Hadoop at a Tier2

Amdamjam 2010

Brian Bockelman

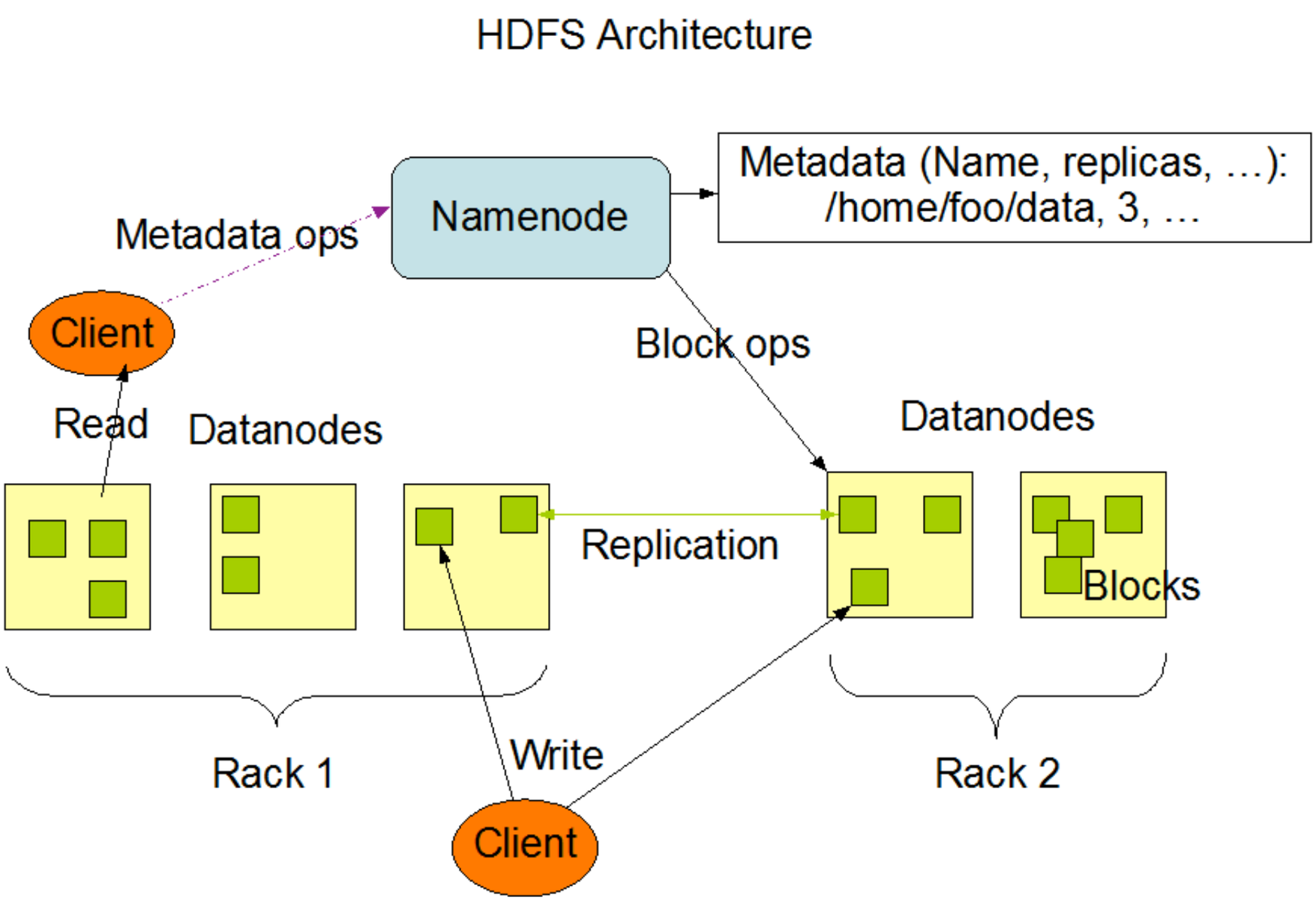
Hadoop Primer

- Hadoop is an open source data processing system based upon the designs of the Google Filesystem and Google MapReduce.
- There is a scheduling component (also called MapReduce) and a filesystem component (Hadoop Distributed Filesystem, HDFS).
  - We are going to talk about HDFS today.

HDFS

- HDFS is designed to be a fault-tolerant, highly scalable file system which runs on the worker node.
  - Stuff your WN full of 2TB disks...
  - Fault tolerance means WN crash reduces capacity but not file availability.

Standard Diagram



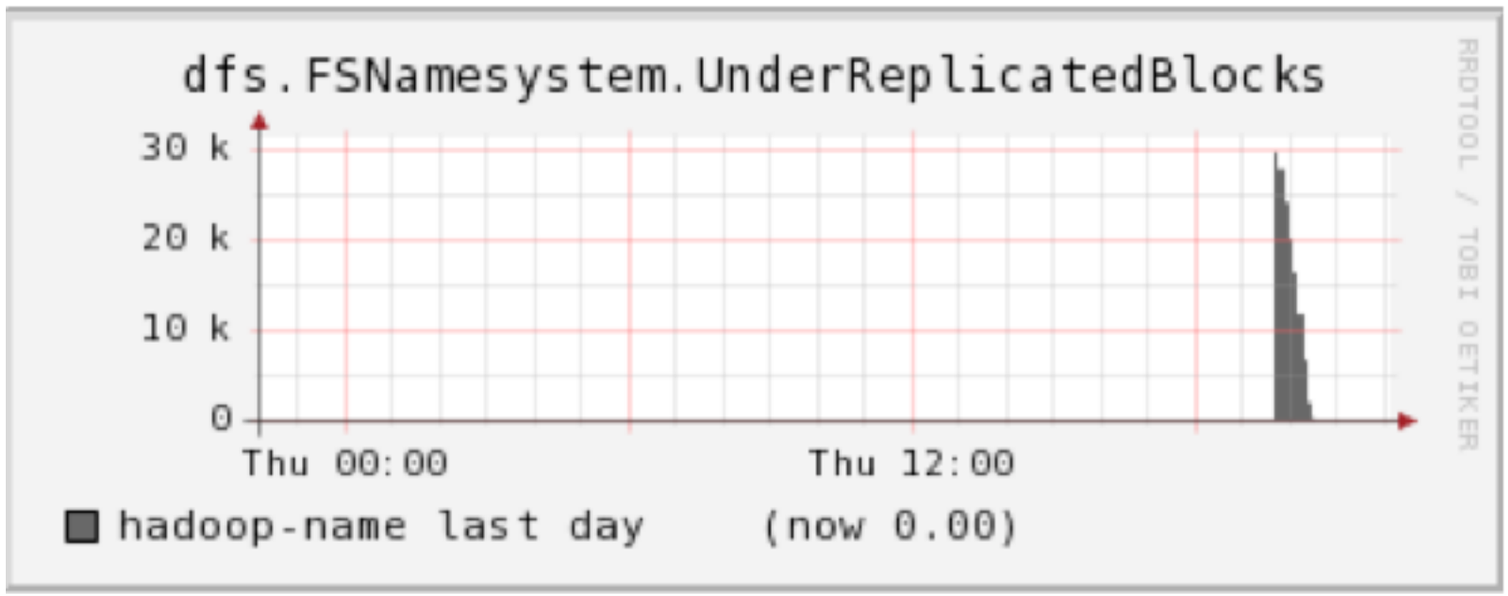
http://hadoop.apache.org/common/docs/r0.20.0/hdfs_design.html

Why HDFS?

- In order, this is why we are excited about HDFS:
  - Less management.
  - More reliability (fsck!).
  - Better scalability.
  - Usability.

Reliability

- Clients will automatically connect to a different datanode if one fails during a read.
- Blocks will automatically re-replicate -- and quickly! Often, we will recover from a loss in an hour.
  - Namenode controls this. We have it set to re-replicate if a node hasn't checked in for 10 minutes.
- All data is checksummed on read.



HDFS on the Grid

- On top of HDFS, we layer:
  - BeStMan SRM server (lightweight, wicked fast [200Hz], don't use true dynamic space tokens, stateless).
  - Globus GridFTP + HDFS module.
  - Xrootd/HDFS integration.
  - FUSE (POSIX-like access for clients).

Highlights

- Block-based (128MB blocks). No hotspots.
- Fast effective replication. Balancing.
- Background checksumming of all data.
- Easy to add/remove nodes.
- Lightning-fast metadata server (126k opens/sec, 5k writes/sec).
- Yum/RPM packaging.
- Pull the plug on a rack, no errors for clients.

FUSE bindings

- Never underestimate how much users *love* "Is".
  - All the "grid stuff" can get out of their hair.
    - "User love" is much harder to quantify than GB/s.
      - Is "user love" more important than scalability? Can we have both?
- And having the Linux VFS working on our side is pretty nice too...

Drawbacks

- Single metadata server is a SPOF.
- GridFTP servers are memory hungry.
  - Globus gridftp does a poor job of resource provisioning.
- POSIX-like, not POSIX!
- Different Frame Of Mind (structural changes)

HadoopViz Demo

- (Dependent on network connectivity)
- http://www.youtube.com/watch?v=qoBoEzOkeDQ

Xrootd/HDFS

- HDFS's security model... isn't.
  - You don't want to expose it to the outside world (although you can do some reasonable).
  - We wanted to efficiently export our HDFS to our users worldwide (or the folks on the other side of the campus).
  - So, we have a nice plugin for Scalla xrootd which utilizes the HDFS C API.
    - Unlocks the Scalla features while keeping the HDFS highlights.

Demonstration again

- If the network fails, there's no YouTube backup for this one...

Conclusions

- HDFS is a highly reliable system.
- Replication reduces admin time.
- Great for T2 usage.
- Focussed on layering on modular components to extend functionality of very solid core.

Links

- http://www.usenix.org/publications/login/2010-04/openpdfs/shvachko.pdf
- http://iopscience.iop.org/1742-6596/180/1/012047
- https://twiki.grid.iu.edu/bin/view/Storage/Hadoop
- http://dcache-head.unl.edu:8088/