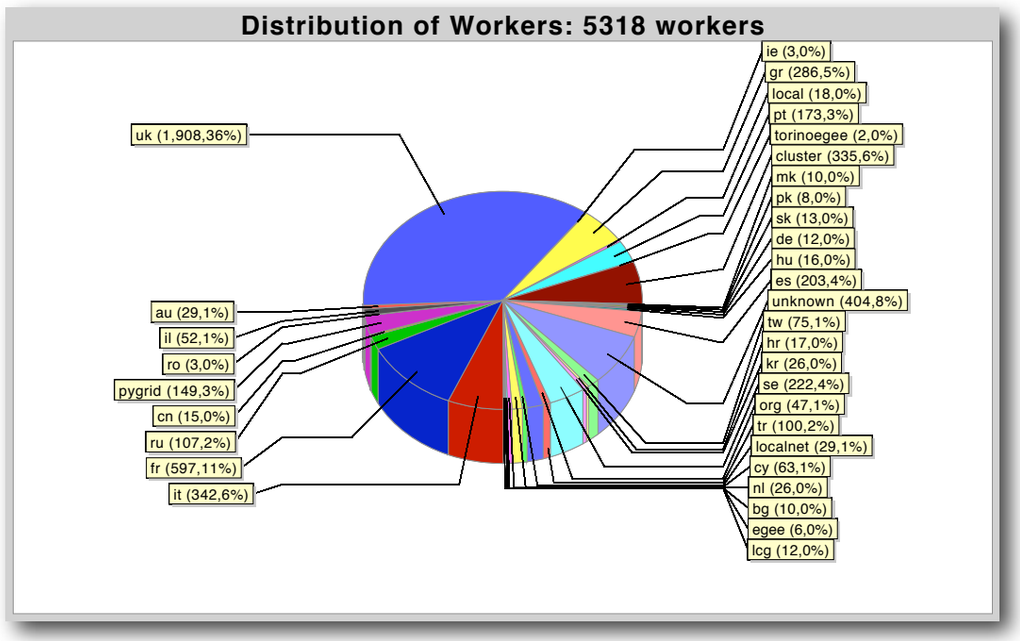
DIANE: a grid overlay



DIANE: drug discovery use case

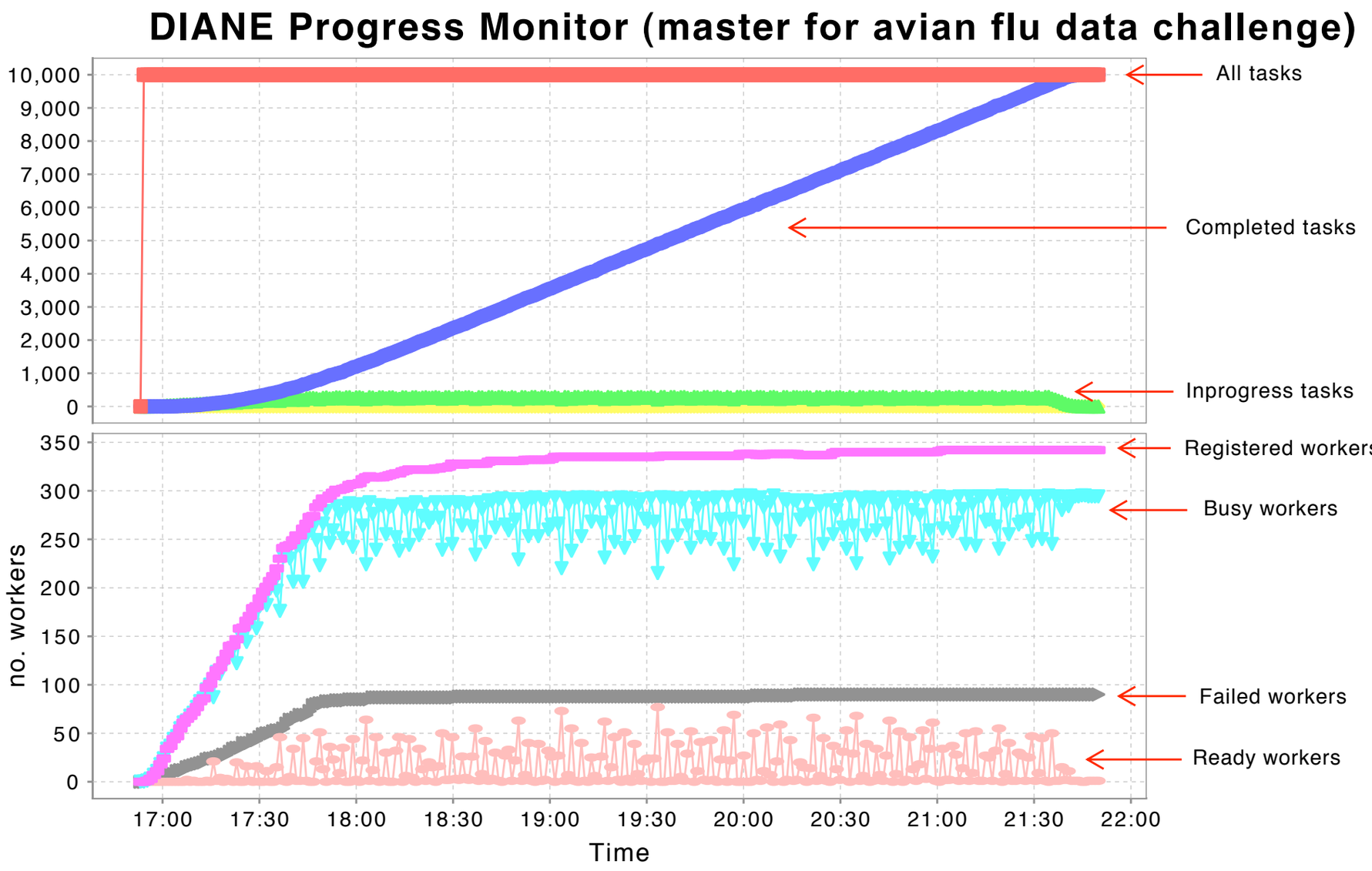






- > 100 distributed DIANE instances
- > 5000 CPUs
- > 1.2 Million simulations
- > 22 CPU years on 1 CPU
- < 1 months

DIANE: application reliability



AMGA: motivation

http://cern.ch/ganga

- Distributed data needs to be indexed
  - not only logical name ⇔ physical name (URI) mapping
  - but also high-level metadata (data about data)
- Relational database is another data storage
  - interoperability between different RDB technologies
  - intensive access through internet
- Data security, persistency and availability

AMGA: metadata concept

- Metadata - List of attributes associated with entries
  - Attribute – key/value pair with type information
  - Type – The type (int, float, string,…)
  - Name/Key – The name of the attribute
  - Value - Value of an entry's attribute
- Schema – A set of attributes
- Collection – A set of entries associated with a schema
- Think of schemas as tables, attributes as columns, entries as rows

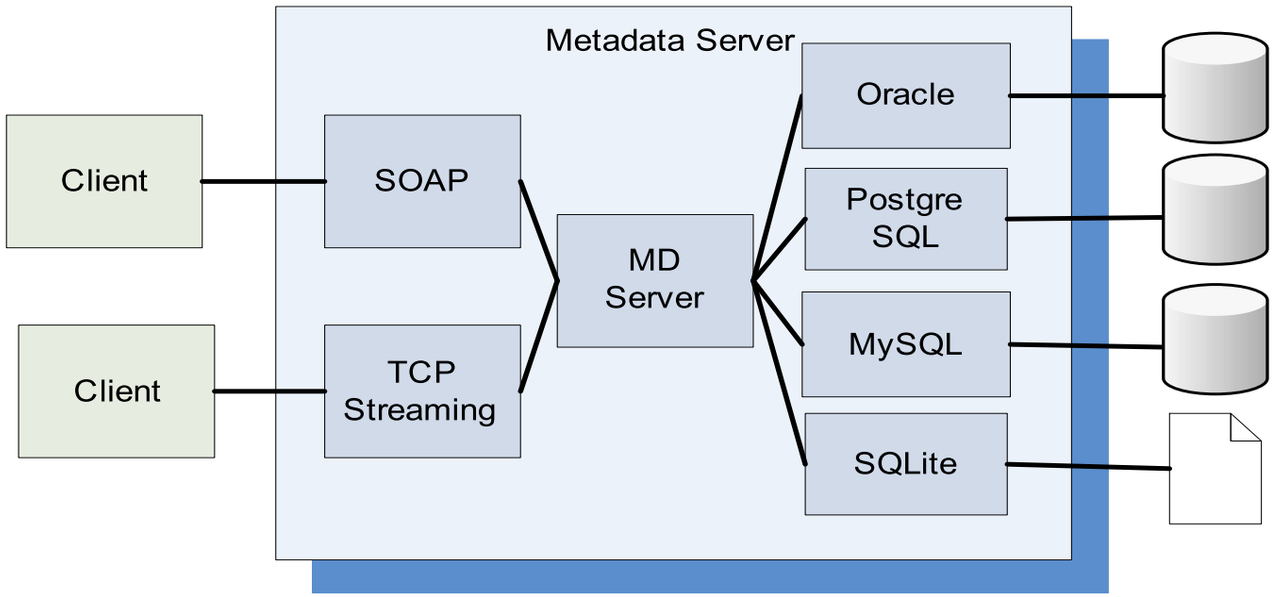
AMGA: metadata catalogue features

- Dynamic Schemas
  - Schemas can be modified at runtime by client
  - Create, delete schemas
  - Add, remove attributes
- Metadata organized as an hierarchy
  - Collections can contain sub-collections
  - Analogy to file system:
    - Collection ⇔ Directory
    - Entry ⇔ File
- Flexible Queries
  - SQL-like query language
  - Joins between schemas

```
Supporting filesystem-like
operations:

ls
mv
mkdir
```

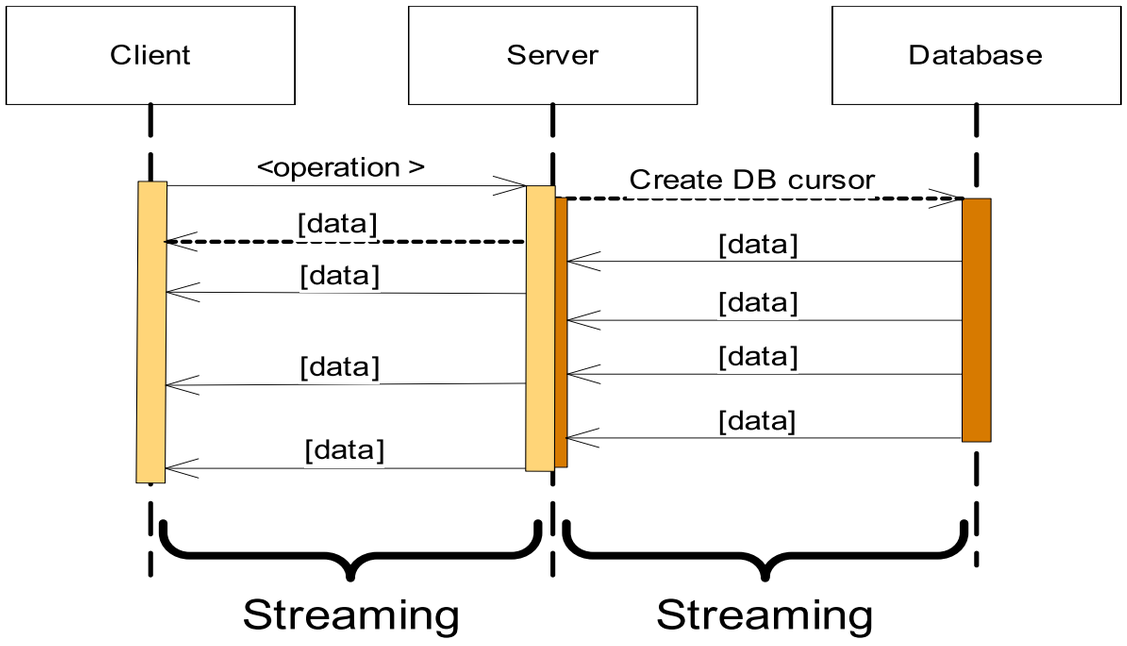
```
QUERY EXAMPLE:

selectattr /gLibrary:FileName \
           /gLibrary:Author \
          `/gLibrary:FILE=/gLAudio:FILE \
           and \
           like(/gLibrary:FileName,"%.mp3")`
```

AMGA: client-server model



- RDB neutral
- TCP streaming for performance
- SOAP for adoptability

- Asynchronous streaming reducing operation overhead

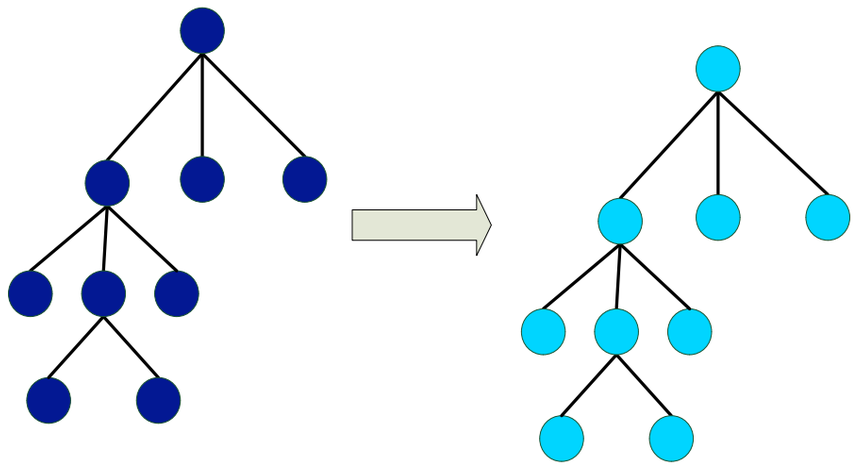


AMGA: security

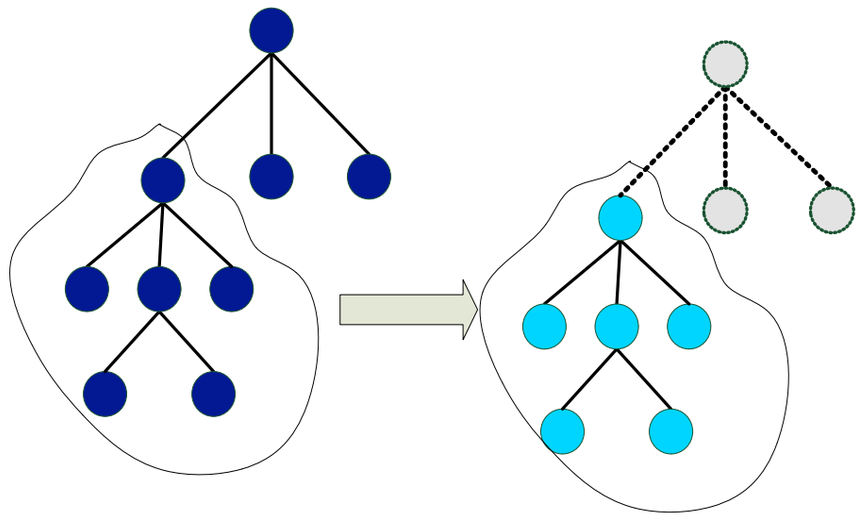
- Unix style permissions
- ACLs – per-collection or per-entry
- Secure connections – SSL
- Client Authentication based on
  - Username/password
  - General X509 certificates
  - Grid-proxy certificates
- Access control via a Virtual Organization Management System (VOMS)

AMGA: metadata preservation / replication

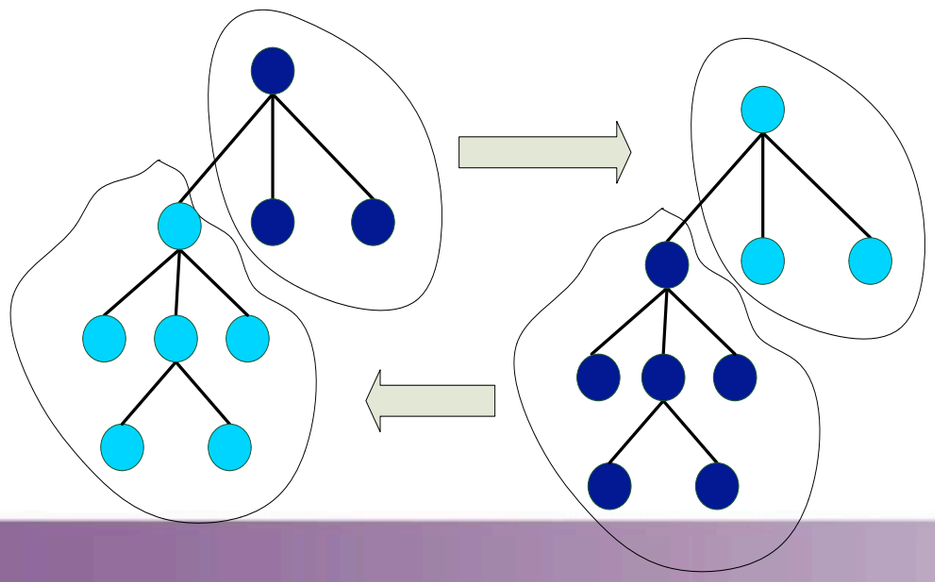
Full replication



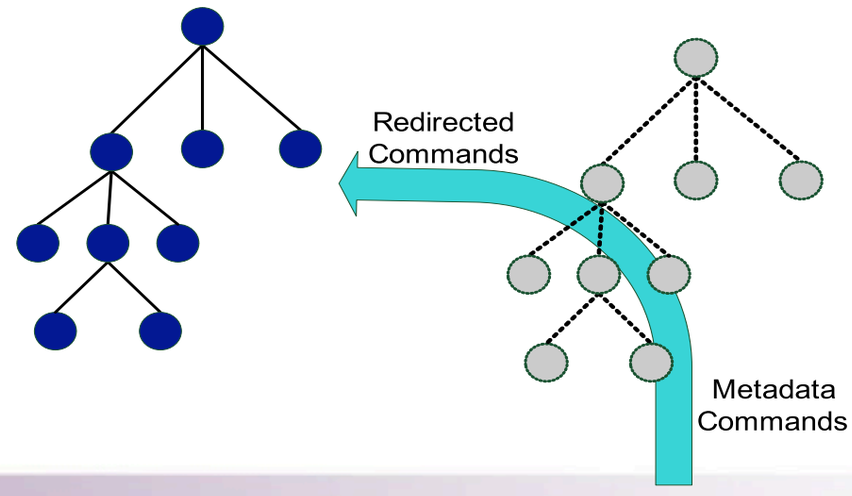
Partial replication



Federation



Proxy



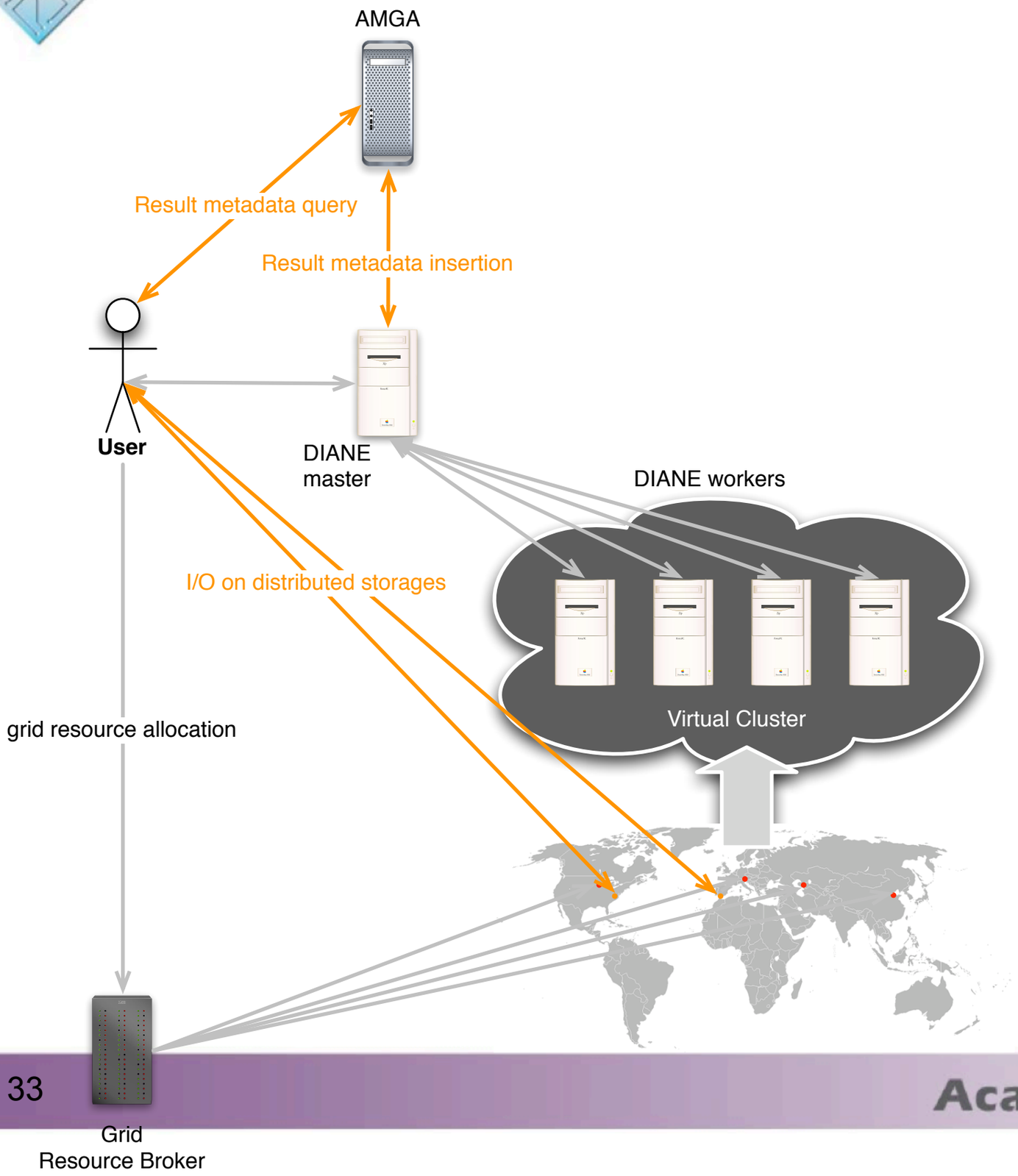
AMGA: drug analysis use case

- Questions:
  - large-scale analysis is a long process; can I investigate my result of the completed simulations during the runtime?
  - there are too many simulation results distributed on several storages; can I just access some of them that I have interests (according to some criteria)?

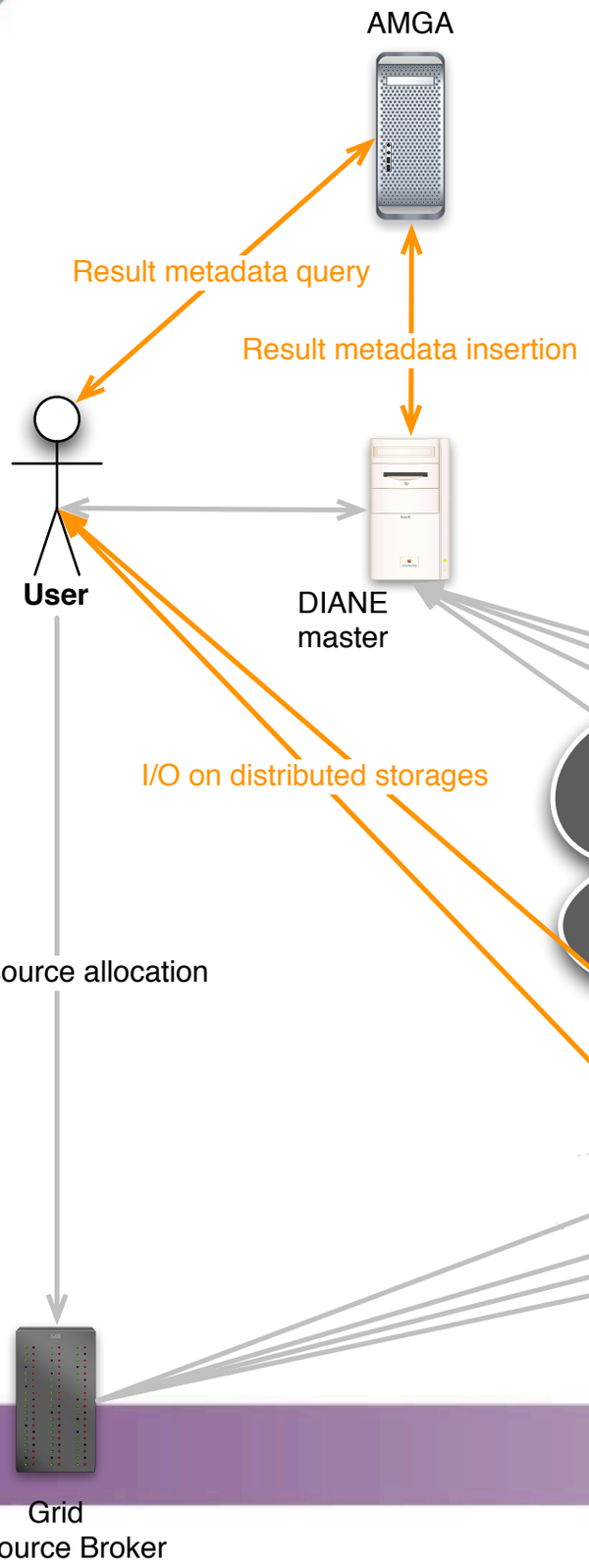
- Answer:

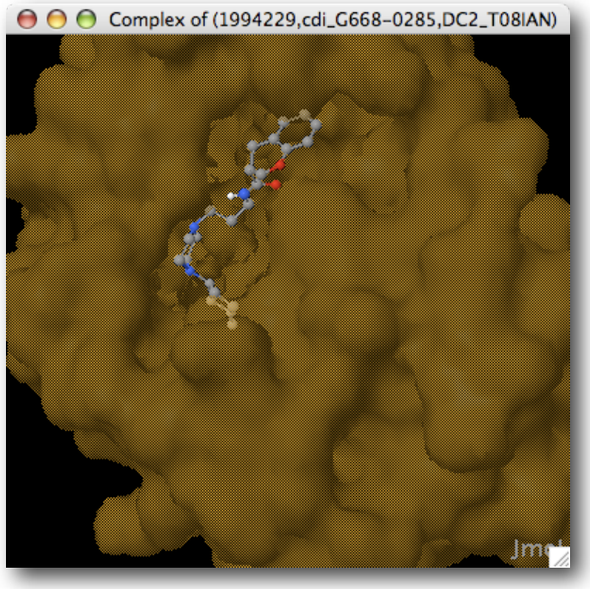
  YES! We need to indexing results with their essential attributes ... ⇒ Metadata catalogue

AMGA: drug analysis use case

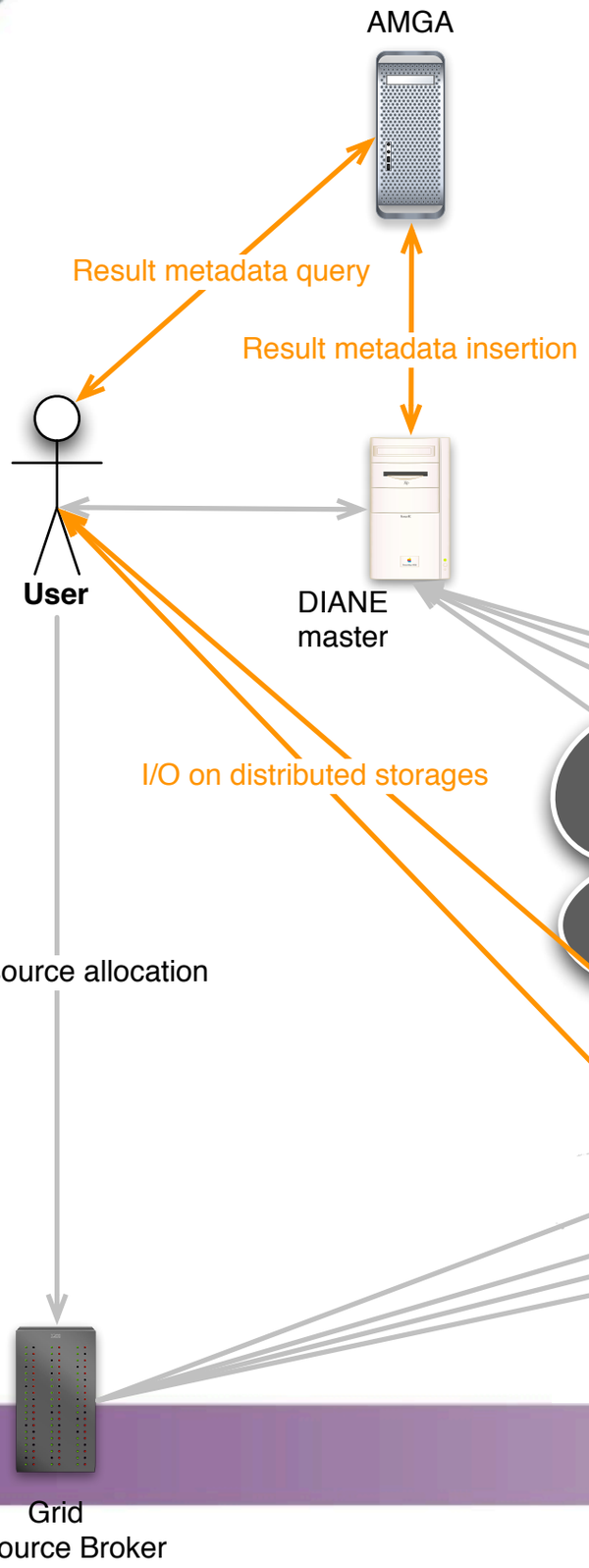


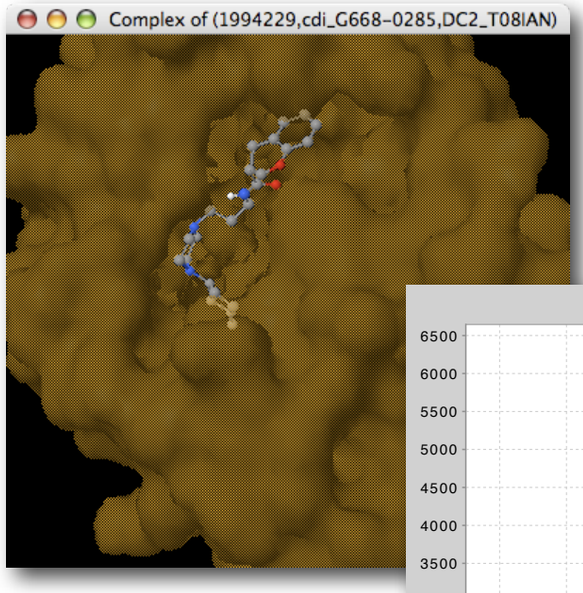
AMGA: drug analysis use case

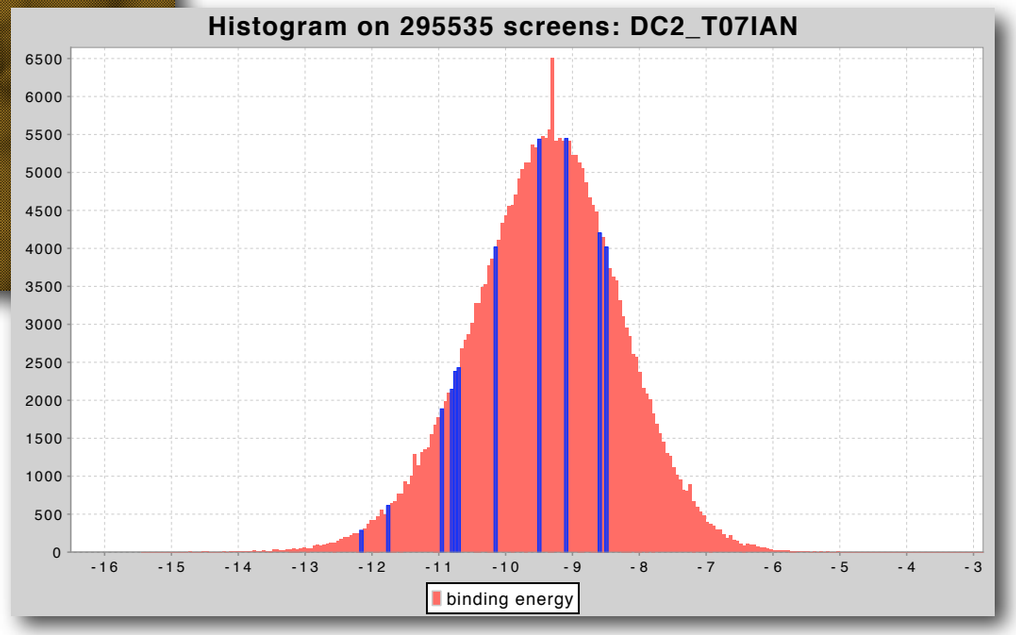




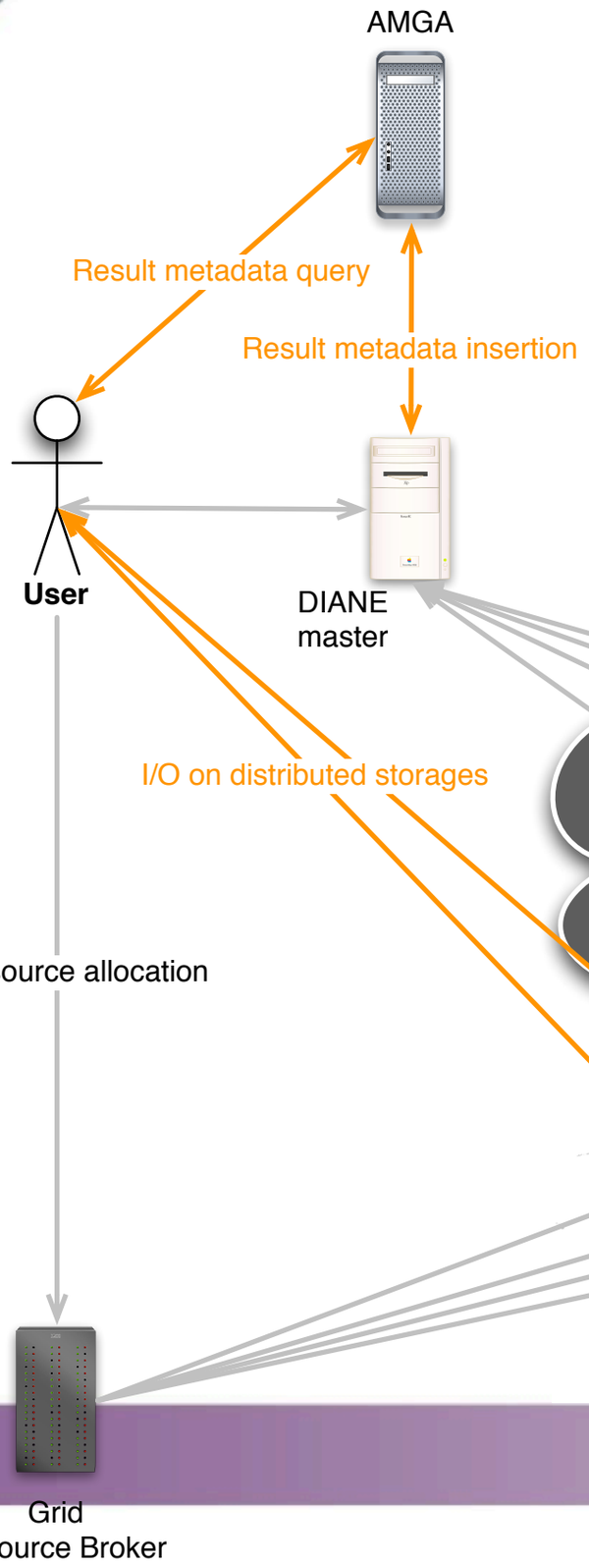
AMGA: drug analysis use case

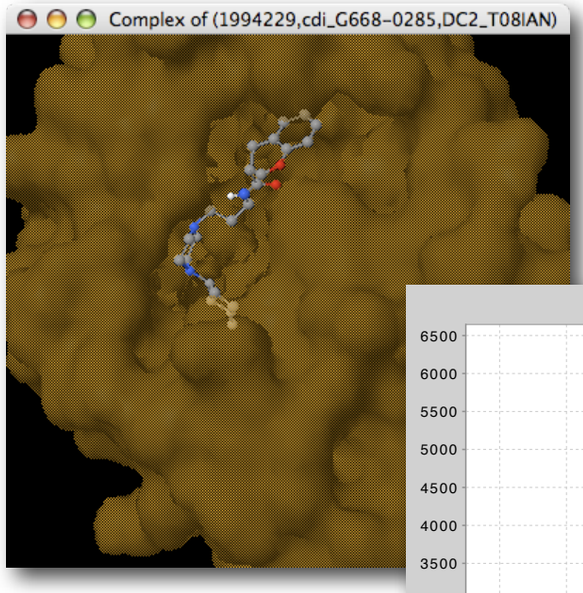


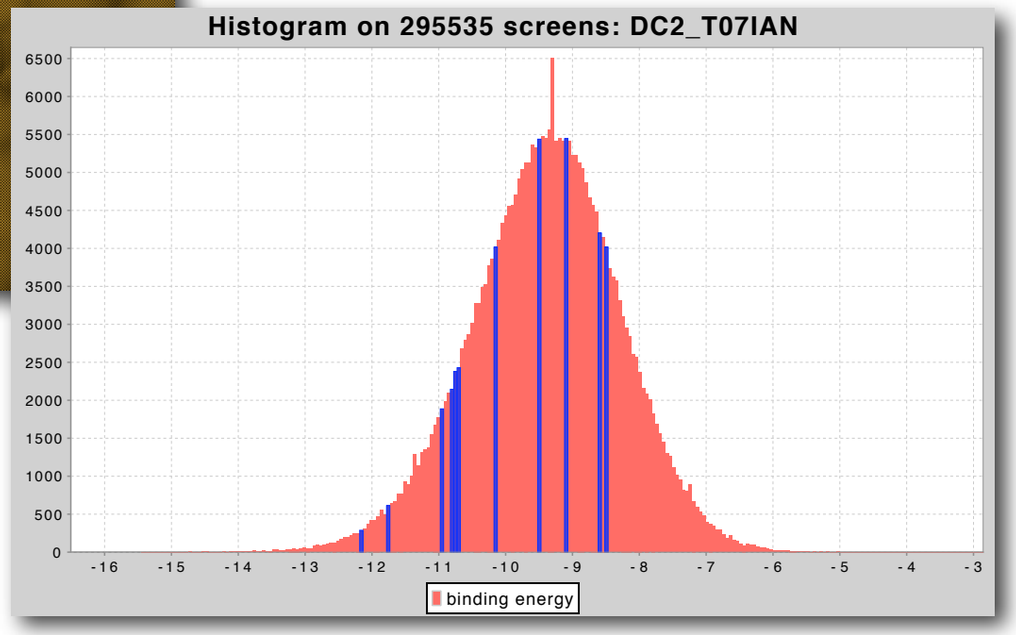


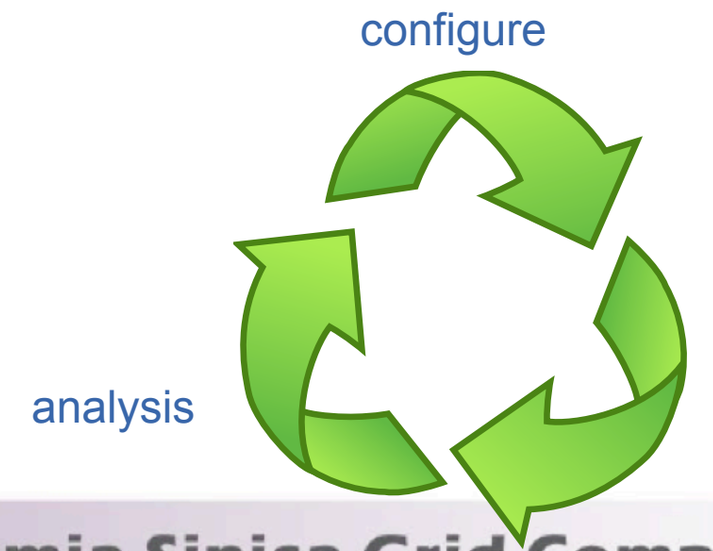


AMGA: drug analysis use case









Conclusion

- Grid is about computing resource sharing
- Grid development is a software engineering
  - working on top of the existing computing infrastructure
  - driven by applications
- High-level utilities are needed in order to fill up the gap between the applications and the middleware
  - where the open source community could contribute

Thank you!

TWGRID: www.twgrid.org