DDM: Future technologies

Database

Vincent.Garonne@cern.ch

ADC Retreat, Napoli, Febuary 2011

Current limitations of Oracle

- Hard to scale
- De-normalization for better performance
- Complicated setup/configuration with apache
- Bad for large, random and I/Os intensive applications

NoSQL Databases

- ‘Scale easily’ – But different architecture
- ‘Get the work done’ on lower cost infrastructure
- Same application level reliability
- Google, Facebook, Amazon, Yahoo!

$\Rightarrow$ Buzz words ☺ or new fundamental concepts?

Proposal: Evaluation and integration

- Strong requirements
  - Stability, backward compatibility, no performance losses
- Non critical or new components
  - Tracer, Popularity, Accounting, Dataset meta-data
  - Limitations with Oracle and relevant for NoSQL
  - Characteristics: Column-oriented, permutation, statistics, data analysis – Ref. Future computing session
- Redundant and complementary approach
  - Self-contained Oracle backup table
  - Easier to stream to another back-end
  - Validation and comparison of results

Message queue

- Key and critical component – $\mathcal{A}$ctive$\mathcal{MQ}$ ☺
- Simple, reliable and scalable mechanism
  - Load balancing, persistency, transactions
  - Message ordering, less polling
- Many use cases
  - $\mathcal{DDM}$ internal
  - External applications
    - AMI/PANDA/DASHBOARD/LFC/DPM
  - Fresh meta-data in distributed caches ?
  - Inter-backend synchronization

First experiences with NoSQL

- Open source projects
  - Bigtable vs Cassandra vs Simpledb vs Dynamo vs Couchdb vs Hypertable vs Riak vs Hadoop Hbase vs etc.
- User community and commercial support
- Common ADC Testbed cluster
  - 4 VM nodes, 2 Intel(R) Xeon(R) CPU 2.27GHz, 4G
- Infrastructure monitoring tools
- Applications
  - Cassandra: Tracer, Panda monitoring/historical data
  - Hbase: Accounting, dataset meta-data

Cassandra evaluation

- Thanks to Maxim Potekhin and Donal Zang !
- On the learning curve
  - High write speed
  - Bugs but good responsiveness of developer team
  - Instability due to a not-so-optimal setup

  $\Rightarrow$ Non VM nodes and performance tuning
    - Memory for key, row caching and throughput
    - Two hard drives to isolate commit log and data file
    - Large disks for compaction and cluster reconfiguration
    - RAID0 for read speed
    - Fast multicore CPUs to calculate bloom filters during writes
- Invitation to other applications to join us !

How to automate a policy change ?

Policy and rule management

Observations

- Lots of code changes are related to policy changes
- Examples: $\mathcal{DQ}2$, AKTR, Datri, Panda
- Distinction between the implementation and the policies ?

Rule-based data management system ?

- Store the rules in a common place
- Open source projects
  - iRODS/Wave – `www.irods.org`,
    Pyke – `http://pyke.sourceforge.net`,
    Pyclips – `http://pyclips.sourceforge.net`
- Demonstrator with ATLAS data management rules ?

Thanks !