Exploiting analytics techniques in CMS computing monitoring

Eric Vaandering
on behalf of Daniele Bonacorsi, Valentin Kuznetsov, Nicolò Magini, Luca Menichetti and Aurimas Repečka
CMS Data Management system

PhEDEx - CMS data transfer service and replica catalog

Oracle DB

Dataset

Block

Block

File

File

File
PhEDEx monitoring

- PhEDEx status is constantly monitored in Oracle DB
- Live data is regularly flushed to preserve performance
- Many monitoring data aggregated in historical tables
  - Transfer rates, latencies, total data volumes
- A lot of other data are lost
  - E.g. replica location history
Extending PhEDEx monitoring

- We want to monitor space used by different datasets to plan distribution and cleaning
- Preserve replica history exporting daily snapshots to HDFS with Sqoop
- CERN IT analytix Hadoop cluster
  - 38 nodes, 2.8 PB
- Extends monitoring without impact on live service
Monitoring Process Structure

- Input
- CSV on HDFS
- Spark
- Processing
- HDFS and Elasticsearch
- Output
- Visualization
- Kibana
Input – Snapshots

- Daily replica catalog snapshots in CSV format
- Size of a snapshot: 2.0-3.5 GB, 5.5M-8.5M rows
  - For 1 year: 1 TB of data
- Fields
  - Snapshot date
  - Storage node name (Node Tier)
  - Dataset name (Acquisition Era, Data Tier)
  - Block name
  - Block replicas node bytes
  - Block replicas node files
  - etc…
### Processing

- Spark jobs to process input data in parallel on the cluster
- Running with customizable parameters for aggregation
  - Group keys
  - Result values
  - Aggregation functions
    - Sum, count, first, last, min, max, mean
    - Daily variation over a time interval, daily avg over a time interval
  - Data ordering
  - Data filtering
Performance

- Example of a typical aggregation query
- Sum of the size of all dataset replicas, grouped per day, per disk/tape storages, per user groups, and per dataset types (acquisition era, data tier)

| Interval   | Input     | Cores | Output    | Duration  |
|------------|-----------|-------|-----------|-----------|
| 1 day      | ~3GB      | 65    | ~600KB    | ~1.6min   |
| 1 month    | ~100GB    | 65    | ~18MB     | ~4.3min   |
| 1 year     | ~1.1TB    | 65    | ~186MB    | ~28min    |
Output

HDFS → Spark → Elastic Search → Kibana

STDOut
Visualization in Kibana — Space by Campaign

Analysis space

Production space

2016 data
2016 MC
2015 data
2015 MC
2015 Sim
2016 Sim
Pileup

run2016b
runispring16dr80
run2016g
runifall15dr76
runispring15dr74
run2016d
run2016e
run2016c
run2015d
run2016f
runisummer15gs
runispring15prepmix
runisummer15wmlhegs
run2015e
runiwinter15gs
runisummer16dr80ba...
runispring16fprepmix
Visualization — Evolution of space by data tier
Status and future plans

▪ Current status
  ◦ Scheduling daily Spark jobs on analytix cluster for basic aggregations
  ◦ Deploying Elasticsearch/Kibana on a server managed by CMS computing

▪ Future plans
  ◦ Set up additional scheduled aggregations
  ◦ Enable submission of one-time custom queries
  ◦ Aggregate data from different CMS services e.g. dataset popularity
  ◦ Migrate visualization to central CERN IT Elasticsearch/Kibana service
Conclusion

- Enabled CMS dataset replica monitoring using analytics tools to aggregate and visualize data
- Efficient
  - Can process 1 TB of input data for 1 year in 30 minutes
- Fully-covered
  - We can afford to keep raw input data indefinitely
- Highly configurable
  - Aggregations can be customized for specific analyses
Backup
Hadoop-Elasticsearch connection

- Script to export HDFS files directly to Elasticsearch
- Reading one or multiple HDFS files with any number of partitions
- Configurable Elasticsearch parameters: node, port, resource (index/type)
- Data schema applied dynamically from user specified json file