ATLAS experience with the hadoop based analytics

Ilija Vukotic
2014/11/26

ATLAS needs

A way to do an in-depth, custom investigations from multiplicity of different data sources.

Already having a number of concrete questions and analyses to do:

- Repeat the scrutiny group study
- Studies on network performance data (FTS/XRootD,...)
- Check the effects of implemented user’s jobs prioritization model
- Job startup times analysis
- Detailed map of overflow jobs
- Effects of move to Rucio on jobs throughput
- ...

Will have multiple users.

Some services need a dedicated servers to churn time critical summary data.

Organization

e-group: atlas-adc-data-analytics

TWiki: https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/ATLASAnalytics

Jira: **ATLAS data analytics (ATLASMINER)**

git repo: https://git.cern.ch/reps/ATLAS-data-analytics

Clusters

lxhadoop

- Probably OK for most of us.
- Panda data are being uploaded there.
- Rucio data already there.
- No ACL’s on e-groups - that would be nice to have

Dashboard analytics cluster

- fully optimized for the Dashboard needs
    - Data format (e.g. Avro)
    - HDFS directory structure
    - all other Hadoop details
- will have data from multiple VOs
- multiple protocols (FTS, xrootd, http,...)

While other clusters can access data from there it won’t be very performant.

Getting access

Account at lxhadoop is easy to get. But logging is not straightforward:

```
ssh ivukotic@lxhadoop.cern.ch  -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
```

Attempt to get access to Rucio data lead to an enormous complication of creating and setting up private Agile, puppetized, VMs…

Getting the data

Options:

1. scp it to /tmp than hdfs dfs -put <file> <where>

   slow, error prone, cumbersome
2. “Sqoop” it from Oracle DB

   fast, one line command, still debugging issues with it.
3. Get it from AMQ

   ideal for the data already sent using AMQ (for example fax collector data)

   is or will be used for the transfers dashboard.

Storing the data

We should create root ATLAS data directory.

Now everything in personal directories.

While HBase could be good option for some special cases it is not yet tested sufficiently.

Choice of how data are written (CSV, TSV, AVRO, compressed or not, etc.) will strongly impact performance.

Analysing the data

- writing MR in java
  - the most performant code
  - the most expensive in terms of your time
  - do it only if a very special optimizations are needed
- pig
  - hides MR from user
  - much shorter and cleaner code
  - a lot of hidden optimizations
  - steep learning curve
  - simple visualizations can be done by copy/paste to Excel
- R/python - not yet tested

To do

Find out for all the data that we want:

- sizes
- how it will be used (in conjunction with what?)
- how often it will be used
- how much processing power it will need

With that, IT people can plan on distributing the data, so as to minimize “remote” data access and data replication.

To do

We want anybody to be able to re-run an analytics task. For that we need:

- Prepare simple getting started guides
- To extensively document the data: it’s origin, people responsible, descriptions down to column.
- A library of algorithms with detailed explanations on what do they do and how to interpret results.
- An automatic periodic check of each script in our git.
- An automatic check of the description.
- Measure performance of different formats and code