Hadoop Overview and Installation
August 10, 2010

Michael Thomas

What is Hadoop

Map-Reduce plus the HDFS filesystem implemented in java

Map-Reduce is a highly parallelized distributed computing system

HDFS is the distributed cluster filesystem

- **This is the feature that we are most interested in**

Open source project hosted by Apache

Used throughout Yahoo. Yahoo is a major contributor to the Apache Hadoop project.

HDFS

Distributed Cluster filesystem

Extremely scalable – Yahoo uses it for multi-PB storage

Easy to manage – few services and little hardware overhead

Files split into blocks and spread across multiple cluster datanodes

- **64MB blocks default, configurable**
- **Block-level decomposition avoids 'hot-file' access bottlenecks**
- **Block-level decomposition means the loss of multliple data nodes will result in the loss of more files than file-level decomposition**

Not 100% posix compliant

- **non-sequential writes not supported**
- **Not a replacement for NFS**

HDFS Services

Namenode – Manages the filesystem namespace operations

- File/directory creation/deletion
- Block allocation/removal
- Block locations

Datanode – Stores file blocks on one or more disk partitions

Secondary Namenode – Helper service for merging namespace changes

Services communicate through java RPC, with some functionality exposed through http interfaces

Namenode (NN)

Purpose is similar to dCache PNFS

Keeps track of entire fs image

- The entire filesystem directory structure
- The file block -> datanode mapping
- Block replication level
- ~1GB per 1e6 blocks recommended

Entire namespace is stored in memory, but persisted to disk

- Block locations not persisted to disk
- All namespace requests served from memory
- Fsck across entire namespace is really fast

Namenode Journals

NN fs image is read from disk only once at startup.

Any changes to the namespace (mkdir, rm) are written to one or more journal files (local disk, NFS, ...)

Journal is periodically merged with the fs image

Merging can temporarily require extra memory to store two copies of fs image at once.

Secondary NN

The name is misleading... this is NOT a backup namenode or hot spare namenode. It does NOT respond to namespace requests.

Optional checkpoint server for offloading the NN journal -> fsimage merges

- **Download fs image from namenode (once)**
- **Periodically download journal from namenode**
- **Merge journal and fs image**
- **Uploaded merged fs image back to namenode**

Contents of merged fsimage can be manually copied to NN in case of namenode corruption or failure.

Datanode (DN)

Purpose is similar to dCache pool

Stores file block metadata and file block contents in one or more local disk partitions. Datanode scales well with # local partitions

- **Some sites have used 40+ individual disks/partitions**

Sends heartbeat to namenode every 3 seconds

Sends full block report to namenode every hour

Namenode uses report + heartbeats to keep track of which block replicas are still accessible

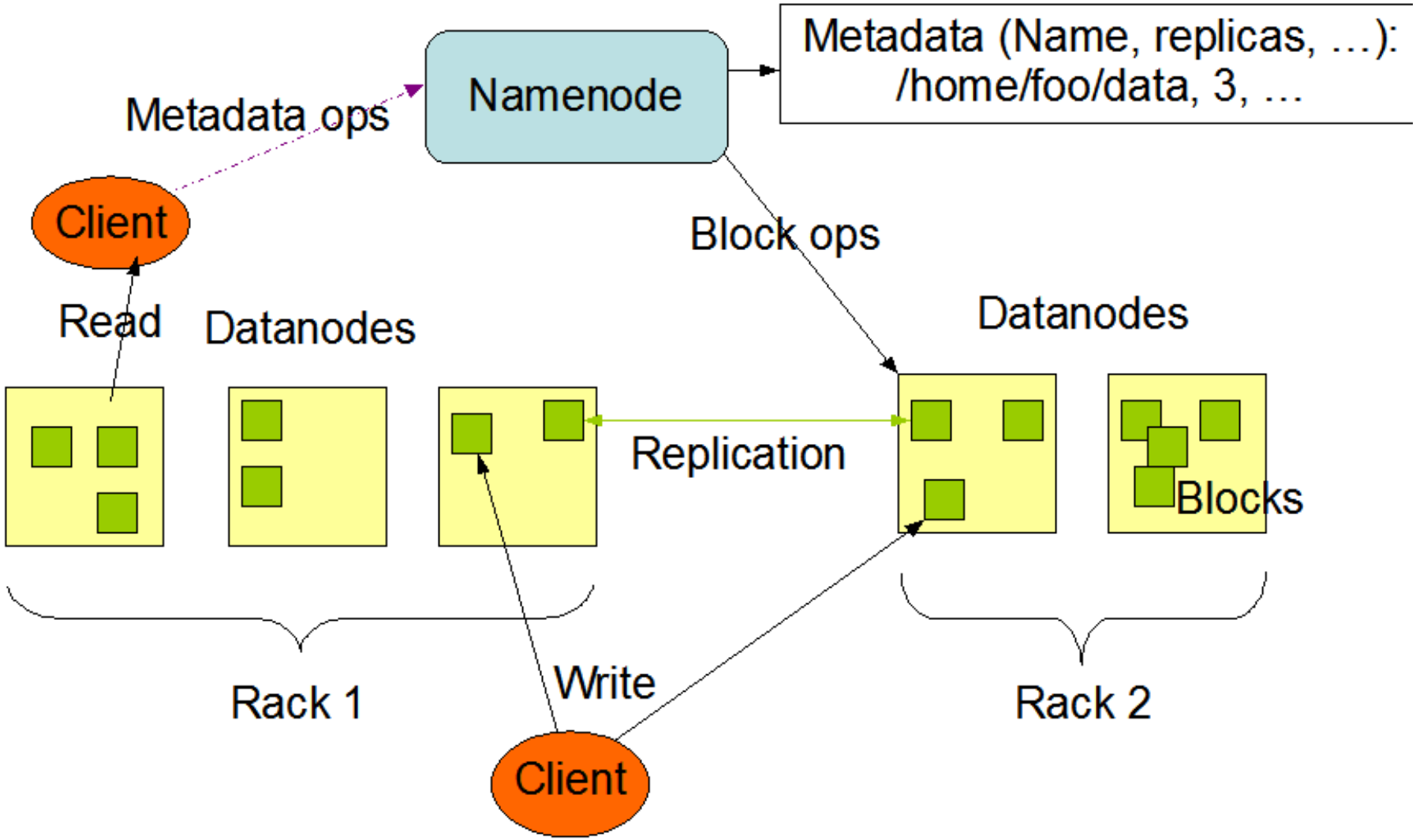
Client access

When a client requests a file, it first contacts the namenode for namespace information.

The namenode looks up the block locations for the requested files, and returns the datanodes that contain the requested blocks

The client contacts the datanodes directly to retrieve the file contents from the blocks on the datanodes

Hadoop Architecture



Native client

A native java client can be used to perform all file and management operations

All operations use native Hadoop java APIs

Example:

```
$ hadoop fs -ls /store/user
$ hadoop fsck /
$ hadoop dfsadmin -refreshNodes
```

FUSE client

FUSE == Filesystem in Userspace

Presents a posix-like interface to arbitrary backend storage systems (ntfs, lustre, ssh)

HDFS fuse module provides posix interface to HDFS using the HDFS APIs. Allows the use of rm, mkdir, cat, and other standard filesystem commands on HDFS.

HDFS does not support non-sequential (random) writes

- **root TFile can't write directly to HDFS fuse, but not really necessary for some VOs**

Random reads are ok

Gridftp/SRM clients

Gridftp could write to HDFS+FUSE with a single stream

Multiple streams will fail due to non-sequential writes

UNL developed a GridFTP dsi module to buffer multiple streams so that data can be written to HDFS sequentially

Bestman SRM can perform namespace operations by using FUSE

- **Running in gateway mode**
- **srmrm, srmls, srmmkdir**
- **Treats hdfs as local posix filesystem**

Installation

https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/HadoopInstallationHandsOn