Part 2

Part 2

Continue analysis of forensicswiki.org log files to determine:

—*How many web pages could be cached by a Content Distribution Network (CDN)?*

We will:

- Compute the distribution of (*page name*, *page length*) tuples.
- See how a 1-day cache policy impacts savings.

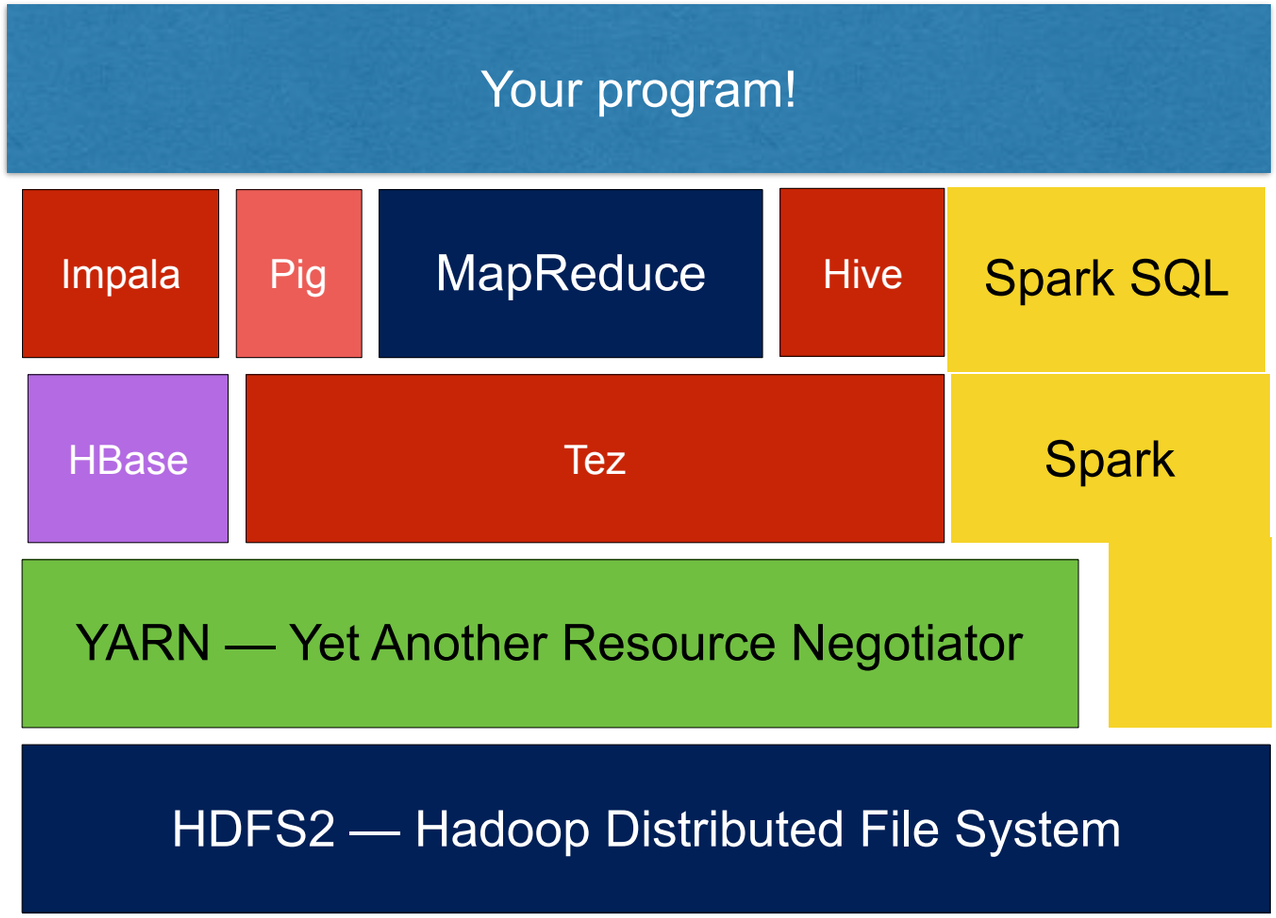
We will also:

- See how to write more reliable Python code with **py.test**.

Hadoop Design and Architecture

Hadoop Architecture: We are using Hadoop Version 2

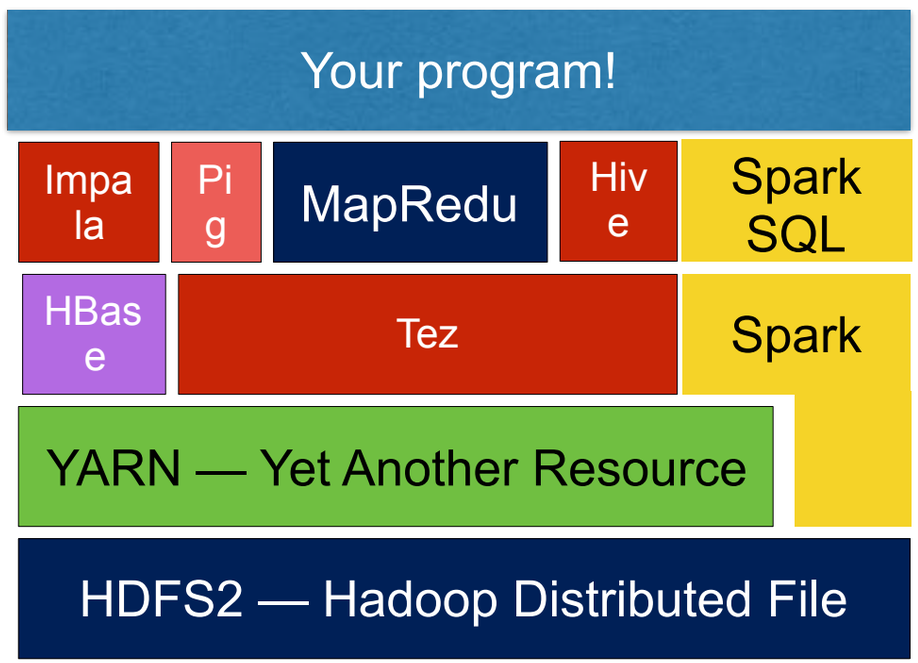
Hadoop is a collection of parts:



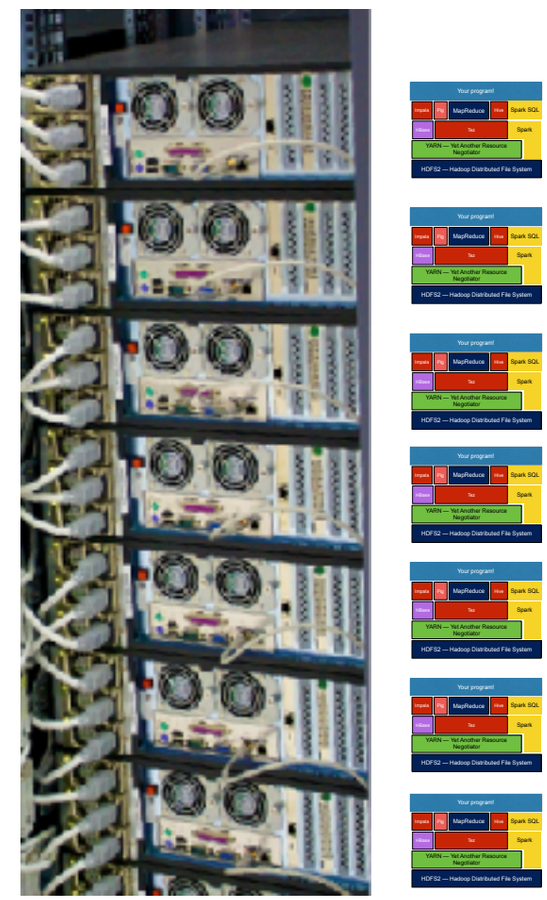
The stack runs on every node in the cluster

Each “node” is a rack-mounted computer:

- Running Linux
- Running multiple JVMs (1 per core)



Rack



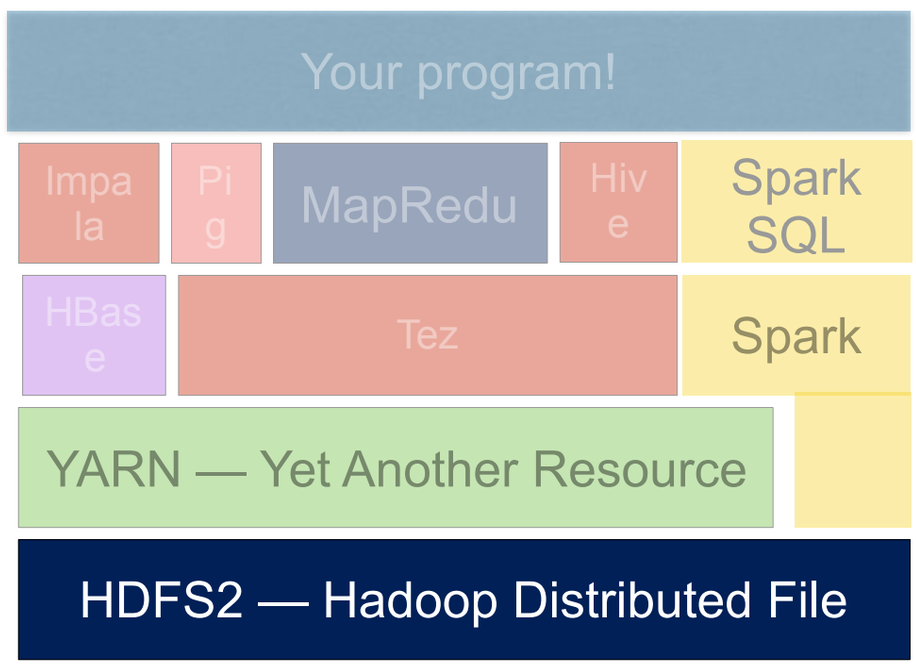
HDFS layer manages storage of data

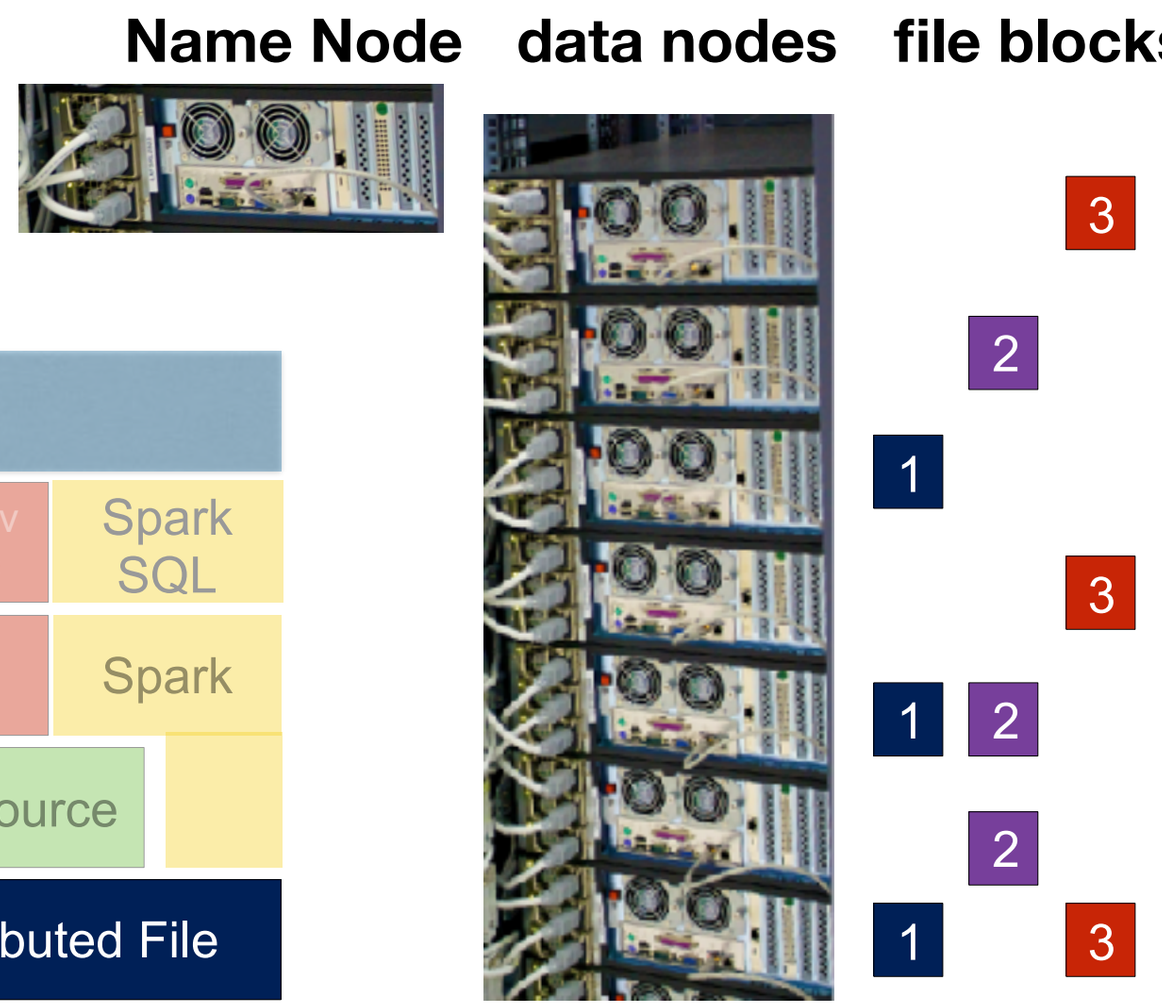
Data:

- split into blocks (64MB each)
- stored redundantly (no RAID)

Name node:

- Tracks location of every file

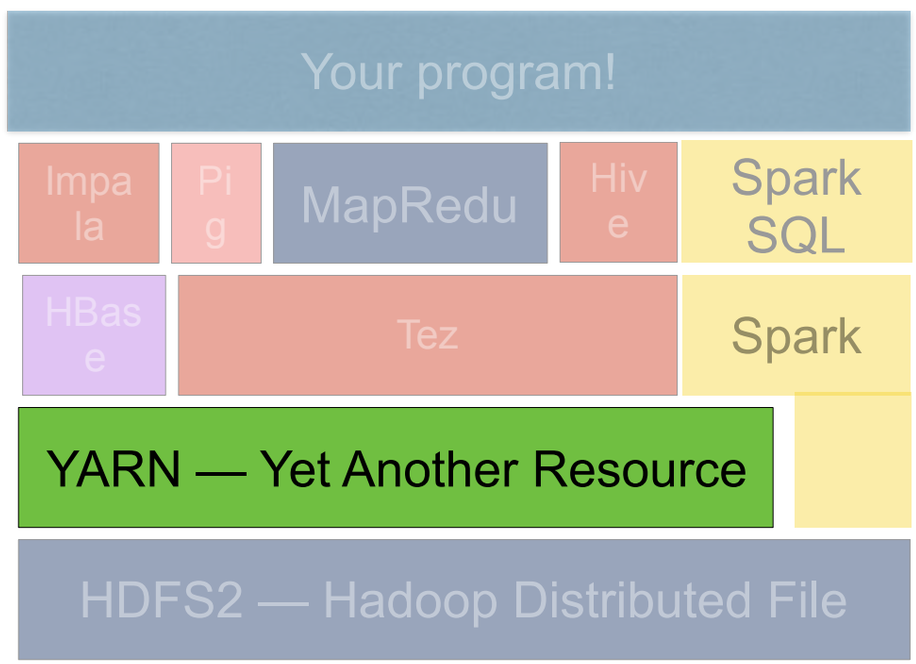




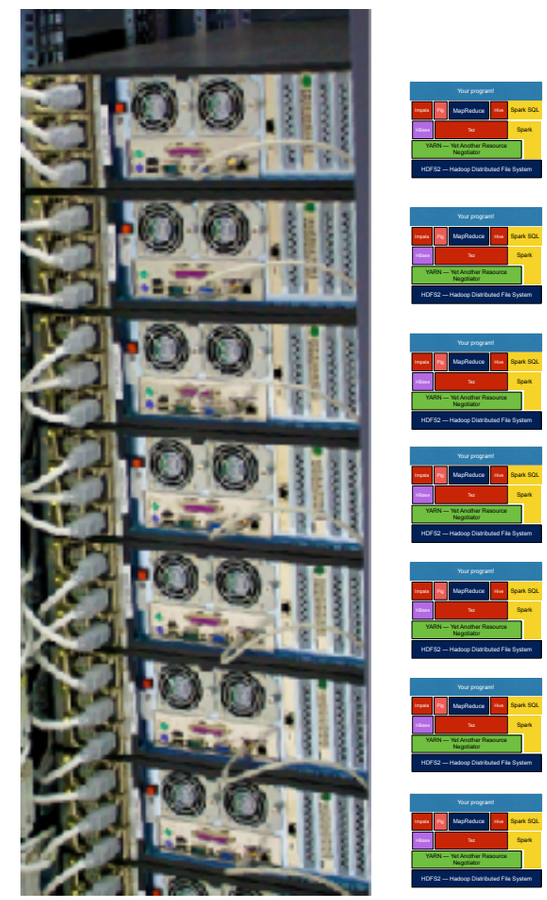
YARN — Yet Another Resource Negotiator
Manages the resources for the entire cluster.

Receives submitted jobs.

Allocates nodes to tasks

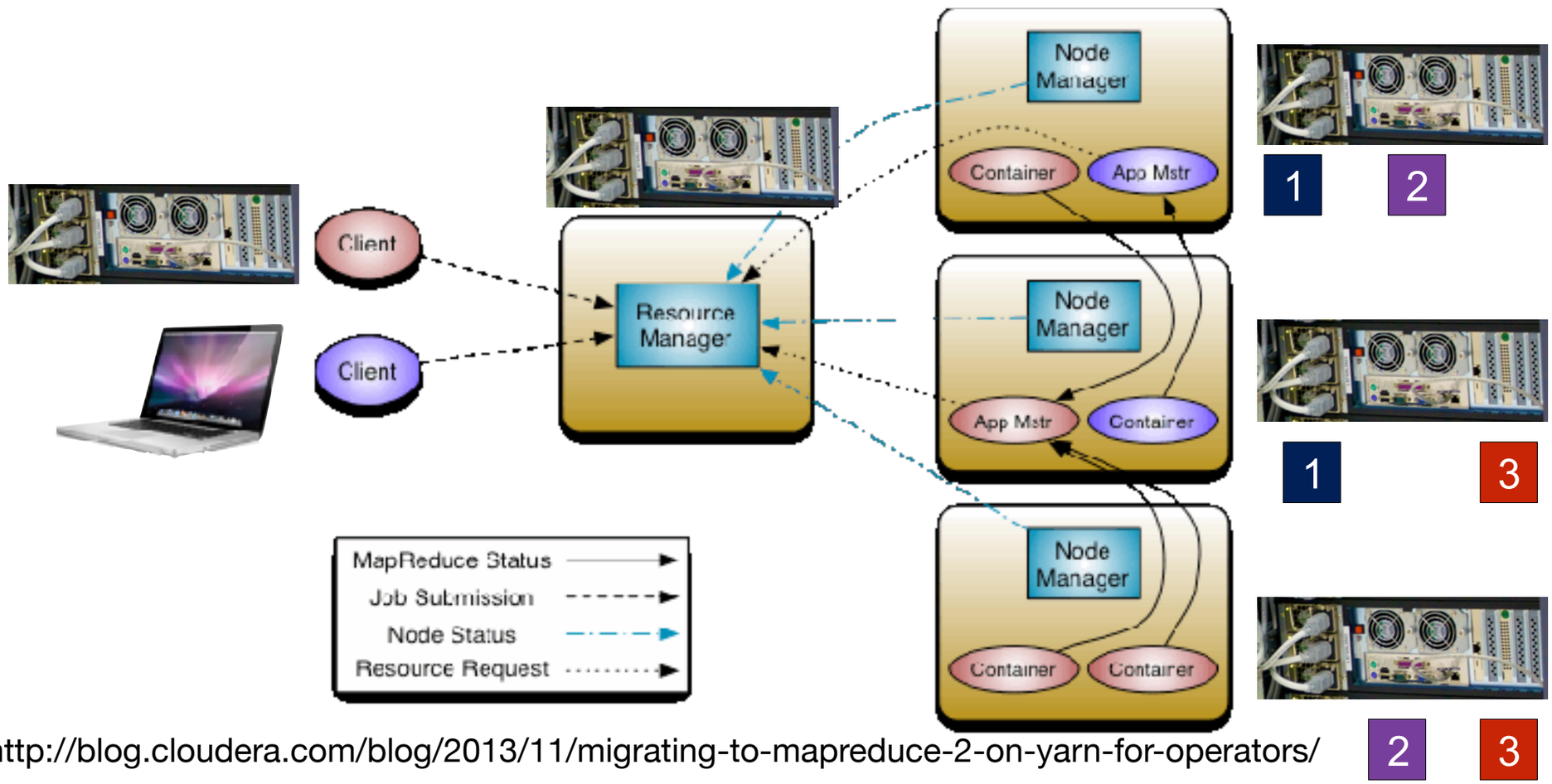


Rack



Yarn receives jobs from “clients” and sends them to the nodes.

Tasks are sent to the nodes that have the data (if possible)



- http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-operators/

YARN is a persistent Java program and a command-line utility.

/usr/bin/yarn — bash script that runs /usr/lib/hadoop-yarn/bin/yarn

/usr/lib/hadoop-yarn/bin/yarn — bash script that runs the appropriate Java class

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

```
[cloudera@quickstart ~]$ yarn
Usage: yarn [--config confdir] COMMAND
where COMMAND is one of:
  resourcemanager -format-state-store   deletes the RMStateStore
  resourcemanager                       run the ResourceManager
  nodemanager                           run a nodemanager on each slave
  timelineserver                        run the timeline server
  rmadmin                               admin tools
  version                               print the version
  jar <jar>                             run a jar file
  application                           prints application(s) report/kill
application
  applicationattempt                    prints applicationattempt(s) report
  container                             prints container(s) report
  node                                  prints node report(s)
  queue                                 prints queue information
  logs                                  dump container logs
  classpath                             prints the class path needed to get the
Hadoop jar and the required libraries
  daemonlog                             get/set the log level for each daemon
  top                                   run cluster usage tool
 or
  CLASSNAME                             run the class named CLASSNAME

Most commands print help when invoked w/o parameters.
[cloudera@quickstart ~]$
```

Basic Yarn commands

“yarn jar” — Submits a Hadoop job:

```
$ export STREAMING_JAR=/usr/lib/hadoop-mapreduce/hadoop-
streaming.jar
$ yarn jar $STREAMING_JAR -mapper mapper.py -reducer reducer.py \
  -input /user/myusername/input -output /user/myusername/
gutenberg_output -file mapper.py -file reducer.py
```

$ yarn top — shows running jobs

- https://hadoop.apache.org/docs/current/hadoop-streaming/HadoopStreaming.html

More commands...

```
$ yarn node -list -all
17/02/07 00:03:44 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-39-51.ec2.internal/172.31.39.51:8032
Total Nodes:1
         Node-Id       Node-State   Node-Http-Address   Number-of-Running-Containers
ip-172-31-39-51.ec2.internal:8041           RUNNING     ip-172-31-39-51.ec2.internal:8042                                0
[hadoop@ip-172-31-39-51 ~]$
```

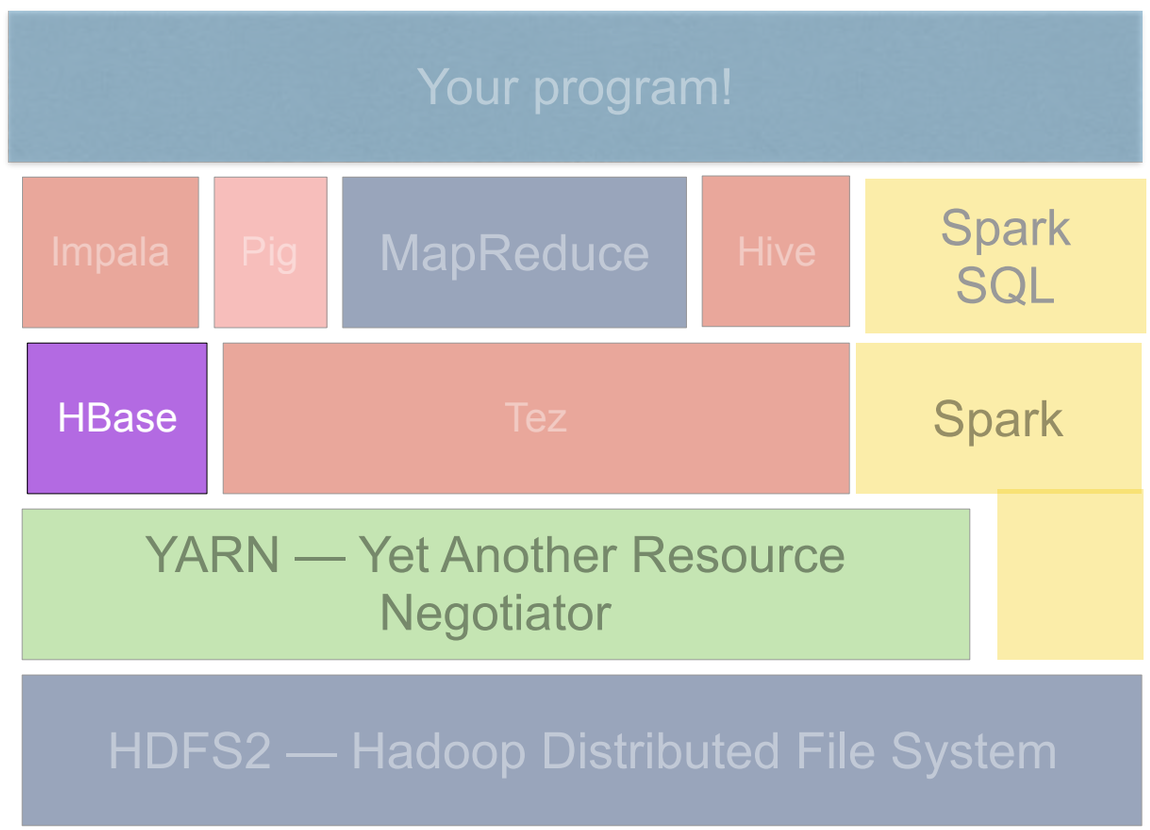
Yarn is a full cluster management system.

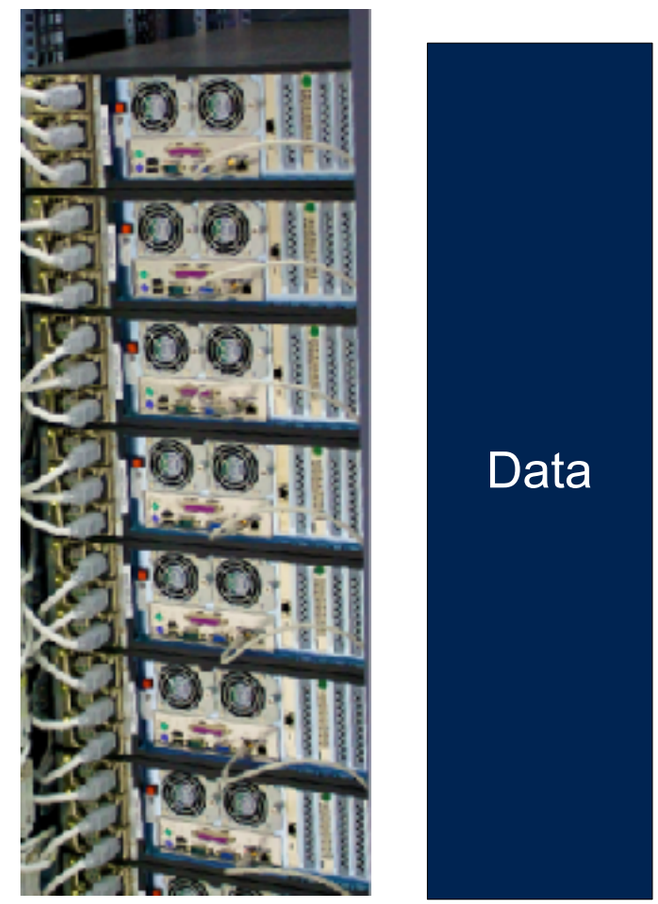
- http://www.slideshare.net/cloudera/introduction-to-yarn-and-mapreduce-2

HBase layer manages a distributed database.

HBase:

- Column-oriented database, stores data in HDFS
- Key-value store
- Based on Google's "Big Table"
- Not an RDBMS



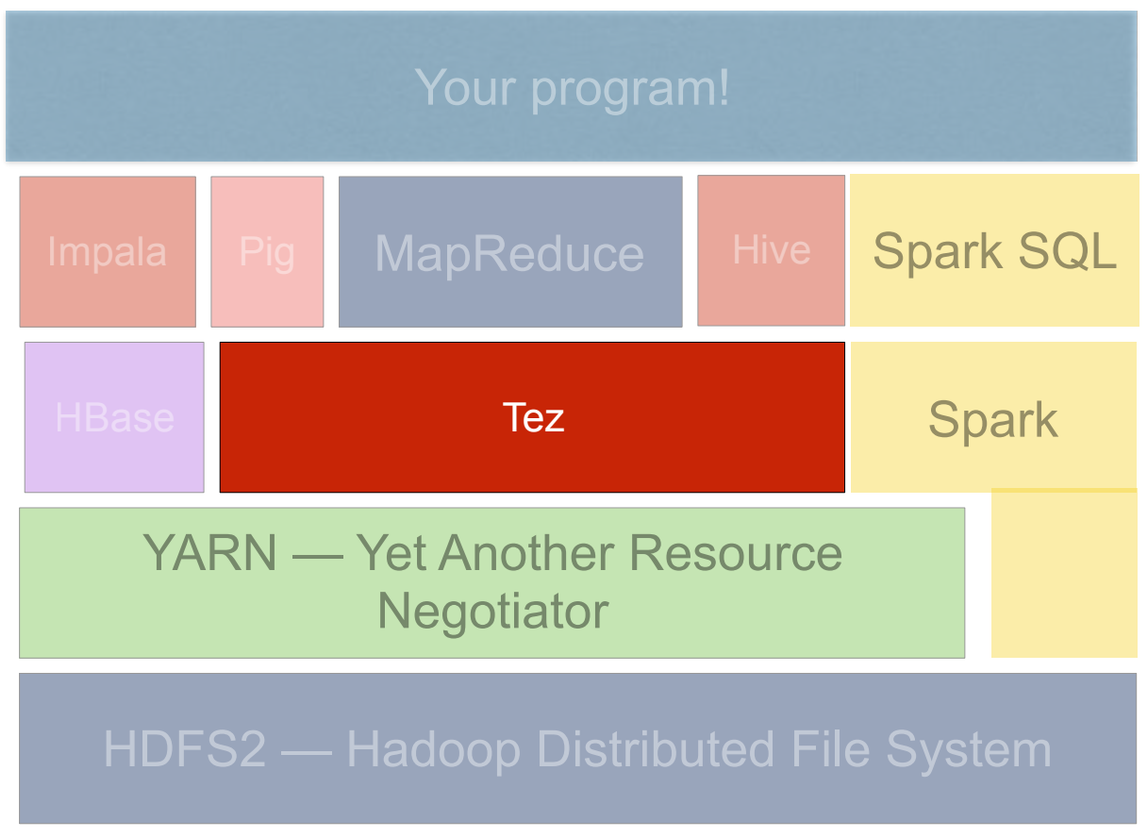


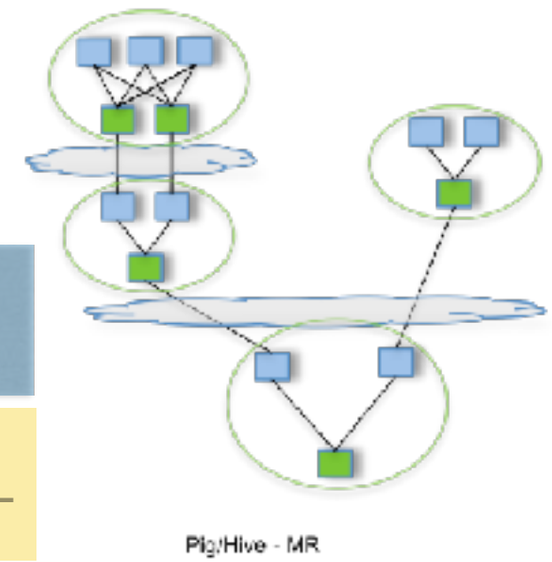
Tez* performs data-flow analysis using “directed-acyclic-graphs”

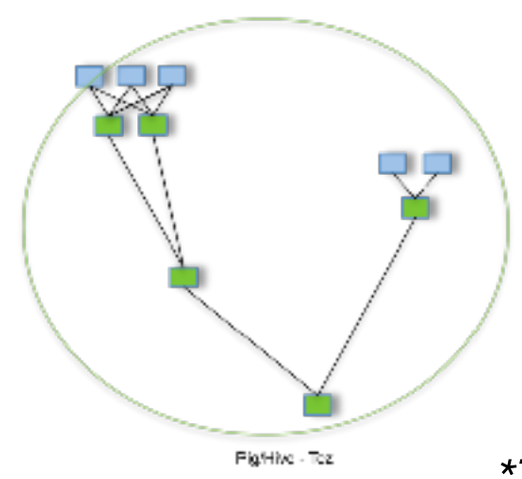
Removes redundant jobs.

Manages workloads

- https://tez.apache.org/
- http://hortonworks.com/hadoop/tez/





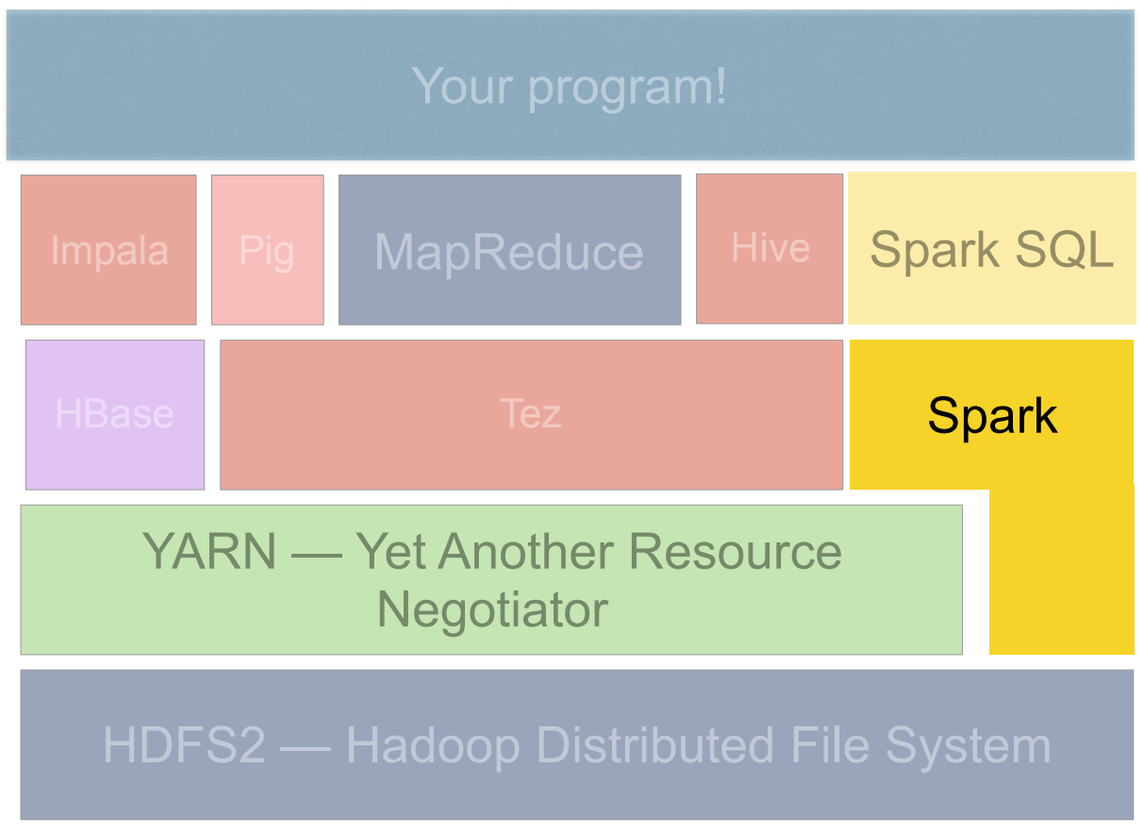


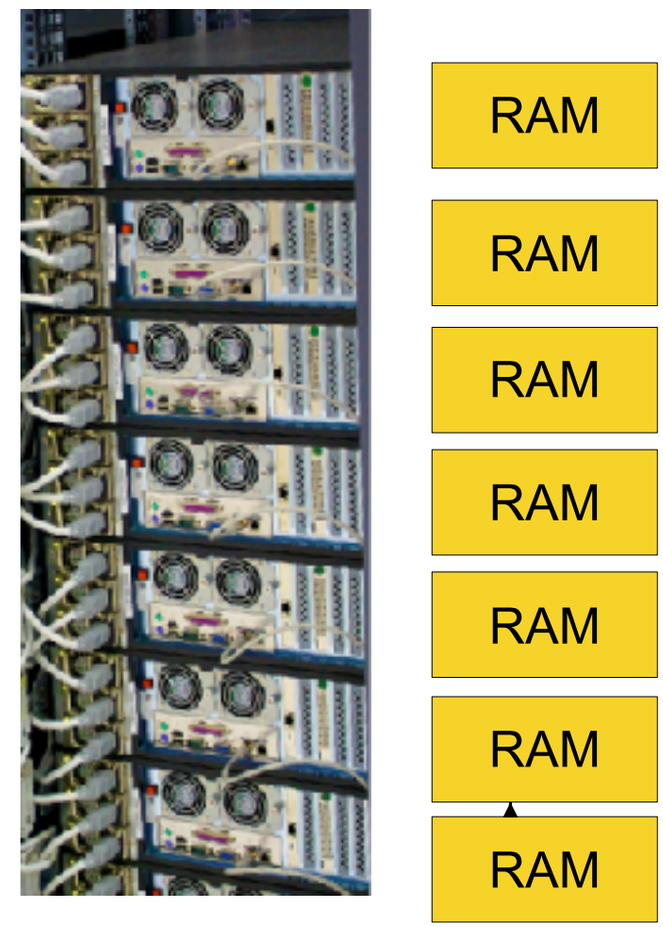
*Tez is hindi for

Spark performs calculations in RAM

Spark:

- Data moves RAM→RAM without being written to disk.
- 5x - 50x faster than MapReduce!
- Easier to program!
- Can run on YARN or on bare metal.

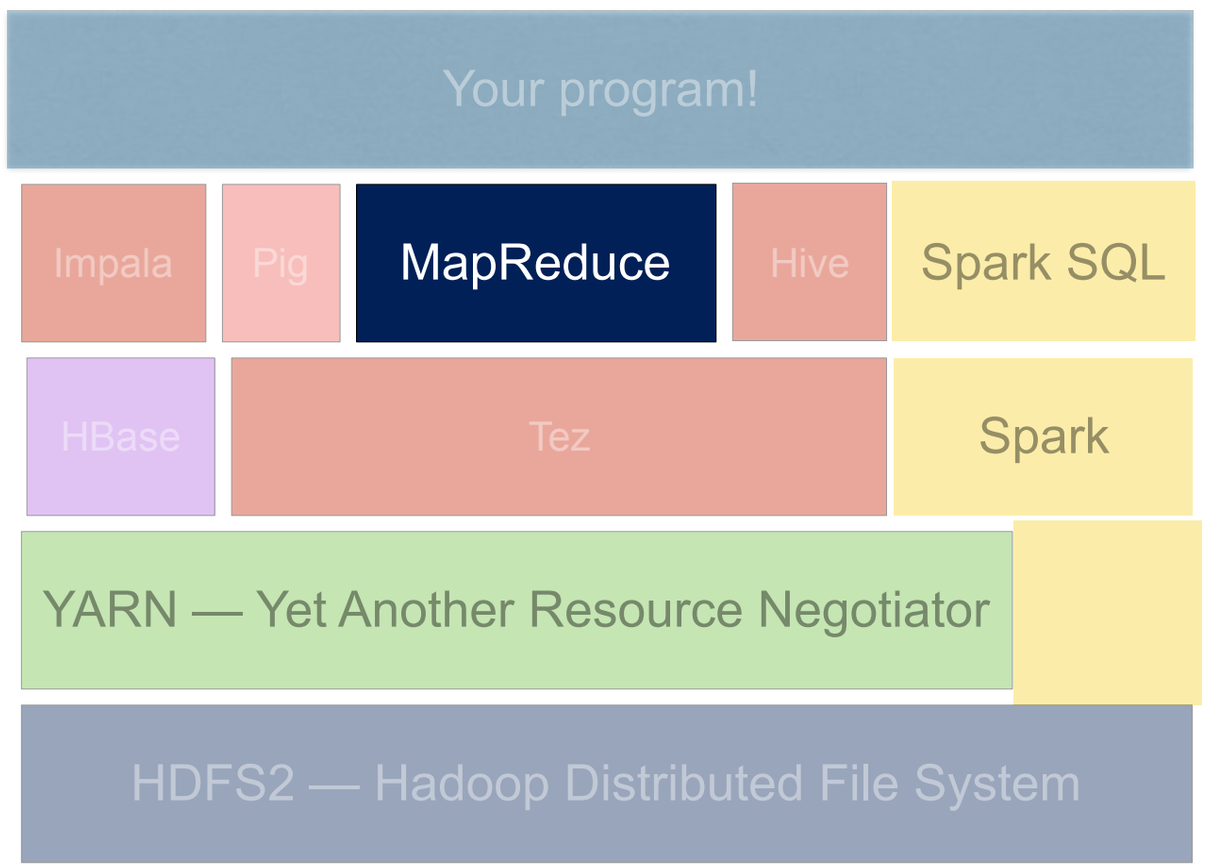




MapReduce — Handles Map/Combiner/Partition/Shuffle/Reduce

Hooks for calling Java programs as mappers, combiners, partitioners, & reducers

Hadoop streaming run as a special Java class.



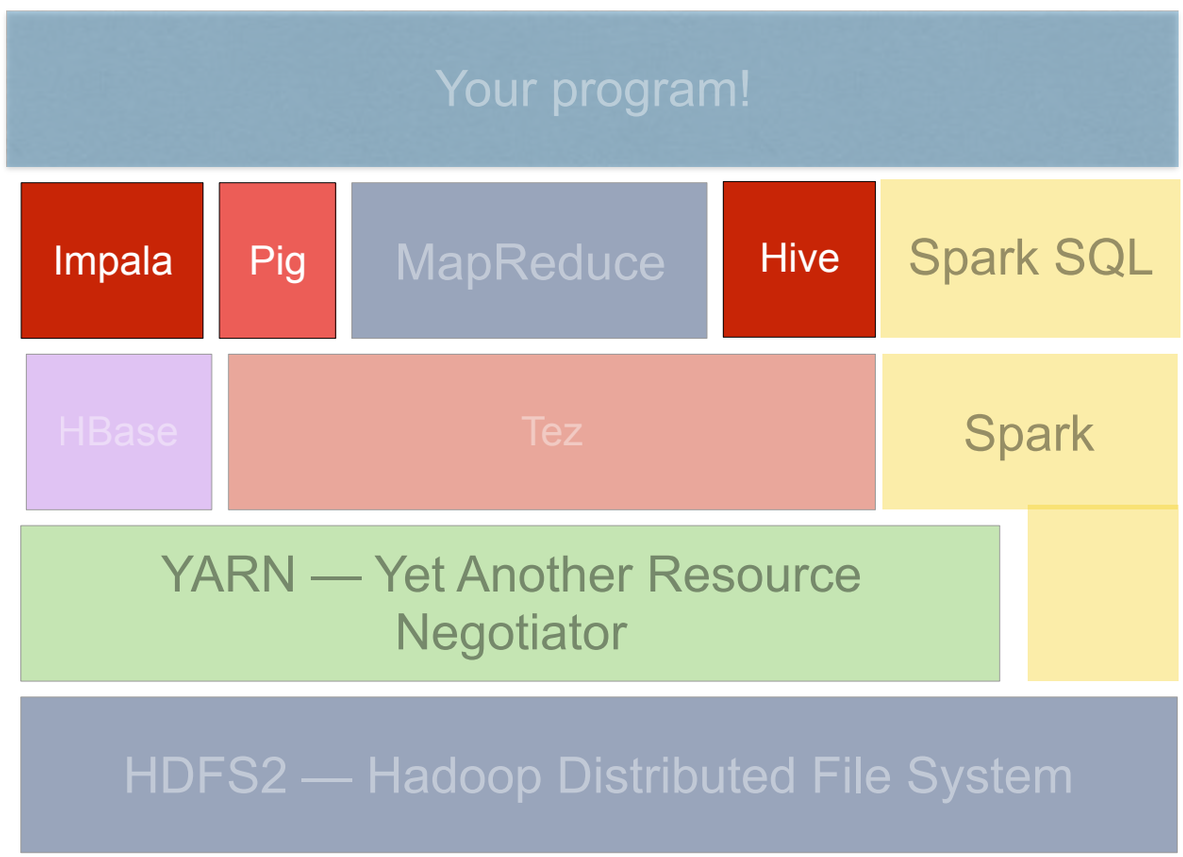
Impala, Pig, Hive, Sqoop & Spark SQL: SQL-interfaces for Hadoop

Impala — transfers between Hadoop and relational databases.

Hive — manages large datasets in HDFS

Pig — Compiles SQL-like queries
in “Pig Latin” to MapReduce jobs

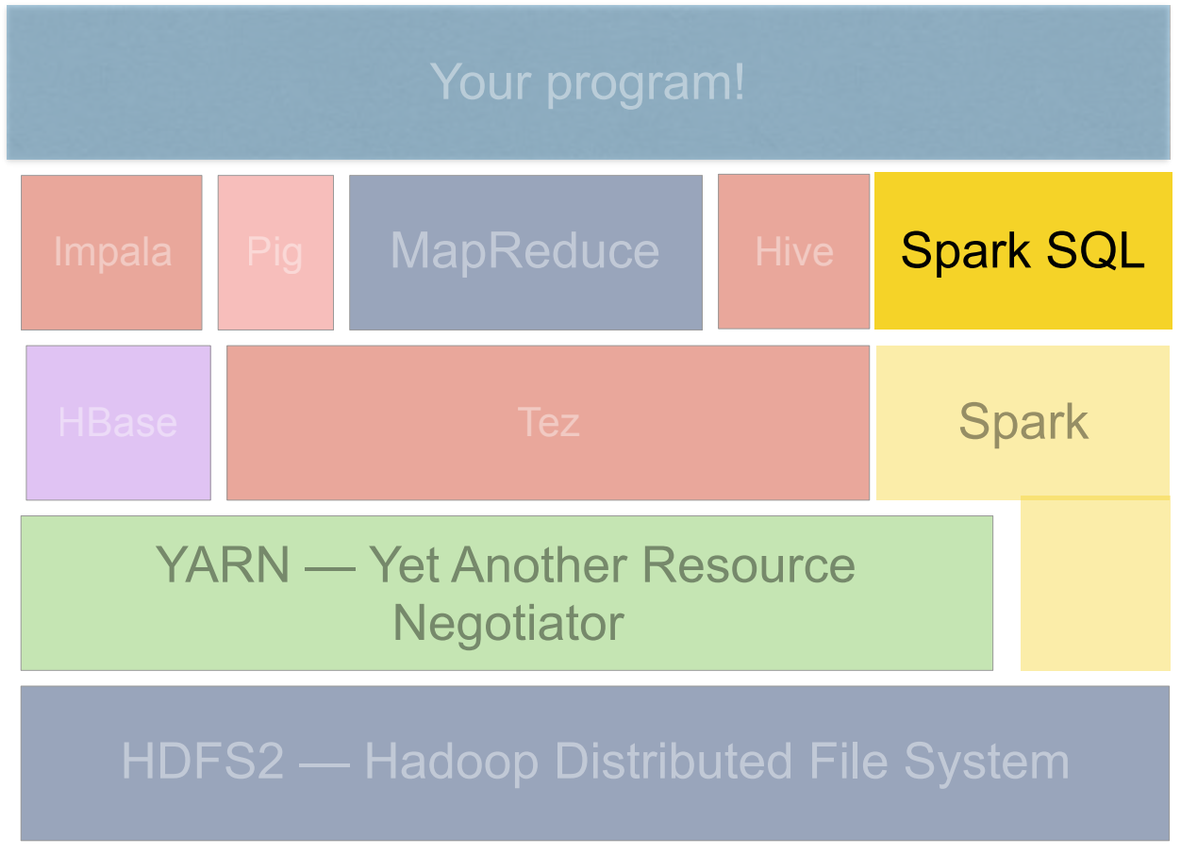
Sqoop — batch interface between SQL and HDFS



SparkSQL

Another SQL-like interface to Spark, HBase, & HDFS

Well integrated with Spark.



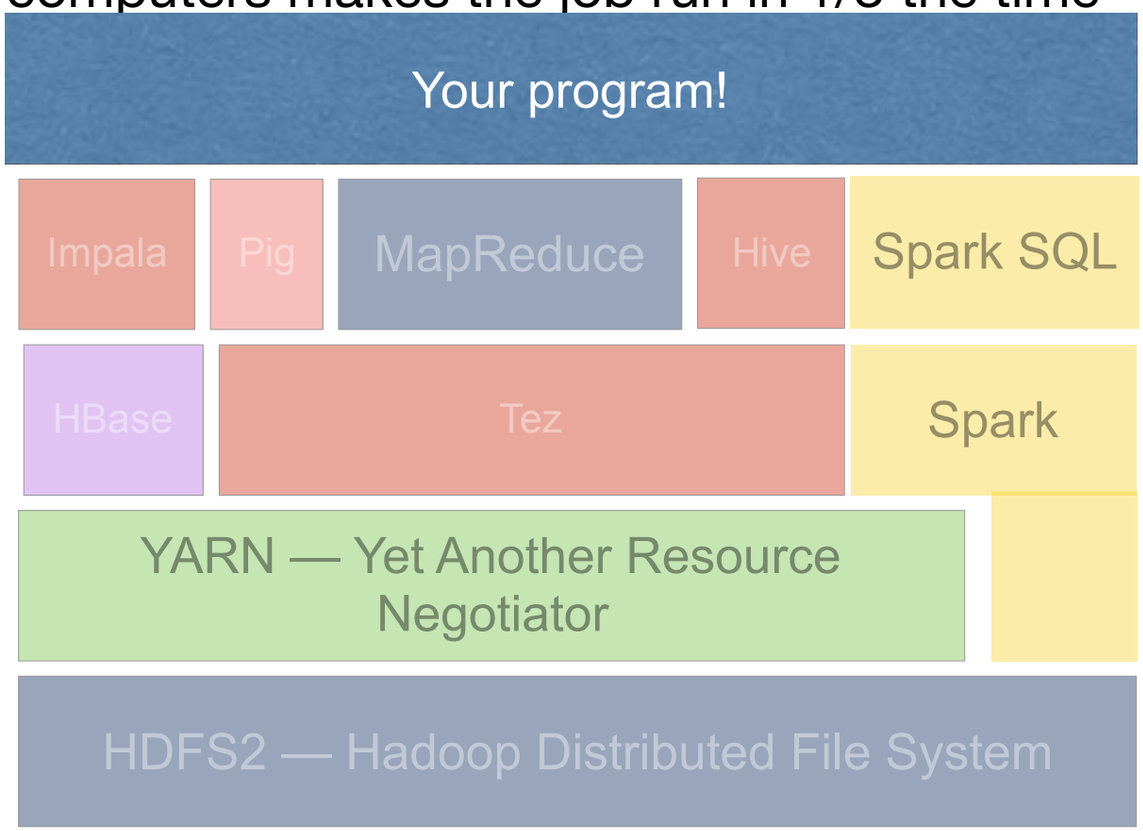
Your program!

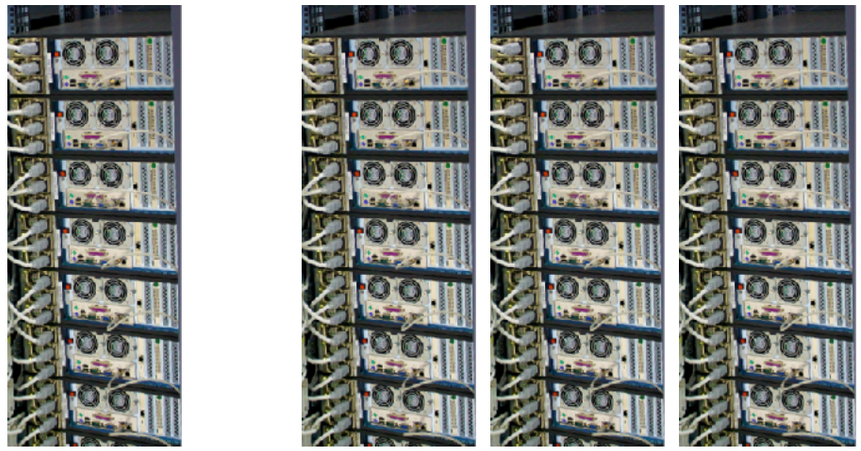
Most jobs consist of multiple operations:

- Fetch data from Database/ HDFS / S3
- Process (multiple MapReduce or Spark steps)
- Save data in HDFS or S3 (if large) or to STDOUT (if small)

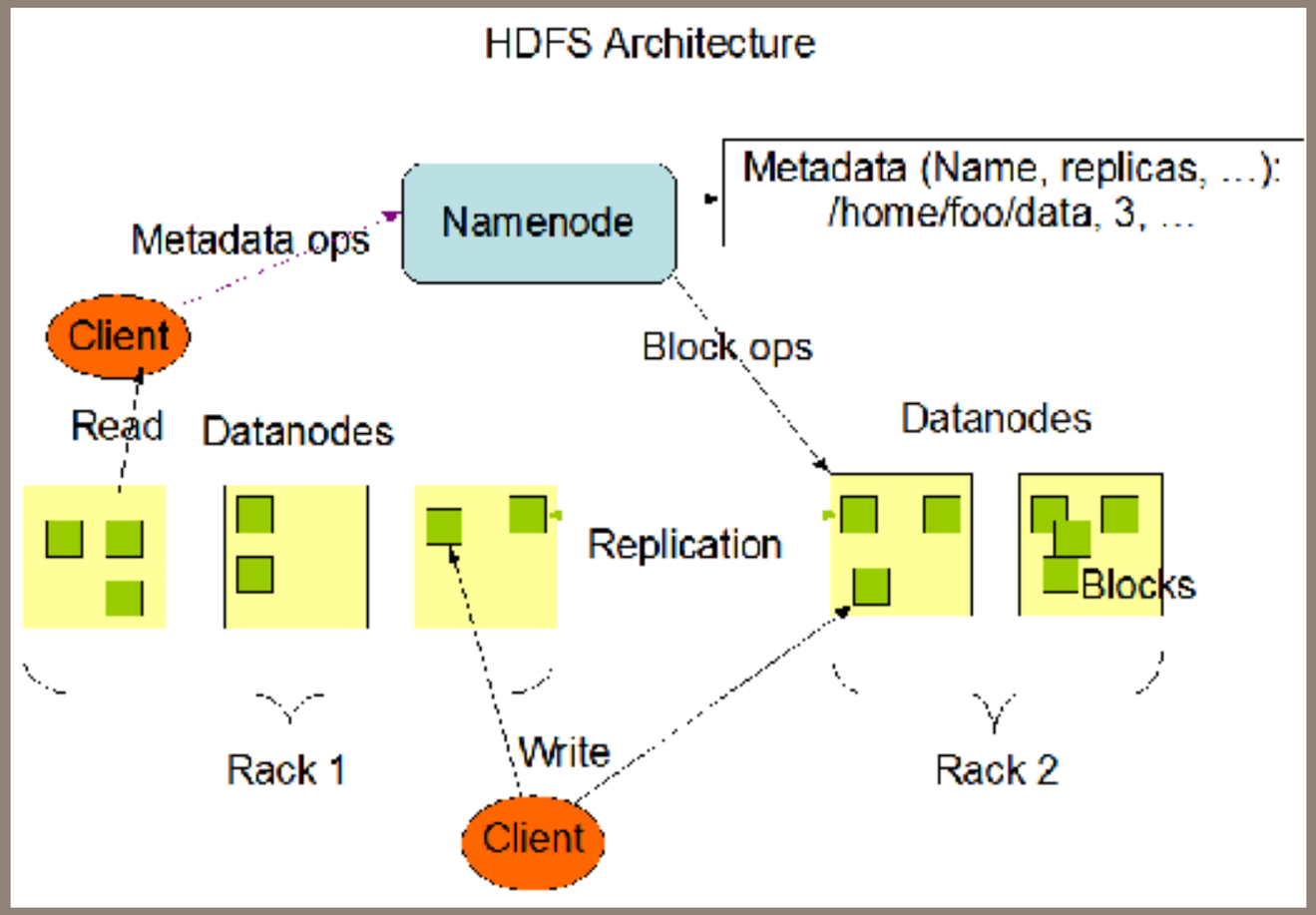
Because you use Hadoop...

- 3x computers makes the job run in 1/3 the time





6 hours **2 hours**



HDFS: The Hadoop Distributed File System

HDFS Design Considerations

Commodity hardware

- No special RAID controllers, high-performance interconnect, etc.

Scale to thousands of nodes.

High aggregate throughput

- One node: 10 min to read 60GB (100 MBB/sec)
- 100 nodes: 10 min to read 6TB (10,000 MB/sec)
  - *Goal is to read 6TB in 10 min, not to read 60GB in 10 seconds*

Designed for batch processing of data:

- Data stored in sequential files.
- Files can be read, written, or appended.
  - *Data cannot be changed after written.*

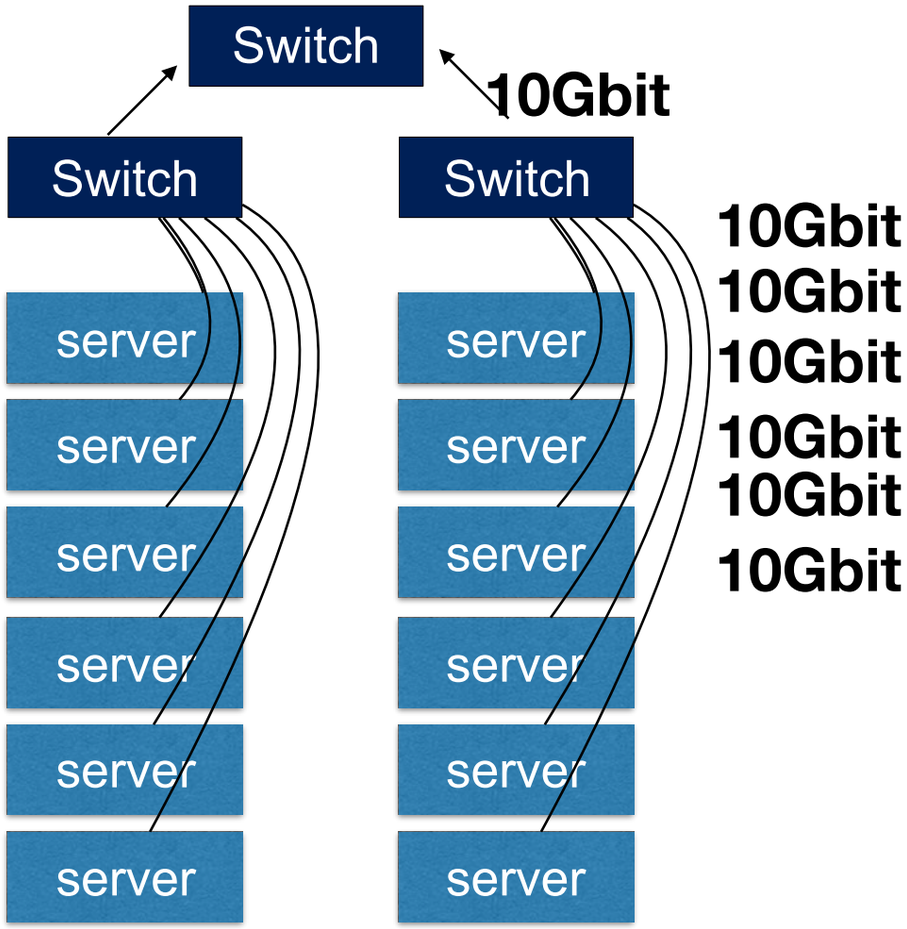
Data Replication — handles failures and parallelism

- If a drive or server fails, clients read from another server.

Rack-aware architecture: Computers are deployed in racks

Rack-aware architecture

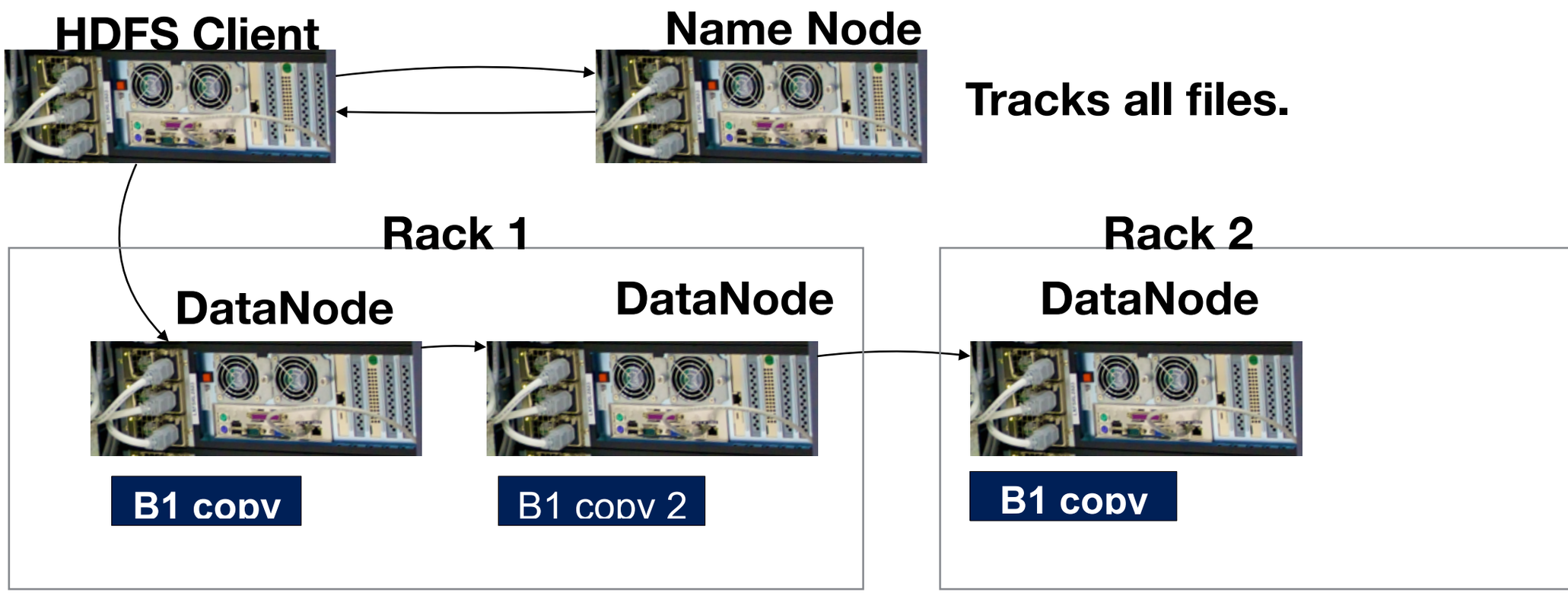
- More bandwidth within racks than between racks



HDFS Architecture: Writing

Each file => 1 or more blocks

Each block => 1 or more nodes (replication factor)



Heartbeat — Resiliency

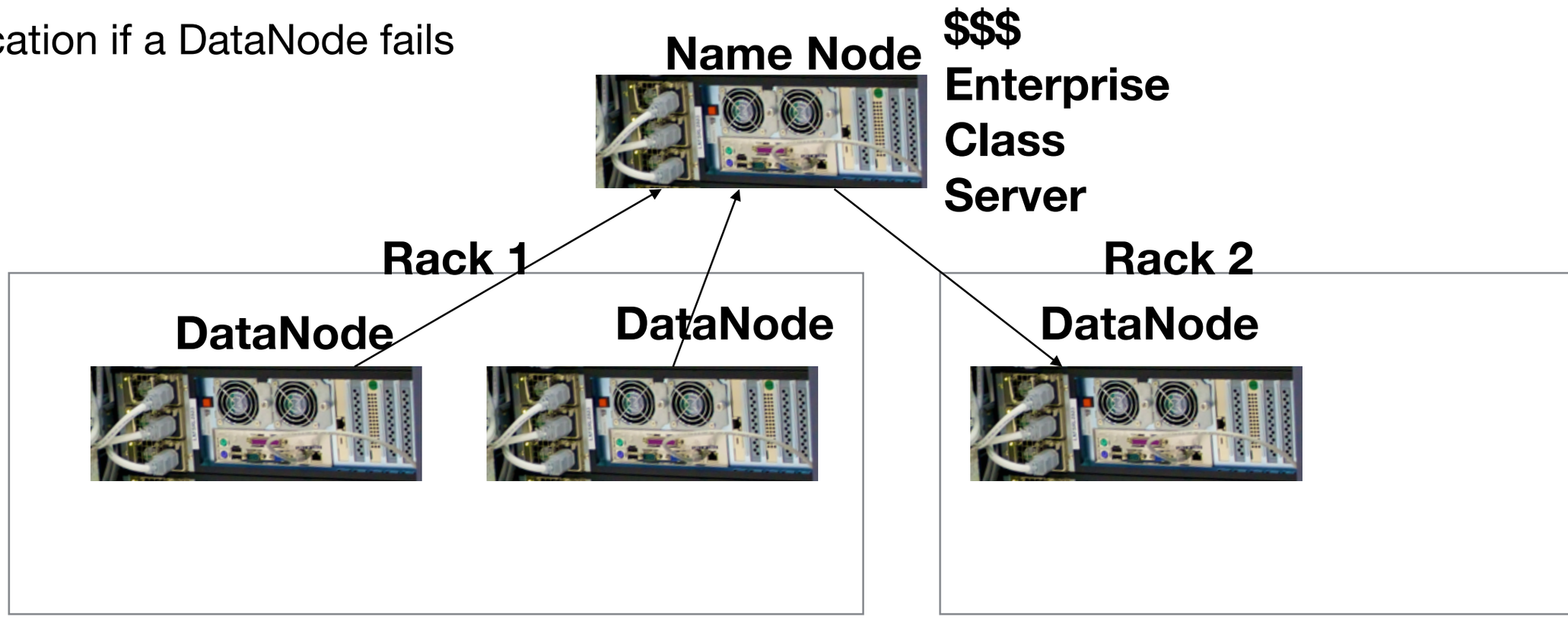
dfs.heartbeat.interval — default 3 seconds

Reports:

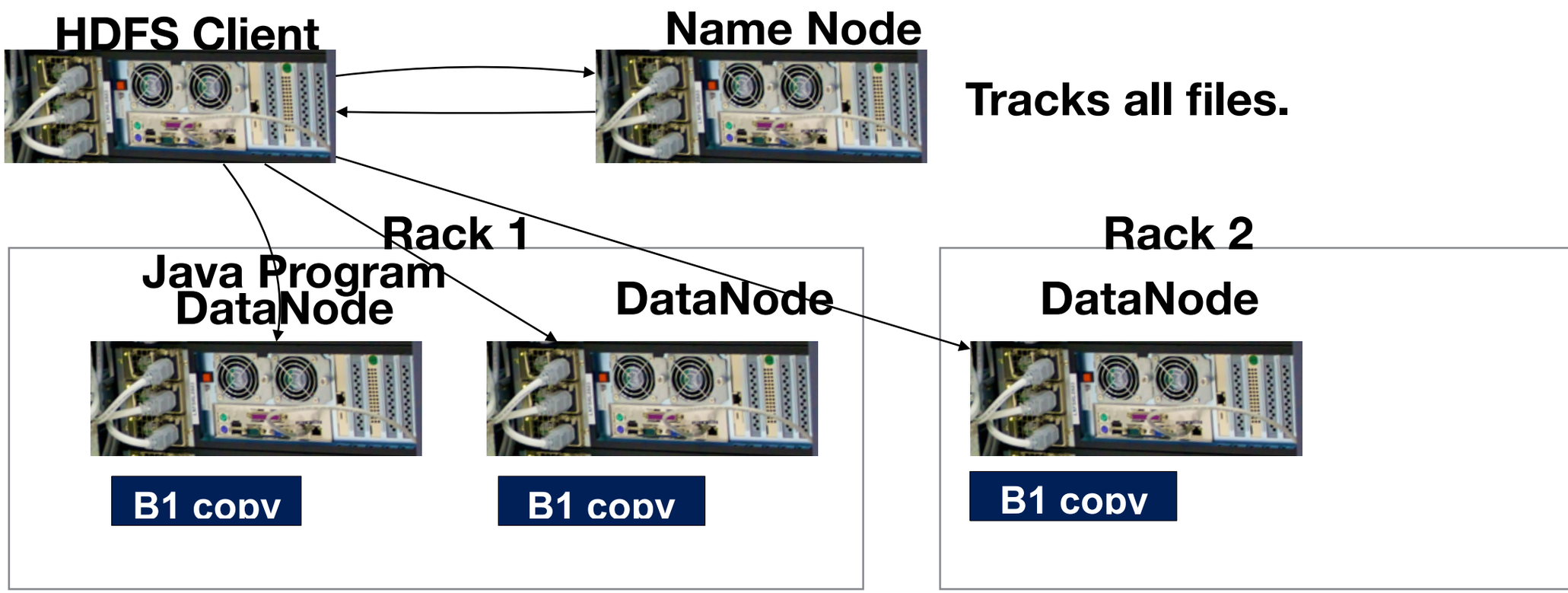
- "I'm alive!"
- Block Report every 10th heartbeat:

Name Node:

- Schedules replication if a DataNode fails



HDFS Architecture: Reading



Replication — Determines Performance and Overhead

Replication = 1 (no replication)

- Cloudera VM
- Amazon EMR with 1 data node
- No backup against failure.

Replication = 3 (typical)*

- Each block stored 3 times
- Name Node keeps track which blocks on which servers
  - —*If a server fails, Name Node tells replicants to make a copy.*

Block Size — default is 64MiB

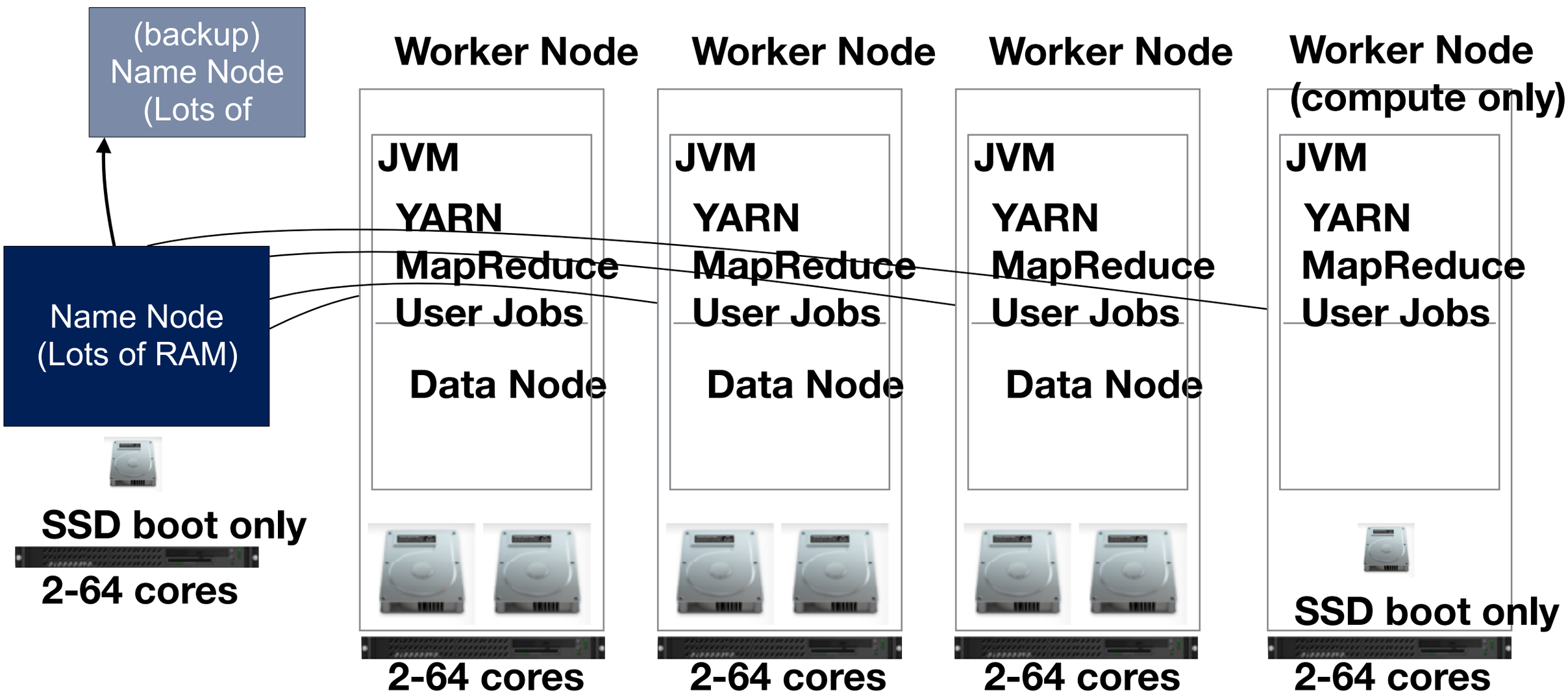
- Don't store small files!
- Hadoop "Sequence File" stores lots of small files in a single "file."

∴ Storing a 1GiB file with replication 3 takes 3GiB (1024 ÷ 64 x 3 = 48 blocks)

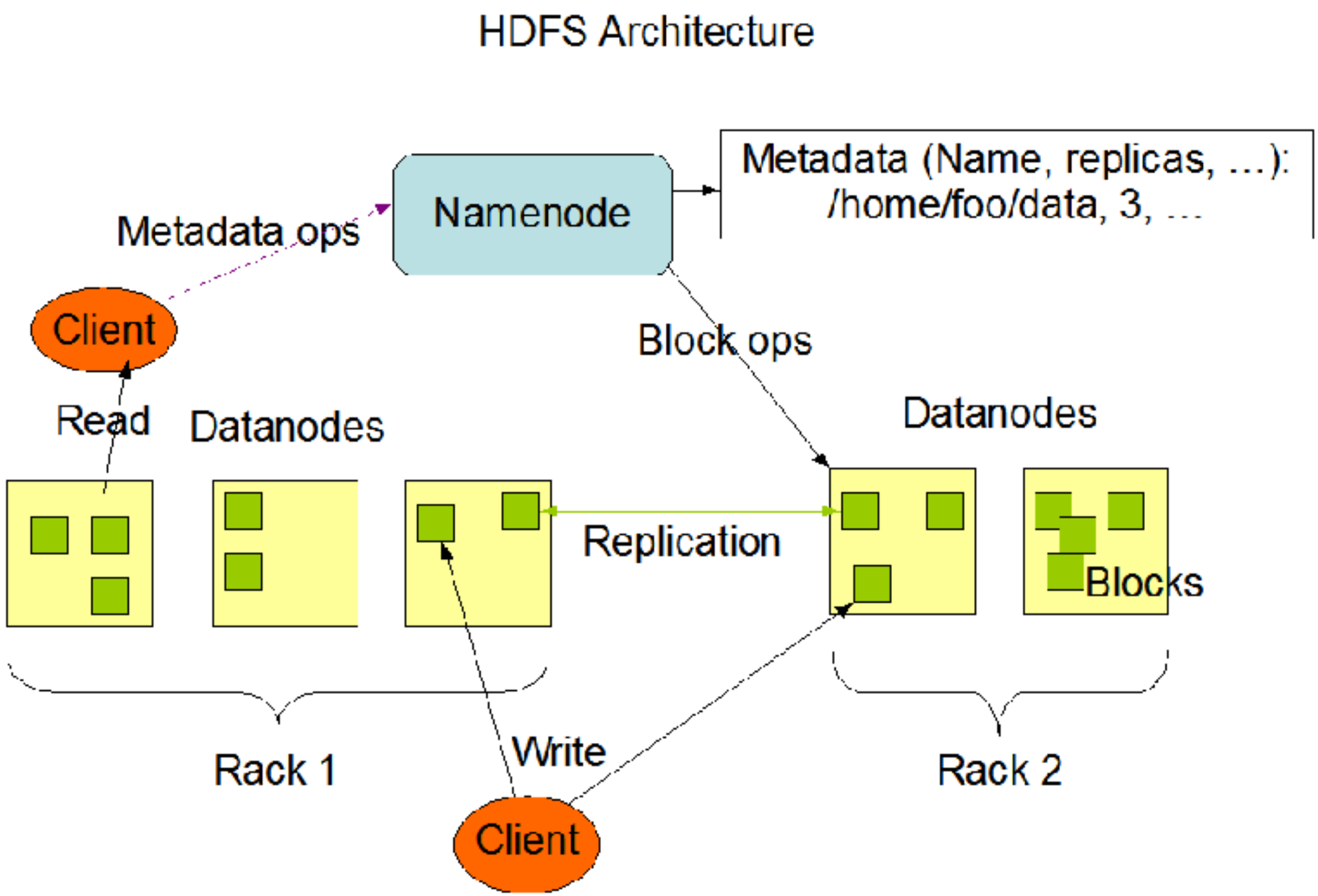
∴ Storing 100 1MiB files with replication 3 takes 19.2GiB (100 x 3 = 300 blocks)

- * HDFS3 will use erasure coding instead of replication for low I/O files. http://blog.cloudera.com/blog/2015/09/introduction-to-hdfs-erasure-coding-in-apache-hadoop/

Remember, each "worker node" is potentially both a data node and a compute node.



Apache's "official" HDFS architecture diagram

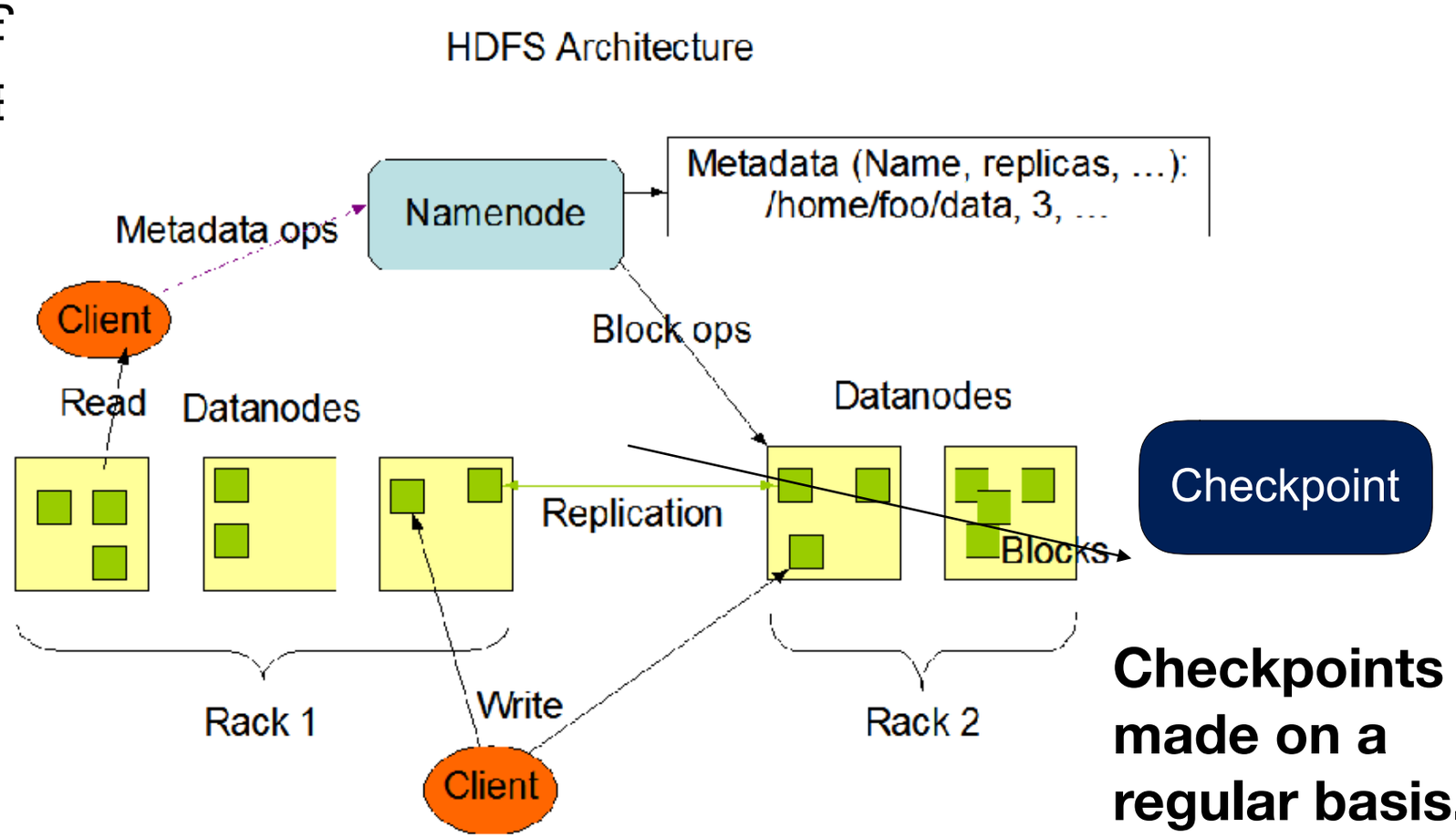


NameNode scaling issues: The Small Files Problem

NameNode keeps entire file system in memory!

Each file requires:

- A file inode reference (≈150 bytes) + a block reference (≈150 bytes) = ≈300 bytes
- A million files — 300MB (a typical laptop has
- A billion files — 300GE
- Ten billion files? — 3TE
  - —*Not happening!*



Checkpoints made on a regular basis.

https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Other problems with small files

More files means:

- More wasted space (64MB block size)
- More map tasks (each file needs its own map task)
- Map tasks are largely wasted.

Therefore — Keep your files big!

- Combine many small files into a few big files:
  - —Hadoop SequenceFiles — splittable, compressible, for working with large amounts of binary data. Java Only.
  - —Hadoop MapFiles — Indexed sequence files.
  - —Hadoop Archive Files
- HBase — Database abstraction from a few large files
- S3 — Easily handles lots of files

If your total data isn't >500MB, you probably shouldn't be using Hadoop!

Interacting with HDFS: “hdfs dfs” command

“hdfs” is the primary command for interacting with the Hadoop file system

Local file system	hadoop file system
`cat`	`hdfs dfs -cat`
`cp`	`hdfs dfs -cp`
`df`	`hdfs dfs -df`
`du`	`hdfs dfs -du`

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

Make a remote directory; copy files; list them; cat them

```
$ hdfs dfs -rm -R tmp
rm: `tmp': No such file or directory
$ cat README.md
This is a readme file.
one
two
three.
Just another file

$ hdfs dfs -ls tmp
ls: `tmp': No such file or directory
$ hdfs dfs -mkdir tmp
$ hdfs dfs -put README.md tmp/
$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   1 hadoop hadoop         58 2015-12-10 19:41 tmp/README.md
$ hdfs dfs -cat tmp/README.md
This is a readme file.
one
two
three.
Just another file

$
```

Informational Commands

```
[hadoop@ip-172-31-39-51 ~]$ hdfs dfs -ls /
Found 3 items
drwxrwxrwt   - hdfs hadoop          0 2017-02-05 18:11 /tmp
drwxr-xr-x   - hdfs hadoop          0 2017-02-05 18:11 /user
drwxr-xr-x   - hdfs hadoop          0 2017-02-05 18:11 /var
[hadoop@ip-172-31-39-51 ~]$

[hadoop@ip-172-31-39-51 ~]$ hdfs dfs -du
587214  tmp
[hadoop@ip-172-31-39-51 ~]$
```

hdfs dfsadmin — administrative commands

```
[hadoop@ip-172-31-39-51 ~]$ hdfs dfsadmin -report
Configured Capacity: 74033651712 (68.95 GB)
Present Capacity: 73748590592 (68.68 GB)
DFS Remaining: 73741402112 (68.68 GB)
DFS Used: 7188480 (6.86 MB)
DFS Used%: 0.01%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (1):

Name: 172.31.39.51:50010 (ip-172-31-39-51.ec2.internal)
Hostname: ip-172-31-39-51.ec2.internal
Decommission Status : Normal
Configured Capacity: 74033651712 (68.95 GB)
DFS Used: 7188480 (6.86 MB)
Non DFS Used: 285061120 (271.86 MB)
DFS Remaining: 73741402112 (68.68 GB)
DFS Used%: 0.01%
DFS Remaining%: 99.61%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Tue Feb 07 00:10:35 UTC 2017


[hadoop@ip-172-31-39-51 ~]$
```

Tuning HDFS for performance and robustness

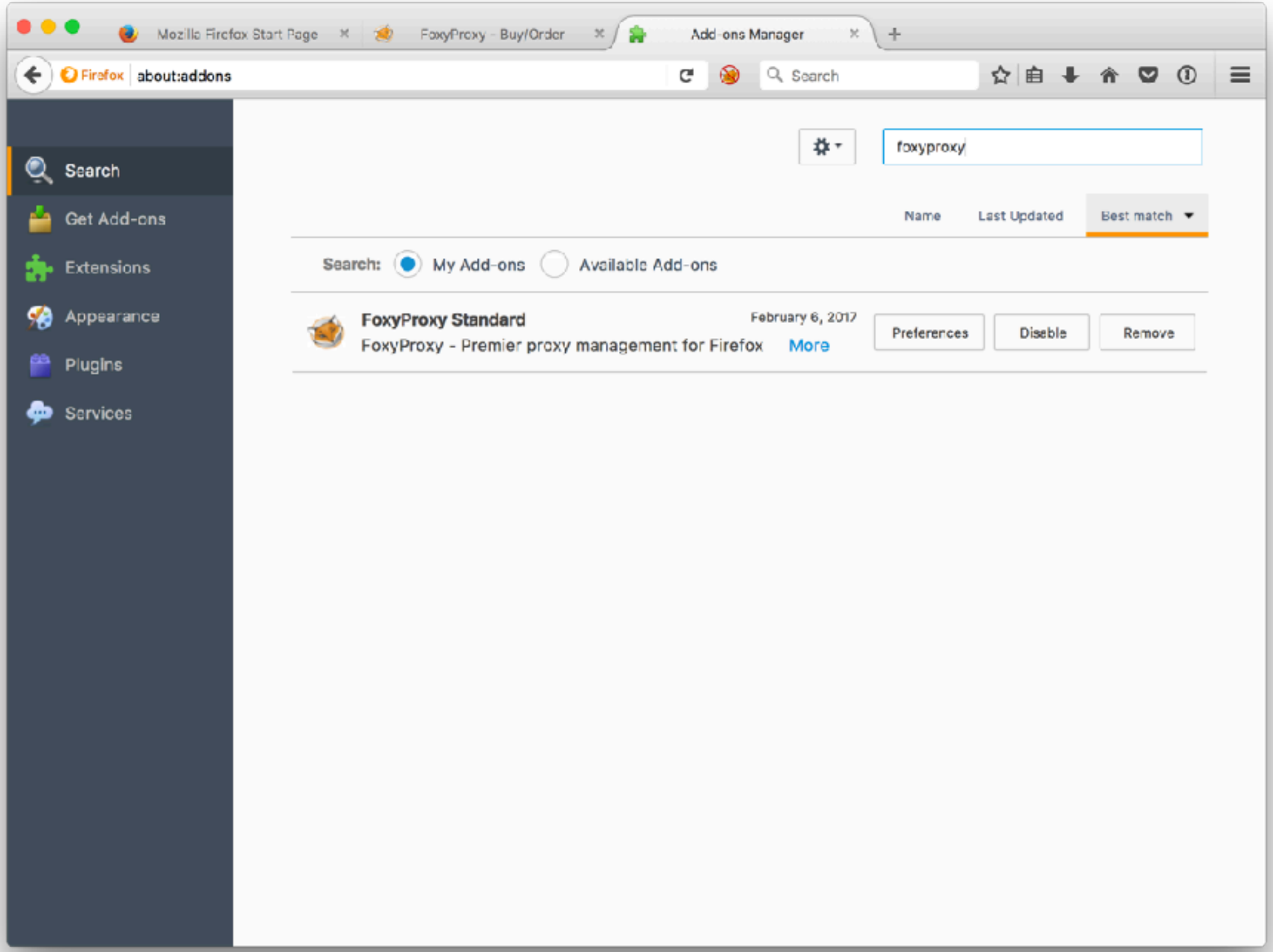
File system configuration parameters: /etc/hadoop/conf/hdfs-site.xml

```
<configuration>

  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///mnt/hdfs,file:///mnt1/hdfs</value>
  </property>

  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///mnt/namenode,file:///mnt1/namenode</value>
  </property>
...

  <property>
    <name>dfs.datanode.available-space-volume-choosing-policy.balanced-space-preference-fraction</name>
    <value>1.0</value>
  </property>

  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.datanode.available-space-volume-choosing-policy.balanced-space-threshold</name>
    <value>10737418240</value>
  </property>

  <property>
    <name>dfs.datanode.du.reserved</name>
    <value>536870912</value>
  </property>
```

https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml

—"You can tune a file system, but you can tune a fish." — UNIX Man Page, circa

Install FoxyProxy Standard



"View Cluster Details"

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#

Services | Resource Groups | Simson Garfinkel | N. Virginia | Support

Amazon EMR

- Cluster list
- Security configurations
- VPC subnets
- Help

Create cluster | View details | Clone | Terminate

Filter: All clusters | Filter clusters ... | 11 clusters (all loaded)

	Name	ID	Status	Creation time (UTC
	EMR 4.8.3 MRJOB Demo	j-QDDOQXJKFW3P	Waiting Cluster ready	2017-02-05 13:05 (
	EMR 5.2.1 MRJOB Demo	j-1Z4OB7ZVQVHJD	Terminated User request	2017-02-05 13:04 (
	EMR 5.2.1 MRJOB Demo	j-3HTDI2HB57JAN	Terminated User request	2017-02-05 13:04 (
	EMR5.2.1	j-2VZKRRDIPY2HI	Terminated	2017-02-01 22:28 (

Summary

Master public DNS: ec2-52-86-169-133.compute-1.amazonaws.com

Termination protection: Off Change

Tags: -- View All / Edit

Hardware

Master: Running 1 m3.xlarge (Spot: .2)

Core: --

Task: --

View cluster details | View monitoring details

Steps | Add Step

Name	Status	Start time (UTC-5)
Setup hadoop debugging	Completed	2017-02-05 13:12 (UTC-5)

Feedback | English | © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back! | Refresh Firefox...

"Enable Web Connection"

Mozilla Firefox Start Page | Install Add-on | 1Password Extension | EMR – AWS Console

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#clu

Services | Resource Groups | Simson Garfinkel | N. Virginia | Support

Amazon EMR

- Cluster list
- Security configurations
- VPC subnets
- Help

Add step | Resize | Clone | Terminate | AWS CLI export

Cluster: EMR 4.8.3 MRJOB Demo Waiting Cluster ready after last step completed.

Enable Web Connection – Resource Manager ... (View All)

Master public DN: ec2-52-86-169-133.compute-1.amazonaws.com SSH

Tags: -- View All / Edit

Summary

ID: j-QDDOQXJKFW3P
Creation date: 2017-02-05 13:05 (UTC-5)
Elapsed time: 1 day, 6 hours
Auto-terminate: No
Termination protection: Off Change

Configuration Details

Release label: emr-4.8.3
Hadoop distribution: Amazon 2.7.3
Applications: --
Log URI: s3://aws-logs-489362128722-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled

Network and Hardware

Feedback | English | © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back! Refresh Firefox...

Open an SSH tunnel…

Mozilla Firefox Start Page | Install Add-on | 1Password Extension | EMR – AWS Console

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#clu

Search

Enable Web Connection

Setup Web Connection

Hadoop, Ganglia, and other applications publish user interfaces as web sites hosted on the master node. For security reasons, these web sites are only available on the master node's local web server.

To reach the web interfaces, you must establish an SSH tunnel with the master node using either dynamic or local port forwarding. If you establish an SSH tunnel using dynamic port forwarding, you must also configure a proxy server to view the web interfaces.

Step 1: Open an SSH Tunnel to the Amazon EMR Master Node - Learn more

Windows | Mac / Linux

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.
2. To establish an SSH tunnel with the master node using dynamic port forwarding, type the following command. Replace ~/mucha.pem with the location and filename of the private key file (.pem) used to launch the cluster.

```
ssh -i ~/mucha.pem -ND 8157 hadoop@ec2-52-86-169-133.compute-1.amazonaws.com
```

   Note: Port 8157 used in the command is a randomly selected, unused local port.
3. Type yes to dismiss the security warning.

Step 2: Configure a proxy management tool - Learn more

Chrome | Firefox

1. Download and install the standard version of FoxyProxy from: http://foxyproxy.mozdev.org/downloads.html
2. Restart your browser after installing FoxyProxy.

Close

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox...

Create the foxyproxy-settings.xml

Mozilla Firefox Start ... | Install Add-on | 1Password Extension | EMR – AWS Console | FoxyProxy - Downloa...

https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#clu

Search

Enable Web Connection

Chrome | Firefox

1. Download and install the standard version of FoxyProxy from:
   http://foxyproxy.mozdev.org/downloads.html
2. Restart your browser after installing FoxyProxy.
3. Using a text editor create a file named foxyproxy-settings.xml containing the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<foxyproxy>
    <proxies>
        <proxy name="emr-socks-proxy" id="2322596116" notes="" fromSubscription="false" enabled="true"
mode="manual" selectedTabIndex="2" lastresort="false" animatedIcons="true" includeInCycle="true" color="#0055E5"
proxyDNS="true" noInternalIPs="false" autoconfMode="pac" clearCacheBeforeUse="false" disableCache="false"
clearCookiesBeforeUse="false" rejectCookies="false">
            <matches>
                <match enabled="true" name="*ec2*.amazonaws.com*" pattern="*ec2*.amazonaws.com*" isRegEx="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />
                <match enabled="true" name="*ec2*.compute*" pattern="*ec2*.compute*" isRegEx="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />
                <match enabled="true" name="10.*" pattern="http://10.*" isRegEx="false" isBlackList="false"
isMultiLine="false" caseSensitive="false" fromSubscription="false" />
                <match enabled="true" name="*10*.amazonaws.com*" pattern="*10*.amazonaws.com*" isRegEx="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />
                <match enabled="true" name="*10*.compute*" pattern="*10*.compute*" isRegEx="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />
                <match enabled="true" name="*.compute.internal*" pattern="*.compute.internal*" isRegEx="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />
                <match enabled="true" name="*.ec2.internal*" pattern="*.ec2.internal*" isRegEx="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />
            </matches>
            <manualconf host="localhost" port="8157" socksversion="5" isSocks="true" username="" password=""
domain="" />
        </proxy>
    </proxies>
</foxyproxy>
```

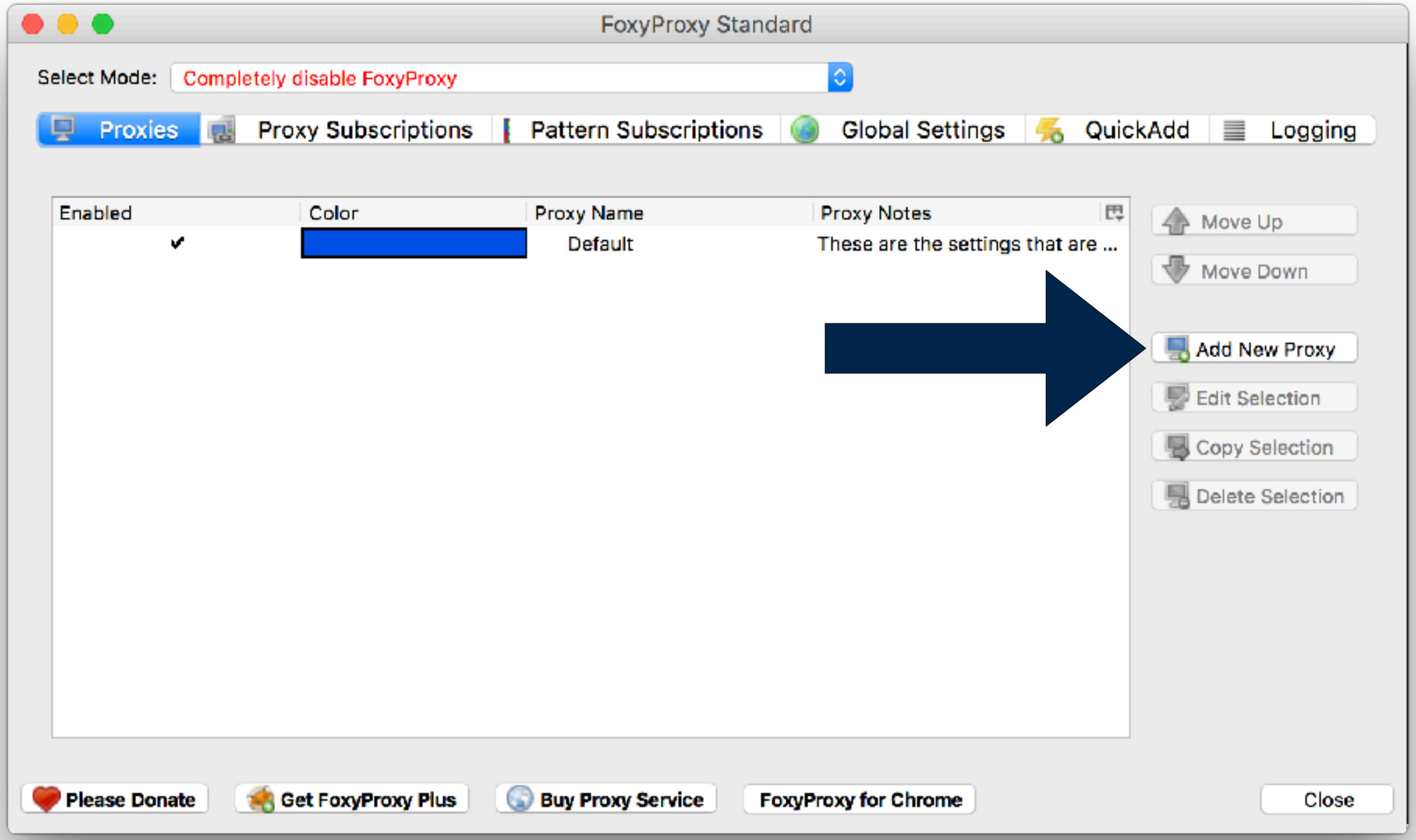
Close

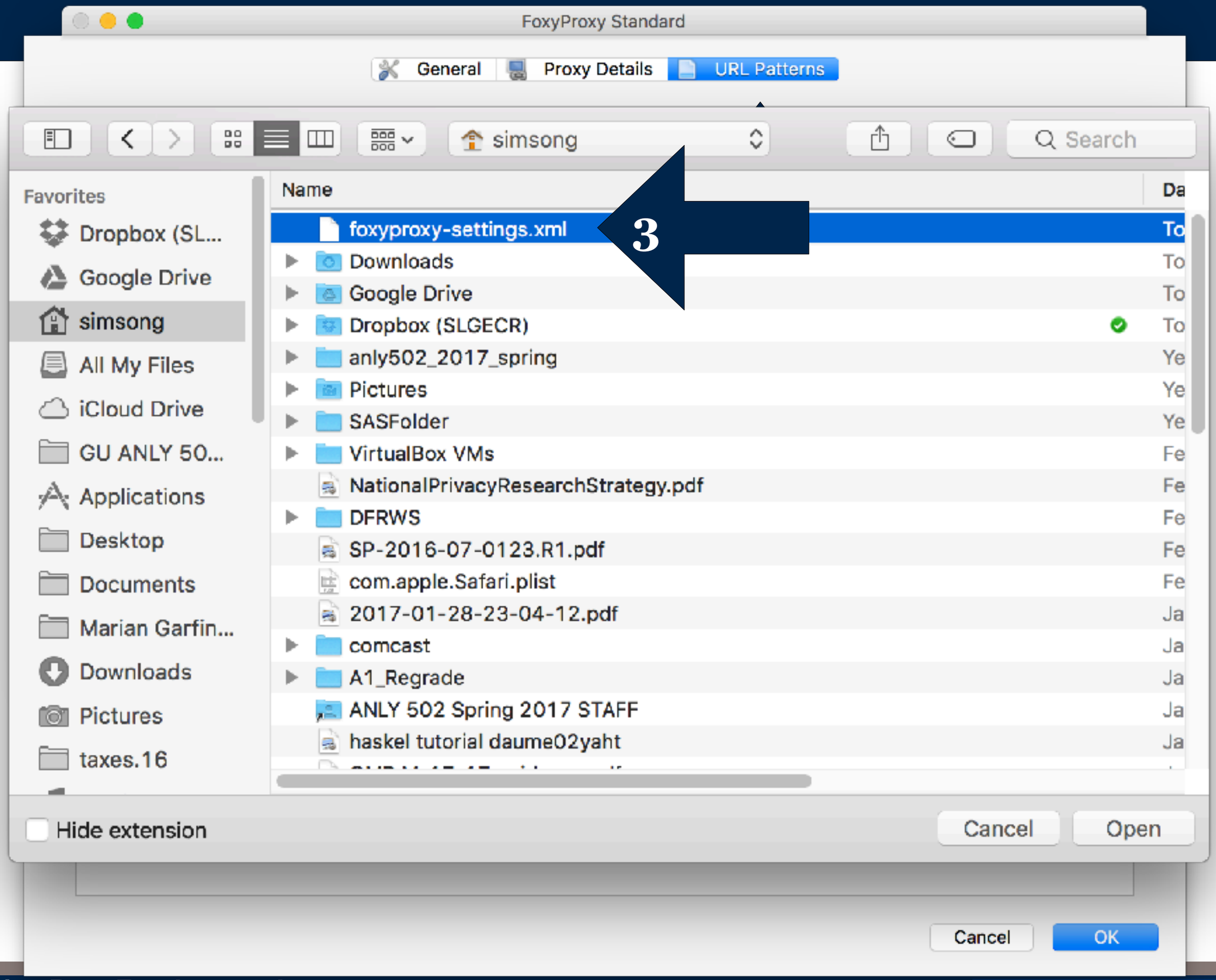
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back! | Refresh Firefox...

Created foxyproxy-settings.xml (using cat, here)

simsong — -bash — 80×45

```
Last login: Mon Feb  6 19:23:21 on ttys000
You have new mail.
.bash_profile
.bashrc
Missing filename ("less --help" for help)
[Dance ~ 19:26:08]$ cat > foxyproxy-settings.xml
<?xml version="1.0" encoding="UTF-8"?>
<foxyproxy>
    <proxies>
        <proxy name="emr-socks-proxy" id="2322596116" notes="" fromSubscription=
"false" enabled="true" mode="manual" selectedTabIndex="2" lastresort="false" ani
matedIcons="true" includeInCycle="true" color="#0055E5" proxyDNS="true" noIntern
alIPs="false" autoconfMode="pac" clearCacheBeforeUse="false" disableCache="false
" clearCookiesBeforeUse="false" rejectCookies="false">
            <matches>
                <match enabled="true" name="*ec2*.amazonaws.com*" pattern="*ec2*
.amazonaws.com*" isRegEx="false" isBlackList="false" isMultiLine="false" caseSen
sitive="false" fromSubscription="false" />
                <match enabled="true" name="*ec2*.compute*" pattern="*ec2*.compu
te*" isRegEx="false" isBlackList="false" isMultiLine="false" caseSensitive="fals
e" fromSubscription="false" />
                <match enabled="true" name="10.*" pattern="http://10.*" isRegEx=
"false" isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscr
i="false" />
                <match enabled="true" name="*10*.amazonaws.com*" pattern="*10*.a
mazonaws.com*" isRegEx="false" isBlackList="false" isMultiLine="false" caseSensi
tive="false" fromSubscription="false" />
                <match enabled="true" name="*10*.compute*" pattern="*10*.compute
*" isRegEx="false" isBlackList="false" isMultiLine="false" caseSensitive="false"
 fromSubscription="false" />
                <match enabled="true" name="*.compute.internal*" pattern="*.comp
ute.internal*" isRegEx="false" isBlackList="false" isMultiLine="false" caseSensi
tive="false" fromSubscription="false" />
                <match enabled="true" name="*.ec2.internal*" pattern="*.ec2.inte
rnal*" isRegEx="false" isBlackList="false" isMultiLine="false" caseSensitive="fa
lse" fromSubscription="false" />
            </matches>
            <manualconf host="localhost" port="8157" socksversion="5" isSocks="t
rue" username="" password="" domain="" />
        </proxy>
    </proxies>
</foxyproxy>
[Dance ~ 19:26:14]$
```


FoxyProxy Standard
Select Mode:
Completely disable FoxyProxy
Proxies
Proxy Subscriptions
Pattern Subscriptions
Global Settings
QuickAdd
Logging
Enabled
Color
Proxy Name
Proxy Notes
Default
These are the settings that are ...
Move Up
Move Down
Add New Proxy
Edit Selection
Copy Selection
Delete Selection
Please Donate
Get FoxyProxy Plus
Buy Proxy Service
FoxyProxy for Chrome
Close


FoxyProxy Standard
General
Proxy Details
URL Patterns
simsong
Search
Favorites
Dropbox (SL...
Google Drive
simsong
All My Files
iCloud Drive
GU ANLY 50...
Applications
Desktop
Documents
Marian Garfin...
Downloads
Pictures
taxes.16
Name
foxyproxy-settings.xml
3
Downloads
Google Drive
Dropbox (SLGECR)
anly502_2017_spring
Pictures
SASFolder
VirtualBox VMs
NationalPrivacyResearchStrategy.pdf
DFRWS
SP-2016-07-0123.R1.pdf
com.apple.Safari.plist
2017-01-28-23-04-12.pdf
comcast
A1_Regrade
ANLY 502 Spring 2017 STAFF
haskel tutorial daume02yaht
Hide extension
Cancel
Open
Cancel
OK

Additional Resources

Useful references

Documentation:

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

Hadoop Architecture:

- http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/
- http://www.edureka.co/blog/apache-hadoop-hdfs-architecture/

Interview Questions

—*HDFS. http://www.edureka.co/blog/hadoop-interview-questions-hdfs-2/*

—*Hadoop Cluster. http://www.edureka.co/blog/hadoop-interview-questions-hadoop-cluster/*

Hadoop Online Tutorial - http://hadooptutorial.info/

- Big Data | Hadoop | Map Reduce | Hive | Pig | HBase | Flume
- http://hadooptutorial.info/hadoop-certification-dump-questions/
- http://hadooptutorial.info/hadoop-interview-questions-and-answers-part-2/

Resources for understanding EC2

Current instances:

- http://www.ec2instances.info/

Getting started with AWS and Python:

- https://aws.amazon.com/articles/Python/3998

mrjob:

- Donald Miner PyCon 2015 - https://www.youtube.com/watch?v=b8HLYUp_fA8

AWS Slideshow about EMR:

- http://www.slideshare.net/AmazonWebServices/deep-dive-amazon-elastic-map-reduce

S3 and Hadoop performance:

- http://blog.mortardata.com/post/58920122308/s3-hadoop-performance
- https://aws.amazon.com/blogs/aws/amazon-s3-performance-tips-tricks-seattle-hiring-event/

Nice explanation of Hadoop joins:

- https://chamibuddhika.wordpress.com/2012/02/26/joins-with-map-reduce/
- http://blog.matthewrathbone.com/2013/02/09/real-world-hadoop-implementing-a-left-outer-join-in-hadoop-map-reduce.html

AWS re:Invent (Amazon's trade show)

https://reinvent.awsevents.com/

https://www.youtube.com/user/AmazonWebServices/Cloud

"State of the Union: AWS Storage Services" — http://bit.ly/1khmDP6

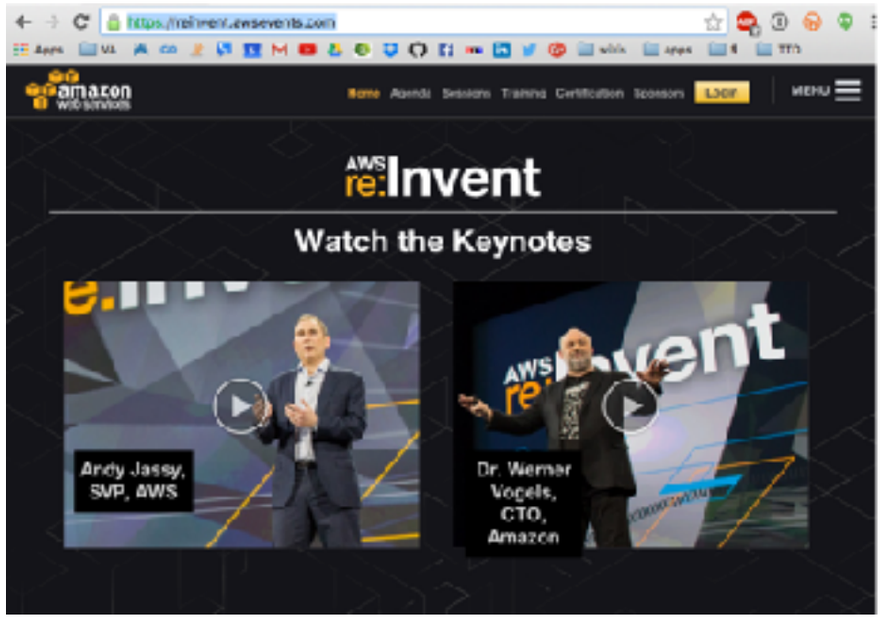
"Self-service Cloud Services" — http://bit.ly/1khmIm5

"Real-World Smart Applications with Amazon Machine Learning"

- http://bit.ly/1khmLyj

"Amazon RDS for PostgreSQL"

- http://bit.ly/1khmM5g



git - Get to know it

Use git for:

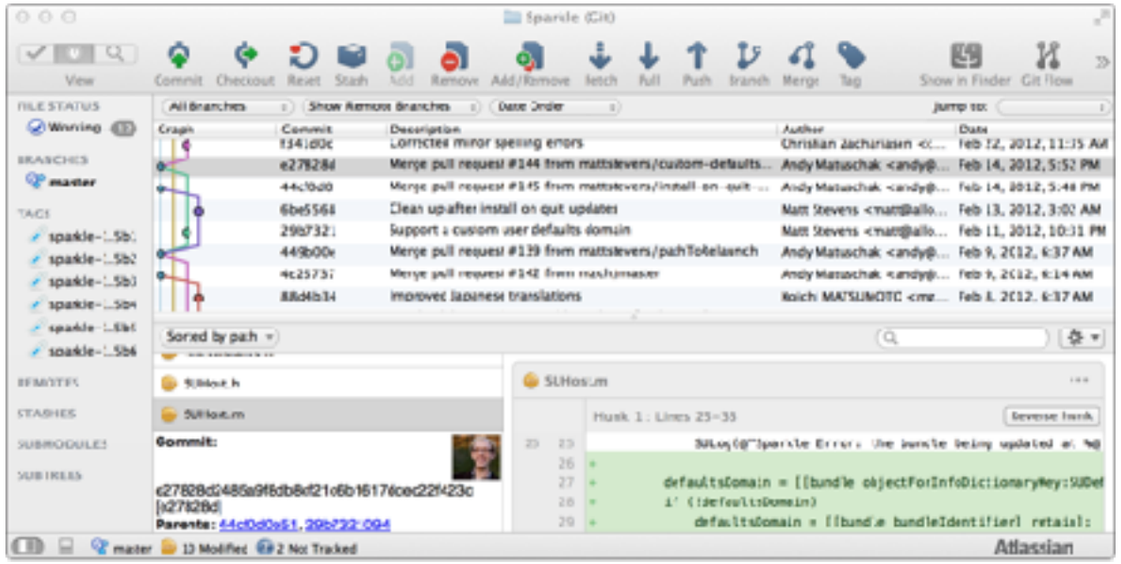
- Storing your work
- (soon: Submitting your homework programs)

Git tutorials:

- https://try.github.io/levels/1/challenges/1
- http://git-scm.com/docs/gittutorial
- https://www.atlassian.com/git/tutorials/

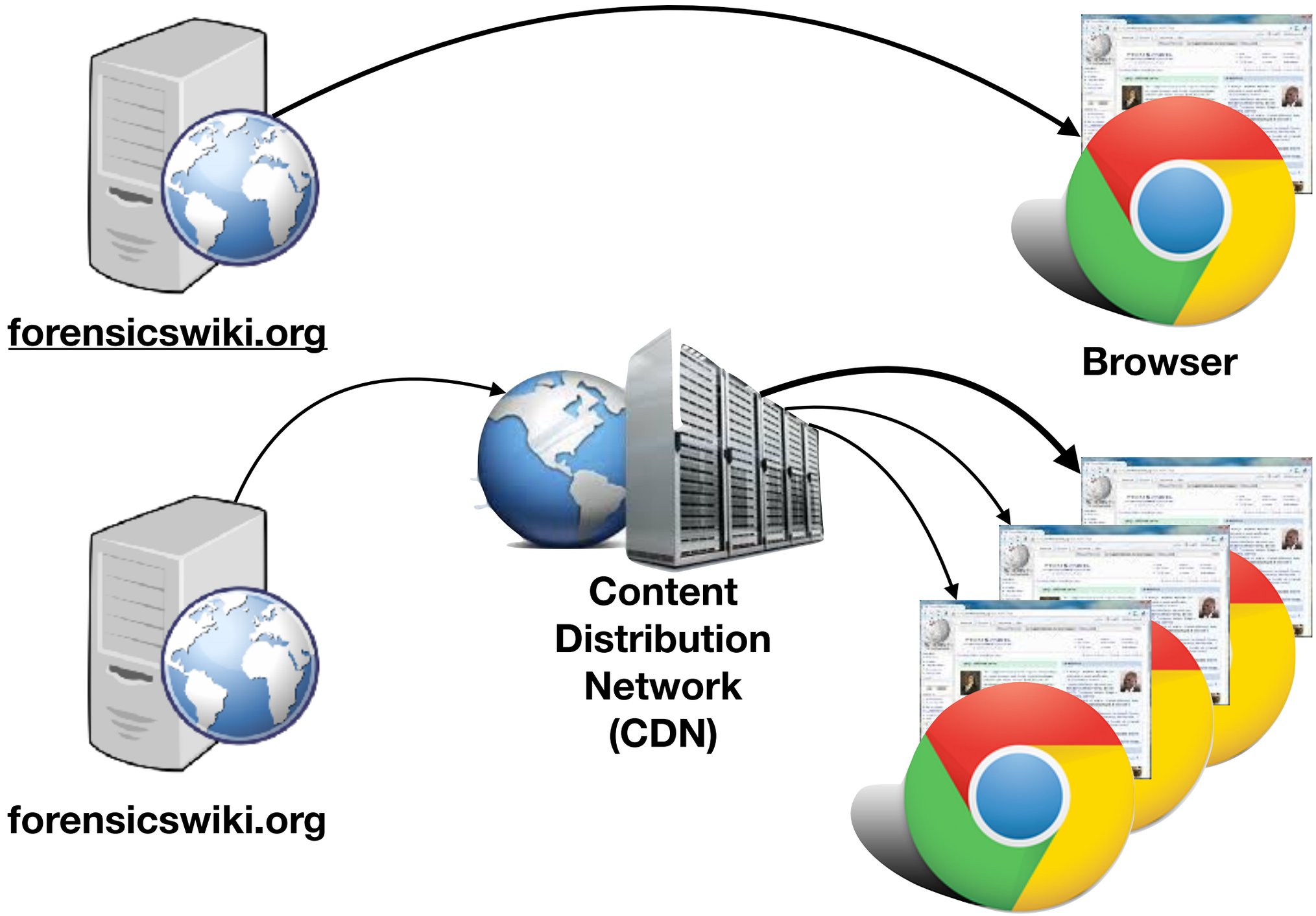
Git GUI: Atlassan SourceTree:

- https://www.sourcetreeapp.com/

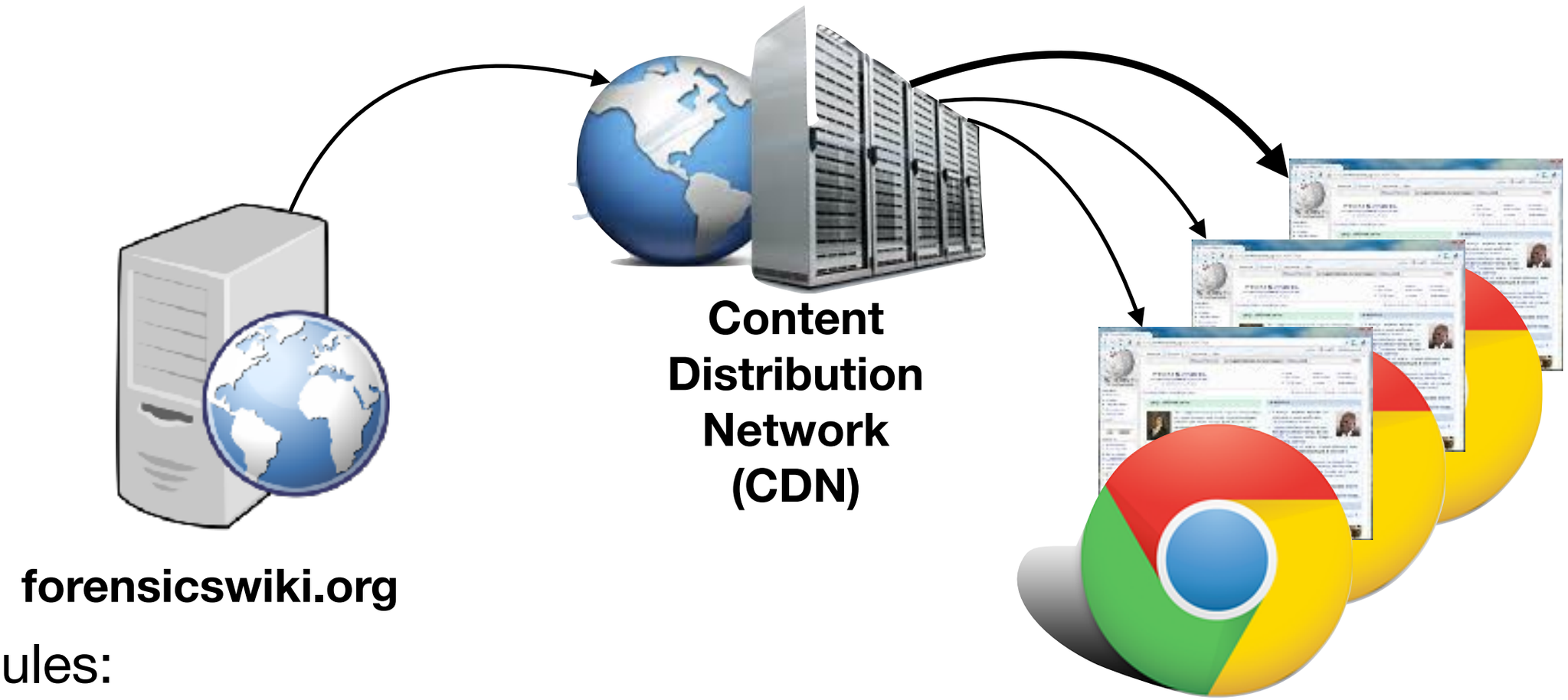


A worked example

Basic Web Architecture



CDN — Key points



CDN Rules:

1. Page moves from web server to CDN *with first request*.
2. Subsequent requests satisfied by CDN.
3. After time T, the page is removed from the CDN.

To answer:

1. How many page fetches would benefit from a CDN?

The Forensics Wiki logfile:

On S3:

```
$ aws s3 ls s3://gu-anly502/
                                   PRE A1/
                                   PRE A2/
                                   PRE A3/
                                   PRE L04/
                                   PRE gutenberg/
                                   PRE logs/
                                   PRE maxmind/
                                   PRE new folder/
                                   PRE ps03/
                                   PRE ps04/
```

.

```
$ aws s3 ls s3://gu-anly502/logs/
2017-01-08 23:56:48 4268793922 forensicswiki.2012.txt
```

.

```
$ aws s3 cp s3://gu-anly502/logs/forensicswiki.2012.txt - | head -2
77.21.0.59 - - [01/Jan/2012:00:35:03 -0800] "GET /wiki/Write_Blockers
HTTP/1.1" 200 5742 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
77.21.0.59 - - [01/Jan/2012:00:35:04 -0800] "GET /w/skins/common/
wikibits.js?270 HTTP/1.1" 200 31165 "http://www.forensicswiki.org/wiki/
Write_Blockers" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
```

s3://gu-anly502/logs/forensicswiki.2012.txt

```
77.21.0.59 - - [01/Jan/2012:00:35:04 -0800] "GET /w/skins/common/
wikibits.js?270 HTTP/1.1" 200 31165 "http://www.forensicswiki.org/
wiki/Write_Blockers" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/
534.52.7"
```

77.21.0.59 – Source IP Address

[01/Jan/2012:00:35:04 -0800] – Time

/w/skins/common/wikibits.js?270 – URL

http://www.forensicswiki.org/wiki/Write_Blockers – Refer (prev page)

31165 – Bytes Transferred

We want to compute the distribution of (*URL, Bytes Transferred*) tuples.

Basic Plan — "Word count" pattern on (URL,size)

Input:

```
77.21.0.59 - - [01/Jan/2012:00:35:04 -0800] "GET /w/skins/common/
wikibits.js?270 HTTP/1.1" 200 31165 "http://www.forensicswiki.org/
wiki/Write_Blockers" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/
534.52.7"
```

Mapper:

```
Input -> (/w/skins/common/wikibits.js?270, 31165)  : 1
```

Reducer:

```
  (/w/skins/common/wikibits.js?270, 31165)  : 1
  (/w/skins/common/wikibits.js?270, 31165)  : 1
  (/w/skins/common/wikibits.js?270, 31165)  : 1
  (/w/skins/common/wikibits.js?270, 31165)  : 1
+ ------------------------------------------------
  (/w/skins/common/wikibits.js?270, 31165)  : 4
```

Trick: Get the parsing right!

Basic word-count MRJOB

```
#!/usr/bin/env python34
#
# Analyze forensics wiki input file
# For information on MRJOB see:
# https://pythonhosted.org/mrjob/guides/quickstart.html

from mrjob.job import MRJob
import re

WORD_RE = re.compile(r"[\w']+")

class FWikiAnalyzer(MRJob):

    def mapper(self, _, line):
        for word in WORD_RE.findall(line):
            yield word.lower(), 1

    def reducer(self, word, counts):
        yield word, sum(counts)


if __name__ == '__main__':
    MRWordFreqCount.run()
```

Change this

Change this

Modify for Apache Combined Log file

```
from mrjob.job import MRJob
import re

CLR_RE = re.compile(r'^(\S+) (\S+) (\S+) \[([^\]]+)\] "(\S+) (\S+)
\S+" (\S+) (\S+) "([^"]*)" "([^"]*)"')

class FWikiAnalyzer(MRJob):

    def mapper(self, _, line):
        m = CLR_RE.search(line)
        ( verb,path,http ) = m.group(5).split(" ")
        yield ( path,m.group(7) ) , 1

    def reducer(self, url_count, ones):
  yield url_count, sum(ones)


if __name__ == '__main__':
    FWikiAnalyzer.run()
```

This is my first try.
Will it work? Probably not!

Use py.test to help develop your code

in fwiki_cdn1.py:

```
CLR_RE = re.compile(r'^(\S+) (\S+) (\S+) \[([^\]]+)\] "(\S+) (\S+) \S+" (\S+)
(\S+) "([^"]*)" "([^"]*)"')
```

Test the regular expression with assert statements in a test program.

```
#!/usr/bin/env python34
import fwiki_cdn1

log='77.21.0.59 - - [01/Jan/2012:00:35:04 -0800] "GET /w/skins/common/
wikibits.js?270 HTTP/1.1" 200 31165 "http://www.forensicswiki.org/wiki/
Write_Blockers" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/
534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"'

# create a test
def test_CLR_RE():
    m = fwiki_cdn1.CLR_RE.search(log)
    assert m.group(1)=="77.21.0.59"
    assert m.group(2)=="-"
    assert m.group(3)=="-"
    assert m.group(4)=="01/Jan/2012:00:35:04 -0800"
    assert m.group(5)=="GET /w/skins/common/wikibits.js?270 HTTP/1.1"
    assert m.group(6)=="200"
    assert m.group(7)=="31165"
    assert m.group(8)=="Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
```

imports the code for testing!

references the RE!

Test

```
$ ls -l
-rw-rw-r-- 1 hadoop hadoop  609 Feb  6 11:28 fwiki_cdn1.py
-rw-rw-r-- 1 hadoop hadoop  849 Feb  6 11:37 fwiki_cdn1_test.py
```

Test

```
$ ls -l
-rw-rw-r-- 1 hadoop hadoop  609 Feb  6 11:28 fwiki_cdn1.py
-rw-rw-r-- 1 hadoop hadoop  849 Feb  6 11:37 fwiki_cdn1_test.py

[hadoop@ip-172-31-39-51 L04]$ py.test
============================= test session starts ==============================
platform linux -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /home/hadoop/anly502_2017_spring/L04, inifile:
collected 1 items

fwiki_cdn1_test.py F

=================================== FAILURES ===================================
_____________________________________ test_CLR_RE _____________________________________

    def test_CLR_RE():
        m = fwiki_cdn1.CLR_RE.search(log)
        assert m.group(1)=="77.21.0.59"
        assert m.group(2)=="-"
        assert m.group(3)=="-"
        assert m.group(4)=="01/Jan/2012:00:35:04 -0800"
>       assert m.group(5)=="GET /w/skins/common/wikibits.js?270 HTTP/1.1"
E       assert 'GET' == 'GET /w/skins/common/wikibits.js?270 HTTP/1.1'
E         - GET
E         + GET /w/skins/common/wikibits.js?270 HTTP/1.1

fwiki_cdn1_test.py:17: AssertionError
========================== 1 failed in 0.23 seconds ===========================
[hadoop@ip-172-31-39-51 L04]$
```

py.test caught the mistake!

Old code:

```
        assert m.group(1)=="77.21.0.59"
        assert m.group(2)=="-"
        assert m.group(3)=="-"
        assert m.group(4)=="01/Jan/2012:00:35:04 -0800"
        assert m.group(5)=="GET /w/skins/common/wikibits.js?270 HTTP/1.1"
        assert m.group(6)=="200"
        assert m.group(7)=="31165"
        assert m.group(8)=="Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
    AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
```

New code:

```
    assert m.group(1)=="77.21.0.59"
    assert m.group(2)=="-"
    assert m.group(3)=="-"
    assert m.group(4)=="01/Jan/2012:00:35:04 -0800"
    assert m.group(5)=="GET"
    assert m.group(6)=="/w/skins/common/wikibits.js?270"
    assert m.group(7)=="200"
    assert m.group(8)=="31165"
    assert m.group(9)=="http://www.forensicswiki.org/wiki/
Write_Blockers"
    assert m.group(10)=="Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
```

Correct test results:

```
[hadoop@ip-172-31-39-51 L04]$ py.test
========================= test session starts =========================
platform linux -- Python 3.4.3, pytest-3.0.6, py-1.4.32,
pluggy-0.4.0
rootdir: /home/hadoop/anly502_2017_spring/L04, inifile:
collected 1 items

fwiki_cdn1_test.py .

========================= 1 passed in 0.18 seconds ====================
```

So the regular expression is okay!
I had the fields wrong in the test.
How about the mapper and reducer?

mapper (fwiki_cdn1.py)

```
...
class FWikiAnalyzer(MRJob):

    def mapper(self, _, line):
        m = CLR_RE.search(line)
        ( verb,path,http ) = m.group(5).split(" ")
        yield ( path,m.group(7) ) , 1
...
```

mapper test:

```
def test_mapper():
    for (key, value) in fwiki_cdn1.FWikiAnalyzer.mapper(None,"",log):
        assert key[0] == "/w/skins/common/wikibits.js?270"
        assert key[1] == 31165
        assert value  == 1
```

We need a "for" loop for even a single value, because mapper() is an *iterator*.

test run

```
========================================= test session starts =========================================
platform linux -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /home/hadoop/anly502_2017_spring/L04, inifile:
collected 2 items

fwiki_cdn1_test.py .F

================================================ FAILURES =================================================
_____________________________________________ test_mapper _____________________________________________

    def test_mapper():
>       (key,value) = fwiki_cdn1.FWikiAnalyzer.mapper(None,"",log)

fwiki_cdn1_test.py:25:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

self = None, _ = ''
line = '77.21.0.59 - - [01/Jan/2012:00:35:04 -0800] "GET /w/skins/common/wikibits.js?270 HTTP/
1.1" 200 31165 "http://www.fore...Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/
534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"'

    def mapper(self, _, line):
        m = CLR_RE.search(line)
>       ( verb,path,http ) = m.group(5).split(" ")
E       ValueError: need more than 1 value to unpack

fwiki_cdn1.py:16: ValueError
============================ 1 failed, 1 passed in 0.22 seconds ============================
```

I forgot to fix the mapper!

Old mapper:

```
    def mapper(self, _, line):
        m = CLR_RE.search(line)
        ( verb, path, http ) = m.group(5).split(" ")
        yield ( path, m.group(7) ) , 1
```

New mapper:

```
    def mapper(self, _, line):
        m = CLR_RE.search(line)
        yield ( m.group(6), int( m.group(8) ) ) , 1
```

New test results:

```
[hadoop@ip-172-31-39-51 L04]$ py.test
================================= test session starts =================================
platform linux -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /home/hadoop/anly502_2017_spring/L04, inifile:
collected 2 items

fwiki_cdn1_test.py ..

================================= 2 passed in 0.19 seconds ==================================
```

Test the reducer

Reducer:

```
    def reducer(self, url_count, ones):
        yield url_count, sum(ones)
```

Reducer Test:

```
def test_reducer():
    key = ("/w/skins/common/wikibits.js?270",31165)
    values = [1,1,1,1]
    for (key2, value2) in fwiki_cdn1.FWikiAnalyzer.reducer(None, key, values):
        assert key2[0] == "/w/skins/common/wikibits.js?270"
        assert key2[1] == 31165
        assert value2  == 4
```

Results:

```
[hadoop@ip-172-31-39-51 L04]$ py.test
========================================= test session starts =========================================
platform linux -- Python 3.4.3, pytest-3.0.6, py-1.4.32, pluggy-0.4.0
rootdir: /home/hadoop/anly502_2017_spring/L04, inifile:
collected 3 items

fwiki_cdn1_test.py ...

================================== 3 passed in 0.18 seconds ===================================
[hadoop@ip-172-31-39-51 L04]$
```

Worked the first time!

Run locally with a small set of data

```
$ aws s3 cp s3://gu-anly502/logs/forensicswiki.2012.txt - | \
  head -100 > /tmp/forensicswiki.2012-100.txt
download failed: s3://gu-anly502/logs/forensicswiki.2012.txt to -
[Errno 32] Broken pipe


$ python34 fwiki_cdn1.py -r local /tmp/forensicswiki.2012-100.txt
Using configs in /home/hadoop/.mrjob.conf
Creating temp directory /tmp/fwiki_cdn1.hadoop.
20170206.122122.483854
Running step 1 of 1...
Streaming final output from /tmp/fwiki_cdn1.hadoop.
20170206.122122.483854/output...
["/", 388]        1
["/favicon.ico", 275]    4
["/logo.png", 173]       1
["/logo.png", 47549]     1
["/logo.png", 47550]     2
["/w/skins/common/ajax.js?270", 5069]   1
```

Run with Hadoop:

```
$ python34 fwiki_cdn1.py -r hadoop \
    s3://gu-anly502/logs/forensicswiki.2012.txt \
    -o s3://anly502-slg/L04-1/
...
```

Run with Hadoop:

```
$ python34 fwiki_cdn1.py -r hadoop s3://gu-anly502/logs/forensicswiki.2012.txt -o s3://anly502-slg/L04-1/
```

Lots of errors!

```
[hadoop@ip-172-31-39-51 L04]$ python34 fwiki_cdn1.py -r hadoop s3://gu-anly502/logs/forensicswiki.2012.txt -o s3://
anly502-slg/L04-1/
Using configs in /home/hadoop/.mrjob.conf
Using Hadoop version 2.7.3
Looking for Hadoop streaming jar in /usr/lib/hadoop...
Found Hadoop streaming jar: /usr/lib/hadoop/hadoop-streaming.jar
Creating temp directory /tmp/fwiki_cdn1.hadoop.20170206.121721.581934
Copying local files to hdfs:///user/hadoop/tmp/mrjob/fwiki_cdn1.hadoop.20170206.121721.581934/files/...
Running step 1 of 1...
  packageJobJar: [] [/usr/lib/hadoop/hadoop-streaming-2.7.3-amzn-1.jar] /tmp/streamjob2087561882746549198.jar tmpDir=null
  Connecting to ResourceManager at ip-172-31-39-51.ec2.internal/172.31.39.51:8032
  Connecting to ResourceManager at ip-172-31-39-51.ec2.internal/172.31.39.51:8032
  Loaded native gpl library
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev 8d8f819de6dcf9e76105e2d20797d885f3804dee]
  Total input paths to process : 1
  number of splits:64
  Submitting tokens for job: job_1486318287298_0002
  Submitted application application_1486318287298_0002
  The url to track the job: http://ip-172-31-39-51.ec2.internal:20888/proxy/application_1486318287298_0002/
  Running job: job_1486318287298_0002
  Job job_1486318287298_0002 running in uber mode : false
   map 0% reduce 0%
  Task Id : attempt_1486318287298_0002_m_000000_0, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1
    at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:322)
    at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:535)
    at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
    at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
    at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
    at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:455)
    at org.apache.hadoop.mapred.MapTask.run(MapTask.java:344)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:164)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:415)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1698)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:158)

Container killed by the ApplicationMaster.
Container killed on request. Exit code is 143
Container exited with a non-zero exit code 143
```

Our code has no error checking!

Old mapper:

```
    def mapper(self, _, line):
        m = CLR_RE.search(line)
        yield ( m.group(6), int( m.group(8) ) ) , 1
```

New mapper:

```
    def mapper(self, _, line):
        m = CLR_RE.search(line)
        try:
            if m:
                yield ( m.group(6), int( m.group(8) ) ) , 1
        except RuntimeError as e:
            pass

    def reducer(self, url_count, ones):
        yield url_count, sum(ones)
```

Catching the errors, it works...

```
[hadoop@ip-172-31-39-51 L04]$ python34 fwiki_cdn1.py -r hadoop s3://gu-anly502/logs/
forensicswiki.2012.txt -o s3://anly502-slg/L04-1/
Using configs in /home/hadoop/.mrjob.conf
Using Hadoop version 2.7.3
Looking for Hadoop streaming jar in /usr/lib/hadoop...
Found Hadoop streaming jar: /usr/lib/hadoop/hadoop-streaming.jar
Creating temp directory /tmp/fwiki_cdn1.hadoop.20170206.134438.888865
Copying local files to hdfs:///user/hadoop/tmp/mrjob/fwiki_cdn1.hadoop.
20170206.134438.888865/files/...
Running step 1 of 1...
  packageJobJar: [] [/usr/lib/hadoop/hadoop-streaming-2.7.3-amzn-1.jar] /tmp/
streamjob6735522675049733530.jar tmpDir=null
  Connecting to ResourceManager at ip-172-31-39-51.ec2.internal/172.31.39.51:8032
  Connecting to ResourceManager at ip-172-31-39-51.ec2.internal/172.31.39.51:8032
  Loaded native gpl library
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev
8d8f819de6dcf9e76105e2d20797d885f3804dee]
  Total input paths to process : 1
  number of splits:64
  Submitting tokens for job: job_1486318287298_0003
  Submitted application application_1486318287298_0003
  The url to track the job: http://ip-172-31-39-51.ec2.internal:20888/proxy/
application_1486318287298_0003/
  Running job: job_1486318287298_0003
  Job job_1486318287298_0003 running in uber mode : false
   map 0% reduce 0%
   map 1% reduce 0%
   map 2% reduce 0%
   map 3% reduce 0%
   map 4% reduce 0%
   map 5% reduce 0%
```

```
$ aws s3 ls s3://anly502-slg/L04-1/
2017-02-06 13:57:22          0 _SUCCESS
2017-02-06 13:57:08   45785080 part-00000
2017-02-06 13:57:06   45788680 part-00001
2017-02-06 13:57:20   45784681 part-00002
$
```

Copy to the local system:

```
$ aws s3 cp --recursive s3://anly502-slg/L04-1 .
download: s3://anly502-slg/L04-1/_SUCCESS to ./_SUCCESS
download: s3://anly502-slg/L04-1/part-00001 to ./part-00001
download: s3://anly502-slg/L04-1/part-00002 to ./part-00002
download: s3://anly502-slg/L04-1/part-00000 to ./part-00000
```

Notice — copy was not in order!

(Time stamps were not in order either.)

However, it turns out that the files themselves are sorted!

It turns out that the files are sorted…

Here is a sample of the part-00000 file:

```
$ tail part-00000
["http://www.forensicswiki.org/wiki/Volatility_Framework", 20031]
  1
["http://www.forensicswiki.org/wiki/WFT", 13806] 1
["http://www.forensicswiki.org/wiki/Websites/", 9931]   1
["http://www.forensicswiki.org/wiki/Xplico", 15271]     1
["http://www.forensicswiki.org/wiki/Zfone", 10226]      1
["http://www.forensicswiki.org/wiki/index.php?
title=Special:UserLogin&type=signup&returnto=Main+Page", 3895]  3
["http://www.forensicswiki.org/wiki/null?
debug=null&lang=null&modules=jquery.checkboxShiftClick%2Ccookie%2Cma
keCollapsible%2CmessageBox%2CmwPrototypes%2Cplaceholder%7Cmediawiki.
language%2Cuser%2Cutil%7Cmediawiki.legacy.ajax%2Cwikibits%7Cmediawik
i.page.ready&skin=null&version=20120414T164358Z&*", 3973]       1
["http://www.forensicswiki.org/wiki/wp-login.php", 488] 5
["http://www.forensicswiki.org/wikka.php?wakka=UserSettings", 438]
  1
$
```

A program to verify sort order:

```
#!/usr/bin/env python34
#
# Verify sort order
#
if __name__=="__main__":
    import sys,json
    for fname in sys.argv[1:]:
        a0 = None
        linenumber = 0
        for line in open(fname,"rU"):
            linenumber += 1
            (k,v) = line.strip().split("\t")
            a = json.loads(k)
            if a0 and a[0] < a0[0]:
                print("{}: {}".format(linenumber-1,a0))
                print("{}: {}".format(linenumber,a))
            a0 = a0
```

Run the program, and there is no output:

```
$ python sortcheck.py part-0000?
$
```

```
:type partitioner: str
:param partitioner: Optional name of a Hadoop partitioner class, e.g.
                    ``'org.apache.hadoop.mapred.lib.HashPartitioner'``.
                    Hadoop streaming will use this to determine how
                    mapper output should be sorted and distributed
                    to reducers.
```