Monitoring and Accounting for the Distributed Computing System of the ATLAS Experiment

Dario Barberis\textsuperscript{1}, Alberto Aimar\textsuperscript{2}, Aleksandr Alekseev\textsuperscript{3}, Pedro Andrade\textsuperscript{2}, Thomas Beermann\textsuperscript{4}, Robert Gardner\textsuperscript{5}, Borja Garrido Bear\textsuperscript{2}, Tatiana Korchuganova\textsuperscript{3}, Luca Magnoni\textsuperscript{2}, Siarhei Padolski\textsuperscript{6}, Eric Schanet\textsuperscript{7}, Nikolay Tsvetkov\textsuperscript{2}, Ilija Vukotić\textsuperscript{5}, Torre Wenaus\textsuperscript{6}

\textsuperscript{1) Genoa Univ./INFN, 2) CERN-IT, 3) Moscow State Univ. & Plekhanov Russian Univ. of Economics, 4) Wuppertal Univ., 5) Univ. of Chicago, 6) Brookhaven National Laboratory, 7) LMU Munich}
Introduction

- ATLAS used during LHC Run 1 and Run 2 a monitoring and accounting infrastructure for the Distributed Computing (ADC) applications developed ~10 years ago by CERN-IT together with ATLAS members.
  - These “old dashboards” started showing aging effects in the last few years:
    - Slowness of data retrieval due to the massive amount of data in Oracle databases
    - Lack of in-depth knowledge for maintenance as original developers left long ago
    - Lack of flexibility and impossibility to develop new views and/or data correlations across different data sources
  - This system worked well enough for general monitoring till the end of Run 2 last year but was evidently in need of a good refurbishing.

- Since 2016 the CERN-IT MonIT group started developing a new infrastructure and environment for monitoring and accounting applications base on modern Open Source components.
  - ATLAS started implementing “new” dashboards using this infrastructure, for data and workload accounting and global monitoring.

- In the meantime the BigPandaMon application was developed for user and task oriented monitoring of the jobs submitted to the ATLAS Grid/Cloud/HPC resources through PanDA.
  - This is now the workhorse of user-level job monitoring.

- In the recent years many Analytics tools appeared on the market. They can be used for more detailed investigations and to correlate data from different sources.
  - The Analytics cluster provided by the University of Chicago allows a more interactive use of monitoring data.
Topics

- ADC dashboards in the MonIT infrastructure
- User level job/task monitoring with BigPandaMon
- Analytics cluster at UC and its applications
Topics

- ADC dashboards in the MonIT infrastructure
- User level job/task monitoring with BigPandaMon
- Analytics cluster at UC and its applications
The MonIT Infrastructure

- **Monitoring Mission:**
  - Provide Monitoring as a Service for CERN Data Centre (DC), IT Services and the WLCG collaboration
    - e.g. Dashboards, Alarms, Search, Archive
  - Collect, transport, store and process metrics and logs for applications and infrastructure

- **Challenges:**
  - **Rate & Volume**
    - from ~ 40k machines
    - > 3 TB/day
    - ~ 100 kHz
  - **Variety**
    - > 150 producers
  - **Reliability**
    - spikes in rate and volume
The MonIT Infrastructure

Collectors / Collectd
- What is Collectd
  "collectd is a daemon which collects system and application performance metrics periodically and provides mechanisms to store the values in a variety of ways"
- Collectd for CERN DC and IT Services
  - Default HW/OS plugins on all DC machines
  - Additional plugins configured by IT Service Managers
    - Community-plugins or custom-made
  - Provide local alarms (~ Nagios Check) and actuators
  - Sampling intervals: 1 to 5 minutes

Collectors / Flume & Logstash
- Flume for basic log forwarding
  - Running on all DC nodes for syslog
- Logstash for advanced use cases
  - Many read/write plugins
  - Allows log parsing and tokenization
- Other lightweight options getting popular
  - Filebeat, Fluentd

Connectors / Apache Flume
- We rely on several Flume features:
  - Transactional nature
  - Pull/push modes
  - Protocol-based agents (sources and sinks)
    - AVRO, JDBC, JMS, HDFS, Elasticsearch, HTTP, Kafka
  - Interceptor / Morphlines for event transformation and validation

Transport / Apache Kafka
- Kafka as rock-solid core of our pipeline
  - decouples producers / consumers
  - enables stream processing
  - resilient (72 hours data retention)
The MonIT Infrastructure

**Storage / InfluxDB**
- Time Series DB for time series data (metrics)
- Data kept for ~5 years (auto down-sampling)
- >30 instances (from 8 to 128GB mem each)
- Data cardinality matters
- Aggregated write: ~65k points per second
- Gross read (single instance): ~100k pps

**Storage / HDFS**
- HDFS for long term archive
- Compressed JSON (or Parquet)
- Data kept ~ forever (or by GDPR agreement)
- Works well with batch and data-intensive analytics workflows
  - e.g. Hadoop, Spark

**Storage / Elasticsearch**
- Distributed search and indexing engine
- Powerful query language to browse and explore
  - Apache Lucene based
- 3 clusters (syslog, service logs and metrics)
  - ~100 TB (total storage size)
- Data kept for 1 month

**Visualization**
- Grafana for dashboards
  - Users can create their own
- Kibana for data exploration
  - Data discovery and logs
- SWAN for analytics (notebooks)
ATLAS Dashboards

- “MonIT Dashboards” for Data and Job monitoring and accounting (and a number of other applications too)
  - Collect data from the Rucio and PanDA databases in Oracle and other sources
  - Transfer to BigData infrastructure using Kafka - possibility of data enrichment and correlations with other sources of information
    - (e.g. AGIS for the relation between PanDA queues, sites, federations, countries, pledges etc.)
  - Storage in HDFS with aggregation in InfluxDB
  - Display with Grafana
    - Still far from perfect as display tool for us but usable
    - Used by ADC shifters and site admins (monitoring) and computing managers (accounting)

- Three groups of dashboards:
  - Production
  - Development (pre-production)
  - Playground (free for all)
Data Management (DDM) Dashboards

- Traces and events reported constantly by Rucio to the message brokers and further processed in the MonIT infrastructure
- Rucio data are dumped periodically from Oracle to HDFS for site accounting views
  - Then used from there to fill InfluxDB
- Many many views available
  - Historical views and snapshots
  - Data storage volumes
  - Transfer rates and efficiencies

Volume per experiment_site

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Min</th>
<th>Max</th>
<th>Avg</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNL-ATLAS</td>
<td>3.01</td>
<td>19.10</td>
<td>16.23</td>
<td>18.75</td>
</tr>
<tr>
<td>CERN-PROD</td>
<td>7.96</td>
<td>20.08</td>
<td>16.32</td>
<td>18.03</td>
</tr>
<tr>
<td>RAL-LCG2</td>
<td>0 B</td>
<td>11.66</td>
<td>5.43</td>
<td>11.66</td>
</tr>
<tr>
<td>FZK-LCG2</td>
<td>4.69</td>
<td>10.23</td>
<td>8.66</td>
<td>10.20</td>
</tr>
<tr>
<td>IN2P3-CC</td>
<td>3.35</td>
<td>9.65</td>
<td>7.45</td>
<td>9.62</td>
</tr>
</tbody>
</table>
Job Dashboard

- Data collected from the PanDA DB every 10 minutes and grouped in 1-hour bins
- Transient job stats:
  - Pending, running, finalising jobs
- Permanent job stats:
  - Submitted and completed jobs
- All job parameters on submission and completion
  - Plus site info from AGIS
  - All available for selections and display
- Data imported from start of Run 1

2019: Cores by Activity

2019: Cores by Site

2019: HS06-sec Power vs Pledges
Other Dashboards

- Several lighter dashboards have been implemented for service monitoring
- Also for short-term site oriented job monitoring
• Derived collection of statistics and plots
  ▪ Jobs, data storage, transfers, services
  ▪ Inputs from different dashboards
  ▪ Also directly from services and Elogs
• Contains links to information sources for deeper investigations
• Used by shifters and managers for daily monitoring
  ▪ Refreshed hourly

- Live Page

- Derived collection of statistics and plots
  - Jobs, data storage, transfers, services
  - Inputs from different dashboards
  - Also directly from services and Elogs
- Contains links to information sources for deeper investigations
- Used by shifters and managers for daily monitoring
  - Refreshed hourly
Topics

- ADC dashboards in the MonIT infrastructure
- User level job/task monitoring with BigPandaMon
- Analytics cluster at UC and its applications
- A window into the PanDA system
- > 100 different views
- From production dashboards to logs
- Covers scope in range $1 \ldots 10^{11}$ events
Dario Barberis: ATLAS ADC Monitoring

Architecture & current usage

17000 json requests a day

- 6.5 (+3%) user queries a day
- From 1 to 626 pages a day per user
- 1110 monthly active users
- 342 daily users

Is a primary tool ATLAS wide for shifter, experts and ADC in general
BigPandaMon Data-flow diagram

1. **Cron job every 10 min**
   - HTTPS request
   - **BP view**
     - Data pre-processing
     - Rucio
     - Aggregates raw data
     - Oracle DB

2. **Redis storage**
   - JSON
   - Getting cached data
   - JSON
   - **BP view**
     - Data aggregation and preparation
     - HTML table
     - Elasticsearch cluster
     - MONIT

3. **BigPandaMon monitor**
   - DataTables
   - Kibana visualization
   - Plot D3.js
   - Plot Matplotlib
   - Dashboard histogram
   - Image

Sources:
- BigPandaMon Data-flow diagram
- ATLAS ADC Monitoring

Event:
- NEC2019 - Budva (ME) 30 September 2019

Presenter:
- Dario Barberis
NEC2019 - Budva (ME) 30 September 2019

User & task page

- User page: all tasks
- Task page:
  - All jobs
  - Exec CPU & memory plots
  - Task exec time

### User page: all tasks

The user page displays all tasks associated with the user. It includes details such as task name, task status, input files, total/remaining events, modified date, state changed, priority, and notes.

### Task page

- **All jobs**
- **Exec CPU & memory plots**
- **Task exec time**

#### ATLAS PANDA

The ATLAS PANDA dashboard provides a visual representation of task performance, including execution times and resource usage. It shows the execution profile for a specific task, indicating where it is in the process and its status.

---

### Task Details

- **Task ID**: 11016615
- **Request Type**: AP_GLOBAL
- **User**: dmshe
- **Campaign**: MC16_MC17
- **Task Status**: running
- **Average CPU time**: 6033 sec
- **Average memory**: 5057 MB
- **Execution time**: 2017-03-23 06:30:37
- **Status**: ["running"]

---

**Execution profile for task 11016615**:

- **Job ID**: 11016615
- **Request Type**: AP_GLOBAL
- **User**: dmshe
- **Campaign**: MC16_MC17
- **Task Status**: running
- **Average CPU time**: 6033 sec
- **Average memory**: 5057 MB
- **Execution time**: 2017-03-23 06:30:37
- **Status**: ["running"]
Jobs Page

specifying parameters

ATLAS PanDA
Dash - Tasks - Jobs - Errors - Users - Sites - Incidents - Search - Admin
PanDA jobs, last 12 hours. Params: limit=10000

Warning: limit 20000 per job table. Task has 0 jobs in job table. Jobs archived, limit is set to 20000.

Total jobs found = 147000

30013 jobs in this selection

Job modification times in this listing range from 2017-09-25 20:50:36 to 2017-09-26 08:50:36.
Job current priorities in this listing range from 100000000 to -100000000. See priority range in the job attribute summary to see how priorities are distributed.

Job attribute summary

<table>
<thead>
<tr>
<th>Job attribute</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlas release</td>
<td>41</td>
</tr>
<tr>
<td>AttemptNR</td>
<td>59</td>
</tr>
<tr>
<td>Cloud</td>
<td>13</td>
</tr>
<tr>
<td>Computing site</td>
<td>284</td>
</tr>
</tbody>
</table>

Overall error summary

<table>
<thead>
<tr>
<th>Category:code</th>
<th>Attempt list</th>
<th>Errors</th>
<th>Sample error description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ddm:200 jobs</td>
<td></td>
<td>3</td>
<td>Could not get GUID/LFN/MD5/FSIZE/SURL from pilot XML</td>
</tr>
</tbody>
</table>

Job list

Only the most recent 100 jobs are shown. Remove the limit and sort by PandID, since last state change, ascending mod time, descending mod time, priority, attemptNR, ascending duration, descending duration.

<table>
<thead>
<tr>
<th>PandID</th>
<th>Owner Group</th>
<th>Request Task ID</th>
<th>Transformation</th>
<th>Status</th>
<th>Created</th>
<th>Time to start d/h/m/s</th>
<th>Duration d/h/m/s</th>
<th>Mod</th>
<th>Cloud Site</th>
<th>Priority</th>
<th>Job info</th>
</tr>
</thead>
<tbody>
<tr>
<td>3315291794 Attempt 1</td>
<td>igoronzale_AP_SOFT</td>
<td>11341 11081388</td>
<td>POOL:bel_fy</td>
<td>activated</td>
<td>2017-04-01 06:07:53</td>
<td>00:52:59</td>
<td>2017-04-01 07:00:43</td>
<td>WORLD SARA-MATRIX</td>
<td>online no active blacklisting rules defined</td>
<td>880</td>
<td>*data_evid\00311365.physics_Main.eventindex.\0264_p3083_t14_3315291794 #1</td>
</tr>
</tbody>
</table>

Datasets: input: data\16\1371\data\16_1371\16_00311365.physics\_Main.merge.AOD.r9264_p3083_t111038617_00
Output: data\evid\00311365.physics\_Main.eventindex.log.r9264_p3083_t14_111038688_00

---

Total number of jobs matched to specified parameters
Topics

- ADC dashboards in the MonIT infrastructure
- User level job/task monitoring with BigPandaMon
- Analytics cluster at UC and its applications
It provides an interactive environment to:

- Develop additional or alternative dashboards
- Investigate correlations between several data sources
- Complementary to the monitoring and accounting infrastructure at CERN

Analytics Infrastructure

**CERN**

- DB - Oracle, Hadoop, Ingress, Elastic
- Processing - Sqoop, pig, Spark, SWAN, Kubernetes cluster

**UC**

- Elasticsearch
- K8s cluster, ML platform

Data sources:

- Oracle (panda, jedi, rucio)
- MySQL (BOINC)
- AMQ (FTS)
- RMQ (PerfSonar)
- Pilot (benchmarks)
- User codes (xAOD usage)
- Frontiers (full logs)
- HC jobs (cost matrix)
- SLATE
- XCachec

Data collection infrastructure - CERN

- All services dockerized
- All running in a single kubernetes cluster at CERN. Very stable.

Data collection infrastructure - UChicago

- All services dockerized. All running at UChicago kubernetes cluster.
- Running:
  - logstash collectors
  - rollup jobs (DDM)
  - Alarm & Alert service
  - REST interfaces (benchmarking, cost matrix, xcache backend)
Analytics Cluster at UC

- It provides an interactive environment to:
  - Develop additional or alternative dashboards
  - Investigate correlations between several data sources
- Complementary to the monitoring and accounting infrastructure at CERN

Analytics platforms

While all the data can be analysed using kibana, or JupyterLab notebooks, some projects would benefit from a custom analytics platforms:
- Extra processing and result caching
- Dedicated web site
- Custom searches
- Custom visualizations

We have a way to run these platforms in UC River2 K8s cluster. Currently we support two - Frontier and Perfsonar analytics platforms. Expect more to come.

Frontier analytics tools

Perfsonar analytics platform

- Made by Евгений Третьяков Сергеевич (MEPhI)
- Django + ES + JavaScript libraries: three.js, d3-force-3d, jquery
- Future (short term)
  - Full production deployment
  - Pagination
  - Adaptive interface Full HD+ (Now only FullHD)
  - Advanced search
- Future (long term)
  - Integration of derived data
  - Other Perfsonar indices
ATLAS Distributed Computing has a coherent set of monitoring and accounting dashboards and interactive tools.

- Technologies evolve all the time - we follow them
  - Trying to use Open Source solutions as much as possible
  - Even if at times some home-made parts are inevitable
    - See the low number of display options in Grafana

- The future is in more interactive environments providing the possibility to correlate information from many different sources
  - This is real BigData in action
    - Not just tens of billions of statistically equivalent event records!

- Any way we are quite well set for the start of LHC Run 3
Thank you!

Thank you for listening
Thanks to all authors of this work

This work was funded in part by the Russian Science Foundation under contract No. 19-71-30008
(research is conducted in the Plekhanov Russian University of Economics)