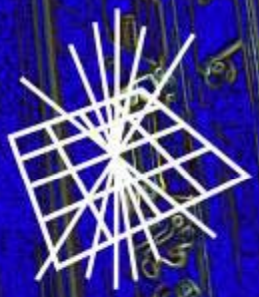


GridPP

UK Computing for Particle Physics

Real Time Monitor of Grid Job Executions

Janusz Martyniak

Imperial College London

What is the RTM

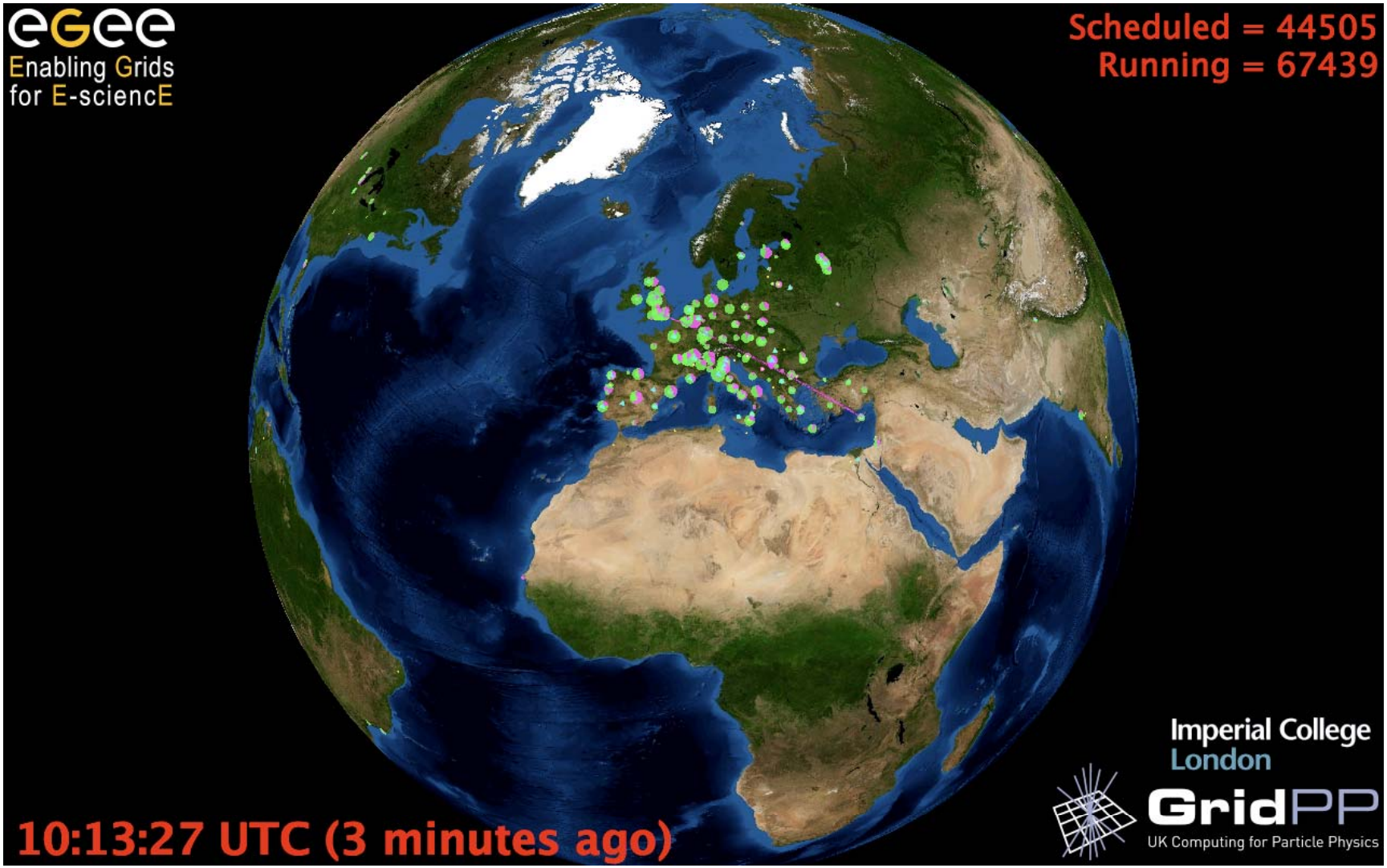
•RTM is a GRID monitoring system which gives on overview of current state of the infrastructure.
•The RTM's view (‘map’) of the Grid is based solely on information retrieved from BDII services
• Job information is obtained by querying EGEE Logging and Bookkeeping servers
•The information gathered is published in XML format and available for clients to use
•RTM supports 2 graphics client programs using satellite imagery from NASA, these clients display the Grid as it is geographically spread over the world.
•There is a text version as well (formatted, ordered by country/site).
•Any site using EGEE compatible technology (i.e LB servers/RB/WMS) may be displayed on the map. If the LB is open for RTM access this happens automatically.
•Webpage: ***http://gridportal.hep.ph.ic.ac.uk/rtm/***

History

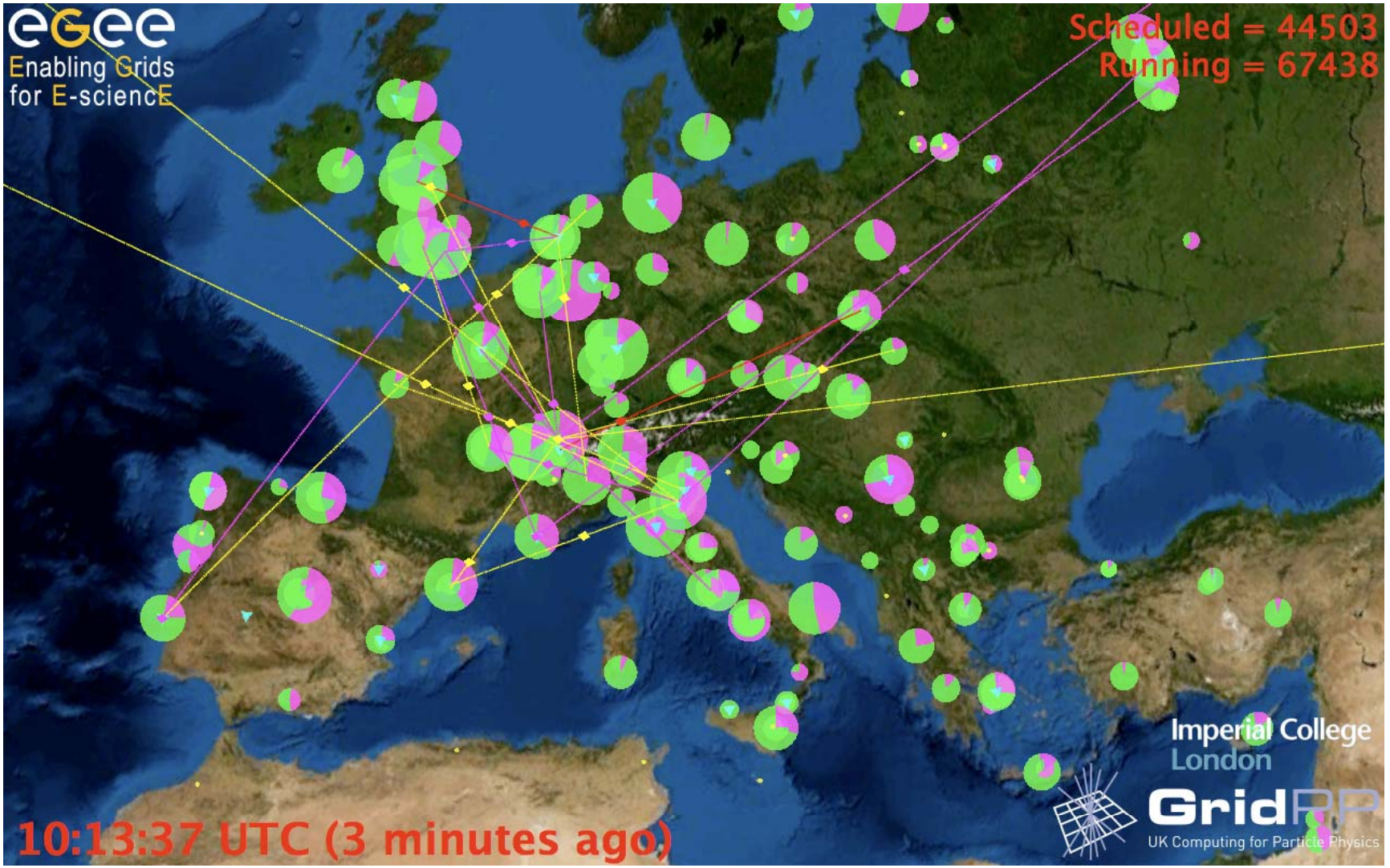
Development started in summer 2002

- initial version 2002, Perl / CGI mix
- 2003, java application/applet – 2D map, ‘stacked’ job display
- 2005-2007, present graphics
  - 2D map
  - 3D globe
  - selectable VO and RB/WMS
  - RB/CE/jobs info on request (graphs and tables)
- 2008 - DB restructuring, WMS support, revised threading, many bug fixes and improvements.

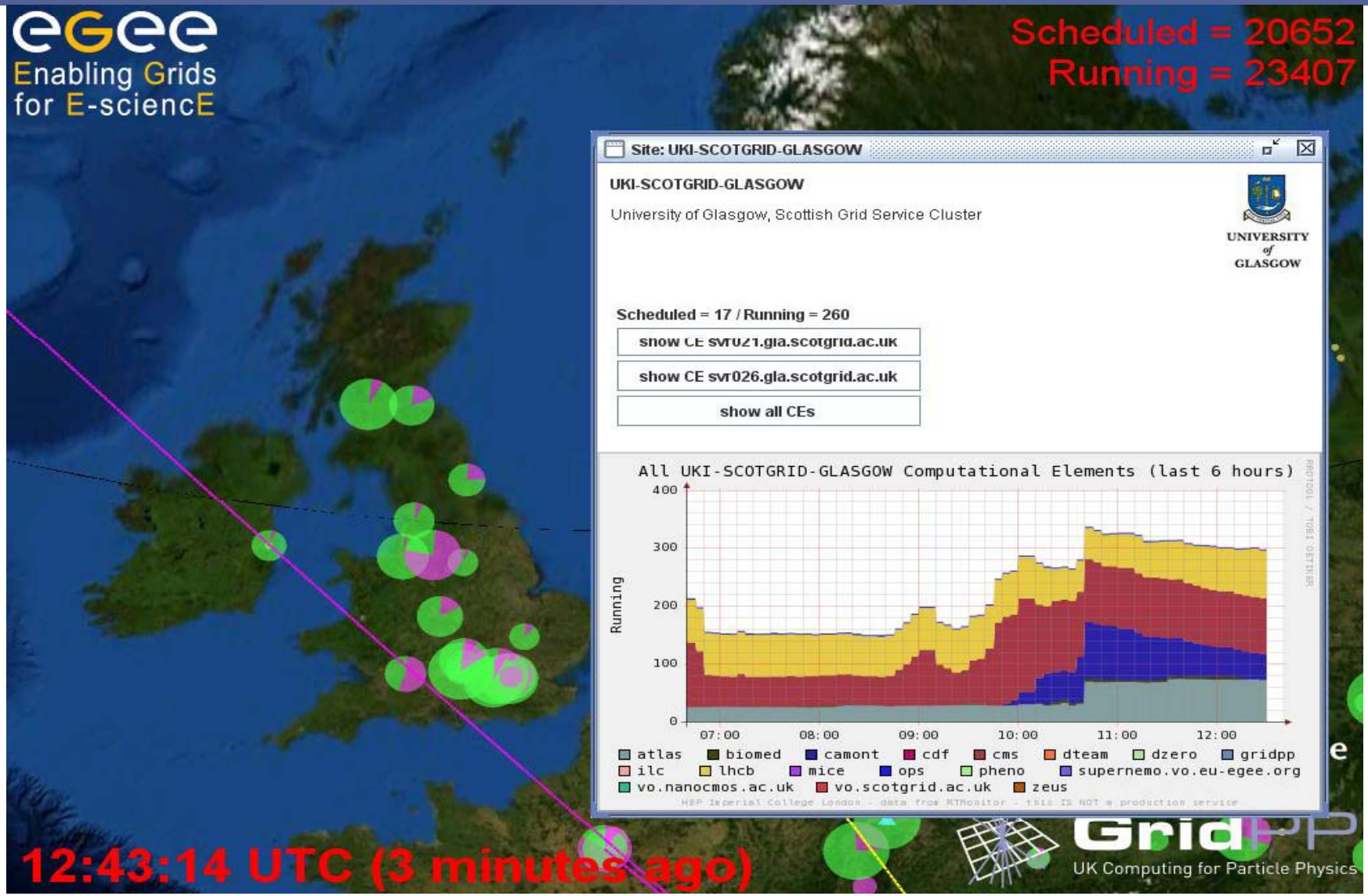
3D Client


eGee
Enabling Grids
for E-sciencE
Scheduled = 44505
Running = 67439
Imperial College
London
GridPP
UK Computing for Particle Physics
10:13:27 UTC (3 minutes ago)

3D client, zoomed in view



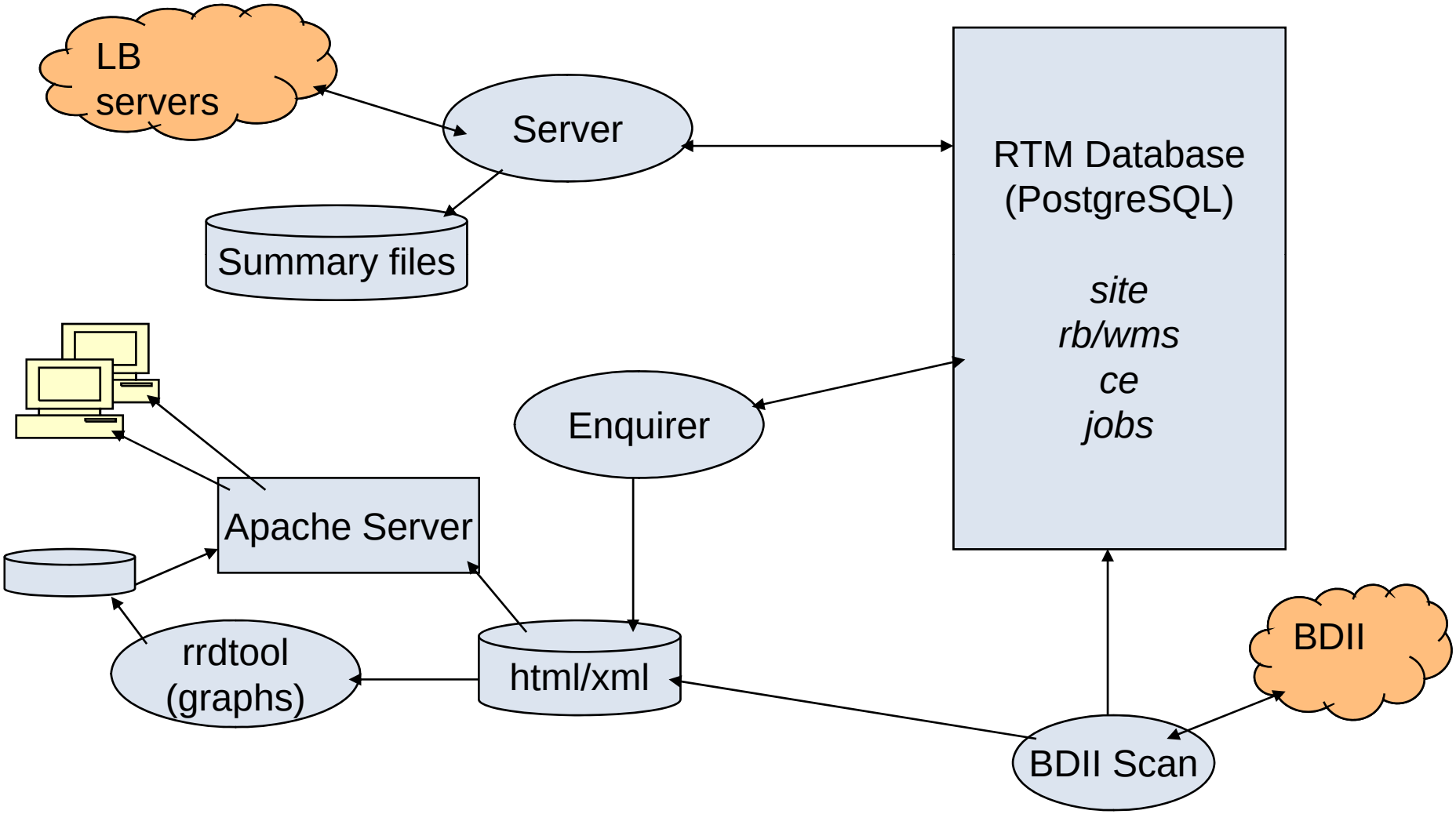
Site Information



Components

- 2D java application/applet client
- 3D client, requires OpenGL support
- Server – gets information from LB databases and stores relevant info in RTM DB by periodically querying LBs.
- BDIIscan – upkeeps BDII/RTM DB synchronisation
- Enquirer – reads RTM DB and writes XML for java clients into Apache file system (so clients query Apache, not the database)
- Resource scanner – checks if LB DB accepts RTM requests (once per day)
- Data analysis (java/Root) – creates CE reports.

RTM Architecture



RTM Data Streams

xml formatted data is made available to clients:

- 2D/3D java clients
- CMS dashboard
- ???

Job Summaries are used by the Grid Observatory (http://www.grid-observatory.org/). The data is publicly available from their website.

Data is also analysed at Imperial, some examples will follow. We have a poster as well [023] on Thu.

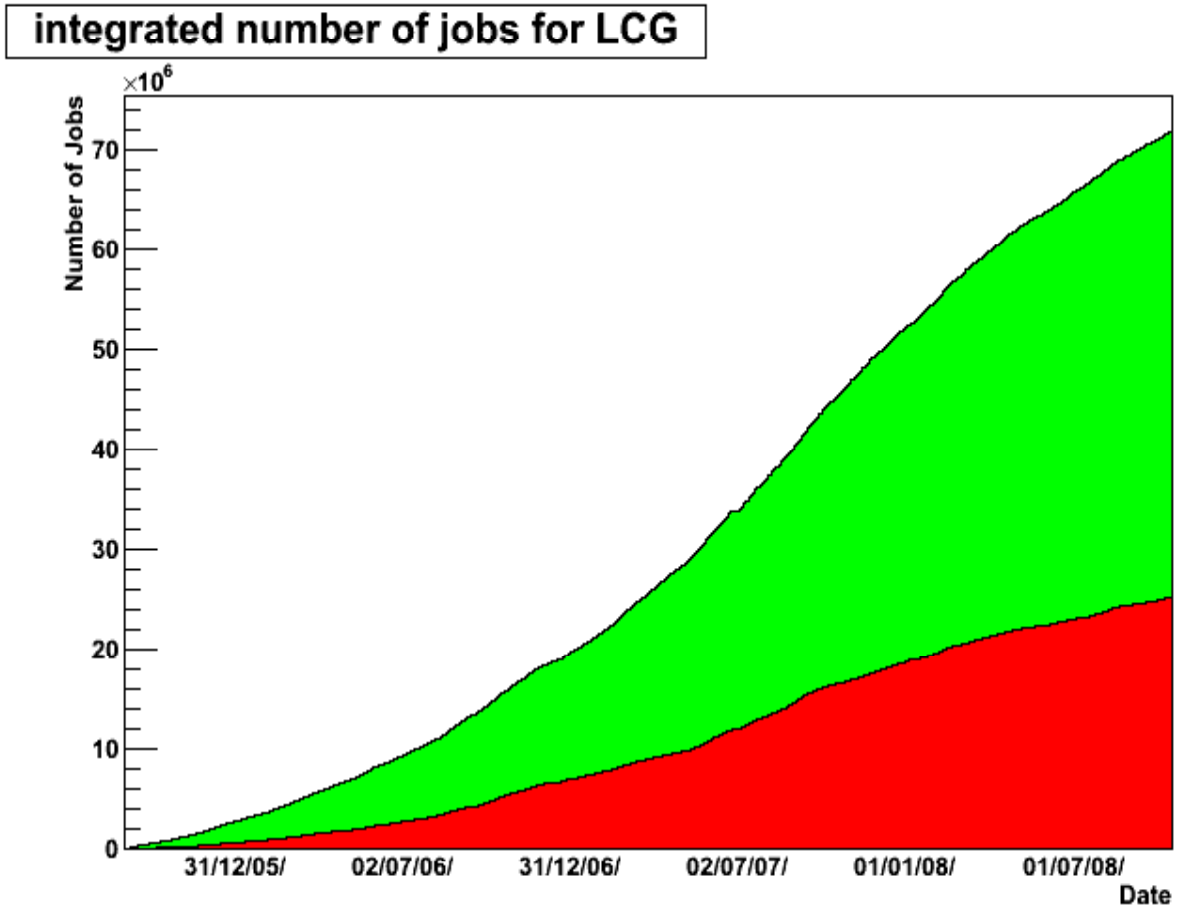
Performance Metrics

We use the following metrics:

- **Job Submissions** in a given day.
- **Cumulative job submissions** in a given day.
- **Job Hours** – total number of hours consumed by a VO on a given day.
- ERAW = $\frac{\text{Total number of successful jobs in a day}}{\text{Total number of jobs in that day}}$
- EUSER = $\frac{\text{Time job is running}}{\text{Time Job is in system}}$.

Growth of the EGEE Grid

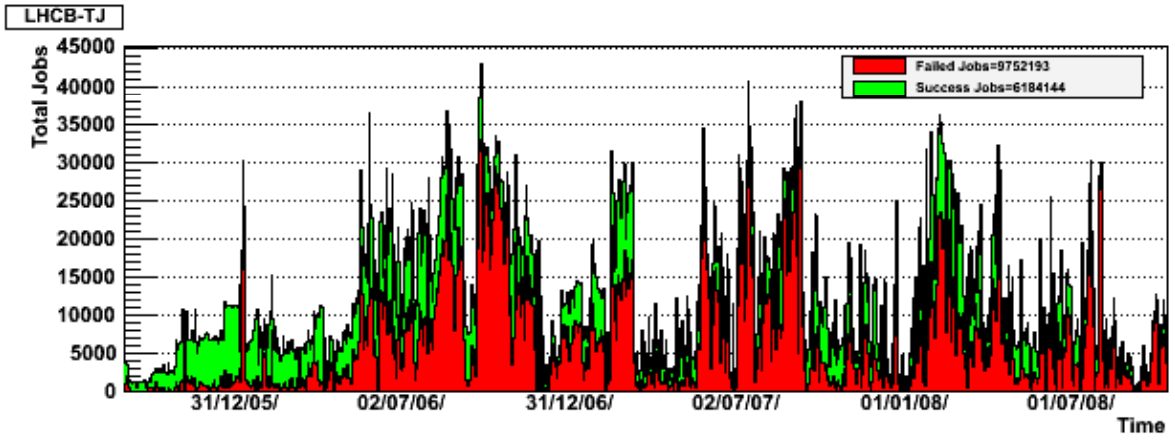
The Cumulative job submissions graph show a greater than linear increase since September 2005.



LHCb

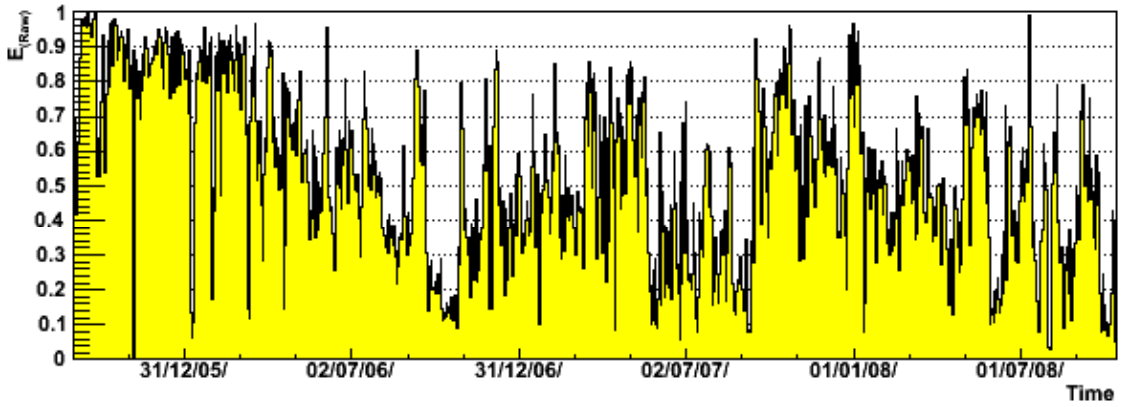
Job submissions vs time

Production runs mid 2006 and 2007. They use pilot jobs which fail quickly if resource is wrong or no waiting job.

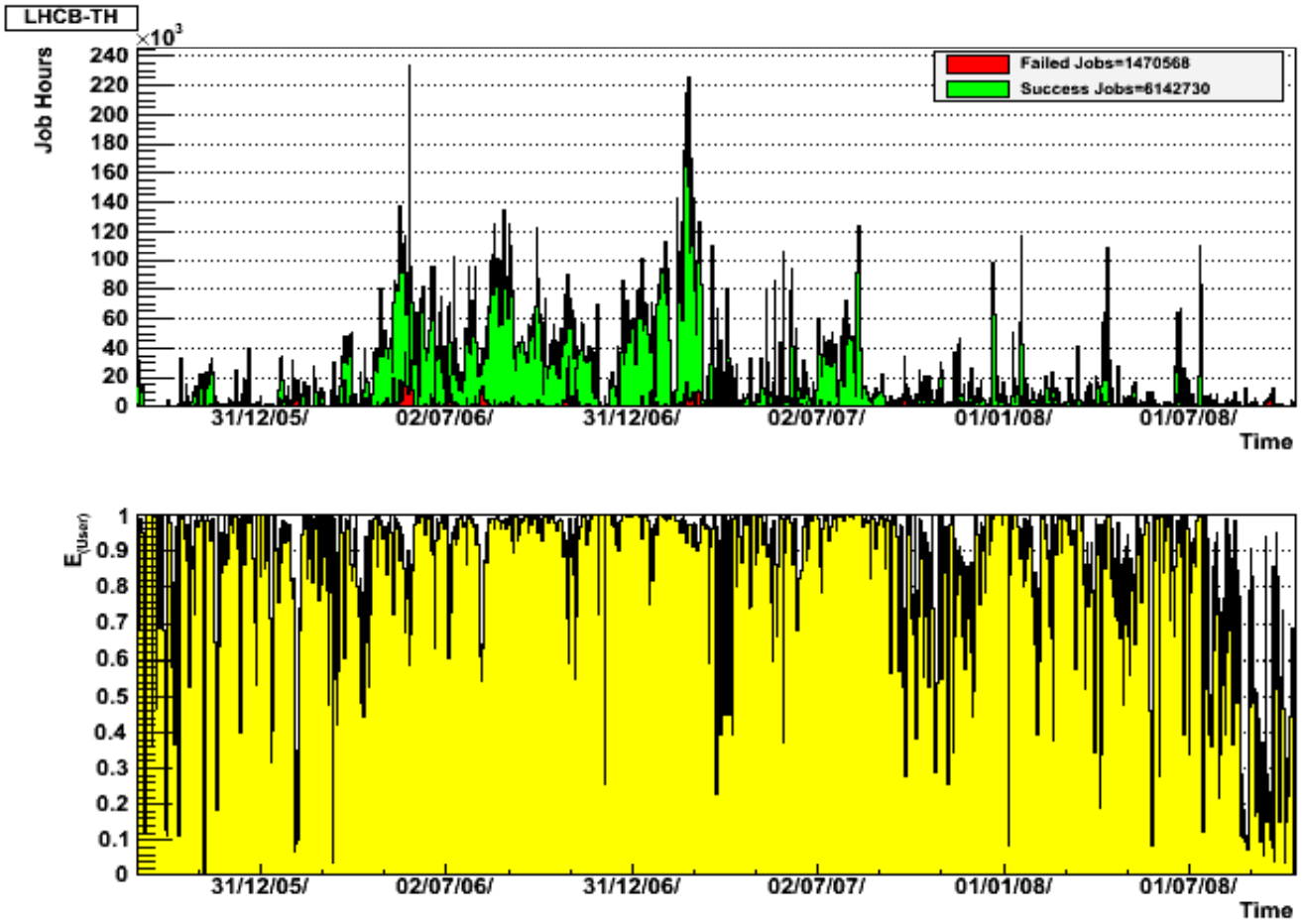


E_{RAW} vs time

Pilot jobs make the RAW efficiency low. Especially when in production runs.



LHCb



Job Hours vs time

Though many failed pilot jobs these take very little time so very little time is spent running failed jobs.

E_{USER} vs time

User efficiency is very good as jobs which run do so quickly and in general there is a high job load from this VO.

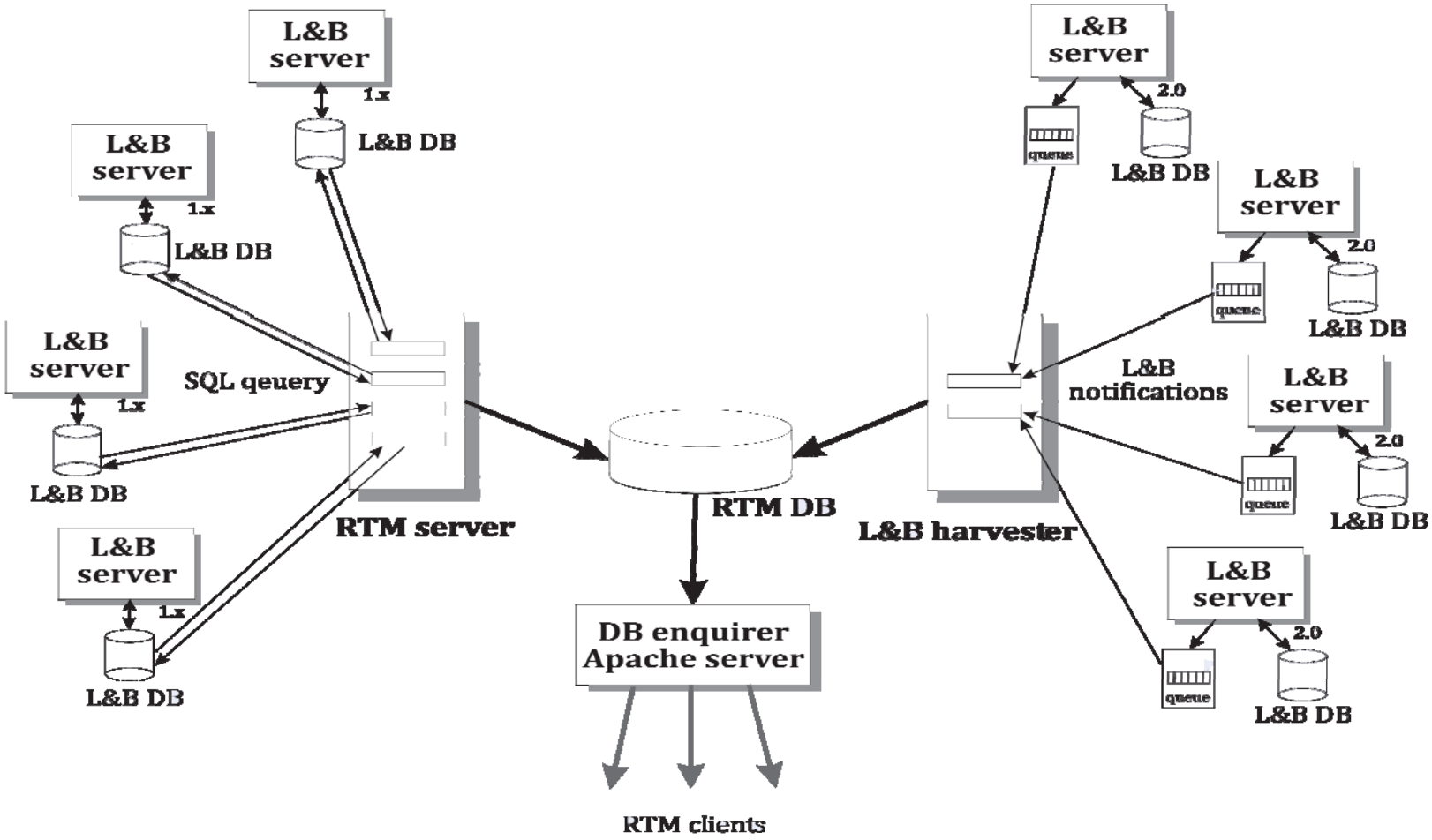
Future....

- We are considering changing the current RTM server-side architecture which relies on direct LB DB access in a pull mode.
- It uses one thread per LB serves which does not scale well
- It suffers from internal changes in LB – current code will not work with next version of LB (2.0)
- Amount of information RTM retrieves is not always optimal
- Privileged DB access make RTM non-portable.

LB notification based architecture

- LB and RTM teams have implemented a prototype which uses push rather than pull mode.
- Replacement RTM module **LB harvester** was developed from scratch, based on **LB notifications**
- The module subscribes to **receive messages** on job state changes from all registered LB's.
- The messages are processed and stored into **internal RTM database**
- Other RTM components remain unchanged
- One harvester thread accepts messages from multiple LBs Limiting factor is the **total number of incoming messages**, not number of connections
- A message on a specific job is generated only on job state change.
- Access to LB uses its stable **public interface** only, not depending on changes in LB internals.
- Uses GSI.

Current and Proposed Systems Combined



We have successfully implemented a working prototype of the harvester:

- **Job state fully synchronized between LB and RTM**
- **Reduced LB load by information push**
- **AuthN/Z with X509 instead of DB password**
- **Promising performance**
- **Better scaling due to “1 thread : many LB’s” architecture**
- **Smooth migration from LB 1.9 to LB 2.0**

Conclusions

- RTM is a reliable monitoring system
- It has been widely used as a dissemination tool to both specialists and non-specialists all over the world.
- Works well on multiple platforms, however there are still issues with some graphics cards and OpenGL support
- There are ongoing discussions with EGEE on how to improve the tool