Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

Evaluation of visualization software for CMS storage usage monitoring

Alessandro Lazzari **Supervisor: Natalia Ratnikova**

Final presentation

Sep 21, 2016

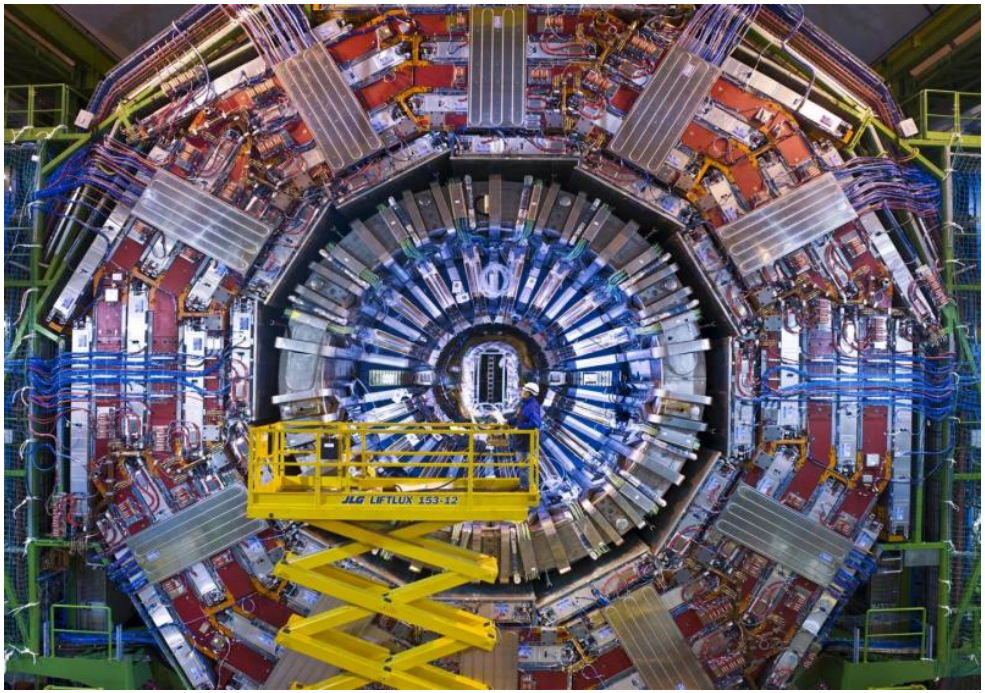
OUTLINE

- Introduction to CMS space monitoring
- Goals of visualization
- My contribution
  - Populated data source for visualization and checked data quality
  - Compared features of visualization tools
  - Implemented visualization examples on CMS monitoring data
  - Documentation of the steps above
- Summary

INTRODUCTION TO CMS

The Compact Muon Solenoid (CMS) is a particle detector, operating at the Large Hadron Collider (LHC) at CERN, that was conceived to observe particles and phenomena produced in high-energy collisions. The range of physics investigated by the CMS experiment include extra dimensions, the search for the Higgs boson and particles that could make up dark matter.

CMS **physics data** are produced at **high rates** and then **distributed** to many **sites** for storage. Multiple users and systems will subsequently make use of these data according to their purposes.

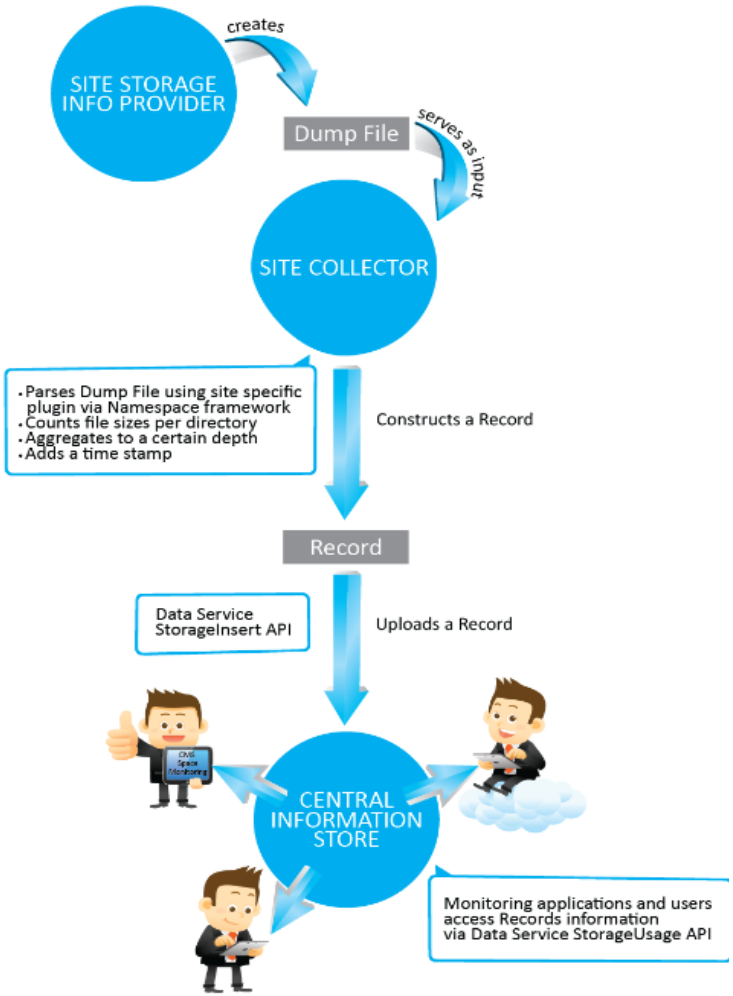


https://home.cern/about/experiments/cms

INTRODUCTION TO CMS SPACE MONITORING

A CMS space monitoring system, called SpaceMon, was designed and developed, in order to provide an overall view of the distributed storage based on sites local storage information. That information is aggregated locally at the sites, then uploaded into a central database at CERN.

The data space is organized according to CMS data namespace conventions and a web data service provides interfaces to the oracle database where space usage monitoring results are stored.



From Ratnikova N et al. 2014 CMS Space Monitoring J. Phys.: Conf. Ser. 513 042036

GOALS OF VISUALIZATION

The main goal of visualization, with respect to the SpaceMon infrastructure, is to represent space monitoring information in a convenient form, according to the needs of CMS computing management as well as site administrators and individual storage users (use cases).

GOALS OF MY TASK

My job consisted in exploring the functionalities of a predefined set of visualization tools, namely Elasticsearch, Grafana and Kibana, and test these tools on real use cases.

SOFTWARE TOOLS

Elasticsearch

Highly scalable open-source full-text search and analytics engine.

- Allows you to store, search, and analyze big volumes of data quickly and in near real time
- Generally used as the underlying engine/technology that powers applications that have complex search features and requirements

→ Used to store monitoring data

Kibana

Open source analytics and visualization platform.

- Allows you to search, view, and interact with data stored in Elasticsearch
- Allows you to perform advanced data analysis and visualize your data in a variety of charts, tables, and maps

→ Used to visualize monitoring data

Grafana

Open source analytics and visualization platform.

- Works with Graphite, Elasticsearch and other data sources
- Allows you to perform advanced data analysis and visualize your data in a variety of charts, tables, and maps

→ Used to visualize monitoring data

ELASTICSEARCH

https://www.elastic.co/products/elasticsearch

- Data types
  - Core: string, numeric, date, *boolean, binary*
  - Geo datatypes
  - *Complex: array, json object*
- Populated ES with CMS monitoring data:
  - Retrieving the data (i.e. 'records') from the data service
  - Converting the data into an ES suitable format
    - script: comments, examples, git versioning
  - Loading the data into ES
    - using either ES python library or 'curl' command
    - 2'225'403 records belonging to the period May-July 2016
- Described additional data service API for ES
  - such as to avoid data conversion step and facilitate ES data loading

Note: in the end the 'curl' method, instead of the python library, was chosen to load the data in ES, since that turned out to be faster.

E.g.: 2.9s vs 1min52s (user+sys time) to load 100k records

ES MAPPING

The mapping specifies the format of the records to load in ES, including the field data types.

```
{
     "mappings" : {
          "_default_" : {
               "properties" : {
                    "timestamp": {"type": "date", "format": "epoch_second"},
                    "name": {"type": "string", "index" : "not_analyzed" },
                    "space":  {"type": "long"},
                    "rlvl": {"type":"integer"},
                    "dir": {"type": "string", "index" : "not_analyzed" }
                }
           }
     }
}
```

DATA SERVICE API DESCRIPTION (1)

```
NAME
DMWMMON::Web::API::totstor – Total site storage.

DESCRIPTION
Weekly-sync total site storage. Output formatted for ES.

OPTIONS
 node            node name, could be multiple, all(T*), required
---options from DumpSpaceQuery
 time_since      former time range, since this time, if not specified, time_since=0
 time_until      later time range, until this time
                 if not specified, time_until=10000000000
                 if both time_since and time_until are not specified,
                 the latest record will be selected

OUTPUT
For every week, a series of 'records' (one per site) reporting their total storage
With same timestamps.
```

DATA SERVICE API DESCRIPTION (2)

EXAMPLE

Input:

```
...?node=T1_*
```

Output:

```
{"index":{"_type":"sync-
weekly","_id":"4d8245c97c93890b0f04118dcbc1c6f0","_index":"test-sync"}}
{"timestamp":1472204564,"name":"T1_AAA","space":56418073}
{"index":{"_type":"sync-
weekly","_id":"4d8245c97c93890b0f04118dcbc1c6f1","_index":"test-sync"}}
{"timestamp":1472204564,"name":"T1_BBB","space":26418073}
{"index":{"_type":"sync-
weekly","_id":"4d8245c97c93890b0f04118dcbc1c6f2","_index":"test-sync"}}
{"timestamp":1472204564,"name":"T1_CCC","space":16418073}
...
```

DATA SERVICE FINDINGS

While analyzing the data returned by the data service, some findings I reported, e.g. undefined values or inconsistent formats, were investigated and understood by the experts. That will eventually help to improve the robustness of the system.

Inconsistencies found in monitoring data from May, 2016 to July, 2016.

Site	Expected PFN for «/store/»	Actual PFN for «/store/»
T2_US_Florida	/cms/data/store/	/store/
T2_KR_KNU	/pnfs/knu.ac.kr/data/cms/store/	/store/
T1_DE_KIT_Disk	/grid/fzk.de/mounts/pnfs/cms2/disk-only/store/	/pnfs/gridka.de/cms/disk-only/store/
T2_HU_Budapest	/store/	/dpm/kfki.hu/home/cms/phedex/store/
T2_FR_IPHC	/dpm/in2p3.fr/home/cms/phedex/store/	/dpm/in2p3.fr/home/cmsphedex/store/

A way to fix such inconsistencies, is preprocessing the records in a script prior to load them into ES, adding an exception for each one of them.

EVALUATION OF KIBANA VS GRAFANA

https://www.elastic.co/products/kibana

http://grafana.org/

- Explored their functionalities
  - using: docs, tutorial examples and real CMS monitoring data
  - several kinds of diagrams
  - capability of assembling a dashboard containing multiple visualizations
  - dashboard sharing options
- Created visualizations with CMS monitoring data
- Presented dashboards in developers' meetings

EVALUATION TABLE

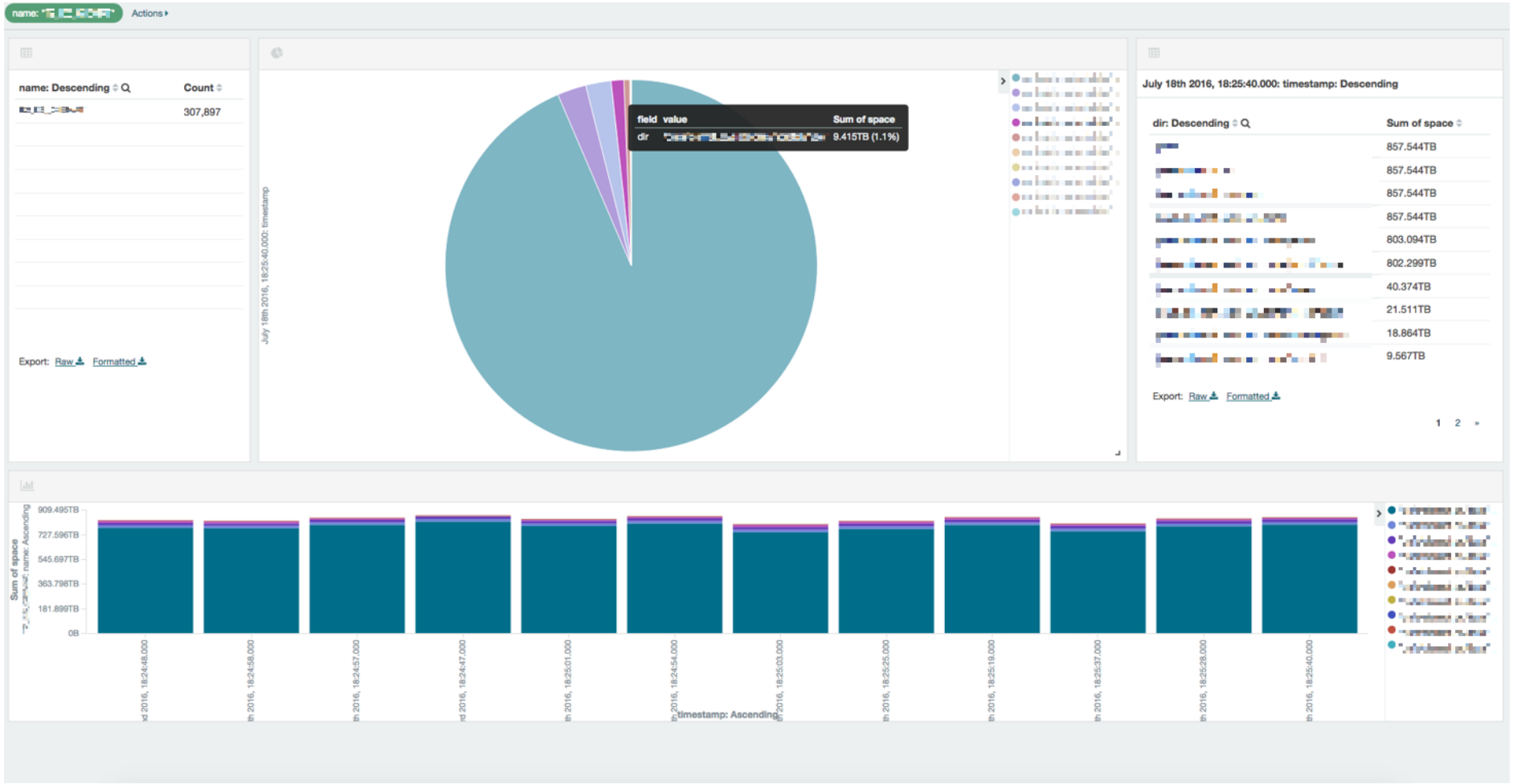
Drawing on documentation and experience gained using this tools:

feature	Kibana	Grafana
license	open source	open source
documentation	well documented	well documented
data sources	Elasticsearch	Elasticsearch, *Graphite*, *CloudWatch*, ...
access via:	browser	browser, *http api*
role-based access	no. available for purchase (Shield)	yes. many auth options (usr/psw, *github oauth*, ...)
data discovery	yes	no
data search	Lucene query, Query DSL based on JSON	Lucene query (for ES data sources)
charts	several, including pie charts, histograms and tables	several, including time graphs and tables
chart plugins	yes (heatmap, ...)	yes (pie charts, ...)
dashboard	yes	yes. very flexible
sharing	(embedded) dashboard	(embedded) dashboard, snapshots
import/export	yes. JSON format	yes. JSON format
more remarks:	rather intuitive	less intuitive at first
	general purpose charts	charts focused on time-series
	possibility to save and reuse intermediate objects (searches, charts, ...)	intermediate objects cannot be saved or reused across different dashboards

VISUALISATION EXAMPLES

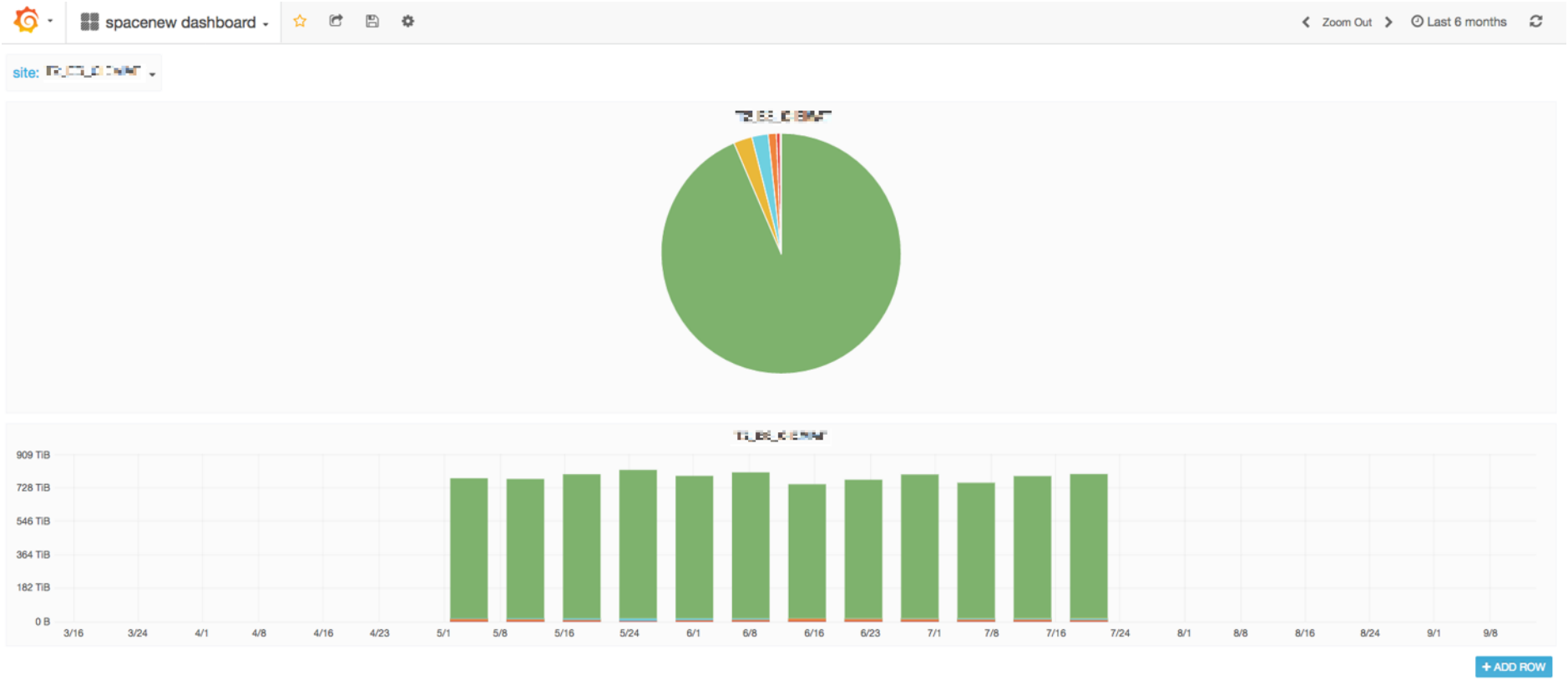
- Point out what folders alongside the official CMS data folder (“/store/”) take up site’s available storage capacity
- Trend over time of the above
- In/out “/store” total space usage over time

KIBANA – DASHBOARD



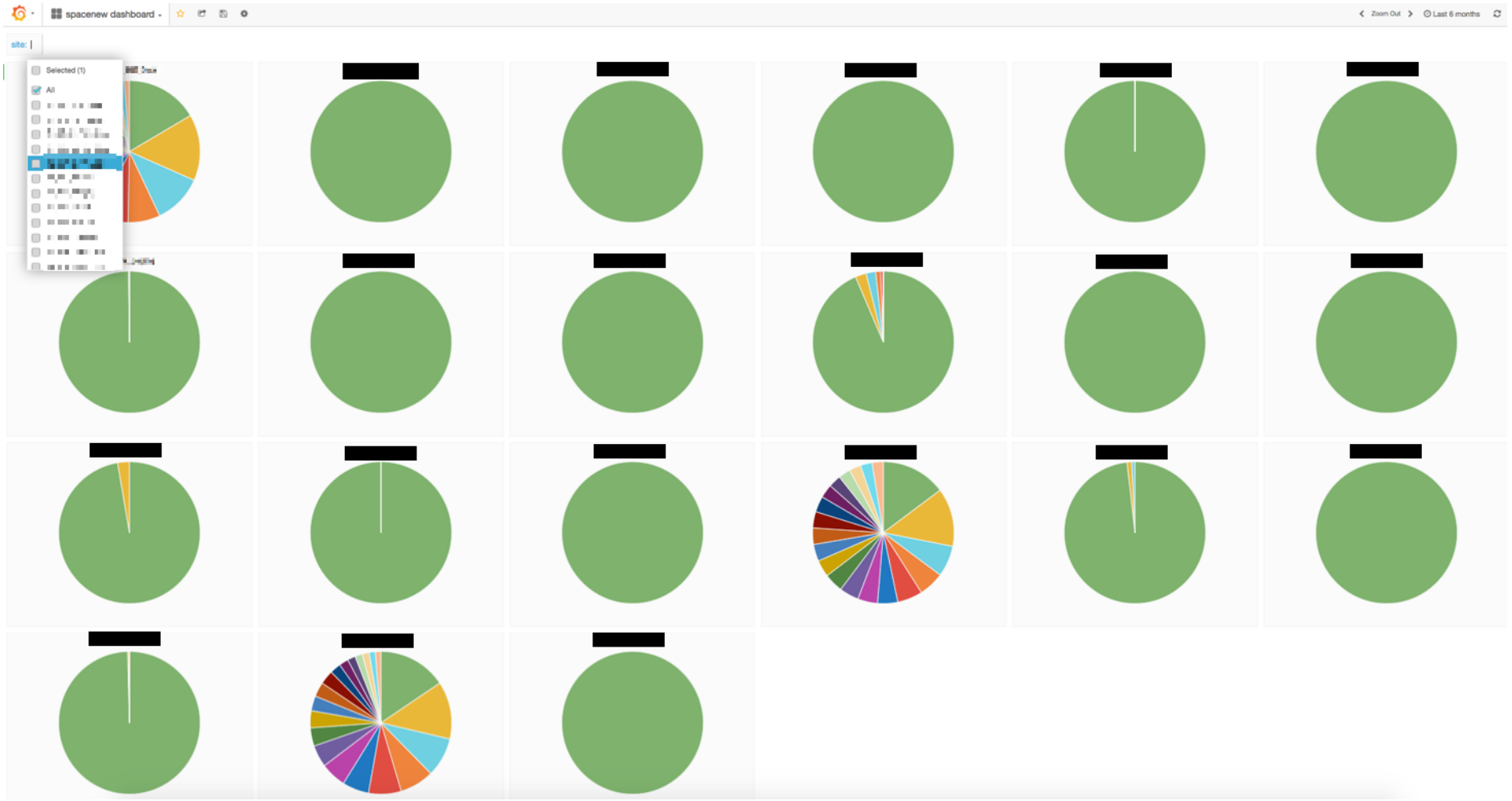
Fermilab

GRAFANA – DASHBOARD



Fermilab

GRAFANA – CHART REPETITION



FINDINGS (1)

- As for chart repetition on the basis of sites, which was of interest to this evaluation,
  - It was NOT possible in Kibana
  - It was possible in Grafana (thanks to the “templating” feature)

- Kibana and Grafana offer a good set of charts to represent your data, along with a few options to customize those charts. However, being mere visualization tools, they offer very limited chances to process data within them (only simple numeric operations on numeric fields).

- Therefore, if the data needs to be processed, it should be done prior to Kibana and Grafana.

FINDINGS (2)

- As a result, an intermediate step, in which the monitoring data is arranged according to the use cases of interest, is needed. That can be a different data service API or a different software tool prior to ES/Kibana/Grafana.

- In fact, in order to draw the visualizations related to most use cases a different arrangement of monitoring data is required.

- For example, in one case such visualizations could not be drawn due to the fact that monitoring data is sampled asynchronously whereas ES/Kibana/Grafana do not provide flexible time aggregation.

DOCUMENTATION

In support of my findings, I spent time documenting my work online.

- Work logs (Redmine)
- Twiki (Redmine)
- Source code with comments (GitHub)

Note: Redmine is a project management web application widely used at Fermilab.

SUMMARY

Goals accomplished:

- Learned about data management process in a distributed computing environment
- Got familiar with the software development model of a large international scientific collaboration
- Gained experience in testing and evaluating software products

THANK YOU