

# *Hadoop in Application*

P95922005 廖宜財

P95922006 曾威霖

P95922009 洪家頌

## **Abstract**

Hadoop is a software platform that lets one easily write and run applications that process vast amounts of data. In this introduction, we will present how to adopt this framework into our daily job/task and how to enhance our current process system.

## **1. Introduction**

Hadoop is a software platform that lets one easily write and run applications that process vast amounts of data. Here's what makes Hadoop especially useful:

- **Scalable:** Hadoop can reliably store and process petabytes.
- **Economical:** It distributes the data and processing across clusters of commonly available computers. These clusters can number into the thousands of nodes.
- **Efficient:** By distributing the data, Hadoop can process it in parallel on the nodes where the data is located. This makes it extremely rapid.
- **Reliable:** Hadoop automatically maintains multiple copies of data and automatically redeploys computing tasks based on failures.

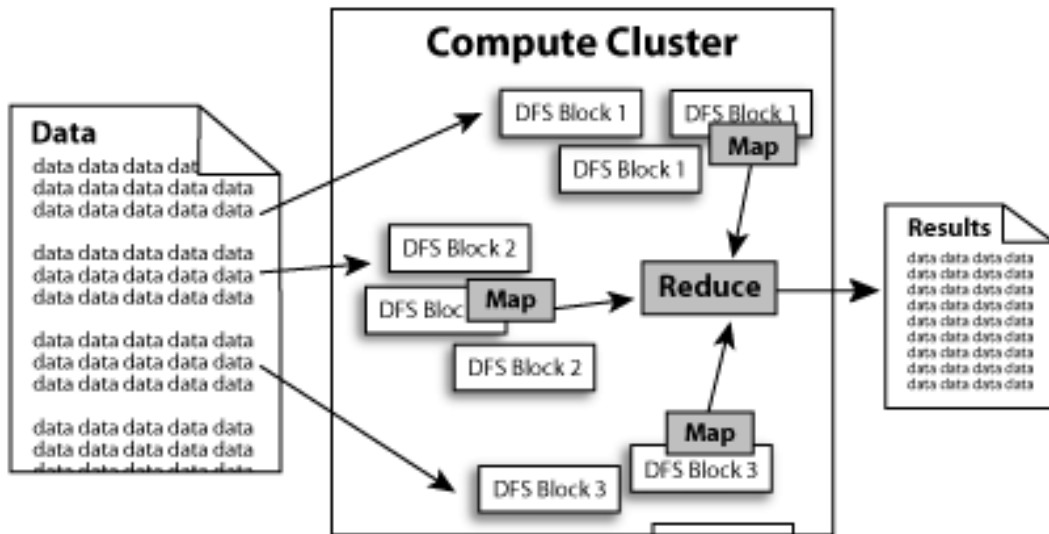
Hadoop implements MapReduce, using the Hadoop Distributed File System (HDFS) (see figure below.) MapReduce divides applications into many small blocks of work. HDFS creates multiple replicas of data blocks for reliability, placing them on compute nodes around the cluster. MapReduce can then process the data where it is located.

Hadoop has been demonstrated on clusters with 2000 nodes. The current design target is 10,000 node clusters.

Here are Hadoop' information from this introduction:

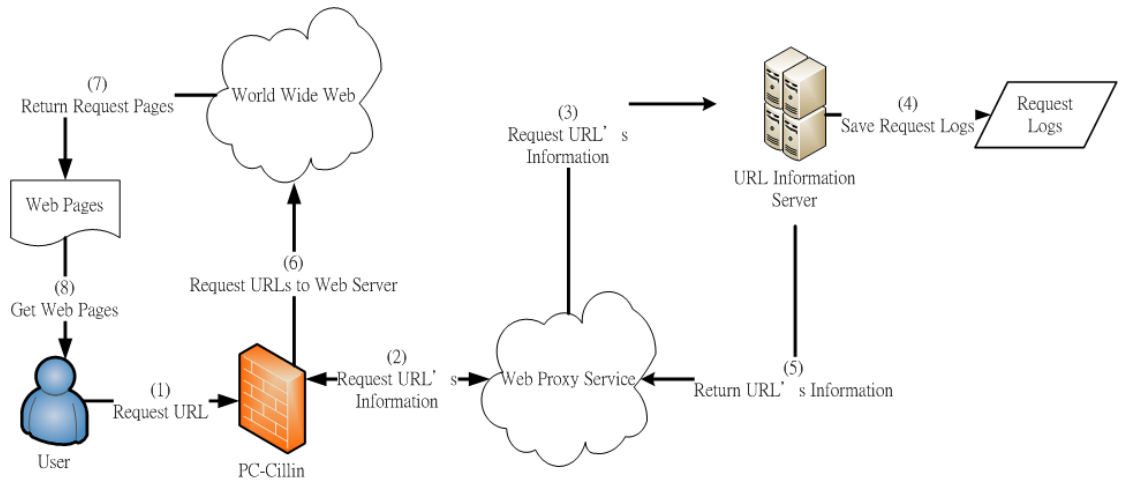
- Operate scalability
  - Terabytes(perabytes) of data
  - Large than RAM, disk i/o required
- Operate economically
  - minimize \$ per cycle, ram, & i/o

- thus use network of commodity PCs
- Operate reliably



## 2. Our Problem

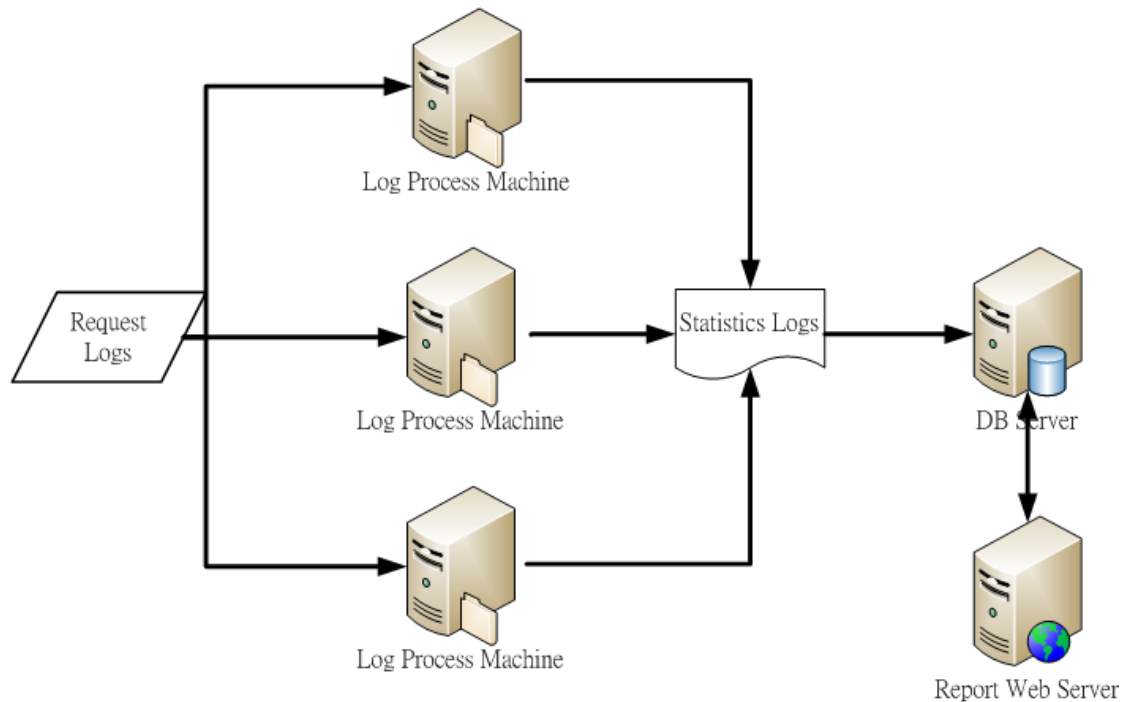
- Project: Web Thread Prevention(WTP)



Our request is almost 3.5 billion of URLs. We need to process and analyze them in the realtime.

- Input:
  - Total Request: 3.5 billions of URLs
  - Total Volume: Avg. 7.5TB
  - Hits: Avg. 40,000 hits/sec
  - Peak: Avg. 80,000 hits/sec
  - All the data is increasing...
- Output: Several kinds of report for analyzing web threats.
  - Unique URLs

- Top 10 hits(URLs/Domains)
- Countries distribution
- ...
- Real-time
- Current Architecture:



- Weak:
  - Process scalability(Ram, Disk spaces, I/O...)
  - Difficult to modify process
  - Response to slow(12~16 hours/day)
  - Machines maintenance

### 3. Why/How Hadoop

- Scalable
  - Hadoop can reliably store and process petabytes.
- Economical
  - It distributes the data and processing across clusters of commonly available computers. These clusters can number into the thousands of nodes.
- Efficient
  - By distributing the data, Hadoop can process it in parallel on the nodes

where the data is located. This makes it extremely rapid.

- Reliable

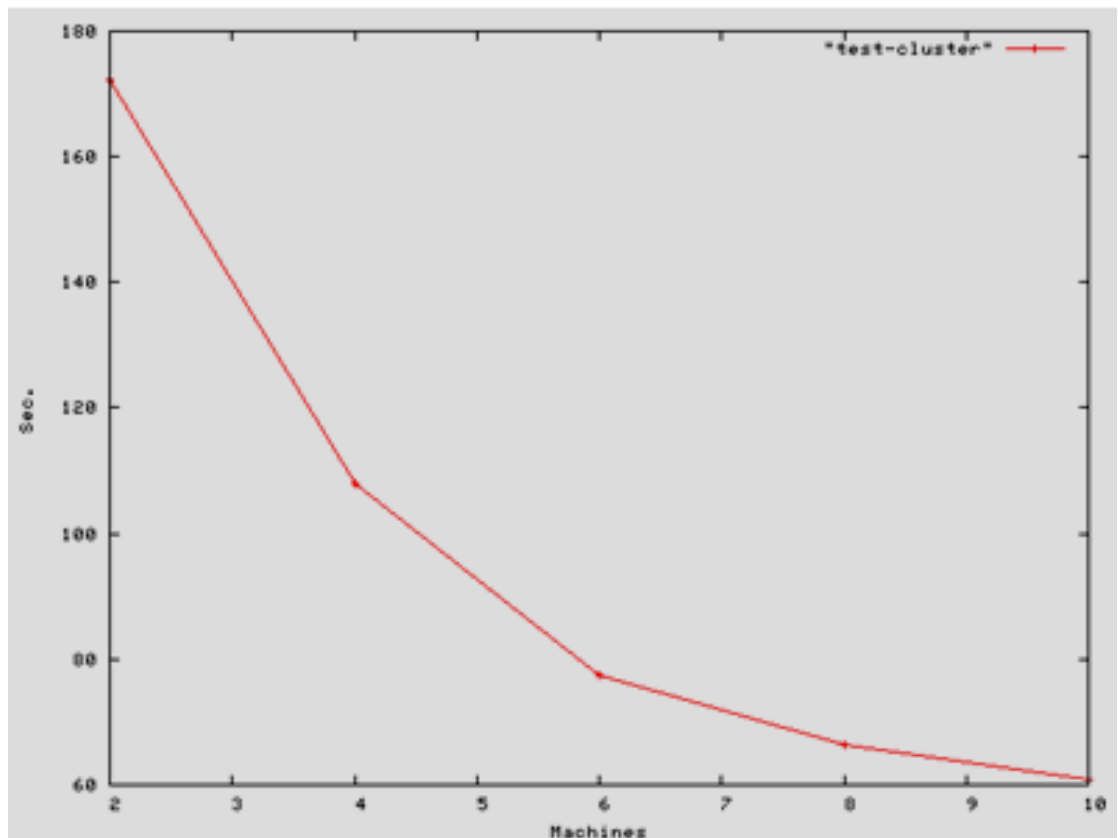
- Hadoop automatically maintains multiple copies of data and automatically redeploys computing tasks based on failures.

- Benchmark from Hadoop

- 1.5 GB access\_log on 10 node cluster
- This test should include the data load time for the MySQL column, not just the SQL time.

	MySql 5.0.27	Hadoop-0.15.2	Hadoop-0.15.2
Data	B-tree disk table (MyISAM)	Text files (access_log)	Text files (access_log)
Machine	1	2	4
Rows	5,914,669	5,914,669	5,914,669
Results	100	100	100
Time	4.43 sec	172.30 sec	108.01 sec

Hadoop-0.15.2	Hadoop-0.15.2	Hadoop-0.15.2
Text files (access_log)	Text files (access_log)	Text files (access_log)
6	8	10
5,914,669	5,914,669	5,914,669
100	100	100
77.41 sec	66.30 sec	60.78 sec



- Our Testing
- Machine Spec.
  - P4-Xeon 2.8G(dual cpu), 2G Ram, 250G HD
- Process
  - Using hash data in perl to compute URLs
- Output
  - Unique URLs
- Our Map Program

```

root@hadoop-jobtracker:/usr/hadoop/hadoop-0.16.0
#!/usr/bin/perl -w

while (<STDIN>)
{
    chomp ;
    my $strLine = $_ ;
    my @LineBuf = split("\t", $strLine) ;
    print "$LineBuf[0]\t1\n" ;
}

```

- Our Reduce Program

```

root@hadoop-jobtracker:/usr/hadoop/hadoop-0.16.0
#!/usr/bin/perl -w

my %URLData ;
while (<STDIN>)
{
    chomp ;
    my $strLine = $_ ;
    my @LineBuf = split("\t", $strLine) ;
    $URLData{$LineBuf[0]} += int($LineBuf[1]) ;
}

foreach my $strURL (keys(%URLData))
{
    print "$strURL\t$URLData{$strURL}\n" ;
}

```

Network 100M	single machine	Hadoop 0.16.0
Machine	1	1
Rows	2M (2120224)	2M (2120224)
Process Time	24.6s	2m4.8s

Network 100M	single machine	Hadoop 0.16.0
Machine	1	1
Rows	8M (8238366)	8M (8238366)
Process Time	1m36s	7m27s

Network 100M	single machine	Hadoop 0.16.0
Machine	1	1

Rows	<b>80M (82383660)</b>	80M (82383660)
Process Time	<b>18m11.6s</b>	80m57s

Hadoop 0.16.0	Hadoop 0.16.0	Hadoop 0.16.0
2	4	7
2M (2120224)	2M (2120224)	2M (2120224)
1m21.6s	57.4s	48.4s

Hadoop 0.16.0	Hadoop 0.16.0	Hadoop 0.16.0
2	4	7
8M (8238366)	8M (8238366)	8M (8238366)
4m4s	2m38s	2m4s

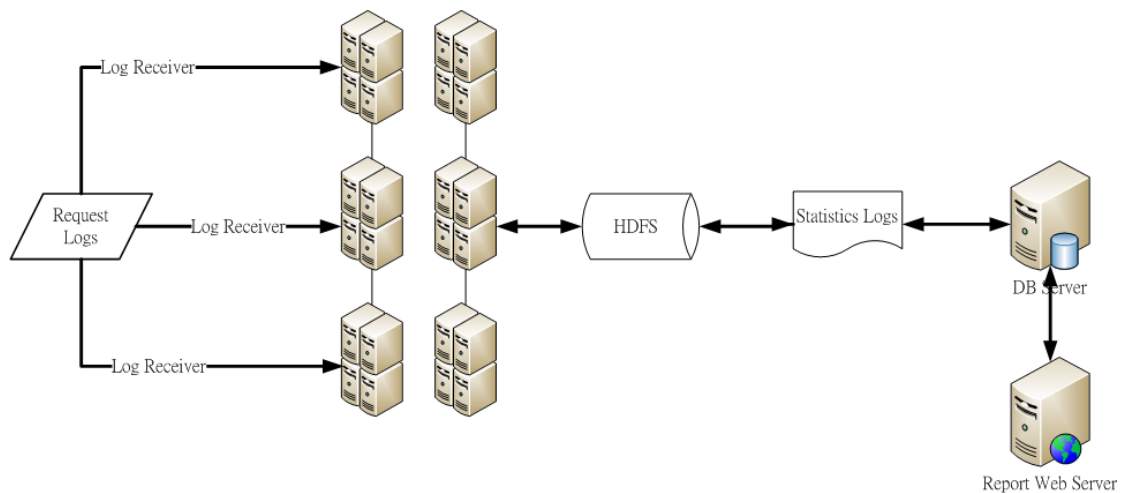
Hadoop 0.16.0	Hadoop 0.16.0	<b>Hadoop 0.16.0</b>
2	4	<b>7</b>
80M (82383660)	80M (82383660)	<b>80M (82383660)</b>
43m1s	21m15s	<b>15m7s</b>

- Very simple things will not be better than using Hadoop
- Huge data also could use simple way to process it, but the process will be

more complicated than using Hadoop.

- Easy to implement & run process with Hadoop.
- Difficult to (implement) maintain or optimize the process without Hadoop.
- Fault Tolerance Control

## 4. Application to Hadoop



- For any computing process(all kinds of reports), just only to provide [Map/Reduce] function
- The machine could be expand more easily
- Only for computing, Not a DB
- For data writing into HDFS needs to cost some time
- Data convert from HDFS to public spaces
- Name-Node(Master)'s META data backup & restore.

## 5. In the future

- Where to pay for use?
  - Running on Amazon EC2 & SC3
  - <http://wiki.apache.org/hadoop/AmazonEC2>
  - <http://wiki.apache.org/hadoop/AmazonS3>
- For Hadoop
  - User Access Control

- Separating data by rack
- Others...
- In Trend Micro Inc.
  - Web Threat Prevention in the real-time(in the cloud service)
  - SVM for content of web pages analysis
  - E-Mail Clustering System for Spam/Phishing
  - Others...
- In Yahoo Inc.
  - URL Content Analysis
  - Spam Mail Analysis
  - Cooperate with Taiwan IT Company
  - Others...

## 6. Reference

- <http://hadoop.apache.org/core/>
- <http://wiki.apache.org/hadoop/>
- <http://en.wikipedia.org/wiki/Hadoop>
- <http://www.cs.washington.edu/education/courses/cse490h/07sp/index.html>
- **Trend Micro Inc. Internal Projects**