BIG DATA WITH HADOOP EXPERIMENTATION

HARRISON CARRANZA

School of Computer Science and Mathematics

Marist College

APARICIO CARRANZA

NYC College of Technology – CUNY

ECC Conference 2016 – Poughkeepsie, NY, June 12 -14, 2016

Marist College

AGENDA

- WHAT IS AND WHY USE HADOOP?
- HISTORY AND WHO USES HADOOP
- KEY HADOOP TECHNOLOGIES
- HADOOP STRUCTURE
- RESEARCH AND IMPLEMENTATION
- CONCLUSIONS

WHAT IS AND WHY USE HADOOP?

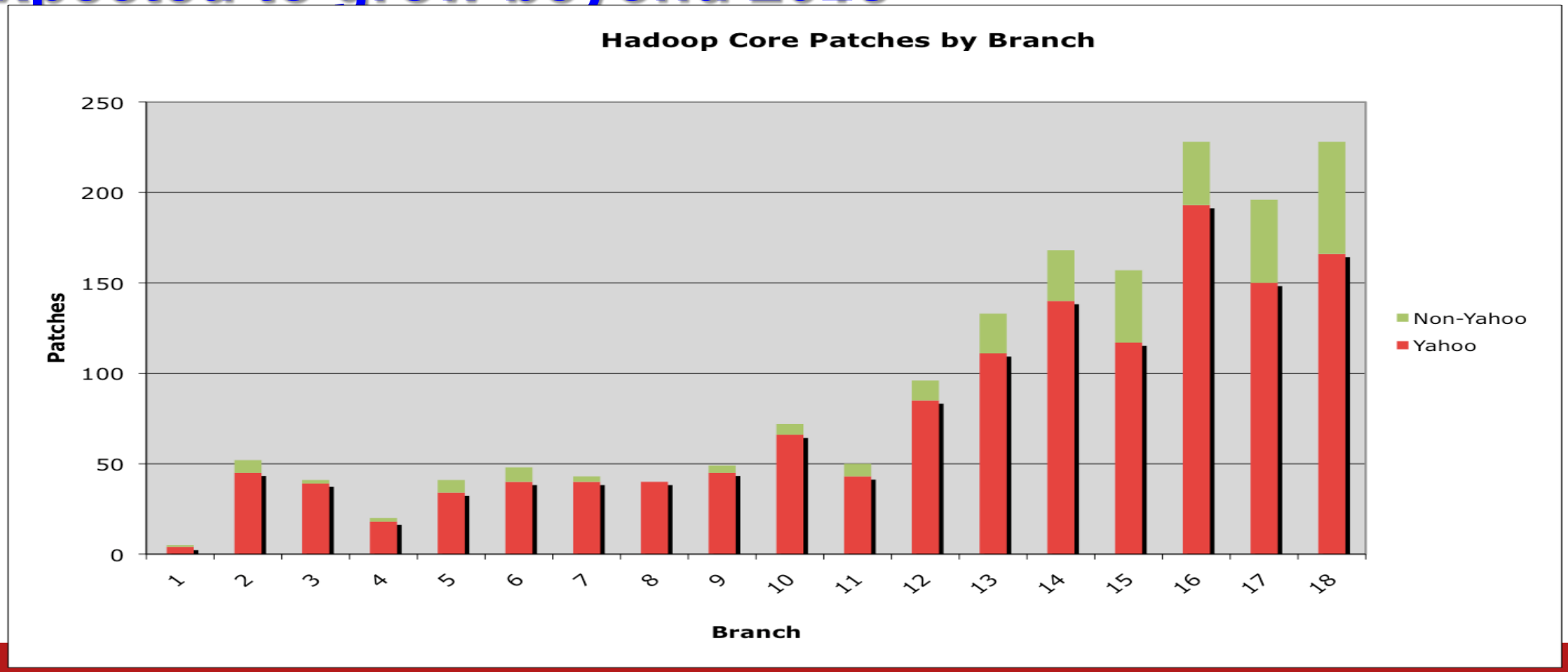
- **Open Source Apache Project**
- **Hadoop Core includes:**
  - *Distributed File System - distributes data*
  - *Map/Reduce - distributes application*
- **Written in Java**
- **Runs on**
  - *Linux, Mac OS/X, Windows*

WHAT IS AND WHY USE HADOOP?

- How do you scale up applications?
  - *Run jobs processing hundreds of terabytes of data*
  - *Takes 11 days to read on 1 computer*
- Need lots of cheap computers
  - *Fixes speed problem (15 minutes on 1000 computers)*
  - *Reliability problems*
    - In large clusters, computers fail every day
    - Cluster size is not fixed
    - Information on one computer gets lost
- Need common infrastructure
  - *Must be efficient and reliable*

HISTORY AND WHO USES HADOOP

- Yahoo! became the primary contributor in 2006
- It increased over the next decade up until now & is expected to grow beyond 2018



HISTORY AND WHO USES HADOOP

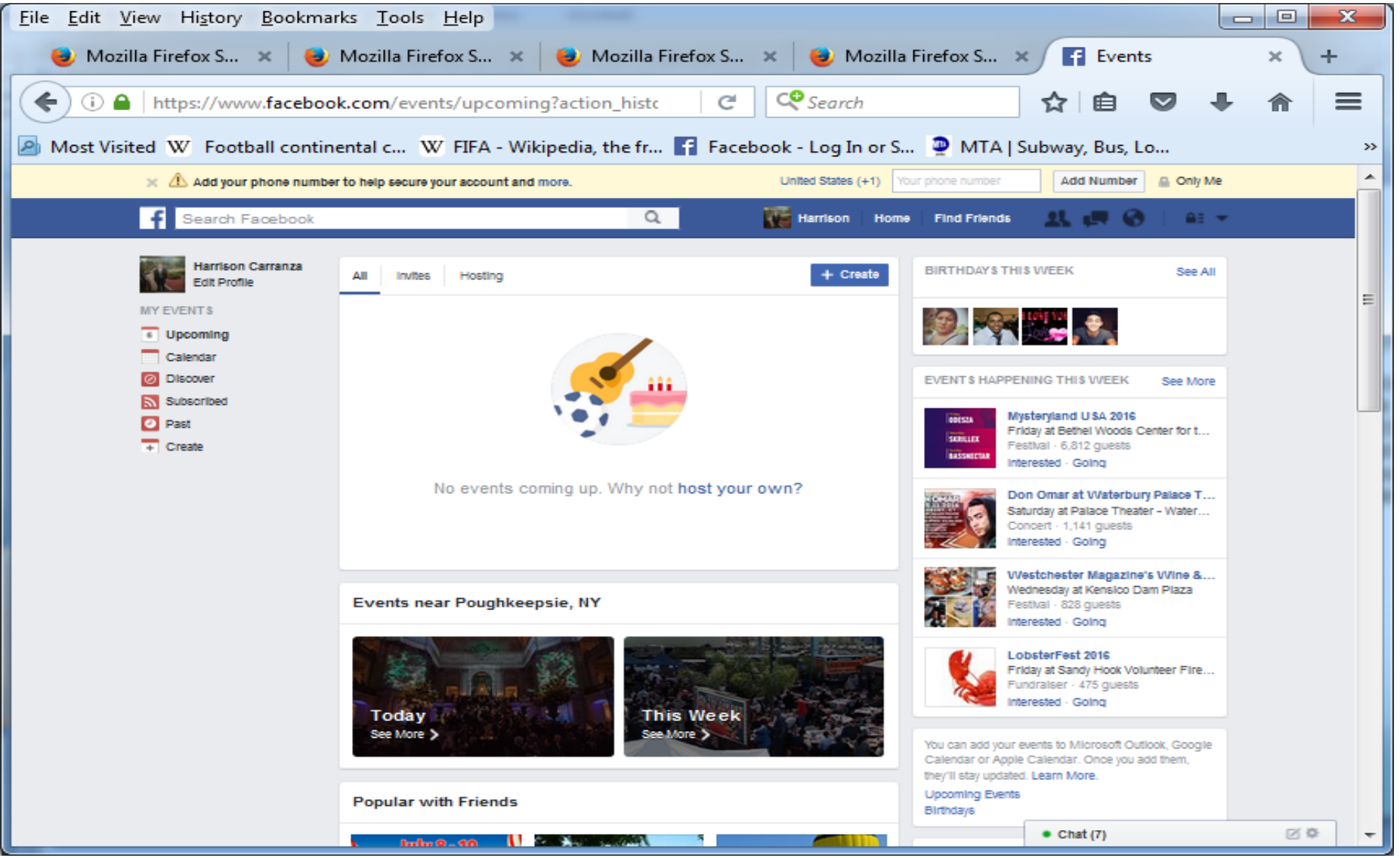
- Yahoo! deployed large scale science clusters in 2007
- Tons of Yahoo! Research papers emerge:
  - WWW
  - CIKM
  - SIGIR
  - VLDB
  - ……
- Yahoo! began running major production jobs in Q1 2008

HISTORY AND WHO USES HADOOP

- Amazon/A9
- AOL
- Facebook
- Fox interactive media
- Google
- IBM
- New York Times
- PowerSet (*now Microsoft*)
- Quantcast
- Rackspace/Mailtrust
- Veoh
- Yahoo!
- New York Times

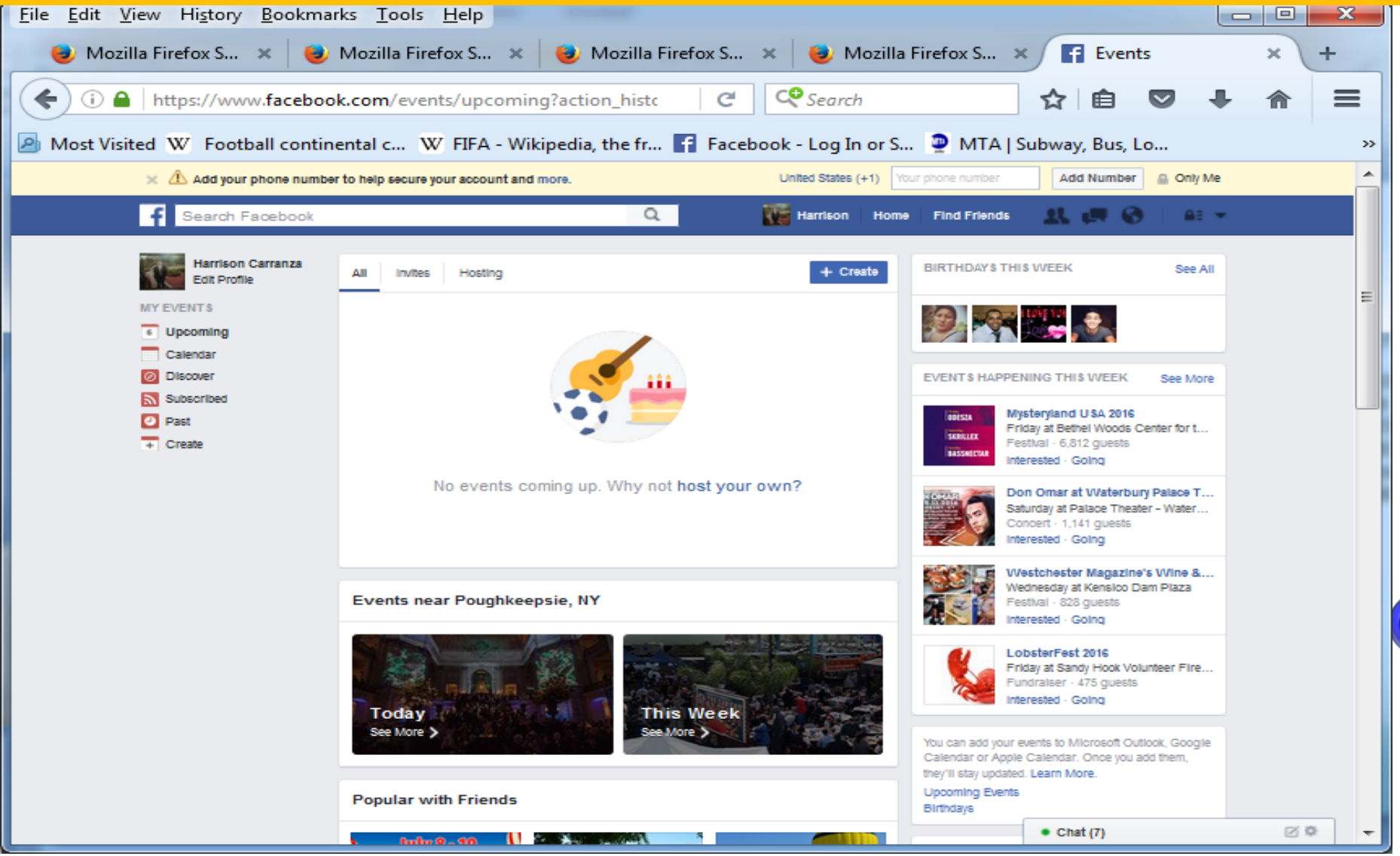
- More at http://wiki.apache.org/hadoop/PoweredBy

HISTORY AND WHO USES HADOOP



- ***When you visit Facebook, you are interacting with data processed with Hadoop!***

HISTORY AND WHO USES HADOOP



Content Optimization

Search Index

- When you visit Facebook, you are interacting with data processed with Hadoop!

Ads Optimization

Content Feed Processing

HISTORY AND WHO USES HADOOP NOWADAYS

- Facebook! has ~20,000 machines running Hadoop
- The largest clusters are currently 2000 nodes
- Several petabytes of user data (compressed, unreplicated)
- Facebook runs hundreds of thousands of jobs every month

KEY HADOOP TECHNOLOGIES

- **Hadoop Core**
  - *Distributed File System*
  - *MapReduce Framework*

The Apache Software Foundation
http://www.apache.org/

- **Pig (initiated by Yahoo!)**
  - *Parallel Programming Language and Runtime*
- **Hbase (initiated by Powerset)**
  - *Table storage for semi-structured data*

- **Zookeeper (initiated by Yahoo!)**
  - *Coordinating distributed systems*
- **Hive (initiated by Facebook)**
  - *SQL-like query language and metastore*

HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

HDFS is designed to reliably store very large files across machines in a large cluster

It is inspired by the Google File System

Hadoop DFS stores each file as a sequence of blocks, all blocks in a file except the last block are the same size

Blocks belonging to a file are replicated for fault tolerance

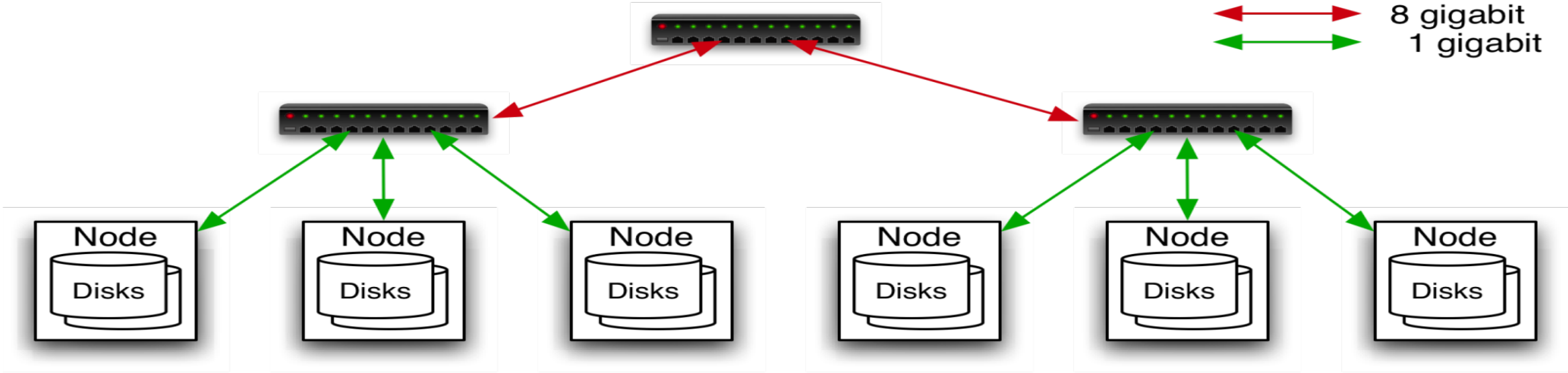
The block size and replication factor are configurable per file

Files in HDFS are "write once" and have strictly one writer at any time

Hadoop Distributed File System – Goals:

- ***Store large data sets***
- ***Cope with hardware failure***
- ***Emphasize streaming data access***

HADOOP STRUCTURE



- **Commodity hardware**
  - ***Linux PCs with local 4 disks***
- **Typically in 2 level architecture**
  - ***40 nodes/rack***
  - ***Uplink from rack is 8 gigabit***
  - ***Rack-internal is 1 gigabit all-to-all***

HADOOP STRUCTURE

- **Single namespace for entire cluster**
  - *Managed by a* ***single namenode***
  - *Files are single-writer and append-only*
  - *Optimized for streaming reads of large files*
- **Files are broken in to large block**
  - *Typically 128 MB*
  - *Replicated to* ***several datanodes****, for reliability*
- **Client talks to both namenode and datanodes**
  - *Data is not sent through the namenode*
  - *Throughput of file system scales nearly linearly with the number of nodes*
- **Access from Java, C, or command line**

HADOOP STRUCTURE

- **Java and C++ APIs**
  - *In Java use Objects, while in C++ bytes*
- **Each task can process data sets larger than RAM**
- **Automatic re-execution on failure**
  - *In a large cluster, some nodes are always slow or flaky*
  - *Framework re-executes failed tasks*
- **Locality optimizations**
  - *Map-Reduce queries HDFS for locations of input data*
  - *Map tasks are scheduled close to the inputs when possible*

RESEARCH & IMPLEMENTATION

There were three different multi node clusters produced each utilizing:

- Microsoft Windows 7
- Linux Ubuntu
- Mac OS X Yosemite

Hadoop functions were then compared in each of these three different environments

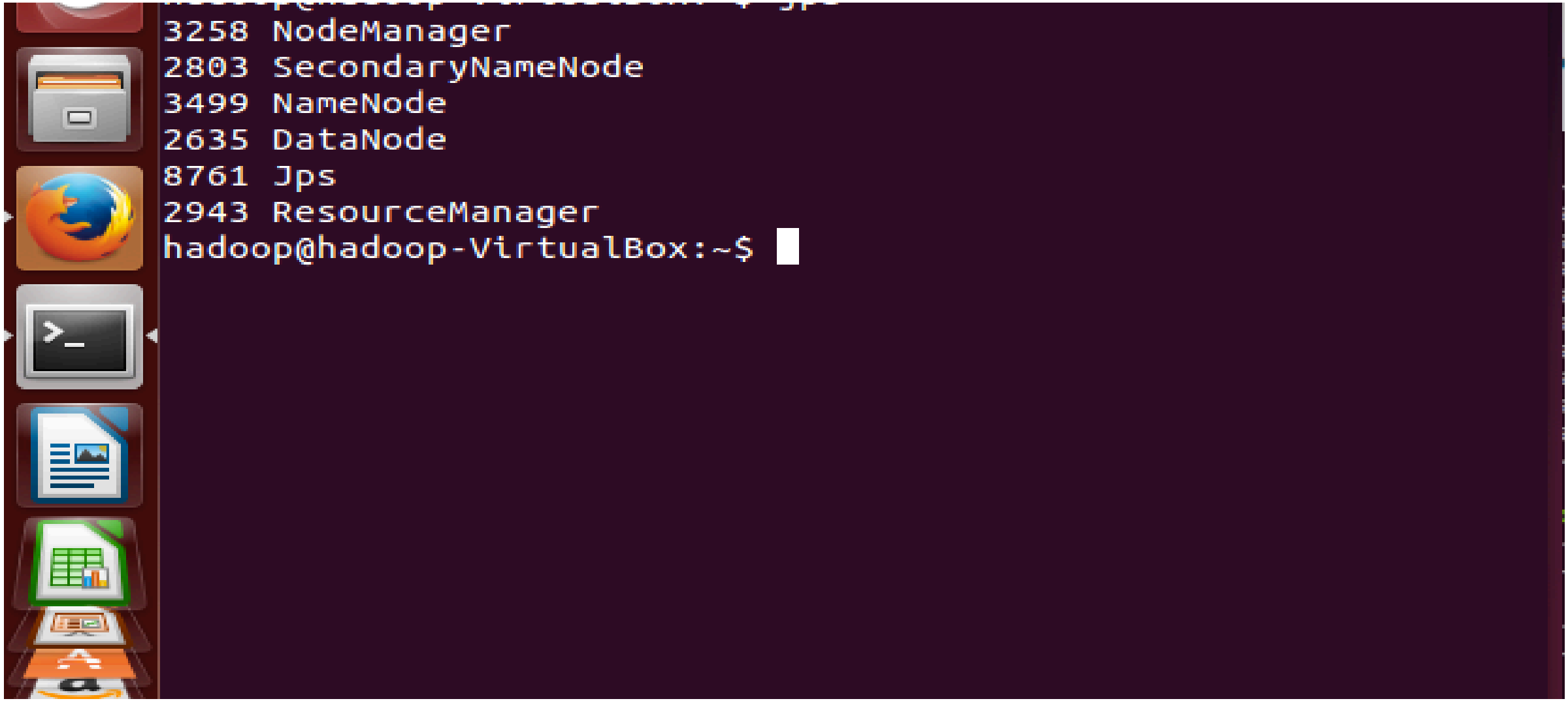
The data that was used in this experiment were from Wikipedia's data dump site

RESEARCH & IMPLEMENTATION

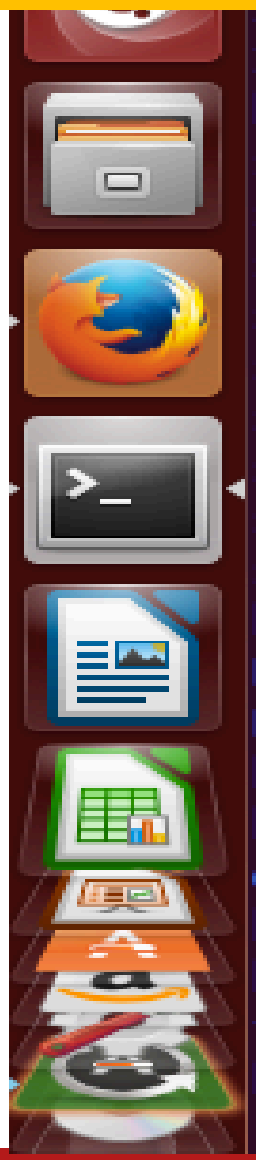
Index of /enwiki/latest/

../		
enwiki-latest-abstract.xml	16-May-2015 10:07	4562685095
enwiki-latest-abstract.xml-rss.xml	16-May-2015 10:07	751
enwiki-latest-abstract1.xml	16-May-2015 06:07	748232458
enwiki-latest-abstract1.xml-rss.xml	16-May-2015 10:03	754
enwiki-latest-abstract10.xml	16-May-2015 05:27	160692277
enwiki-latest-abstract10.xml-rss.xml	16-May-2015 10:05	757
enwiki-latest-abstract11.xml	16-May-2015 07:08	154936835
enwiki-latest-abstract11.xml-rss.xml	16-May-2015 10:05	757
enwiki-latest-abstract12.xml	16-May-2015 08:36	172178507
enwiki-latest-abstract12.xml-rss.xml	16-May-2015 10:05	757
enwiki-latest-abstract13.xml	16-May-2015 09:50	163055214
enwiki-latest-abstract13.xml-rss.xml	16-May-2015 10:05	757
enwiki-latest-abstract14.xml	16-May-2015 09:57	156122136
enwiki-latest-abstract14.xml-rss.xml	16-May-2015 10:05	757
enwiki-latest-abstract15.xml	16-May-2015 09:06	142645608
enwiki-latest-abstract15.xml-rss.xml	16-May-2015 10:05	757
enwiki-latest-abstract16.xml	16-May-2015 09:45	153311991
enwiki-latest-abstract16.xml-rss.xml	16-May-2015 10:05	757
enwiki-latest-abstract17.xml	16-May-2015 07:28	150961156
enwiki-latest-abstract17.xml-rss.xml	16-May-2015 10:06	757
enwiki-latest-abstract18.xml	16-May-2015 10:03	138412668
enwiki-latest-abstract18.xml-rss.xml	16-May-2015 10:06	757
enwiki-latest-abstract19.xml	16-May-2015 06:45	150821976
enwiki-latest-abstract19.xml-rss.xml	16-May-2015 10:06	757

RESEARCH & IMPLEMENTATION

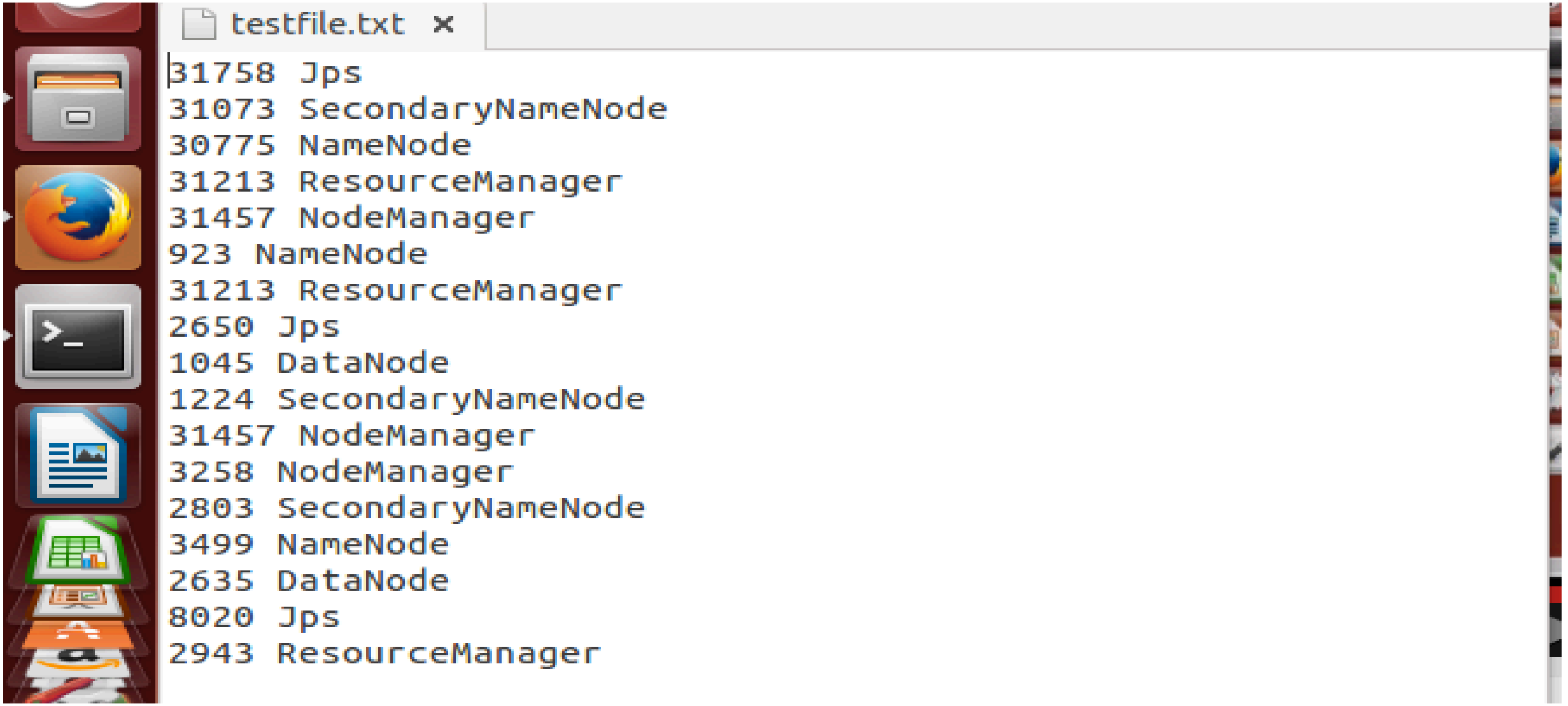


RESEARCH & IMPLEMENTATION

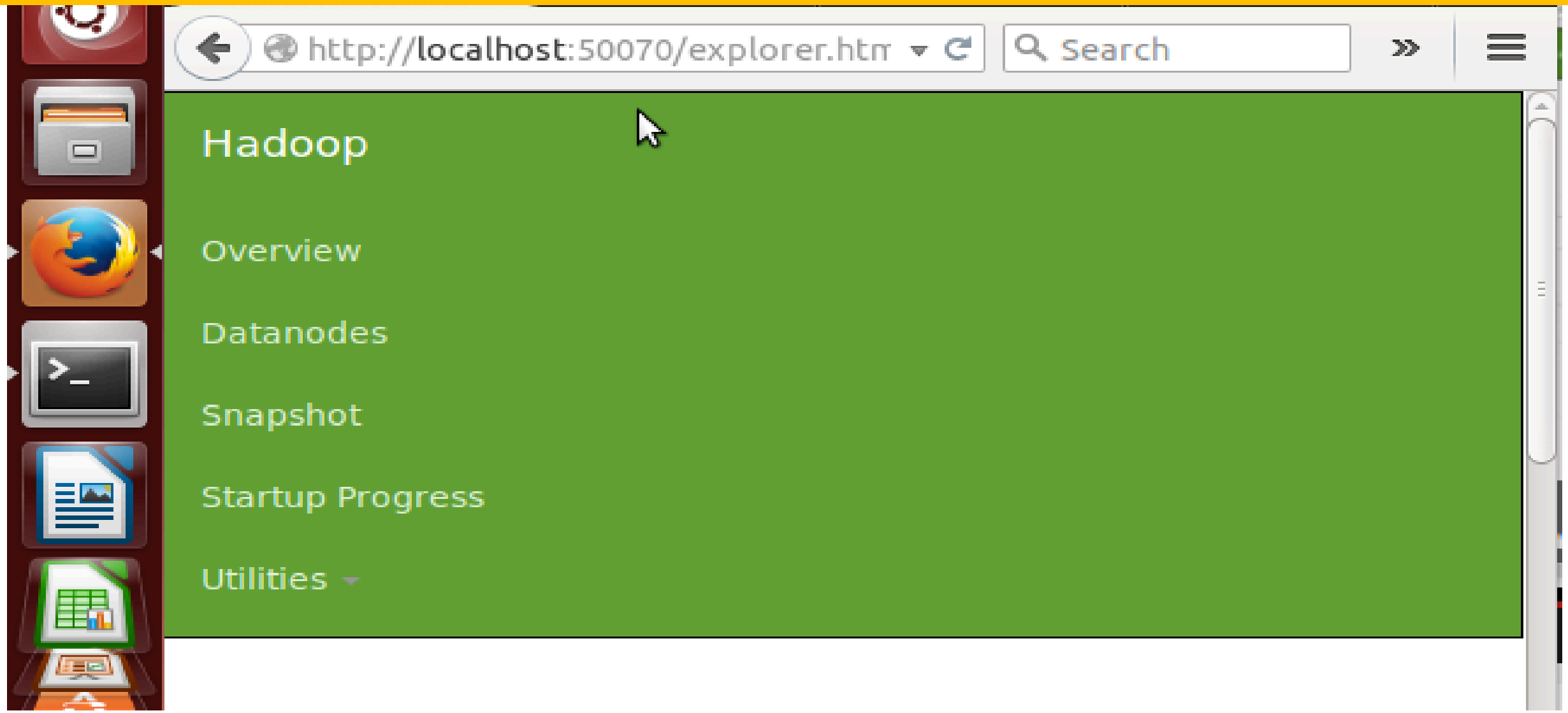


```
2943 ResourceManager
hadoop@hadoop-VirtualBox:~$ sta
start                          start-pulseaudio-x11
start-all.cmd                  start-secure-dns.sh
start-all.sh                   start-stop-daemon
start-balancer.sh              startx
start-dfs.cmd                  start-yarn.cmd
start-dfs.sh                   start-yarn.sh
startpar                       stat
startpar-upstart-inject        static-sh
start-pulseaudio-kde           status
hadoop@hadoop-VirtualBox:~$ ls
Desktop    Downloads           Music     Public     Videos
Documents  examples.desktop    Pictures  Templates
hadoop@hadoop-VirtualBox:~$ cd Desktop/
hadoop@hadoop-VirtualBox:~/Desktop$ ls
data  hadoop-2.6.0.tar.gz
hadoop@hadoop-VirtualBox:~/Desktop$ cd data/
hadoop@hadoop-VirtualBox:~/Desktop/data$ ls
testfile.txt
hadoop@hadoop-VirtualBox:~/Desktop/data$ jps >>testfile.txt
hadoop@hadoop-VirtualBox:~/Desktop/data$
```

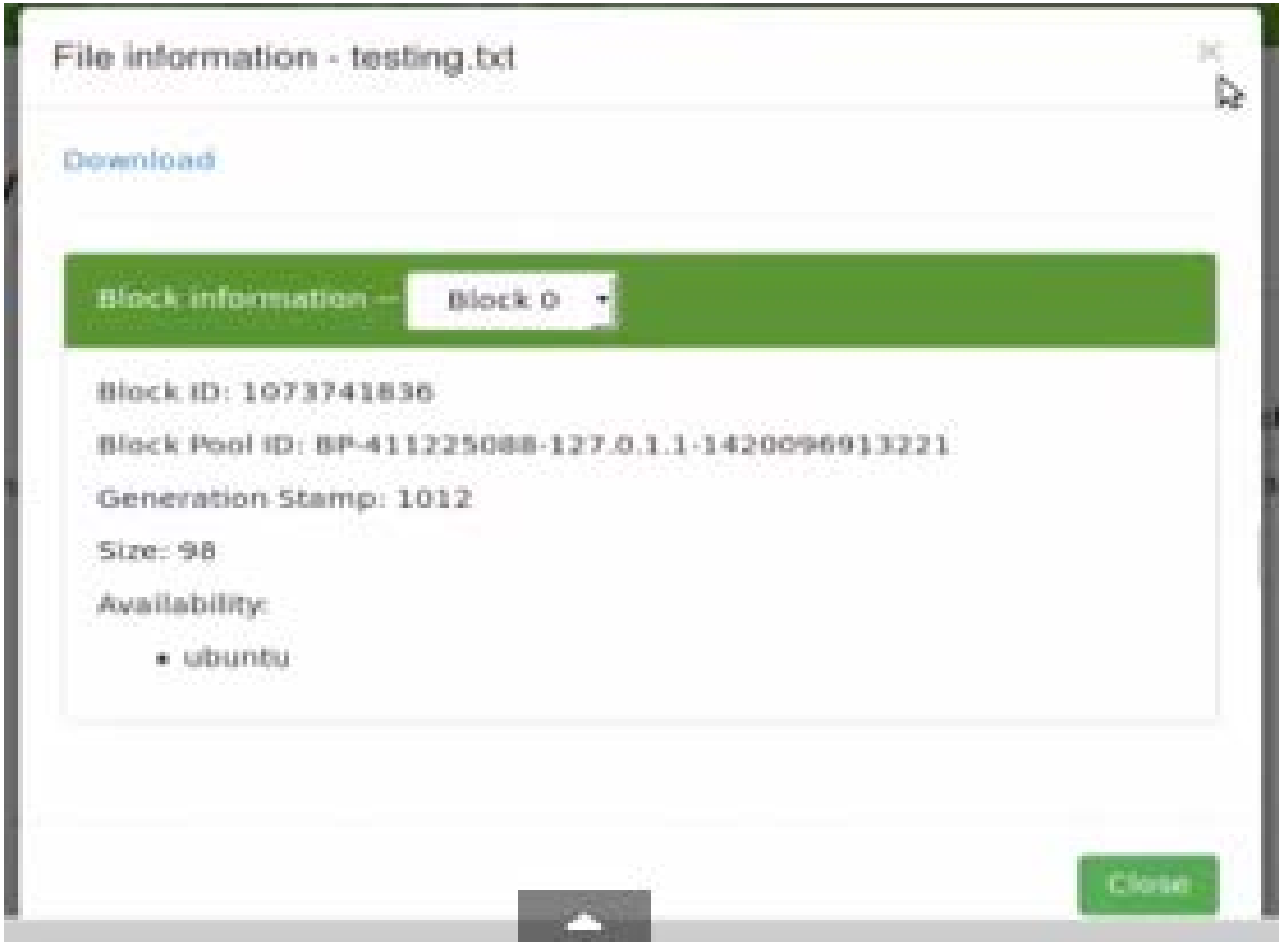
RESEARCH & IMPLEMENTATION



RESEARCH & IMPLEMENTATION



RESEARCH & IMPLEMENTATION



CONCLUSIONS

- When name node memory usage is higher than 75%, reading and writing HDFS files through Hadoop API shall fail; caused by Java Hashmap

- Name node's QPS (query per second) can be optimized by fine-tuning the handler counts higher than the number of tasks that are run in parallel

- The memory available affects the namenode's performance in each OS platform; therefore allocation of memory is essential

- Since memory has such an effect performance, the operating system utilizes less RAM for background and utility features to yield better results

- Ubuntu Linux would be the ideal environment since it uses the least amount of memory, followed by Mac OSX and then finally Windows

THANK YOU!!!

QUESTIONS