The Panda System

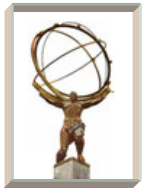
Mark Sosebee (for K. De)

University of Texas at Arlington

dosar workshop

March 30, 2006

Outline

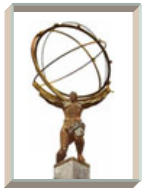


- What is Panda?
- Why Panda?
- Panda Goals and Expectations
- Panda Performance
- Panda Design
- Summary of Panda Features and Components
- Panda Contributors

What is Panda?

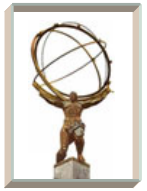
- PanDA – Production and Distributed Analysis system
- Project started Aug 17, 2005
- Baby Panda growing!
- New system developed by U.S. ATLAS team
  - Rapid development from scratch
  - Leverages DC2/Rome experience
  - Inspired by Dirac & other systems
  - Already in use for CSC production in the U.S.
  - Better scalability/usability compared to DC2 system
  - Will be available for distributed analysis users in few months
- "One-stop shopping" for all ATLAS computing users in U.S.

Why Panda?



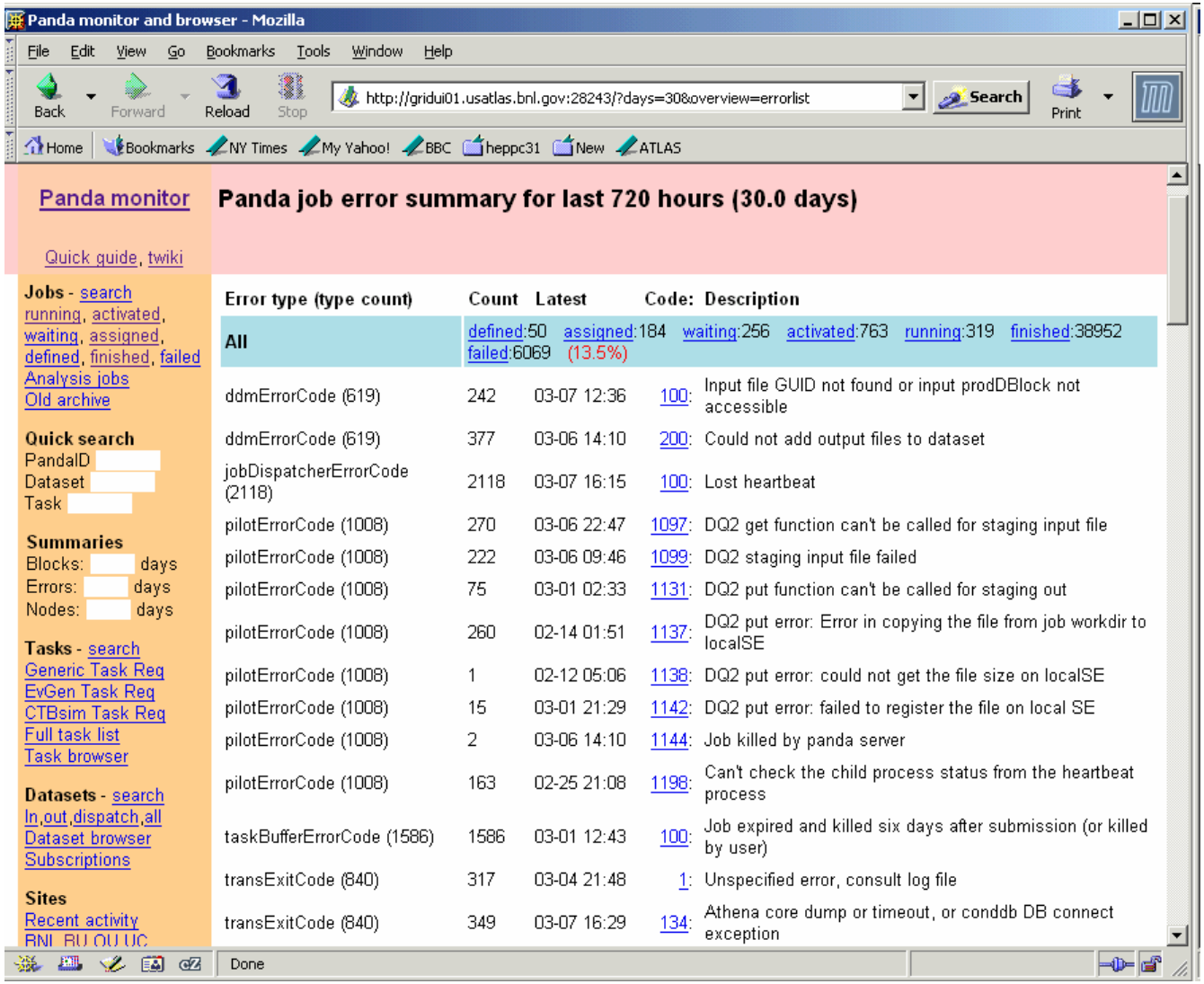
- ATLAS used supervisor/executor system for Data Challenge 2 (DC2) and Rome production in 2004-2005
  - Windmill supervisor common for all grids – developed by KD
  - U.S. executor (Capone) developed by UC/ANL team
  - Four other executors were available ATLAS-wide
- Large scale production was very successful on the grid
  - Dozens of different workflows (evgen, G4, digi, pile-up, reco…)
  - Hundreds of large MC samples produced for physics analysis
- DC2/Rome experience led to development of Panda
  - Operation of DC2/Rome system was too labor-intensive
  - System could not utilize all available hardware resources
  - Scaling problems – hard to scale up by required factor of 10-50
  - No distributed analysis system available, no data management

Original Panda Goals



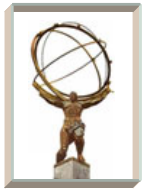
- Minimize dependency on unproven external components
  - Panda relies only on proven and scalable external components – Apache, Python, MySQL
  - Backup/failover systems built into Panda for all other components
    - Local pilot submission backs up CondorG scheduler
    - GridFTP backs up FTS
- Integrated with ATLAS prodsys
- Maximize automation – reduce operations crew
  - Panda operating with smaller (half) shift team and already exceeded DC2/Rome production scale
  - Expect to scale production rate by factor of 10 without increasing operations support significantly
  - Additionally, provide support for distributed analysis users (hundreds of users in the U.S.) – without large increase in personnel

Extensive Error Analysis and Monitoring in Panda Simplifies Operations



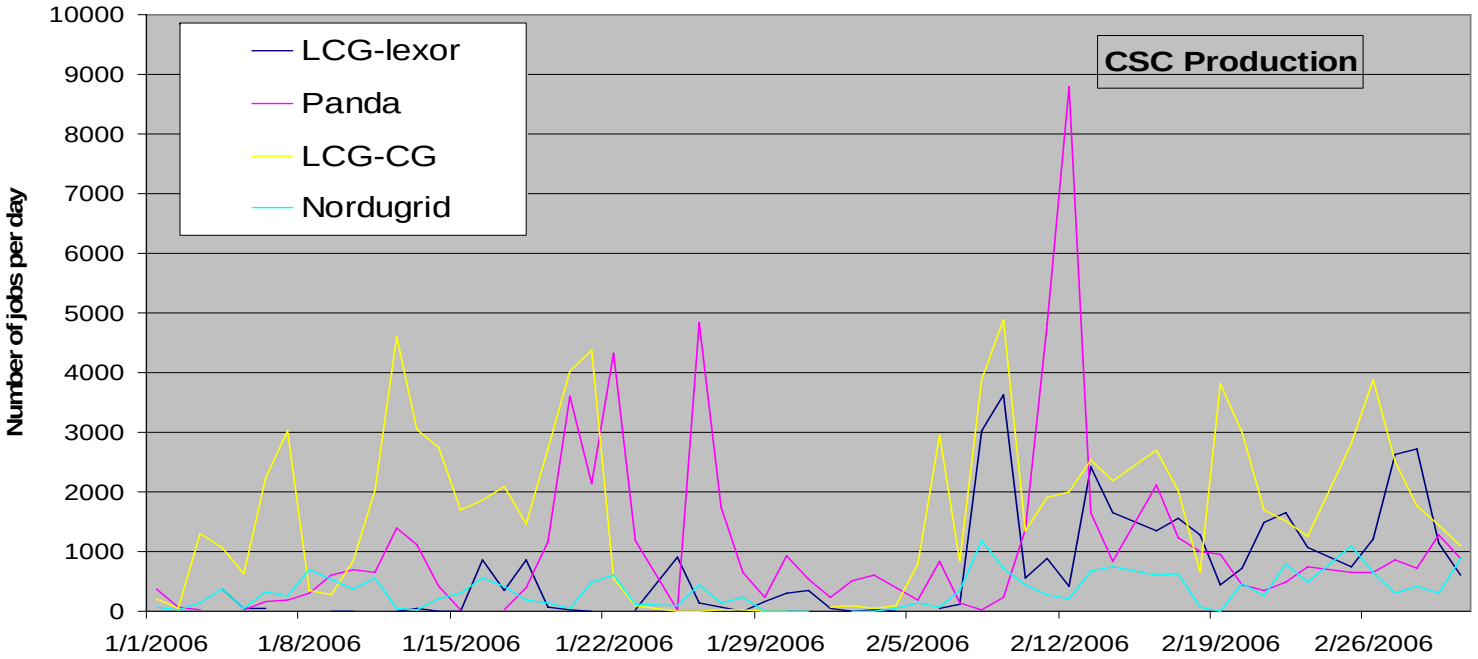
Error type (type count)	Count	Latest	Code:	Description
All	defined:50 assigned:184 waiting:256 activated:763 running:319 finished:38952 failed:6069 (13.5%)			
ddmErrorCode (619)	242	03-07 12:36	100:	Input file GUID not found or input prodDBlock not accessible
ddmErrorCode (619)	377	03-06 14:10	200:	Could not add output files to dataset
jobDispatcherErrorCode (2118)	2118	03-07 16:15	100:	Lost heartbeat
pilotErrorCode (1008)	270	03-06 22:47	1097:	DQ2 get function can't be called for staging input file
pilotErrorCode (1008)	222	03-06 09:46	1099:	DQ2 staging input file failed
pilotErrorCode (1008)	75	03-01 02:33	1131:	DQ2 put function can't be called for staging out
pilotErrorCode (1008)	260	02-14 01:51	1137:	DQ2 put error: Error in copying the file from job workdir to localSE
pilotErrorCode (1008)	1	02-12 05:06	1138:	DQ2 put error: could not get the file size on localSE
pilotErrorCode (1008)	15	03-01 21:29	1142:	DQ2 put error: failed to register the file on local SE
pilotErrorCode (1008)	2	03-06 14:10	1144:	Job killed by panda server
pilotErrorCode (1008)	163	02-25 21:08	1198:	Can't check the child process status from the heartbeat process
taskBufferErrorCode (1586)	1586	03-01 12:43	100:	Job expired and killed six days after submission (or killed by user)
transExitCode (840)	317	03-04 21:48	1:	Unspecified error, consult log file
transExitCode (840)	349	03-07 16:29	134:	Athena core dump or timeout, or conddb DB connect exception

Original Panda Goals (cont.)

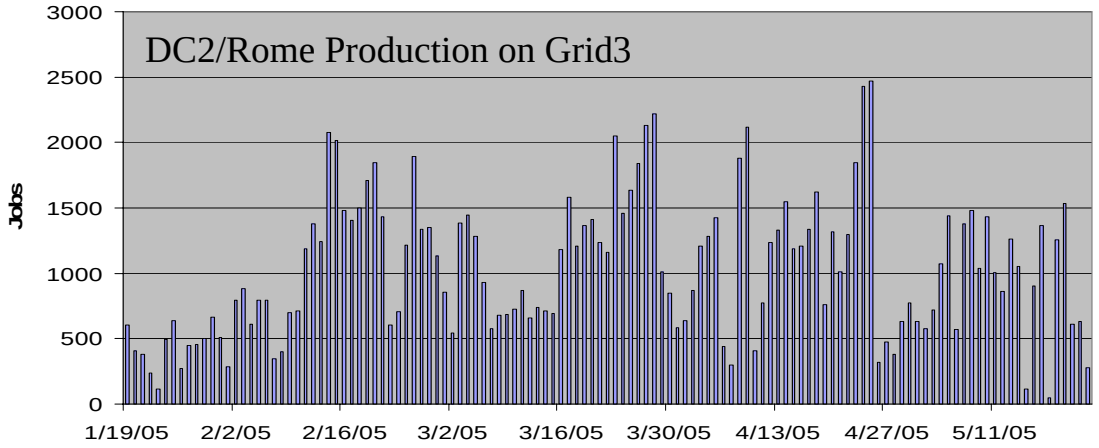


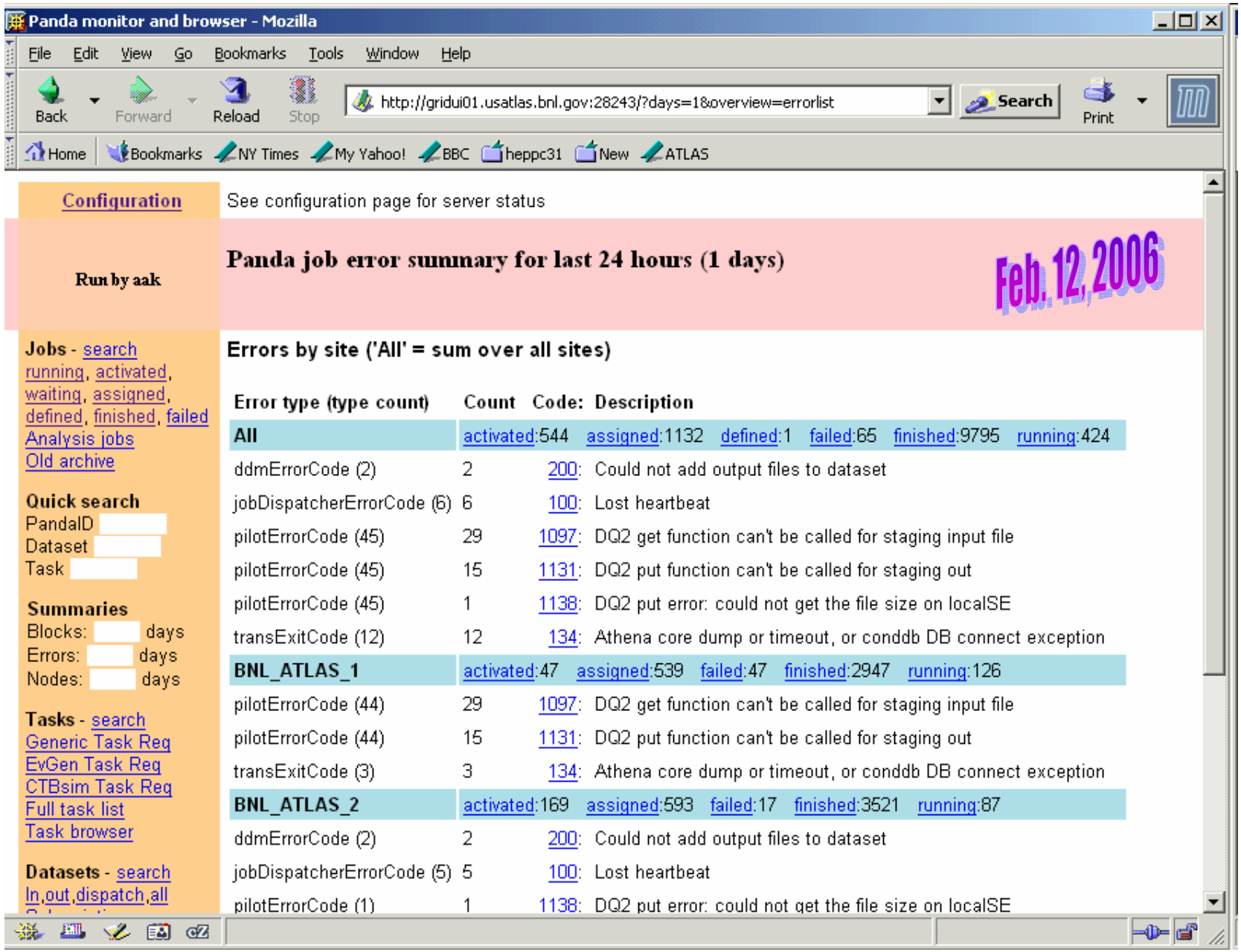
- Efficient utilization of available hardware
  - So far, saturated all available T1 and T2 resources during sustained CSC production over February
  - Available CPU's are expected to increase by factor of 2-4 soon (~one month): Panda expected to continue with full saturation
- Demonstrate scalability required for ATLAS in 2008
  - Already seen factor of 4 higher peak performance compared to DC2/Rome production
  - No scaling limits expected for another factor of 10-20
  - Many techniques available if scaling limits reached – based on proven Apache technology, or deployment of multiple Panda servers, or based on proven MySQL cluster technology

Panda in CSC Production

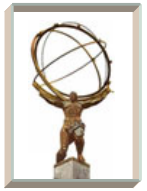


>400 CPU's



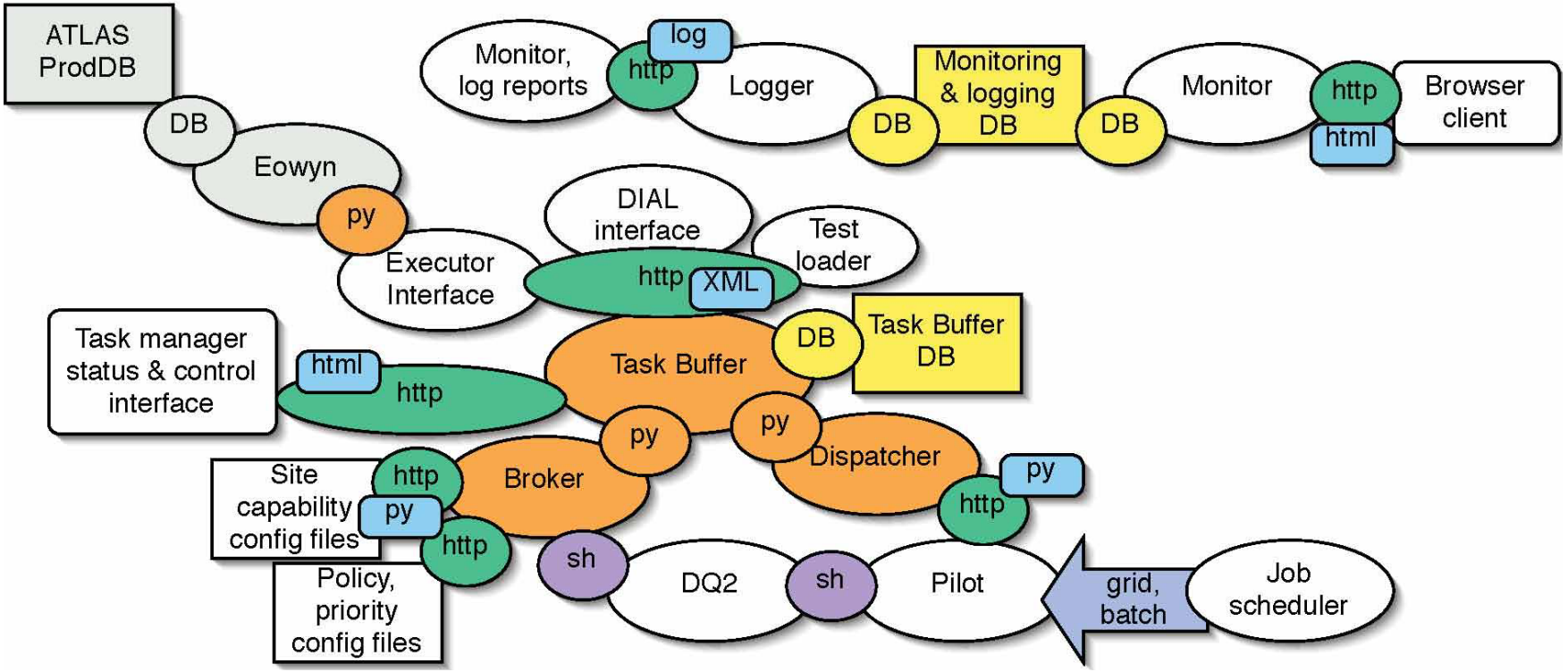

Panda monitor and browser - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://gridui01.usatlas.bnl.gov:28243/?days=1&overview=errorlist
Search Print
Home Bookmarks NY Times My Yahoo! BBC heppc31 New ATLAS
Configuration
See configuration page for server status
Run by aak
Panda job error summary for last 24 hours (1 days)
Feb. 12, 2006
Jobs - search
running, activated,
waiting, assigned,
defined, finished, failed
Analysis jobs
Old archive
Quick search
PandaID
Dataset
Task
Summaries
Blocks: days
Errors: days
Nodes: days
Tasks - search
Generic Task Req
EvGen Task Req
CTBsim Task Req
Full task list
Task browser
Datasets - search
In,out,dispatch,all
Errors by site ('All' = sum over all sites)
Error type (type count) Count Code: Description
All activated:544 assigned:1132 defined:1 failed:65 finished:9795 running:424
ddmErrorCode (2) 2 200: Could not add output files to dataset
jobDispatcherErrorCode (6) 6 100: Lost heartbeat
pilotErrorCode (45) 29 1097: DQ2 get function can't be called for staging input file
pilotErrorCode (45) 15 1131: DQ2 put function can't be called for staging out
pilotErrorCode (45) 1 1138: DQ2 put error: could not get the file size on localSE
transExitCode (12) 12 134: Athena core dump or timeout, or conddb DB connect exception
BNL_ATLAS_1 activated:47 assigned:539 failed:47 finished:2947 running:126
pilotErrorCode (44) 29 1097: DQ2 get function can't be called for staging input file
pilotErrorCode (44) 15 1131: DQ2 put function can't be called for staging out
transExitCode (3) 3 134: Athena core dump or timeout, or conddb DB connect exception
BNL_ATLAS_2 activated:169 assigned:593 failed:17 finished:3521 running:87
ddmErrorCode (2) 2 200: Could not add output files to dataset
jobDispatcherErrorCode (5) 5 100: Lost heartbeat
pilotErrorCode (1) 1 1138: DQ2 put error: could not get the file size on localSE

Original Panda Goals (cont. 2)

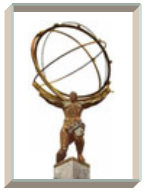


- Tight integration with DQ2 data management system
  - Panda is the only prodsys executor fully integrated with DQ2
  - Panda has played a valuable role in testing and hardening DQ2 in real distributed production environment
  - Excellent synergy between Panda and DQ2 teams
- Provide distributed analysis service for U.S. users
  - Integrated/uniform service for all U.S. grid sites
  - Variety of tools supported
  - Scalable system to support hundreds of users

Panda Design

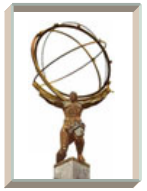


Key Panda Features



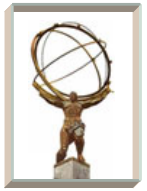
- **Service model** – Panda runs as an integrated service for all ATLAS sites (currently U.S.) handling all grid jobs (production and analysis)
- **Task Queue** – provides batch-like queue for distributed grid resources (unified monitoring interface for production managers and all grid users)
- **Strong data management** (lesson from DC2) – pre-stage, track and manage every file on grid asynchronously, consistent with DQ2 design
- **Block data movement** – pre-staging of output files is done by optimized DQ2 service based on datasets, reducing latency for distributed analysis (jobs follow the data)
- **Pilot jobs** – are prescheduled to batch systems and grid sites; actual ATLAS job (payload) is scheduled when CPU becomes available, leading to low latency for analysis tasks
- **Support all job sources** – managed or regional production (ATLAS ProdSys), user production (tasks, DIAL, Root, pAthena, scripts or transformations, GANGA…)
- **Support any site** – minimal site requirement: pilot jobs (locally or through grid), outbound http, and integration with DQ2 services

Panda Components



- Job Interface – allows injection of jobs into the system
- Executor Interface – translation layer for ATLAS prodsys/prodDB
- Task Buffer – keeps track of all active jobs (job state is kept in MySQL)
- Brokerage – initiates subscriptions for a block of input files required by jobs (preferentially choose sites where data is already available)
- Dispatcher – sends actual job payload to a site, on demand, if all conditions (input data, space and other requirements) are met
- Data Service – interface to DQ2 Data Management system
- Job Scheduler – send pilot jobs to remote sites
- Pilot Jobs – lightweight execution environment to prepare CE, request actual payload, execute payload, and clean up
- Logging and Monitoring systems – http and web-based
- All communications through REST style HTTPS services (via mod_python and Apache servers)

Job States in the System

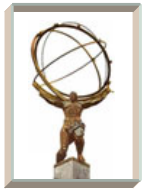


defined → assigned → activated → running → finished/failed

(If input files are not available: defined → waiting. Once files are available, chain picks up again at “assigned” stage)

defined : inserted in panda DB
assigned : dispatchDBlock is subscribed to site
waiting : input files are not ready
activated: waiting pilot requests
running : running on a worker node
finished : finished successfully
failed : failed

Job States in the System (cont.)



What triggers a change in job status?

defined → assigned / waiting: automatic

assigned → activated: received a callback which DQ2 site service sends when dispatchDBlock is verified. If a job doesn't have input files, it is activated without a callback.

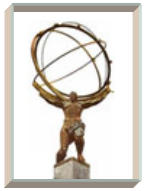
activated → running: picked up by a pilot

waiting → assigned: received a callback which DQ2 site service sends when destionationDBlock is verified

Panda Contributors

- Project Cordinators: Torre Wenaus – BNL, Kaushik De – UTA
- Lead Developer: Tadashi Maeno – BNL
- Panda team
  - Brookhaven National Laboratory (BNL): Wensheng Deng, Alexei Klimentov, Pavel Nevski, Yuri Smirnov, Tomasz Wlodek, Xin Zhao; University of Texas at Arlington (UTA): Nurcan Ozturk, Mark Sosebee; Oklahoma University (OU): Karthik Arunachalam, Horst Severini; University of Chicago (UC): Marco Mambelli; Argonne National Laboratory (ANL): Jerry Gerialtowski; Lawrence Berkeley Lab (LBL): Martin Woudstra
  - Distributed Analysis team (from Dial): David Adams – BNL, Hyunwoo Kim – UTA
  - DQ2 developers (CERN): Miguel Branco, David Cameron

Panda Performance Relative to the Other Grids



- How to compare this system to the resources needed to staff equivalent non-PanDA-based production elsewhere in ATLAS? Here is one possibility:
  - Panda has a single shift crew, 30k CSC jobs completed
  - NG has a single shift crew, 12k CSC jobs completed, approximately same number of available CPU's as Panda
  - LCG (Lexor) has two shift crews now, third one in training, 32k CSC jobs completed
  - LCG-CG has two shift crews now, third one in training, 51k CSC jobs completed
  - LCG + LCG-CG has 4-5 times the number of available CPU's as Panda (shared between two executors)