BigPanDA monitoring system evolution in the ATLAS experiment

Tatiana Korchuganova, Aleksandr Alekseev, and Alexei Klimentov on behalf of the ATLAS Computing Activity

May 08, 2023 to May 12, 2023 | 26th International Conference on Computing in High Energy & Nuclear Physics

Monitoring in ATLAS Distributed Computing

Place of BigPanDAmon in monitoring infrastructure

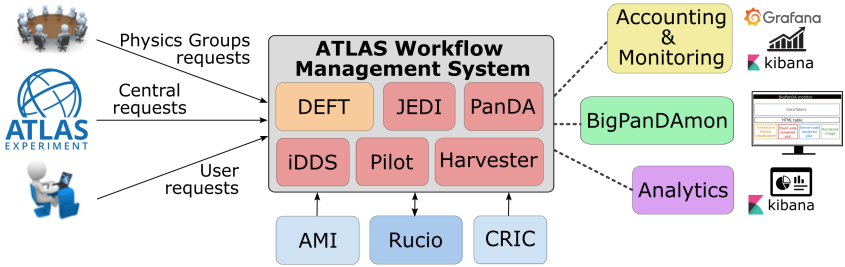


Figure 1 – Monitoring in ATLAS Distributed Computing

- DEFT: Database Engine For Tasks
- JEDI: Job Execution and Definition Interface
- PanDA: Production ANd Distributed Analysis System
- Harvester: resource-facing service between the PanDA and collection of pilots
- Pilot: the execution environment on a worker node
- iDDS: Intelligent Data Delivery System
- AMI: ATLAS Metadata Interface
- CRIC: Computing Resource Information Catalogue
- Rucio: Distributed Data Management System

- *Accounting & Monitoring**: saving snapshots of the system state regularly to observe trends in the time-series visualizations
- *BigPanDAmon*: monitoring of the current system state, and providing a wide range of views from the top-level summaries to a single computational job and its logs (the next generation system inherited from PanDA monitor in 2014)
- *Analytics*: finding correlations, trends, anomaly detection and building models for prediction of system behaviour in future

↓

The BigPanDAmon is essential for fast error troubleshooting and tracking progress of running production and analysis

*see more details in "Accounting and monitoring tools enhancement for Run 3 in the ATLAS distributed computing" poster

BigPanDAmon system overview

Architecture & technology stack

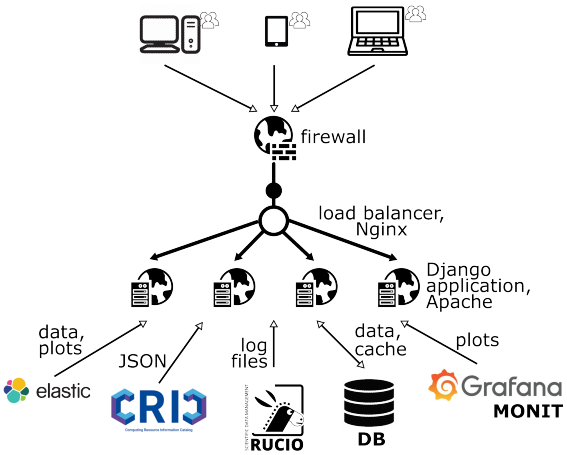


Figure 2 – BigPanDAmon architecture

- Using Model-Template-View approach provided by Django framework
- Various DB backends (**Oracle**, PostgreSQL, MySQL)
- Apache + Web Server Gateway Interface (WSGI)
- NGINX server as load balancer
- Ceph as shared file storage between nodes
- OAuth2 for SSO provided by CERN, Google, GitHub, and IndigoIAM
- Angular and Ajax for dynamic data delivery
- Responsive Web designing with ZURB Foundation
- DataTables plugin
- D3.js for advanced plots generation on client side
- Matplotlib for plots generation on server side
- ELK-stack for self-monitoring system

BigPanDAmon evolution

Project structure

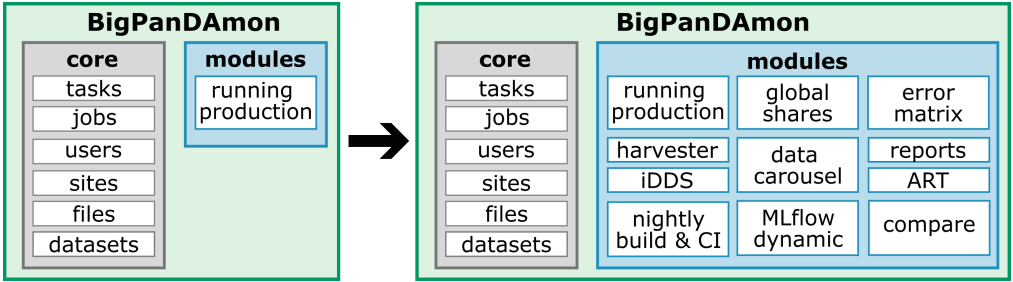


Figure 3 — BigPanDAmon structure evolution from 2017 to 2023

- Over the past few years of the PanDA WMS advancement[†‡] in the ATLAS Distributed Computing several new components were developed, such as Harvester, iDDS, Data Carousel, and Global Shares, etc.
- All new components are related closely to the key WMS objects and need to be monitored
- BigPanDAmon naturally grew into a platform where the relevant data from all PanDA WMS components and accompanying services are accumulated and displayed in the form of interactive charts and tables.

[†] Ref T.Maeno CHEP23 talk “Utilizing Distributed Heterogeneous Computing with PanDA in ATLAS”
[‡] Ref R.Walker talk at WLCG/HSF Pre-conference Workshop “Brokering to heterogeneous resources: ATLAS”

Harvester module

Harvester worker attributes summary

computingelement (257)	gate03.aglt2.org:9619 **(1145)** gridgk07.racf.bnl.gov:9619 **(1370)** calc2.t1.grid.kiae.ru:443 **(140)** gridgk02.racf.bnl.gov:9619 **(1338)** gridgk06.racf.bnl.gov:9619 **(1549)** gate01.aglt2.org:9619 **(1804)** ce02.tier2.hep.manchester.ac.uk **(507)** ce03.tier2.hep.manchester.ac.uk **(273)** lcgce1.shef... **more**
computingsite (185)	AGLT2 **(2780)** BNL **(8245)** RRC-KI-T1 **(350)** UKI-NORTHGRID-MAN-HEP **(1397)** UKI-NORTHGRID-SHEF-HEP **(129)** UKI-SOUTHGRID-BHAM-HEP **(57)** SWT2_LANCIUM **(114)** GoeGrid_LODISK **(37)** GOOGLE_EUW1 **(2780)** MWT2_K8S_UCORE **(86)** RIVR.UM **(108)** ... **more**
harvesterid (13)	CERN_central_A **(39220)** CERN_central_B **(27801)** test_aipanda185 **(114)** CERN_central_k8s **(5535)** CERN_central_ACTA **(22654)** CERN-dev **(183)** SuperMUC **(122)** cern_cloud **(908)** CERN_central_1 **(1546)** CERN_central_0 **(8546)** tacc-frontera **(9)** CERN_central_dask **(3)** NERSC_perlmutter_test **(1)**
jobtype (8)	managed **(55627)** user **(46111)** prod_test **(606)** ANY **(3061)** panda **(944)** DUMMY **(28)** rc_alrb **(142)** rc_test2 **(123)**
nativestatus (15)	running **(43002)** completed **(13555)** idle **(19712)** removed **(552)** **(13488)** starting **(4981)** done **(7152)** unknown **(64)** donefailed **(691)** donecancelled **(191)** held **(882)** tovalidate **(15)** transferring **(1)** sent **(2354)** finished **(2)**
resourcetype (4)	MCORE **(40253)** SCORE_HIMEM **(6749)** SCORE **(59632)** MCORE_HIMEM **(8)**
status (7)	running **(48416)** finished **(22670)** submitted **(33889)** cancelled **(1192)** missed **(117)** failed **(279)** idle **(79)**

Workers | Jobs | Worker Statistics | Dialog Messages

Number of workers 1000 Reload

Show 10 entries Search:

Workers submitted by harvesters

Instance	Worker ID	Jobs	Last update	Status	Batch ID	Computingsite	Computing element	Submit time	Start time
CERN_central_A	463520836	1	2023-04-04 02:10:49	running	15428467.34	AGLT2	gate03.aglt2.org:9619	2023-04-04 11:15:02	2023-04-04 11:50:23
CERN_central_A	463520835	1	2023-04-04 02:10:49	running	4356492.16	AGLT2	gate01.aglt2.org:9619	2023-04-04 11:15:02	2023-04-04 11:50:23
CERN_central_A	463520833	1	2023-04-04 12:59:26	cancelled	13859329.0	RRC-KI-T1	calc2.t1.grid.kiae.ru:443	2023-04-04 11:15:02	2023-04-04 11:50:02
CERN_central_A	463520832	1	2023-04-04 02:10:42	running	4356492.15	AGLT2	gate01.aglt2.org:9619	2023-04-04 11:15:02	2023-04-04 11:50:22
CERN_central_A	463520831	1	2023-04-04 02:00:37	running	15428467.33	AGLT2	gate01.aglt2.org:9619	2023-04-04 11:15:02	2023-04-04 11:20:11
CERN_central_A	463520830	1	2023-04-04 01:50:37	finished	15428467.32	AGLT2	gate03.aglt2.org:9619	2023-04-04 11:15:02	2023-04-04 11:50:22
CERN_central_A	463520829	1	2023-04-04 12:33:45	finished	4356492.14	AGLT2	gate01.aglt2.org:9619	2023-04-04 11:15:02	2023-04-04 11:50:22

Figure 4 – Harvester workers summary

Harvester

is a resource-facing service between the workflow management system (WFMS) and the collection of pilots for resource provisioning and workload shaping. Harvester worker = pilot or VM or MPI job.

The BigPanDAmon Harvester module consists of the following views:

- Harvester instances list
- Harvester workers summary (fig 4):
  - workers attribute summary allows to drill down to workers of interest
  - list of workers with links to batch logs
  - list of associated PanDA jobs
  - workers statistics
  - diagnostic messages to spot internal problems of a Harvester instance
- Harvester worker info page accumulates all information related to the selected worker
- Harvester slots page represents the number of slots for special resources (HPCs) working in push mode

GlobalShares module

Global Shares

enable central control of resources by continuously determining fraction of total resources available to ATLAS to allocate to a given activity. Nested 3-level structure.

Special dashboard was developed which consists of the following blocks:

- an overview with plots (fig. 5)
- a table with list of global shares representing target & actual values linked to the corresponding PanDA job list
- a set of tabs with tables showing global shares distribution across computing sites, types of computing resources (GRID, HPC, Cloud etc), resource types (single/multi core, ordinary/high memory), and PanDA job statuses.

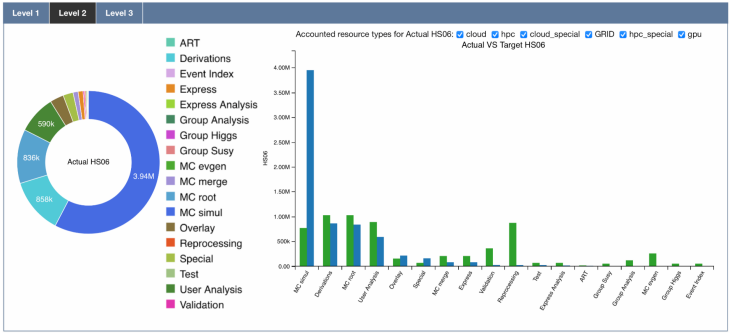


Figure 5 – GlobalShares dashboard

iDDS module

iDDS (intelligent Data Delivery Service)

aims to intelligently orchestrate workflows and data management systems, decoupling data pre-processing, delivery, and primary processing in large scale workflows (including workflows defined via DAG (Directed Acyclic Graph)).

Requests:

Show 10 entries Search:

request id	username	workflow status	graph	workflow name	created on (UTC)	total tasks	tasks
459289	User 1	SubFinished	plot	mc16_valid.900248.PG_singlepion_flatPt2to50.simul.HITS.e8312_s3238_tid26378578_00	2023-03-21 08:42:16	6	Finished(4) Failed(2)
459287	User 2	Failed	plot	pseudo_input.2023_03_21_08_31_15_068502719	2023-03-21 08:31:16	5	Failed(5)
457051	User 2	Finished	plot	pseudo_input.2023_03_07_13_41_14_791409569	2023-03-07 13:41:15	1	Finished(1)
456849	User 3	Failed	plot	pseudo_input.2023_03_06_21_06_55_862884428	2023-03-06 21:06:56	5	Failed(5)
456473	User 3	Finished	plot	pseudo_input.2023_03_06_14_55_55_63443933	2023-03-06 14:55:56	1	Finished(1)

Showing 1 to 5 of 5 entries

Figure 6 – Workflows progress view

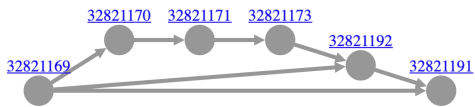


Figure 7 – A graph plot of one of DAG Workflows

The iDDS module in the BigPanDAmon contains 3 views:

- workflows progress view (fig. 6)
- a graph plot for DAG workflows (fig. 7)
- a general iDDS view that allows to drill down through iDDS object hierarchy: requests → transform → collection (input/output/log) → list of files for a collection.

Data Carousel module

Data Carousel[§]

orchestrates data processing between workload management, data management, and storage services with the bulk data resident on offline storage. The processing is executed by staging and promptly processing a sliding window of inputs onto faster buffer storage, such that only a small percentage of input data are available at any one time.

- dedicated dashboard has been developed for monitoring staging activities
  - various visualizations of staging datasets/files/their volume
  - a table with staging requests and associated tasks that have input on tape
- a notification mechanism of stalled staging requests was put in place
  - it regularly searches staging requests that have not progressed for 10 days
  - it sends an email report to the list of relevant experts

Figure 8 — Data Carousel dashboard

[§]Ref. X.Zhao CHEP2023 talk “Updates to the ATLAS Data Carousel Project”

MyBigPanDA page

Motivation

Most requests to add/fix something come from experts (shifters, production managers, WFMS component developers etc)

↓

The BigPanDAmon naturally tends towards becoming more complicated expert-focused system that is hard to learn how to use for newly joined experiment members performing their analysis

↓

Needs a simple starting page that collects all analysis tasks submitted by a user and provides insight into task progress and help debugging problems

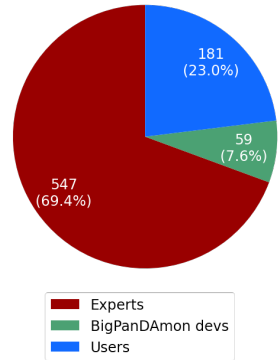
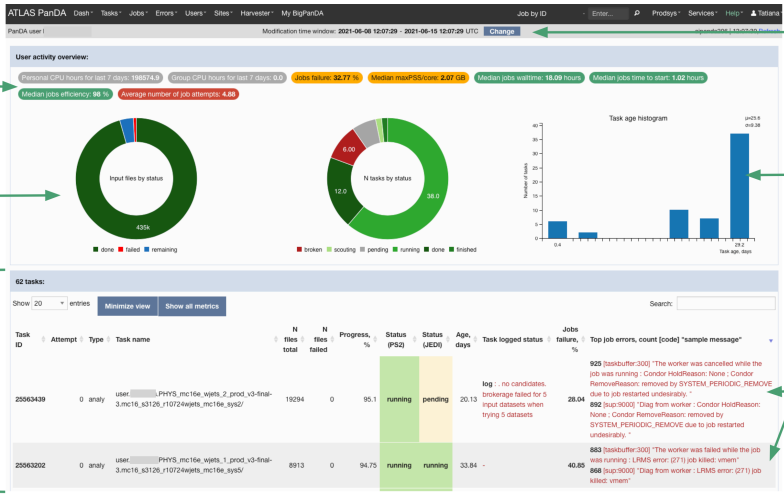


Figure 9 – Roles of improvement requester

MyBigPanDA page

Implementation



Time window interactive selection

Color coded metrics: Alert, warning, OK

Overall progress of tasks in number of input files

Task age

List of tasks

Top job errors per task with link to /jobs/ page

Figure 10 — MyBigPanDA page

Results & Outcome

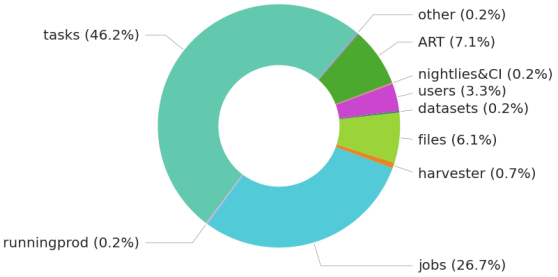


Figure 11 – BigPanDAmon usage statistics

- from 1 to 626 pages a day per user
- 342 unique daily users
- 1110 unique monthly active users

- The BigPanDAmon evolved into a modular monitoring platform that allows to integrate new modules easily
- Number of modules increased from 1 to 11 in last 6 years
- The BigPanDAmon was initially designed for ATLAS but now it is widely used in many experiments (COMPASS, sPHENIX, and Vera C. Rubin)
- The core views are still the most popular among users because of their commonality
- The MyBigPanDA page has been developed to accumulate information of all analysis conducted by a user. We got only positive feedback about it
- It is essential to keep the monitoring views user-friendly and simple despite the changes initiated by experts