ATLAS Grid Computing: Distributed Analysis and Monitoring

Dario Barberis

(CERN & Genoa University/INFN)

ATLAS Computing Co-ordinator

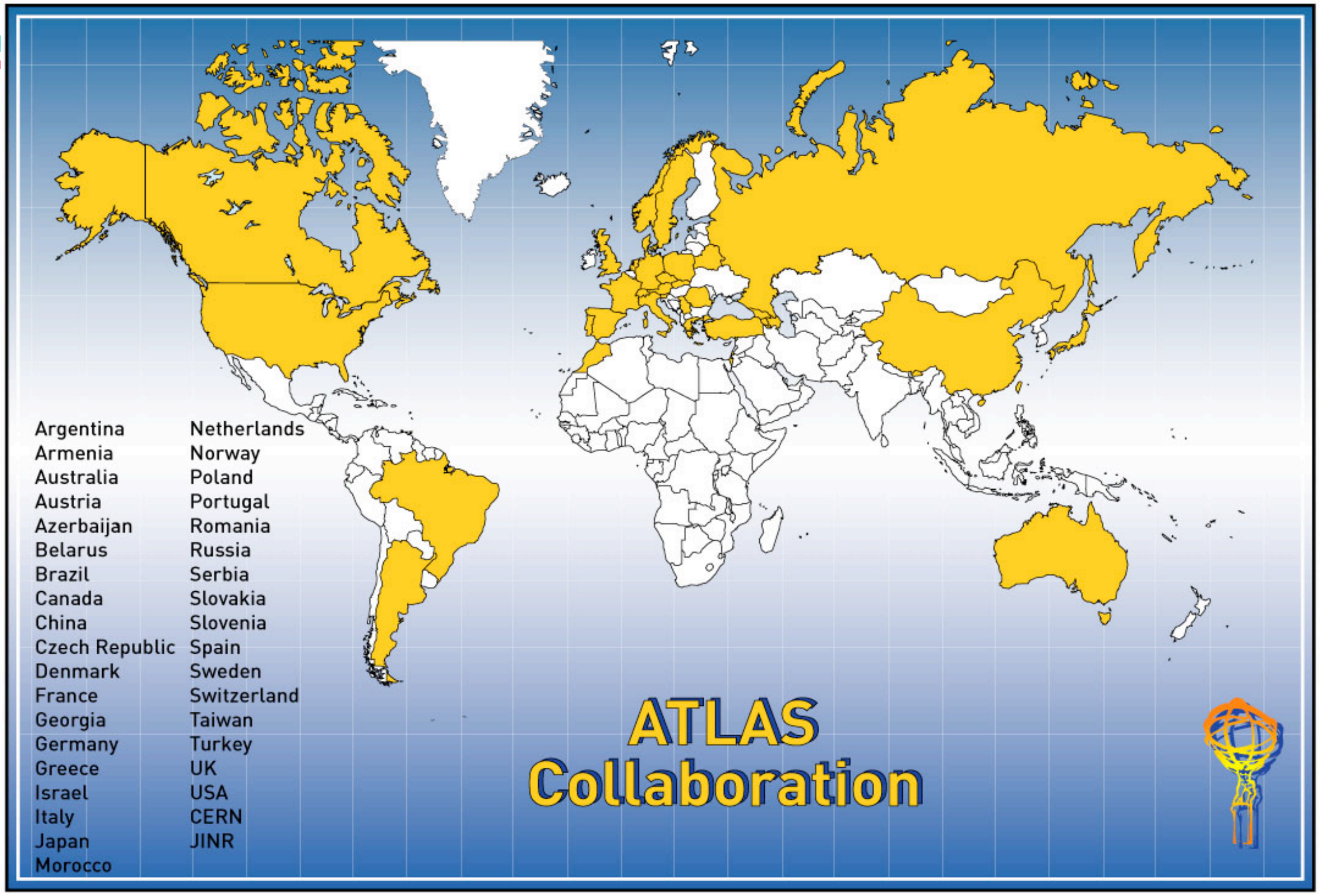
Outline

- Introduction
  - The ATLAS Collaboration
  - Setting the scale: event sizes, rates, distribution
- ATLAS Grid computing architecture
  - Data Management
  - Production System
- Distributed Analysis
  - Principles
  - Experience
- Monitoring
  - Jobs and CPUs
  - Data management
- Conclusions

The ATLAS Collaboration

(As of September 2006)

35 Countries
161 Institutions
1650 Scientific Authors total
(1300 with a PhD, for M&O share)



Argentina	Netherlands
Armenia	Norway
Australia	Poland
Austria	Portugal
Azerbaijan	Romania
Belarus	Russia
Brazil	Serbia
Canada	Slovakia
China	Slovenia
Czech Republic	Spain
Denmark	Sweden
France	Switzerland
Georgia	Taiwan
Germany	Turkey
Greece	UK
Israel	USA
Italy	CERN
Japan	JINR
Morocco	

Albany, Alberta, NIKHEF Amsterdam, Ankara, LAPP Annecy, Argonne NL, Arizona, UT Arlington, Athens, NTU Athens, Baku,
IFAE Barcelona, Belgrade, Bergen, Berkeley LBL and UC, Humboldt U Berlin, Bern, Birmingham, Bologna, Bonn, Boston, Brandeis,
Bratislava/SAS Kosice, Brookhaven NL, Buenos Aires, Bucharest, Cambridge, Carleton, Casablanca/Rabat, CERN, Chinese Cluster, Chicago, Clermont-Ferrand, Columbia, NBI Copenhagen, Cosenza, INP Cracow, FPNT Cracow, DESY, Dortmund, TU Dresden,
JINR Dubna, Duke, Frascati, Freiburg, Geneva, Genoa, Giessen, Glasgow, LPSC Grenoble, Technion Haifa, Hampton, Harvard, Heidelberg, Hiroshima, Hiroshima IT, Indiana, Innsbruck,
Iowa SU, Irvine UC, Istanbul Bogazici, KEK, Kobe, Kyoto, Kyoto UE, Lancaster, UN La Plata, Lecce, Lisbon LIP, Liverpool, Ljubljana, QMW London, RHBNC London, UC London, Lund,
UA Madrid, Mainz, Manchester, Mannheim, CPPM Marseille, Massachusetts, MIT, Melbourne, Michigan, Michigan SU, Milano, Minsk NAS, Minsk NCPHEP, Montreal, McGill Montreal,
FIAN Moscow, ITEP Moscow, MEPhI Moscow, MSU Moscow, Munich LMU,
MPI Munich, Nagasaki IAS, Naples, New Mexico, New York U, Nijmegen, BINP Novosibirsk, Ohio SU, Okayama, Oklahoma, Oklahoma SU, Oregon, LAL Orsay, Osaka, Oslo, Oxford,
Paris VI and VII, Pavia, Pennsylvania, Pisa, Pittsburgh, CAS Prague,
CU Prague, TU Prague, IHEP Protvino, Ritsumeikan, UFRJ Rio de Janeiro, Rochester, Rome I, Rome II, Rome III,
Rutherford Appleton Laboratory, DAPNIA Saclay, Santa Cruz UC, Sheffield, Shinshu, Siegen, Simon Fraser Burnaby,
Southern Methodist Dallas, NPI Petersburg, SLAC, Stockholm, KTH Stockholm, Stony Brook, Sydney, AS Taipei, Tbilisi, Tel Aviv, Thessaloniki, Tokyo ICEPP, Tokyo MU, Toronto,
TRIUMF, Tsukuba, Tufts, Udine, Uppsala, Urbana UI, Valencia, UBC Vancouver, Victoria, Washington, Weizmann Rehovot, Wisconsin, Wuppertal, Yale, Yerevan

ATLAS Data Collection

- Protons flying in opposite directions will collide with a centre-of-mass energy of 14 TeV (~14000 times the proton rest mass) in the centre of the ATLAS detector
- Each such collision produces several (tens of) particles that are absorbed and detected by the ATLAS detector
- The ensemble of the electronic signals produced in all detector components by a single collision is called an “event”
- Events can take place at rates up to 40 MHz, but “interesting” ones will occur much more rarely (100-1000 Hz)
- The online data acquisition system will collect together all signals that belong to the same event and select “interesting” ones (max. rate 200 Hz, limited by bandwidth and offline processing)
- These events are sent to the CERN computing centre (Tier-0) for processing and distribution

Event data flow from online to offline

- Events are written in "ByteStream" format by the Event Filter farm in ~2 GB files
  - ~1000 events/file (nominal size is 1.6 MB/event)
  - 200 Hz trigger rate (independent of luminosity)
  - Currently several streams are foreseen:
    - Express stream with "most interesting" events to be processed immediately
    - ~5 event streams, separated by trigger signature (e.g. muons, electromagnetic, hadronic jets, taus, minimum bias)
    - Calibration events
    - "Trouble maker" events (for debugging)
  - One 2-GB file every 5 seconds will be available from the Event Filter
  - Data will be transfered to the Tier-0 input buffer at 320 MB/s (average)

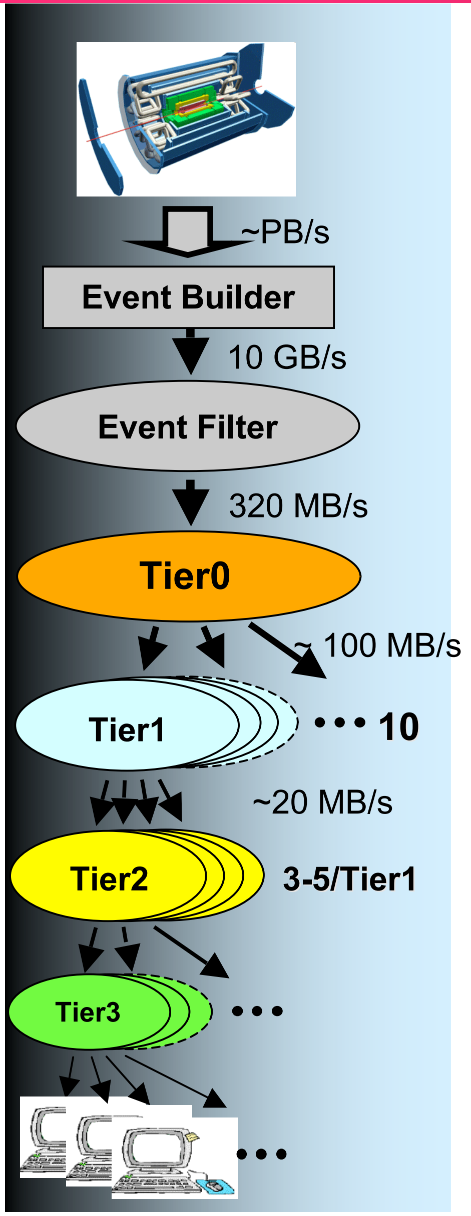
Event Data Model

- RAW:
  - "ByteStream" format, ~1.6 MB/event
- ESD (Event Summary Data):
  - Full output of reconstruction in object (POOL/ROOT) format:
    - Tracks (and their hits), Calo Clusters, Calo Cells, combined reconstruction objects etc.
  - Nominal size 1 MB/event initially, to decrease as the understanding of the detector improves
    - Compromise between "being able to do everything on the ESD" and "not enough disk space to store too large events"
- AOD (Analysis Object Data):
  - Summary of event reconstruction with "physics" (POOL/ROOT) objects:
    - electrons, muons, jets, etc.
  - Nominal size 100 kB/event
- TAG:
  - Database used to quickly select events in AOD and/or ESD files

Computing Model: central operations

- Tier-0:
  - Copy RAW data to Castor tape for archival
  - Copy RAW data to Tier-1s for storage and subsequent reprocessing
  - Run first-pass calibration/alignment (within 24 hrs)
  - Run first-pass reconstruction (within 48 hrs)
  - Distribute reconstruction output (ESDs, AODs & TAGS) to Tier-1s
- Tier-1s:
  - Store and take care of a fraction of RAW data (forever)
  - Run "slow" calibration/alignment procedures
  - Rerun reconstruction with better calib/align and/or algorithms
  - Distribute reconstruction output to Tier-2s
  - Keep current versions of ESDs and AODs on disk for analysis
- Tier-2s:
  - Run simulation (and calibration/alignment when appropriate)
  - Keep current versions of AODs on disk for analysis

Data replication and distribution



In order to provide a reasonable level of data access for analysis, it is necessary to replicate the ESD, AOD and TAGs to Tier-1s and Tier-2s.

RAW:

- Original data at Tier-0
- Complete replica distributed among all Tier-1
  - Randomized dataset to make reprocessing more efficient

ESD:

- ESDs produced by primary reconstruction reside at Tier-0 and are exported to 2 Tier-1s
- Subsequent versions of ESDs, produced at Tier-1s (each one processing its own RAW), are stored locally and replicated to another Tier-1, to have globally 2 copies on disk

AOD:

- Completely replicated at each Tier-1
- Partially replicated to Tier-2s (~1/3 – 1/4 in each Tier-2) so as to have at least a complete set in the Tier-2s associated to each Tier-1
  - Every Tier-2 specifies which datasets are most interesting for their reference community; the rest are distributed according to capacity

TAG:

- TAG databases are replicated to all Tier-1s (Oracle)
- Partial replicas of the TAG will be distributed to Tier-2 as Root files
  - Each Tier-2 will have at least all Root files of the TAGs that correspond to the AODs stored there

Samples of events of all types can be stored anywhere, compatibly with available disk capacity, for particular analysis studies or for software (algorithm) development.

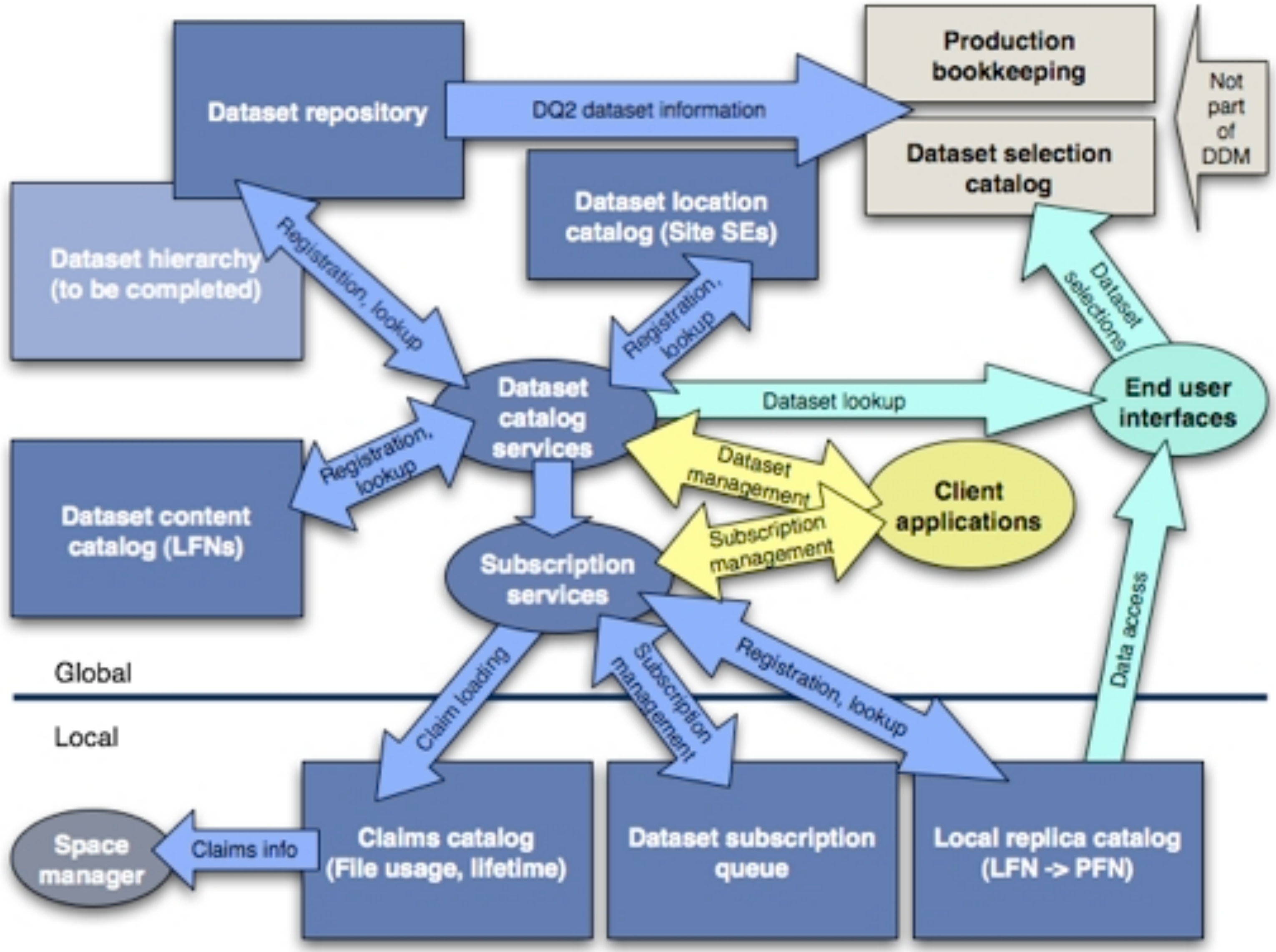
ATLAS Grid Architecture

- The ATLAS Grid architecture is based on 4 main components:
  - Distributed Data Management (DDM)
  - Distributed Production System (ProdSys)
  - Distributed Analysis (DA)
  - Monitoring and Accounting
- DDM is the central link between all components
  - As data access is needed for any processing and analysis step!
- A first version of the production and data management systems was developed in 2003 and used for DC2 (Data Challenge 2) in 2004 and large-scale productions ("Rome" production) in 2005
  - It relied on Grid middleware functionality, performance and reliability
    - None of which turned out to be available at the time (and also later on)
- In 2005 there was a global re-design of ProdSys and DDM to address the shortcomings of the Grid m/w, and allow easier access to the data for distributed analysis
  - At the same time, the first implementations of DA tools were developed

Distributed Data Management

- ATLAS reviewed all its own Grid distributed systems (data management, production, analysis) during the first half of 2005
  - In parallel with the WLCG “Baseline Services WG” activity
- A new Distributed Data Management System (DDM) was designed, based on:
  - Datasets (rather than files) as the unit of storage, replication and processing
  - Central dataset catalogues
  - Data blocks as units of file storage and replication
  - Distributed file catalogues
  - Automatic data transfer mechanisms using distributed services (dataset subscription system)
- The DDM system allows the implementation of the basic ATLAS Computing Model concepts, as described in the Computing TDR (June 2005):
  - Distribution of raw and reconstructed data from CERN to the Tier-1s
  - Distribution of AODs (Analysis Object Data) to Tier-2 centres for analysis
  - Storage of simulated data (produced by Tier-2s) at Tier-1 centres for further distribution and/or processing
  - End-user access to the data

ATLAS DDM Organization



Central vs Local Services

- One fundamental feature is the presence of distributed file catalogues and (above all) auxiliary services
  - Clearly we cannot ask every single Grid centre to install ATLAS services
  - We decided to install "local" catalogues and services at Tier-1 centres
  - Then we defined "regions" which consist of a Tier-1 and all other Grid computing centres that:
    - Are well (network) connected to this Tier-1
    - Depend on this Tier-1 for ATLAS services (including the file catalogue)
- We believe that this architecture scales to our needs for the LHC data-taking era:
  - Moving several 10000s files/day
  - Supporting up to 100000 organized production jobs/day
  - Supporting the analysis work of >1000 active ATLAS physicists
- In the ideal world (perfect network communication hardware and software) we would not need to define default Tier-1—Tier-2 associations
- In practice, it turns out to be convenient (robust?) to partition the Grid so that there are default (not compulsory) data paths between Tier-1s and Tier-2s
  - FTS channels are installed for these data paths for production use
  - All other data transfers go through normal network routes
- In this model, a number of data management services are installed only at Tier-1s and act also on their "associated" Tier-2s:
  - VO Box
  - FTS channel server (both directions)
  - Local file catalogue (part of DDM/DQ2)

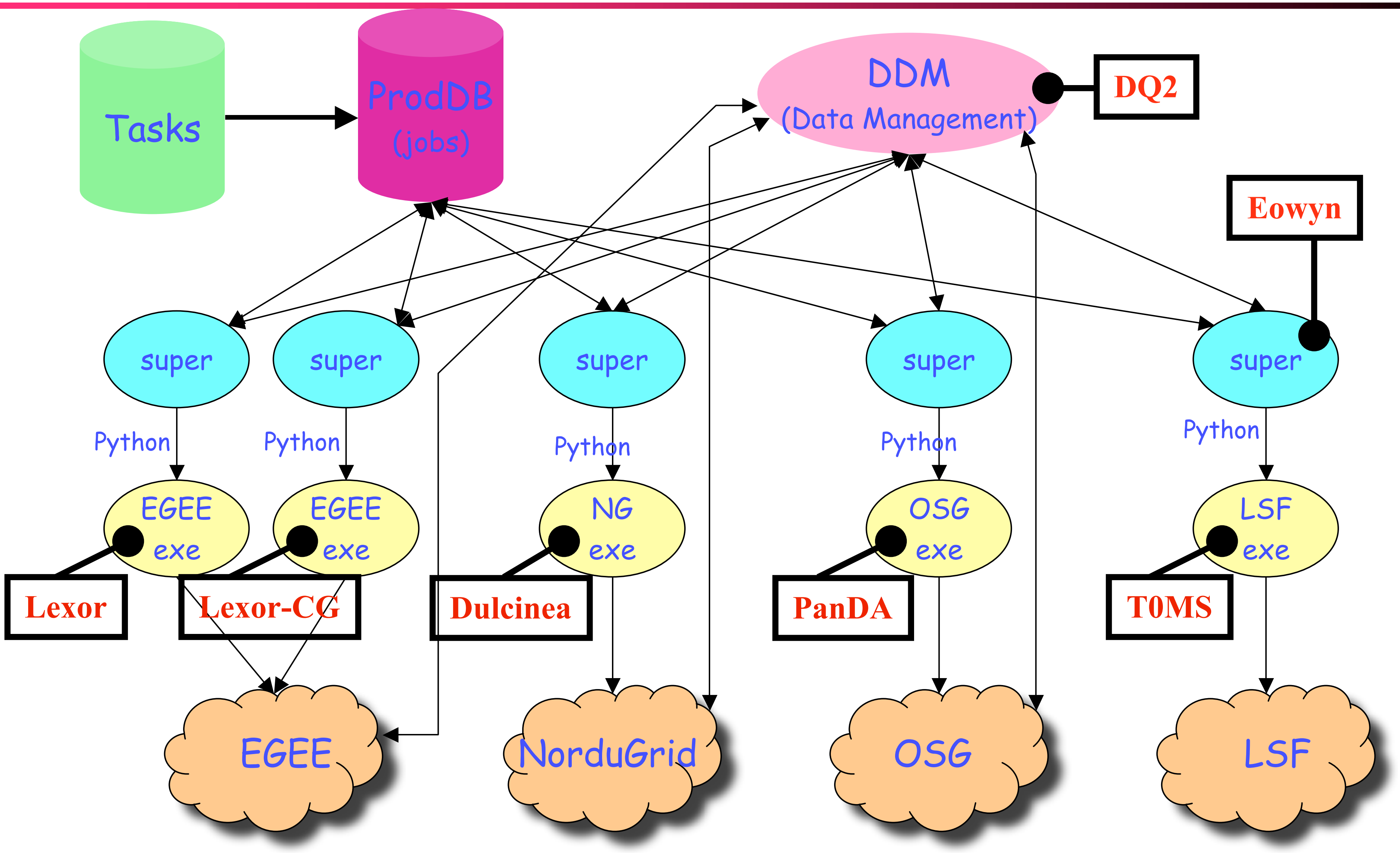
Data Management Considerations

- It is therefore "obvious" that the association must be between computing centres that are "close" from the point of view of:
  - network connectivity (robustness of the infrastructure)
  - geographical location (round-trip time)
- Rates are not a problem:
  - AOD rates (for a full set) from a Tier-1 to a Tier-2 are nominally:
    - 20 MB/s for primary production during data-taking
    - plus the same again for reprocessing from 2008 onwards
    - more later on as there will be more accumulated data to reprocess
  - Upload of simulated data for an "average" Tier-2 (3% of ATLAS Tier-2 capacity) is constant:
    - 0.03 * 0.2 * 200 Hz * 2.6 MB = 3.2 MB/s continuously
- Total storage (and reprocessing!) capacity for simulated data is a concern
  - The Tier-1s must store and reprocess simulated data that match their overall share of ATLAS
    - Some optimization is always possible between real and simulated data, but only within a small range of variations

Job Management: Productions

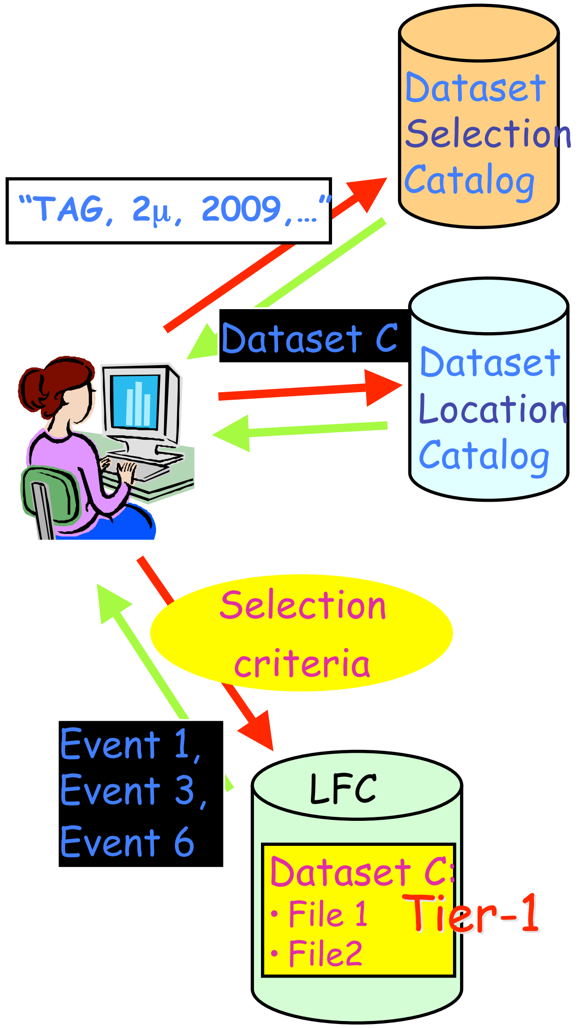
- Once we have data distributed in the correct way (rather than sometimes hidden in the guts of automatic mass storage systems), we can rework the distributed production system to optimise job distribution, by sending jobs to the data (or as close as possible to them)
  - This was not the case previously, as jobs were sent to free CPUs and had to copy the input file(s) to the local WN, from wherever in the world the data happened to be
- Next: make better use of the task and dataset concepts
  - A "task" acts on a dataset and produces more datasets
  - Use bulk submission functionality to send all jobs of a given task to the location of their input datasets
  - Minimise the dependence on file transfers and the waiting time before execution
  - Collect output files belonging to the same dataset to the same SE and transfer them asynchronously to their final locations

ATLAS Production System (2006)



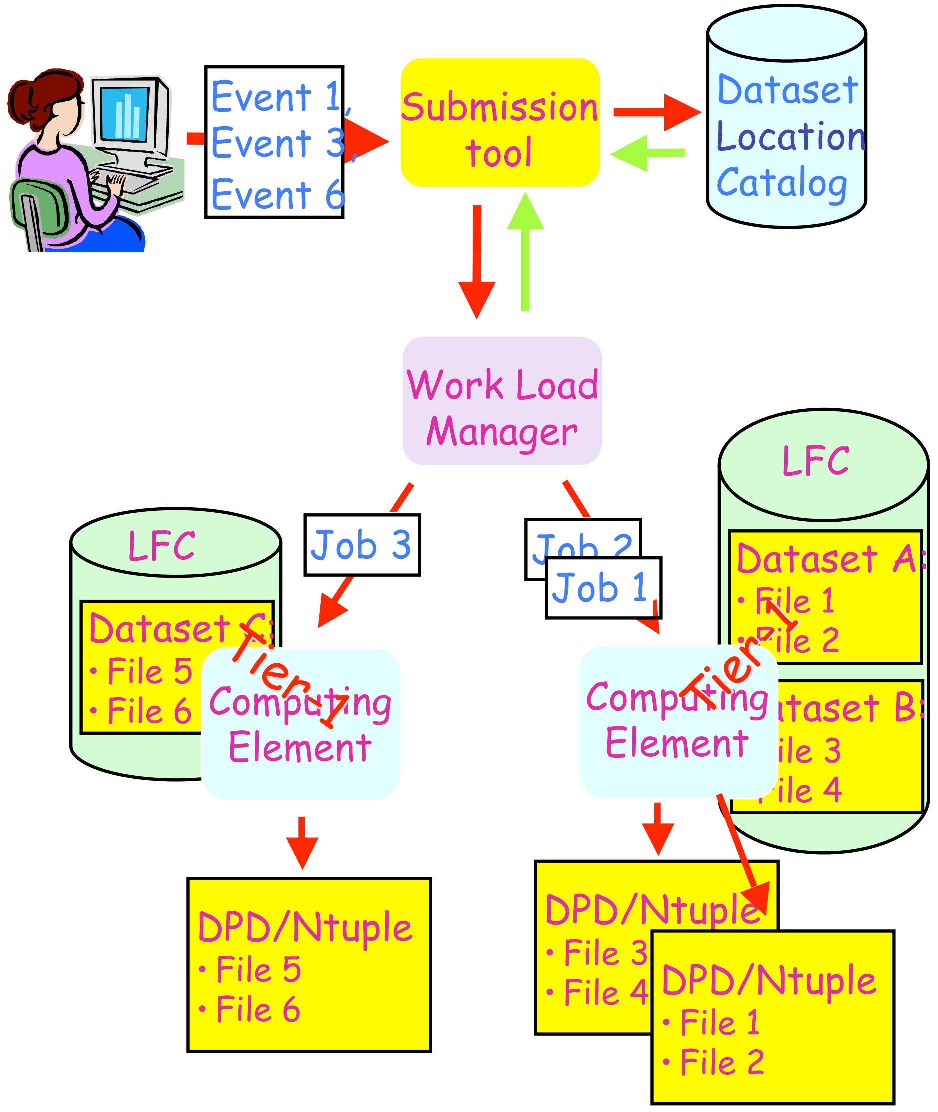
Distributed Analysis: Data selection

- The user opens an analysis session through an interface that allows to execute all analysis operations.
- Logical steps to perform (manually or automatically through a script or web form):
  - Interrogate the Dataset Selection Catalogue, which contains the metadata for each dataset of the desired type, to find interesting datasets
    - Example of a query: give me the list of TAG datasets with 2μ triggers taken in 2009, version x.y.z
  - Through the Dataset Location Catalogue, the site(s) where the dataset is present is (are) found
  - In every Tier-1 there is a local catalogue (Local File Catalogue) that allows to go from datasets to single files that are resident in the cluster of the Tier-1 and associated Tier-2s
  - Apply the selection algorithm on the chosen datasets and produce a list of accepted events



Distributed Analysis: Data processing

- Once events are selected, the user can submit analysis jobs to the Grid :
  - Through the Dataset Location Catalogue sites with datasets containing accepted events are identified
  - In each Tier-1 the Local File Catalogue resolves the datasets into files at the Tier-2 or associated Tier-2s
- A job can produce as output a new event collection, which can be registered as a new dataset in the catalogues for later access
  - A user can also extract from the analysed data (for example in AOD format) files pf Derived Physics Data (such as an n-tupla or any other public or private format) for subsequent interactive local analysis.



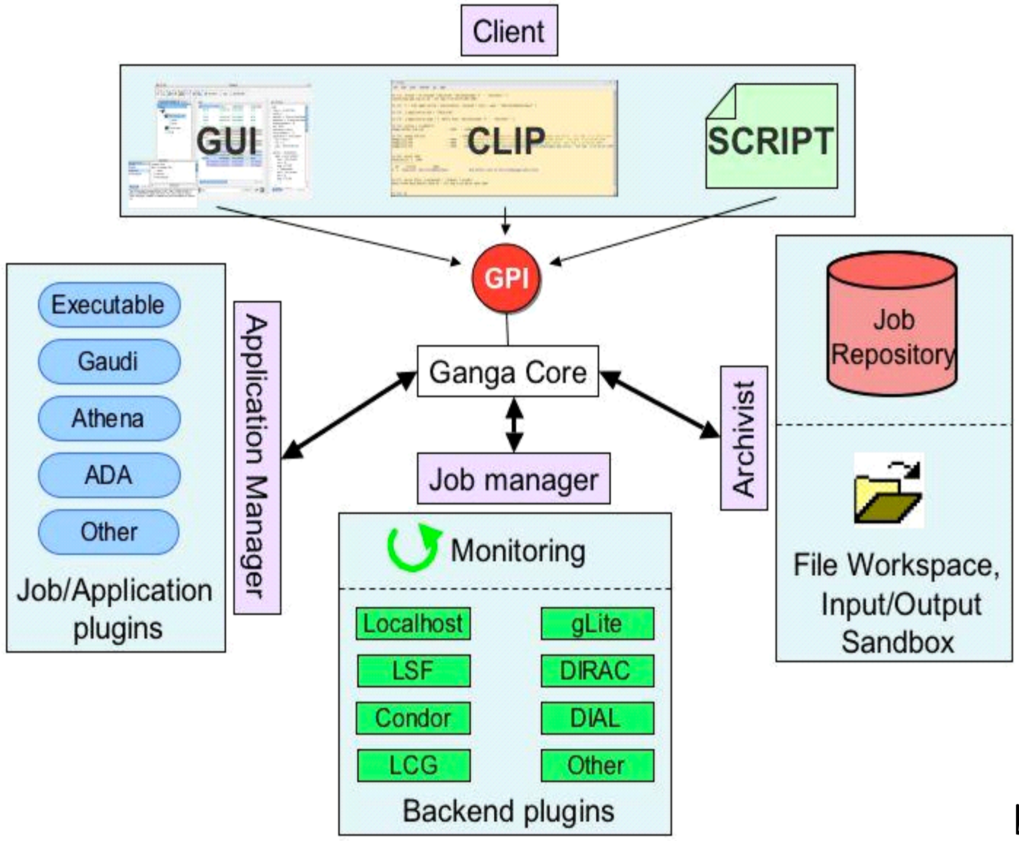
Event 1, Event 3, Event 6
Submission tool
Dataset Location Catalog
Work Load Manager
LFC
Dataset C: • File 5 • File 6
Job 3
Job 2
Job 1
LFC
Dataset A: • File 1 • File 2
Dataset B: • File 3 • File 4
Tier-1
Computing Element
Tier-1
Computing Element
DPD/Ntuple • File 5 • File 6
DPD/Ntuple • File 3 • File 4
DPD/Ntuple • File 1 • File 2

Job Management: Analysis

- The Grid turns out to be a complex environment with different middleware. There is always a competition between central, regional and even local interests. A fully centralized systems does not address everybody's need.
- To address this aspect, tools with complementary features have been developed for different activities.
- The PanDA system developed for production on OSG supports also user analysis jobs using pathena. It is naturally integrated with the DDM.
  - Pathena can generate Athena jobs that act on a dataset and submits them to PanDA on the OSG Grid
- The GANGA framework has been developed to support user analysis on EGEE. A plug-in approach allows to extend the framework to different submission systems and different applications.
- Currently two modes of operation are supported by GANGA:
  - User Analysis: perform AOD analysis jobs on the Grid integrated with the DDM.
  - User production: run small to medium sized production using ATLAS transformations. The program has been evolved from the LJSF submission framework and has been integrated with the DDM.
- The system has been integrated with several submission systems, for ATLAS the main submission system are EGEE and local batch. A PanDA plug-in is being developed and we have discussions on a NorduGrid plug-in.
- In addition, GANGA provides a lot of user-level functionality:
  - Job splitting and bookkeeping
  - Several submission possibilities
  - Collection of output files
- We have a strong collaboration with D-Grid on this issue (Johannes Elmsheuser @ LMU München).
  - **This effort is very important for us.**
  - It would be very interesting for us to further extend this activity in the direction of more interactive analysis (with a potential student).

Distributed Analysis: Ganga

- GANGA (Gaudi/Athena aNd Grid Alliance) is an "easy-to-use" user interface that is meant to drive all analysis related operations:
  - Event selection and identification of sites with selected datasets
  - Splitting of analysis jobs into shorter jobs to be executed in parallel on the Grid
  - Job sumbission and book-keeping
  - Merging of results



- From the implementation side, GANGA:
  - Allows uniform access to local or remote (Grid) resources
  - Offers built-in support for all applications based on the Gaudi/Athena framework, as it is developed by an ATLAS-LHCb collaboration
  - Has a component architecture, which allows easily the extension of functionality
  - Is written in Python, so that it can be integrated with other user scripts

ATLAS Analysis Work Model

1. Job preparation:

Local system (shell)

Prepare JobOptions → Run Athena (interactive or batch) → Get Output

2. Medium-scale testing:

Local system (Ganga)

Prepare JobOptions
Find dataset from DDM
Generate & submit jobs

→

Grid
Run Athena

→

Local system (Ganga)

Job book-keeping
Get Output

3. Large-scale running:

Local system (Ganga)

Prepare JobOptions
Find dataset from DDM
Generate & submit jobs

→

ProdSys
Run Athena on Grid
Store o/p on Grid

→

Local system (Ganga)

Job book-keeping
Access output from Grid
Merge results

Analysis Jobs at Tier-2s

- Analysis jobs must run where the input data files are
  - As transferring data files from other sites may take longer than actually running the job
- Most analysis jobs will take AODs as input for complex calculations and event selections
  - And most likely will output Athena-Aware Ntuples (AAN, to be stored on some close SE), or equivalent, and histograms (to be sent back to the user)
- We assume that people will develop their analyses and run them on reduced samples many many times before launching runs on a complete dataset
  - There will be a large number of failures due to people's code!
- In order to assure execution of analysis jobs with a reasonable turn-around time, we are setting up a priority system that separates centrally organised productions from analysis tasks

Analysis and Grid Services

- There are two layers of services that must work very reliably to support analysis jobs:
  - Grid services, both local (CE, SE, SRM) and at the Tier-1s (LFC, FTS, VObox)
    - Otherwise the common complaint "I cannot get (to) my data" will become even more frequent!
  - ATLAS DDM services at the Tier-1s on top of Grid services (subscriptions, local catalogues)
    - Other common complaint: "My dataset is not getting transferred to its destination"
- Analysis jobs are more sensitive to temporary interruptions of service that production jobs
  - Shorter turn-around time, necessity to operate "now" on "this" dataset
  - The paradigm of sending jobs to the data needs assured knowledge of data location and accessibility
- We are working on improving the robustness of these services, but we are not there yet
  - A lot of baby-sitting and trouble-shooting is still needed
  - Dedicated (wo)manpower for each ATLAS/Tier-1 cloud

Local vs distributed jobs

- In ATLAS one can produce events locally (and then register them), or using Ganga or ProdSys

Tool	Pro	Con
Local Prod	Easy to start Under direct control	Logistics, Bookkeeping, Provenance Limited resources Publication of files (DDM, LFC) Manual and local operation No accounting (user jobs) Tricky validation
Ganga	Easy to learn Automatic book-keeping File publishing (DDM, LFC)	Locale book-keeping Provenance Local operation Accounting still to be implemented (user space)
ProdSys	Bookkeeping, Provenance File publishing (DDM, LFC) Distributed operation Centralized priorities Automatic accounting	System complexity

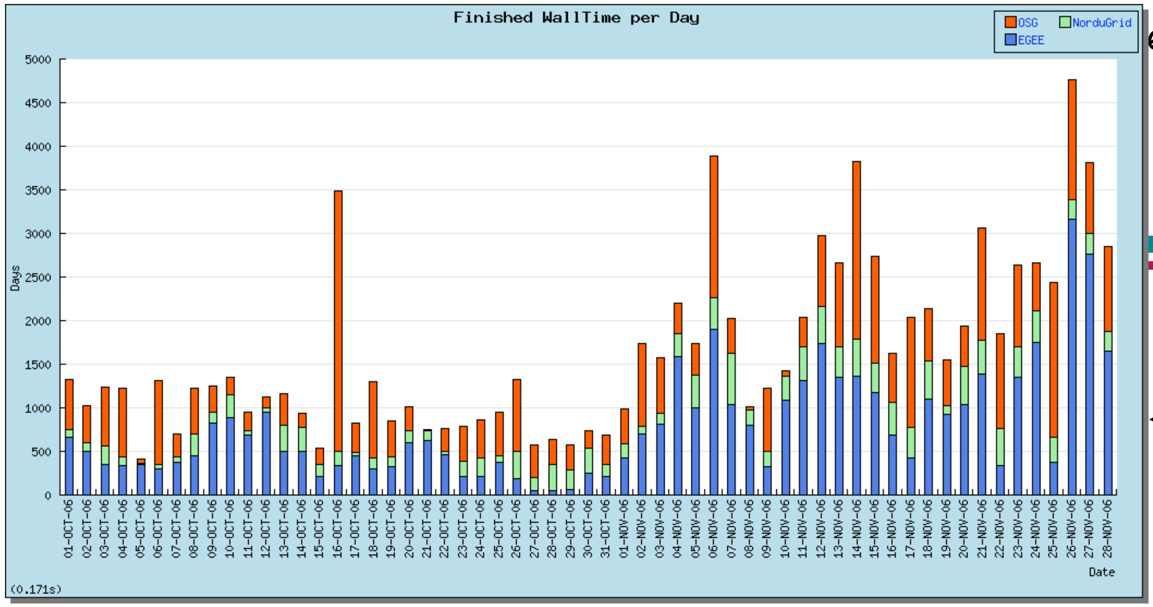
- In practice:
  - Ganga is good for medium-sized productions, limited in time
  - ProdSys is surely better for large productions (>>1000 jobs, repeated several times)

ATLAS Job/CPU Resource Monitoring

- Several job monitoring tools have been developed as part of Grid middleware
  - Unfortunately none of them is complete, reliable, and covers our needs
- We therefore had to develop our own strategy:
  - Develop a robust monitoring tool for ProdSys, based on the production database (ProdDB) (John Kennedy @ LMU München)
    - This is reliable, as it is entirely under our (ATLAS) control
  - Take data from the Grid Operations Centre at RAL
    - Data collected from all EGEE Computing Elements with integration of OSG data
      - Still incomplete as the data transmission system is not yet very reliable
  - Take data from the Imperial College London monitoring system
    - Data collected from the EGEE Resource Brokers
      - Will always be incomplete as there are different ways to submit jobs to EGEE CEs (Condor-G, glide-ins)
  - Take data from the MonaLisa monitoring service that runs on OSG
- Merge all these data and provide a display tool that allows to select data types and time ranges in a user-friendly way
  - The ARDA Dashboard (LCG project in collaboration with other LHC experiments)

ProdSys Monitoring

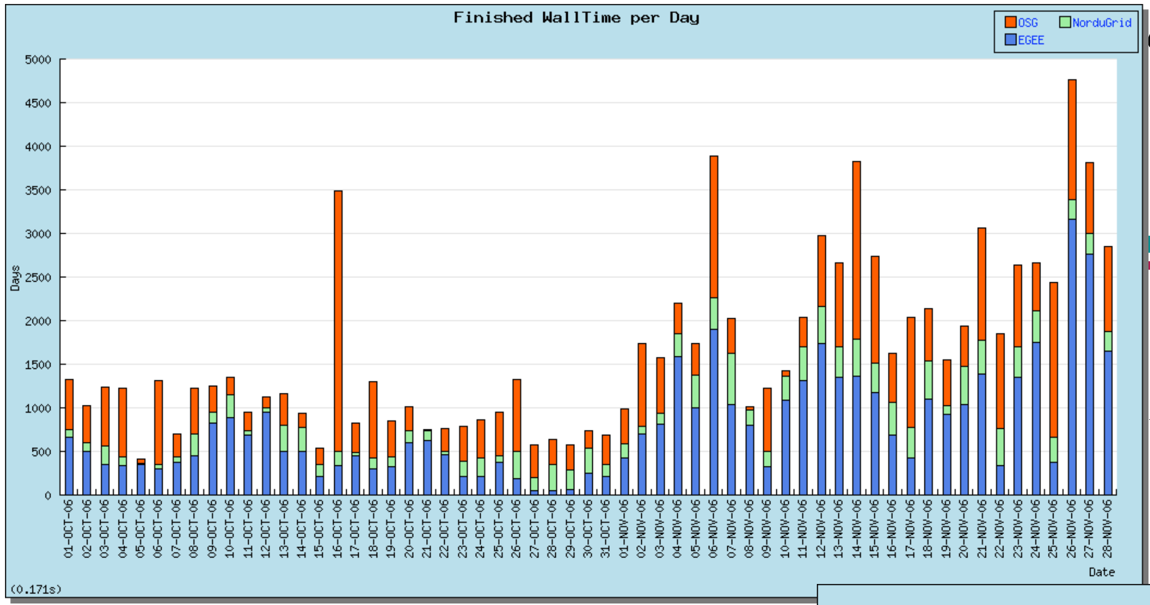
er 2006



1 Oct - 28 Nov 2006
Wall clock time used by successful ProdSys jobs by Grid type

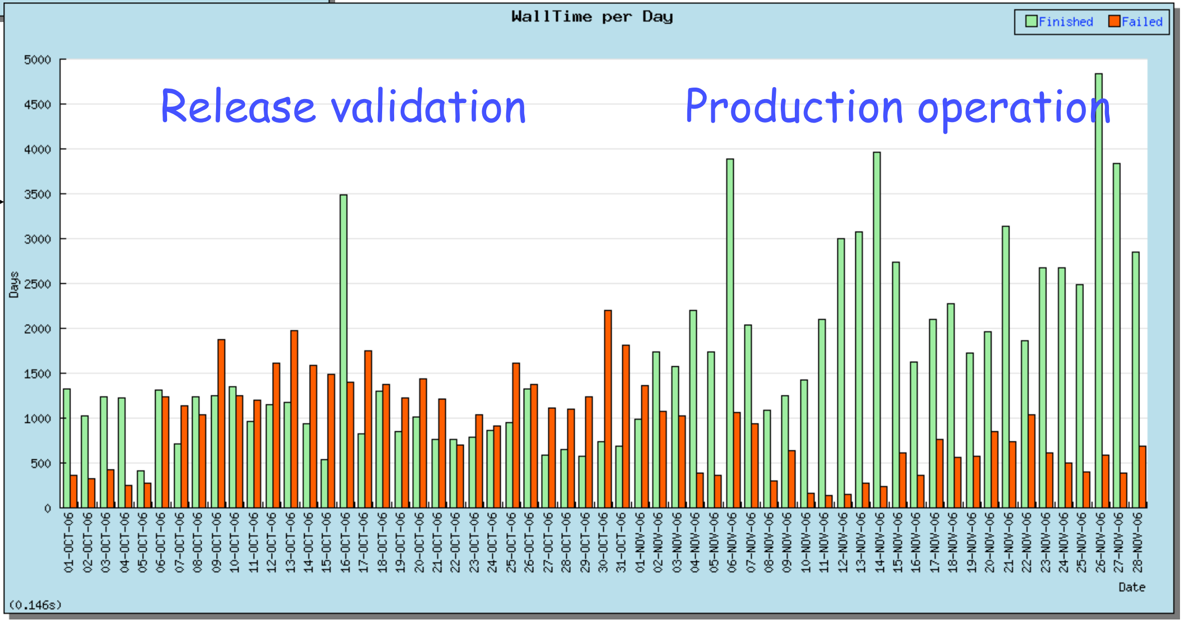
er 2006

ProdSys Monitoring

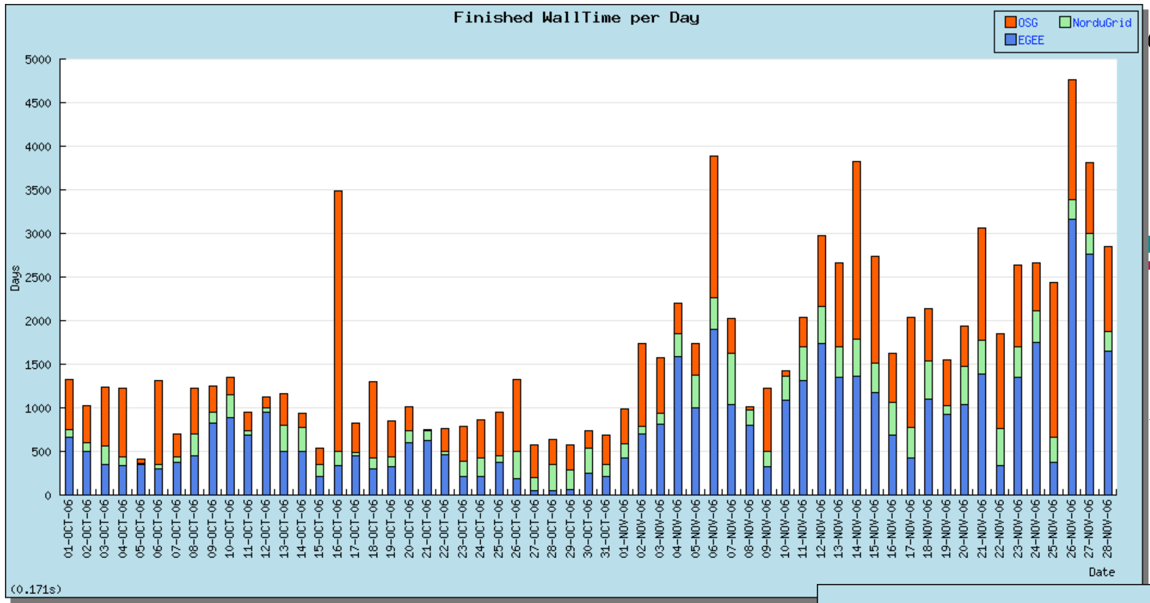


1 Oct - 28 Nov 2006
Wall clock time used by successful ProdSys jobs by Grid type

1 Oct - 28 Nov 2006
Wall clock time used by successful and failed ProdSys jobs

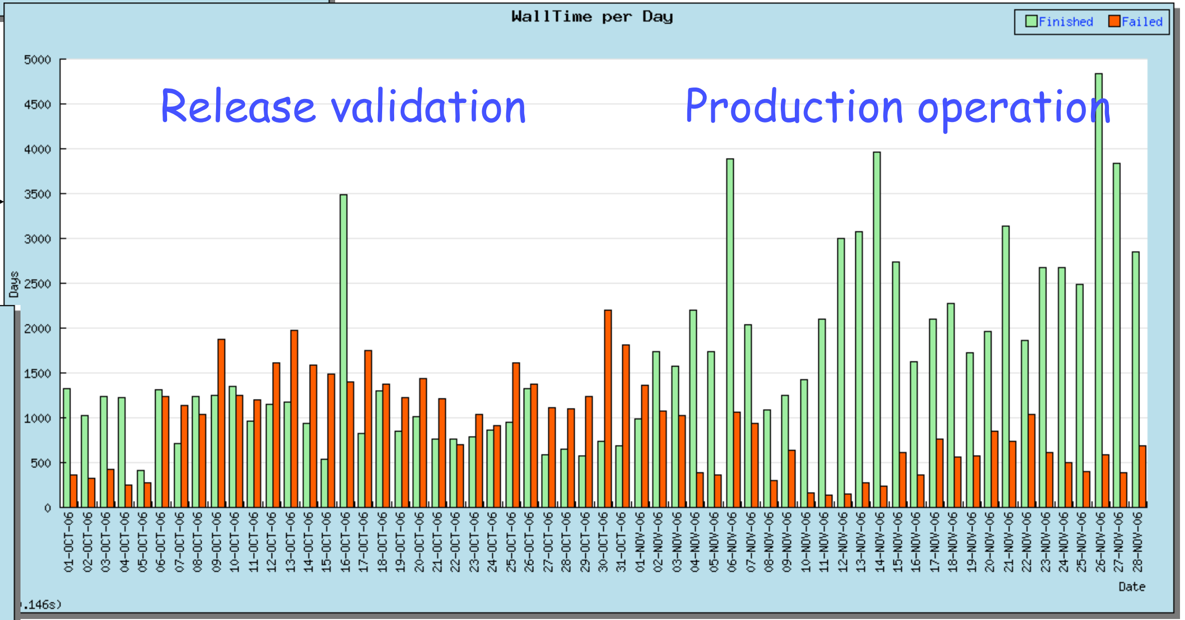


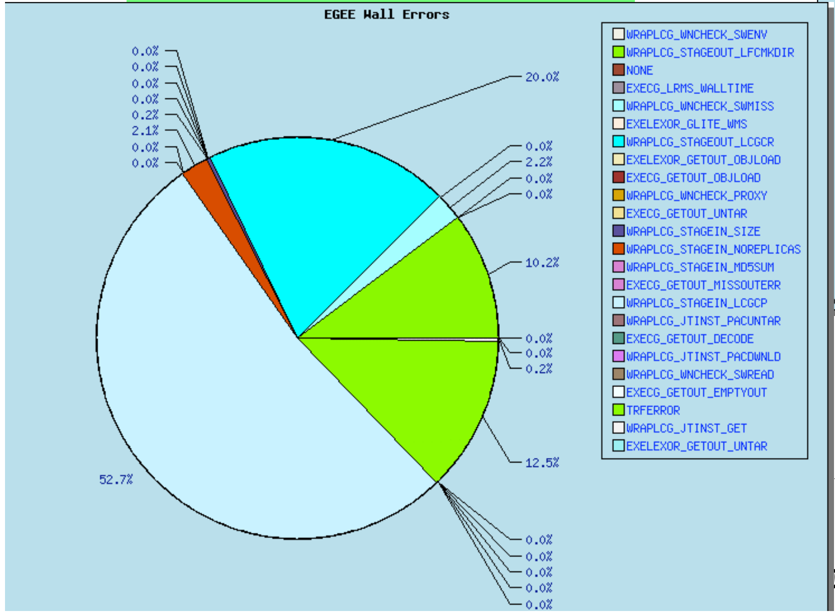
ProdSys Monitoring



1 Oct - 28 Nov 2006
Wall clock time used by successful ProdSys jobs by Grid type

1 Oct - 28 Nov 2006
Wall clock time used by successful and failed ProdSys jobs

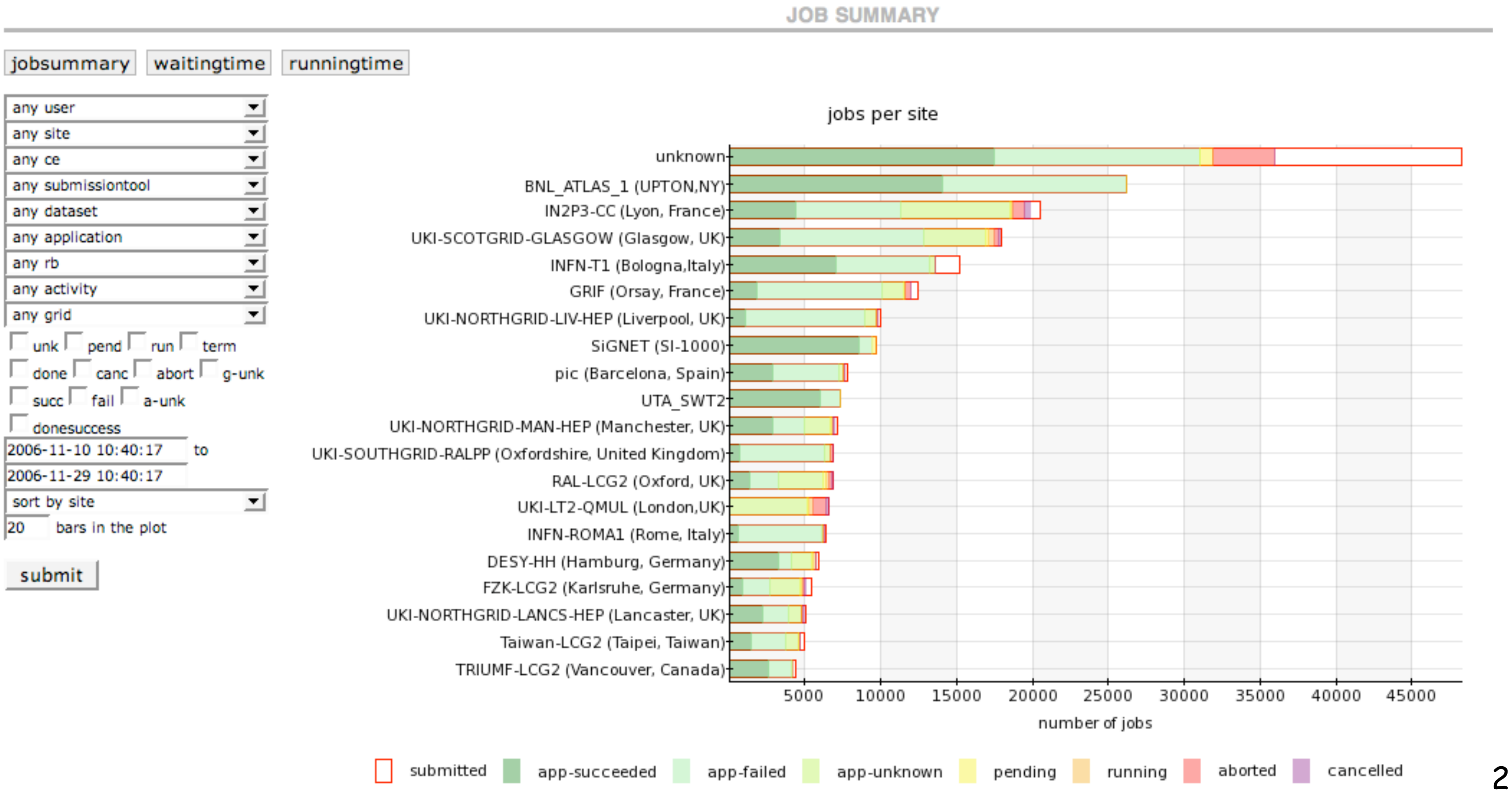




11-28 Nov 2006
Wall clock time used by failed ProdSys jobs on the EGEE Grid: error analysis

Grid Job Monitoring Dashboard

- Example of information retrievable from the ARDA dashboard:
  - Job status per Site/User/CE/SubmissionTool/Dataset/Application/RB/Grid within a given time window
  - Many other combined views are available (see the menus on the left)
  - Clicking on any part of the graph, one can get more detailed info on the jobs


JOB SUMMARY

jobsummary | waitingtime | runningtime

any user
any site
any ce
any submissiontool
any dataset
any application
any rb
any activity
any grid

☐ unk ☐ pend ☐ run ☐ term
☐ done ☐ canc ☐ abort ☐ g-unk
☐ succ ☐ fail ☐ a-unk
☐ donesuccess

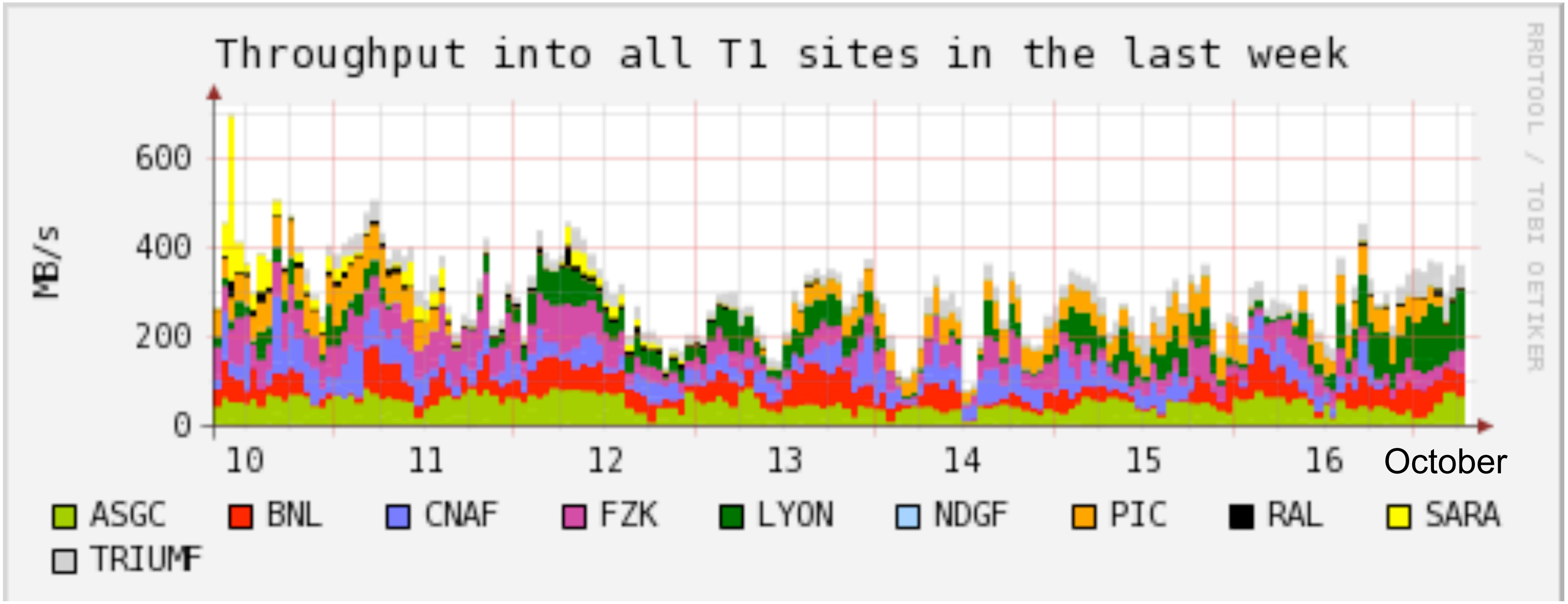
2006-11-10 10:40:17 to
2006-11-29 10:40:17

sort by site
20 bars in the plot

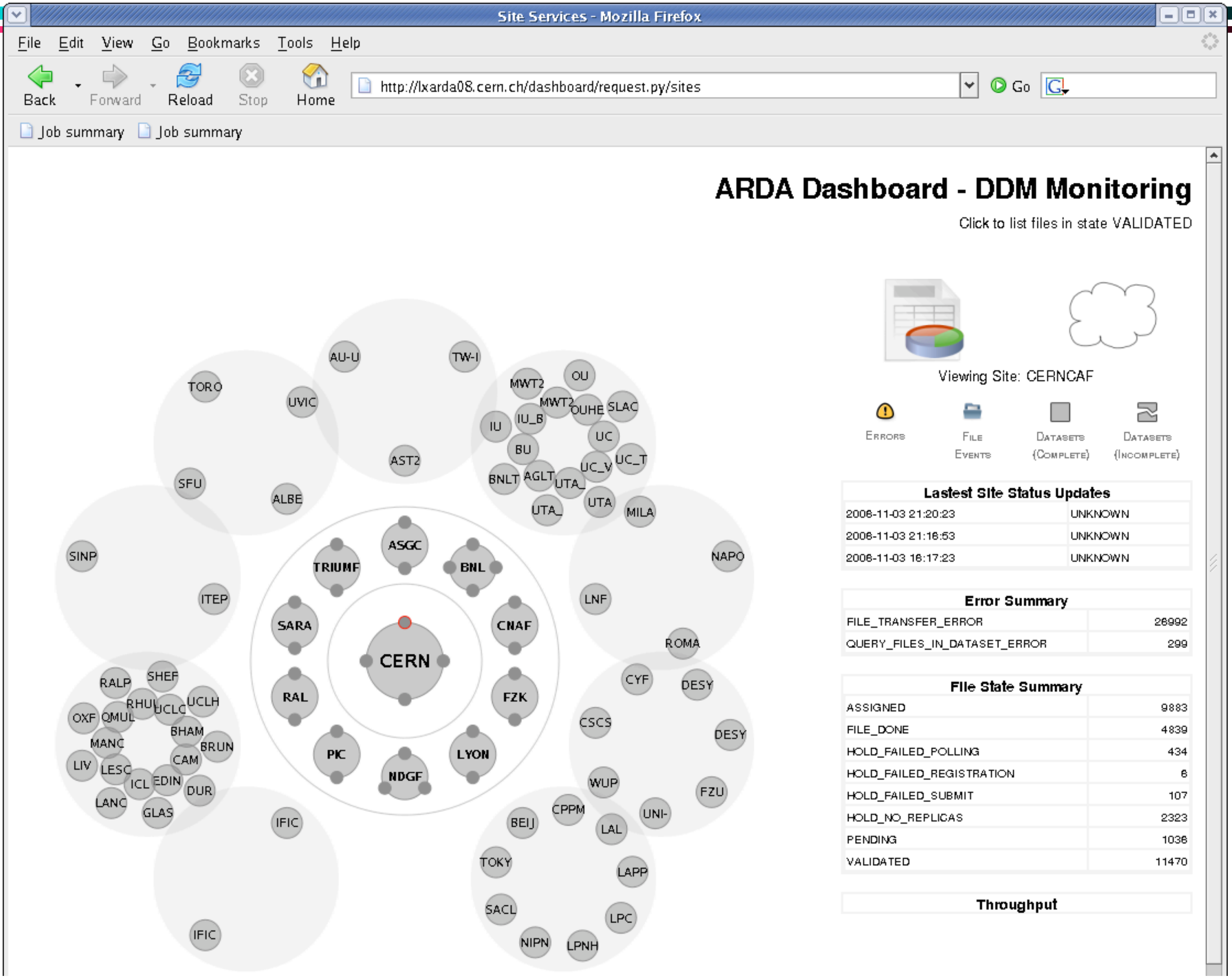
submit

Data Distribution Monitoring

- We also developed a monitoring system for the data distribution tests in the context of SC4 (2006)
  - As always, it was meant for a well-defined function, but it had to be extended to monitor all components of DDM and also site storage and transfer tools
- A completely new (more scalable) version is in preparation and will become the production version in a few weeks
  - It is now integrated with the ARDA dashboard, so users will have the same "look and feel" and use some of the same underlying tools for job and data monitoring



DDM Dashboard


Site Services - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Back Forward Reload Stop Home
http://lxarda08.cern.ch/dashboard/request.py/sites
Go
Job summary
Job summary
CERN
ASGC
BNL
CNAF
FZK
LYON
NDGF
PIC
RAL
SARA
TRIUMF
Errors
File Events
Datasets (Complete)
Datasets (Incomplete)

ARDA Dashboard - DDM Monitoring

Click to list files in state VALIDATED

Viewing Site: CERNCAF

Lastest Site Status Updates

2006-11-03 21:20:23	UNKNOWN
2006-11-03 21:16:53	UNKNOWN
2006-11-03 16:17:23	UNKNOWN

Error Summary

FILE_TRANSFER_ERROR	26992
QUERY_FILES_IN_DATASET_ERROR	299

File State Summary

ASSIGNED	9883
FILE_DONE	4839
HOLD_FAILED_POLLING	434
HOLD_FAILED_REGISTRATION	6
HOLD_FAILED_SUBMIT	107
HOLD_NO_REPLICAS	2323
PENDING	1036
VALIDATED	11470

Throughput

Conclusions

- ATLAS has an integrated programme of Grid tools developments
  - Distributed Data Management (DDM) supports the Production System and Distributed Analysis
- While ProdSys has now been in operation for several years, only this year the Distributed Analysis tools have reached a level of maturity that makes them useful to the Collaboration
  - But the system can only work satisfactorily if the lower layers (Grid middleware and ATLAS components) perform reliably
  - Currently operations (wo)manpower is needed to keep the system working
- Monitoring tools are used to:
  - Find operations problems and address them in real time
  - Investigate the underlying causes of service interruptions, together with the Grid middleware experts, and provide solutions (or at least work-arounds)
- A non-negligible amount of work is still needed in all these areas before the beginning of data-taking in about a year's time
  - Everybody is welcome to contribute to the common effort!