Applying Big Data solutions for log analytics in the PanDA infrastructure

F. Barreiro Megino, A. Alekseev, A. Klimentov, T. Korchuganova, T. Maeno, S. Padolski

2017
Outline

● ATLAS, PanDA, JEDI, BigPanDA

● Reasons for choosing filebeat and logstash and results of implementation for PanDA, JEDI and BigPanDA logs

● Results of integration with BigPanDA monitoring

● BigPanDA logs processing. Notification system for BigPanDA monitoring
Introduction

- **ATLAS** is one of two general-purpose detectors at the Large Hadron Collider (LHC). 1PB/sec from all sub-detectors reduces to ~1GB/sec raw data which is saved to Data Center.
- **PanDA (Production and Distributed Analysis)** is the workflow management system of the ATLAS experiment at the LHC and is responsible for generating, brokering and monitoring up to two million jobs per day across 150 computing centers in the Worldwide LHC Computing Grid.
- **JEDI (Job Execution and Definition Interface)** is an intelligent component in the PanDA system to have capability for task-level workload management.
- **BigPanDA** is a monitoring system which provides a comprehensive and coherent view of the tasks and jobs executed by the system, from high level summaries to detailed drill-down job diagnostics.
Motivation

- Centralized storage for PanDA server, JEDI and BigPanDA logs
- 400GB daily logs
- Average size of the one log file can be up to 2 GB
- Quick search of errors and potential problems
- Notification features
- ELK (Elasticsearch+Logstash+Kibana) have more performance than Flume and parses/indexes messages
ELK stack and Flume

- **ELK stack**
  - **Logstash** is an special open source software for collecting, filtering and normalizing logs. It is used to collect log events from different log types.
  - **ElasticSearch** is a distributed open source software for storing and searching information. In our case, Logstash writes all log events into the ElasticSearch repository.
  - **Kibana** is an open source data visualization plugin for Elasticsearch. It is used for visualization of data from Elasticsearch cluster.

- **Apache flume (or Flume)** is a distributed service for collecting, aggregating, and moving large amounts of log data.
Scale of the Problem

PanDA server

~ 210 GB per week in ES (34 types of log filters/files)

JEDI

~ 73 GB per week in ES (27 types of log filters/files)
Example of log processing

Input line

2016-10-20
01:37:34,202
panda.log.JobGenerator: INFO
<jediTaskID=9672755
datasetID=101428269>
successfully submitted 25/25 for VO=atlas cloud=None queue=user status=running

Grok filter

timeEvent: 2016-10-20
01:37:16,377
logName: panda.log.JobGenerator
logLevel: INFO
message:
<jediTaskID=9672755
datasetID=101428269>
successfully submitted 25/25 for VO=atlas cloud=None queue=user status=running

KV filter
(from the message field)

jediTaskID: 9672755
datasetID: 101428269
VO: atlas
cloud: None
queue: user
status: running

Ruby filter
(from the message field)

jobs_submitted: 25
jobs_tried: 25

Output result

timeEvent: 2016-10-20
01:37:16,377
logName: panda.log.JobGenerator
logLevel: INFO
jediTaskID: 9672755
datasetID: 101428269
VO: atlas
cloud: None
queue: user
status: running
jobs_submitted: 25
jobs_tried: 25
Examples: Panda server health metrics

- Average response time (seconds)
- Timestamp per 30 minutes
- Request count

Possibility to filter by host
### Integration with BigPanDA monitoring. Logs page

#### Log types ordered by interest

<table>
<thead>
<tr>
<th>Type</th>
<th>Level</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>atlasprodraskbroker (?)</td>
<td>INFO</td>
<td>14687</td>
</tr>
<tr>
<td>atlasanaljobbroker (?)</td>
<td>INFO</td>
<td>4216525</td>
</tr>
<tr>
<td>atlasprodpodjobbroker (?)</td>
<td>INFO</td>
<td>919857</td>
</tr>
<tr>
<td>jobgenerator (?)</td>
<td>DEBUG</td>
<td>1378069</td>
</tr>
<tr>
<td>atlasprodpodjobthrottler (?)</td>
<td>INFO</td>
<td>83318</td>
</tr>
<tr>
<td>jobsplitter (?)</td>
<td>DEBUG</td>
<td>495025</td>
</tr>
<tr>
<td>taskrefiner (?)</td>
<td>ERROR</td>
<td>606763</td>
</tr>
</tbody>
</table>

#### Log level and number of entries

<table>
<thead>
<tr>
<th>Type</th>
<th>Level</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrialmodule (?)</td>
<td>INFO</td>
<td>369999</td>
</tr>
<tr>
<td>dbproxyfiltered (?)</td>
<td>INFO</td>
<td>70979</td>
</tr>
<tr>
<td>serveraccess (?)</td>
<td>INFO</td>
<td>8202664</td>
</tr>
<tr>
<td>entry (?)</td>
<td>INFO</td>
<td>7990110</td>
</tr>
<tr>
<td>userif (?)</td>
<td>INFO</td>
<td>72</td>
</tr>
<tr>
<td>pilotrequest (?)</td>
<td>DEBUG</td>
<td>3717655</td>
</tr>
<tr>
<td>servererror (?)</td>
<td>INFO</td>
<td>22364</td>
</tr>
<tr>
<td>activatorlog (?)</td>
<td>DEBUG</td>
<td>43772</td>
</tr>
<tr>
<td>copyarchive (?)</td>
<td>DEBUG</td>
<td>70888</td>
</tr>
<tr>
<td>finisher (?)</td>
<td>DEBUG</td>
<td>604927</td>
</tr>
</tbody>
</table>

**Tooltip:**

- Generates tasks in JEDI from definitions found in DEFT
Integration with BigPanDA monitoring. Dashboards implementation
Notification system. Example of notification message

Error description


More information here:

https://es-atlas.cern.ch/kibana/app/kibana#/discover?_g=(refreshInterval:(display:Off,pause:1f,value:0),time:(from:'2017-09-04T05:31:00.000Z',mode:absolute,to:'2017-09-04T05:35:00.000Z'))&_a=(columns:!(request),index:'atlas_bigpanda-*',interval:auto,query:(query_string: (analyze_wildcard:t,lowercase_expanded_terms:f,query:'*'))),sort:![('@timestamp',desc)]

Detailed information
Notification system. Information about error in ES

Total entries within 4 minutes

Django log with error description

Apache log with user request
Summary

- Logstash infrastructure is set up to process PanDA server, JEDI and BigPanda logs
  - 67 filters for PanDA, JEDI and BigPanDA logs are ready and in production
- Improve the lifetime model of the logs
  - for the moment we are only keeping all of them 20 days
- The system has a lot of potential to monitor, investigate and understand PanDA and ProdSys
  - The infrastructure is ready and several dashboards for real-life issues have been implemented
  - We need to continue looking/understanding the data and identify further necessary visualizations
- Results were integrated with BigPanDA monitoring
  - The special page with was created in BigPanDA monitoring
  - Further integration points with BigPanDA will be added
- Notification system was created
  - Error messages are sent to the BigPanDA developers immediately
  - Error messages are stored in ES
Thank you for your attention!