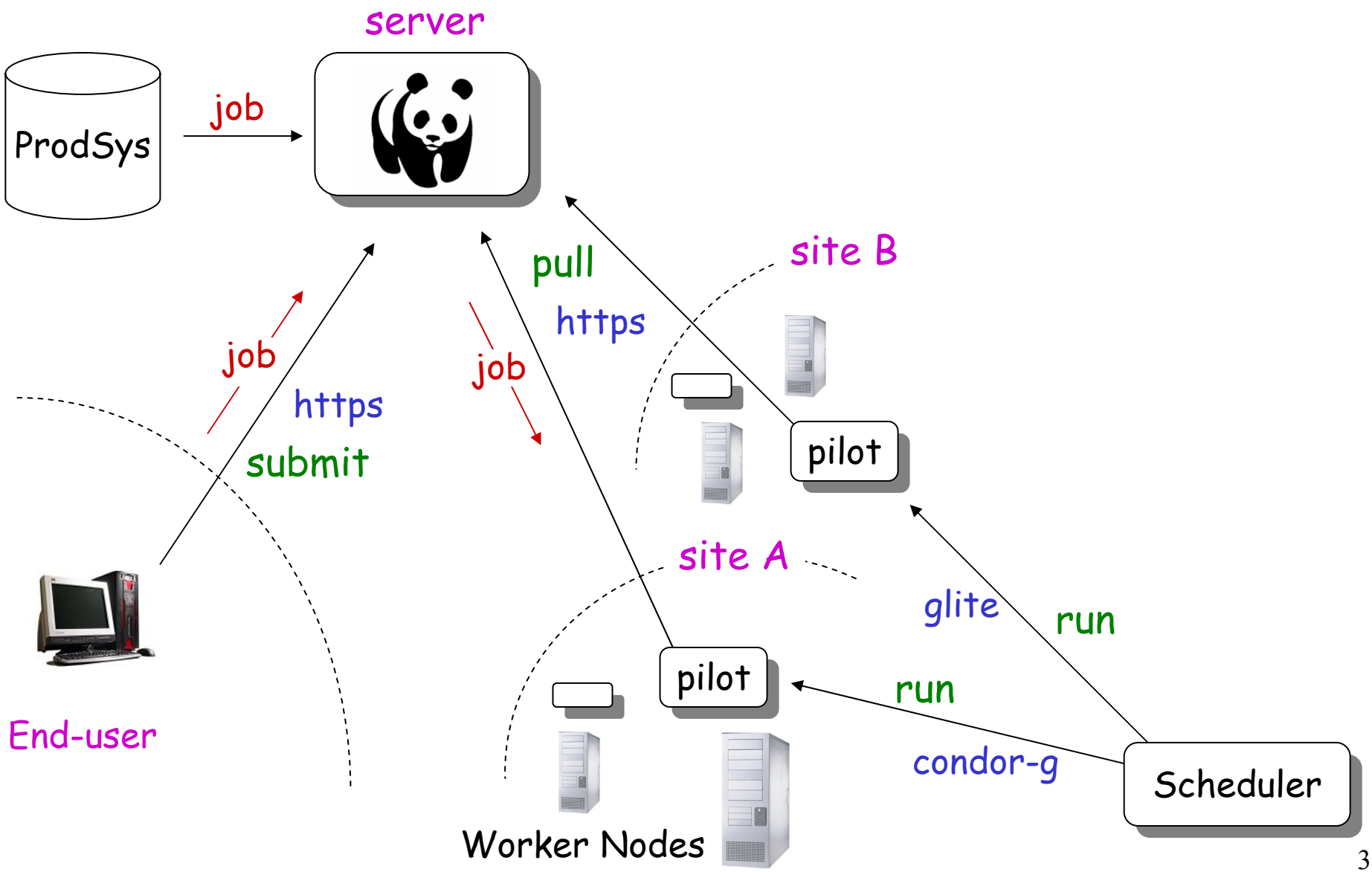
PANDA
Distributed Production and Distributed Analysis System for ATLAS

Tadashi Maeno (BNL)
on behalf of PANDA team

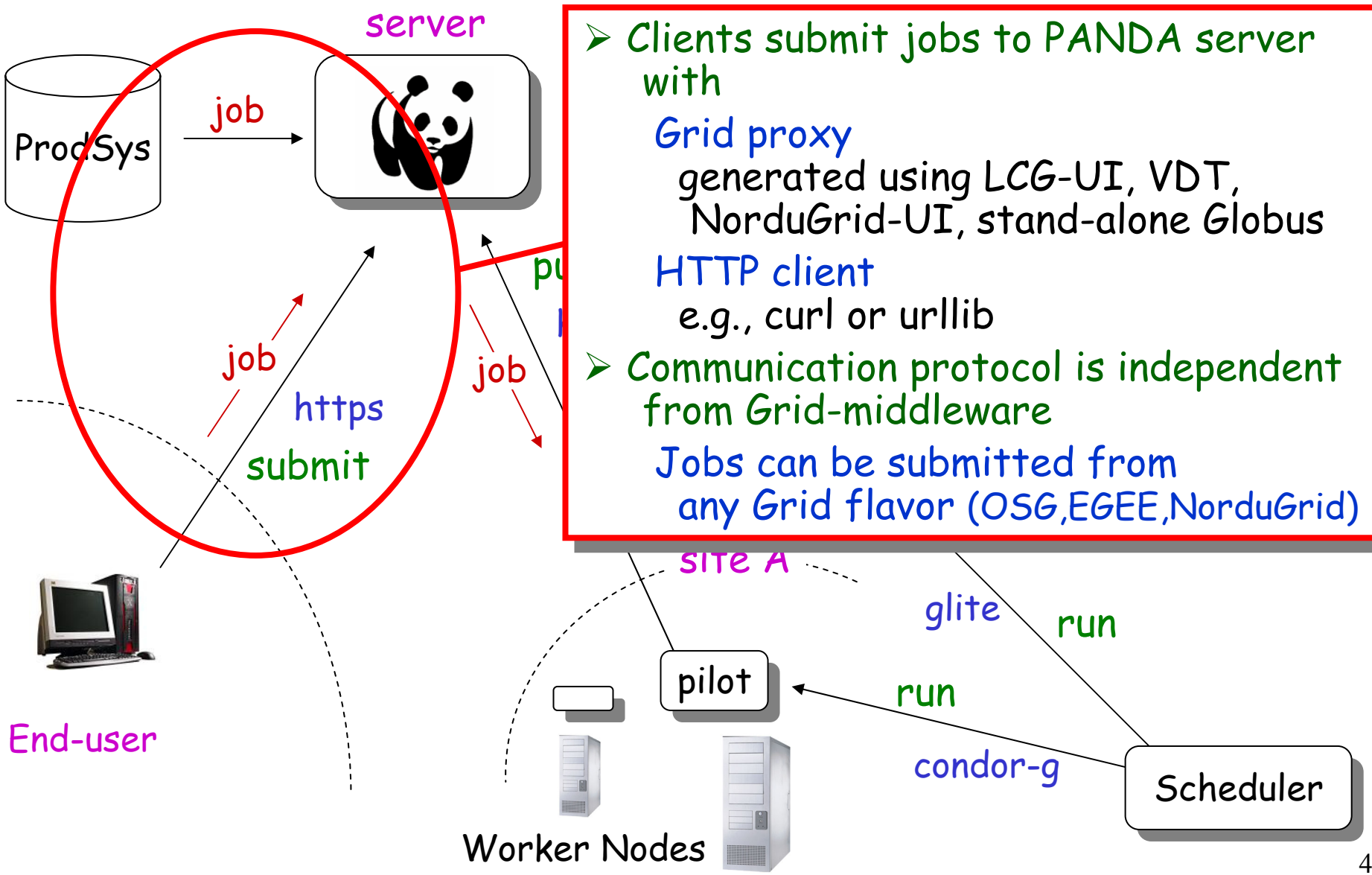
PANDA

- PANDA = Production ANd Distributed Analysis system
  - Designed for analysis as well as production
  - Project started Aug 2005, prototype Sep 2005, production Dec 2005
  - Works both with OSG and EGEE middleware
- A single task queue and pilots
  - Apache-based Central Server
  - Pilots retrieve jobs from the server as soon as CPU is available → low latency
- Highly automated, has an integrated monitoring system, and requires low operation manpower
- Integrated with ATLAS Distributed Data Management (DDM) system
- Not exclusively ATLAS: has its first OSG user CHARMM

PANDA System

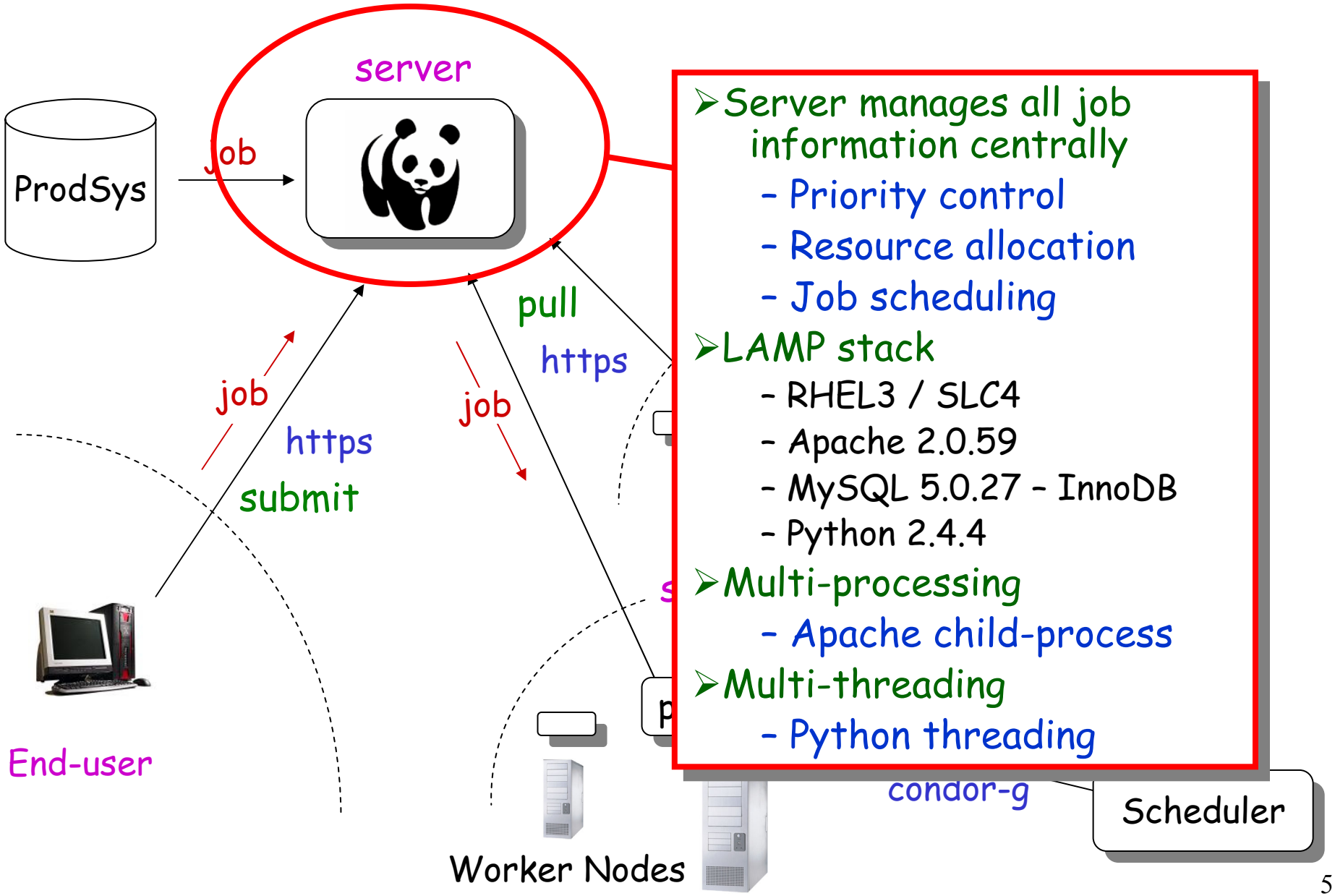


PANDA System



- Clients submit jobs to PANDA server with
  - Grid proxy
    - generated using LCG-UI, VDT, NorduGrid-UI, stand-alone Globus
  - HTTP client
    - e.g., curl or urllib
- Communication protocol is independent from Grid-middleware
  - Jobs can be submitted from any Grid flavor (OSG,EGEE,NorduGrid)

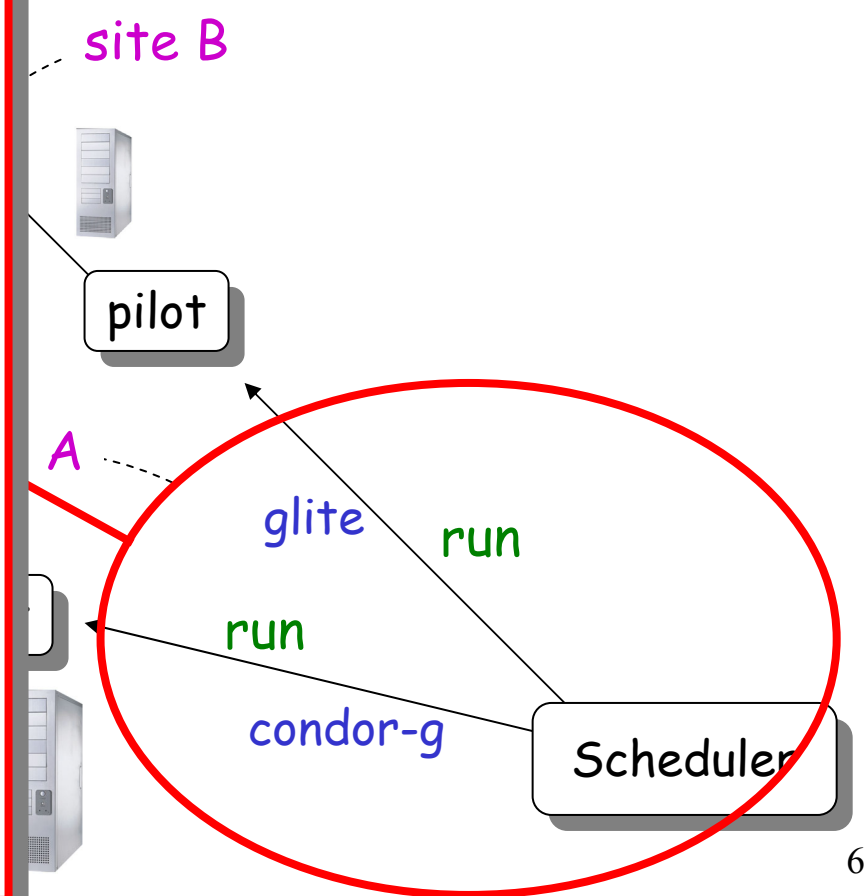
PANDA System



- Server manages all job information centrally
  - Priority control
  - Resource allocation
  - Job scheduling
- LAMP stack
  - RHEL3 / SLC4
  - Apache 2.0.59
  - MySQL 5.0.27 - InnoDB
  - Python 2.4.4
- Multi-processing
  - Apache child-process
- Multi-threading
  - Python threading

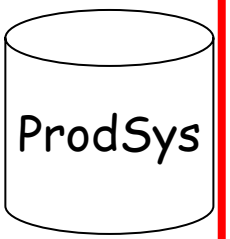
PANDA System

- Scheduler sends pilots to the batch system and Grid
- Three kinds of scheduler
  - –CondorG scheduler
    - For most US ATLAS OSG sites
  - –Local scheduler
    - BNL(condor) and UTA(PBS)
    - Very efficient and robust
  - –Generic scheduler
    - Supports also non-ATLAS OSG VOs and LCG
    - Being extended through OSG Extensions project to support Condor-based pilot factory
    - Move pilot submission from a global submission point to a site-local pilot factory, which itself is globally managed as a Condor glide-in



PANDA System

- Pilots are prescheduled to batch system and grid sites, and run jobs as soon as CPU becomes available
- Use resources efficiently
  - -Exit immediately if no job available
  - -Rate is regulated according to workload
- Multi-tasking
  - -Job execution
  - -Status reporting
  - -Zombie detection
  - -Error recovery
- Paul Nilsson will give a detailed talk on the pilot (# 170)



End-user

site A

glite

run

pilot

run

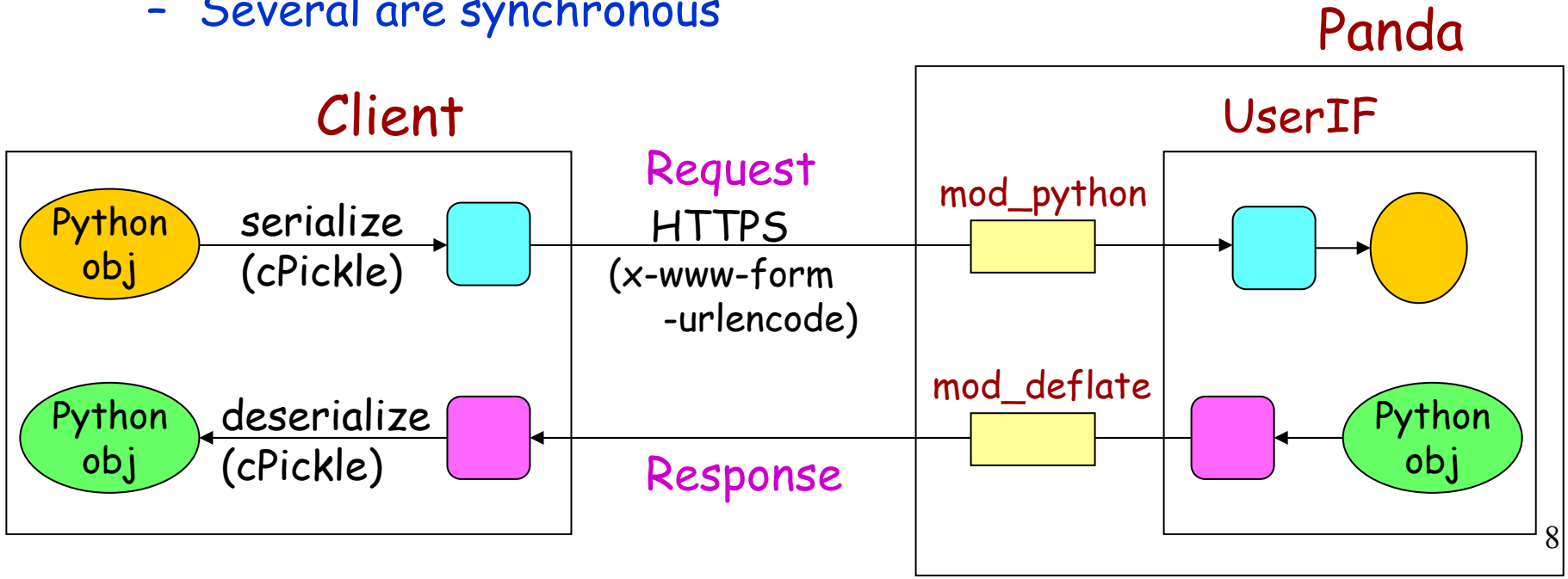
condor-g

Scheduler

Worker Nodes

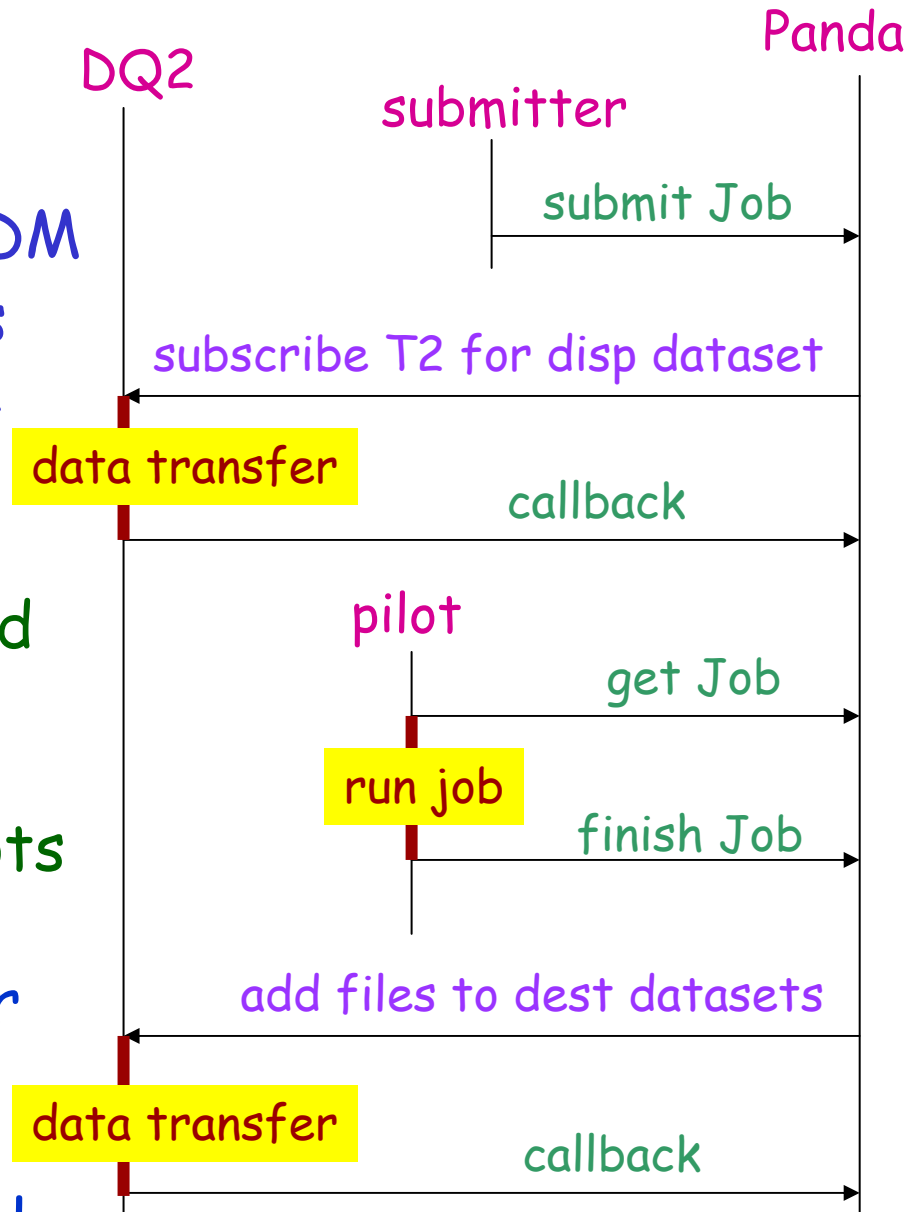
Client-Server Communication

- HTTP/S-based communication (curl+grid proxy+python)
- GSI authentication via mod_gridsite
- Most of communications are asynchronous
  - Panda server runs python threads as soon as it receives HTTP requests, and then sends responses back immediately. Threads do heavy procedures (e.g., DB access) in background → better throughput
  - Several are synchronous



Data Transfer

- Rely on ATLAS DDM
  - Panda sends requests to DDM
  - DDM moves files and sends notifications back to Panda
  - Panda and DDM work asynchronously
- Dispatch input files to T2s and aggregate output files to T1
- Jobs get 'activated' when all input files are copied, and pilots pick them up
  - Pilots don't have to wait for data arrival on WNs
  - Data-transfer and Job-execution can run in parallel



Production vs Analysis

- Run on same infrastructures
  - Same software, monitoring system and facilities
  - No duplicated manpower for maintenance
- Separate computing resources
  - Different queues → different CPU clusters
  - Production and analysis don't have to compete with each other
- Different policies for data transfers
  - Analysis jobs don't trigger data-transfer
    - Jobs go to sites which hold the input files
  - For production, input files are dispatched to T2s and output files are aggregated to T1 via DDM asynchronously
    - Controlled traffics

Operation/Service Model

- End-users are insulated from GRID
  - Communicate with the Panda (HTTP) server
  - Lower threshold especially for physicists
- Scheduler sends pilots to WNs using GRID middleware
  - Only the operator of the scheduler needs to have enough expertise on GRID
- Production and Analysis run on the same infrastructure
  - Production should suffer from the same problem as analysis
  - Once production team (one shift crew) fix the problem for official production, analysis get cured automatically
    → no additional manpower is needed for analysis

Current Status (1/2)

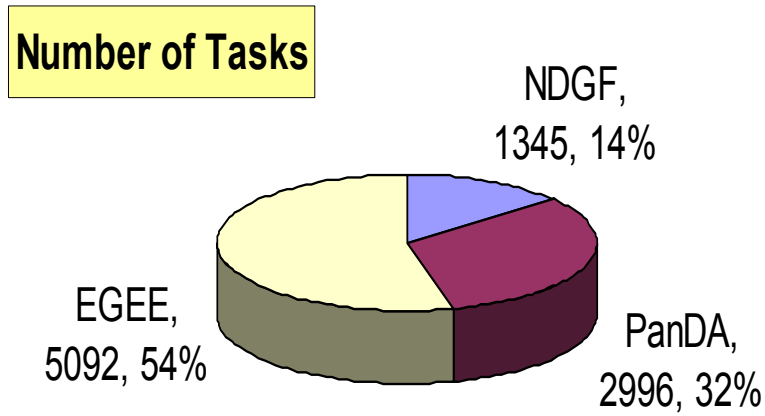
- ATLAS MC production
  - Computer System Commissioning (CSC) is on going
  - Massive MC samples produced for software validation, physics studies, calibration and commissioning
  - Many hundreds of different physics processes fully simulated with Geant 4
  - More than 10k CPU's participated in this exercise
- CSC with PANDA performing very well
  - All managed US production
    - ~2500 CPU's
    - ~35% of total ATLAS production
  - Canadian LCG sites have just come
    - TRIUMF, Toronto, WestGrid, Victoria …
    - > 25% of PANDA data has been produced in Canada for the past couple of weeks

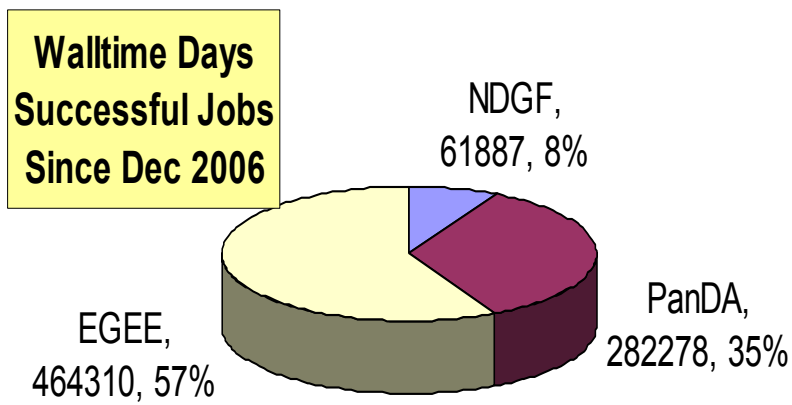
Current Status (2/2)

- Distributed Analysis effort
  - Has been in general use since June 2006
  - Popular with users (~100) and has been interested in ATLAS outside US which we're working to satisfy
    - e.g., LYON, and French Tier2s …
- Development is not complete and ended. But the changes should be incremental because steady operation is important for coming data-taking period

CSC Production Statistics

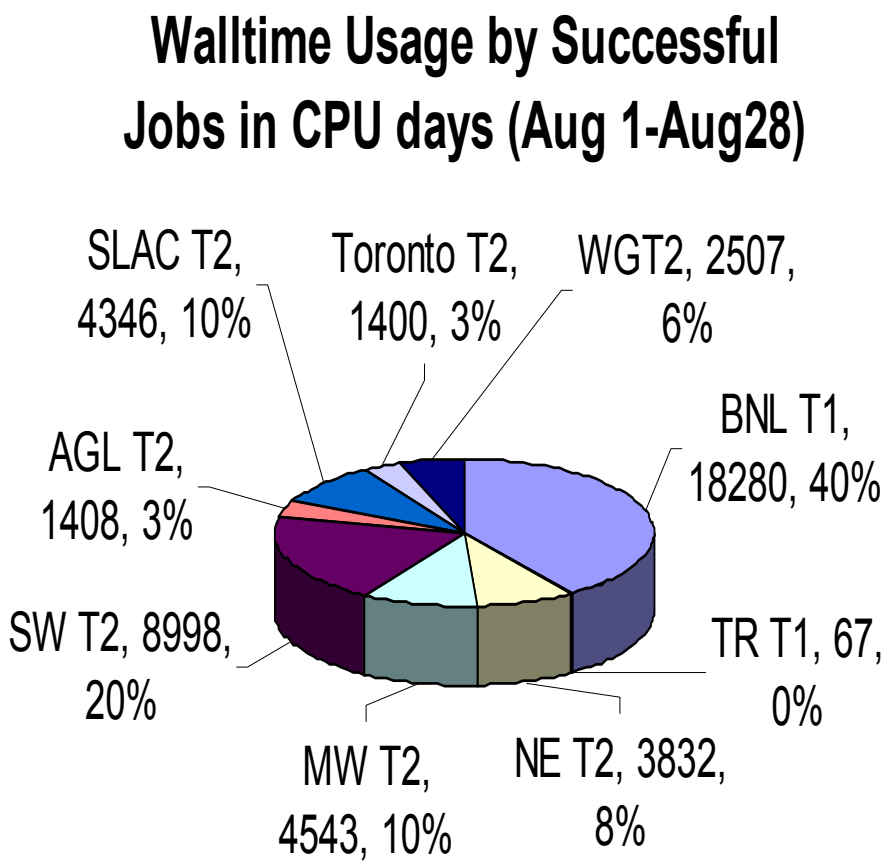
- Many hundreds of physics processes have been simulated
- Tens of thousands of tasks spanning two major releases
- Dozens of sub-releases (about every three weeks) have been tested and validated
- Thousands of 'bug reports' fed back to software and physics
- 50M+ events done from CSC12
- >300 TB of MC data on disk





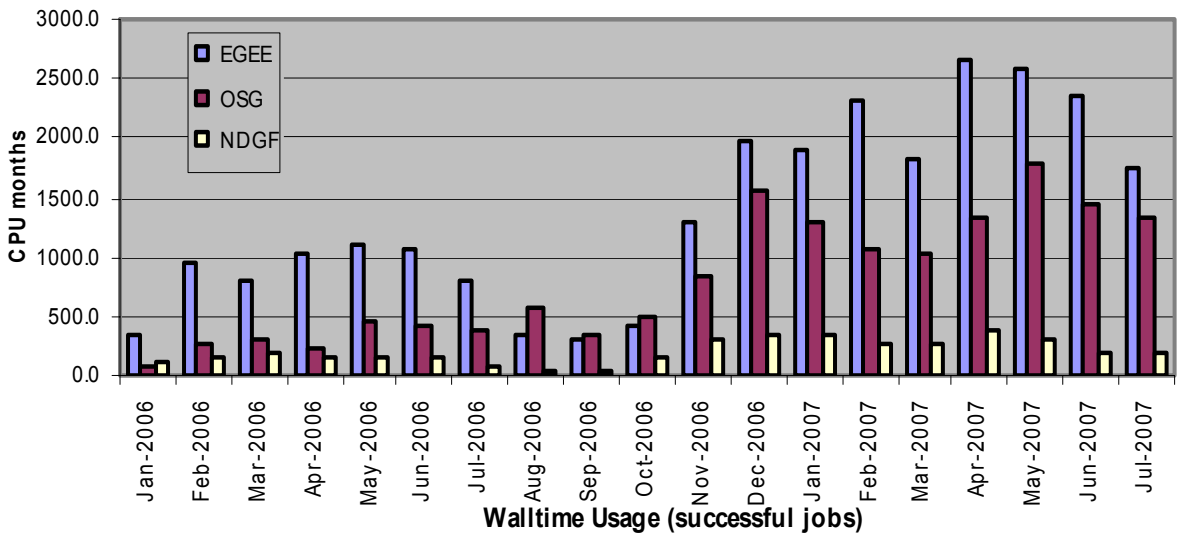
PanDA (US, Canada) Production Statistics

- PanDA has completed ~30M fully simulated physics events (simul+digit step), >30% of total central production
- Also successfully completed >15M single particle events
- Since November, all available CPU's occupied (ran out of jobs only for few days, plus few days of service outages)
- About 400 TB of original data stored at BNL T1 (includes data generated on other grids)
- Additional ~100 TB of replicas kept at U.S. ATLAS Tier 2 sites
- Canadian sites are now using PanDA for ATLAS production

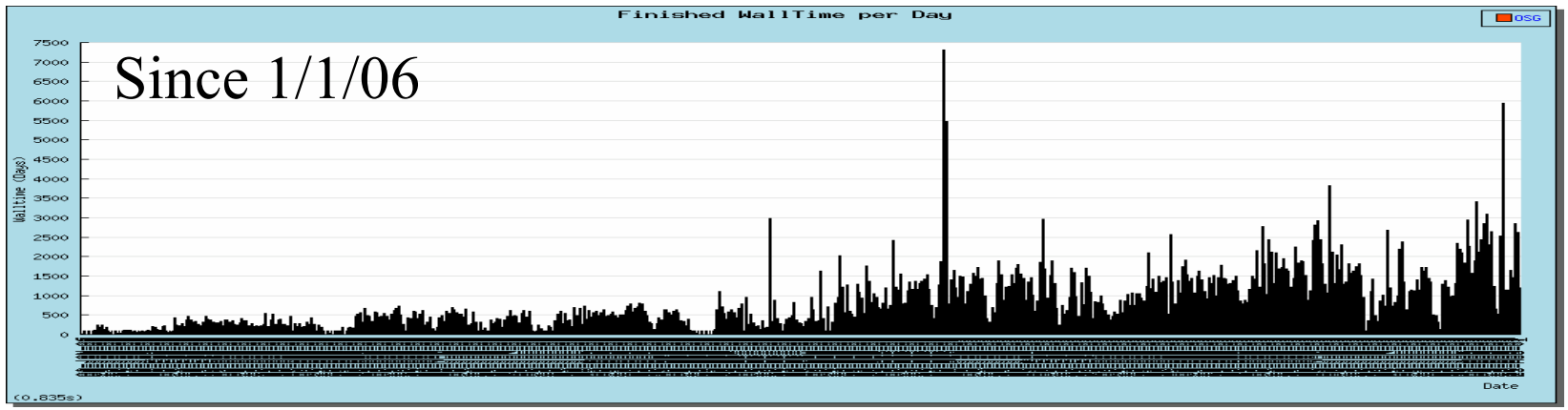


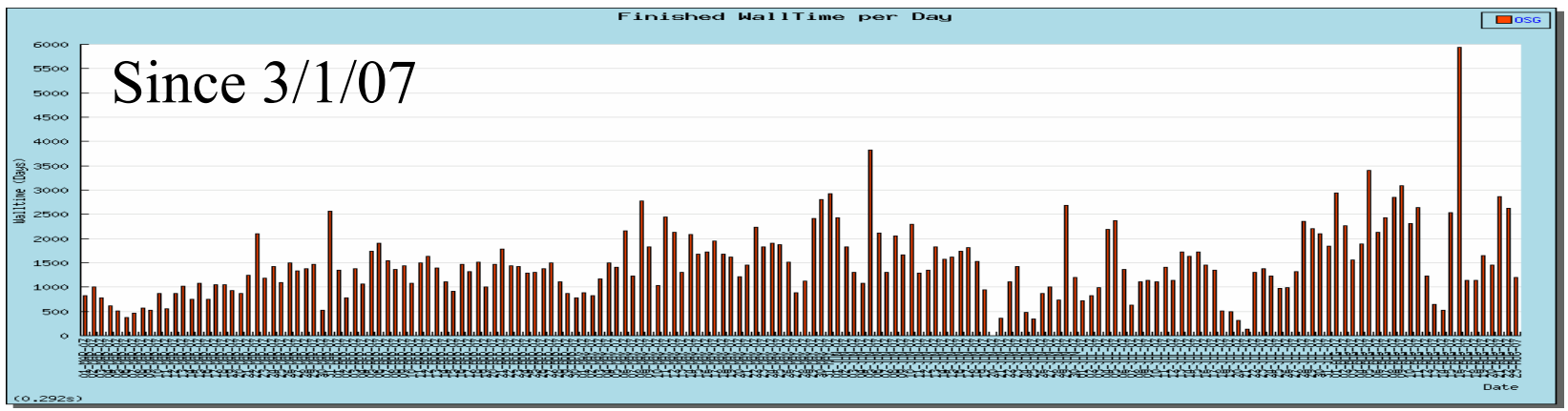
Resource Usage

- CPU and disk resources available to ATLAS rising steadily
- Production system efficiencies are steadily increasing
- But much more will be required for data taking
- Additional resources are coming online soon

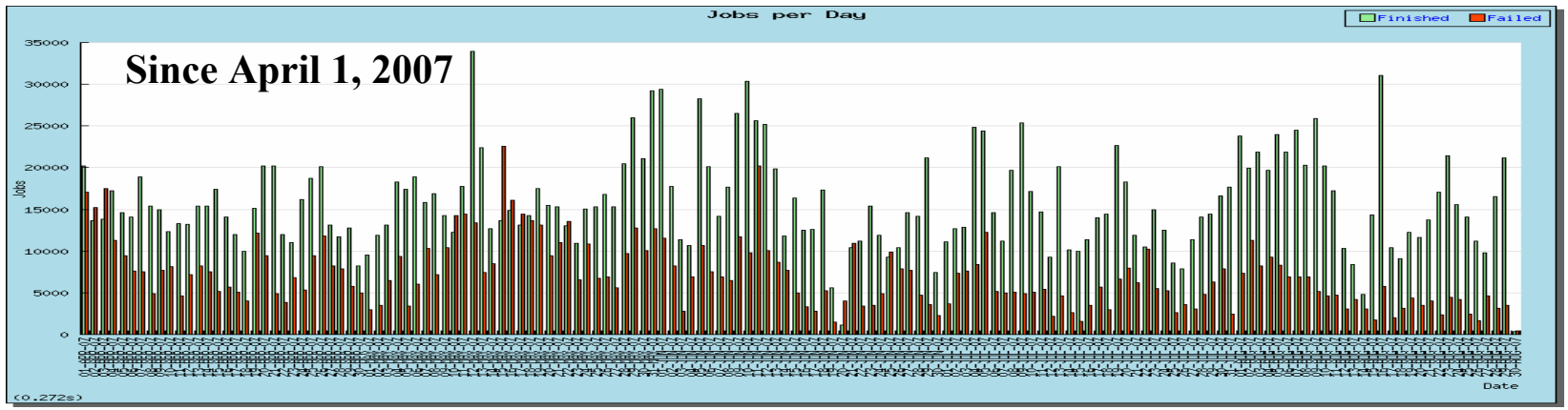


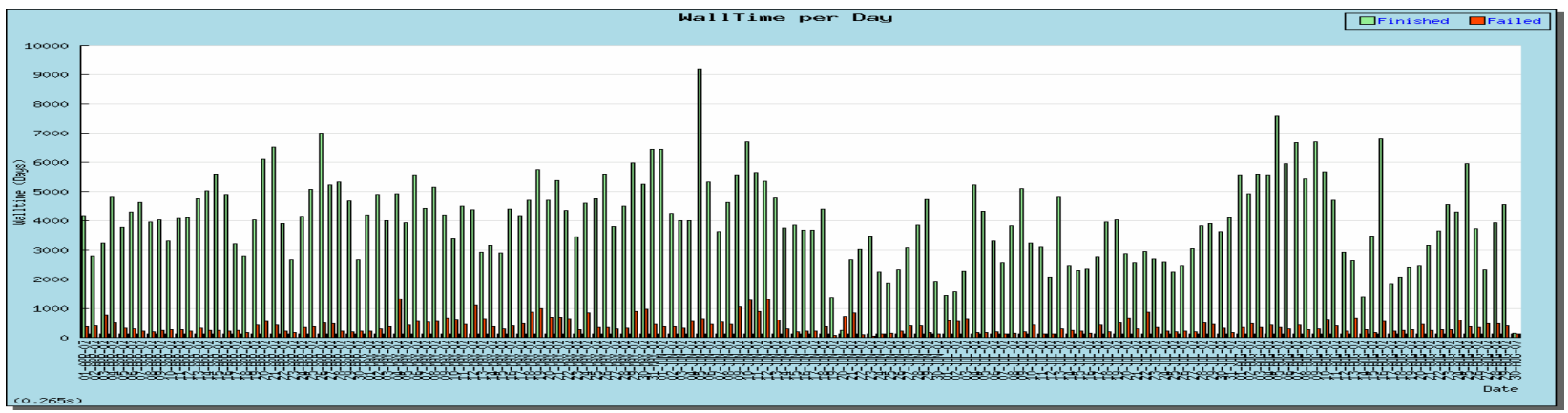
Panda Central Production



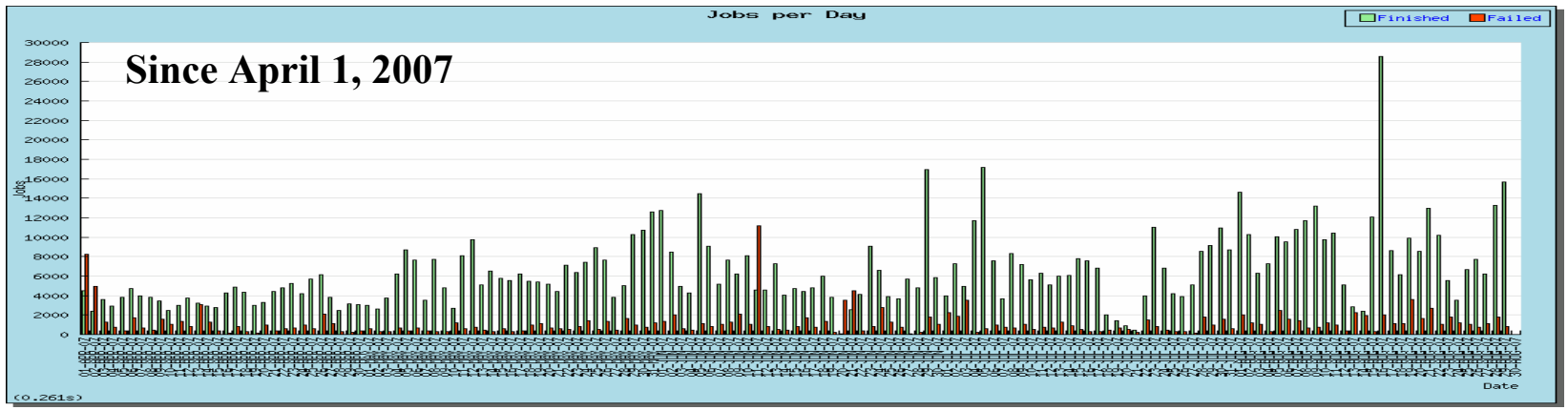


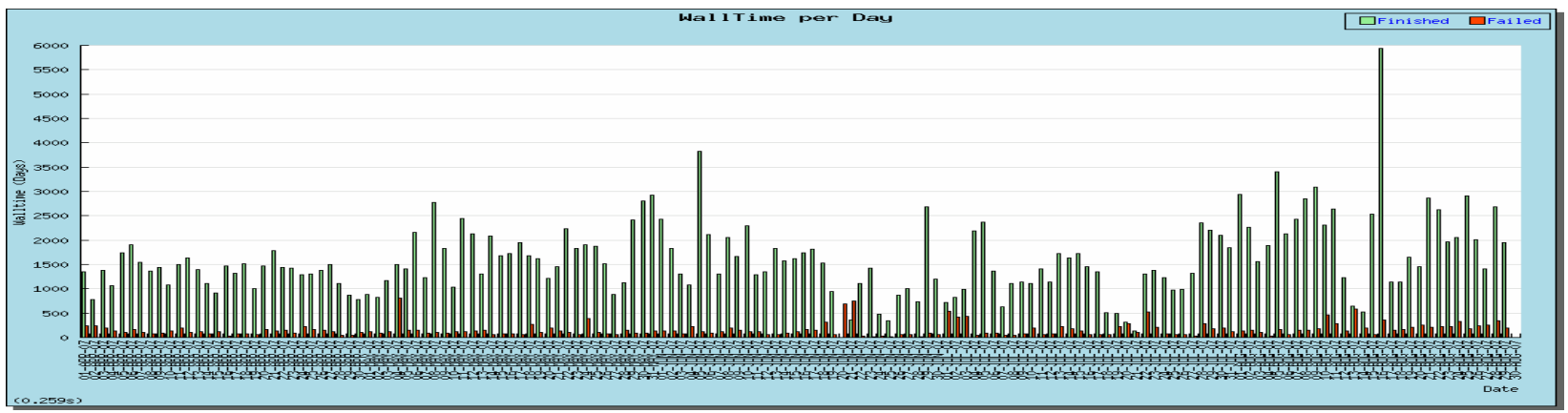
Job/Walltime Efficiencies for ATLAS





Job/Walltime Efficiencies for PANDA





Near-Term Plans

- Use generic scheduler/pilot system deployed on OSG and LCG to support ATLAS production and analysis across these grids
- Glide-ins, pilot factory and further Condor integration
  - Through OSG extensions project, collaborating with Condor and CMS
- Panda-native DDM
  - Self-scheduled data-transfers
- Additional resources coming soon

Conclusions

- PANDA performing very well both for production and analysis
  - High volume MC production
  - Huge computing resources available for individual analysis
- Ready to provide stable and robust service for coming data-taking of ATLAS experiment
  - Many new challenges to come but no big-bang migration
- Broadening deployment and supporting scale-up
  - LCG sites, OSG Tier3s, …
  - Condor extensions/integration to support scale-up