PANDA

FIFE Workshop at FNAL

June 4th, 2013

Maxim Potekhin

BNL

potekhin@bnl.gov

PanDA: Production and Distributed Analysis System

Brief History

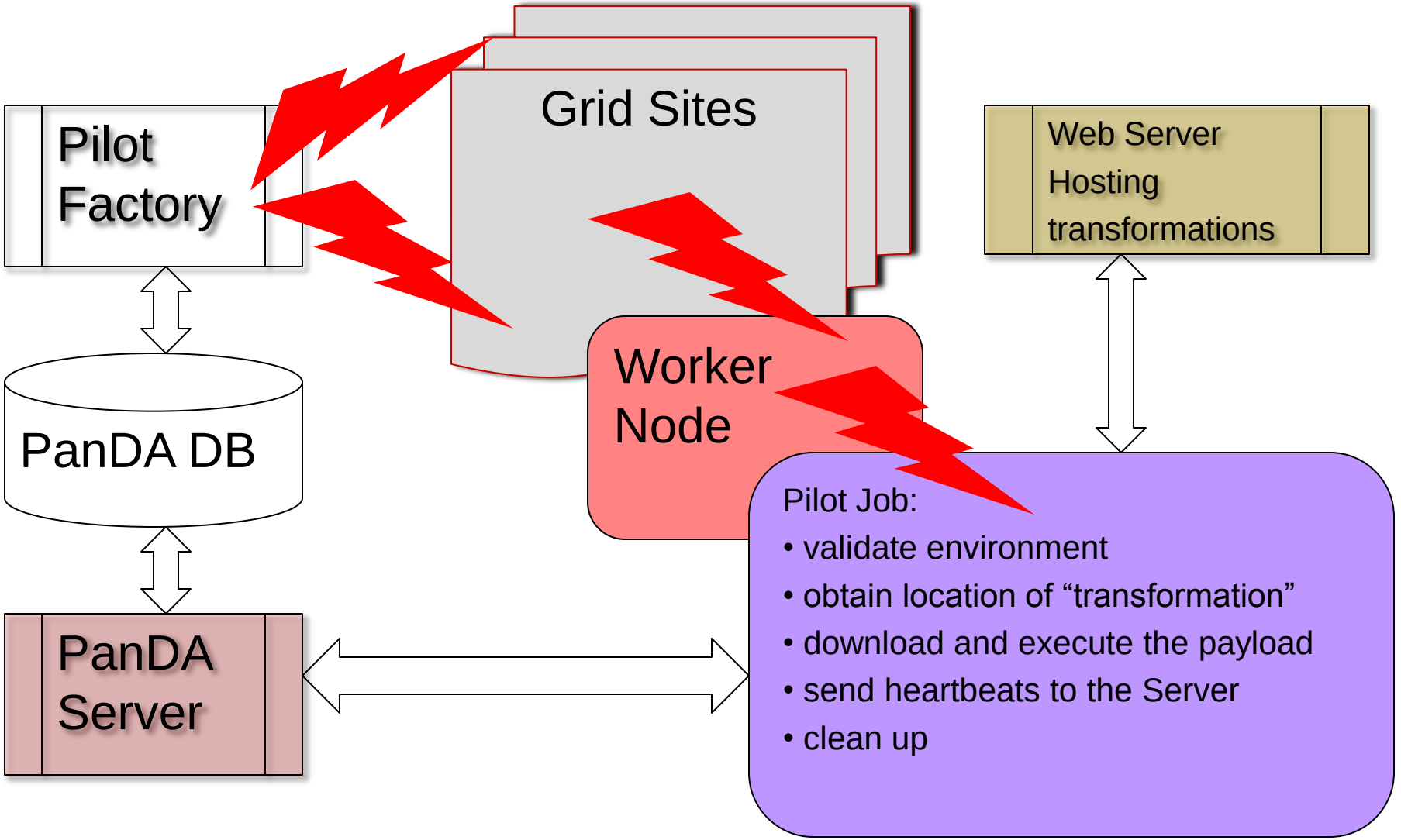
The PanDA system has been developed by US ATLAS (mainly at BNL and UTA) since 2005 to meet ATLAS requirements for a data-driven workload management system for production and distributed analysis capable of operating at the LHC data processing scale. Computational load handled by ATLAS places challenging requirements on throughput, scalability, robustness, efficient resource utilization, minimal operations manpower, and tight integration of data management with processing workflow. In October 2007 PanDA was adopted by the ATLAS Collaboration as the sole system for distributed processing and production across the Collaboration.

Pilot Job Framework

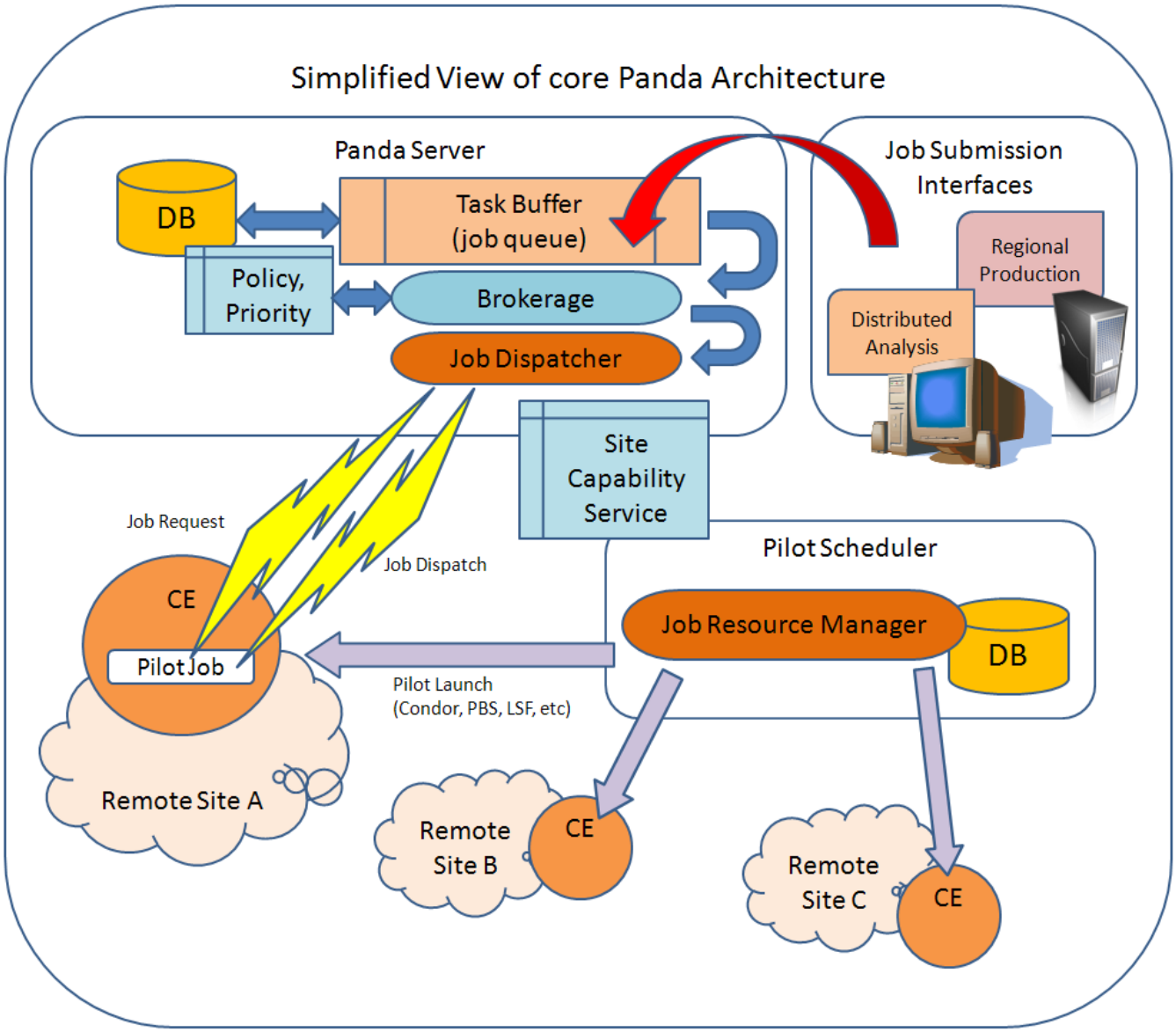
Use of pilot jobs for acquisition of processing resources: payload jobs are assigned to successfully activated and validated “pilots” based on PanDA-managed brokerage criteria. This “late binding” of payload jobs to processing slots prevents latencies and failure modes in slot acquisition from impacting the jobs, and maximizes the flexibility of job allocation to resources based on the dynamic status of processing facilities and job priorities. Payloads are defined as scripts (termed “transformations”) which the pilot job obtains from a URL specified by the PanDA server, in the course of communication between the two.

The pilot is also a principal 'insulation layer' for PanDA, encapsulating the complex heterogeneous environments and interfaces of the grids and facilities on which PanDA operates. This concept is not unique or exclusive to PanDA, it has proven itself in other projects (cf DIRAC, glideinWMS etc)

Lifecycle of the PanDA Pilot (simplified)



Pilots and Job Dispatcher in PanDA



ATLAS: scale and complexity of workload

Main job types in ATLAS, ordered by resource consumption

- Monte Carlo production
- User Analysis
- Group Production
- Validation
- Data processing and re-processing
- Testing

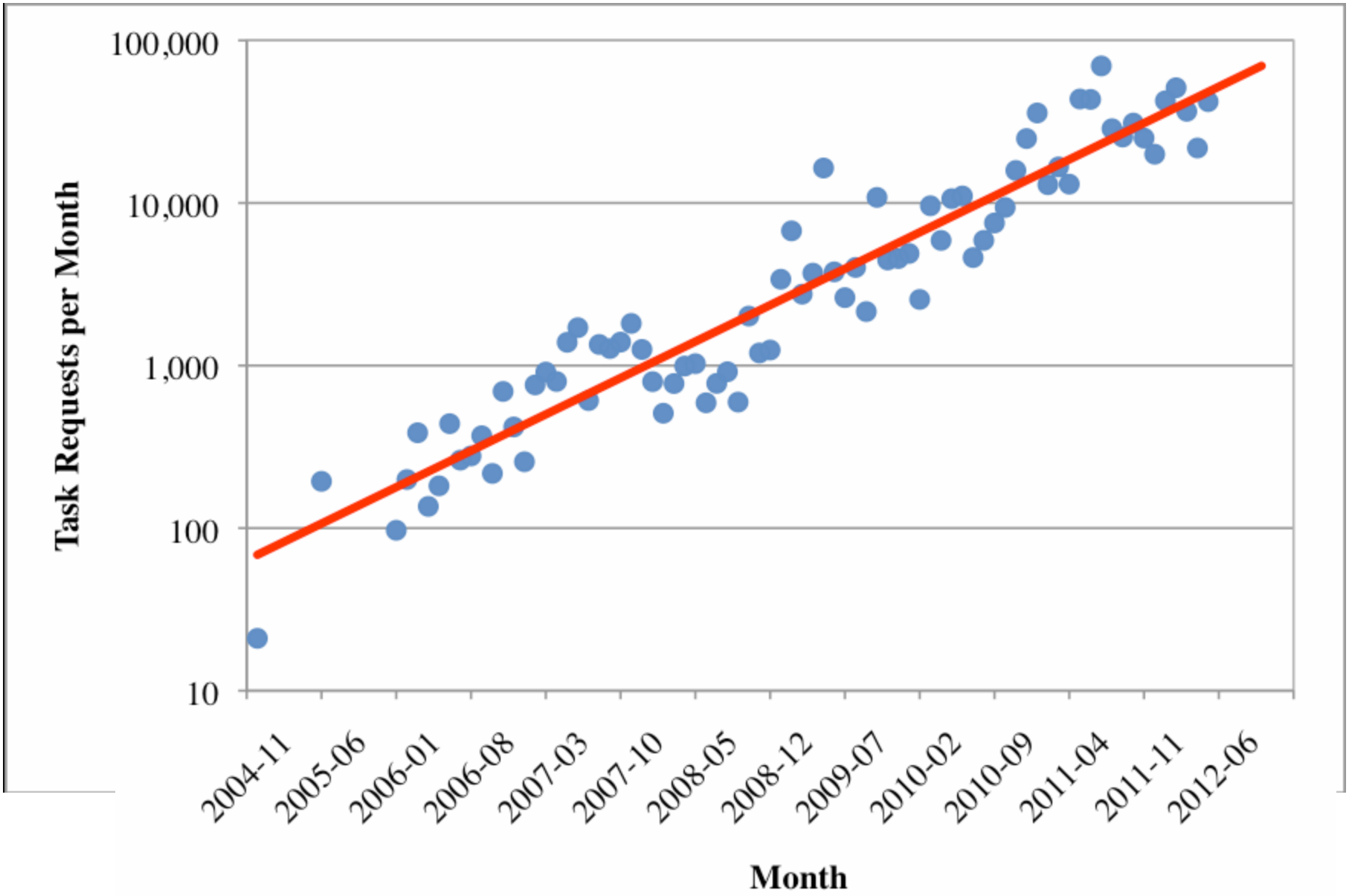
What is the scale of the workload?

- ATLAS generates and processes ~10^6 jobs per day
- ~10^5 jobs running at any given time, submitted by ~10^3 users
- Monte Carlo: 5×10^9 events in 2011 alone

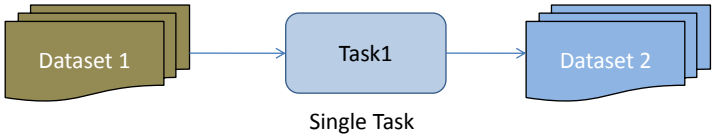
How to manage complexity?

- Given the sheer number of jobs processed in ATLAS , individual job definition is out of question, in particular due to the fact that the large number of job configuration parameters makes it error-prone
- Instead, many jobs are grouped into a *TASK*, i.e. a set of jobs with similar parameters
- Use patterns emerged in the past few years whereby we need to support the “Meta-Task” concept, which is a group of interrelated tasks forming a *workflow*
- Work is under way to upgrade the Workflow Management capabilities of PanDA

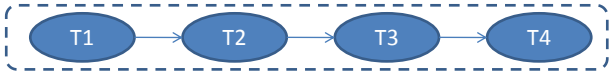
ATLAS: exponential growth of task request rate



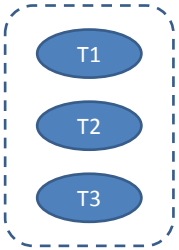
ATLAS: managing workflows (evolution of the Production System)



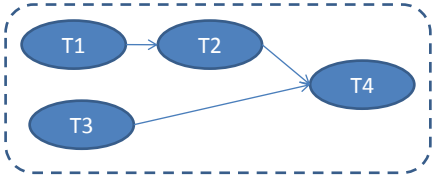
Single Task



Chain Meta-Task



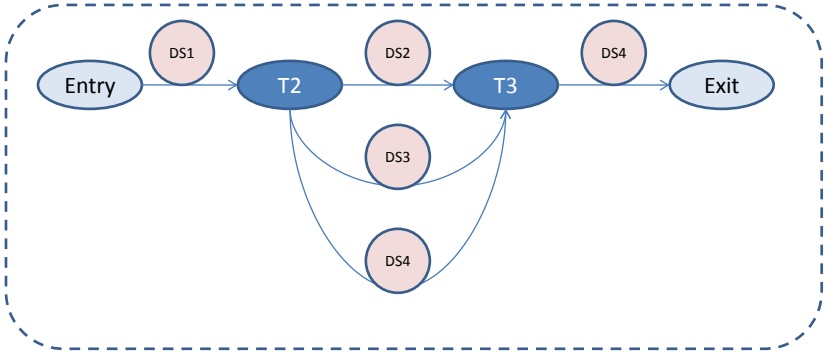
Bag Meta-Task



An example of a more generic Meta-Task

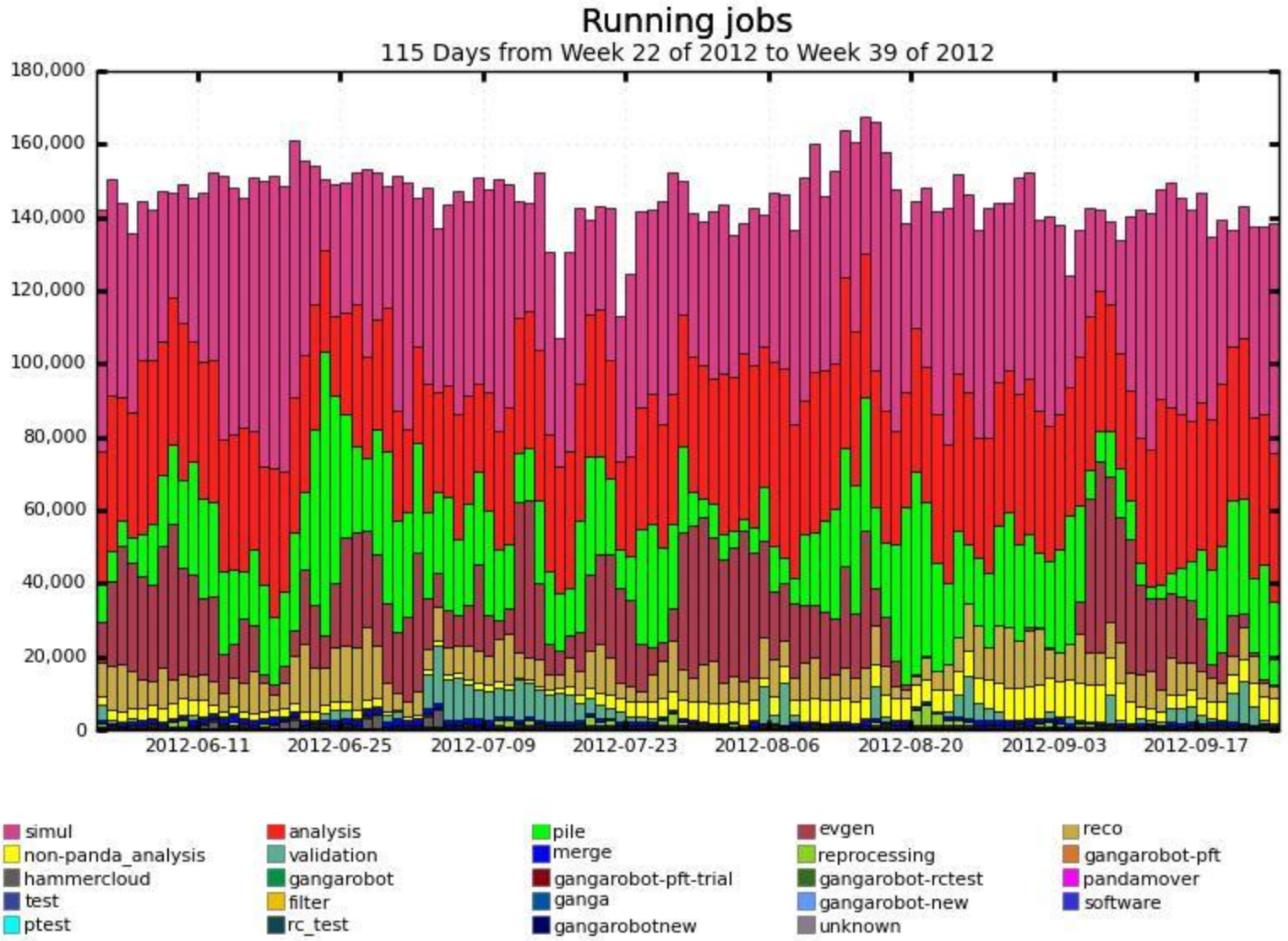
Meta-Task: a collection of tasks connected by data dependencies.

Can be modeled as DAG



Revised Chain Meta-Task Topology Example With Multiple Data Edges

An example of ATLAS job execution history in 2012, broken down by job type



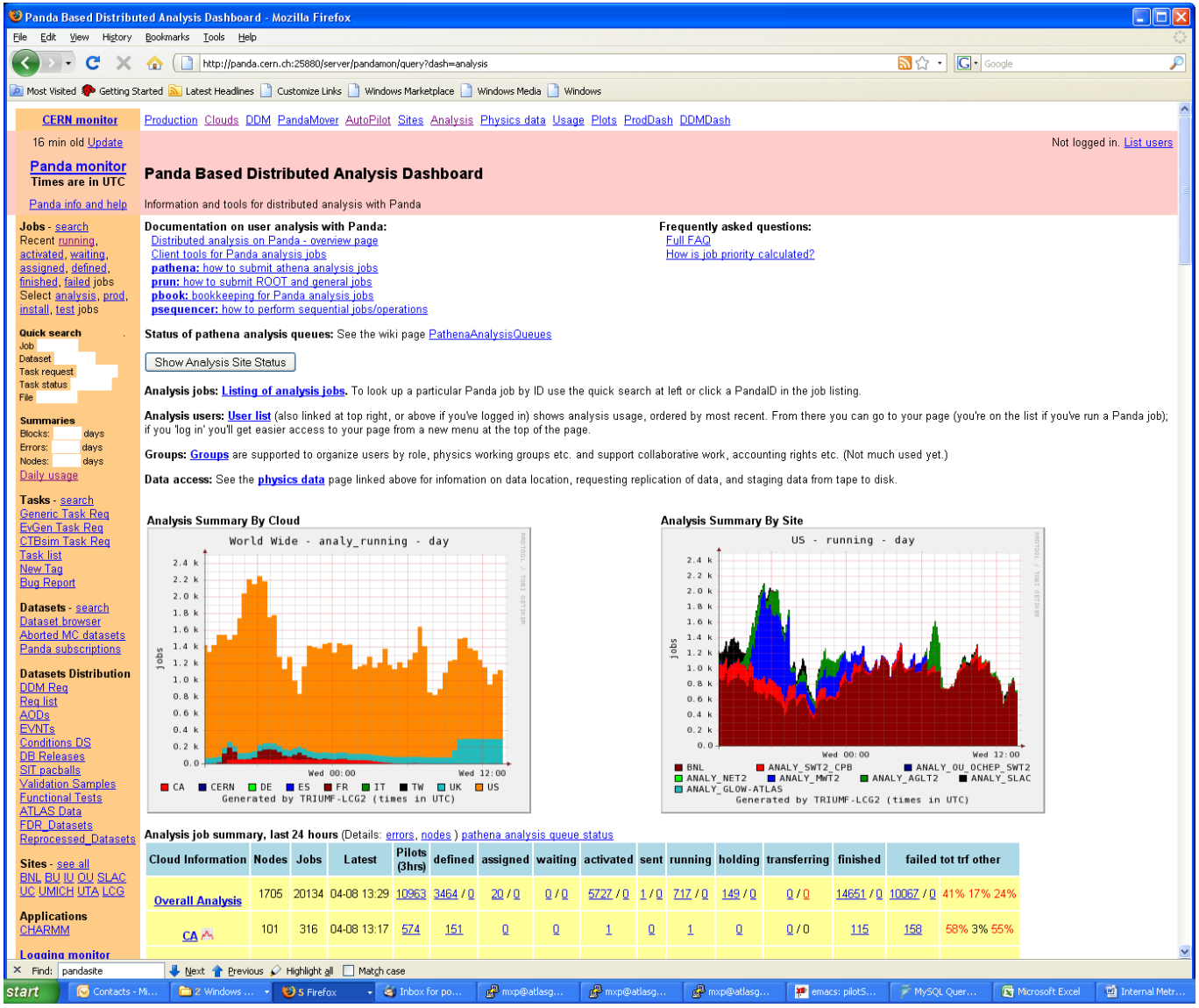
Highlights of PanDA principal design features

- Support for both managed production and individual users (analysis) so as to benefit from a common WMS infrastructure and to allow analysis to leverage production operations support, thereby minimizing overall operations workload.
- A coherent, homogeneous processing system layered over diverse and heterogeneous processing resources, which may simultaneously include local resources such as a farm located at the site, and any number of Grid Sites distributed worldwide. This helps insulate production operators and analysis users from the complexity of the underlying processing infrastructure. It also maximizes the amount of PanDA systems code that is independent of the underlying middleware and facilities actually used for processing in any given environment.
- Extensive direct use of Condor (particularly Condor-G), as a pilot job submission infrastructure of proven capability and reliability. This functionality is currently contained in the recently developed "AutoPyFactory".
- Coherent and comprehensible system view afforded to users, and to PanDA's own job brokerage system, through a system-wide job database that records comprehensive static and dynamic information on all jobs in the system. To users and to PanDA itself, the job database appears essentially as a single attribute-rich queue feeding a worldwide processing resource.

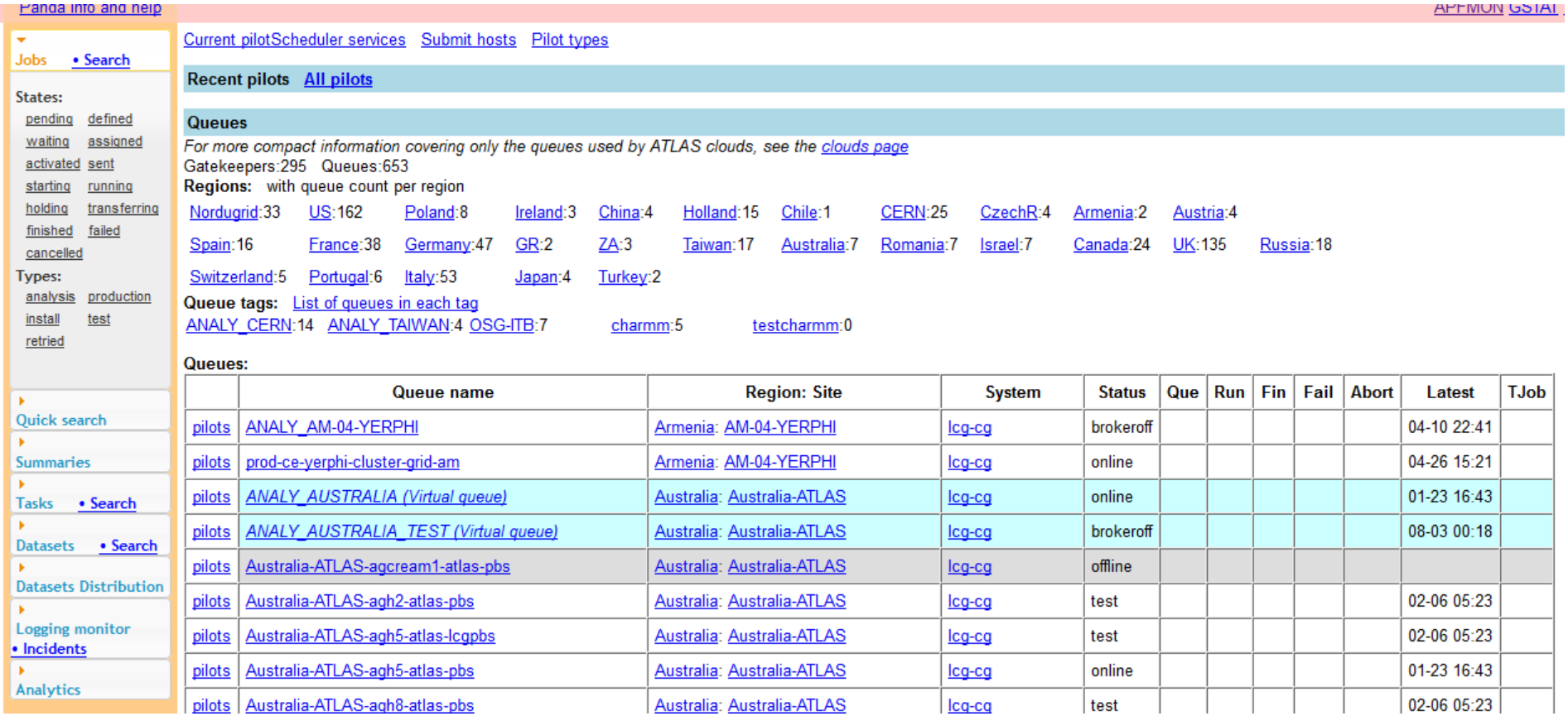
PanDA in a nutshell

- The users never have to deal with specific details of highly heterogeneous computing sites, their batch systems and storage facilities. They don't have to know anything about the sites' gatekeepers or other information that would normally be necessary to submit a job in the Grid environment
- Despite the description of the pilot job framework used in PanDA, as presented in these slides, its users don't have to know much or anything about pilots
- Grid sites become an abstraction know by their mnemonic names, such UTA_SWT2, and are often thought of as queues for job execution
- Stage-in and stage-out of ATLAS data is fully automatic and handled by the Dynamic Data Management System, description of which is outside the scope of this presentation
- Due to intelligence built into the brokerage mechanism in the PanDA server, jobs utilizing a particular dataset are guided to sites that already have these data
- Jobs are fed into PanDA either by individual users, via a suite of command line utilities, or by an automated process (a robot) which processes tasks created by the Task Request system and stored in the database.
- At any given point in time, the users have access to comprehensive information regarding their task, job and data status, aggregated data characterizing operation of sites and clouds, and individual details of the pilot and payload job execution. All of this is delivered to the user via the PanDA Monitor – see the following next slides.
- One way to look at PanDA is this: it effectively creates a virtual supercomputer with ~1.5×10^5 cores

PanDA Monitoring System: a few screenshots



PanDA Monitoring System: a few screenshots



Panda Info and Help

APFMON GSTAT

Jobs • Search

States:
pending defined
waiting assigned
activated sent
starting running
holding transferring
finished failed
cancelled

Types:
analysis production
install test
retried

Quick search

Summaries

Tasks • Search

Datasets • Search

Datasets Distribution

Logging monitor
• Incidents

Analytics

Current pilotScheduler services Submit hosts Pilot types

Recent pilots All pilots

Queues

For more compact information covering only the queues used by ATLAS clouds, see the clouds page

Gatekeepers:295 Queues:653

Regions: with queue count per region

Nordugrid:33 US:162 Poland:8 Ireland:3 China:4 Holland:15 Chile:1 CERN:25 CzechR:4 Armenia:2 Austria:4
Spain:16 France:38 Germany:47 GR:2 ZA:3 Taiwan:17 Australia:7 Romania:7 Israel:7 Canada:24 UK:135 Russia:18
Switzerland:5 Portugal:6 Italy:53 Japan:4 Turkey:2

Queue tags: List of queues in each tag

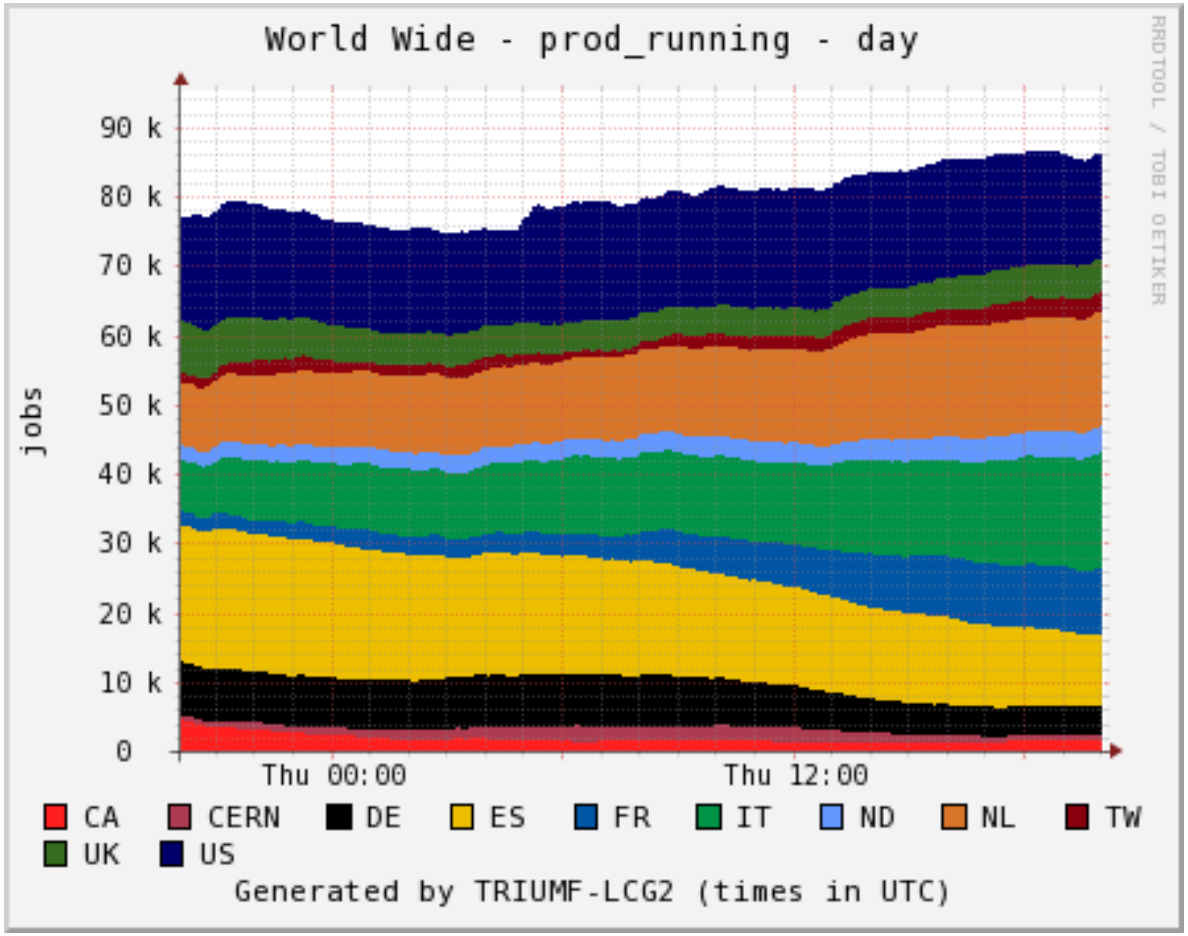
ANALY_CERN:14 ANALY_TAIWAN:4 OSG-ITB:7 charmm:5 testcharmm:0

Queues:

	Queue name	Region: Site	System	Status	Que	Run	Fin	Fail	Abort	Latest	TJob
pilots	ANALY_AM-04-YERPHI	Armenia: AM-04-YERPHI	lcg-cg	brokeroff						04-10 22:41	
pilots	prod-ce-yerphi-cluster-grid-am	Armenia: AM-04-YERPHI	lcg-cg	online						04-26 15:21	
pilots	*ANALY_AUSTRALIA (Virtual queue)*	Australia: Australia-ATLAS	lcg-cg	online						01-23 16:43	
pilots	*ANALY_AUSTRALIA_TEST (Virtual queue)*	Australia: Australia-ATLAS	lcg-cg	brokeroff						08-03 00:18	
pilots	Australia-ATLAS-agcream1-atlas-pbs	Australia: Australia-ATLAS	lcg-cg	offline							
pilots	Australia-ATLAS-agh2-atlas-pbs	Australia: Australia-ATLAS	lcg-cg	test						02-06 05:23	
pilots	Australia-ATLAS-agh5-atlas-lcgpbs	Australia: Australia-ATLAS	lcg-cg	test						02-06 05:23	
pilots	Australia-ATLAS-agh5-atlas-pbs	Australia: Australia-ATLAS	lcg-cg	online						01-23 16:43	
pilots	Australia-ATLAS-agh8-atlas-pbs	Australia: Australia-ATLAS	lcg-cg	test						02-06 05:23	

A view of clouds and queues

PanDA Monitoring System: a few screenshots



* Numbers above represent MC production only

PanDA Monitoring System: a few screenshots

wolfgang.ehrenfeld@(233849)
Pilot counts are for the last 3 hours. Error rates above 5% are shown in red.

This is the **"region" view**. For the "cloud" view click here. For an explanation of the difference click Help above.

Region	Pilots	Latest	pending	defined	waiting	assigned	activated	sent	starting	running	holding	transferring	finished	failed	cancelled
ALL			0	110	0	15508	80921	4	2119	86598	1671	43248	114165	6434	3045
CA	1356	10-04 19:44	0	3	0	308	2278	1	13	5023	200	4641	12954	113	15
CERN	1688	10-04 19:44	0	0	0	595	4853	0	1	2635	65	550	6783	2226	1149
DE	2471	10-04 19:44	0	102	0	4520	7296	3	26	8619	131	4841	10341	366	339
ES	1016	10-04 19:44	0	1	0	68	1635	0	25	3505	57	1723	4761	163	18
FR	3485	10-04 19:44	0	1	0	3112	11756	0	5	10834	269	6374	11802	430	200
IT	1605	10-04 19:44	0	0	0	725	4652	0	5	5801	77	2887	4345	71	166
ND	266	10-04 19:44	0	0	0	504	4489	0	2026	5823	42	764	3004	313	14
NL	1875	10-04 19:44	0	1	0	603	6516	0	14	5275	310	5105	7872	520	350
TW	478	10-04 19:44	0	0	0	0	1157	0	0	1832	25	4	2648	13	2
UK	2881	10-04 19:44	0	0	0	2030	26544	0	1	17119	107	6100	13591	1094	760
US	10328	10-04 19:44	0	2	0	3043	9745	0	3	20132	388	10259	36064	1125	32
OSG (brokeroff)			0	0	0	0	0	0	0	0	0	0	0	0	0

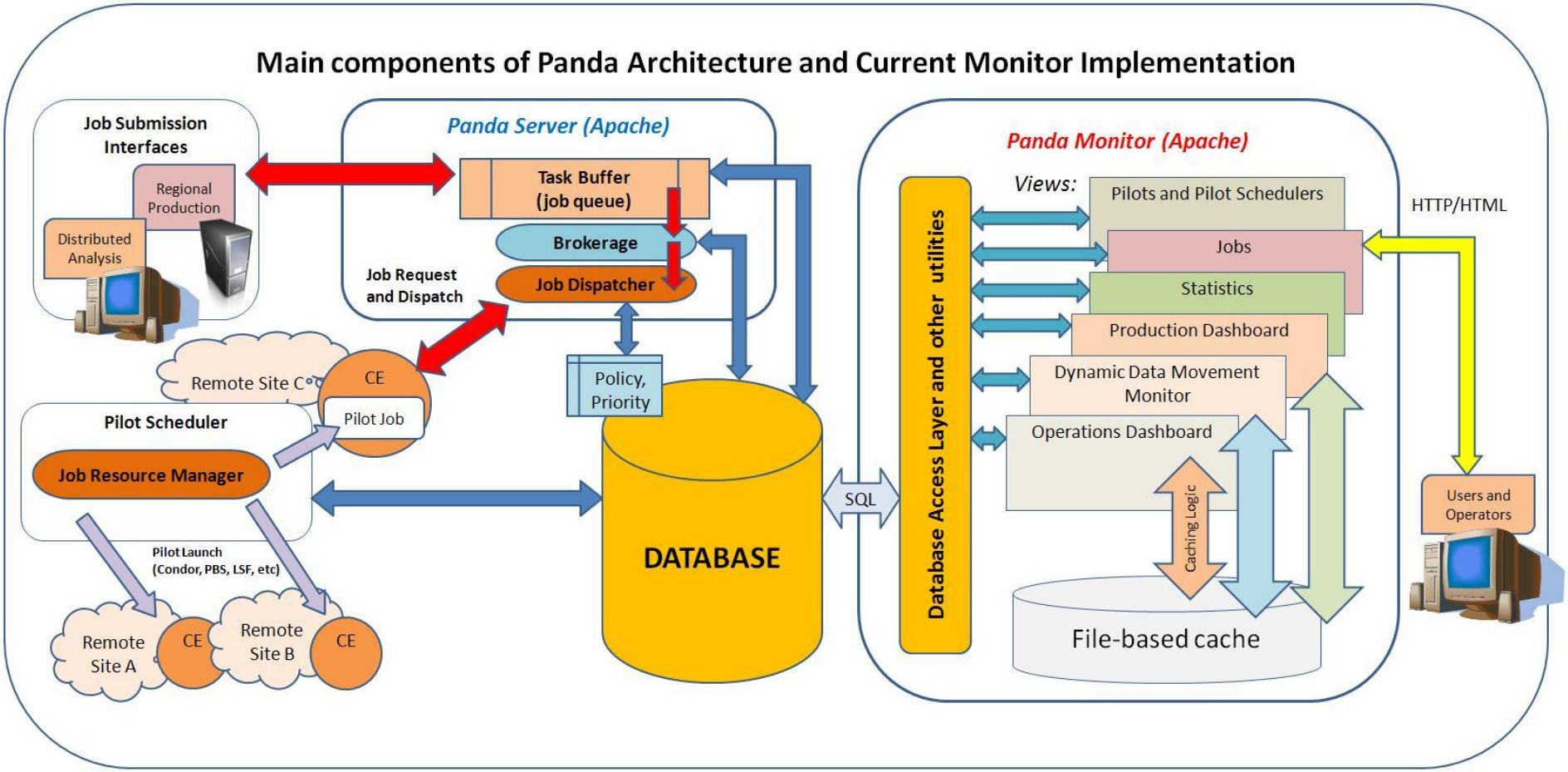
Job Summary from the MultiCore Sites

* PanDA statistics for managed production tasks, excluding chaotic user analysis jobs

PanDA Monitoring System

- **PanDA Monitoring System is its distinguishing and powerful feature**
  - It intuitively models relationships between queues, pilots and jobs, as well as I/O datasets, in the its Web interface
  - Provides central access point to the job's *stdout*, *stderr* and pilot submission log, via the Web interface – extremely useful for troubleshooting
  - Provides plethora of other information:
    - Status of Pilot Factories
    - Statistics on site usage and activity
    - Software releases
    - etc
  - Links to Dynamic Data Management Operations and Atlas Dashboard
  - Provides helpful links to documentation, help and problem reporting
- **Current Monitor technology platform and status:**
  - Written in Python
  - Oracle RDBMS as the data store, *cx_Oracle* is used for database connectivity (MySQL or Postgres capability can be added if needed).
  - Ongoing R&D and development of prototypes to apply noSQL technology
  - Web service based on Apache server instrumented with *mod_python/mod_wsgi*
  - The existing Panda monitoring system represents a significant investment of manpower and expertise. Its evolution was guided by user requirements, and that is reflected in the feature set.
- **Current Development:**
  - The Monitor is being upgraded with better and reorganized code base, which takes advantage of JSON/AJAX technology

An Integrated View of PanDA server and Monitoring System



A few tools and platforms used in ATLAS and PanDA

- **Web services**
  - Web services form the core of PanDA components (server, monitor, etc)
  - based on Apache server instrumented with *mod_python,* and other plug-ins as necessary
  - Increasing use of JSON/AJAX and jQuery to serve and render content
  - Oracle RDBMS is used as backend storage for most components of PanDA
  - Recent migration to noSQL technology for some of the more demanding components: Hadoop, HDFS and HBase. Prior to that a Cassandra-based prototype was built and tested.
- **Language platform for most tools and services**
  - Python is overwhelmingly preferred as the language platform for infrastructure development.
  - Java used where necessary
- **Additional tools and components**
  - The software stack provided by the Open Science Grid
  - HTCondor system is widely used both by itself, and/or as a basis for building software components.
  - Grid job submission (in case of PanDA limited to the Pilot Job submission) is primarily done via Condor-G
  - Message Queue technology is used in a few components in ATLAS
- **Software build, validation and distribution**
  - A well managed process of nightly builds, validation jobs and distribution via CVMFS

PanDA use outside of ATLAS

- **Prior Experience (2007-2010)**
  - Life Sciences: protein folding code (CHARMM) was run opportunistically on OSG sites using PanDA, resulting in publications
  - Daya Bay neutrino experiment group did a few Monte Carlo test runs (production level statistics) on RACF utilizing PanDA, again using some of RHIC owned resources in opportunistic mode
- **Current developments (2011-2013)**
  - The Alpha Magnetic Spectrometer (AMS) collaboration is successfully using PanDA to run Monte Carlo Simulations and is working to create an instance of PanDA server in Taiwan
  - The Large Synoptic Survey Telescope project (LSST): PanDA integration with the LSST workflow management tools is under way, with encouraging first results; appears a good match because of PanDA's ability to manage a large number of concurrent jobs, and effectively federate computational resources of many sites.
  - According to a recent initiative, both CMS and ATLAS experiments at the LHC are exploring the possibility of creating shared components of their computing infrastructure
  - There has been a successful effort aimed at utilization of Cloud-based resources under PanDA management (including Google Compute Engine)

PanDA use outside of ATLAS, cont'd

- **What is needed to use PanDA to its full potential?**
  - All software components need to be Grid-ready, i.e. site-agnostic and easily configurable to adapt to the local file system layout, software location etc
  - Translated into English: no hardcoded paths, or too rigid conventions regarding paths
  - Software build and distribution system must me in place (PanDA can make use of CVMFS for software deployment)
  - PanDA will handle loading the correct environment provided the payload is configurable
  - What's not needed: there is little to no software to be installed by prospective PanDA users on their desktops or interactive nodes, outside of a few job submission scripts.

- **"Threshold of usefulness" is that at which managing/operating the processing takes significant effort in the absence of a tool like PanDA**
  - Several processing sites and/or several individuals/groups doing processing that requires global prioritization/fair share
  - Enough job throughput that automated error handling, retry, bookkeeping tools, diagnostic & historical monitoring are important
  - Sufficient number of participating sites in order to benefit from economies of scale, in terms of operational and direct user support.
  - Value of workflow management

PanDA: a comprehensive upgrade

- **ASCR**
  - ASCR funds have been allocated to the development of the new generation of PanDA, which builds on the strengths of the original design while benefitting from extensive operational experience and new technologies that became available
  - “New PanDA” is focused on providing a great deal of value to a variety of projects in any of the “Frontiers”, which of course includes the Intensity Frontier
- **Perfect Timing**
  - Because of its scope and mandate, the ASCR project is ready to consider requirements (and maybe even contributions) from the Intensity Frontier experiments. It has the potential to bring benefits to the FIFE stakeholders by leveraging resources and expertise in the community
  - Dedicated manpower already assigned and started work
- **The Fabric**
  - The “New PanDA” is a continuation of the approach that enables users and organizations to have transparent access to widely distributed and heterogeneous resources
  - Combined with workflow management and improved monitoring, it will provide a more dynamic and efficient working environment

Summary

- **PanDA is one of a few proven large-scale Workload Management systems:**
  - Handling petabytes of data
  - Processing a million jobs daily on hundreds of sites worldwide
  - Supporting automated production as well as individual user and group tasks
  - Offering extensive monitoring facilities leading to a more transparent and efficient environment
  - Track record in supporting cutting-edge physics research and discovery
  - Upgrade program under way to improve flexibility and scalability of the system, its robustness and ease of use for the researchers