BROOKHAVEN NATIONAL LABORATORY

PanDA

TOMSK POLYTECHNIC UNIVERSITY

The BigPanDA monitoring system architecture

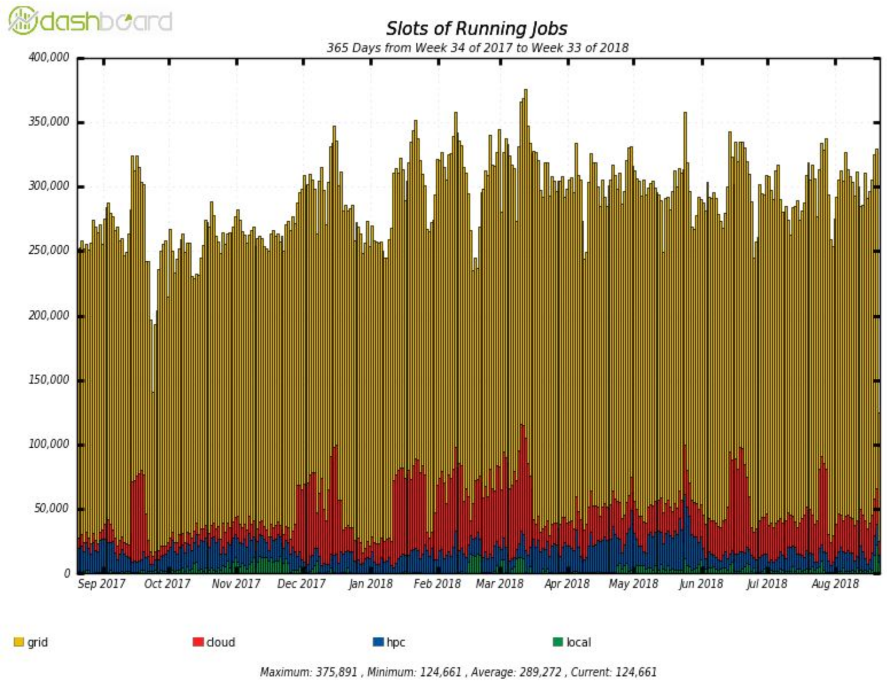
A. Alekseev, A. Klimentov, T. Korchuganova, S. Padolski, T. Wenaus
for the ATLAS Collaboration

The 8th International Conference
"Distributed Computing and Grid-technologies in Science and Education"
10-14.09.2018, Dubna, Russia

Outlines

- ➔ ATLAS PanDA Workload Management
- ➔ Requirements to BigPanDA monitoring system
- ➔ Architecture of BigPanDA monitor instance at ATLAS
- ➔ Data flow of BigPanDA monitoring system
- ➔ Adaptation for COMPASS experiment at CERN SPS
- ➔ Architecture of Django application
- ➔ Future plans and challenges
- ➔ Summary

ATLAS PanDA Workload Management



PanDA (Production and Distributed Analysis) Workload Management System is managing the workflow for all data processing on over 170 grid data centers, local, and opportunistic resources such as commercial and academic clouds and supercomputers.

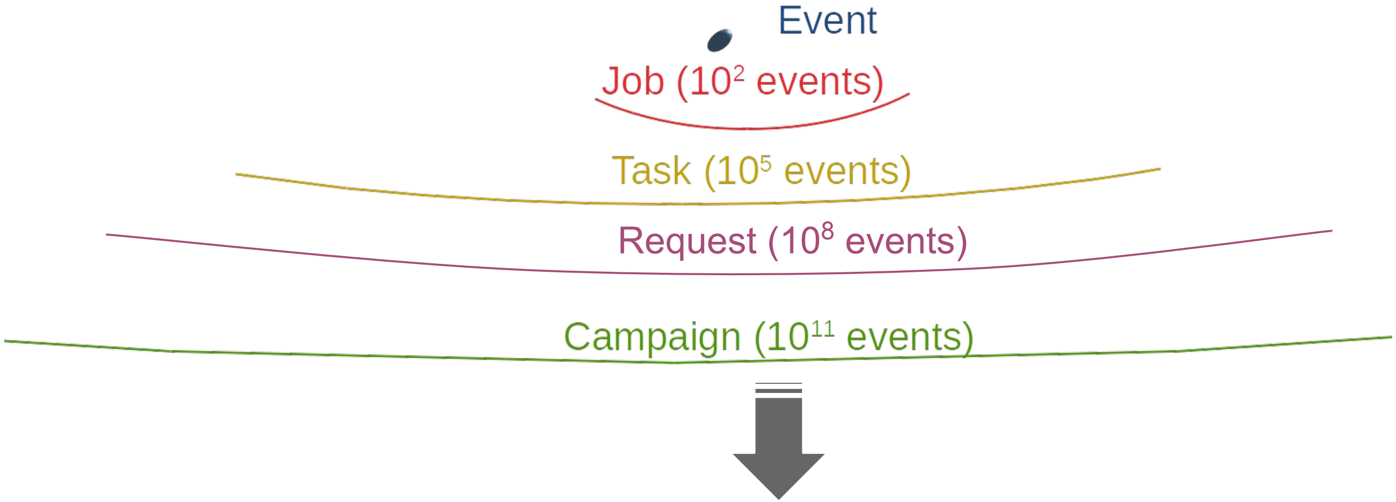
~250k simultaneous jobs

~2M jobs a day

~3k users

See talk “BigPanDA Experience on Titan for the ATLAS Experiment at the LHC” by A. Klimentov

PanDA processing scales at ATLAS



Event

Job (10^{2} events)

Task (10^{5} events)

Request (10^{8} events)

Campaign (10^{11} events)

Time to process:

~ 30s

~ 5 hours

~ 14 days

~ 6 weeks

~ 1 year

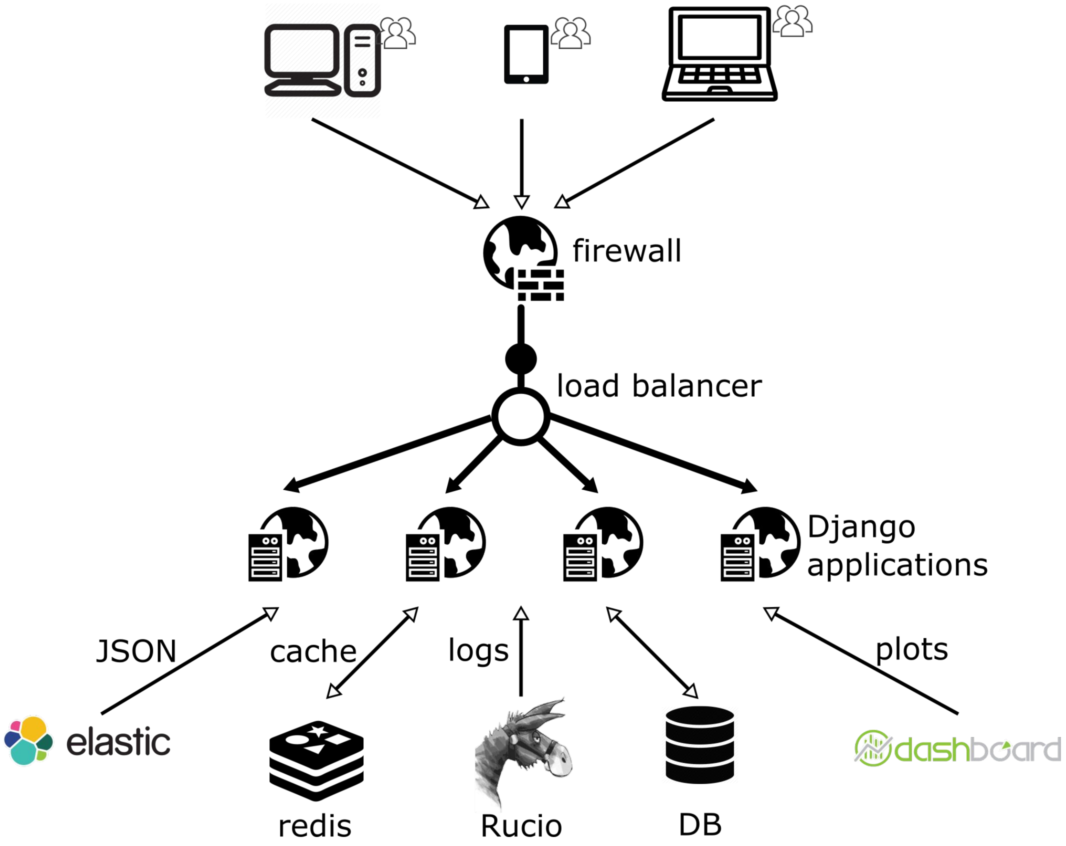
- ➔ Multilevel granularity of data to aggregate and display
- ➔ Providing the real-time and long-term-history (up to 2 year) of aggregated information
- ➔ Drilling-down from high level summaries to detailed diagnostics data

Goal of monitoring system is the rapid identification of failures and monitoring of progress of a distributed physics analysis and production.

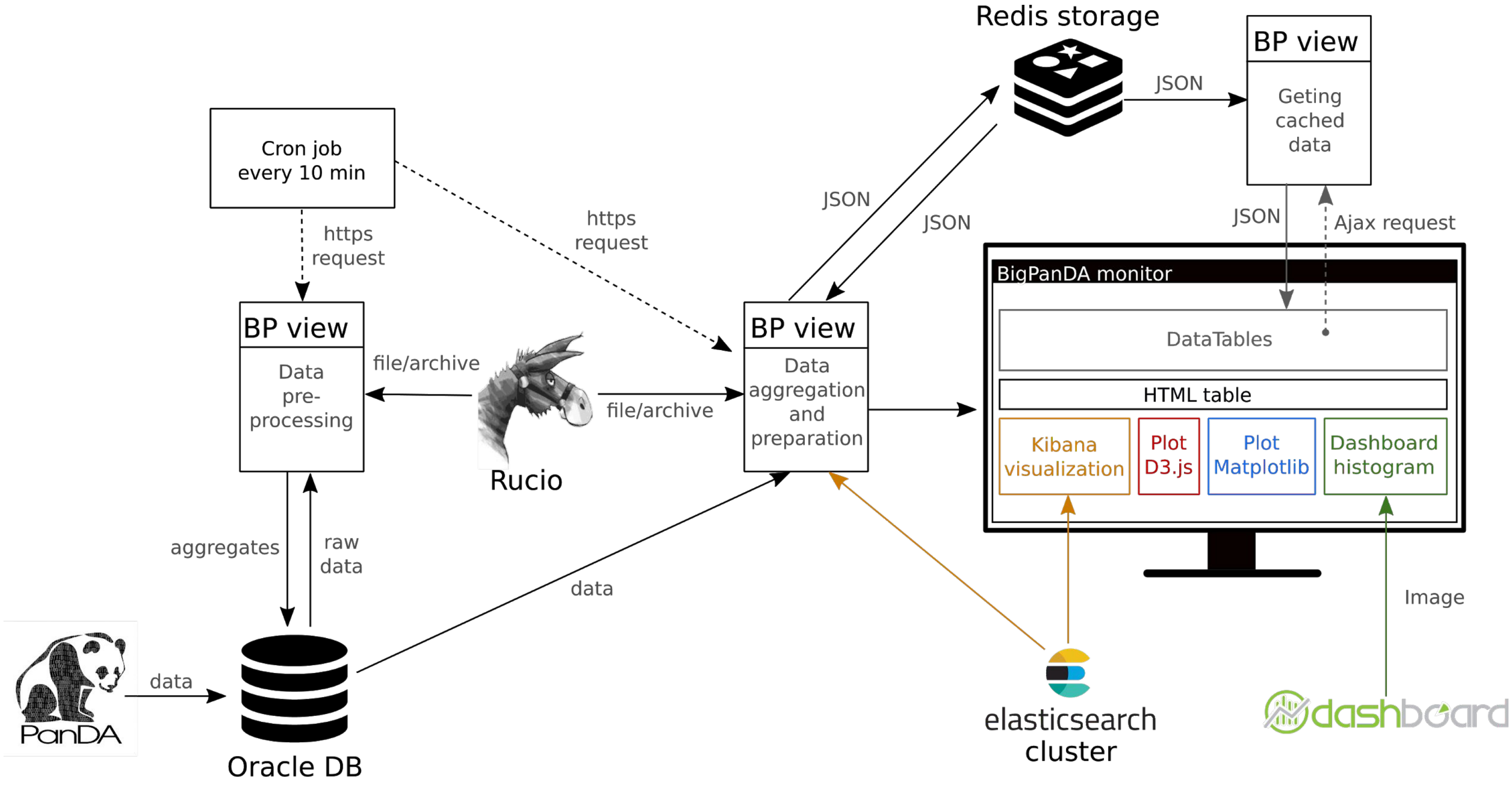
Requirements to BigPanDA monitor

- ➔ System state monitoring and troubleshooting by immediate access to logs
- ➔ 24/7 accessibility
- ➔ Possibility to aggregate data from various sources:
  - DB (Oracle, MySQL)
  - Indexed data from ElasticSearch and Kibana visualizations
  - log files in text or JSON format from Rucio
  - plots from Dashboard accounting service
- ➔ Ability to scale
- ➔ Acceptable response time for different usage scenarios
- ➔ Developing a common visualization mechanism to satisfy monitoring needs of different groups of users:
  - Operators
  - Physicists
  - Computing site administrators
  - Software developers
  - Managers
  - Coordinators

Architecture of BigPanDA monitor instance at CERN



Data flow of BigPanDA monitoring system



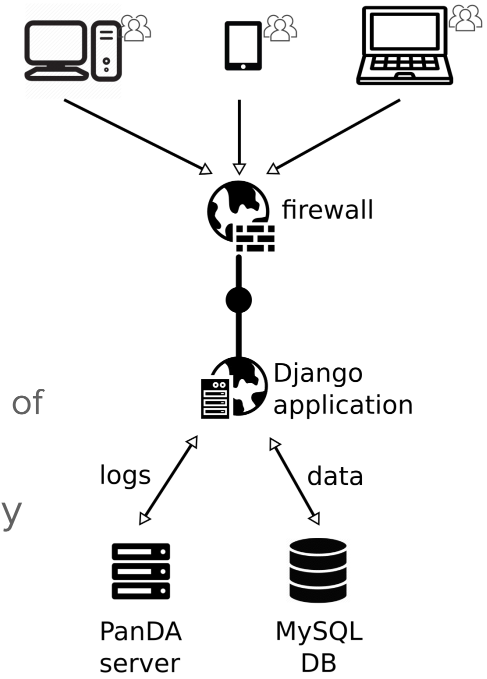
BigPanDA monitoring system features

- ➔ Using Model-Template-View approach provided by Django framework
- ➔ Various DB backends (Oracle or MySQL)
- ➔ Aggregating data from various sources (ElasticSearch, Rucio - Distributed Data Management system, Dashboard - WLCG accounting service)
- ➔ Data processing algorithms split between DB and application backends
- ➔ Responsive Web designing with Foundation
- ➔ Advanced plots generation on client side using d3.js library or matplotlib on server side
- ➔ Dynamic data delivery (Ajax)
- ➔ Proactive caching
- ➔ Using NGINX server as load balancer
- ➔ API for retrieving aggregated data in JSON format
- ➔ Self-monitoring alarm system based on ELK stack [see dedicated talk - “The BigPanDA self-monitoring alarm system for ATLAS” by A. Alekseev]
- ➔ Delivers advanced analytics against Big Data volumes using ES clusters

jQuery AJAX Foundation matplotlib django + python

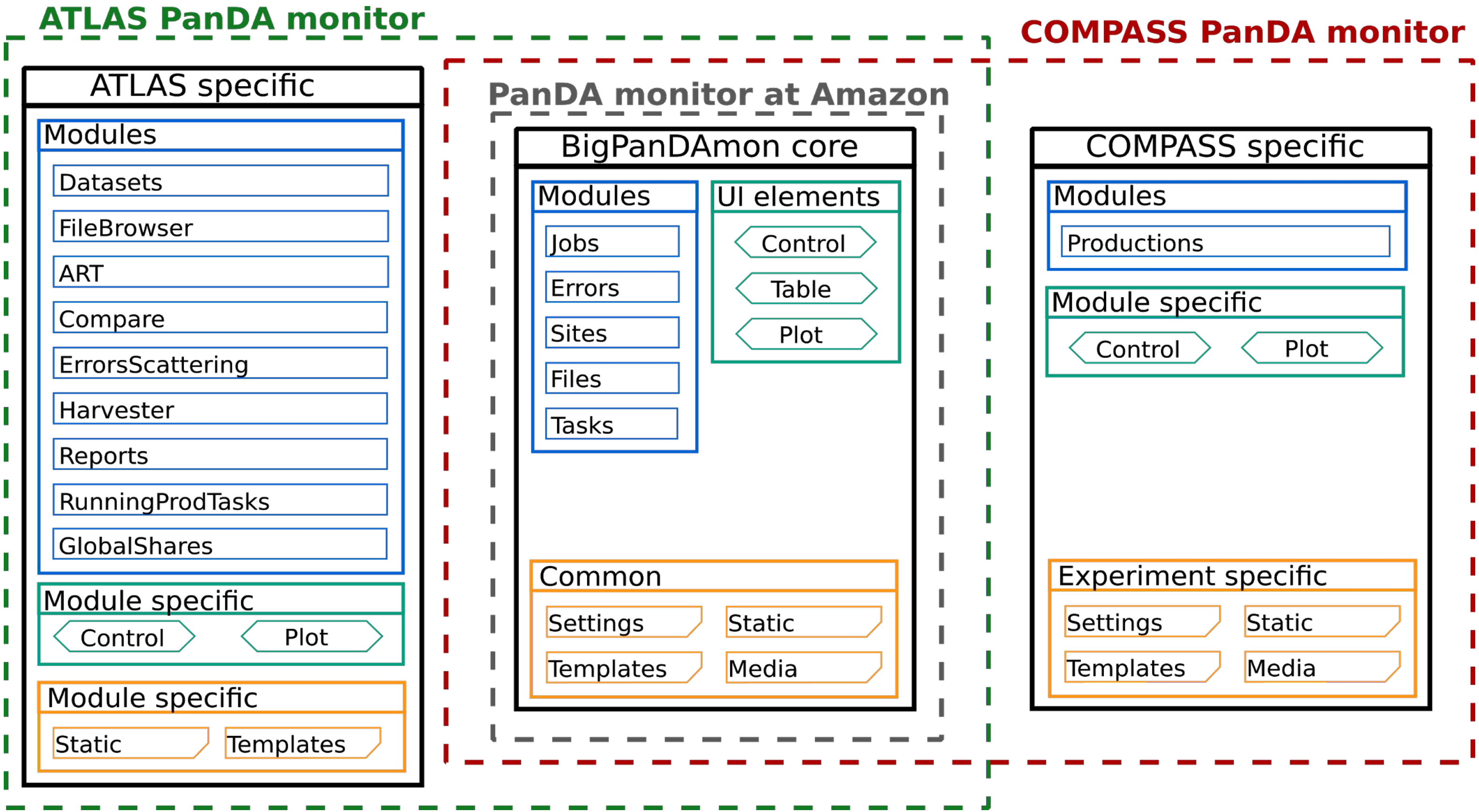
Adaptation for COMPASS experiment at CERN SPS

- ➔ BigPanda monitoring initially developed as experiment agnostic system
- ➔ Continuously evoluted in ATLAS
- ➔ Recent version was migrated to COMPASS
  - a new instance at JINR, Dubna
  - core views (jobs, files, dash, errors, sites) needed minor changes
  - tasks module was adapted to COMPASS production system
  - a completely new module was designed and implemented on the top of existed architecture
- ➔ It is in production since November 2017 serving ~300 requests/day



See talk of PanDA usage experience at COMPASS - “COMPASS Production System: Processing on HPC” by A. Petrosyan

Architecture of Django application for BigPanDA monitor



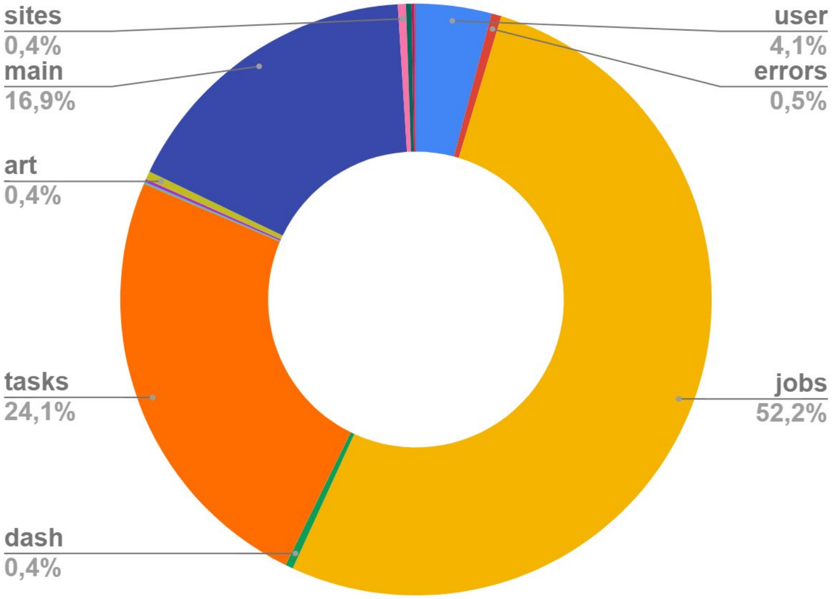
Future plans and challenges

- ➔ Performance
  - ◆ Switch to pre-produced aggregates in order to avoid latencies of on-demand requests
  - ◆ Development of algorithms for dispatching and aggregating the PanDA data stream at the one-pass processing
- ➔ Improvement of user experience
  - ◆ Enhance personalization features of BigPanDA monitoring
  - ◆ New modules, dashboards, views
- ➔ Capabilities
  - ◆ Extend functionality of ATLAS specific modules of BigPanda monitoring - Harvester, ART etc
  - ◆ Develop new functionality for better using BigPanDA as an intermediate data processing engine
  - ◆ Extension of analytics features
  - ◆ Integrate results of R&D projects into BigPanDA monitor, see related talks:
    - “The ATLAS Production System Predictive Analytics service: an approach for intelligent task analysis” by M. Titov
    - “Search for Anomalies in the Computational Jobs of the ATLAS Experiment with the Application of Visual Analytics” by M. Grigorieva

Summary

- ➔ BigPanDA Monitoring system is in production since the middle of 2014 and developing continuously;
- ➔ Developed architecture allows to serve **35k** requests daily including **20k** API requests in August 2018;
- ➔ Architecture of Django application allows to easily adapt the monitoring system beyond ATLAS (COMPASS, instance at Amazon for other experiments)
- ➔ It provides a comprehensive and coherent view of the tasks and jobs executed by the PanDA system, from high level summaries to detailed drill-down job diagnostics (**~23** views are available in ATLAS instance);

Usage statistics of the BigPanDA monitor views



ART - ATLAS Release Tester
Dash is a view which associates jobs with job queues on the basis of the region in which the processing site is located.