# The ATLAS **Panda** Production and Distributed Analysis System

Torre Wenaus, BNL

Technology Meeting
BNL
March 13, 2006

BROOKHAVEN
NATIONAL LABORATORY

# Outline

- Panda background
- Design and implementation
- Data management in Panda
- Monitoring & Logging
- Panda for distributed analysis
- Performance and status
- Program of work
- Conclusion
- More information

# What is Panda?

- PanDA - Production and Distributed Analysis system
- Started August 2005 as a full redesign to achieve performance, scalability, ease of operation needed for ATLAS datataking (up to 100-200k jobs/day)
  - Leverages past production experience
  - Inspired by DIRAC (LHCb) and other systems
  - Designed to inherently support analysis
  - 'One stop shopping' for distributed processing
- In production since Dec 2005
  - Ambitious development milestones met
  - Thanks to productive development team
  - Still in rapid development, esp. analysis

# Panda Team

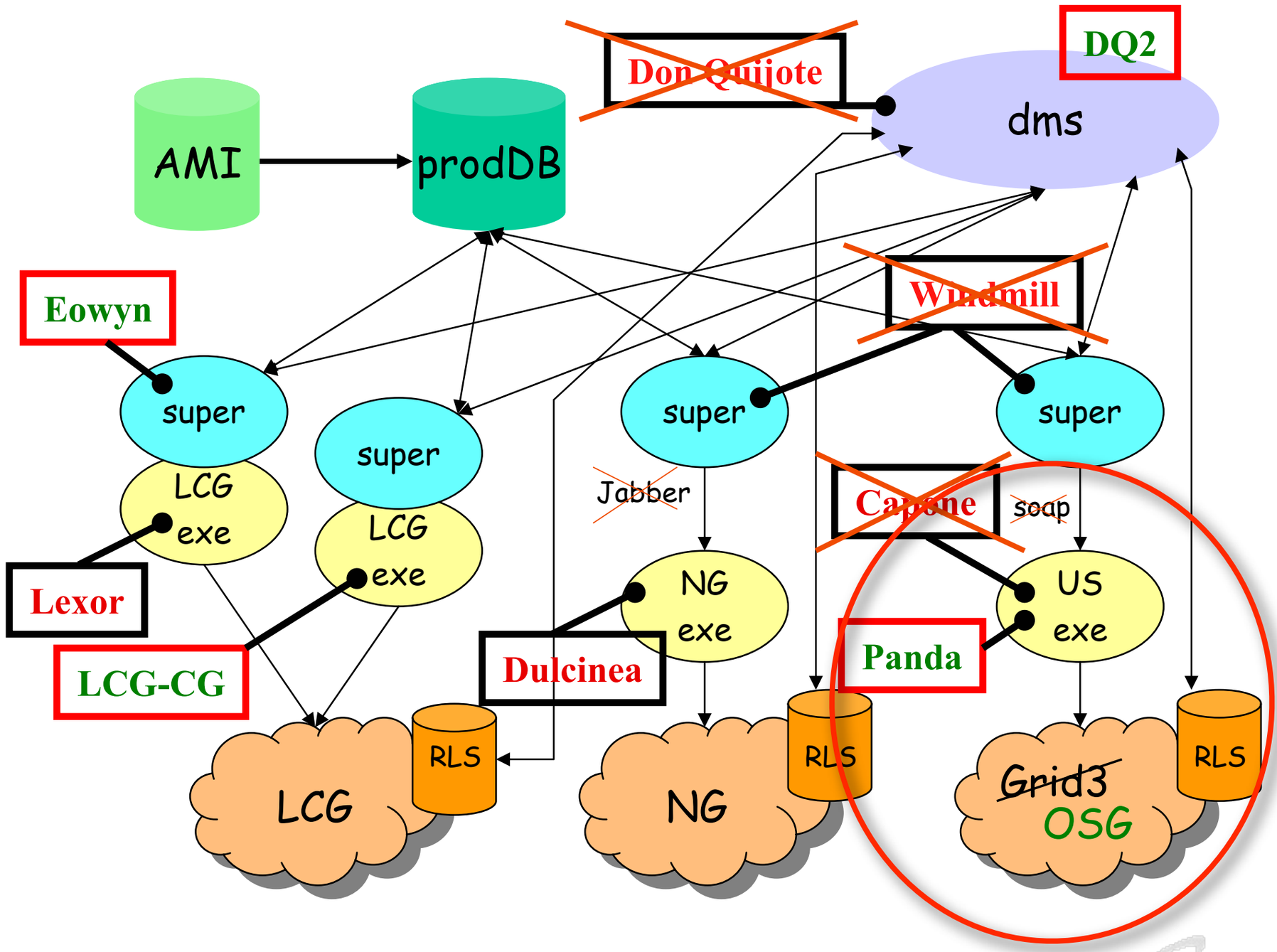
- Project Cordinators: Torre Wenaus – BNL, Kaushik De – UTA
- Lead Developer: Tadashi Maeno – BNL
- Panda team
  - Brookhaven National Laboratory (BNL): Wensheng Deng, Alexei Klimentov, Pavel Nevski, Yuri Smirnov, Tomasz Wlodek, Xin Zhao; University of Texas at Arlington (UTA): Nurcan Ozturk, Mark Sosebee; 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

# Why Panda?

- Previous system used for ATLAS Data Challenge 2 (DC2) and Rome physics workshop production 2004-2005
  - US developed ATLAS-wide 'supervisor', 'Windmill', and US grid specific 'executor', 'Capone' (one of five in ATLAS)
- Large scale grid-based production was very successful
  - Dozens of workflows (evgen, simu, digi, pile-up, reco, ...)
  - Hundreds of large MC samples for physics analysis
- However, the experience exposed problems as well
  - Very labor-intensive; manual fixes for frequent failures in software, middleware, facilities
  - Not scalable: unable to utilize all available resources
  - No real data management
  - No analysis support
- Panda developed to address the shortcomings
  - Within the unchanged overall supervisor/executor structure of ATLAS production

# ATLAS Production System Evolution

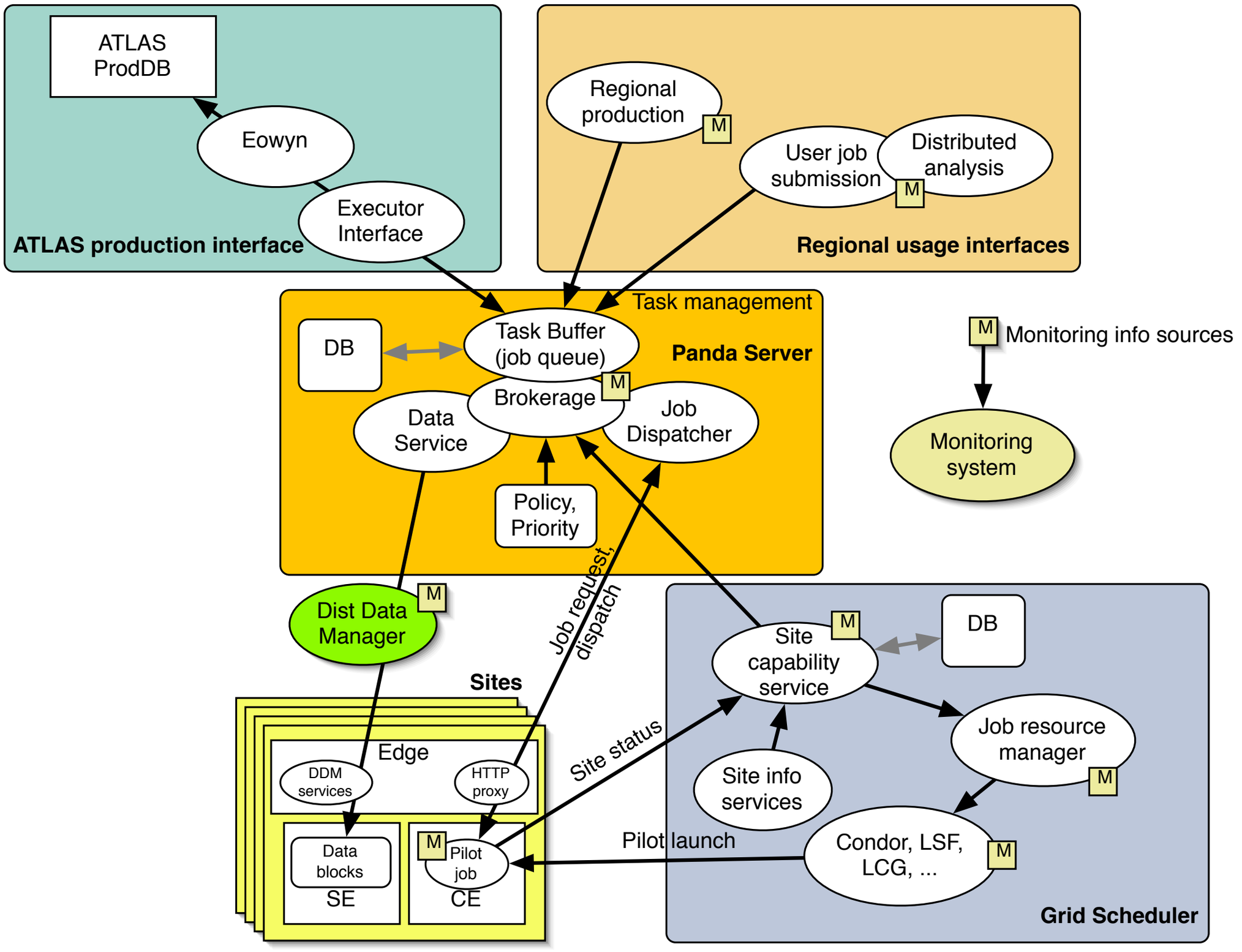
- Overall architecture unchanged
- Major changes:
  - Redesigned data manager Don Quixote 2 (DQ2)
  - Redesigned US executor **Panda**
- Minor changes:
  - Eowyn supervisor similar to previous
  - Other executors similar to previous



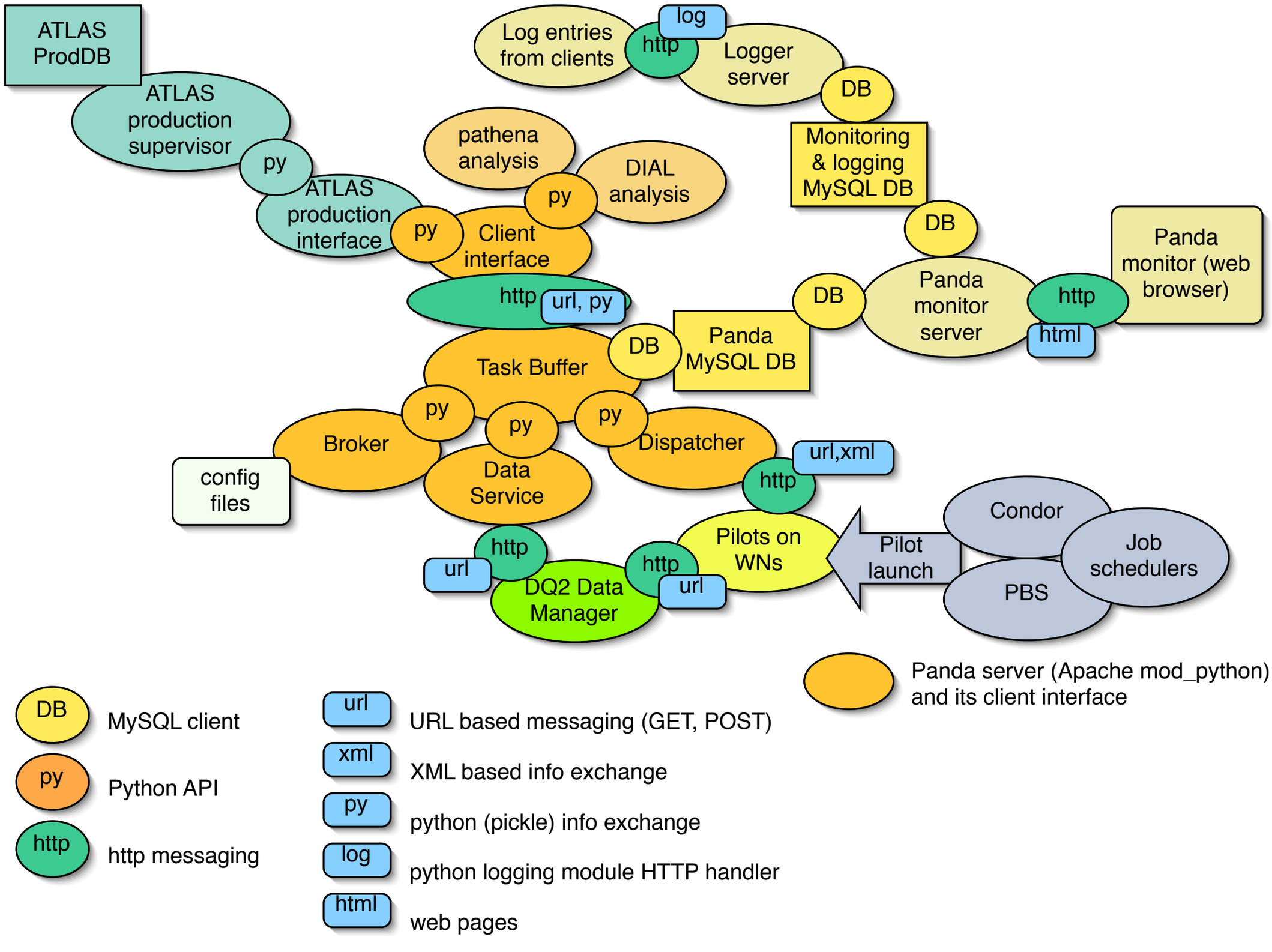
# Key Panda Design Elements

- Support all job sources: ATLAS production, regional / group / user production, interactive analysis
- Tight integration with data management system
  - Data-driven, dataset (file collection) based workflow
  - Data pre-staging as strict prerequisite to job dispatch
- Job scheduling and brokerage done within the system; no reliance on middleware (resource broker)
  - Full, dynamic control of work allocations, priorities, workflow
  - Unified view of the system as effectively one queue
- Job dispatching done within the system rather than via grid submission ('late binding' or 'just-in-time' dispatch)
  - Generic 'pilots' delivered to worker nodes via grid or batch systems; running pilots then request work from Panda
  - Latency for job delivery to a WN can be seconds
- Support any site: minimal requirements
  - Pilot delivery, outbound http, data management

# Panda Architecture

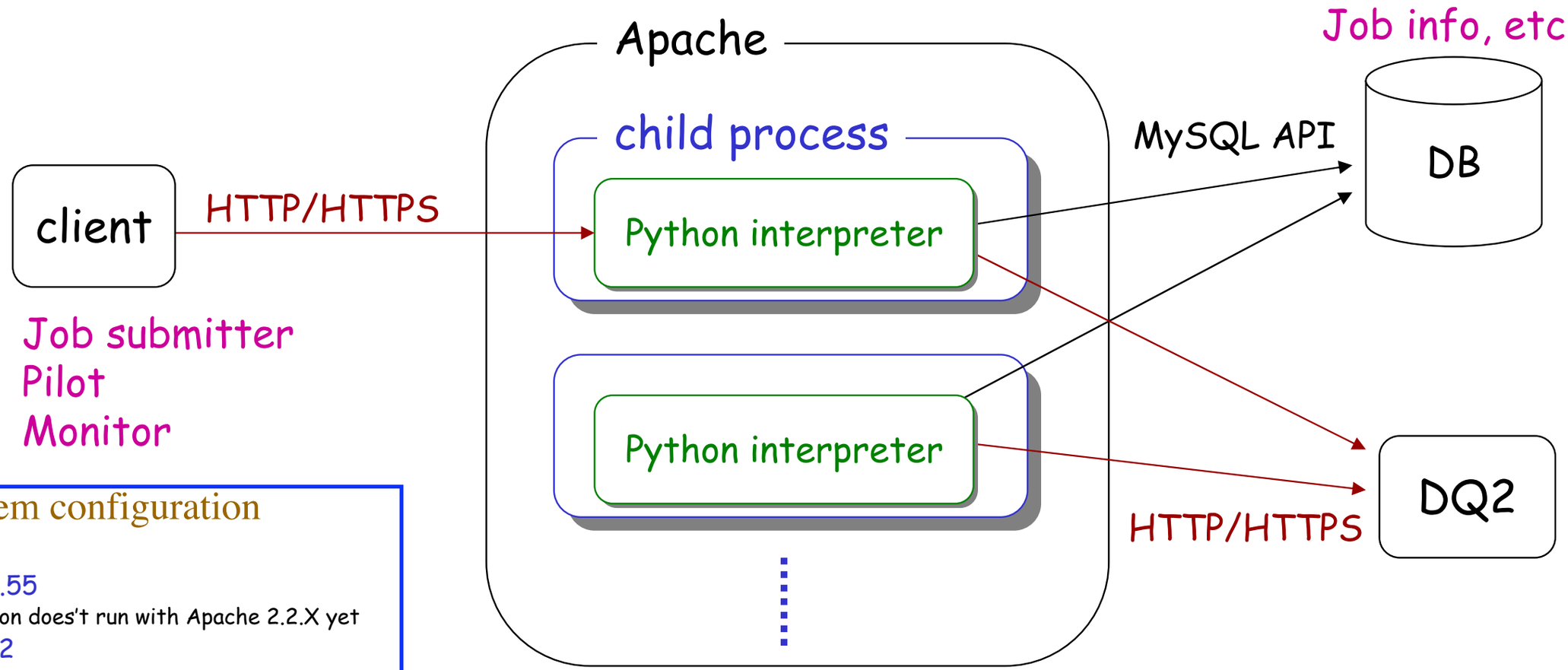


# Panda Implementation



# Panda Server Design

- Apache-based
- Communication via HTTP/HTTPS
- Multi-process
- Global info in the memory resident database

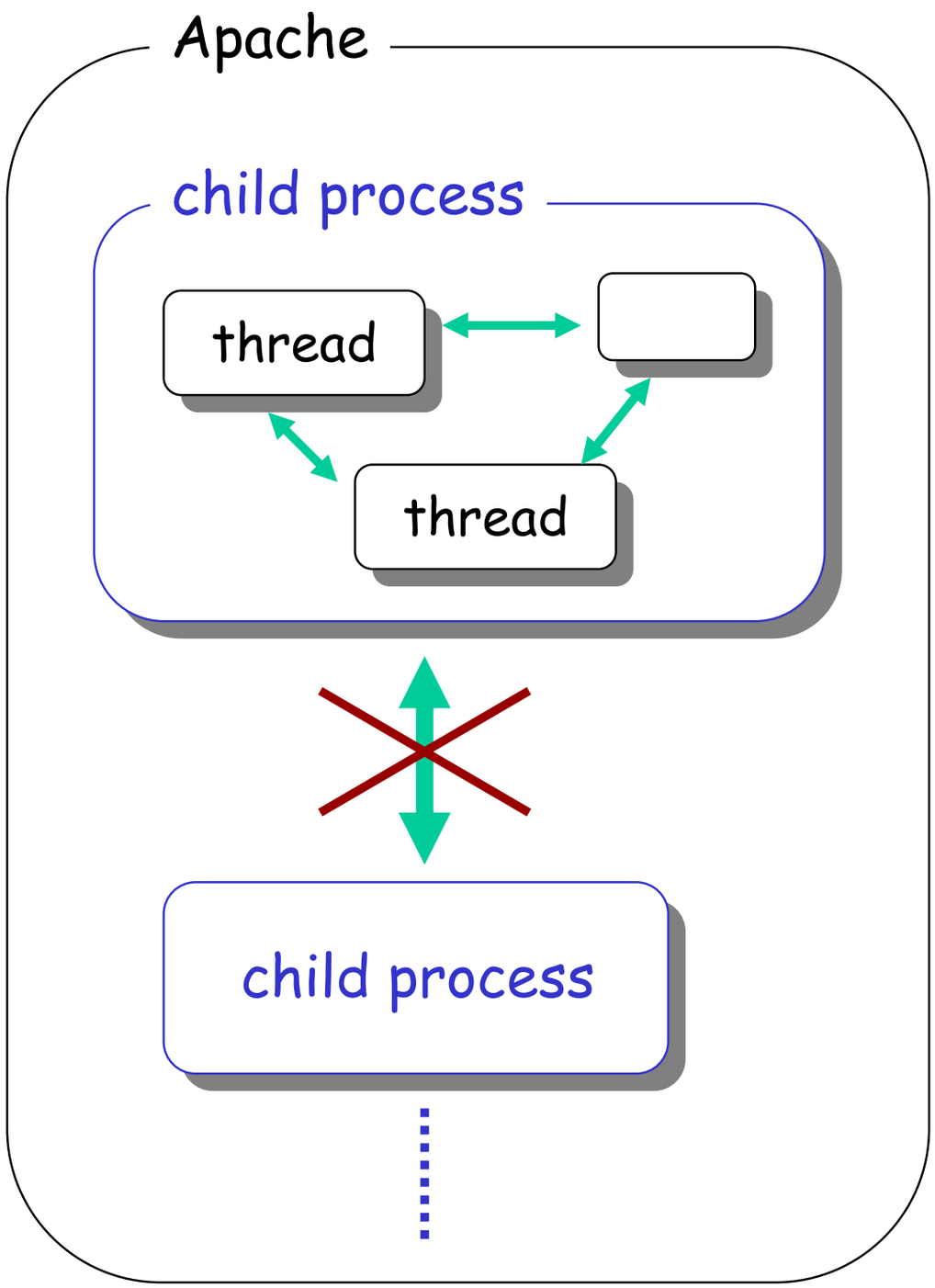



## System configuration

- Components
  - Apache 2.0.55
    - mod_python does't run with Apache 2.2.X yet
  - Python 2.4.2
  - mod_python 3.1.8
  - mod_gridsite 1.1.15
  - MySQL-python 1.1.8
- The panda server doesn't have dependence on special grid-middleware
  - can run even at CERN if needed
  - needs to be deployed near the database

T. Maeno

# Memory-resident MySQL Database



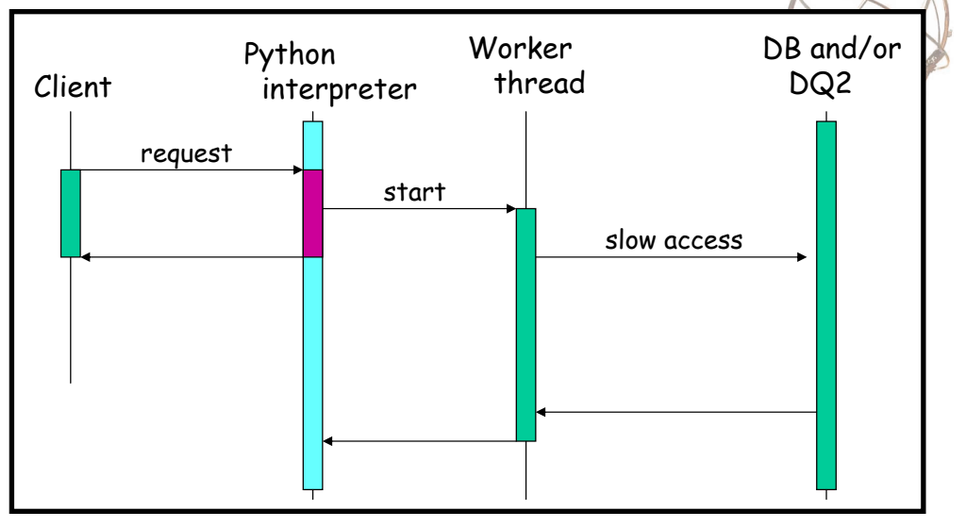

Threads in a child process
share a memory space
→ each child process can
have internal state

Child processes cannot
communicate with each other
without IPCs
→ the Apache system can not have
consistent state

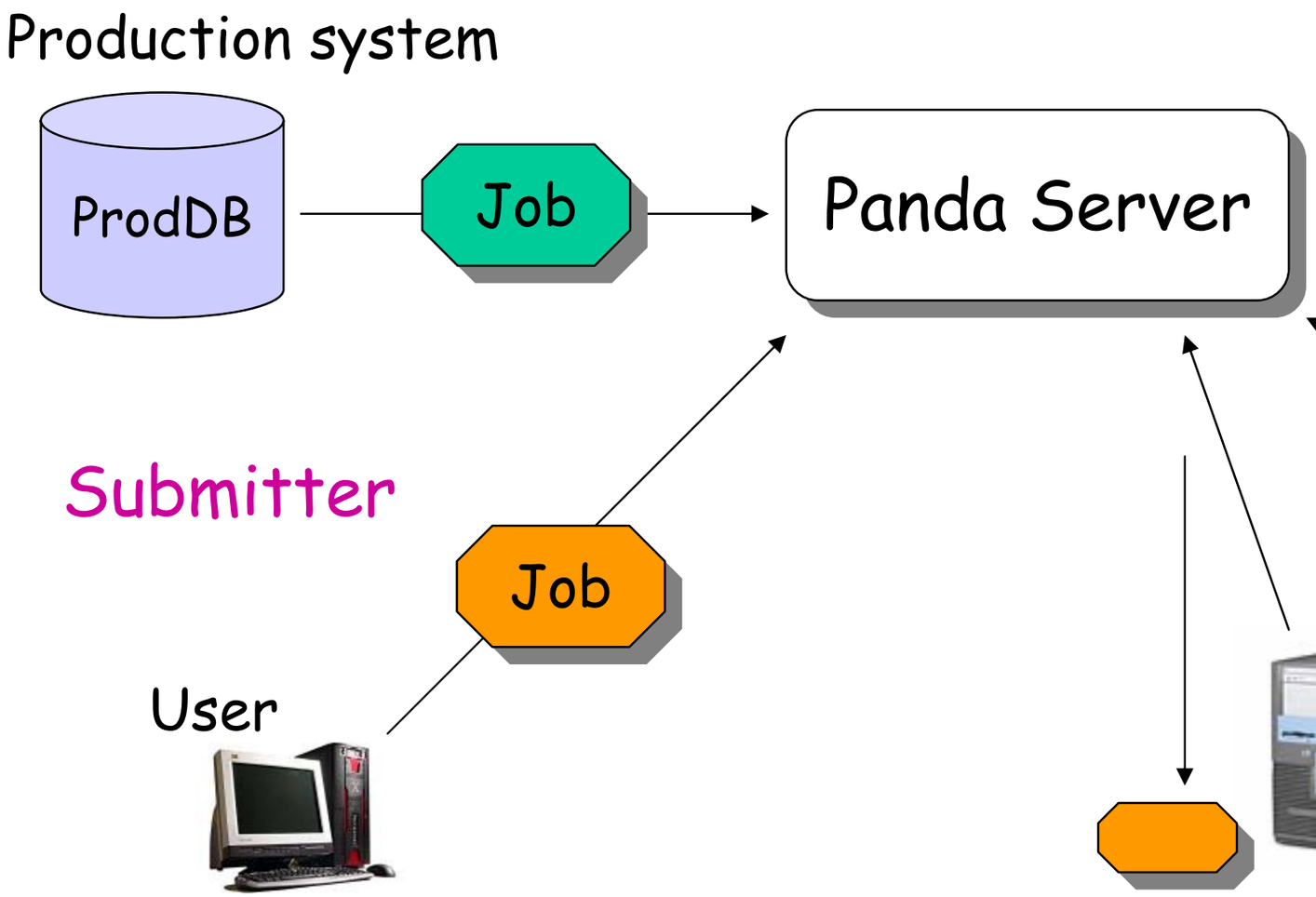
The state information is in the DB
which guarantees synchronization.
The DB is used as a buffer. Records
are moved to a disk-resident DB
after several days.

T. Maeno

# Job Flow




Client-server communication:
mostly asynchronous

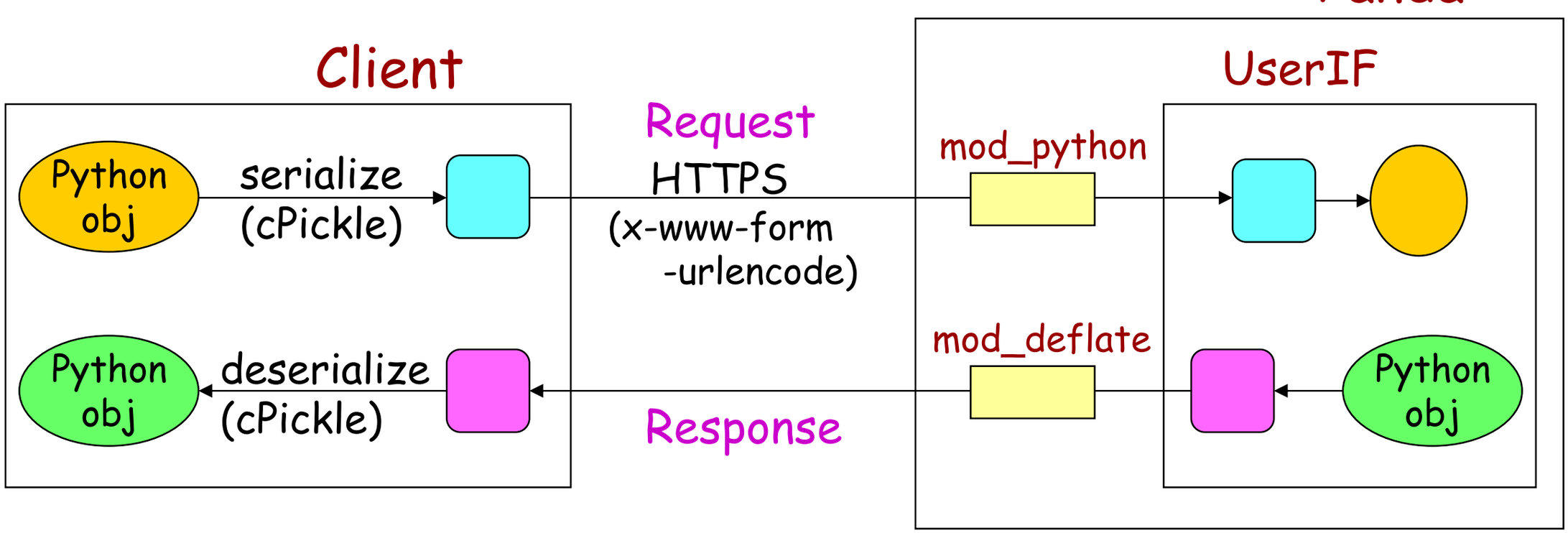



Each pilot runs on a worker node
1. send a request
2. receives a job
3. runs the job

T. Maeno

# Client Interface to Panda Server

- Pickle module of Python and native curl
  - External component is not needed
  - Pickle is faster than XML parser
- Client require python 2.3 or higher, curl and a grid-proxy, but not grid middleware
- Simple, light-weight, independent from grid-flavor

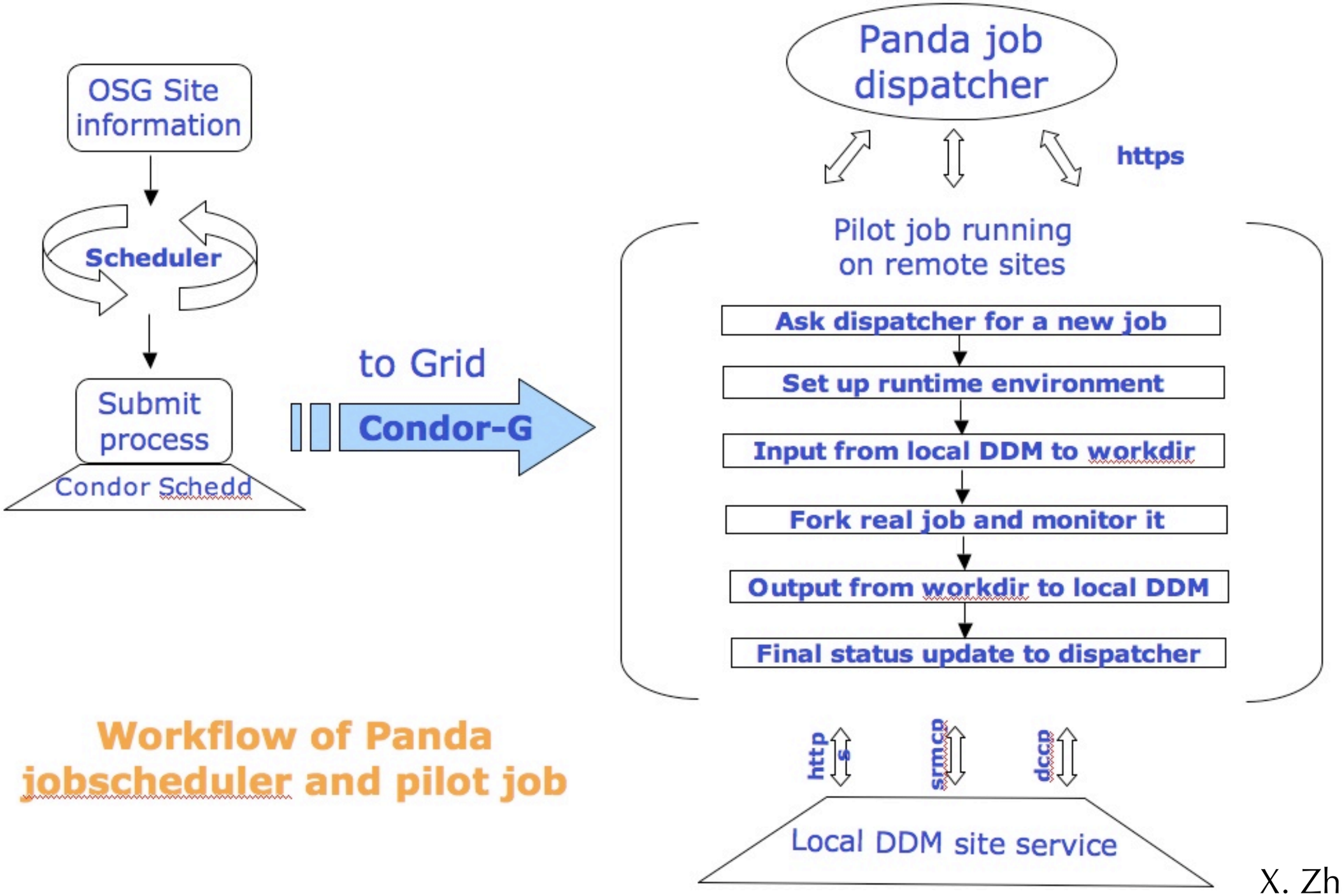



T. Maeno

# Panda Brokerage

- Core Panda server incorporates all **brokerage** managing where jobs (and associated data) are sent based on job characteristics, data locality, priorities, user/group role, site resources & capabilities matched to job needs, dynamic site info & throughput from Panda logging, ...
  - Not all there yet; being progressively implemented
  - Need improvements in (particularly dynamic) site info gathering
- Principal brokerage function is managed data flow: broker sends 'dispatch dataset' associated with set of jobs to a site, receives notification of transfer completion, and then releases jobs
  - ie file transfer and job execution are pipelined

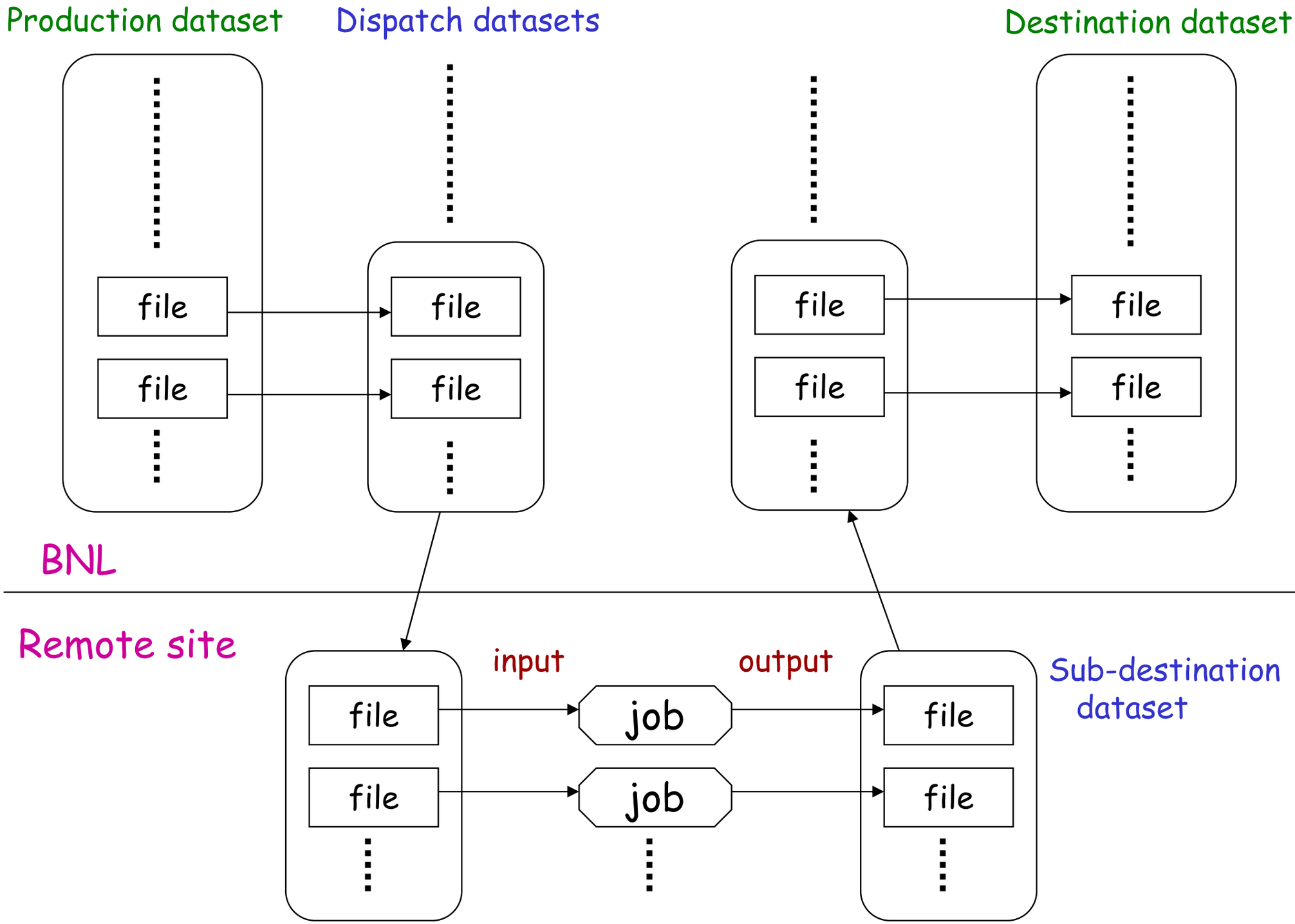
# Scheduler and Pilot




**Workflow of Panda jobscheduler and pilot job**

X. Zhao

# Dataset-based Data Flow




T. Maeno

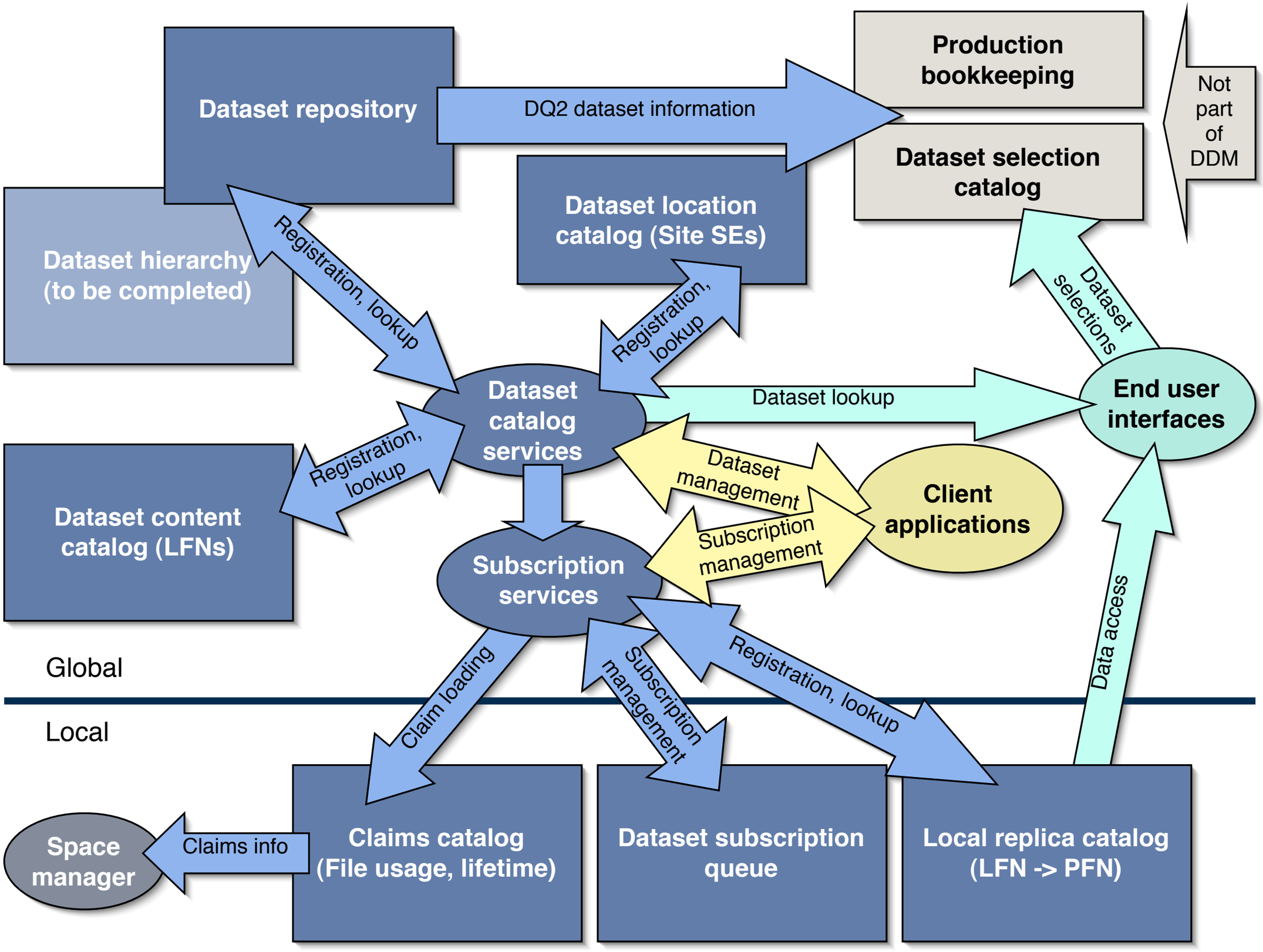
# DDM in ATLAS - DQ2

- Don Quijote 2 (DQ2) initiated in March 2005 to provide a comprehensive DDM tool for current needs and for datataking
  - Development met an aggressive schedule
  - Deployed in production in fall 2005
- Design driven by scalability, manageability, no unsafe dependencies, match to computing model
- Supports
  - all ATLAS grid environments (LCG, OSG, NG as needed)
  - any file-based data (event data, conditions, other)
  - all stages of the data flow, EF -> T0 -> Grid tiers -> Institutes, laptops, end users
  - all processing environments, managed production to individual analysis
  - both global ATLAS and regional data management

# DQ2 Principal Features

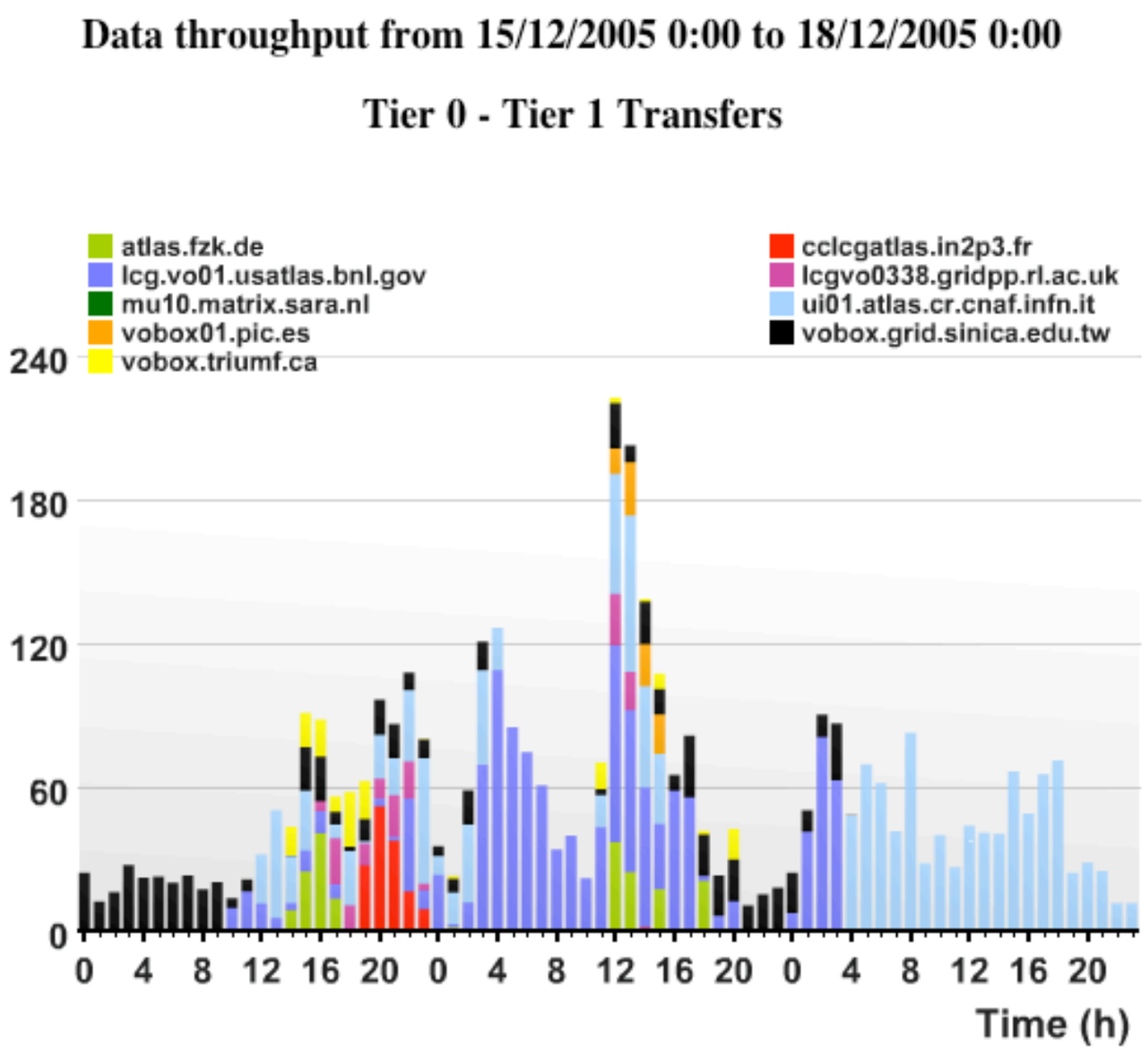
- Scalable global data discovery and access via catalog hierarchy: *datasets* as collections of logical files
  - May be mutable & versionable, or immutable (blocks)
- *Data blocks* are datasets specifically designed for replication and global data discovery
  - Content is immutable so that replica completeness is well defined, and dataset-based lookup is reliable without complex mechanisms to maintain global consistency
- Data movement based on *subscriptions* to datasets: client subscribes to trigger DS replication to a local site
- No global physical file replica catalog
  - Physical file info available and managed locally only
- Grid certificate based security/authentication

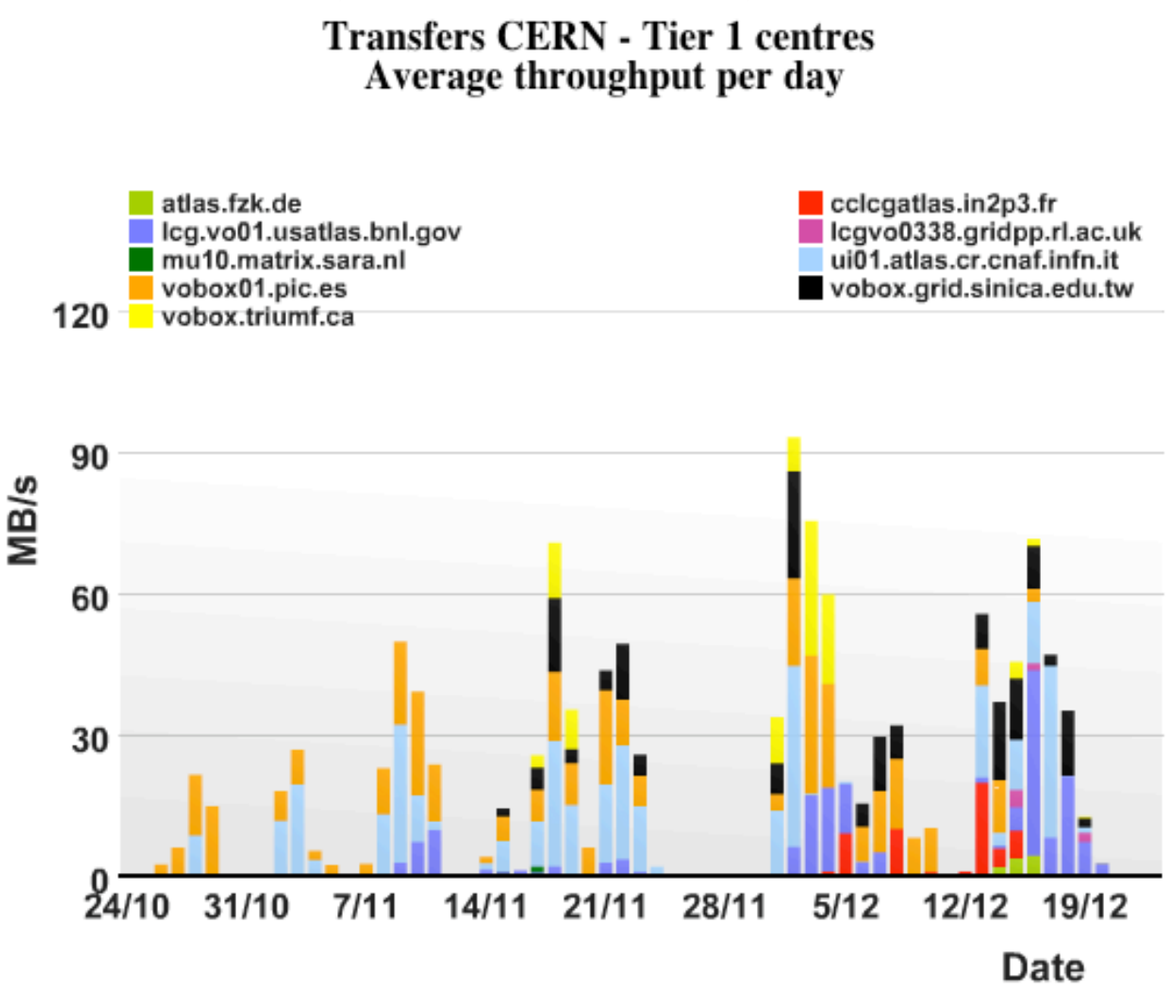
# DQ2 Architecture



# DQ2 Scaling in Nov-Dec ATLAS Tier0 Test

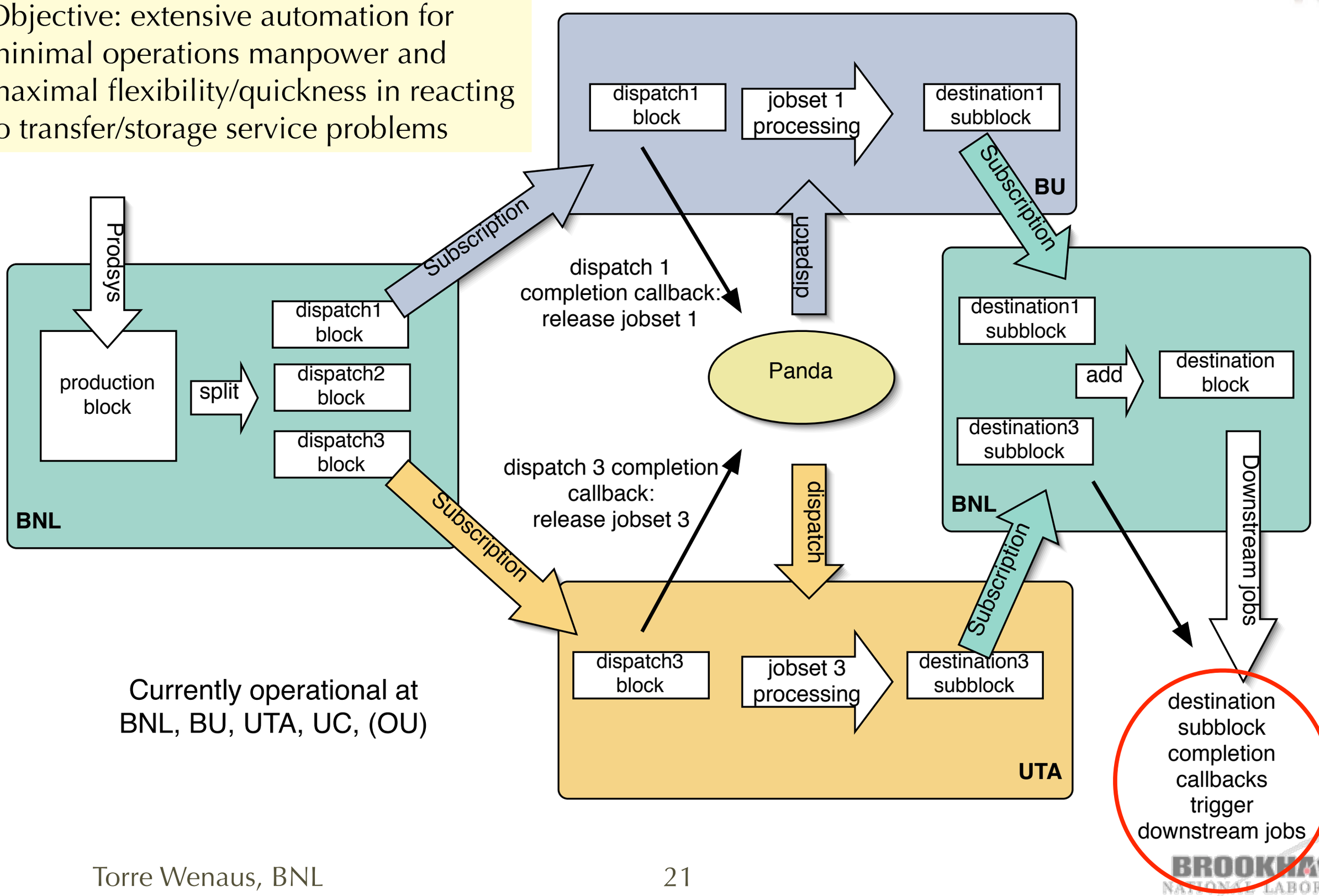
- Goal of 10% nominal throughput (70 MB/s) out of T0 exceeded
  - 90MB/s for 24h, 200MB/s peak
- Interruptions due to site instabilities
- No DQ2 scaling limits seen (none were expected)
- Repeat in ~June 2006 (SC4) with goal of 100% (720 MB/s to T1s)



# DQ2 Based Data Handling in Panda

Objective: extensive automation for minimal operations manpower and maximal flexibility/quickness in reacting to transfer/storage service problems




Currently operational at BNL, BU, UTA, UC, (OU)

# DDM for Panda at US Sites

- DDM-capable sites:
  - BNL
  - BU
  - UTA
  - UC
  - OU in progress
  - IU in progress
  - SLAC in progress

**Storages currently in use for production:**
gsiftp://atlas.bu.edu/data2/dq-cache/
gsiftp://atlas.bu.edu/data3/dq2-cache/
gsiftp://atlas.dpcc.uta.edu/data33/ATLAS/dq2/
gsiftp://atlas.dpcc.uta.edu/data43/ATLAS/dq2/
gsiftp://atlas.dpcc.uta.edu/data81/ATLAS/dq2/
gsiftp://osg-itb.dpcc.uta.edu/data33/ATLAS/dq2/
gsiftp://osg-itb.dpcc.uta.edu/data43/ATLAS/dq2/
gsiftp://osg-itb.dpcc.uta.edu/data81/ATLAS/dq2/
gsiftp://tier2-03.uchicago.edu/share/data6/atlas_se_ddm/
srm://dcsrm.usatlas.bnl.gov/pnfs/usatlas.bnl.gov/data/prod/pandadev/

**DQ2 site services:**
http://yakut04.slac.stanford.edu:8000/dq2/
http://g3aux.avidd.iu.edu:8000/dq2/
http://ouhep00.nhn.ou.edu:8000/dq2/
http://atlas001.bu.edu:8000/dq2/
http://osg-itb2.dpcc.uta.edu:8000/dq2/
http://tier2-01.uchicago.edu:8000/dq2/
http://dms02.usatlas.bnl.gov:8000/dq2/

# Panda DDM Experience with DQ2

- Extremely effective overall in its performance and its effect on production through *managed* data flow
- Central dataset catalogs, services
  - Stably functional, still with some optimization to do for performance/scalability
  - Panda just migrated easily to CERN-based central catalogs; transatlantic latency visible but manageable
- Site-level services
  - Principal DQ2 problem: site services hard to deploy
    - Packaging, documentation, dependencies, complexity, all need improvement
    - New version next week should address
    - DQ2 in use at 5 ATLAS Tier 1s + 6 smaller US sites
    - Tier 1s and, in US, Tier 2s will act as 'regional hubs' for DDM; no need to deploy to all active sites
- Middleware
  - DQ2 has minimal middleware dependencies (file catalog, data mover) and supports fallbacks/alternates
  - Not seeing frequent or protracted downtime due to it (thanks sometimes to fallbacks)

# Tasks/Datasets In Action

Run by wenaus

**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
Subscriptions

**Sites**
Site specs
BNL BU OU UC
UTA LCG NG All

System statistics

Logging monitor

## Task browser

Click for help

Current selection: STATUS=running&GRID=osg Clear selection

**Click to show and select physics types**

**Releases:** v11000301 (46) v11000302 (77) v11000303 (61) v11000304 (12) v11000305 (37) v11000306 (9) v11000307 (6) v11000308 (33) v11000401 (1)
**Stages:** digit (99) evgen (74) merge (18) reco (24) recon (67)
**Outputs:** AOD (106) CBNT (97) ESD (88) EVNT (74) HIST (9) HITS (102) RDO (102) TAG (18)
**Grids:** lcg (28) anygrid (1) nordic (52) osg (81) lcg-cg (120)
**Status:** aborted (12) done (115) finished (13) rejected (1) running (108) submitted (33)

Total selected events=2160000 jobs=21095 jobs done=16645

| Task name | Task ID | Status | Grid | Total jobs | Done jobs | Events | Input files | Release | Formats |
|---|---|---|---|---|---|---|---|---|---|
| mc11.007204.singlepart_mu4.recon.v11000302 | 694 | running | osg | 2200 | 1498 | 220000 | 2200 | 11.0.3 | ESD.AOD.CBNT |
| mc11.007430.singlepart_singlepi_pt2.digit.v11000308 | 667 | running | osg | 500 | 331 | 100000 | 20 | 11.0.3 | RDO.HITS |
| mc11.007200.singlepart_mu2.recon.v11000303 | 579 | running | osg | 500 | 464 | 50000 | 500 | 11.0.3 | ESD.AOD.CBNT |
| mc11.007200.singlepart_mu2.digit.v11000303 | | | | | | | | | RDO.HITS |
| mc11.007222.singlepart_mu26.recon.v1100030 | | | | | | | | | ESD.AOD.CBNT |
| mc11.007216.singlepart_mu18.recon.v1100030 | | | | | | | | | ESD.AOD.CBNT |
| mc11.007211.singlepart_mu10.recon.v1100030 | | | | | | | | | ESD.AOD.CBNT |
| mc11.007207.singlepart_mu6.recon.v1100030 | | | | | | | | | ESD.AOD.CBNT |
| mc11.007216.singlepart_mu18.digit.v1100030 | | | | | | | | | RDO.HITS |
| mc11.007207.singlepart_mu6.digit.v11000302 | | | | | | | | | RDO.HITS |
| mc11.005001.pythia_minbias.recon.v1100030 | | | | | | | | | ESD.AOD.CBNT |
| mc11.005001.pythia_minbia digit.v11000304 | | | | | | | | | RDO.HITS |
| mc11.005800.JF17_pythia iet_filter.reco | | | | | | | | | ESD.AOD.CBNT |
| mc11.005800.JF17_pythia_lo filter.digit.v | | | | | | | | | RDO.HITS |
| mc11.005055.PythiaPhotonJet1. v1 03 | | | | | | | | | ESD.AOD.CBNT |
| mc11.005056.PythiaPhotonJet2.reco 3 | | | | | | | | | ESD.AOD.CBNT |
| mc11.005056.PythiaPhotonJet2.di | | | | | | | | | RDO.HITS |

## Task mc11.007216.singlepart_mu18.recon.v11000303

**Datasets for task mc11.007216.singlepart_mu18.recon.v11000303**

mc11.007216.singlepart_mu18.recon.ESD.v11000303
mc11.007216.singlepart_mu18.recon.AOD.v11000303
mc11.007216.singlepart_mu18.recon.CBNT.v11000303

**Parameters for task mc11.007216.singlepart_mu18.recon.v11000303**

| | |
|---|---|
| **Task ID** | **562** |
| Project | mc11 |
| Input dataset | mc11.007216.singlepart_mu18.digit.v11000302 |
| Task name | mc11.007216.singlepart_mu18.recon.v11000303 |
| Formats | ESD.AOD.CBNT |
| Transformation | csc.reco.trf |
| Trf Version | 11.0.3.3 |
| Release | 11.0.3 |
| Owner | i_hinchliffe@lbl.gov |
| CPU/event | 100 |
| Memory usage | 600 |
| First inputfile number | 1 |
| Input files | 2200 |
| Events | 220000 |
| Events/file | 100 |
| Grid | osg |

## Task browser

Click for help

Current selection: STATUS=running&GRID=osg

| Physics type | Events |
|---|---|
| A3_Ztautau_tightfilter | 20000 |
| AlpgenJimmyW4jet | 80000 |
| Bs_Jpsi_mu6mu3_phi_KplusKminus | 10000 |
| Electron_Pt_25 | 20000 |
| Electrons_e100 | 40000 |
| FJ1_fwjets_e200 | 20000 |
| FJ2_pythia_jetjet | 100000 |
| Gmm_500_pythia_photos | 250000 |
| H3_120_gamgam | 20000 |
| J1_pythia_jetjet | 150000 |
| J2_Pt_35_70 | 20000 |
| J2_pythia_jetjet | 150000 |
| J3_Pt_70_140 | 20000 |
| J3_pythia_jetjet | 150000 |
| J4_pythia_jetjet | 150000 |
| J5_Pt_280_560 | 20000 |
| J5_pythia_jetjet | 150000 |
| J6_Pt_560_1120 | 30000 |
| J6_pythia_jetjet | 50000 |
| J7_pythia_jetjet | 30000 |
| J8_pythia_jetjet | 30000 |
| JF17_pythia_jet_filter | 3000000 |
| JF17_pythia_loosejet_filter | 1800000 |
| LRSM_WR_1800_300 | 60000 |
| M1_minbias | 20000 |
| McAtNloWenu | 45000 |
| McAtNloWmunu | 60000 |
| P5P_Single211 | 40000 |
| P7P_Single211 | 20000 |
| Photon_Pt_60 | 20000 |
| Photons_e100 | 40000 |
| PythiaH120gamgam | 240000 |
| PythiaH130zz4l | 200000 |

**Tasks** define production tasks and record their associated metadata

**Task Query Form**

- *Text fields don't require the exact matching*
- Queries in *italic* not implemetted yet

**MC Task**

| | |
|---|---|
| Project : | mc11 |
| Input Dataset | |
| Transformation : | csc.reco.trf |
| Transformation Version : | 11.0.4.1 |
| Grid Flavour : | osg |
| Requested By | |
| Output Task Name | |
| Priority | |
| Status : | running |

**Datasets** define and organize the task inputs and outputs

BROOKHAVEN NATIONAL LABORATORY

# DQ2 Datasets In Action

**DQ2** catalogs datasets...

...provides user-level access: dq2_get, dq2_ls, (dq2_put coming)...

...provides automated management/movement tools for use in production...

...organizes validation...

...and ultimately lets you get at the files!

**Dataset mc11.007216.singlepart_mu18.recon.AOD.v11000303**

**Task corresponding to dataset:** mc11.007216.singlepart_mu18.recon.v11000303

**Produced on grid(s):** osg

**Availability in DQ2:**
Registered in osg DQ2, creation date Mon Jan 23 02:25:50 2006, vuid = 58aed7fc-4057-4b3c-b6e4-73f1f2fa152c
Not registered in lcg DQ2 instance

**Data access**
Instructions follow on how to access the data on the grids producing or holding it.
The instructions will become simpler once all grids are using DQ2.
In order to set up your environment to use the commands described,
follow the instructions here for running the setup script and initializing your grid certificate.

**Access to data on OSG and LCG:**
The dq2_get command can be used to retrieve the full dataset

```
dq2_get mc11.007216.singlepart_mu18.recon.AOD.v11000303
```

or a subset of files from it.
Internally dq2_get uses DQ2 where available (OSG) and direct LFC/LCG tools where necessary (LCG).
The dq2_ls command allows wildcarded listings of datasets or files within them, for DQ2 resident datasets.
See the documentation for dq2_get and dq_ls details.

**Dataset info from Panda:**
Type=output Status=running Created 2006-01-23 02:25:54 Modified 2006-01-26 14:57:57

Look for recent or all Panda jobs using this dataset for input or output
Look for recent or all Panda jobs using this dataset for input
Look for recent or all Panda jobs using this dataset for output

Click to show 512 subdatasets

BNL-validated dataset mc11.007216.singlepart_mu18.recon.AOD.v11000303_bnl available

OSG sites holding the dataset: BNL

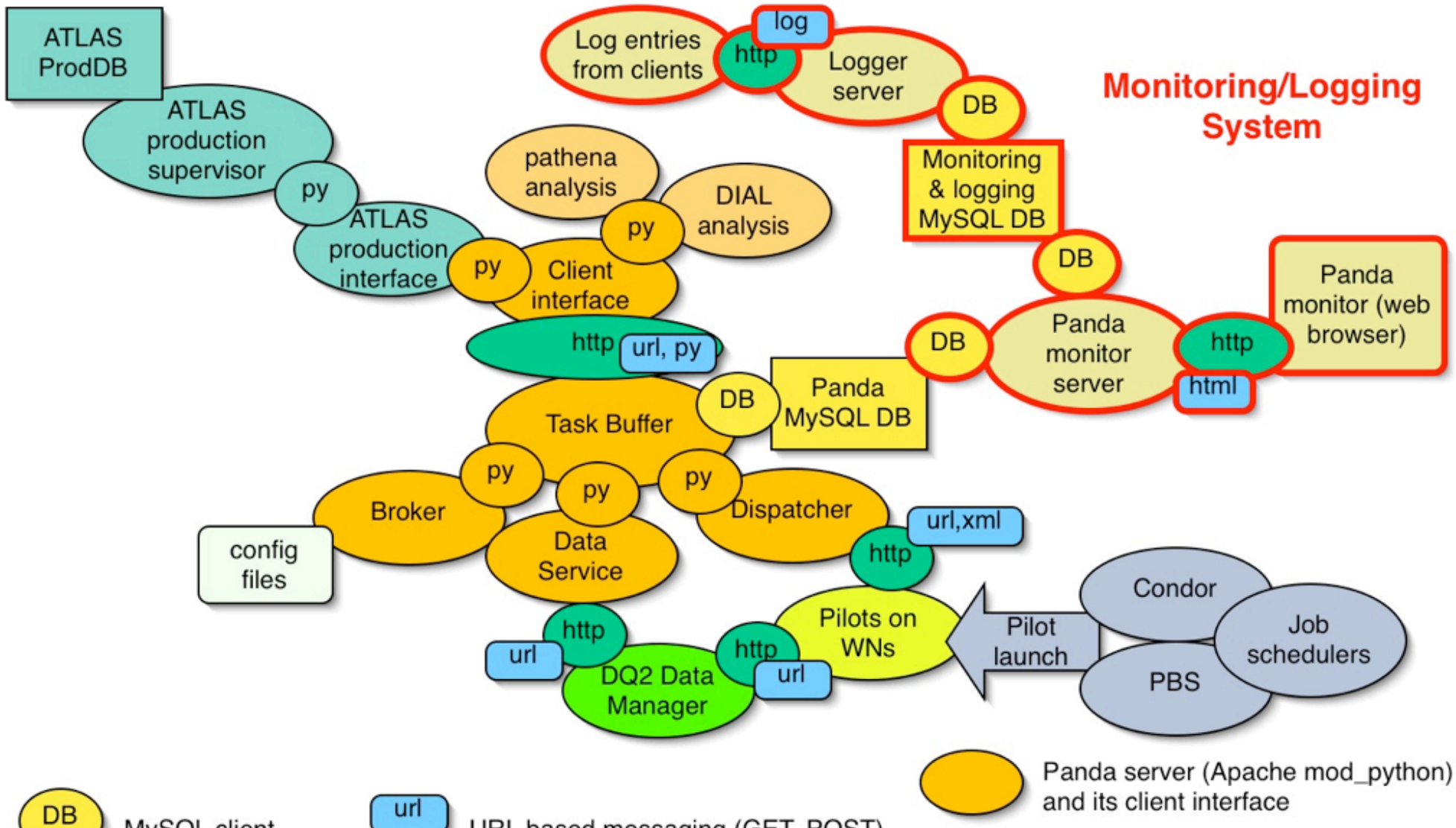
338 constituent logical files in dataset mc11.007216.singlepart_mu18.recon.AOD.v11000303:

| LFN | Click to view site replicas | |
|---|---|---|
| mc11.007216.singlepart_mu18.recon.AOD.v11000303._00001.pool.root.1 | BNL | At BNL |
| mc11.007216.singlepart_mu18.recon.AOD.v11000303._00002.pool.root.1 | BNL | At BNL |
| mc11.007216.singlepart_mu18.recon.AOD.v11000303._00003.pool.root.1 | BNL | At BNL |
| mc11.007216.singlepart_mu18.recon.AOD.v11000303._00004.pool.root.1 | BNL | At BNL |

# Monitoring & Logging in Panda

- Single comprehensive web interface to
  - Make the internals of the system open and visible
  - Reduce operations workload through easier problem discovery, diagnostics
    - Quickly identify underutilized and dysfunctional resources
    - Provide dynamic, real-time info repository on resource availability, performance for intelligent brokering
    - Gather statistics on performance, efficiency, etc.
    - Aid users in discovering and using produced data
- Found to be very useful for production operations and lessening workload for people on shift

# Monitoring/Logging System



# Available Via Panda Monitor

- Production monitoring
  - Operations 'dashboard'
  - Throughput, error reporting by site; 30+ error types tracked
  - Job queue content, job level info, log file access
  - DQ2 status, active production datasets, subscriptions
- Site info
  - Production activity, aggregate and to WN level
  - Disk space monitoring, DDM configuration, resident data
  - Grid monitor access (GridCat, MonaLisa, local monitors)
- Data discovery and access
  - Dataset search, browsing and access to datasets/files
- Logging system
  - Client incident reports, dynamic status and history
- MC production task definition & management

Configuration | **Dashboards:** Production Analysis DDM Physics data Task definition

**Panda monitor**

# Panda Production Operations Dashboard

Quick guide, twiki | **Panda shift guide** and calendar

**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
Dataset browser
Subscriptions

**Sites**
Recent activity
BNL BU OU UC
UTA LCG NG All

**System statistics**

**Logging monitor**

**Servers:** Panda:OK Logger:OK DQ2:OK
**Tasks assigned to OSG**

**Jobs updated <12 hrs ago: Running**:489 **Activated**:101
**Jobs updated >12 hrs ago: Activated**:335

**Space available at sites:**

| Site | GB | As of |
|---|---|---|
| BNL_ATLAS_1 | 999999 | 03-05 06:38 |
| BNL_ATLAS_2 | 999999 | 03-05 10:32 |
| BU_ATLAS_Tier2 | 1085 | 03-05 10:44 |
| OUHEP_OSG | 1659 | 03-05 08:21 |
| UC_ATLAS_MWT2 | 635 | 03-04 04:41 |
| UTA-DPCC | 1076 | 03-05 10:45 |

**Recent site activity history**

**Pilot requests for jobs, last 3 hours**
Red indicates <10/hr (prod), <30/hr (analysis)

| | Production | Analysis |
|---|---|---|
| BNL_ATLAS_1 | 0 | 0 |
| BNL_ATLAS_2 | 1266 | 156 |
| BU_ATLAS_Tier2 | 27 | 3 |
| OUHEP_OSG | 3 | 3 |
| OU_OSCER_OSG | 0 | 0 |
| UC_ATLAS_MWT2 | 0 | 0 |
| UTA-DPCC | 10 | 4 |

**Production job summary, last 12 hours** (Details: errors, nodes)

| Site | Nodes | Total jobs | Latest | defined | assigned | waiting | activated | running | finished | failed | %fail total/trans/other |
|---|---|---|---|---|---|---|---|---|---|---|---|
| All | 245 | 1319 | 03-05 10:46 | 0 | 91 | 250 | 93 | 470 | 390 | 25 | 6.0% / 3.4% / 2.7% |
| BNL_ATLAS_1 | 10 | 26 | 03-05 10:46 | 0 | 8 | 0 | 9 | 9 | 0 | 0 | |
| BNL_ATLAS_2 | 103 | 526 | 03-05 10:46 | 0 | 83 | 0 | 71 | 156 | 211 | 5 | 2.3% / 1.9% / 0.5% |
| BU_ATLAS_Tier2 | 28 | 189 | 03-05 10:46 | 0 | 0 | 0 | 0 | 109 | 80 | 0 | 0.0% / 0.0% / 0.0% |
| NULL | 1 | 250 | 03-05 05:15 | 0 | 0 | 250 | 0 | 0 | 0 | 0 | |
| OUHEP_OSG | 13 | 41 | 03-05 10:40 | 0 | 0 | 0 | 13 | 11 | 7 | 10 | 58.8% / 41.2% / 17.6% |
| OU_OSCER_OSG | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| UC_ATLAS_MWT2 | 44 | 84 | 03-05 10:46 | 0 | 0 | 0 | 0 | 84 | 0 | 0 | |
| UTA-DPCC | 54 | 203 | 03-05 10:46 | 0 | 0 | 0 | 0 | 101 | 92 | 10 | 9.8% / 2.9% / 6.9% |

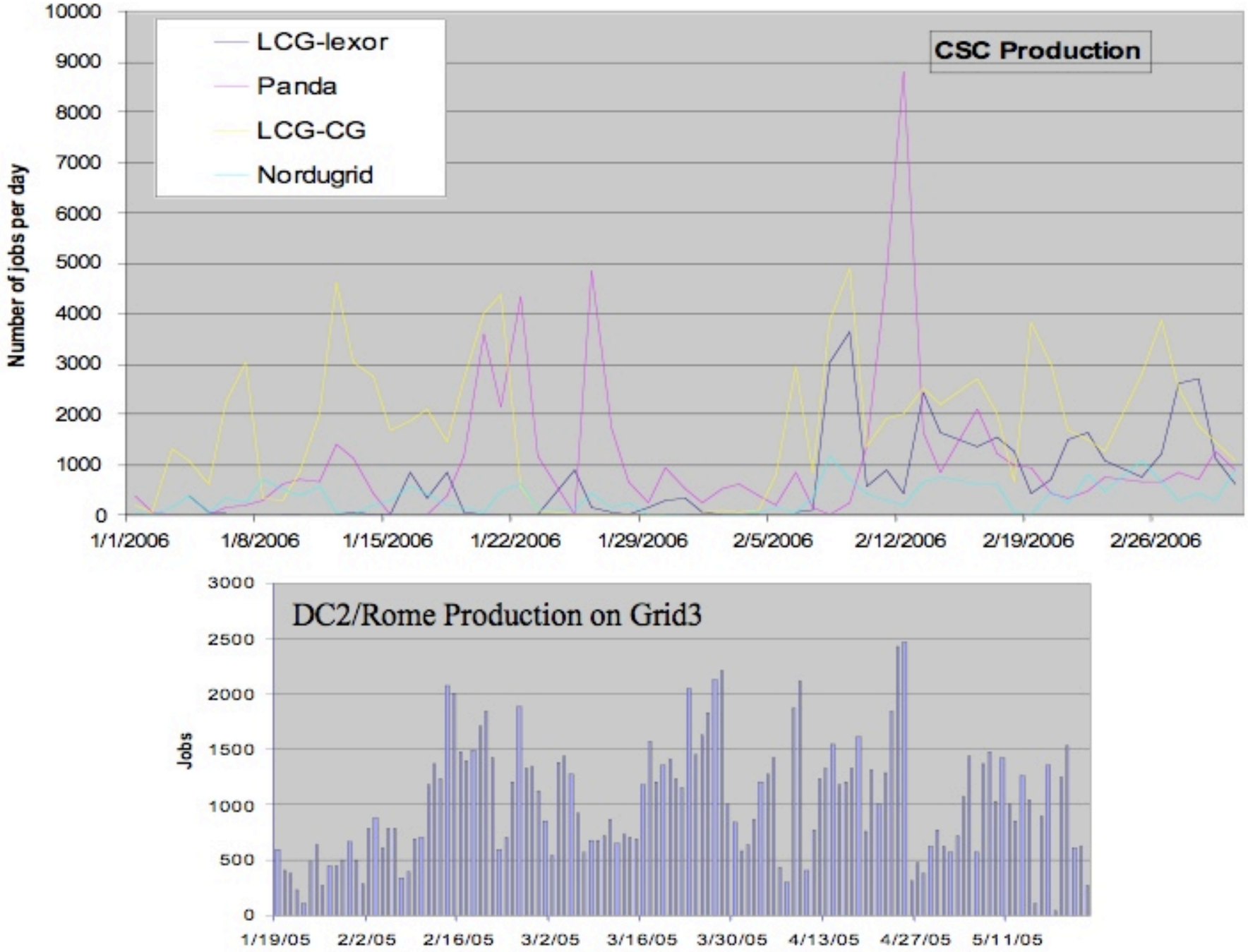
**Analysis job summary, last 12 hours** (Details: errors, nodes )

| Site | Nodes | Total jobs | Latest | defined | assigned | waiting | activated | running | finished | failed | %fail total/trans/other |
|---|---|---|---|---|---|---|---|---|---|---|---|
| All | 5 | 64 | 03-05 06:15 | 0 | 0 | 0 | 0 | 0 | 0 | 64 | |
| BU_ATLAS_test | 1 | 64 | 03-05 06:15 | 0 | 0 | 0 | 0 | 0 | 0 | 64 | |

**Summary of Panda subscriptions, last 12 hours (details here):**

| Dispatch blocks | Destination blocks | Other |
|---|---|---|

# Panda in Production







Steady utilization of our 400-500 CPUs (as long as jobs are available)

~9000 jobs/day in brief 'scaling test'; no scaling limit found, or expected (Target is ~10-20 higher)

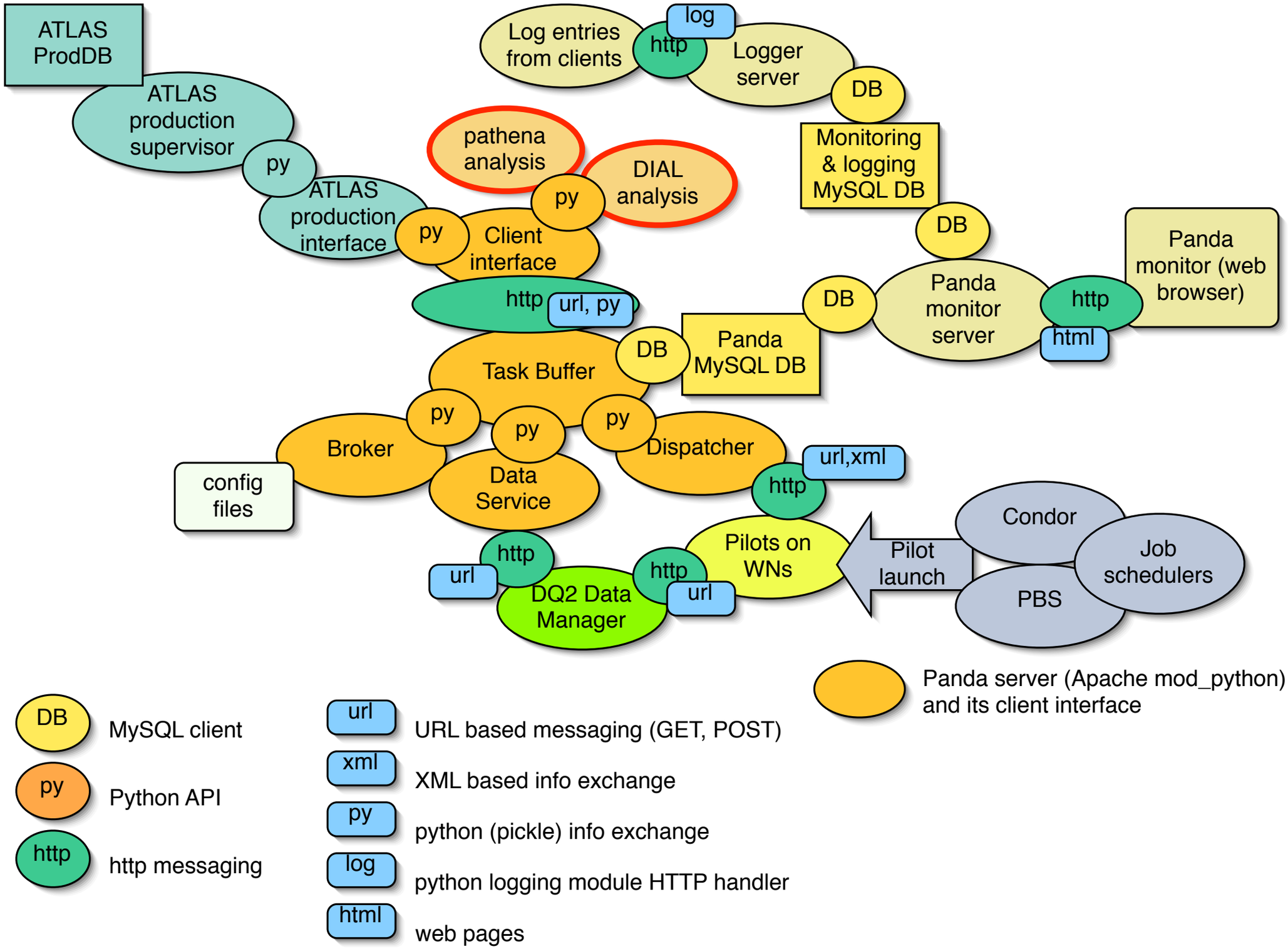
Lowest failure rate among executors (generally well under 10%)

Half the shift manpower compared to previous

Previous system never able to exceed 2500 jobs/day in production

Kaushik De

# Distributed Analysis (DA) in Panda



# Key Panda Features for DA

- Designed from beginning to support **distributed analysis usage**
  - interactive analysis, user-level job submission, regional group production
  - grid-based (CondorG) or farm-based (batch system) resources
- **Dataset** based organization of Panda matches the DDM system and the **analysis work model**
- Use of DDM to **pre-stage input data** and **immediately return outputs**, all asynchronously, minimizes data transport latencies and delivers earliest possible first results
- Management/optimization of workload via **job queue** with **late binding** of jobs to worker nodes gives **dynamic and flexible** system response to **highly variable DA work**
- Use of grid and/or farm batch queues to **pre-stage job wrappers** to worker nodes (pilot jobs) and **directly deliver workloads** from Panda allows **fast injection of DA work** (a work in progress!)

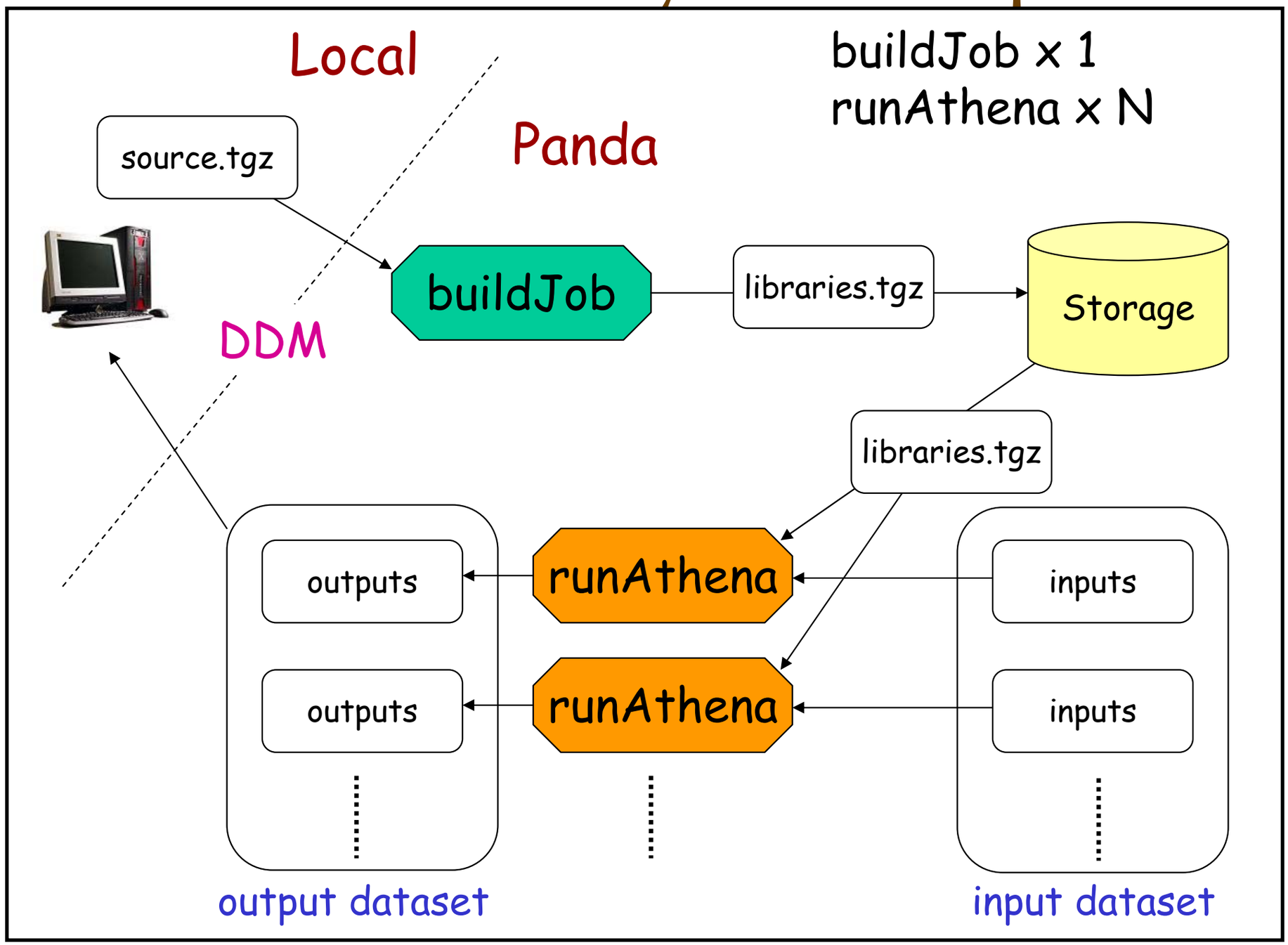
# Key Panda Features for DA (2)

- Support for packaging, deploying, running **arbitrary user code/jobs**
  - Arbitrary scripts can be specified by job and loaded via http retrieval for execution
- **Comprehensible system view** offered to users: **heterogeneous** distributed resources appear as one **uniform** resource accessed through standard interface
- Easy to **integrate your own local resources**: site requirements are pilot delivery (via local batch queue or grid), outbound http, and access to a DQ2-enabled SE (locally or 'nearby') (only Tier 2s so far)
- Easy-to-use **client interface** makes integration with diverse analysis/ interactive front ends easy
- **User ID** built into Panda DB; **monitoring and metadata** extensible to **user level (**User ID (DN) is recorded, user-level extensions not in yet)
- **User-level controls, quotas** directly implementable in Panda's brokerage rules (not implemented yet)

# Athena Based Analysis via Panda

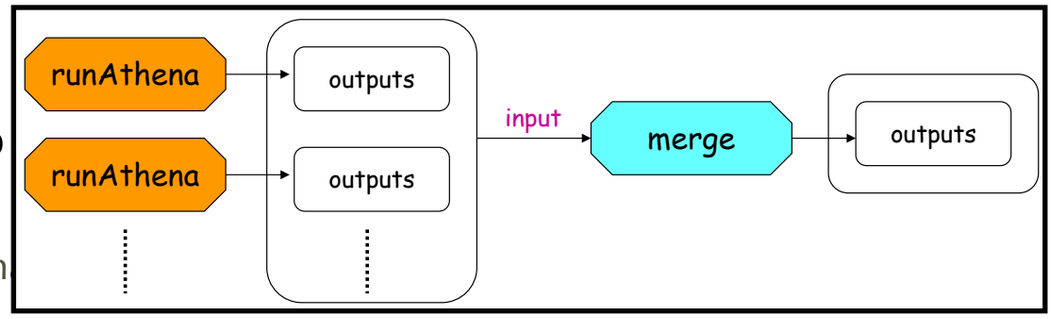
- Athena is ATLAS offline Framework
- **pathena** is an adaptation of the 'athena' command line invocation of framework to process via Panda
  - pathena processes user submission in two steps:
    - Build step: gather up user code, store it (webdav) and ship it to processing site for (http) retrieval by the pilot and site installation/build in user area
    - Run step: run N Athena jobs with user-designated input and output datasets
  - User retrieves output dataset via DDM tool dq2_get
- **DIAL** also supports Panda as processing back end, but main focus going forward will be on direct distributed analysis support in Panda

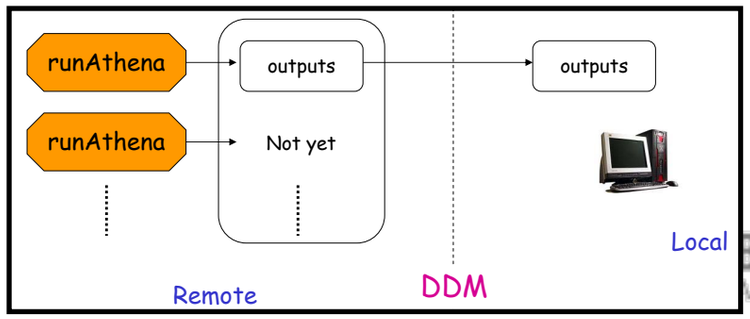
# Panda Based Analysis with pathena




T. Maeno

Can include a third merge step




Partial results delivery

Torre Wen

# User Access to Data

- DQ2 end-user tools
  - Documentation
    https://uimon.cern.ch/twiki/bin/view/Atlas/AccessPandaData
- dq2_get
  - copy files over the grid
  - Files need to be registered in DQ2 catalog
    - OSG/Panda : done
    - LCG : still using LFC
    - NG : on going
  - Working with OSG, LCG (temporary hack). Once NG prod system is integrated with DQ2, NG datasets are available automatically
  - Todo
    - 3rd party transfer
    - Parallel transfer
    - md5 sum check

**Work-flow**

1. Use dq2_ls to look for interesting datasets
2. Submit jobs using pathena
3. Jobs runs on LCG/OSG/NG production system
4. Output datasets are registered to DQ2 catalog automatically
5. Get output by using dq2_get

- dq2_ls
  - List datasets and files in DQ2 catalog
  - Allow users to search for datasets
  - Wildcard search
    > dq_ls mc11.*RDO*

- dq2_put
  - ~~not yet~~
  - Register files in DQ2 catalog
  - Files produced on production system are registered to DQ2 catalog automatically
    - regional production

# Dataset Browser

**DQ2 dataset browser, category csc, datasets at BNL**

Click for help
Dataset lists last updated 03-07 02:18 (updates are currently nightly)

**Select a dataset category** *Category counts are totals, exclusive of selections or site restrictions*

| Category | Count | Description |
|---|---|---|
| all | 16079 | All datasets |
| csc | 333 | Computing system commissioning production |
| ctb | 8 | Combined testbeam production |
| dc2 | 6 | Data Challenge 2 production |
| destination | 9271 | Panda destination sub-blocks |
| dial | 14 | DIAL analysis |
| dispatch | 4307 | Panda dispatch blocks |
| larg | 5 | LAr commissioning |
| mc | 416 | MC production |
| other | 166 | Everything else |
| rome | 77 | Rome physics workshop production |
| testpanda | 553 | Panda test datasets |
| tile | 9 | Tilecal commissioning |

**Restricted to BNL resident datasets** (clear) (Restrict to datasets *not* at BNL)
Choose another site: ASCC BNL BNL-SC BU CERN CNAF FZK IN2P3 OU PIC TRIUMF UC UTA

**Selected category: csc** (clear category)

**csc fields:** **Selections:** release=v11004103 (clear selections)

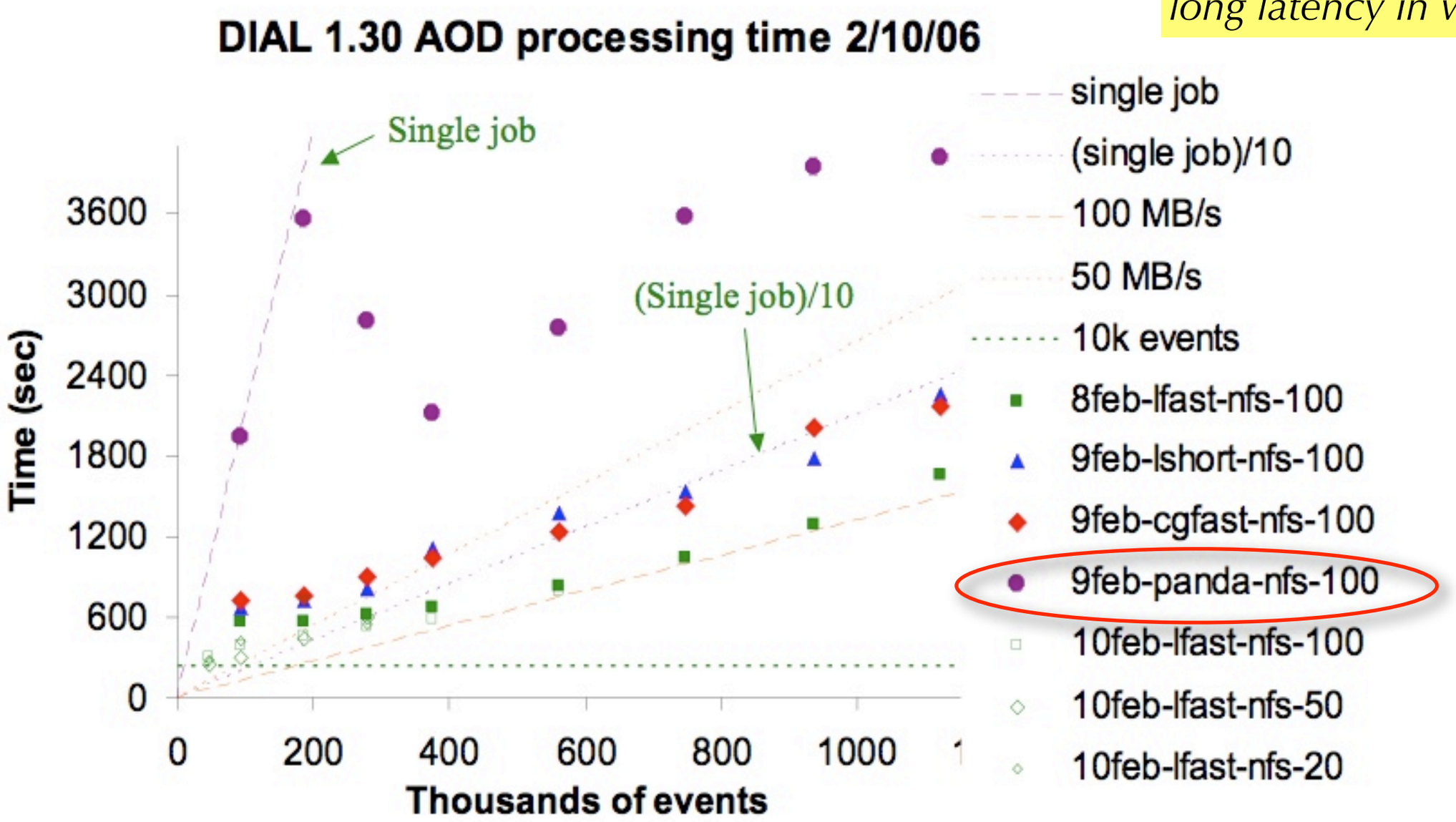
| series (1) | number (29) | physics (29) | stage (4) | format (7) | release (1) |
|---|---|---|---|---|---|
| csc11 (122) | 007086 (4) | WH400uu_pythia (5) | recon (105) | HITS (3) | v11004103 (122) |
| | 005403 (4) | pythia_minbias (4) | evgen (8) | AOD (27) | |
| | 007604 (1) | PythiaWenu_pt100 (4) | digit (6) | log (33) | |
| | 007211 (4) | singlepart_mu6 (4) | simul (3) | ESD (26) | |
| | 007216 (4) | singlepart_mu4 (4) | | RDO (3) | |
| | 005630 (4) | singlepart_gamma_Et60 (4) | | CBNT (26) | |
| | 005857 (5) | singlepart_singlepi_pt2 (4) | | EVNT (4) | |
| | 005854 (9) | PythiaPhotonJet2 (4) | | | |
| | 007401 (4) | PythiaPhotonJet1 (4) | | | |
| | 007085 (4) | singlepart_mu10 (4) | | | |
| | 005015 (4) | SU3_jimmy_susy (4) | | | |
| | 007222 (4) | PythiaZH120gamgam (4) | | | |
| | 005056 (4) | singlepart_mu18 (4) | | | |
| | 005055 (4) | JF17_pythia_loosejet_filter (4) | | | |
| | 005270 (4) | Gee_500_pythia_photos (4) | | | |
| | 005311 (4) | SingleTaupt100 (1) | | | |
| | 007200 (4) | Hplusplus150eemumu (2) | | | |
| | 005621 (9) | singlepart_gamma_E500 (4) | | | |
| | 005620 (4) | PythiaZmumu (4) | | | |
| | 005371 (2) | WH400bb_pythia (9) | | | |
| | 007204 (4) | JimmyWmunu (4) | | | |
| | 007207 (4) | singlepart_gamma_E1000 (4) | | | |
| | 007430 (4) | PythiaLQ300 (4) | | | |
| | 005800 (4) | singlepart_singlepi2 (4) | | | |
| | 005001 (4) | Gmm_500_pythia_photos (9) | | | |
| | 005145 (4) | J6_pythia_jetjet (4) | | | |
| | 005020 (4) | singlepart_mu26 (4) | | | |
| | 007042 (4) | singlepart_mu2 (4) | | | |
| | 005101 (4) | FJ1_pythia_jetjet (4) | | | |

**csc datasets matching selection and present at BNL (122):**
csc11.005001.pythia_minbias.recon.AOD.v11004103
csc11.005001.pythia_minbias.recon.CBNT.v11004103
csc11.005001.pythia_minbias.recon.ESD.v11004103

# But, Panda analysis performance today, er, *sucks!*

Study by D. Adams with DIAL using Panda based submission and LSF, CondorG submission

*We know why it sucks:*
*long latency in waiting for a pilot*



# Analysis Pilot Delivery

- Panda capability for interactive analysis relies on fast pilot delivery/job pickup
- Current analysis pilot delivery scheme: 1 in 10 pilots dedicated to analysis
  - Works poorly when system is saturated with long-running production jobs; new pilots are few and infrequent
- Two alternatives coming online
  - Send analysis pilots to dedicated short queues, decoupling analysis and production pilot delivery and (queue) resources
    - Will require optimization (and a feedback system?) to balance adequate pilot delivery rate against 'thrashing the system' with pilots popping in and out of existence when analysis load is low
    - Or, allow pilots to persist rather than terminating if no immediate work available

# Multitasking Pilots

- Second alternative: ‘multitasking pilots’
  - Currently a pilot manages a single ‘job thread’, running one job (production or analysis)
  - Add to pilots a second ‘job thread’ able to run a second job in parallel with the ‘primary’ job being managed by the pilot
    - ‘primary’ is a (long) production job; ‘secondary’ is a (short) analysis job
    - while primary runs, pilot regularly asks Panda for a secondary
    - a series of secondaries are run serially, while the primary runs
    - once the primary completes, the pilot exits when the current secondary completes
    - primary could be checkpointed or suspended when an analysis job is acquired, but won’t be initial approach

# Multitasking Pilots (2)

- Can open up the *entire Panda-managed resource pool as a live, listening analysis pool* for analysis pickup by pilots
  - Actual *utilization* of that pool for analysis can be easily controlled through brokerage, eg fixed CPU fraction by site
- No thrashing due to analysis-dedicated pilots starting/stopping continuously to be ready to pick up work
  - With 'all' pilots listening for analysis work, system has the dynamic range to cope with highly variable analysis workload
- But, some concerns too...
  - Conflicts with resource usage policies? Can at least do it on US ATLAS owned resources
  - Resource contention, particularly memory
  - Reinventing the wheel? May be able to utilize Condor...

# Panda and Condor

- Recently completed a SciDAC proposal for US ATLAS, Condor, CMS collaboration on “just-in-time workload management”
  - Extending/generalizing Panda into a Condor-enabled generic WM system deployed to OSG
- Basis of the proposal was support and interest from Condor (Miron Livny) in Panda and its pilot job approach
  - Plus CMS interest in pilot scheme also; CDF has a Condor-based pilot system (GlideCAF) and CMS-UCSD is bringing it into CMS
- Miron acknowledges that an overhead cost of grid systems like Condor is a substantial latency for job launch, and that a pilot system can usefully ‘short-circuit’ this for latency sensitive applications like interactive analysis
- Condor itself offers ‘multitasking pilot’ functionality through its VM system, targeted (among many others) for attention in the proposal
- That and many other avenues for Panda leveraging of Condor will be pursued if the proposal is accepted
  - Work will go on largely in the context of OSG’s planned program of ‘middleware extensions’ development

# Panda Program of Work

- Two main fronts:
  - Production - exercise Panda in production at increasing scales and more sites; debug/refine/harden based on performance & feedback
  - Analysis - finish delivering an effective analysis capability in Panda, then proceed as above
- The program, in outline:
  - Jan: resource-limited production throughput & <10% failure (done)
  - Jan: full end-user support: effective data access/discovery and job submission (data access/discovery done; still working on jobs)
  - Jan-Apr: evaluate how to scale Panda up for full analysis workloads
  - May+: Validate Panda as a hardened production system with no scaling limits for production work
  - Apr-Aug: implement Panda scaling for analysis
  - Fall: Integrate DQ2 support for personal datasets, data management and validate Panda for scalable analysis

# Conclusion

- Newly designed and implemented distributed production/ analysis system Panda for US ATLAS now in operation
- Designed for 'one stop shopping' distributed processing for US ATLAS, also interfaced to ATLAS production
- Based on internally managed queue/brokerage, pilot jobs and 'just in time' workload delivery
- Closely aligned with (also redesigned) ATLAS distributed data management
- Shows dramatic increase in throughput/scalability and decrease in operations workload
- Analysis systems in place but latencies too long; should be fixed in a couple weeks
- May spawn a more generic effort in collaboration with Condor, CMS, possibly also STAR

# More Information

- Panda
  - https://uimon.cern.ch/twiki/bin/view/Atlas/PanDA
- Panda monitor/browser
  - http://gridui01.usatlas.bnl.gov:28243/
- Distributed analysis with Panda
  - https://uimon.cern.ch/twiki/bin/view/Atlas/DAonPanda
- Access to Panda data (with DQ2)
  - https://uimon.cern.ch/twiki/bin/view/Atlas/UsingDQ2
- ATLAS DDM (DQ2)
  - https://uimon.cern.ch/twiki/bin/view/Atlas/DistributedDataManagement