PITTSBURGH SUPERCOMPUTING CENTER

Hadoop: An Overview

Bryon Gill
Pittsburgh Supercomputing Center

What Is Hadoop?

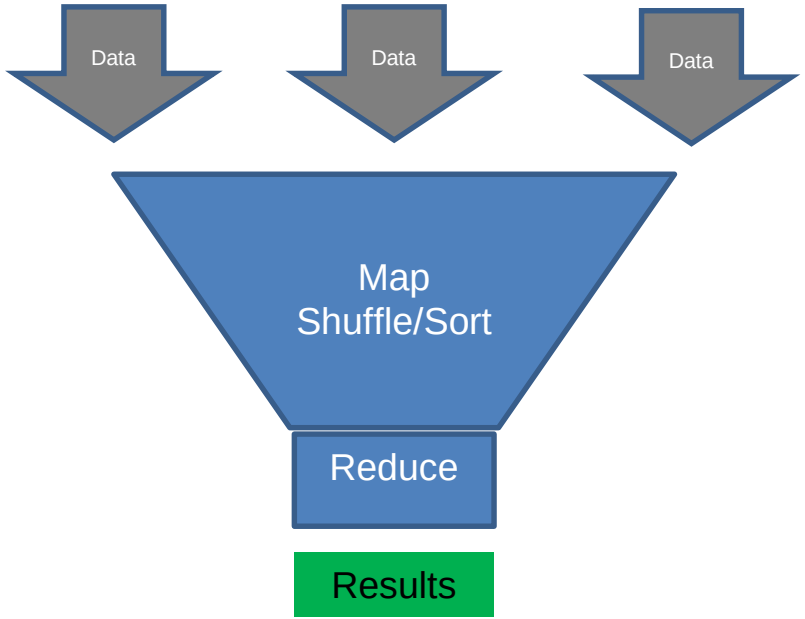
- Programming platform
- Filesystem
- Software ecosystem
- Stuffed elephant

What does Hadoop do?

- Distributes files
  - Replication
  - Closer to the CPU
- Computes
  - Map/Reduce
  - Other

MapReduce

- Map function
  - Maps k/v to intermediate k/v
- Reduce function
  - Shuffle/Sort/Reduce
  - Aggregates results of map



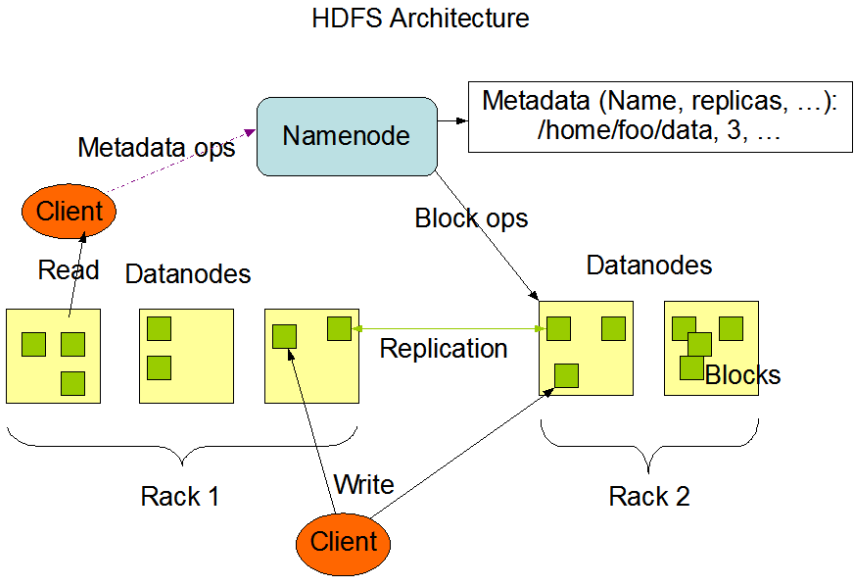
HDFS: Hadoop Distributed File System

- Replication
  - Failsafe
  - Predistribution
- Write Once Read Many (WORM)
  - Streaming throughput
    - Simplified Data Coherency
  - No Random Access (contrast with RDBMS)

HDFS: Hadoop Distributed File System

- Meta filesystem
  - Requires underlying FS
  - Special access commands
  - Exports
    - NFS
    - Fuse
    - Vendor filesystems

HDFS



Source: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

HDFS: Daemons

- Namenode
  - Metadata server
- Datanode
  - Holds blocks
  - Compute node

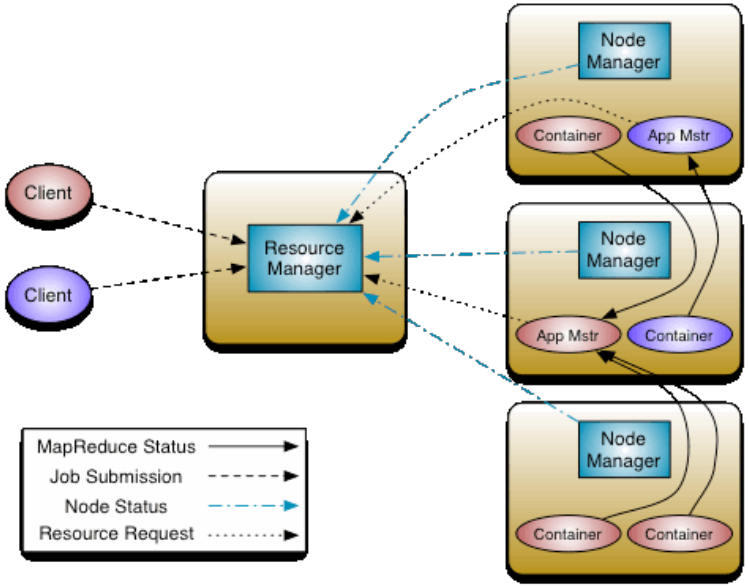
YARN: Yet Another Resource Negotiator

- Programming interface (replaces MapReduce)
- Include MapReduce API (compatible with 1.x)
- Assigns resources for applications

YARN: Daemons

- ResourceManager
  - Applications Manager
  - Scheduler (pluggable)
- NodeManager
  - Worker Node
  - Containers (tasks from ApplicationManager)

YARN



Source: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

Using Hadoop

- Load data to hdfs
  - Fs commands
- Write a program
  - Java
  - Hadoop Streaming
- Submit a job

Fs Commands

- “FTP-style” commands
  - hdfs dfs –put /local/path/myfile /user/$USER/
  - hdfs dfs –cat /user/$USER/myfile # | more
  - hdfs dfs –ls
  - hdfs dfs –get /user/$USER/myfile

Moving Files

```
#on bridges:
hdfs dfs –put /home/training/hadoop/datasets /

# if you don’t have permissions for / (eg. shared cluster)
# you can put it in your home directory
# (making sure to adjust paths in examples):
hdfs dfs –put /home/training/hadoop/datasets
```

Writing a MapReduce Program

- Hadoop Streaming
  - Mapper and reducer scripts read/write stdin/stdout
  - whole line is key, value is null (unless there's a tab)
  - Use builtin utilities (wc, grep, cat)
  - Write in any language (python)
- Java (compile/jar/run)

Simple MapReduce Job (HadoopStreaming)

- cat as mapper
- wc as reducer

```
hadoop jar \
$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming*.jar \
-input /datasets/plays/ -output streaming-out \
-mapper '/bin/cat' -reducer '/usr/bin/wc -l
```

Python MapReduce (HadoopStreaming)

```
hadoop jar
$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming*.jar \
-file ~training/hadoop/mapper.py -mapper mapper.py \
-file ~training/hadoop/reducer.py -reducer reducer.py \
-input /datasets/plays/ -output pyout
```

MapReduce Java: Compile, Jar, Run

```
cp /home/training/hadoop/*.java ./
hadoop com.sun.tools.javac.Main WordCount.java
jar cf wc.jar WordCount*.class
hadoop jar wc.jar WordCount /datasets/compleat.txt output
```

Getting Output

```
hdfs dfs –cat /user/$USER/streaming-out/part-00000 | more

hdfs dfs –get /user/$USER/streaming-out/part-00000
```

Further Exploration

- Write a Hadoop Streaming wordcount program in the language of your choice
- Advanced: Extend the word count example to use a custom input format splitting the text by sentence rather than by line.

Questions?

- Thanks!