Open Science Grid

Panda Overview and Release Schedule

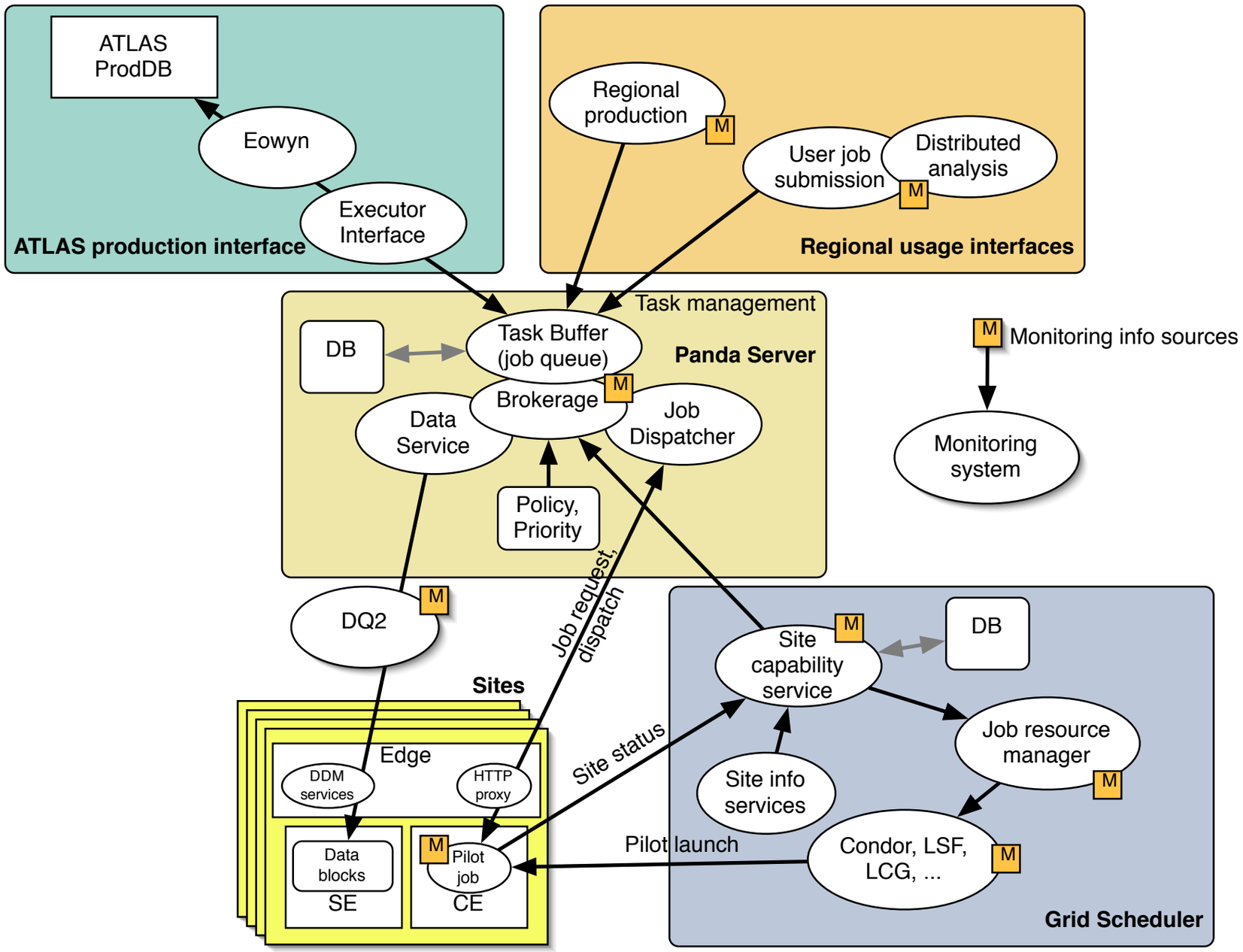
Torre Wenaus (BNL)

US ATLAS Tier 2 Meeting
UCSD
March 8, 2007

BROOKHAVEN NATIONAL LABORATORY

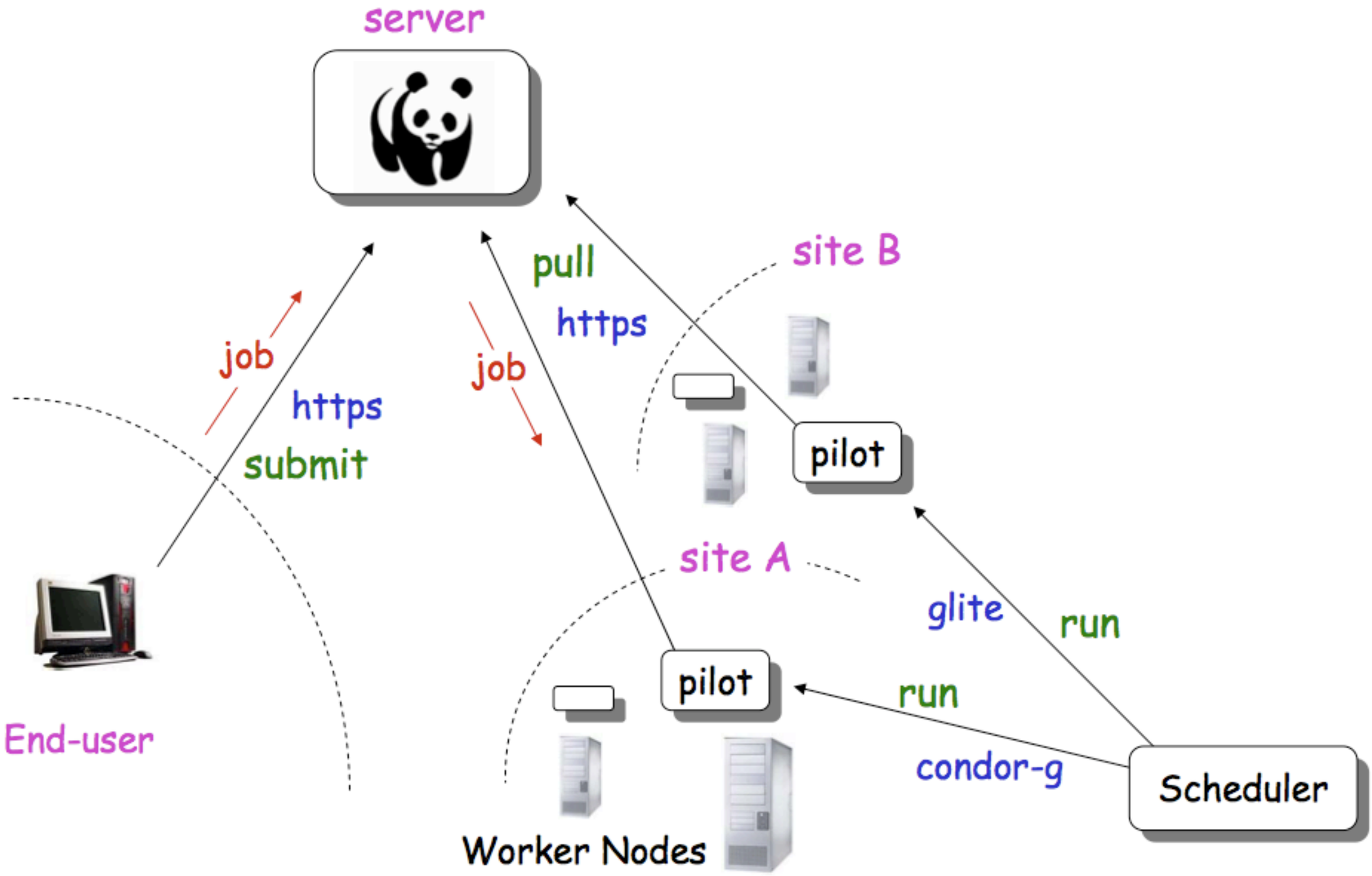
Panda Basics

Workload management system for Production ANd Distributed Analysis



- Launched 8/05 to achieve scalable data-driven workload management
  - Prototype 9/05
  - Production 12/05
- Tightly integrated with DDM
- Pilot-based 'CPU harvesting'
- Designed for analysis as well as production
- Designed for high automation, comprehensive monitoring, low ops manpower

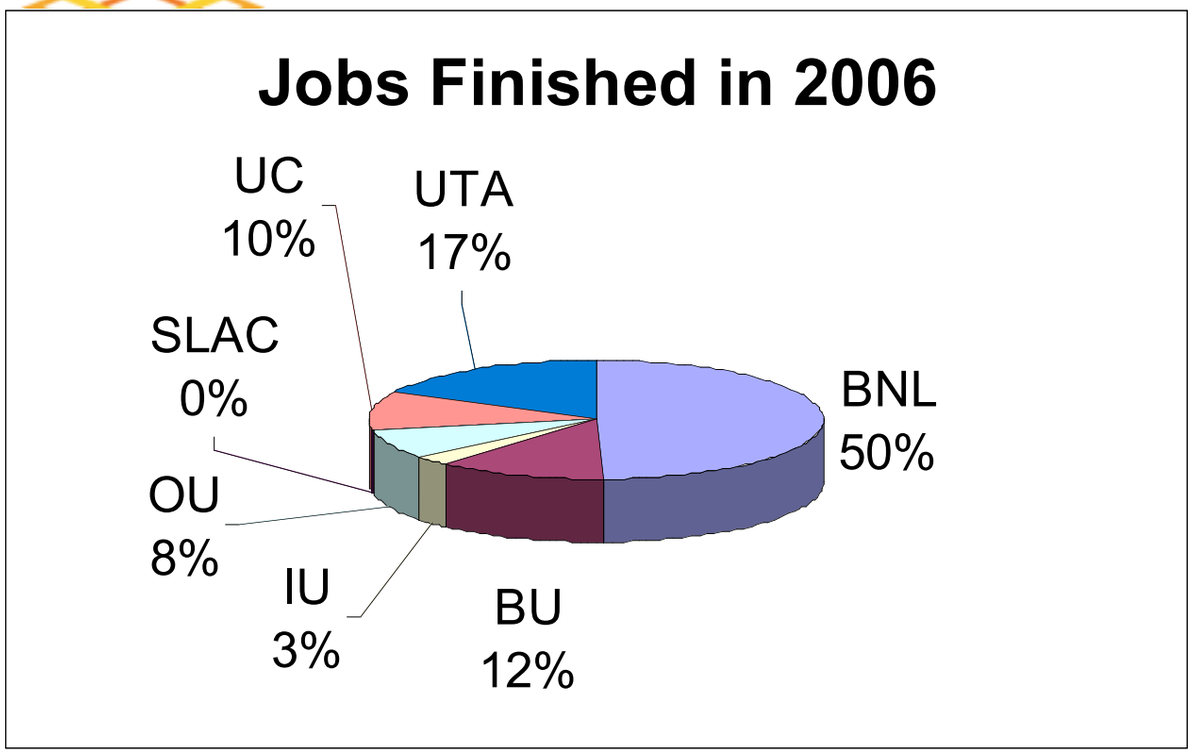
Panda Operation

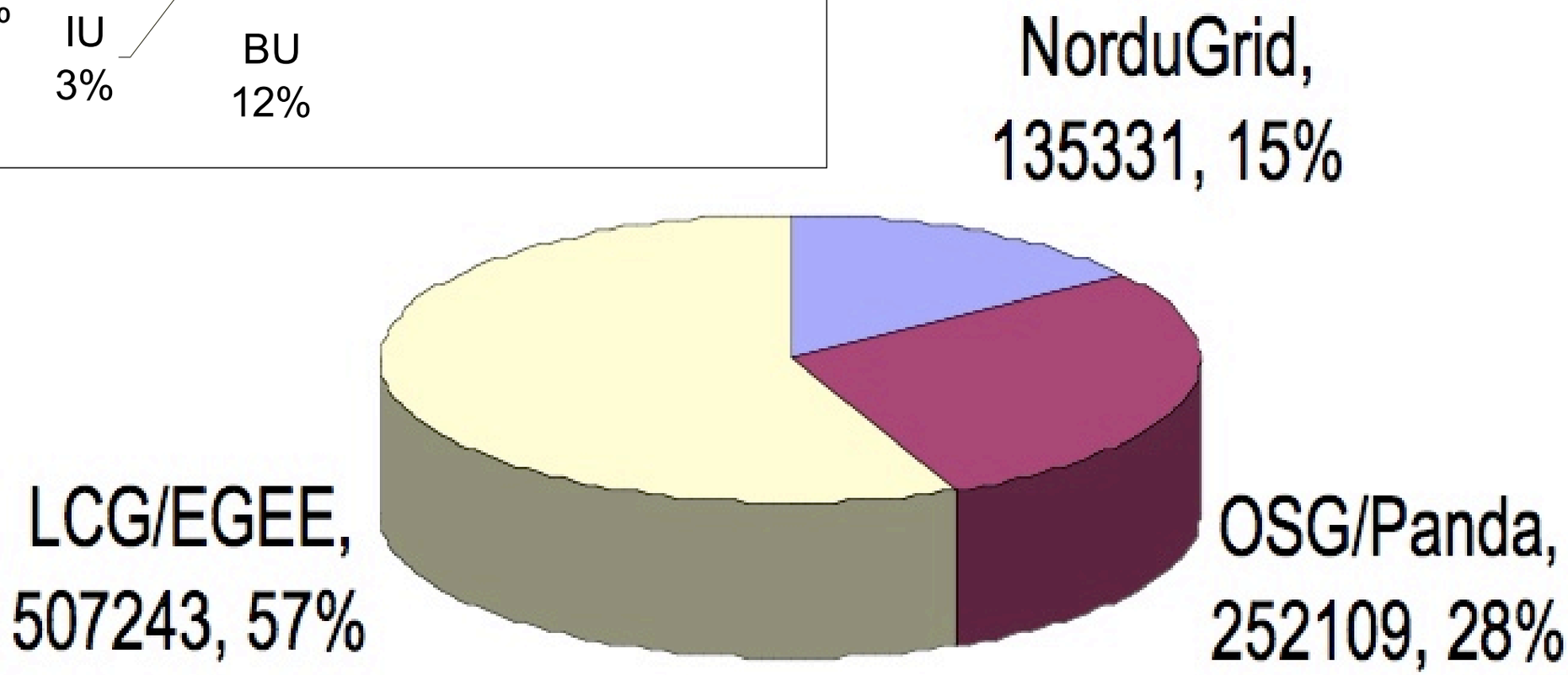


Panda Status

- All US ATLAS production, ~28% of overall ATLAS production with Panda (~50% more than official share)
  - Single shifter, spends <10% of time operating Panda
- Distributed analysis for US ATLAS, also used by Int'l ATLAS, about 50 users
- Recently extended to full OSG, LCG
  - 250 queues at 186 sites, ~200 queues successfully handling test Panda jobs
  - Working on deploying ATLAS production, analysis to these
- OSG Extensions effort on 'just-in-time' workload mgmt
  - ATLAS Panda, CMS glide-in factory, **Condor**
  - First non-ATLAS OSG Panda user starting prod: CHARMM

Completed ATLAS Production Jobs 2006

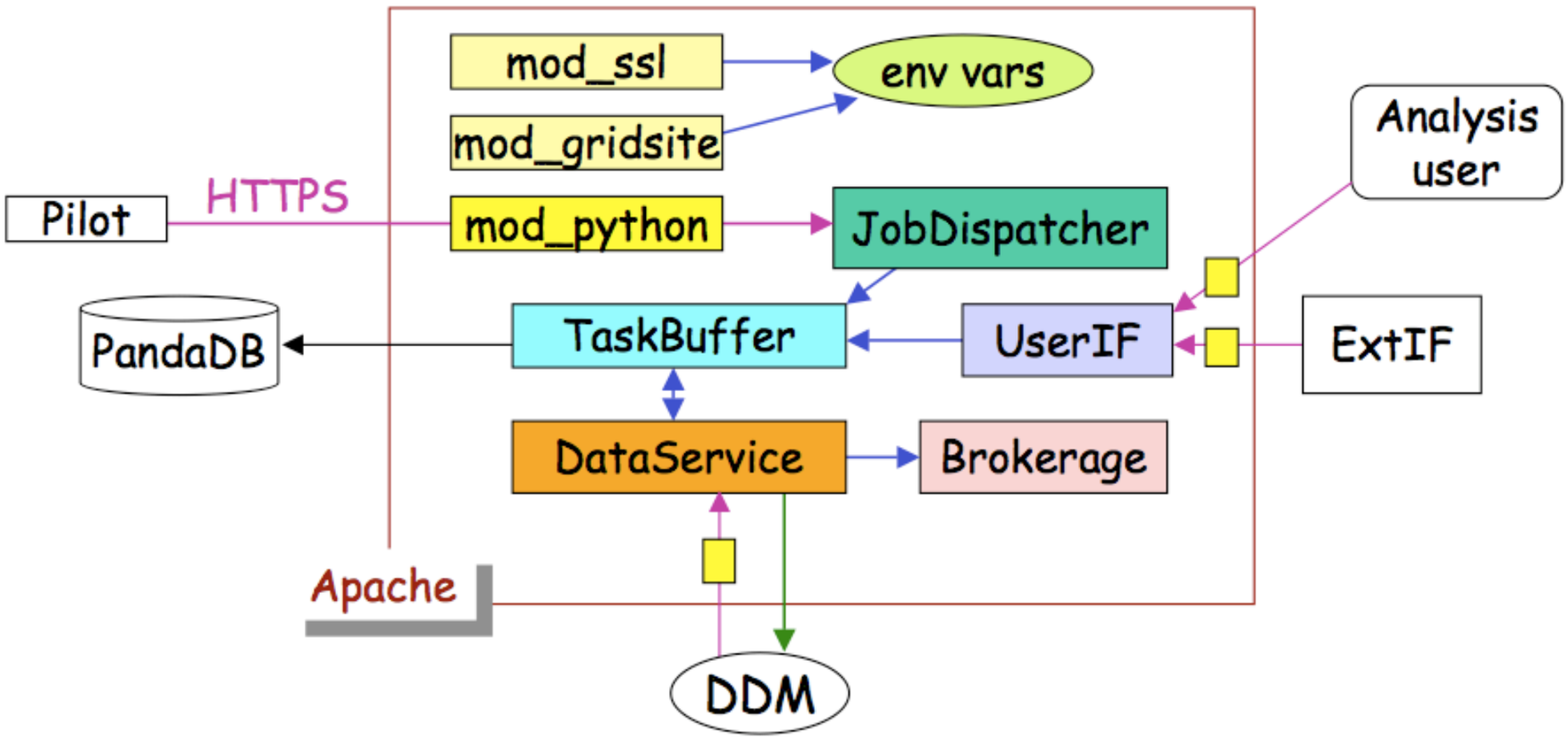




Panda production – 50% of the jobs done on Tier 1 facility at BNL
50% done at U.S. ATLAS Tier 2 sites

Panda Server

- LAMP
  - RHEL3 / SLC4
  - Apache 2.0.59
  - MySQL 5.0.27 - InnoDB
  - Python 2.4.4
- Multi-process (Apache child-processes) and multi-threading (Python threading)



Panda DB

- DB problems fully resolved by transition from cluster to InnoDB
  - Orders of magnitude improvement, consistent with tests conducted at UTA
  - No other change (hardware, memory, schema changes, query optimization) brought anything close to the improvement seen when we dropped the cluster
- So we're on solid ground again with the DB, with scaling headroom
  - and able to proceed with planned scaling measures in a 'confort zone' rather than 'crisis zone'
- Clear lesson: we need guru-level DBA knowledge of MySQL, and comprehensive and attentive monitoring

Panda Server

- Stable; recent and pending changes are small and incremental
  - Minor schema adjustments in schema version 5, to be deployed when we find a quiet time
  - Pilot handling adjustments for expanding scope
    - Multiple pilot types, multiple pilot streams for different apps, multiple VOs (CHARMM as well as ATLAS), new workflows (submitting Panda jobs from Panda jobs)
- Principal development activities in '07
  - **Scaling:** LAMP stack offers robust and well-proven scalability mechanisms which we expect to start activating this year
    - No immediate need but we want to keep ahead of the growth curve
    - Approach: system partitioning, most naturally by site or region (not via distributed deployment, except for the addition of a CERN instance, but via partitioning at the BNL/CERN instance)
  - **Security:** Through OSG, expect to leverage Condor and CMS work to secure job payloads, and to support WN-level identity switching where required
  - **Usage controls:** Ready to be activated for user/group level quotas, when needed

Scheduler/Pilot

- Panda scheduler manages submission of pilots to worker nodes
- Panda supports and uses multiple pilot submission approaches
  - CondorG scheduler, deployed for US ATLAS production at most US production sites
  - Local batch scheduler. PBS at UTA, Condor at BNL
    - Very efficient and robust, but cannot be centrally managed
    - Generic scheduler (new, part of 'TestPilot')
      - Supports local batch, CondorG, LCG, and (soon) pilot factory
        - Extends CondorG support OSG-wide and LCG-wide
        - LCG submission supported but irrelevant because of awful performance, CondorG is a full replacement
        - Adds partial support for central management of local batch
- Important recent improvements in pilots themselves, for robustness and recovery
  - Cleanup and recovery of previous job failures on WN

Panda and Condor Glide-ins

- Condor glide-in: use of GRAM or CondorC to 'glide in' a condor daemon to a remote site, acting as a transparent remote extension of a 'local' resource
  - startd glideins: remote resource becomes an additional worker node in a Condor pool, ie transparently distributed pool
  - schedd glideins: remote resource is a submit daemon, capable of creating jobs (pilots) at the remote site: a remotely-controlled pilot factory
- Use of glide-ins in Panda has been planned since Oct '05 and actively pursued since Sep '06 (when we gained manpower to work on it)
  - Initial target is a new capability for Condor: schedd glide-ins to support site-level pilot factory. Condor only supports startd glide-ins at present
- Note that Panda could use 'cronus' (Condor startd glide-in pool) trivially for pilot scheduling
  - 90% of Panda is at a higher level than this (the prodsys level, not the job submission 'executor' level)
- We are collaborating with CMS (Igor Sfligoi, FNAL) on startd glide-in; Igor has an infrastructure that is a more complete and generic solution than cronus
  - And Igor welcomes collaboration

Scheduler/Pilot

- Migration in progress to new Scheduler/Pilot code to
  - unify schedulers, pilots to one system for all Panda sites/usages
  - extend operation OSG-wide and LCG-wide
    - with improvements in scheduler/pilot control and monitoring to support this scale-up robustly and without taxing operations
      - this is where most of the work has gone to date
  - integrate with the OSG Extensions effort on pilot-based workload management
    - Condor integration (schedd and startd glideins as pilot launchers/hosts), CMS collaboration on same
    - Small meeting on this tomorrow, here
  - support non-ATLAS usages of Panda on the same infrastructure
    - first customer CHARMM now starting production; indications of more interest out of this meeting

Scheduler/Pilot: US Production

- Current production scheduler/pilot operates on US ATLAS OSG sites
  - With recent extensions for DDM support for opportunistic OSG usage
- New scheduler/pilot 'TestPilot' (needs a new name) runs across OSG and LCG, with new pilot-level monitoring layer and centralized DB management of queue selections, configs
  - Plan is to migrate all Panda usages to this
- Current production pilot, 'pilot2' from Paul/Xin, now works within TestPilot, 'wrapped' by TP's generic pilot
  - Objective pre-Munich: deploy to all current US production sites and validate with production jobs for migration
- Ideally, demonstrate TP-based Panda production on LCG pre-Munich also, but higher priority is analysis...

Scheduler/Pilot: US + LCG Analysis

- TP-based pathena deployment to LCG in progress
  - Initial targets are the T1s/clouds successfully importing AODs
    - Pilots flowing to 17 'good quality' queues at CERN, Lyon, FZK, SARA, PIC, RAL, + some T2s around these
- pathena in Panda works in two steps:
  - *User code load and build*. This now works with TestPilot at all the aforementioned LCG sites (and US OSG)
  - *Athena run*. Getting this working on LCG is in progress now, as highest priority
- Panda server supports needed matchmaking (run jobs must follow build jobs; send jobs only where needed release exists)
- Objective pre-Munich (next week): pathena operational at Lyon (minimal) or as many of the aforementioned sites as possible
- Initial deployment: user specifies the site where they want to run
  - Later, will support site assignment by Panda as for production
- Also, Panda being integrated as back end in GANGA
  - Dietrich visits BNL in April

Monitor

- Stable, but not always a good thing!
  - Monitor needs to catch up to some changes -- in particular, reorganized archive tables
- Performance much improved since DB became healthy
- Improvements/extensions in the works
  - Certificate enabled monitor. Will permit monitor taking action on the part of users: submit a job, register a DQ2 subscription
  - Extension of user-level monitor. Authentication, sessions, personalisation. Personal/group jobset & dataset management
  - More efficient DB interactions. Dynamic in-page addition of info ('ajax'), rather than all-or-nothing pages
  - Likely addition of new technology: 'Ruby on Rails', designed for exactly our system architecture, and offering these new functions 'for free'
  - Command line versions of user-level monitor functions

Timeline - Major near-term milestones

('Release schedule' sounds too precise & organized)

- Mar 21 (pre-Munich): ATLAS apps on TestPilot
  - pathena on LCG
  - validation of TestPilot+pilot2 for production in US ATLAS
  - demonstrate ATLAS production with Panda on LCG
- April: production ATLAS apps on TestPilot
  - migrate US production to TP
  - complete and broadly deploy ATLAS production on LCG
  - broaden pathena deployment on LCG based on interest
- May: executor 'shoot-out'
  - lcg-cg/cronus, Lexor, Panda comparison (to what purpose, not clear)
- June: official milestone for 'OSG works for US ATLAS'
  - extend ATLAS production usage to opportunistic OSG sites

Timeline - Other Milestones

(Approximate)

- Apr-May
  - startd-based pilot prototype, with glexec capability
- June-July
  - Panda server partitioning in the LAMP stack
  - Monitor extensions
  - Activate physics working group support, user/group quotas?
  - Production experiment-neutral Panda on OSG
- Aug-Sep
  - Secured job payload delivery to pilots
  - Full deployment of OSG WM Year1 deliverables (presently, glideins, pilot factory, glexec & secured pilot; more depending on manpower)
- Oct-Nov
  - Dedicated US-Panda DQ2 instance?

Conclusions

- Panda in good shape for US ATLAS
- Ready, we believe, to provide stable and robust production and analysis service for US ATLAS when datataking starts
  - Further development planned, but incremental
  - Depends of course on functional DDM
- Ambitious plans for extending the scope of Panda and integrating it further with Condor
  - ATLAS analysis: Offer Panda/pathena ATLAS wide to interested people; integration with Ganga
  - ATLAS production: LCG deployment and executor 'shoot-out'
  - ATLAS approach generally: show them our wares, in the hope that users will be allowed to vote with their feet, and in the belief that they'll walk in our direction!
  - OSG: now have our first non-ATLAS Panda user in production
  - Condor+CMS: OSG Extensions collaboration on WM