# BigPanDA monitoring system evolution in the ATLAS Experiment

*Tatiana* Korchuganova[1,*], *Aleksandr* Alekseev[2], *Alexei* Klimentov[3], *Torre* Wenaus[3], and *Zhaoyu* Yang[3] on behalf of the ATLAS Computing Activity

[1]University of Pittsburgh, Pittsburgh, PA, USA
[2]University of Texas at Arlington, Arlington, TX, USA
[3]Brookhaven National Laboratory, Upton, NY, USA

**Abstract.** Monitoring services play a crucial role in the day-to-day operation of distributed computing systems. The ATLAS Experiment at LHC uses the Production and Distributed Analysis workload management system (PanDA WMS), which allows a million computational jobs to run daily at over 170 computing centers of the WLCG and opportunistic resources, utilizing 600k cores simultaneously on average. The BigPanDA monitor is an essential part of the monitoring infrastructure for the ATLAS Experiment that provides a wide range of views, from top-level summaries to a single computational job and its logs. Over the past few years of the PanDA WMS advancement in the ATLAS Experiment, several new components were developed, such as Harvester, iDDS, Data Carousel, and Global Shares. Due to its modular architecture, the BigPanDA monitor 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. Moreover the system has been adopted by other experiments beyond HEP. In this paper we describe the evolution of the BigPanDA monitor system, the development of new modules, and the integration process into other experiments.

## 1 Introduction

BigPanDA monitor is the next-generation monitoring system of the ATLAS PanDA Workload Management System (WMS) [1], which was developed in the ATLAS Experiment [2] and brought into production in 2014. It has replaced the technologically outdated PanDA monitor web application [3]. Since then, it has been continuously improved in line with the advancement of the PanDA WMS to meet new challenges brought by the ATLAS collaboration, in particular the integration of HPCs and commercial clouds, the central control of computing activity distributed among available resources, and orchestration of the data movement between tape storage and disks. To complete them, the several new components were developed, such as Harvester, Intelligent Data Delivery Service (iDDS) [4], Data Carousel [5], and Global Shares [6]. The overview of the ATLAS Workflow Management System is shown in Figure 1. Central and physics groups requests are submitted via Production System Web-UI

*e-mail: tatiana.korchuganova@cern.ch

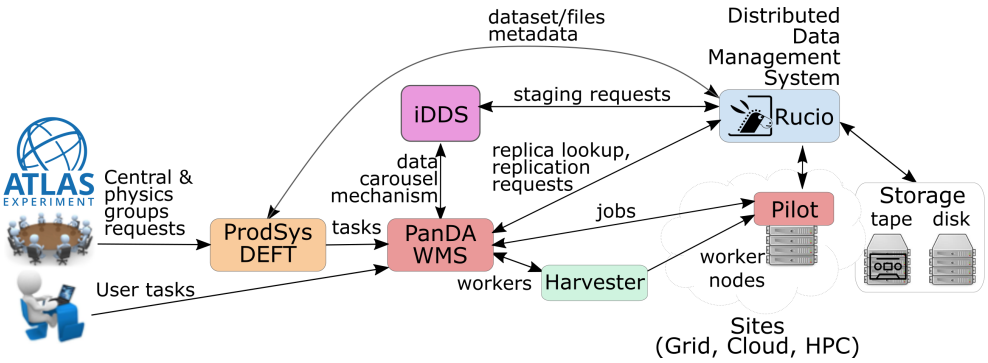



**Figure 1.** ATLAS Workflow Management System overview.

(ProdSys), which is the interface on top of the Database Engine for Tasks (DEFT) [7]. DEFT divides a request into a chain or set of tasks. A task is an entity for passing configuration parameters to the payload submission system. PanDA WMS consumes the tasks and splits each of them into a set of jobs. A job is a single executable workload, which can run on worker nodes across various computing sites (Grid, Cloud, or HPC). PanDA obtains the location of input data, and requests an additional replica if necessary, using the distributed data management system Rucio. If the input data of a task is only available on tape storage, the Data Carousel mechanism is triggered. It orchestrates the data staging from tape to disk storage and delivery to a site where jobs will run in the most optimal way, using the iDDS and the Rucio [8]. Harvester generates and submits pilots using the appropriate communication protocol for each resource provider. A pilot is a transient agent to execute a job on a worker node, periodically reporting various metrics to the PanDA throughout its lifetime.

The success of the PanDA WMS brought interest from other experiments in HEP and beyond. This implied the need for changes in the system development to become experiment-agnostic.

## 2 BigPanDA monitor evolution

BigPanDA monitor is a Django-based web application that collects data from various sources, such as a relational DB (Oracle, PostgreSQL, MySQL), Elasticsearch storage, APIs and prompt clients of other systems (CRIC information system [9], Rucio distributed data management system [8]), aggregates and represents it in the form of interactive tables and charts.

Initially, the project comprised a set of core views covering the basic needs of monitoring key aspects of the PanDA WMS, in particular jobs, tasks, files, datasets, computing sites, and users. With the advancement of the PanDA WMS [1] in the ATLAS Experiment, several new components have been developed, such as Harvester [10], iDDS [4], Data Carousel [5], and Global Shares [6]. All of them are closely related to key WMS objects and have to be monitored in an integrated way. Naturally, BigPanDA monitor started to grow and fulfil new monitoring needs. Also, the interest of other experiments required the ability to easily add or remove components depending on their unique needs. This triggered the transformation of the BigPanDA monitor architecture into a modular platform. From 2017 to 2023, the number of implemented modules increased from 1 to 11 (Figure 2). To perform this transition, we generalized and separated common functions, so they could be reused across modules. The system was updated by:

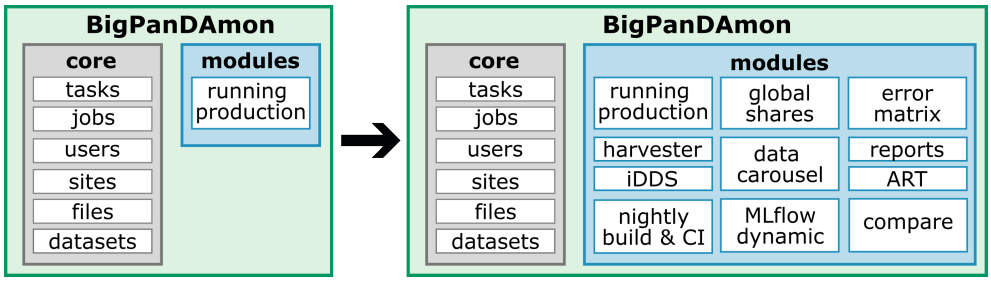



**Figure 2.** BigPanDA monitor structure evolution from 2017 to 2023.

- translating request GET/POST parameters into queries for Django's built-in Object Relational Mapper (ORM) ;
- introducing caching of the prepared-to-be-rendered data, which allows adding user-specific data on top of it for subsequent requests;
- using a common base HTML template and nested blocks of the built-in Django template-rendering engine;
- improving configuration of the application to handle plug in/out of modules;
- and a testing mechanism by creating unit-tests for most of the views.

These improvements make it possible to create new modules and views efficiently and faster because developers only need to work on data querying, aggregation, and visualization with all tools at hand. The next section describes the implementation of several key modules.

## 3 Implementation of new modules

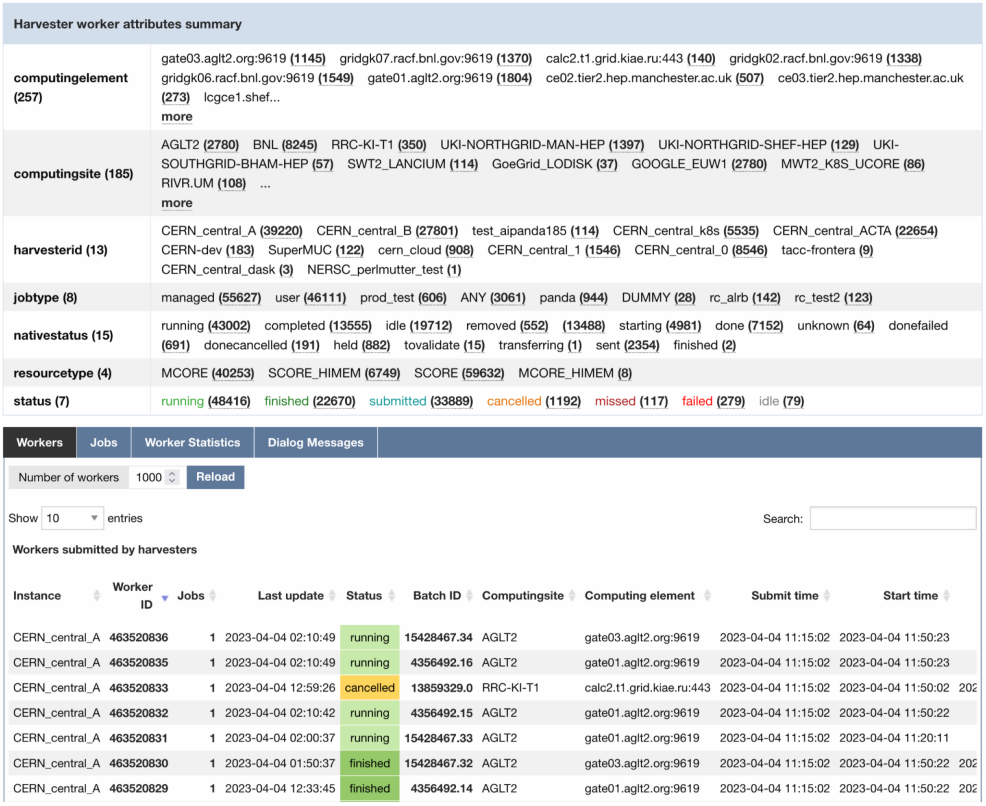
### 3.1 Harvester monitoring module

Harvester [10] is a resource-facing service between the PanDA WMS and pilots for resource provisioning and workload shaping. It has two entities: a Harvester instance that can run centrally with the required plugins activated for different resources or on edge nodes of HPC centres, and a Harvester worker that can be a pilot, VM or MPI job. The Harvester worker has a one-to-many relation with the PanDA job.

The requirements for monitoring Harvester components were: to present a list of all Harvester instances and their states; to be able to search workers per Harvester instance, computing site or computing element; to find a worker for a PanDA job and, vice versa, PanDA job(s) for a worker. To fulfil the needs listed above, we developed four views:

- Harvester instance list;
- Harvester worker summary (Figure 3):
  - summary of worker attributes, allowing for drill-down to workers of interest;
  - list of workers with links to batch logs;
  - list of associated PanDA jobs;
  - workers statistics;
  - diagnostic messages to identify the internal problems of a Harvester instance;

- Harvester worker info page accumulates all information related to the selected worker. It can be found by the worker ID or PanDA job ID;
- Harvester slots page represents the target of cores that can be reached for large, bursty resources (e.g. HPC), for which PanDA needs to pre-queue jobs so ensure a very fast spin up when the resource comes online.;



**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 3.** Harvester worker summary page for the most recent six-hour period.

We faced a performance issue during the development of the Harvester worker summary view. In the ATLAS Experiment, the average number of daily submitted workers is 600K and therefore the time to query and aggregate the data for the last day could reach one minute, which is unacceptable for most users. To mitigate this problem, it was decided to use Elasticsearch, Logstash and Kibana (ELK) stack for global monitoring of Harvester. The data is copied every 10 minutes from the PanDA DB to the central Elasticsearch storage provided by CERN IT [11]. The implemented views of the Harvester monitoring module are mostly used for debugging immediate problems related to a particular worker or PanDA job because BigPanDA monitor reads data directly from the PanDA database without additional delays.

### 3.2 Global Shares monitoring module

The Global Shares component [6] of the PanDA WMS is responsible for optimizing the distribution of workloads across the available computing resources, depending on current priorities

of computing activities set by the ATLAS collaboration. The shares of computing resources may have a nested structure; for example, on the top level, it divides into production and analysis, then production divides into event generation, MC simulation, reconstruction, and derivation activities. The current setup has three levels. The main requirement for the monitoring of this component was the ability to see how close the actual distribution of computing resources is to the target for every share on any level. In addition, it must provide an insight into which type of resources (among the Grid, HPC, Cloud and GPU) contributes the most to each share.

We implemented a dedicated dashboard view to meet these requirements, which consists of three blocks:

- an overview with plots where the level of Global Shares and type of resources can be selected (Figure 4);

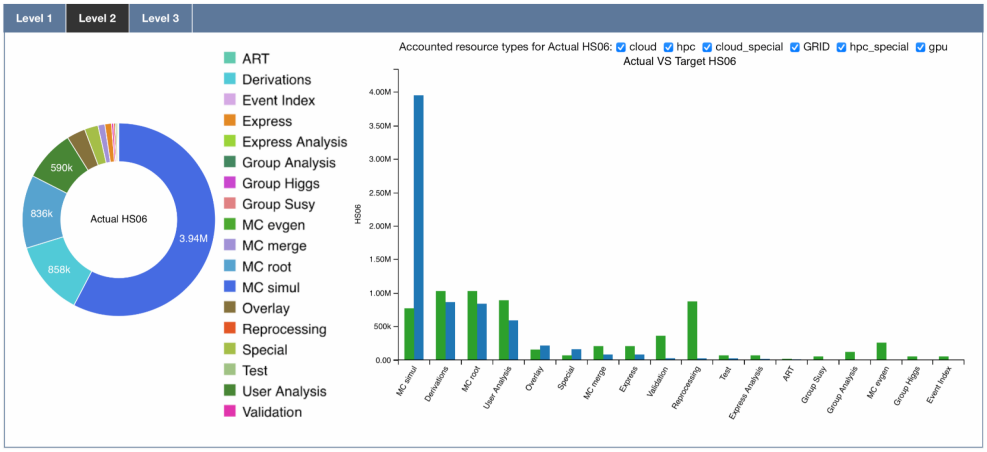



**Figure 4.** Global Shares dashboard.

- a table with the list of Global Shares representing the target and actual values linked to the corresponding PanDA job list;
- a set of tabs containing more detailed information, such as the global share distribution across computing sites, types of computing resources (Grid, HPC, Cloud etc.), resource types (single/multicore, ordinary/high memory), and PanDA job statuses.

### 3.3 Intelligent Data Delivery Service monitoring module

iDDS [4] aims to orchestrate workflows and manage data pre-processing, delivery, and primary processing in large-scale workflows. It supports workflows defined via a Directed Acyclic Graph (DAG).

For monitoring workflows managed by iDDS we created a workflow progress dashboard (Figure 5), which shows a list of workflows, their type and status, and the progress quantified in number of files. iDDS brings flexibility to workflow structures, and connections between its internal steps. Therefore, one of the requirements for monitoring was to build a visualization of workflow complexity. Figure 6 shows an example of such a graph diagram, where each node is a computational task, and edges represent the flow of data. We are planning to improve this visualization by adding more information to it, in particular the status and progress of each task.

Requests:

Show 10 entries Search:

| request id | username | workflow status | graph | workflow name | created on (UTC) | total tasks | tasks | transform type | total files | released files | unreleased files | finished files | failed files |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 508223 | User 1 | SubFinished | plot | pseudo_input.2023_06_26_20_24_18_752267769 | 2023-06-26 20:24:19 | 10 | Finished(8) Failed(2) | Processing | - | 0 | 0 | 95.7% | 4.3% |
| 489653 | User 2 | SubFinished | plot | pseudo_input.2023_05_15_20_22_16_527614670 | 2023-05-15 20:22:17 | 10 | Failed(6) SubFinished(4) | Processing | - | 0 | 0 | 50% | 50% |
| 489639 | User 3 | SubFinished | plot | pseudo_input.2023_05_14_14_56_16_550921349 | 2023-05-14 14:56:17 | 10 | Finished(4) Failed(6) | Processing | - | 0 | 0 | 57.8% | 4.5% |
| 486243 | User 4 | SubFinished | plot | pseudo_input.2023_05_01_01_21_18_02287980 | 2023-05-01 01:21:19 | 10 | Failed(6) SubFinished(4) | Processing | - | 0 | 0 | 50% | 50% |

**Figure 5.** iDDS workflow progress view.

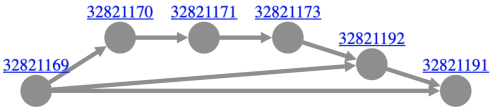



**Figure 6.** Graph plot of one of workflows defined via DAG. Each node is a computational task, which has a unique identification number.

### 3.4 Data Carousel monitoring module

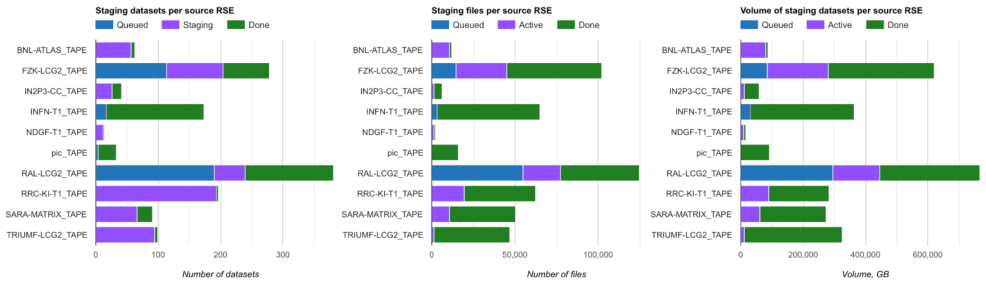
The Data Carousel [5] orchestrates data processing between workload management, data management, and storage services with the bulk data resident on tape. The processing is executed by staging and promptly processing a sliding window of inputs onto faster buffer storage such that only a small percentage of input data are available at any one time.

The development of this module was particularly challenging due to its intermediate position between the workflow and distributed data management systems. The main source of information are the ATLAS Production System [7] DB tables, which store the datasets that need to be staged-in, the corresponding Rucio rules, and their progress. Detailed information regarding staging transfers comes from the central Elasticsearch storage filled by Rucio.

A dedicated dashboard has been developed for monitoring staging activities (Figure 7), which contains various visualizations of staging datasets/files/their volume and a table with staging requests and associated tasks that have input on tape. In addition, a notification mechanism for stalled staging requests was implemented. It regularly searches for staging requests that have not progressed for $N$ days (configurable, currently set to 10) and sends an email report to the list of relevant experts.

## 4 MyBigPanDA view

The development of the new modules described in Section 3 was initiated by requests from experts: WMS component developers, shifters, production managers, and system administrators of computing centres. This group of users is very active with almost half of requests to BigPanDA monitor despite the small number of people (around 100). On the other side, the number of physicists sending their analysis tasks to the system reaches 1500 people in peaks. Only 25% of improvement requests or bug reports come from analysis users, while the other 75% are from experts. Therefore, BigPanDA monitor naturally tends towards becoming a more complicated expert-focused system that is difficult to learn for experiment members performing their analysis. This necessitates the creation of a simple overview page




**Staging datasets:**

Show 10 entries Search:

| Campaign | Request ID | Task ID | Status | P-type | Size [GB] | Total files | Staged files | Progress [%] | Source RSE | Time elapsed | Started at | Rucio rule | Update time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| data17_13TeV | 48500 | 32938794 | done | deriv | 25072.45 | 6202 | 6202 | 100 | SARA-MATRIX_TAPE | 1 day, 22:19:54 | 2023-04-02 17:29:25 | faa...aef | 0:02:17 |
| data17_13TeV | 48500 | 32938466 | done | deriv | 5278.08 | 2787 | 2787 | 100 | FZK-LCG2_TAPE | 22:26:53 | 2023-04-02 17:29:25 | 5b8...2b5 | 21:55:14 |
| data17_13TeV | 48500 | 32938666 | done | deriv | 9529.9 | 4690 | 4690 | 100 | INFN-T1_TAPE | 3 days, 2:08:31 | 2023-04-01 06:12:28 | 957...088 | 0:02:17 |
| data17_13TeV | 48500 | 32938677 | done | deriv | 12835.38 | 4630 | 4630 | 100 | RAL-LCG2_TAPE | 1 day, 0:26:57 | 2023-04-02 17:29:25 | 80d...525 | 14:34:54 |
| data18_13TeV | 46610 | 32975425 | queued | reprocessing | 11293.17 | 4752 | 0 | 0 | RAL-LCG2_TAPE | 8:18:38 | 2023-04-04 07:33:00 | --- | --- |
| MC16 | 47465 | 32925207 | done | simul | 16.87 | 40 | 40 | 100 | INFN-T1_TAPE | 2 days, 16:16:15 | 2023-04-01 06:12:19 | 003...de2 | 5:22:28 |
| MC16 | 47465 | 32925209 | queued | simul | 16.82 | 40 | 0 | 0 | FZK-LCG2_TAPE | 5 days, 12:23:59 | 2023-03-30 03:27:39 | --- | --- |
| MC16 | 47465 | 32925211 | done | simul | 16.76 | 40 | 40 | 100 | INFN-T1_TAPE | 2 days, 18:16:20 | 2023-04-01 06:12:19 | 0e5...0a2 | 11:22:47 |
| MC16 | 47465 | 32925215 | queued | simul | 16.81 | 40 | 0 | 0 | RAL-LCG2_TAPE | 5 days, 12:23:59 | 2023-03-30 03:27:40 | --- | --- |
| MC16 | 47465 | 32925217 | queued | simul | 16.8 | 40 | 0 | 0 | RAL-LCG2_TAPE | 5 days, 12:23:58 | 2023-03-30 03:27:40 | --- | --- |

All All All All

Showing 1 to 10 of 1,382 entries Previous 1 2 3 4 5 ... 139 Next

**Figure 7.** Data Carousel dashboard.

that collects all analysis tasks submitted by a user, provides insight into task progress, and helps debugging problems. An example of this page is shown in figure 8. It consists of two blocks. The first is an overview showing statistics of user activity: the progress of the submitted tasks in the number of processed input files, the distribution of task statuses, a histogram of task age that allows slow tasks to be spotted, and colour-coded job metrics. These metrics can provide clues to possible problems. The second block is a table with a list of tasks. The last column of the table shows the most common errors across jobs for each task and the link to the log for one job. The MyBigPanDA view is actively used; during the last year, 1471 out of 2123 unique users visited it at least once. The number of daily visits by a single user varies from 0 to 212.

## 5 Adapting to other experiments

As of 2023 several other experiments are using the PanDA WMS for distributed data processing and BigPanDA monitor for monitoring, in particular COMPASS [12] at the Super Proton Synchrotron (SPS), Vera Rubin observatory [13], and sPHENIX [14] at Brookhaven National Laboratory (BNL). Each experiment has unique requirements and can choose which components of PanDA WMS to use. The main difficulty in adapting the monitoring system was the requirement to use different database vendors. ATLAS uses Oracle DB, while COMPASS requires MySQL. V. Rubin Observatory and sPHENIX use PostgreSQL. BigPanDA monitor views mostly use the Django built-in Object Relational Mapper (ORM) for querying data from databases. However, there are cases where the functionality is not enough, and for those we have added functions which change the syntax of queries depending on the database provider. Also, we have introduced new configuration variables to manage the content of views depending on the experiment. Another requirement was to run the BigPanDA monitor

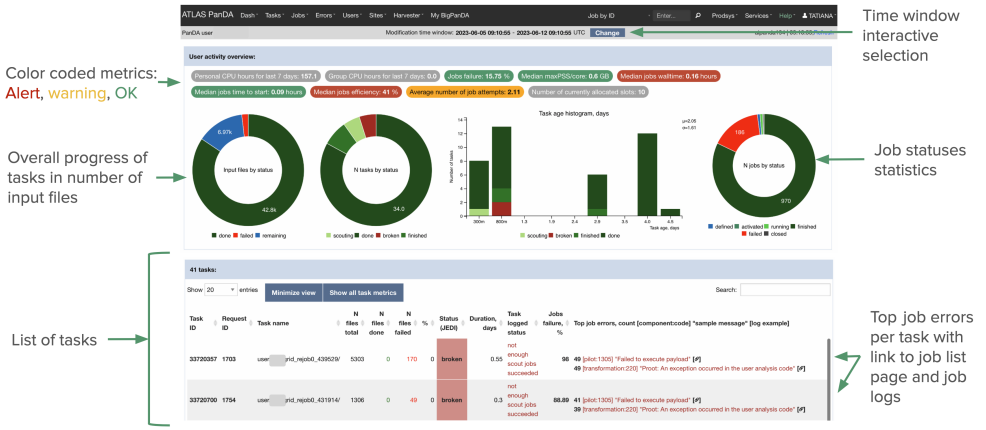



**Figure 8.** MyBigPanDA page. The red median job duration label highlights that jobs are very short (~ 10 min), which affects the overall efficiency of data processing, because typical Grid jobs are expected to run for a duration of order 24 hours. The error message in the last column of the table ("Failed to execute payload. An exception occurred in user analysis code") clearly indicates the problem. In this case, the user can see the log with just one click.

on a Kubernetes cluster, therefore we have created a Docker image and configured a GitHub action to build a container automatically for new releases.

## 6 Summary

The BigPanDA monitor system has significantly evolved since its first release in 2014. The architecture naturally transformed from an application into a multimodule platform which accumulates monitoring views for all components of the PanDA WMS. We are continuously working on improving system usability and transparency for ATLAS collaboration members doing their physics analysis.

## Acknowledgement

We appreciate the assistance of our colleagues from the PanDA team and ATLAS Distributed Computing group. We thank Siarhei Padolsky for a significant contribution to the development and supervising the project for many years.

## References

[1] T. Maeno, K. De, T. Wenaus, P. Nilsson, R. Walker, A. Stradling, V. Fine, M. Potekhin, S. Panitkin, G. Compostella, Journal of Physics: Conference Series **396**, 032071 (2012)
[2] ATLAS Collaboration, J. Inst. **3**, S08003 (2008)
[3] A. Klimentov, P. Nevski, M. Potekhin, T. Wenaus, Journal of Physics: Conference Series **331**, 072058 (2011)
[4] W. Guan et al., EPJ Web Conf. **251**, 02007 (2021)
[5] M. Borodin et al., EPJ Web Conf. **251**, 02006 (2021)

[6] F.H. Barreiro Megino, A. Di Girolamo, K. De, T. Maeno, R. Walker, EPJ Web Conf. **214**, 03025 (2019)

[7] F.H. Barreiro Megino, M. Borodin, K. De, D. Golubkov, A. Klimentov, T. Maeno, R. Mashinistov, S. Padolski, T. Wenaus, on behalf of the ATLAS Collaboration, Journal of Physics: Conference Series **898**, 052016 (2017)

[8] M. Barisits et al., Comput Softw Big Sci **3**, 11 (2019)

[9] A. Anisenkov et al., EPJ Web Conf. **245**, 03032 (2020)

[10] T. Maeno, F.H. Barreiro Megino, D. Benjamin, D. Cameron, J.T. Childers, K. De, A. De Salvo, A. Filipcic, J. Hover, F. Lin et al., EPJ Web Conf. **214**, 03030 (2019)

[11] F.H. Barreiro Megino, A. Alekseev, F. Berghaus, D. Cameron, K. De, A. Filipcic, I. Glushkov, F. Lin, T. Maeno, N. Magini, EPJ Web Conf. **245**, 03010 (2020)

[12] P. Abbon, E. Albrecht, V. Alexakhin, Y. Alexandrov, G. Alexeev, M. Alekseev, A. Amoroso, H. Angerer, V. Anosov, B. Badełek et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **577**, 455 (2007)

[13] Z. Ivezic et al., Astrophys. J., **873**, 111 (2019)

[14] A. Adare et al. (PHENIX) (2015), 1501.06197