Kibana, Grafana and Zeppelin on Monitoring data

Technical Report · August 2016

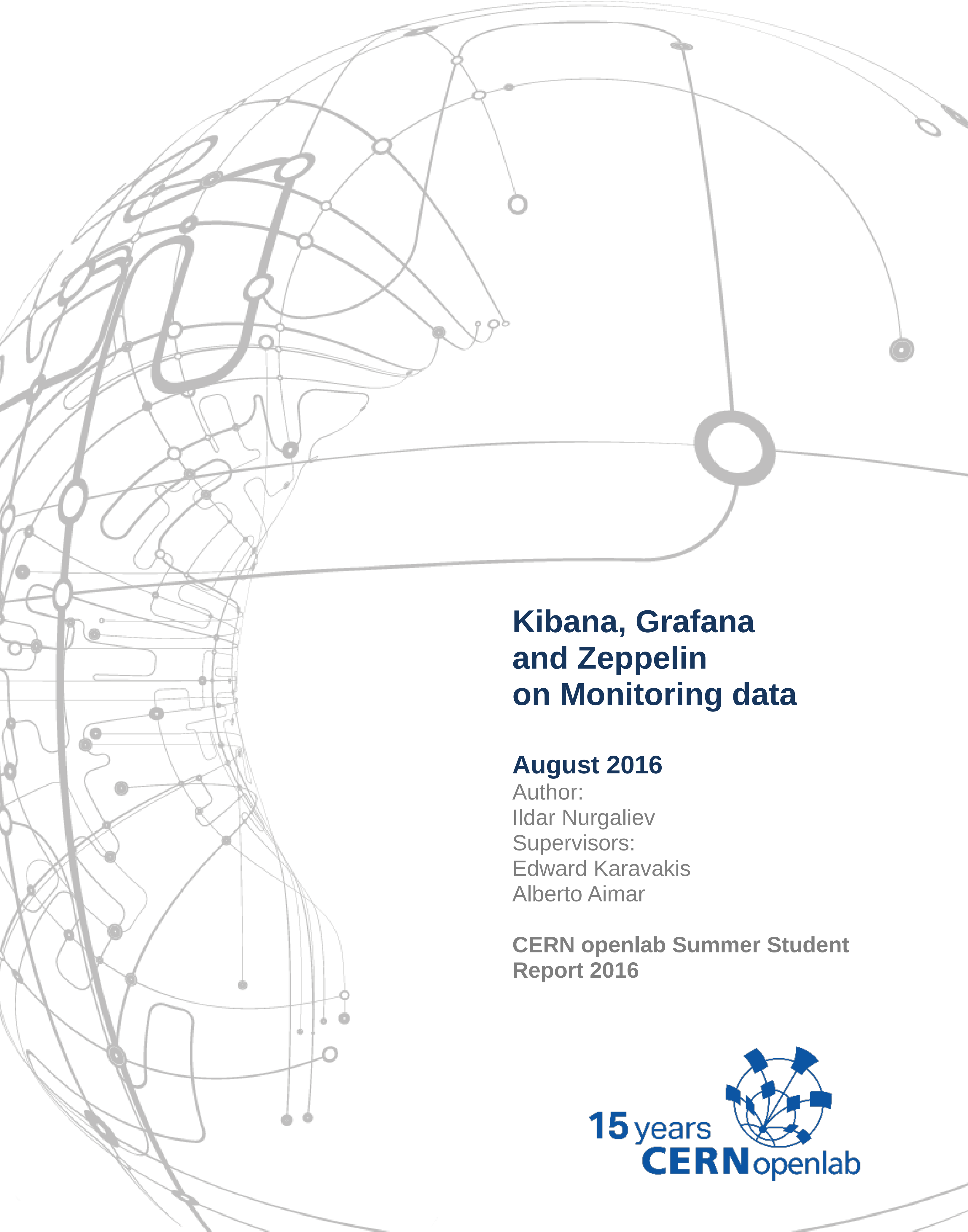
DOI: 10.5281/zenodo.61079

3 authors:

Ildar Nurgaliev
Innopolis University
6 PUBLICATIONS 114 CITATIONS
SEE PROFILE

Edward Karavakis
CERN
363 PUBLICATIONS 21,500 CITATIONS
SEE PROFILE

Alberto Aimar
CERN
21 PUBLICATIONS 162 CITATIONS
SEE PROFILE



Kibana, Grafana and Zeppelin on Monitoring data

August 2016

Author:
Ildar Nurgaliev
Supervisors:
Edward Karavakis
Alberto Aimar

CERN openlab Summer Student Report 2016

Project Specification

The goal of this project is to investigate different solutions and develop some typical monitoring displays, such as the general IT Department overviews and service-specific dashboards of data provided by the IT groups and stored in the IT monitoring repository (ElasticSearch). As a part of the project, the current IT host monitoring dashboard and the Data Centre Overview where migrated and ported from Kibana 3 to Grafana and Kibana 4 in order to study features, performance and improvements of these displays. The second part of the project is related to the report visualization of the ATLAS scrutiny report that was implemented in Apache Zeppelin. The last part of the project is related to the porting of the Apache PIG scripts to the Apache Spark computational framework in order to accelerate the generation of the aggregated summaries and to have direct Hadoop access that would be beneficial to any future project development in the report automation.

Abstract

The CERN/IT monitoring team handles every day millions of monitoring events from the CERN data centers and the WLCG sites. Today this data is provided to users through a number of different tools and dashboards. This project aims at exploring new visualization techniques to provide enhanced dashboards. Using the data being stored in ElasticSearch and HDFS this project relies on tools like Kibana, Grafana, and Zeppelin to develop new dashboards relevant for CERN and the LHC Experiments.

Table of Contents

1 Introduction 4

2 Relevant technologies 5

2.1 Elasticsearch 5

2.2 Kibana 4 5

2.3 Grafana 7

2.4 Kibana and Grafana fast comparing table 9

2.5 Zeppelin 11

2.6 Spark 11

3 Data Centre Overview in Kibana and Grafana 12

4 Porting FTS Monitoring to Grafana 14

5 Porting HOST metrics to Grafana 15

6 ACL and SSO support in Grafana for our server 16

7 Scrutiny report automatization by Zeppelin 17

8 Scrutiny report acceleration: replacing PIG scripts with Spark jobs 19

9 Conclusion 19

References 20

1 Introduction

CERN runs today two large data centers, one in Geneva and a second in Budapest, providing over 120,000 cores, 180 PB of disk space, 100 PB of tape space, and 150 high performance tape drives to serve CERN's computing and storage needs. This capacity is extended by the Worldwide LHC Computing Grid (WLCG), providing resources from different organizations around the globe, as CERN provides around 20% of the total WLCG resources. WLCG is a global collaboration of more than 170 computing centers in 42 countries, linking up national and international grid and cloud infrastructures. At any given time, there are typically more than 300,000 concurrent jobs running and, on a daily basis, more than two million jobs are submitted and terabytes of data are transferred between sites. The Monitoring team of the IT Compute and Monitoring (CM) group is responsible for providing monitoring solutions for these resources by handling millions of monitoring events from the CERN data centers and the WLCG sites on a daily basis.

Therefore, providing a reliable monitoring service is extremely important for the constant improvement and efficient use of the CERN data centers and the WLCG sites. The CERN/IT team has setup an infrastructure specifically designed to collect and display information about the nodes running in the data centers [1] and about the WLCG data transfers, computational jobs and the state of sites and services [2]. The idea behind is to facilitate the observation of the current and historical status and to give the opportunity to understand the state of the system from a glance view and at the same time allow to dig into detailed views in order to investigate problems.

A Dashboard’s primary purpose is to provide relevant and timely information to its audience. As a part of the project we conducted a detailed comparison between two Open Source visualization solutions: Kibana 4 and Grafana. This comparison will help to make strategic decisions for any future dashboard development or porting from an existing JavaScript implementation or from Kibana 3 that does not have support for the latest versions of Elasticsearch.

The first part of the report consists of an overview of the technologies that were used to store the data and to build a comprehensive dashboard. The second part of the report shows the migrated dashboard visualizations of the Data Center Overview, the FTS (File Transfer Service) and the HOST metrics and with the issues encountered.

The last part of the project is related to the report visualization of the ATLAS scrutiny report plots that was implemented in Apache Zeppelin and to the porting of the Apache PIG scripts to the Apache Spark computational framework in order to accelerate the generation of the aggregated summaries and to have direct Hadoop access that would be beneficial to any future project development in the report automation.

2 Relevant technologies

2.1 Elasticsearch

In order to display information about the state of the batch service, we use Elasticsearch, which is a distributed, highly available and scalable search server based on Lucene. It provides real-time search and analytic capabilities [3].

Elasticsearch is fully document oriented. Stores real world entities in Elasticsearch as structured JSON documents. All fields are indexed by default, and all the indices can be used in single query in fast way.

Due to this, ElasticSearch is an ideal candidate when you need to deal with huge amounts of time-series data and you can adapt your project to work with this kind of NoSQL document database. Particularly, they are focusing their attention in real time data, multitenancy and text search, bundled with the advantages of NoSQL document databases such as schema less, distributed computing and storage, high availability and fault tolerant.

In the case of the current project, ElasticSearch is used as a NoSQL document database and it is not used to perform queries on free text. Some studies were conducted before the start of this project [4] that were testing the performance of several technologies with the Experiment Dashboard monitoring data showing that ElasticSearch is the best approach to our current problem in terms of performance.

2.2 Kibana 4

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch [5]. Kibana is used to search, view, and interact with the data stored in the Elasticsearch indices. A user can easily perform advanced data analysis and visualize data in a variety of charts, tables, and maps. Kibana is used to quickly and easily visualize large volumes of data and its browser-based interface enables to quickly create and share dynamic dashboards that display changes to Elasticsearch queries in real time. No code and no additional infrastructure required.

Kibana enables near real-time analysis and visualization of streaming data. Allows interactive data exploration and supports cross-filtering. Having multiple chart types as bar chart, line and scatter plots, histograms, pie charts, maps. It is open-source and has a number of plug-in extensions that can further enhance its functionality.

Pros:

1. Kibana (ELK) is a popular stack for log storage and visualization with a big user community.
2. Rich dynamic visualization dashboards.
3. The visualizations, searches and dashboards are stored as objects, meaning that an object could be shared in multiple dashboards. That saves quite a lot of time as the user doesn't have to repeat common searches, visualizations and dashboards, the common ones could be reused.
4. A Kibana dashboard displays a set of saved visualizations in groups that the user can arrange freely. You can save a dashboard to share or reload at a later time.
5. The search field on the Discover page provides a way to query a specific subset of transactions from the selected time frame. The query syntax is based on the Lucene query syntax. It allows boolean operators, wildcards, and field filtering. Moreover after search it is possible to see the document structure and even filter on some fields.
6. Kibana 4 allows users to create scripts and plant them within the Elasticsearch server. Once the scripts are planted, their returned results can be viewed in real time as if they were data arriving directly from the logs.
7. Good documentation and multiple examples are available online.

Cons:

1. Area plot looks a little bit not-refined in most of the cases. As a result, it is better to change it to a bar plot.
2. There is no way to migrate dashboards from Kibana3 to Kibana 4, thus you need rebuild them from scratch.
3. No Customizable axis labels, i.e. instead of 5000000, a user cannot show 5mi or put measurements in Y-axis (s, ms, terabytes) as it was possible in Kibana 3.
    - Issue tracking https://github.com/elastic/kibana/issues/2386
4. No option to hide a column in a table view.
    - Issue tracking https://github.com/elastic/kibana/issues/4307

Visualizations support:

Area chart	Used to visualize the total contribution of several different series
Data table	Used to display the raw data of a composed aggregation
Line chart	Used to compare different series
Markdown widget	Used to display free-form information or instructions about the dashboard
Metric	Used to display a single number on a dashboard
Pie chart	Used to display each source's contribution to a total

Tile map	Used to associate the results of an aggregation with geographic points
Vertical bar chart	Used as a general-purpose chart
Heat map	Heat-map Plugin for Kibana

Additional visualization and non-visualization options:

Combined plots **(-)**	It is not possible to create plots from two different sources in the same view. Also there is no support for combined types in a plot, i.e. a bar plot with the inclusion of lines as well
Plotting derived fields **(+)**	Scripted fields option and by using the Timelion additional plugin
Exporting plots and sharing dashboards **(+-)**	You can use the Share button on Kibana4 to get an iframe link you can embed, but there aren't any other options at this time. The data of a visualization could be directly downloaded in the CSV and JSON format
ACL (security) **(-)**	Commercial plug-in, no support in the free version

2.3 Grafana

Grafana's support for the Elasticsearch data source is relatively new, so there are some rough edges and not all functions that Elasticsearch provides are currently available in the editor, for example some aggregations. Grafana offers quite a lot of refined looking charts [6].

In Grafana there is no concept of 'reusable objects' to share between dashboards although that there are ways to do that. For example we want to create a common menu for several dashboards, and while the markdown visualization is not shareable, we can create an iframe and then share it. In the same time, Grafana offers an inbuilt menu visualization by using the 'tag search' function.

Pros:

1. Good support of Grafana having an active and big community.

2. Rich dynamic visualization dashboards, with additional features with the templated variables that could be used for data sources, variables as a search parameter, or time widths.
3. Inbuilt menu that could construct navigation to dashboards by tag search.
4. Lots of graph styling options.
5. Direct link to render images for every visualization in a dashboard, so it is very easy to share a report.
6. Customizable Axes that are very useful and it frees from additional work for data transformation for example from bytes to megabytes, we just mention that for the Y-axis we want a representation of the data size in bytes and it automatically transforms it in the right visualization.
7. A dashboard displays a set of saved visualizations in groups that you can arrange freely. You can save a dashboard to share or reload at a later time.
8. Good documentation and examples online.

Cons:

1. There is no appropriate heat map for plotting, no double matrix table, just aggregator by time.
2. If you add annotation events to a dashboard, the dashboard starts sometimes to slow down.
3. Has limited grid template and plot alignments unlike Kibana 4.
4. No way for nominal data to be on the X-axis of a bar chart as Grafana is only for time series. A Pie chart plugin extension could be used instead to visualize the data in a pie chart.
5. Heat maps are not supported in a way of nominal matrix.

Visualizations support:

Area chart	Visualize the total contribution of several different series
Data table	Display the raw data of a composed aggregation
Line chart	Compare different series
Markdown widget	Display free-form information or instructions about a dashboard
Singlestat	Used to display a single number on a dashboard
Pie chart	Pie chart plugin
Geo map	Used to associate the results of an aggregation with geographic points
Vertical bar chart	General-purpose charts
Heat map	EpochHeat Map Plugin for Grafana, no way to make nominal general

	information, this plugin is for time series data

Additional visualization and non-visualization **options**:

Combined plots **(+)**	It is possible to create plots from two different sources in the same view. Also there is support for combined types in a plot (as bar plot with inclusion of lines as well)
In built menu **(+)**	Grafana offers an inbuilt menu visualization by using the 'tag search' function
Plotting derived fields **(+)**	With scripted fields
Exporting plots and sharing dashboards **(+)**	PNG, CSV, url-render of a plot as an image
ACL (security) **(+)**	Out of the box by supporting "Organizations"

2.4 Kibana and Grafana fast comparing table

Feature	Grafana	kibana
Aspect	Plots **look refined** without banners or edit button Expandable visualizations	Not very refined
Search and exploring	Lucene **query for visualization** By dashboard developer	'Data exploration' Show **document structure** Inbuilt search highlights **Save search** as object By dashboard user

Objects reuse	**No** visualization **reusage** **Manual repeat** search query	Visualization, Dashboards, Search are saved as objects (**pluggable into many dashboards**)
General plots	All is **Time-series** based	Time series Nominal axis Comprehensive Heat Map
Export & Share	Visualization -> CSV, PNG, Render image url	Visualization -> CSV, JSON
Role-based access (RBA)	RBA/ACL supported by default as “Organizations”	No built-in RBA Commercial plug-in
Plotting derived fields	**Yes** Scripted fields	**Yes** Scripted fields, TimeLion
Combined plots	**Yes** very flexible visualization	**No**
Plots from 2 different source	**Yes** As many as you wish (ES, Graphite, Influx Db)	**No** One ES source only
Templated variable	**Yes** for search fields, time window	**No**
Templated Repeated visualizations	**Yes** Auto Generate visualizations from limited set of values for a var	**No**

Templated data source	**Yes** Possible to change data source on fly	**No**
Menu	**YES** Generate menu from tag search	**No** Markdown visualization as solution

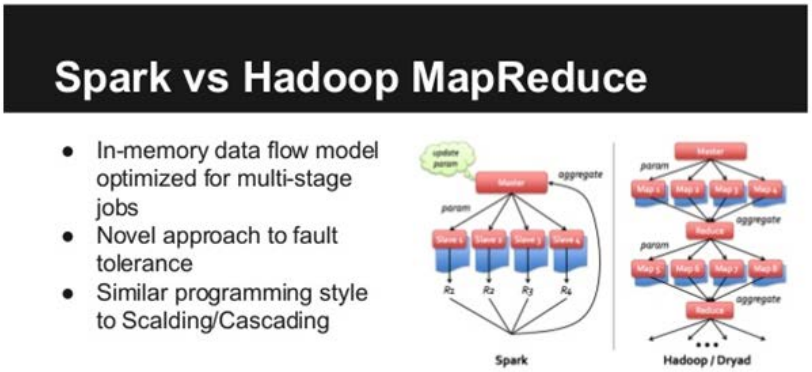
2.5 Zeppelin

Apache Zeppelin is a web-based notebook that enables interactive data analytics with powerful dynamic visualizations and it supports several technologies out of the box (Python, SQL, Spark, Hive, Markdown, Shell, etc.) with the concept of an interpreter that allows any language, data-processing-backend to be plugged into Zeppelin [7]. A Zeppelin Notebook URL can be shared among people. In a shared Zeppelin notebook the changes are seen by all the users in real-time, just like in Google docs. For public publishing, Zeppelin provides an option to display the result only, without the inclusion of Zeppelin's menu and buttons. This way, it can be easily embedded anywhere as an iframe.

2.6 Spark

Apache Spark is an open-source framework for the distributed processing of unstructured and semi-structured data, and it is part of the Hadoop ecosystem projects [8]. Unlike the classic processing from the Hadoop core that implements a two-level concept of MapReduce with disk storage, Spark uses specialized primitives to recursively process data in memory, allowing to obtain a significant gain in the speed work for some classes of problems. In particular, the possibility to access data loaded in the memory multiple times is very useful for data analysis related tasks.

The project provides APIs for the Java language, Scala, Python and R. Spark consists of the Spark Core and a number of extensions, such as Spark SQL (allows you to execute SQL-queries on the data), Spark MLlib (set of libraries machine learning) and etc. It can work both in the Hadoop cluster environment running YARN, and without the core components of Hadoop. Spark supports multiple distributed storage systems - HDFS, OpenStack Swift, NoSQL-databases such as Cassandra, Amazon S3.



Spark vs Hadoop MapReduce

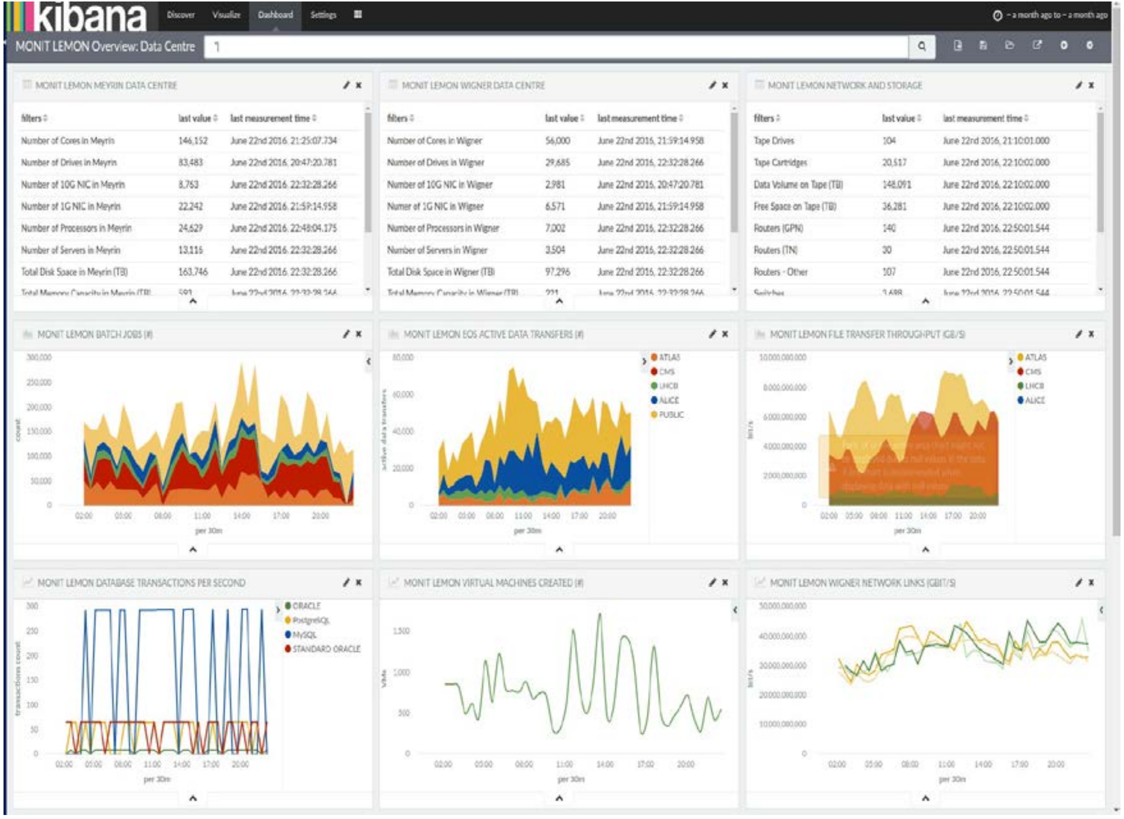
- In-memory data flow model optimized for multi-stage jobs
- Novel approach to fault tolerance
- Similar programming style to Scalding/Cascading

Figure 1. Comparison of Map Reduce and Spark RDD computation paradigms.

3 Data Centre Overview in Kibana and Grafana

The following information is available in the Data Centre Overview dashboard:

- Meyrin and Wigner data center overview
- Network and storage overview
- Batch jobs count plotting
- Active data transfers rate
- File transfer throughput GB/s
- Database transactions per second
- Count of virtual machines created
- WIGNER network overview plotting



kibana
Discover
Visualize
Dashboard
Settings
~ a month ago to ~ a month ago
MONIT LEMON Overview: Data Centre
1

MONIT LEMON MEYRIN DATA CENTRE

filters	last value	last measurement time
Number of Cores in Meyrin	146,152	June 22nd 2016, 21:25:07.734
Number of Drives in Meyrin	83,483	June 22nd 2016, 20:47:20.781
Number of 10G NIC in Meyrin	8,763	June 22nd 2016, 22:32:28.266
Number of 1G NIC in Meyrin	22,242	June 22nd 2016, 21:59:14.958
Number of Processors in Meyrin	24,629	June 22nd 2016, 22:48:04.175
Number of Servers in Meyrin	13,116	June 22nd 2016, 22:32:28.266
Total Disk Space in Meyrin (TB)	163,746	June 22nd 2016, 22:32:28.266
Total Memory Capacity in Meyrin (TB)	593	June 22nd 2016, 22:32:28.266

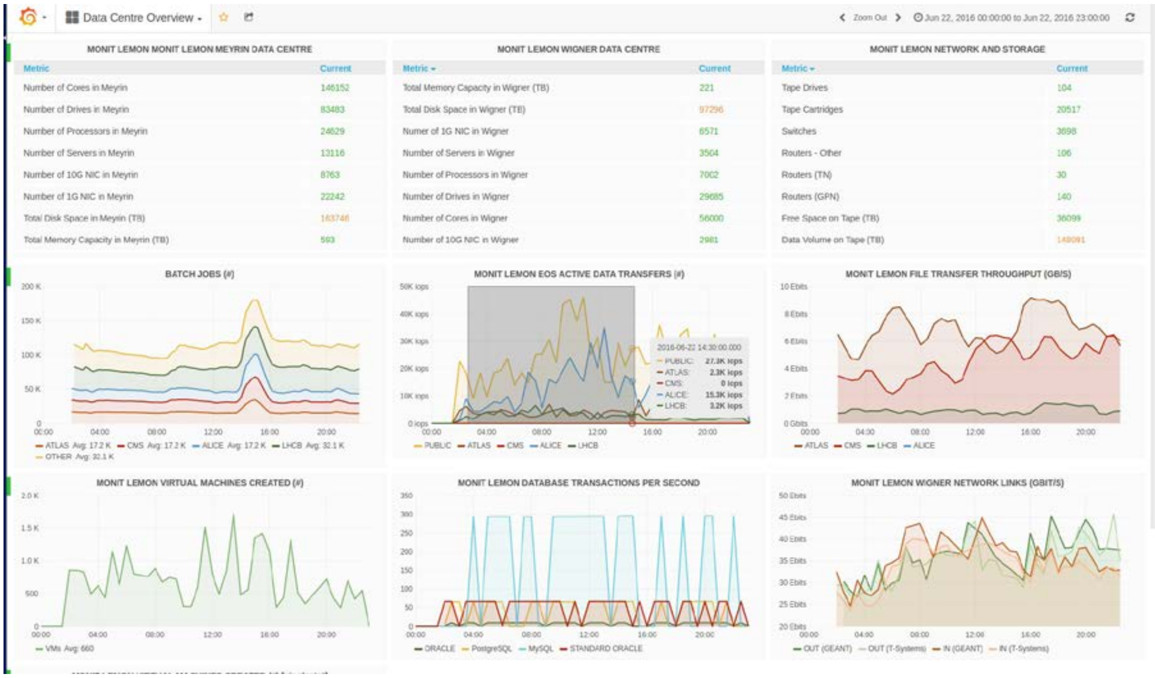
MONIT LEMON WIGNER DATA CENTRE

filters	last value	last measurement time
Number of Cores in Wigner	56,000	June 22nd 2016, 21:59:14.958
Number of Drives in Wigner	29,685	June 22nd 2016, 22:32:28.266
Number of 10G NIC in Wigner	2,981	June 22nd 2016, 20:47:20.781
Numer of 1G NIC in Wigner	6,571	June 22nd 2016, 21:59:14.958
Number of Processors in Wigner	7,002	June 22nd 2016, 22:32:28.266
Number of Servers in Wigner	3,504	June 22nd 2016, 22:32:28.266
Total Disk Space in Wigner (TB)	97,296	June 22nd 2016, 22:32:28.266
Total Memory Capacity in Wigner (TB)	221	June 22nd 2016, 22:32:28.266

MONIT LEMON NETWORK AND STORAGE

filters	last value	last measurement time
Tape Drives	104	June 22nd 2016, 21:10:01.000
Tape Cartridges	20,517	June 22nd 2016, 22:10:02.000
Data Volume on Tape (TB)	148,091	June 22nd 2016, 22:10:02.000
Free Space on Tape (TB)	36,281	June 22nd 2016, 22:10:02.000
Routers (GPN)	140	June 22nd 2016, 22:50:01.544
Routers (TN)	30	June 22nd 2016, 22:50:01.544
Routers - Other	107	June 22nd 2016, 22:50:01.544
Switches	3,698	June 22nd 2016, 22:50:01.544

Figure 2. Data Center Overview reproduced in Kibana 4



MONIT LEMON MONIT LEMON MEYRIN DATA CENTRE

Metric	Current
Number of Cores in Meyrin	146152
Number of Drives in Meyrin	83483
Number of Processors in Meyrin	24629
Number of Servers in Meyrin	13116
Number of 10G NIC in Meyrin	8763
Number of 1G NIC in Meyrin	22242
Total Disk Space in Meyrin (TB)	163746
Total Memory Capacity in Meyrin (TB)	593

MONIT LEMON WIGNER DATA CENTRE

Metric	Current
Total Memory Capacity in Wigner (TB)	221
Total Disk Space in Wigner (TB)	97296
Numer of 1G NIC in Wigner	6571
Number of Servers in Wigner	3504
Number of Processors in Wigner	7002
Number of Drives in Wigner	29685
Number of Cores in Wigner	56000
Number of 10G NIC in Wigner	2981

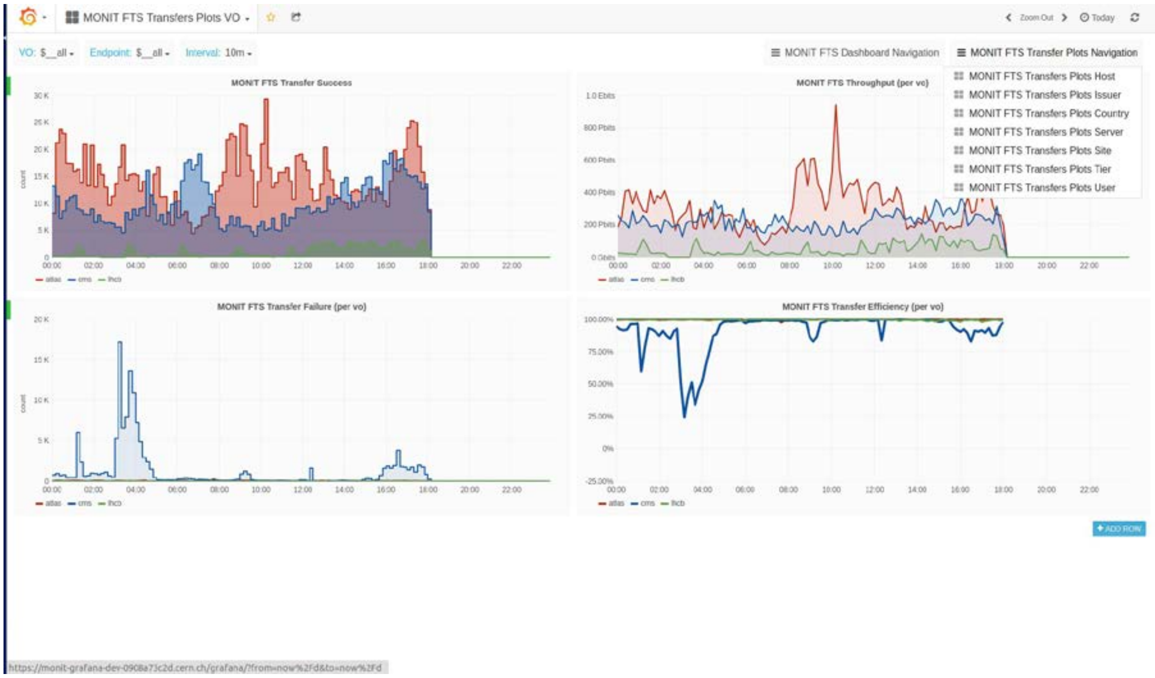
MONIT LEMON NETWORK AND STORAGE

Metric	Current
Tape Drives	104
Tape Cartridges	20517
Switches	3698
Routers - Other	106
Routers (TN)	30
Routers (GPN)	140
Free Space on Tape (TB)	36099
Data Volume on Tape (TB)	148091

Figure 3. Data Center Overview reproduced in Grafana

4 Porting FTS Monitoring to Grafana

File Transfer Monitoring dashboards were fully reproduced in Grafana. The implementation in Grafana, there is more space especially for the plots. The main menu, context menu, filter by VO (Virtual Organizations) and filter by host were moved to the upper part. As shown in the dashboard below, the filtering is represented by template variables on the left up corner, moreover there is an additional time template that allows us to dynamically change the time window from 10 minutes to 30 minutes and to 1 hour. The menu is auto generated from the tag search: the bunch of dashboards have the same tag, thus searching by tag gives us the current menu.



VO: $__all
Endpoint: $__all
Interval: 10m

https://monit-grafana-dev-0908a73c2d.cern.ch/grafana/?from=now%2Fd&to=now%2Fd

Figure 4. FTS Transfer Plots VO (Virtual organizations)

The list of reproduced dashboards is presented below:

- FTS Transfer Plots
  - Transfer by VO (Virtual Organization)
  - Transfer by Server
  - Transfer by Host
  - Transfer by Site
  - Transfer by Country
  - Transfer by Tier
  - Transfer by User
  - Transfer by Issuer
- FTS Ranking Plots
  - Ranking by VO
  - Ranking by Server

  - ○ Ranking by Country
  - ○ Ranking by Site
  - ○ Ranking by Tier
- Correlated Plots
- Queue Monitoring Plots

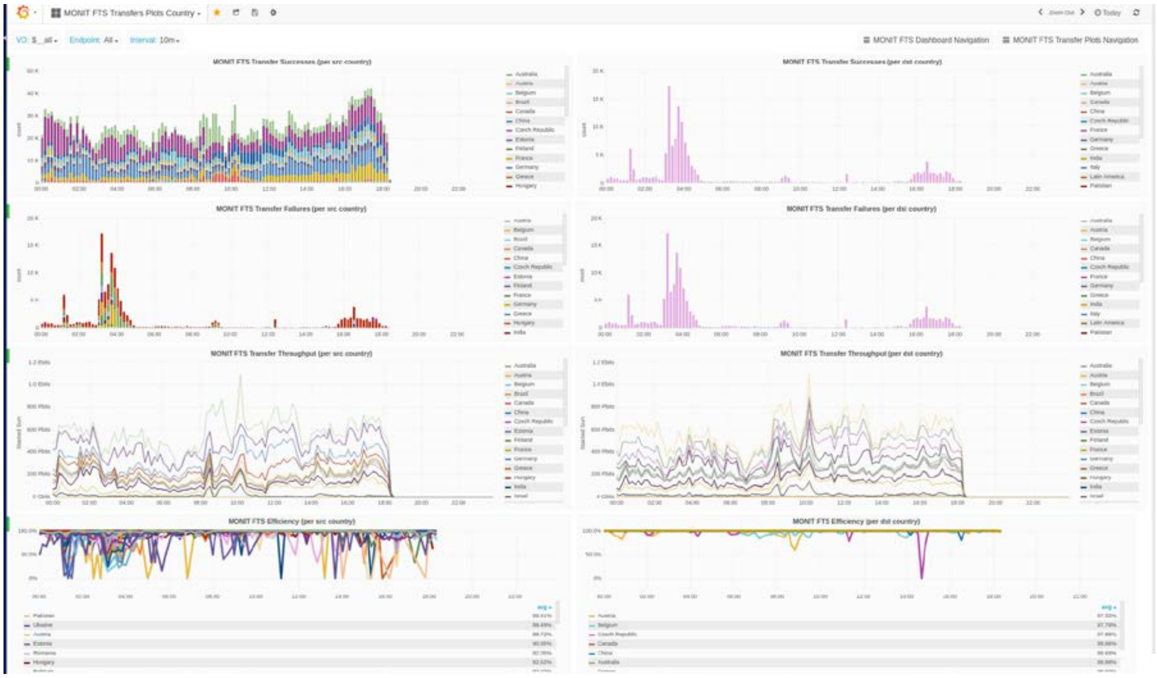

MONIT FTS Transfers Plots Country
Zoom Out
Today
VO: $_all
Endpoint: All
Interval: 10m
MONIT FTS Dashboard Navigation
MONIT FTS Transfer Plots Navigation
MONIT FTS Transfer Successes (per src country)
MONIT FTS Transfer Successes (per dst country)
MONIT FTS Transfer Failures (per src country)
MONIT FTS Transfer Failures (per dst country)
MONIT FTS Transfer Throughput (per src country)
MONIT FTS Transfer Throughput (per dst country)
MONIT FTS Efficiency (per src country)
MONIT FTS Efficiency (per dst country)
count
Stacked Sum
Australia
Austria
Belgium
Brazil
Canada
China
Czech Republic
Estonia
Finland
France
Germany
Greece
Hungary
India
Israel
Italy
Latin America
Pakistan
Ukraine
avg
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
100.0%
50.0%
0%

Figure 5. FTS reproduced in Grafana

5 Porting HOST metrics to Grafana

HOST metrics dashboard has some collapsed groups of visualizations that mostly represent the group of the same visualization with the same visualization parameters, the only difference is the different variable used in the search request. For this, we have templated all the possible values of a requested variable in order to automatically generate the visualization after uncollapsing a group.

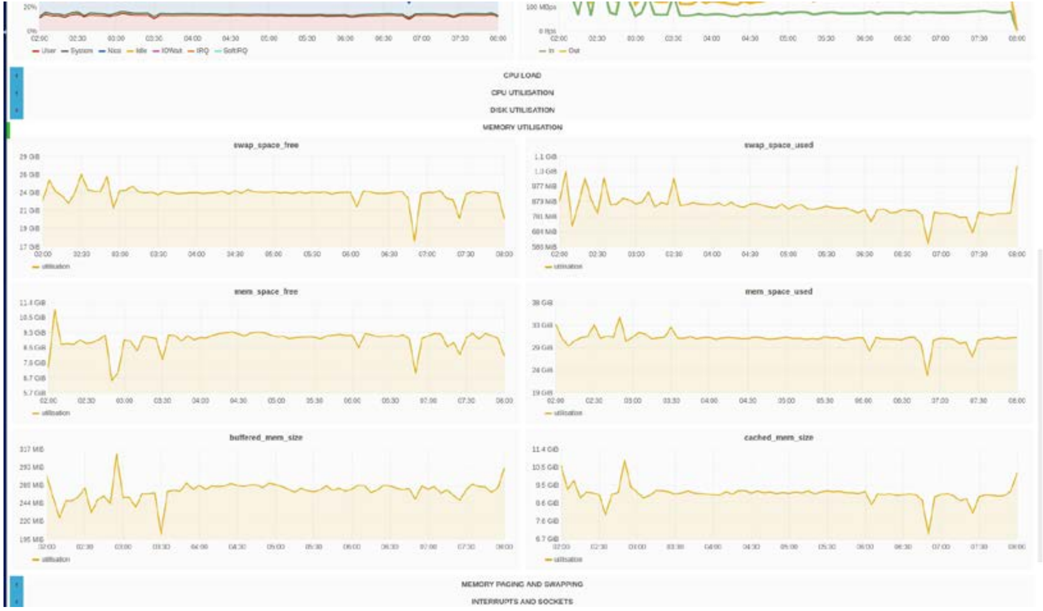


Figure 6. Templated Visualization in action

List of visualization groups available:

- MONIT LEMON Utilization
- CPU Load
- CPU utilization
- Disk utilization
- Memory utilization
- Memory paging and swapping
- Interrupts and sockets
- Processes
- Number of users
- Host details (single stat for **text info** was replaced by a one row table)
- Raw data

6 ACL and SSO support in Grafana for our server

Single Sign-On is a technology that allows a user to move from one section of the portal to another one without authenticating once again. For example, if there are several large independent sections (forum, chat, blog and so on). After successfully passing the authentication process in one of the services, the user automatically gets access to all the rest.

For this purpose we decided to use SSO in front of Grafana by using Shibboleth, an open-source project that provides Single Sign-On capabilities allowing sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner.

Grafana supports Access Control Lists (ACLs) and the definition of ACL rules in the dashboards, i.e. this user can only read this dashboard, this user can read and write and etc. This task consists of understanding how to ACLs work in Grafana, checking Grafana's support for 'single sign on' (sso) and LDAP, and to define some ACL rules in a couple of accounts.

Nested organizations/sub-groups are unsupported in Grafana. Since dashboards, data sources and other configuration items are not shared between organizations, there's no need to create multiple Organizations if you want all your users to have access to the same set of dashboards and data. We could just copy dashboards to new organizations, that is complicated to maintain. Actually users could be assigned to different organizations, so the problem is avoided. A User from different organization can not view any dashboards of others while they are not subscribed in the same organization. For our test-bed, three organizations were created: MONIT - default for all newcomers that get the Viewer permission [contains all the dashboards now], ATLAS and CMS.

By incorporating Shibboleth/SSO in Grafana, new users with a valid CERN account are automatically mapped to a Grafana user and thus, eliminating the need to create a new Grafana account. The default assignment group is MONIT with Viewer permissions. As a future work we could integrate Puppet to automatically deploy and configure a Grafana instance and to have an automated way to copy Grafana data sources and dashboards.

7 Scrutiny report automatization by Zeppelin

ATLAS is intensively using the WLCG resources to store and analyze data coming out of the detector. They are all working to optimize the resource usage by diminishing the derived data formats that are not used and by reducing the number of data replicas in the tiers thanks to the availability of fast networks connecting the WLCG sites. As a result of this optimization, they produce a data popularity plot where we could see the number of accesses in a given time period [9].

For this project, we studied the existing reports based on datasets popularity and unused data in ATLAS. This reports were done with Cron Jobs running some PIG scripts that aggregate the ATLAS datasets access events from Hadoop Summaries and were storing them on a web server as CSV files. Afterwards, the CSV files were manually downloaded and imported in an Excel spreadsheet. For this part, we automated the manual intervention so that the user can select the data and get the required plots that are used in the report directly within Zeppelin. The figure below shows the number of accesses in a given time period. We are interested in particular in first two bins that contain data with zero access over a given time period.

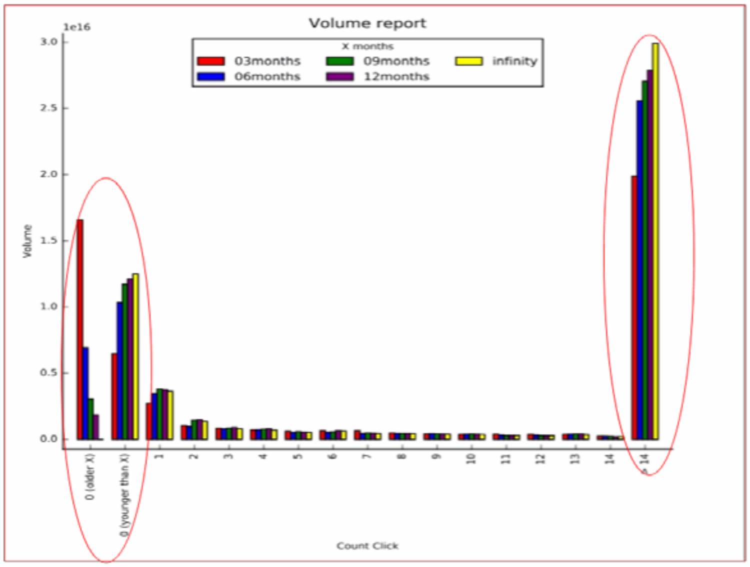
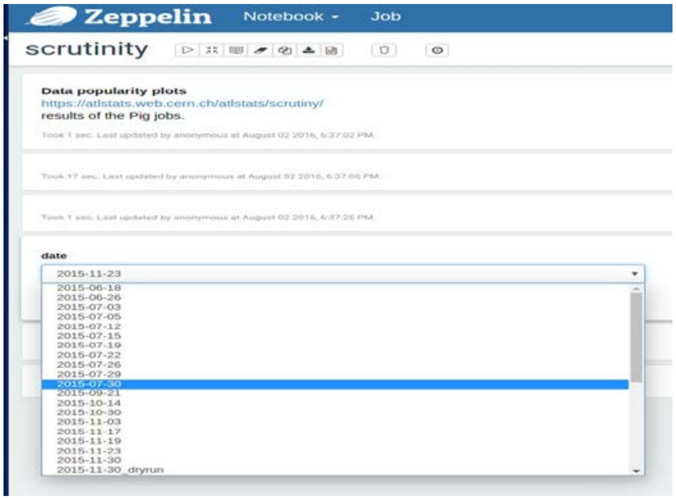


Figure 7. Volume of data versus number of accesses in 3-, 6- ,12-month periods and infinity for ATLAS. For each period X, data created in that period but not accessed is in the second bin. The first bin is for data created before the period began and not accessed during that period.



Data popularity plots
https://atlstats.web.cern.ch/atlstats/scrutiny/
results of the Pig jobs.

Took 1 sec. Last updated by anonymous at August 02 2016, 6:37:02 PM.

Took 17 sec. Last updated by anonymous at August 02 2016, 6:37:06 PM.

Took 1 sec. Last updated by anonymous at August 02 2016, 6:37:25 PM.

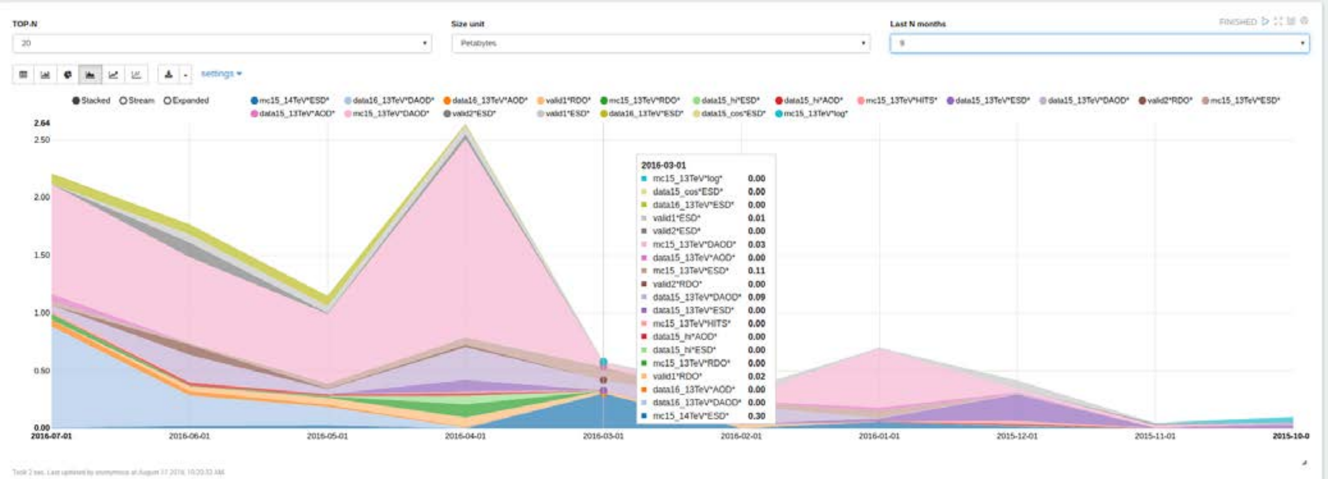
date

2015-11-23

```
2015-06-18
2015-06-26
2015-07-03
2015-07-05
2015-07-12
2015-07-15
2015-07-19
2015-07-22
2015-07-26
2015-07-29
2015-07-30
2015-09-21
2015-10-14
2015-10-30
2015-11-03
2015-11-17
2015-11-19
2015-11-23
2015-11-30
2015-11-30_dryrun
```

Figure 8. A user selects a date and the automated report-file search and load occurs automatically.

We have also automated the data popularity reports by TOP-N unused data within every month, the number of which is also could be dynamical chosen. The current excel report shows just TOP-20 within last 9 months. Our dynamic implementation allows the user to choose the date, get the report and within the report choose the TOP-N, last N months, Represent in Petabytes or Terabytes, select TOP-**N** from last **M** months showing in last **K** months where **K <= M**.



TOP-N: 20

Size unit: Petabytes

Last N months: 9

● Stacked ○ Stream ○ Expanded

2016-03-01	
mc15_13TeV*log*	0.00
data15_cos*ESD*	0.00
data16_13TeV*ESD*	0.00
valid1*ESD*	0.01
valid2*ESD*	0.00
mc15_13TeV*DAOD*	0.03
data15_13TeV*AOD*	0.00
mc15_13TeV*ESD*	0.11
valid2*RDO*	0.00
data15_13TeV*DAOD*	0.09
data15_13TeV*ESD*	0.00
mc15_13TeV*HITS*	0.00
data15_hi*AOD*	0.00
data15_hi*ESD*	0.00
mc15_13TeV*RDO*	0.00
valid1*RDO*	0.02
data16_13TeV*AOD*	0.00
data16_13TeV*DAOD*	0.00
mc15_14TeV*ESD*	0.30

Figure 9. Scrutiny report by date: select **TOP-20** in **Petabytes** slice of the **last 9 months,** with list of dataset names and actual occupied volume of unassessed data.

8 Scrutiny report acceleration: replacing PIG scripts with Spark jobs

The CSV files used are generated from RUCIO traces by the scripts presented under the github storage: https://github.com/ATLAS-Analytics/scrutiny. They produce detailed statistics on unused data by project and datatype by aggregating Rucio traces using Rucio contents dumps (datasets information, storage elements, etc.). The current data aggregator is implemented in various Apache PIG scripts that use the Map Reduce computation paradigm. Nowadays Map Reduce is considered as insufficient with the immediate availability of big RAM and CPU capabilities. The main purpose of porting the PIG scripts to Spark scripts is to reduce the disk access by replacing Map Reduce with the Spark RDD computation paradigm. As a result we have accelerated the generation of the CSV files.

9 Conclusion

The monitoring team handles every day millions of monitoring events from the CERN data centers and the WLCG sites. Today this data is provided to users through a number of different tools and dashboards. The aim of this contribution was to explore existing Open Source visualization techniques in order to provide enhanced dashboards. Two of the most comprehensive and complete dashboard solutions are Kibana and Grafana. We have performed the comparison between them and have also ported some production dashboards from Kibana 3 to Kibana 4 and to Grafana in order to study features, performance and improvements of these displays.

The second contribution of the project is related to the report visualization of the ATLAS scrutiny report that required manual intervention to produce the plots. After this contribution that was implemented in Apache Zeppelin, everything is fully automated thus, eliminating the need for manual intervention.

The last contribution of the project is related to the porting of the Apache PIG scripts to the Apache Spark computational framework in order to accelerate the generation of the aggregated summaries and to have direct Hadoop access that would be beneficial to any future project development in the report automation.

While contributing to the above projects within the OpenLab summer internship, I have studied many technologies such as Elasticsearch, Kibana 3 & 4, Grafana, Apache Zeppelin notebook, Apache Hadoop, Apache PIG and the Spark framework. It will have a very big positive impact in my future data science career. Thank you CERN!

References

[1] P Andrade et al, “Agile Infrastructure Monitoring”, Phys.: Conf. Ser. 513, 062001 doi:10.1088/1742-6596/513/6/062001/

[2] J Andreeva et al, “Experiment Dashboard-a generic, scalable solution for monitoring of the LHC computing activities, distributed sites and services”, 2012 J. Phys.: Conf. Ser. 396 032093 doi:10.1088/1742-6596/396/3/032093

[3] "Elastic · Revealing Insights from Data (Formerly Elasticsearch) | Elastic." 2011. 15 Aug. 2016 <http://www.elastic.co/>

[4] J Andreeva et al, “Processing of the WLCG monitoring data using NoSQL”, 2014 J. Phys.: Conf. Ser. 513 032048 doi:10.1088/1742-6596/513/3/032048

[5] "Kibana 4. Literally. | Elastic." 2015. 15 Aug. 2016 <http://www.elastic.co/blog/kibana-4-literally>

[6] "Grafana - Beautiful Metrics, Analytics, dashboards and monitoring!." 2014. 15 Aug. 2016 <http://grafana.org/>

[7] "Zeppelin." 2015. 15 Aug. 2016 <https://zeppelin.apache.org/>

[8] "Apache Spark™ - Lightning-Fast Cluster Computing." 2014. 15 Aug. 2016 <http://spark.apache.org/>

[9] UK, C Allton et al. "Computing Resources Scrutiny Group." (2016).