facebook

Hadoop Scalability at Facebook

Dmytro Molkov (dms@fb.com)
YaC, Moscow, September 19, 2011

How Facebook uses Hadoop
Hadoop Scalability
Hadoop High Availability
HDFS Raid

How Facebook uses Hadoop

Usages of Hadoop at Facebook

- Warehouse
  - Thousands of machines in the cluster
  - Tens of petabytes of data
  - Tens of thousands of jobs/queries a day
  - Over a hundred million files
- Scribe-HDFS
  - Dozens of small clusters
  - Append support
  - High availability
  - High throughput

Usages of Hadoop at Facebook (contd.)

- Realtime Analytics
  - Medium sized hbase clusters
  - High throughput/low latency
- FB Messages Storage
  - Medium sized hbase clusters
  - Low latency
  - High data durability
  - High Availability
- Misc Storage/Backup clusters
  - Small to medium sized
  - Various availability/performance requirements

Hadoop Scalability

Hadoop Scalability

- Warehouse Cluster - A “Single Cluster” approach

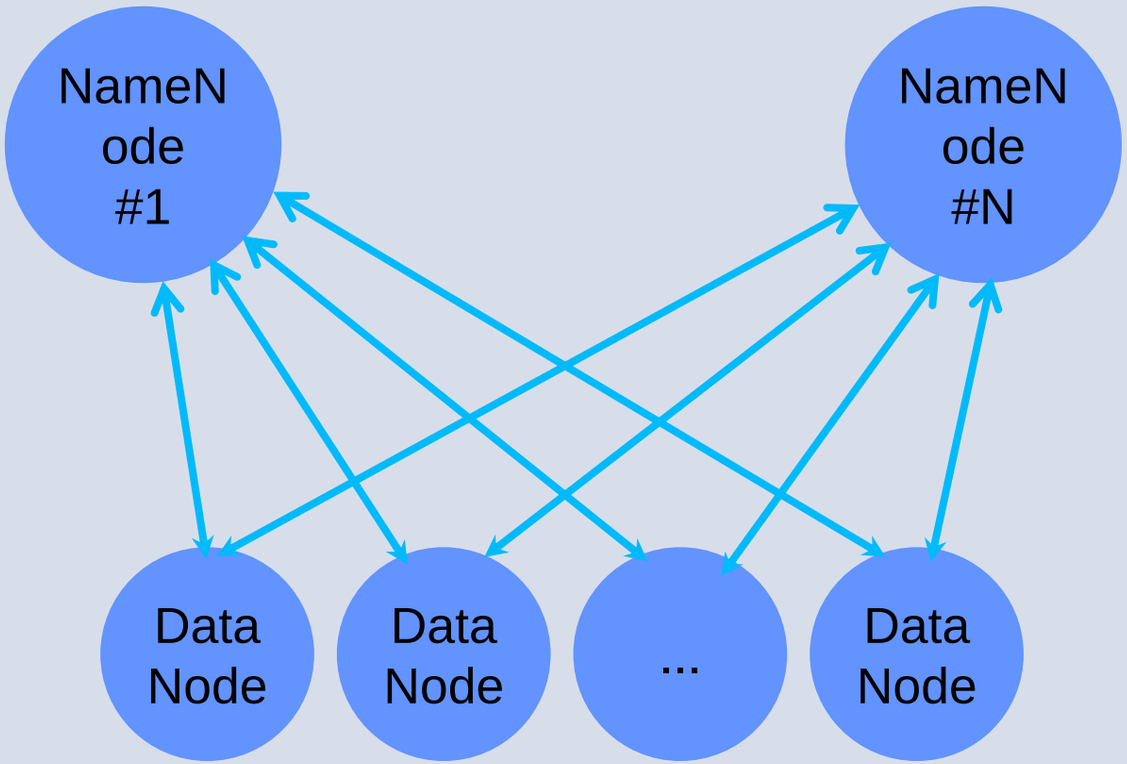
  - Good data locality
  - Ease of data access
  - Operational Simplicity

  - NameNode is the bottleneck
    - Memory pressure - too many files and blocks
    - CPU pressure - too many metadata operations against a single node
  - Long Startup Time
  - JobTracker is the bottleneck
    - Memory Pressure - too many jobs/tasks/counters in memory
    - CPU pressure - scheduling computation is expensive

HDFS Federation Wishlist

- Single Cluster
- Preserve Data Locality
- Keep Operations Simple
- Distribute both CPU and Memory Load

Hadoop Federation Design



HDFS Federation Overview

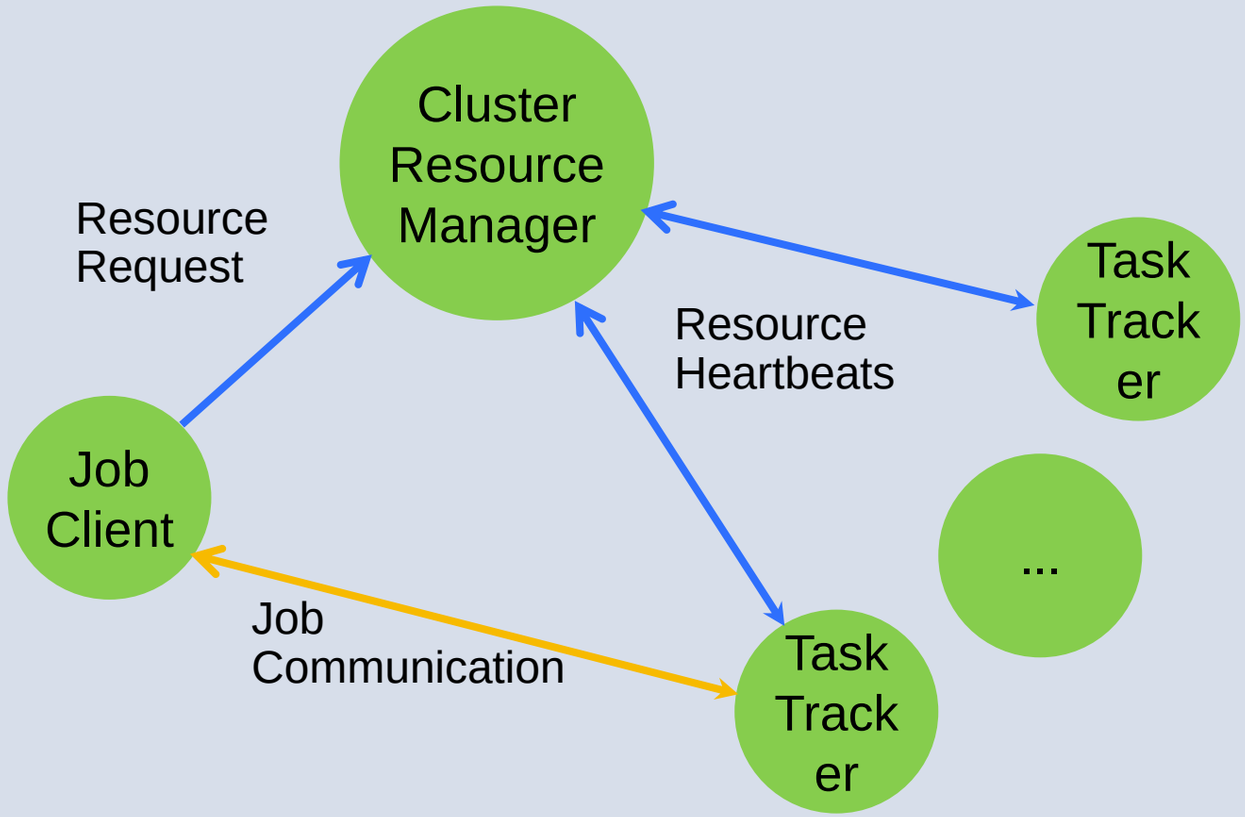
- Each NameNode holds a part of the NameSpace
- Hive tables are distributed between namenodes
- Hive Metastore stores full locations of the tables (including the namenode) -> Hive clients know which cluster the data is stored in
- HDFS Clients have a mount table to know where the data is

- Each namespace uses all datanodes for storage -> the cluster load is fully balanced (Storage and I/O)
- Single Datanode process per node ensures good utilization of resources

Map-Reduce Federation

- Backward Compatibility with the old code
- Preserve data locality
- Make scheduling faster
- Ease the resource pressure on the JobTracker

Map Reduce Federation



MapReduce Federation Overview

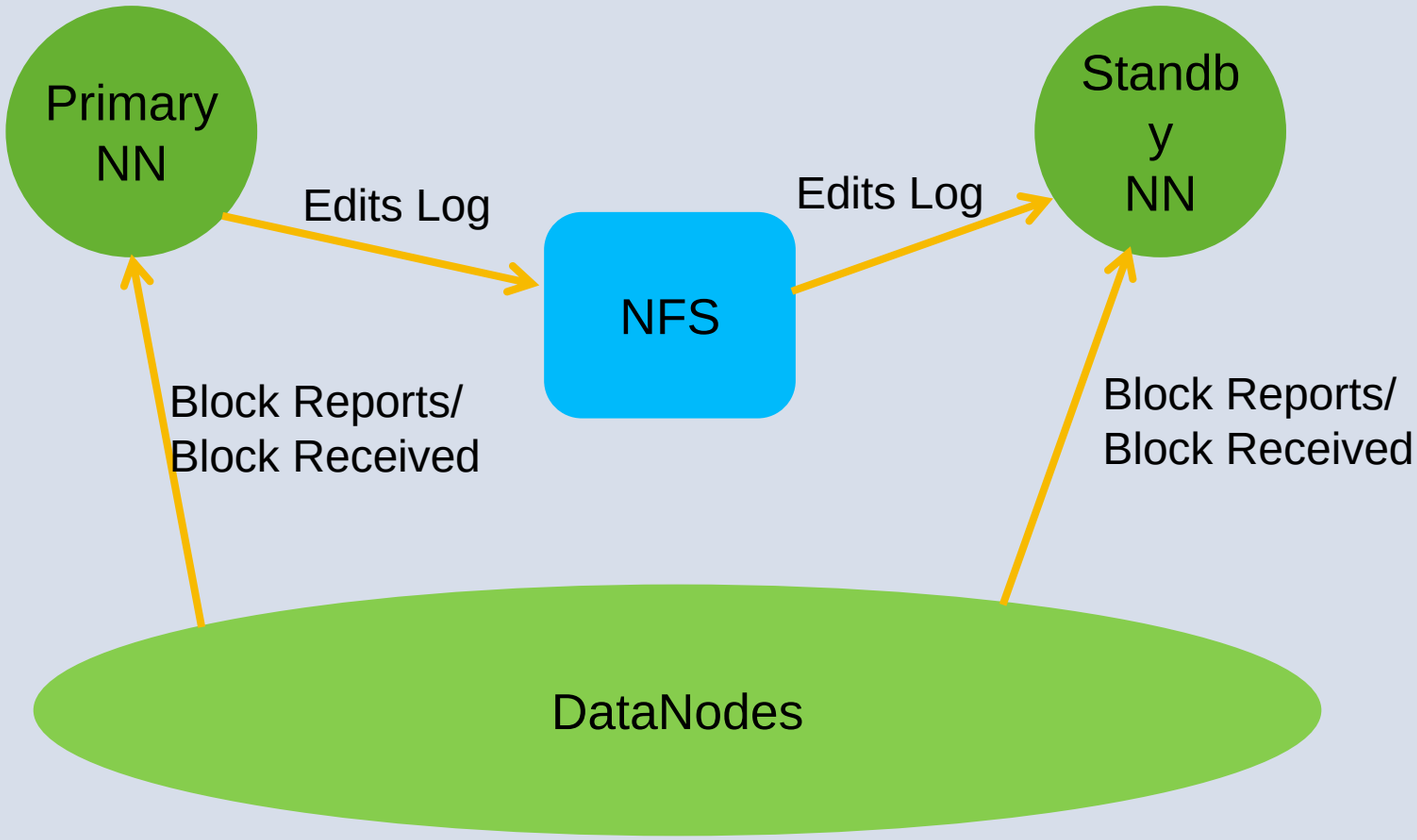
- Cluster Manager only allocates resources
- JobTracker per user -> few tasks per JobTracker -> more responsive scheduling
- ClusterManager is stateless -> shorter restart times -> better availability

Hadoop High Availability

Warehouse High Availability

- Full cluster restart takes 90-120 mins
- Software upgrade is 20-30 hrs of downtime/year
- Cluster crash is 5 hrs of downtime/year
- MapReduce tolerates failures

HDFS High Availability Design



Clients Design

- Using ZooKeeper as a method of name resolution
- Under normal conditions ZooKeeper contains a location of the primary node
- During the failover ZooKeeper record is empty and the clients know to wait for the failover to complete
- On a network failure clients check if the ZooKeeper entry has changed and retry the command agains the new Primary NameNode if the failover has occurred
- For the large clusters Clients also cache the location of the primary on the local node to ease the load on the zookeeper cluster

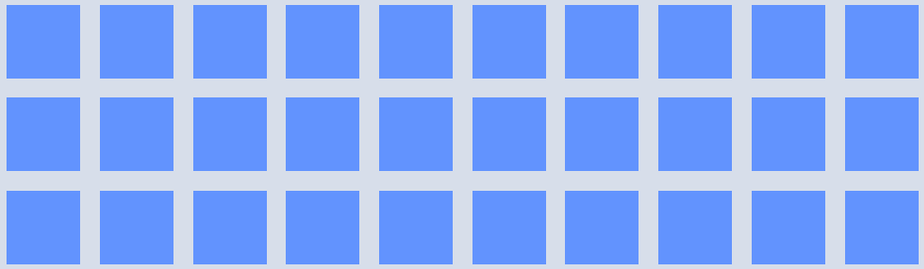
HDFS Raid

HDFS Raid

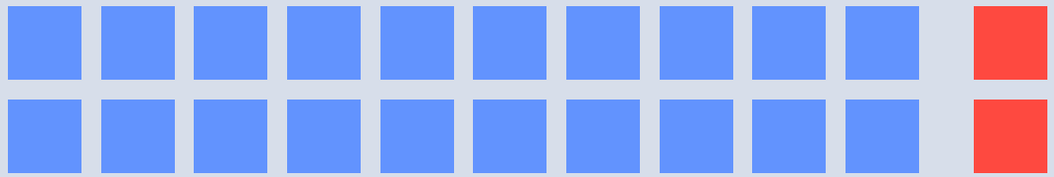
- 3 way replication
  - Data locality - necessary only for the new data
  - Data availability - necessary for all kinds of data
- Erasure codes
  - Data locality is worse than 3 way replication
  - Data availability is at least as good as 3 way replication

HDFS Raid Detais

XOR

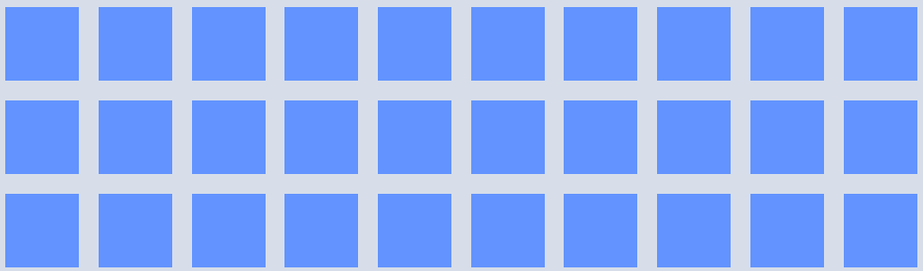


10 blocks replicated 3 times = 30 physical blocks
Effective replication factor 3.0



10 blocks replicated twice +
checksum (XOR) block replicated
twice = 22 physical blocks.
Effective replication factor 2.2

Reed Solomon Encoding



10 blocks replicated 3 times = 30 physical blocks
Effective replication factor 3.0

10 blocks with replication factor 1 +
erasure codes (RS) replicated once =
14 physical blocks.
Effective replication factor 1.4

HDFS Raid Pros and Cons

- Saves a lot of space
- Provides same guarantees for data availability

- Worse data locality
- Need to reconstruct blocks instead of replicating (CPU + Network cost)
- Block location in the cluster is important and needs to be maintained

facebook.com/dms

dms@fb.com

facebook