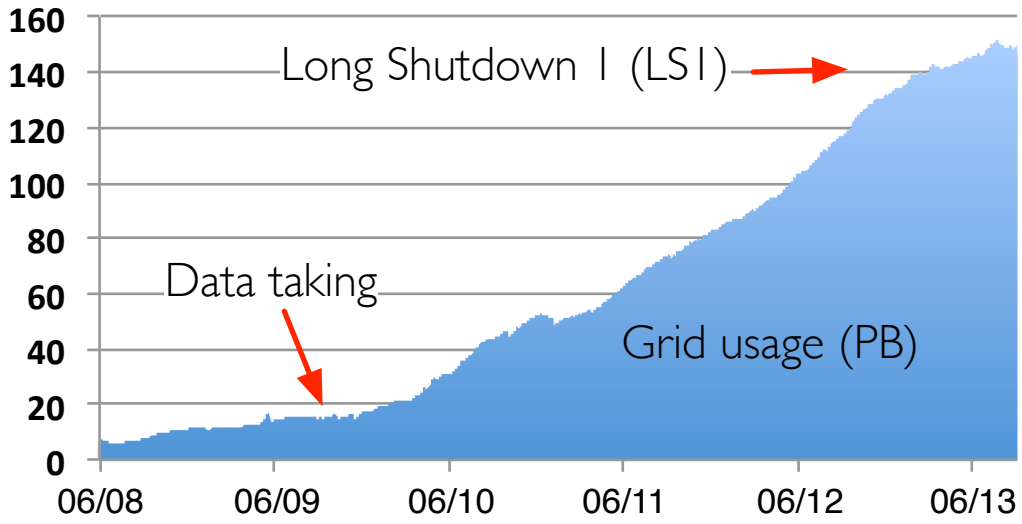
DATA ANALYTICS IN THE ATLAS DISTRIBUTED DATA MANAGEMENT SYSTEM

Vincent.Garonne@cern.ch

Openlab Workshop on Data Analytics Use Cases

DDM Background

- The Distributed Data Management project manages ATLAS data on the grid
  - 150 Petabytes
  - 500 million files
  - 1000 active users
  - 130 sites
  - + history



- The current system is Don Quijote 2 (DQ2) in production since 2004
- The next generation system is Rucio (scheduled for 2014)

Workloads: OLTP & Analytical

- Online transaction processing (OLTP) workload
  - Relational database management system (RDBMS)
  - Main database: Oracle 11g
  - Object-Relational Mapper: SQLAlchemy
  - Rucio supports also MySQL, PostgreSQL, etc.
- Analytical workload
  - RDBMS & Non relational structured storage (NoSQL)
  - Oracle & Hadoop

NoSQL Technology Selection

	MongoDB	Cassandra	Hadoop/HBase
Installation/ Configuration	Download, unpack, run	Download, unpack, configure, run	Distribution, Complex config
Buffered read 256	250'000/sec	180'000/sec	150'000/sec
Random read 256	20'000/sec	20'000/sec	20'000/sec
Relaxed write 256	10'000/sec	19'000/sec	9'000/sec
Durable Write 256	2'500/sec	9'000/sec	6'000/sec
Analytics	Limited MapReduce	Hadoop MapReduce	MapReduce, Pig, Hive
Durability support	Full	Full	Full
Native API	Binary JSON	Java	Java
Generic API	None	Thrift	Thrift, REST

12 node cluster located in CERN IT data:

96 CPU cores (Intel Xeon, 2.27GHz, 8/node), 288 GB RAM (24/node), 24 SATA (1 TB each, 2/node), 1 GigE network

Technology: 

	MongoDB	Cassandra	Hadoop/HBase
Installation/ Configuration	Download, unpack, run	Download, unpack, configure, run	Distribution, Complex config
Buffered read 256	250'000/sec	180'000/sec	150'000/sec
Random read 256	20'000/sec	20'000/sec	20'000/sec
Relaxed write 256	10'000/sec	19'000/sec	9'000/sec
Durable Write 256	2'500/sec	9'000/sec	6'000/sec
Analytics	Limited MapReduce	Hadoop MapReduce	MapReduce, Pig, Hive
Durability support	Full	Full	Full
Native API	Binary JSON	Java	Java
Generic API	None	Thrift	Thrift, REST

Hadoop is a framework for distributed data processing (not only a database) with many components: **HDFS** (distributed filesystem), **MapReduce** (distributed processing of large data sets), **HBase** (distributed data base for structured storage), **Hive**(SQL frontend), **Pig**: data-flow language for parallel execution, …

Use Case : Trace Mining

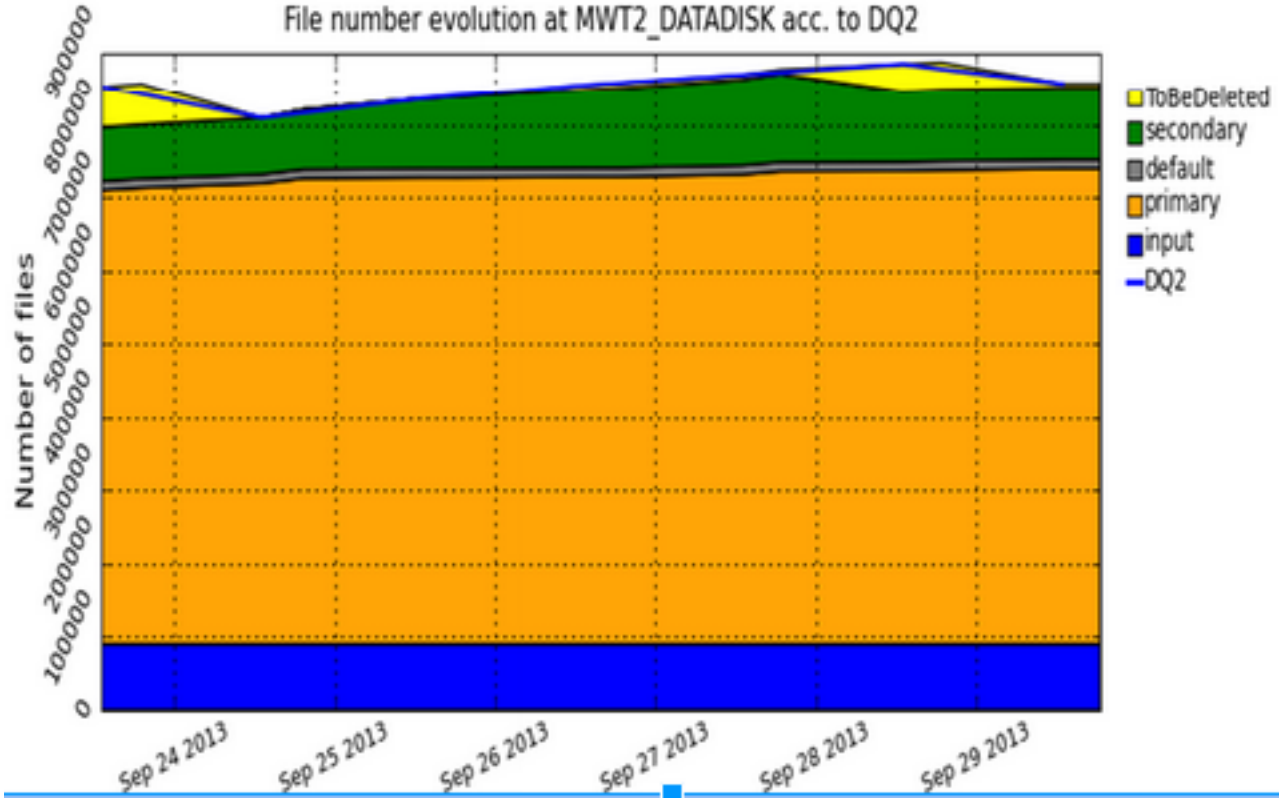
- Client interaction with ATLAS DDM generates traces
  - E.g., downloading a dataset/file from a remote site
- Lots of information (25 attributes) time-based and stored at the file level
  - E.g., Timestamp, dataset / File, User, Site, Transfer times
- Since the start in 2007 almost 7 billion traces have been collected
  - Average rate at 300 insertions/s
  - One month of traces ~80GB

Use Case : Popularity

- Popularity system aggregates at various granularities
  - traces into hourly reports
  - hourly reports into daily reports
  - by day, dataset, event type (local download, analysis, production, ...), sites, user, etc.
- Oracle based implementation
- Moving to hadoop
  - Interesting features: schemaless , Hbase, distributed atomic counters, etc.

Replica Reduction

- Popularity is used for data deletion by Victor
  - If a threshold is reached, it looks at all replicas on the site with no accesses reported for a certain time period
  - Then, if it is a secondary category copy, it will be sent to the deletion service

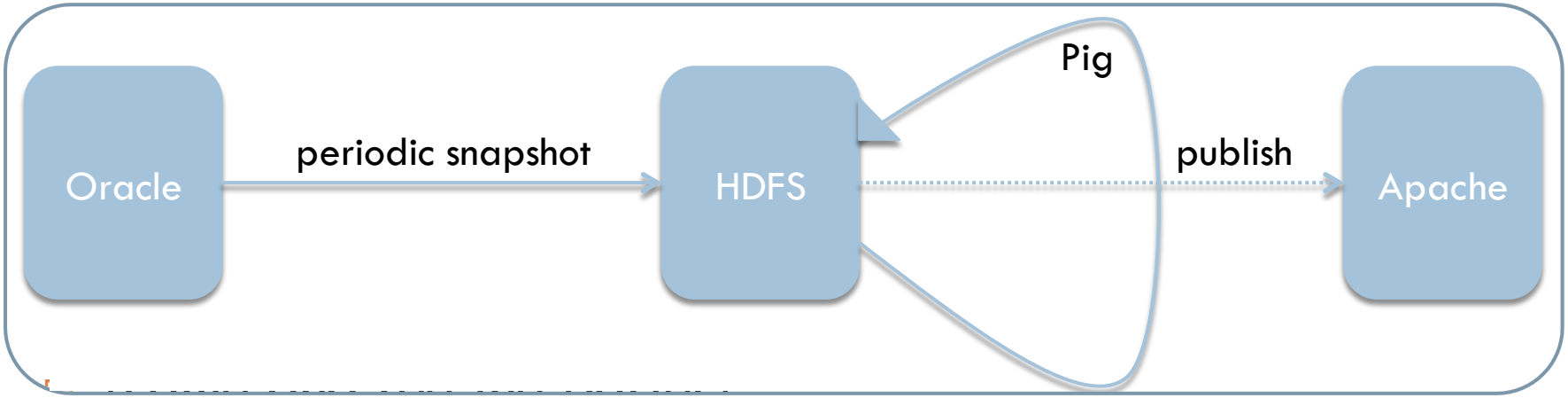


Use Case : Accounting & Popularity

- Regular reports are created
  - For computing management, visualization front-ends, etc
- Break down usage of ATLAS data contents/Popularity
  - Historical free-form meta data queries

```
{site, nbfiles, bytes} := {project=data10*, datatype=ESD, location=CERN*}
```

- A full accounting run takes about 8 minutes
  - Pig data pipeline creates MapReduce jobs
  - 7 GB of input data, 100 MB of output data

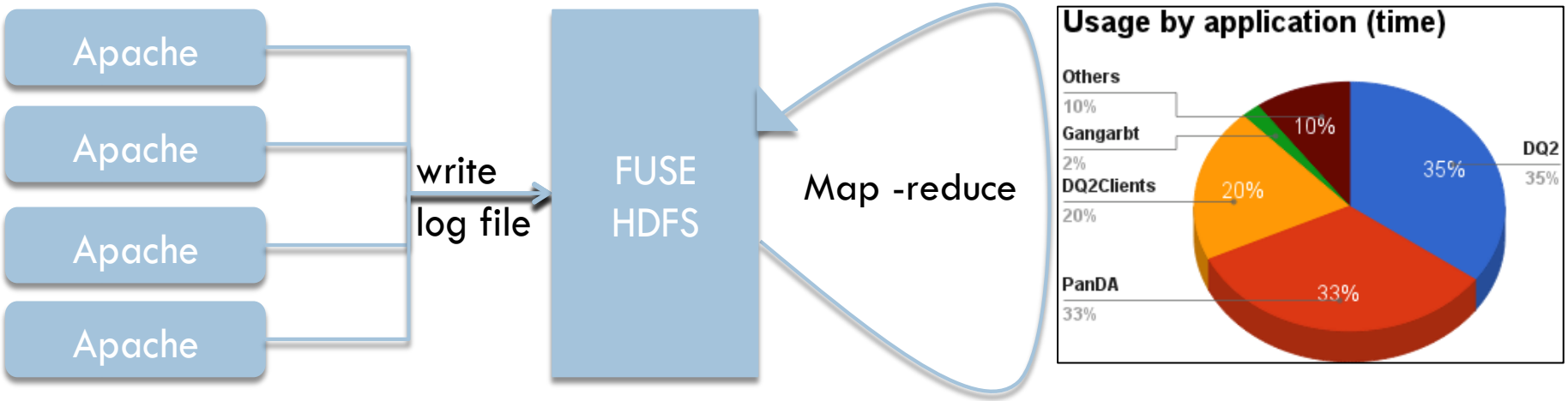


Automated Replica Creation

- Currently popularity is only used in an automated way for deletion but replication policies definition is a static process at the moment
- Idea: Use the popularity also to make new replicas automatically for popular datasets, i.e.,
  - Forecasts about future dataset popularity
  - Decisions how many datasets to delete and where (i.e., how much space to free up
  - Decisions on where to replicate new copies for which datasets
- Ongoing Ph.D student work(thomas.beermann@cern.ch)
  - Static/Linear/neural network prediction
  - Simulation

Use Case : Log File Aggregation

- Monitoring infrastructure based on Hadoop to analyse central catalog traffic



- Daily copies of all the ATLAS DDM log files
- 8 months of logs = 3 TB on HDFS
- Python MapReduce jobs to analyse the log files

Conclusion

- ATLAS Distributed Data Management uses both SQL and NoSQL
  - We see NoSQL complementary to RDBMS, not as a replacement
- DDM Analytic use cases are well covered
  - Hadoop proved to be the correct choice: Stable – reliable – fast – easy to work with
- Happy to work with interested parties
  - Many other groups/projects adopting similar solutions
  - CERN-IT Hadoop testbed in use for ATLAS DDM

Thanks !

http://rucio.cern.ch