Multi-model Databases

- … but one application can actually require different data models for the different data it stores
- Provide support for multiple data models against a single backend:
  - OrientDB supports key-value, document, graph & object models; geospatial data;
  - ArangoDB supports key-value, document & graph models stored in JSON; common query language;
- How to query the different models in a uniform way

Big Data Analytics Stack

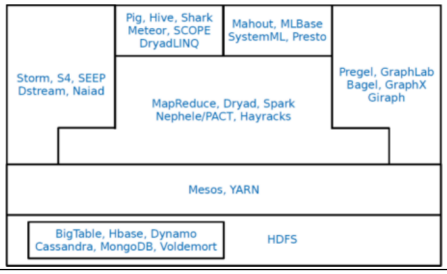


figure from: https://www.sics.se/~amir/dic.htm

HDFS[Hadoop][HDFS][HDFSpaper]

Hadoop Distributed File System

Compute Nodes[Massive]

- Compute node – processor, main memory, cache and local disk
- Organized into racks
- Intra-rack connection typically gigabit speed
- Inter-rack connection slower by a small factor

HDFS (Hadoop Distributed File System)

- Runs on top of the native file system
  - Files are very large divided into 128 MB chunks/blocks
    - To minimize the cost of seeks
  - Caching blocks is possible
  - Single writer, multiple readers
  - Exposes the locations of file blocks via API
  - Fault tolerance and availability to address disk/node failures
    - Usually replicated three times on different compute nodes
- Based on GFS (Google File System - proprietary)

HDFS is Good for …

- Store very large files – GBs and TBs
- Streaming access
  - Write-once, read many times
  - Time to read the entire dataset is more important than the latency in reading the first record.
- Commodity hardware
  - Clusters are built from commonly available hardware
  - Designed to continue working without a noticeable interruption in case of failure

HDFS is currently Not Good for …

- Low-latency data access
  - HDFS is optimized for delivering high throughput of data
- Lots of small files
  - the amount of files is limited by the memory of the namenode; blocks location is stored in memory
- Multiple writers and arbitrary file modifications
  - HDFS files are append only – write always at the end of the file

HDFS Organization

- Namenode (master)
  - Manages the filesystem namespace and metadata
  - Stores in memory the location of all blocks for a given file
- Datanodes (workers)
  - Store and retrieve blocks
  - Send heartbeat to the namenode
- Secondary namenode
  - Periodically merges the namespace image with the edit log
  - **Not** a backup for a namenode, only a checkpoint

HDFS Organization

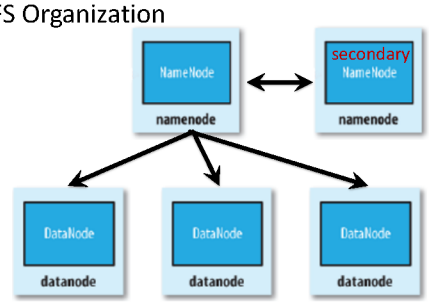
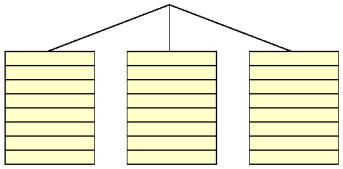


figure based on a figure from [Hadoop]

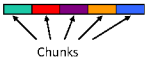
Block Placement and Replication

- Aim – improve data reliability, availability and network bandwidth utilization
- Default replica placement policy
  - No Datanode contains more than one replica
  - No rack contains more than two replicas of the same block
- Namenode ensures the number of replicas is reached
- Balancer tool – balances the disk space usage
- Block scanner – periodically verifies checksums



Racks of Compute Nodes

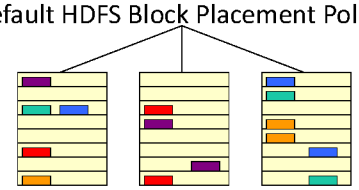
File



Chunks

Source: J. D. Ullman invited talk EDBT 2011

Default HDFS Block Placement Policy



- 1st replica located on the writer node
- 2nd and 3rd replicas on two different nodes in a different rack
- The other replicas are located on random nodes

HDFS – File Reads

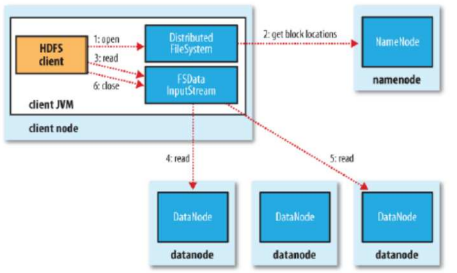


figure from [Hadoop]

HDFS – File Writes

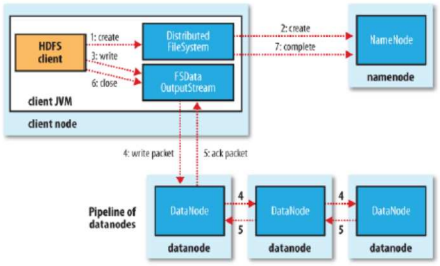


figure from [Hadoop]

HDFS – High Availability

- The namenode is single point of failure:
  - If a namenode crashes the cluster is down
- Secondary node
  - periodically merges the namespace image with the edit log to prevent the edit log from becoming too large.
  - lags the state of the primary prevents data loss but does **not** provide high availability
  - time for cold start 30 minutes
- In practice, the case for planned downtime is more important

HDFS – High Availability

- Pair of namenodes in an active stand-by configuration:
  - Highly available shared storage for the shared edit log
  - Datanodes send block reports to all namenodes
  - Clients must provide transparent to the user mechanism to handle failover
  - The standby node takes checkpoints of the active namenode namespace instead of the secondary node

HDFS commands

- List all options for the hdfs dfs
  - `hdfs dfs -help`
  - `dfs` – run a filesystem command
- Create a new folder
  - `hdfs dfs -mkdir /BigDataAnalytics`
- Upload a file from the local file system to the HDFS
  - `hdfs dfs -put bigdata /BigDataAnalytics`

HDFS commands

- List the files in a folder
  - `hdfs dfs -ls /BigDataAnalytics`
- Determine the size of a file
  - `hdfs dfs -du -h /BigDataAnalytics/bigdata`
- Print the first 5 lines from a file
  - `hdfs dfs -cat /BigDataAnalytics/bigdata | head -n 5`
- Copy a file to another folder
  - `hdfs dfs -cp /BigDataAnalytics/bigdata /BigDataAnalytics/AnotherFolder`

HDFS commands

- Copy a file to a local filesystem and rename it
  - `hdfs dfs -get /BigDataAnalytics/bigdata bigdata_localcopy`
- Scan the entire HDFS for problems
  - `hdfs fsck /`
- Delete a file from HDFS
  - `hdfs dfs -rm /BigDataAnalytics/bigdata`
- Delete a folder from HDFS
  - `hdfs dfs -rm -r /BigDataAnalytics`

References

- A comparison between several NoSQL databases with comments and notes by Bogdan George Tudorica, Cristian Bucur
- nosql-databases.org
- Scalable SQL and NoSQL data stores by Rick Cattel
- [Brewer] Towards Robust Distributed Systems @ACM PODC'2000
- [12 years later] CAP Twelve Years Later: How the "Rules" Have Changed, Eric A. Brewer, @Computer Magazine 2012. https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed
- [Fox et al.] Cluster-Based Scalable Network Services @SOSP'1997
- [Karger et al.] Consistent Hashing and Random Trees @ACM STOC'1997
- [Coulouris et al.] Distributed Systems: Concepts and Design, Chapter: Time & Global States, 5th Edition
- [DataMan] Data Management in cloud environments: NoSQL and NewSQL data stores.

References

- NoSQL Databases - Christof Strauch – University of Stuttgart
- The Beckman Report on Database Research
- [Vogels] Eventually Consistent by Werner Vogels, doi:10.1145/1435417.1435432
- [Hadoop] Hadoop The Definitive Guide, Tom White, 2011
- [Hive] Hive - a petabyte scale data warehouse using Hadoop
- https://github.com/Prokopp/the-free-hive-book
- [Massive] Mining of Massive Datasets
- [HiveManual] https://cwiki.apache.org/confluence/display/Hive/LanguageManual
- [Shark] Shark: SQL and Rich Analytics at Scale
- [SparkSQLHistory] https://databricks.com/blog/2014/07/01/shark-spark-sql-hive-on-spark-and-the-future-of-sql-on-spark.html

References

- [HDFS] The Hadoop Distributed File System
- [Dynamo] Dynamo: Amazon's Highly Available Key-value Store, 2007
- [HBaseInFacebook] Apache hadoop goes realtime at Facebook
- [HBase] HBase The Definitive Guide, 2011
- [HDFSpaper] The Hadoop Distributed File System @MSST2010