Open Science Grid

PanDA-based
GRID Workload Management

Maxim Potekhin
(presenting for BNL Physics Applications Group)
Brookhaven National Laboratory

North Carolina Grid School
April 22-23, 2009
RENCI

Panda Intro

The Panda (Production ANd Distributed Analysis) system has been developed since summer 2005 to meet challenging requirements of ATLAS Collaboration for a large scale, data-driven workload management system for production and distributed analysis.

Since September 2006 Panda has also been a principal component of the US Open Science Grid (OSG) program in just-in-time (pilot-based) workload management. In October 2007 Panda was adopted by the ATLAS Collaboration as the sole system for distributed processing production across the Collaboration.

Panda Intro

In addition to serving the needs of Atlas community, Panda has also been used by scientists from other disciplines, such as a group of researchers at National Institute of Health. Work is under way to make Panda more accessible to smaller research groups by simplifying API and creating better Web-based interfaces.

Since its commissioning, Panda has processed a large number of jobs (~10^8) on dozens of sites around the world. In addition to the production workflow, there are thousands of analysis jobs run daily.

Current metrics indicate 10^4-10^5 jobs running at any given time, with 10^5-10^6 jobs complete in a 24 hrs period, with the amount of data processed is of the order of 30TB.

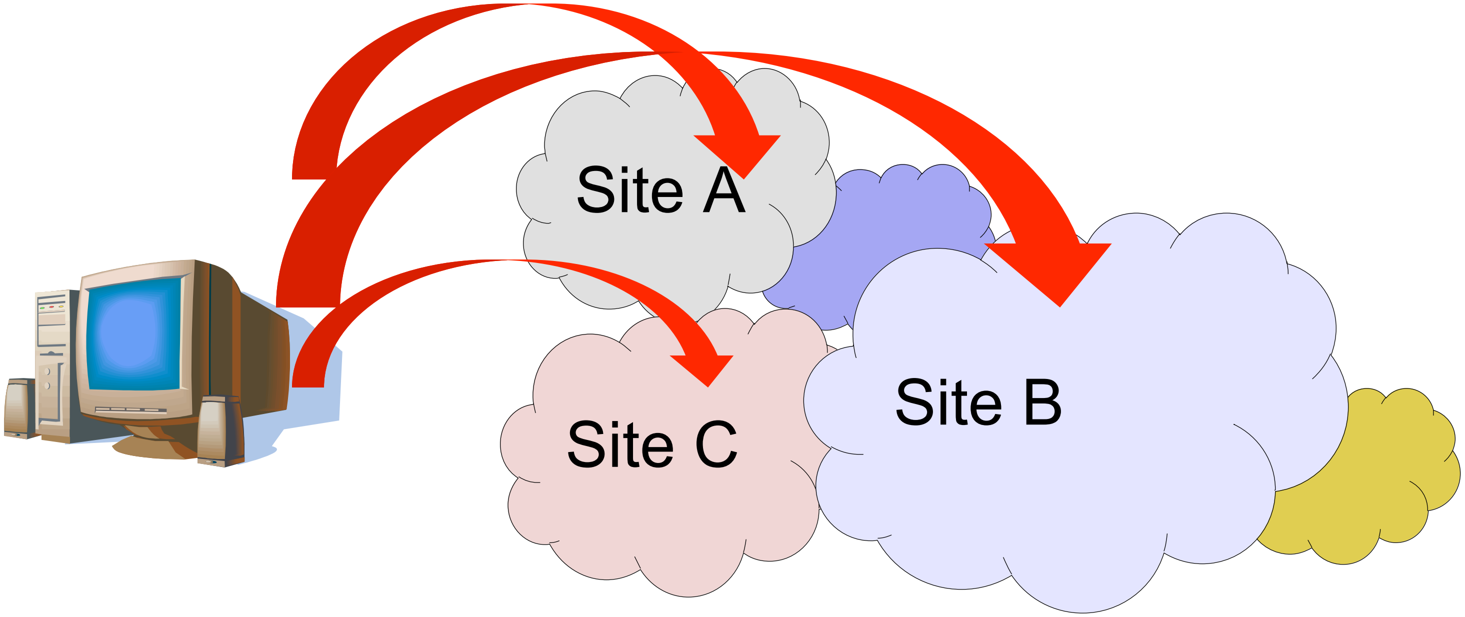
Open Science Grid

Panda Intro

Central for Panda is the concept of Pilot-based workload management. Similar approach has been used in other large-scale projects (CMS, LHCb and others).

In the following slides, we will consider differences between the direct and Pilot-based job management.

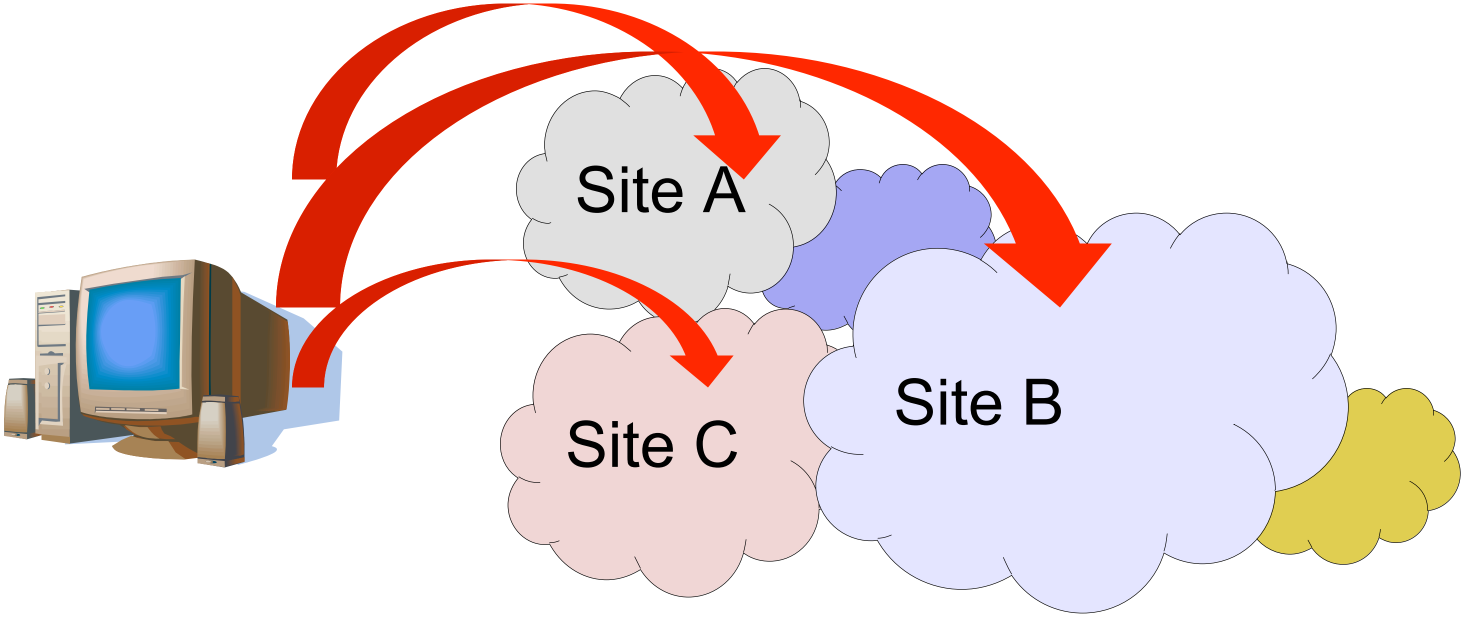
Direct Job Submission (without Panda)



Advantage:

- the user can employ the OSG software stack straight out of the box without having to deploy any extra layers (which, however, may not be easy)

Direct Job Submission (cont'd)



Disadvantages:

• need to interface and manage diverse and heterogeneous processing resources

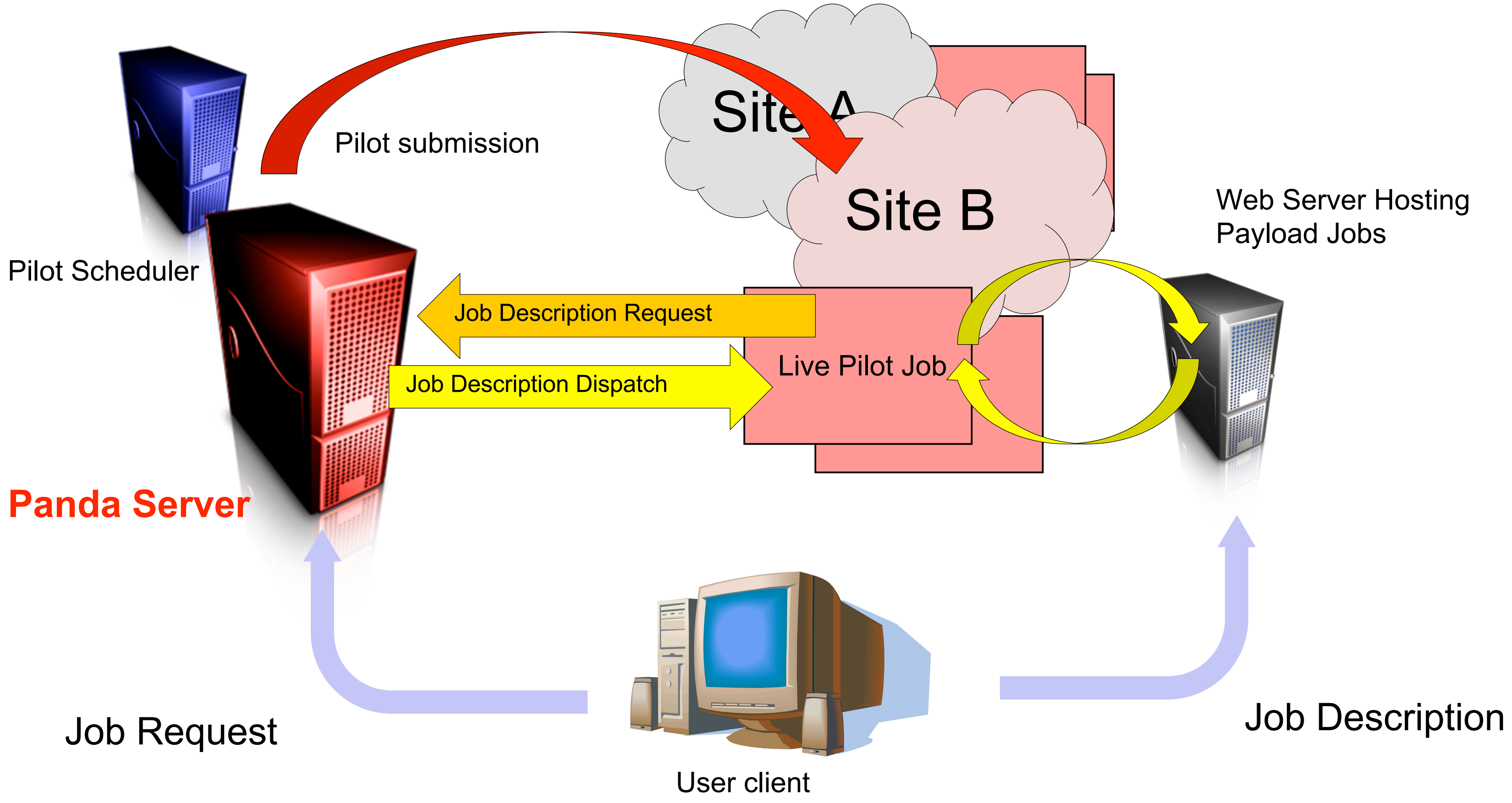
• absence of a system-wide view of job status and progress

• lack of uniform and integrated data management

• hard to control latencies and failure modes inherent in generic in job submission (critical for analysis)

• etc…

Panda Pilot-based job management: the concept



(…next slide)

Panda Pilot-based job management: the concept

Panda's Pilot Framework for Workload Management

• Workload jobs are assigned to successfully activated and validated Pilot Jobs (lightweight processes which probe the environment and act as a 'smart wrapper' for payload jobs), based on Panda-managed brokerage criteria. During this process, the Pilot "dials in" using an outgoing HTTPS connection to the server.

•This 'late binding' of workload 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.

• The pilot also encapsulates the complex heterogeneous environments and interfaces of the grids and facilities on which Panda operates. The users do not need to concern themselves with intricacies of Grid interface – Panda presents them with a unified mode of access to Grid resources.

Panda Pilot-based job management: the concept

Job Submission

- Jobs are submitted via a client interface where the jobs sets, associated datasets, input/output files etc can be defined. Jobs received by the Panda server are placed in the job queue, and the brokerage module to prioritizes and assigns work based on job type, available resources, data location and other criteria.

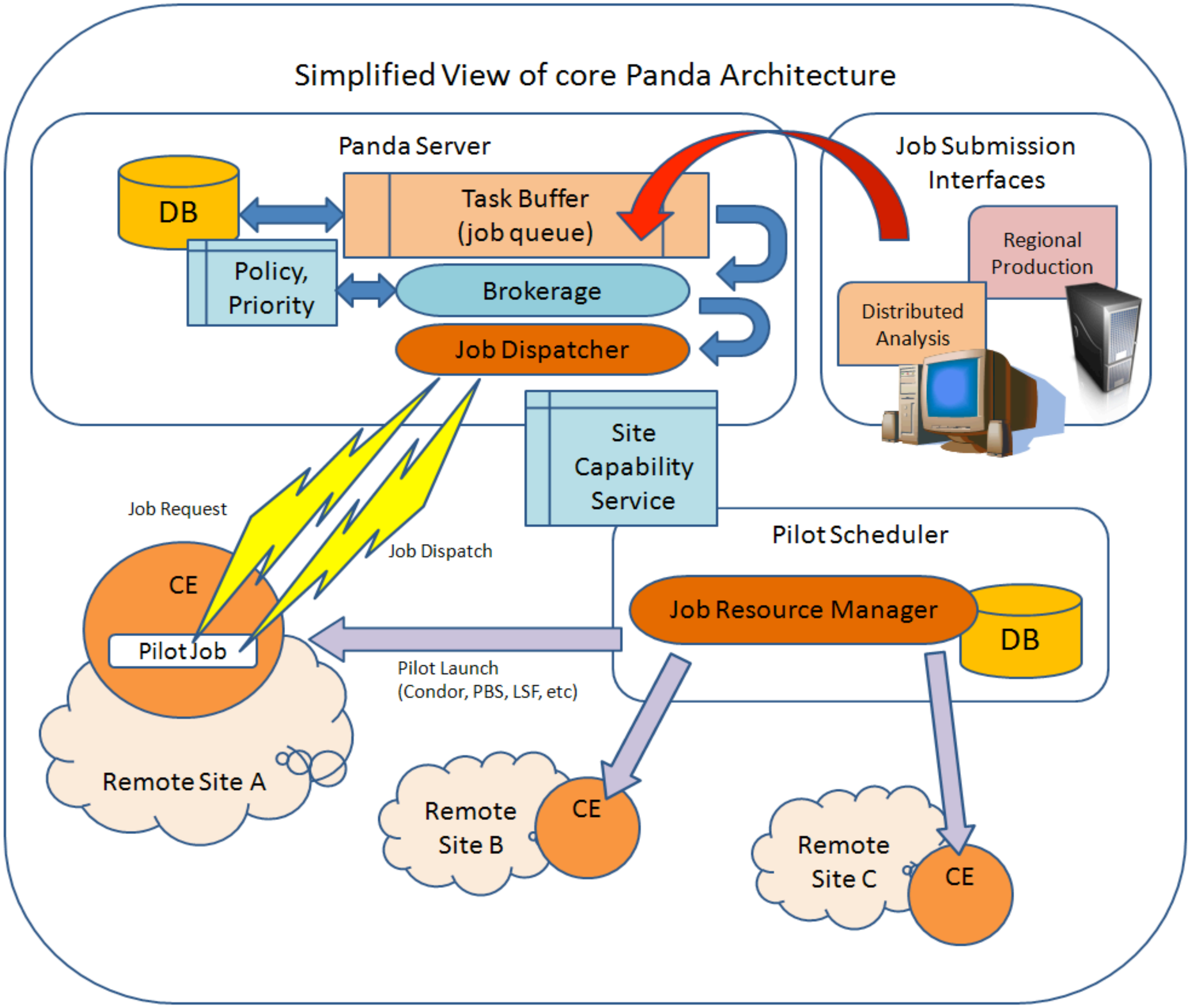
- The payload is not stored on the Panda server - it is defined as a URL from which it can be retrieved, thus improving the scalability and ease of management by the users. To communicate with the Panda and payload servers, the Pilot needs to be capable of outbound HTTP connectivity from the Worker Node on which it is run.

Panda Pilot-based job management: the concept

What Panda doesn’t solve

- It is still necessary to build and make provisions for deployment of the executable, binaries and other resources for the specific target platforms. This can be alleviated by using virtualization, which goes beyond the topic of this presentation.

Panda Architecture



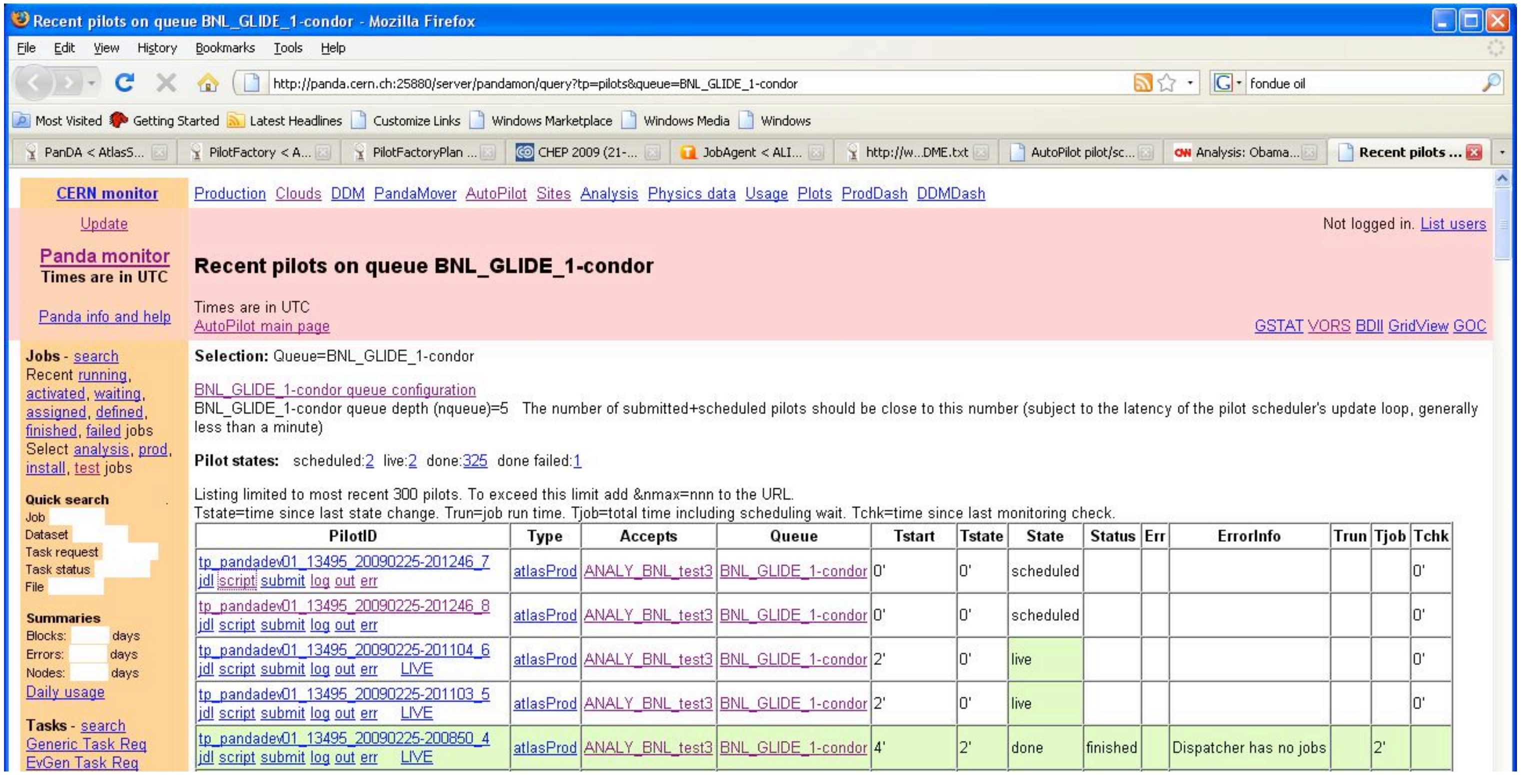
Panda Monitoring

Panda monitoring system is a separate component based on the Apache server, which allows the users and operators to have a comprehensive view of execution flow across multiple sites, and of many aspects of the job progress through the system. One has the capability to “drill down” into job execution detail.

Such details include, for example, log files produced by the pilots, number of running jobs according to various criteria, start and completion time etc.

Panda Monitoring

A sample screenshot of the Panda monitoring system:


Recent pilots on queue BNL_GLIDE_1-condor - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://panda.cern.ch:25880/server/pandamon/query?tp=pilots&queue=BNL_GLIDE_1-condor
fondue oil
Most Visited Getting Started Latest Headlines Customize Links Windows Marketplace Windows Media Windows
PanDA < AtlasS... PilotFactory < A... PilotFactoryPlan ... CHEP 2009 (21-... JobAgent < ALI... http://w...DME.txt AutoPilot pilot/sc... Analysis: Obama... Recent pilots ...

CERN monitor

Production Clouds DDM PandaMover AutoPilot Sites Analysis Physics data Usage Plots ProdDash DDMDash

Update

Not logged in. List users

Panda monitor
Times are in UTC

Panda info and help

Recent pilots on queue BNL_GLIDE_1-condor

Times are in UTC
AutoPilot main page

GSTAT VORS BDII GridView GOC

Jobs - search
Recent running, activated, waiting, assigned, defined, finished, failed jobs
Select analysis, prod, install, test jobs

Quick search
Job
Dataset
Task request
Task status
File

Summaries
Blocks: days
Errors: days
Nodes: days
Daily usage

Tasks - search
Generic Task Req
EvGen Task Req

Selection: Queue=BNL_GLIDE_1-condor

BNL_GLIDE_1-condor queue configuration
BNL_GLIDE_1-condor queue depth (nqueue)=5 The number of submitted+scheduled pilots should be close to this number (subject to the latency of the pilot scheduler's update loop, generally less than a minute)

Pilot states: scheduled:2 live:2 done:325 done failed:1

Listing limited to most recent 300 pilots. To exceed this limit add &nmax=nnn to the URL.
Tstate=time since last state change. Trun=job run time. Tjob=total time including scheduling wait. Tchk=time since last monitoring check.

PilotID	Type	Accepts	Queue	Tstart	Tstate	State	Status	Err	ErrorInfo	Trun	Tjob	Tchk
tp_pandadev01_13495_20090225-201246_7 jdl script submit log out err	atlasProd	ANALY_BNL_test3	BNL_GLIDE_1-condor	0'	0'	scheduled						0'
tp_pandadev01_13495_20090225-201246_8 jdl script submit log out err	atlasProd	ANALY_BNL_test3	BNL_GLIDE_1-condor	0'	0'	scheduled						0'
tp_pandadev01_13495_20090225-201104_6 jdl script submit log out err LIVE	atlasProd	ANALY_BNL_test3	BNL_GLIDE_1-condor	2'	0'	live						0'
tp_pandadev01_13495_20090225-201103_5 jdl script submit log out err LIVE	atlasProd	ANALY_BNL_test3	BNL_GLIDE_1-condor	2'	0'	live						0'
tp_pandadev01_13495_20090225-200850_4 jdl script submit log out err LIVE	atlasProd	ANALY_BNL_test3	BNL_GLIDE_1-condor	4'	2'	done	finished		Dispatcher has no jobs		2'	

Open Science Grid

Panda Pilot-based job management

Submission of Pilot Jobs

Panda makes extensive use of Condor (particularly Condor-G) as a Pilot job submission infrastructure of proven capability and reliability. Pilots are submitted via Pilot Schedulers (Generators), which are typically run by administrators of the Virtual Organization wishing to submit jobs to Panda. Submission rate is regulated by the number of job requests queued in the server, thus eliminating creation of unused pilots and waste of resources.

Other principal design features

- Through a system-wide job database, a comprehensive and coherent view of the system and job execution is afforded to the users.
- Integrated data management is based on the concept of 'datasets' (collections of files), and movement of datasets for processing and archiving is built into the Panda workflow. Asynchronous and automated pre-staging of input data minimizes data transport latencies
- Panda is based on the industry-standard Apache server and therefore renders itself to well understood performance tuning and scalability enhancing procedures. Its security is based on standard Grid tools (such as X.509 certificate proxy-based authentication and authorization)

Summary

- Panda presents a coherent, homogeneous interface to distributes Grid resources to the user, in both production and analysis situation. It mitigates effects of job submission latency and isolates the user from many failure modes that may exist in Grid job submission scenario. It interfaces both automated production submission systems and requests from individuals.

- In addition, it provides integrated data movement capabilities and extensive monitoring tools to users and operators. It has proved itself as a stable and scalable system, capable of addressing computing needs of a large and global organization, as well as of smaller research teams.

- We welcome participation of researchers outside of High Energy and Nuclear Physics (which so far have been the main driver for the project).