



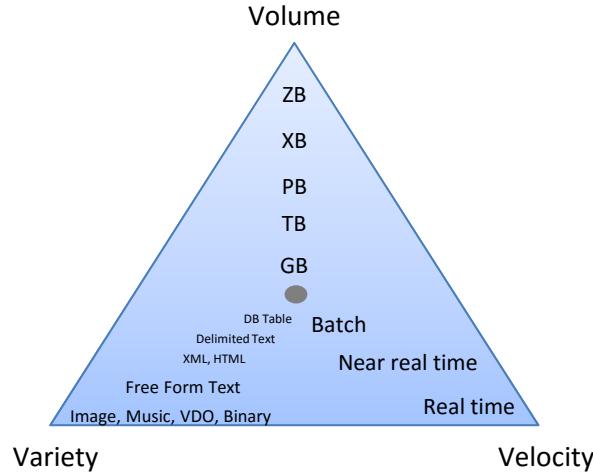
# Learning Big Data

# Hadoop and Spark

- Part 1** Big Data Technology and Example Cases
- Part 2** Hadoop 2.7.2 Architecture
- Part 3** Hands on: Hadoop HDFS and MRv2 on YARN
- Part 4** Spark 2.0 Architecture
- Part 5** Hands on: Spark on Hadoop HDFS
- Part 6** Big Data Project Management

อ.ดนัยรัฐ ธนบดีธรรมจารี  
+668-1559-1446 Line ID: danairat  
FB: <https://www.facebook.com/tdanairat>





# Big Data Introduction

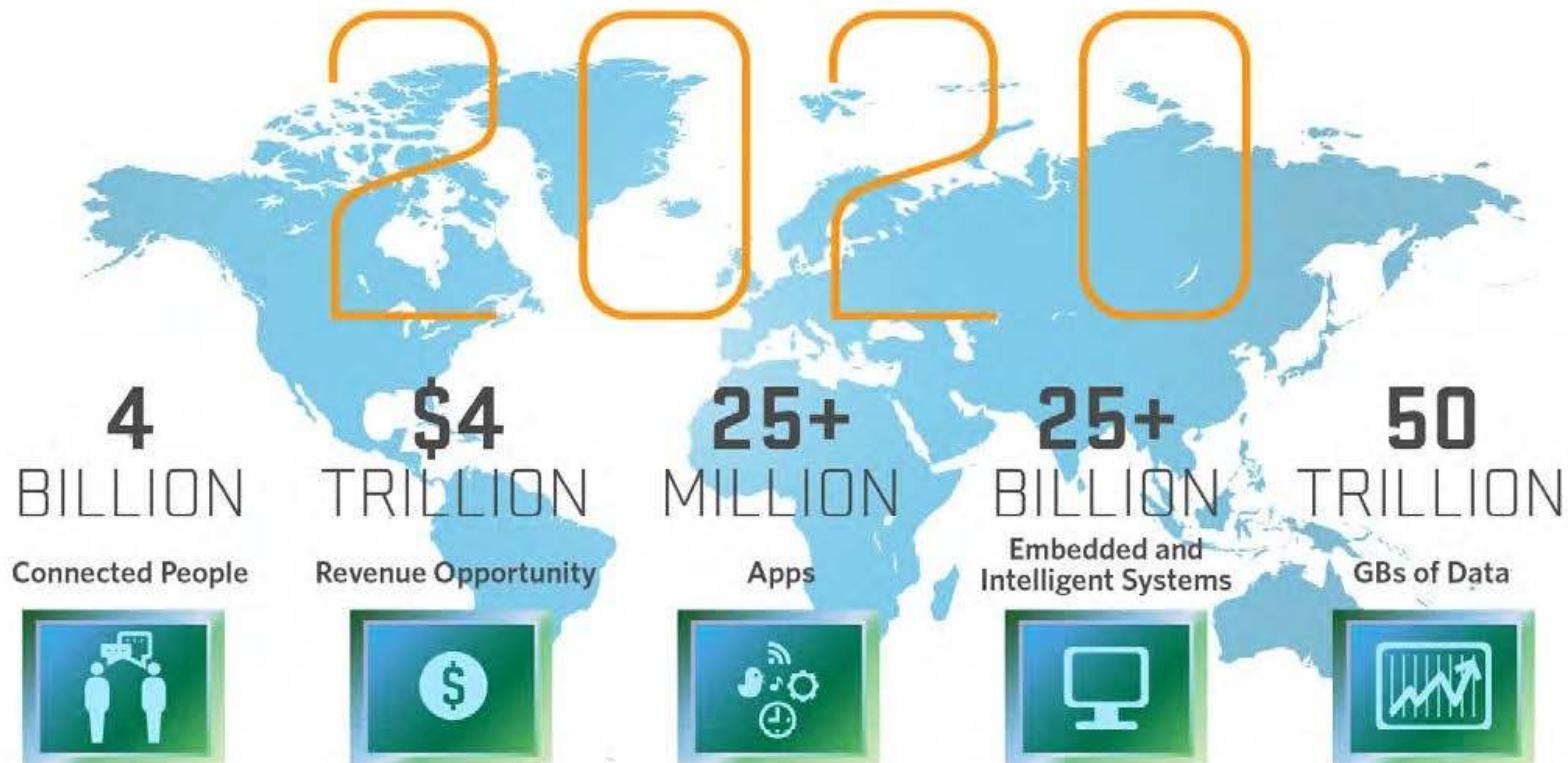
อ.ดนัยรัตน์ ธนบดีธรรมจารี

Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

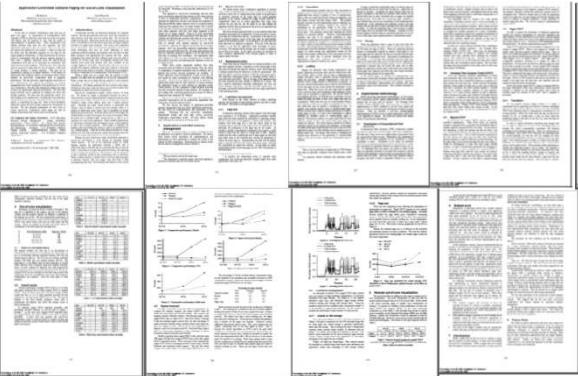
# Why Big Data



Source: Mario Morales, IDC

# A Short History Of Big Data

**October 1997** Michael Cox and David Ellsworth published “[Application-controlled demand paging for out-of-core visualization](#)” in the Proceedings of the IEEE 8th conference on Visualization. When data sets do not fit in main memory (*in core*), or when they do not fit even on local disk, the most common solution is to acquire more resources. It is the first article in the ACM digital library to use the term “**Big Data**”.



**Application-Controlled Demand Paging for Out-of-Core Visualization**

Michael Cox  
MRJ/NASA Ames Research Center  
Microcomputer Research Labs, Intel Corporation  
[<mbc@nas.nasa.gov>](mailto:<mbc@nas.nasa.gov>)

David Ellsworth  
MRJ/NASA Ames Research Center  
[<ellswo@nas.nasa.gov>](mailto:<ellswo@nas.nasa.gov>)

**Abstract**

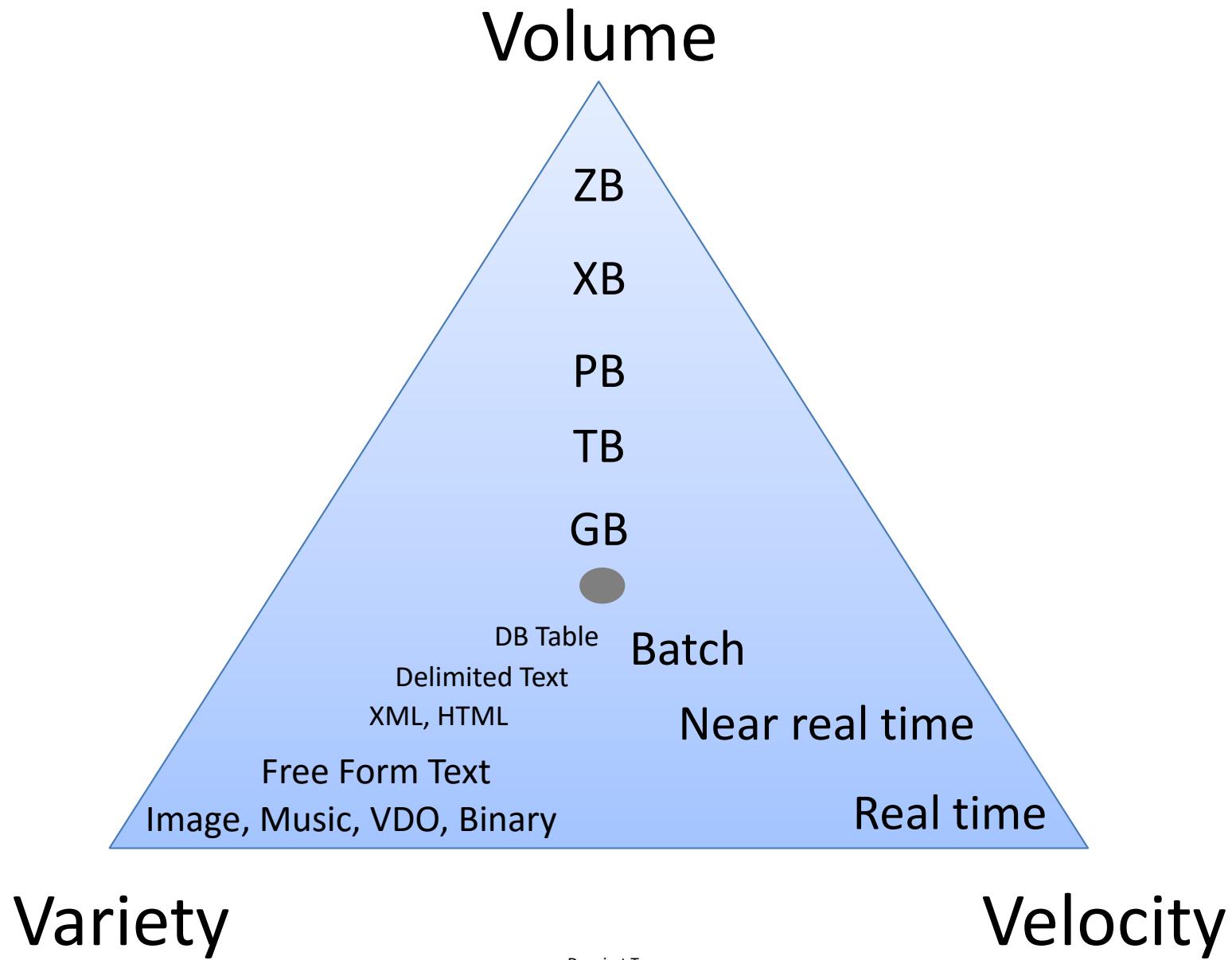
In the area of scientific visualization, input data sets are often very large. In visualization of Computational Fluid Dynamics (CFD) in particular, input data sets today can surpass 100 Gbytes, and are expected to scale with the ability of supercomputers to generate them. Some visualization tools already partition large data sets into segments, and load appropriate segments as they are needed. However, this does not remove the problem for two reasons: 1) there are data sets for which even the individual segments are too large for the largest graphics workstations, 2) many practitioners do not have access to workstations with the memory capacity required to load even a segment, especially since the state-of-the-art visualization tools tend to be developed by researchers with

**1 Introduction**

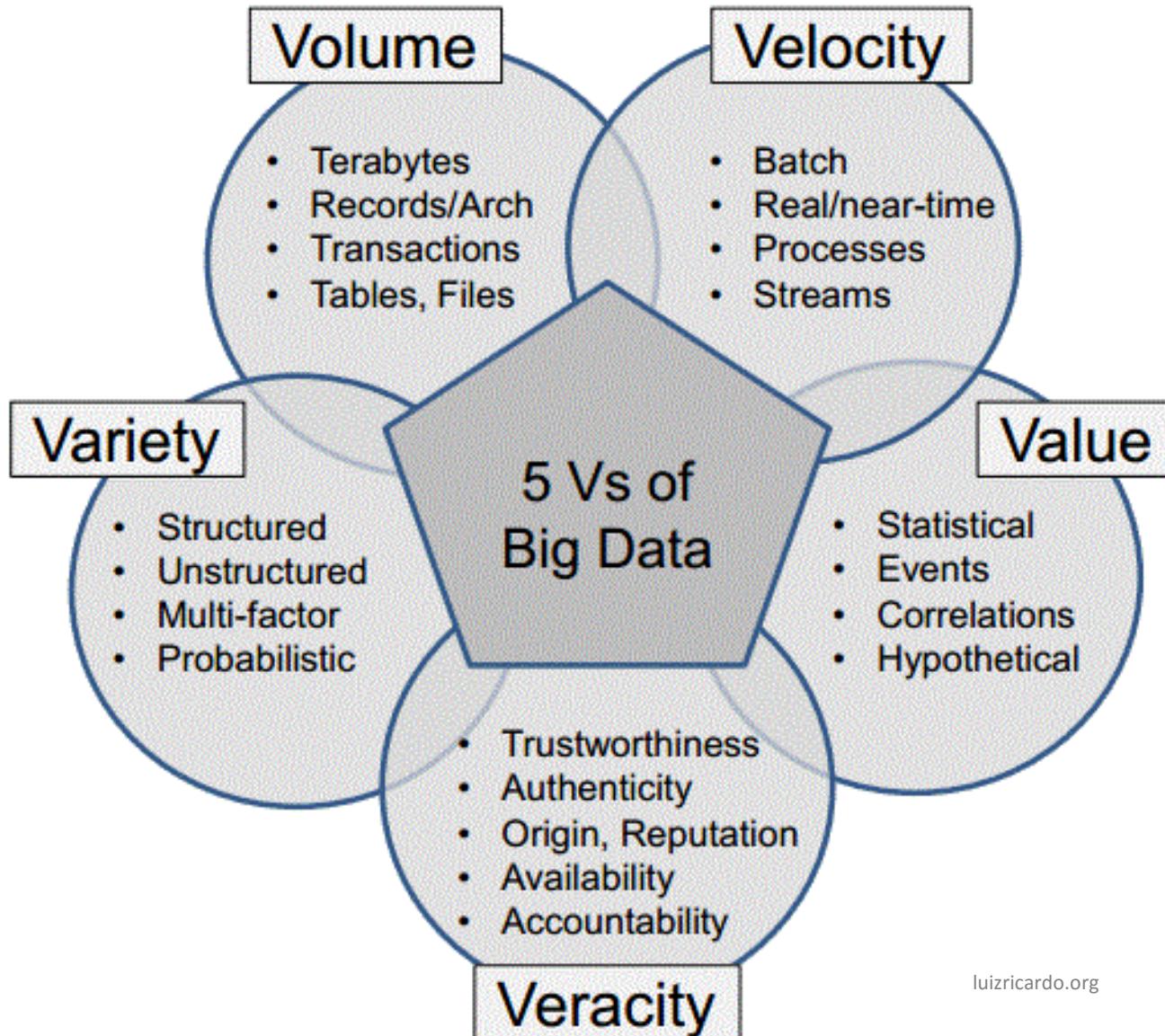
Visualization provides an interesting challenge for computer systems: data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of *big data*. When data sets do not fit in main memory (*in core*), or when they do not fit even on local disk, the most common solution is to acquire more resources. This *write-a-check* algorithm has two drawbacks. First, if visualization algorithms and tools are worth developing, then they are worth deploying to more production-oriented scientists and engineers who may have on their desks machines with significantly less memory and disk. Some researchers have noted that their software tools were not used in practice for several years after development because the tools required more power and memory than were available on the

[https://www.evl.uic.edu/cavern/rg/20040525\\_renabot/Viz/parallel\\_volviz/paging\\_outofcore\\_viz97.pdf](https://www.evl.uic.edu/cavern/rg/20040525_renabot/Viz/parallel_volviz/paging_outofcore_viz97.pdf)

# Introduction to Big Data 3Vs



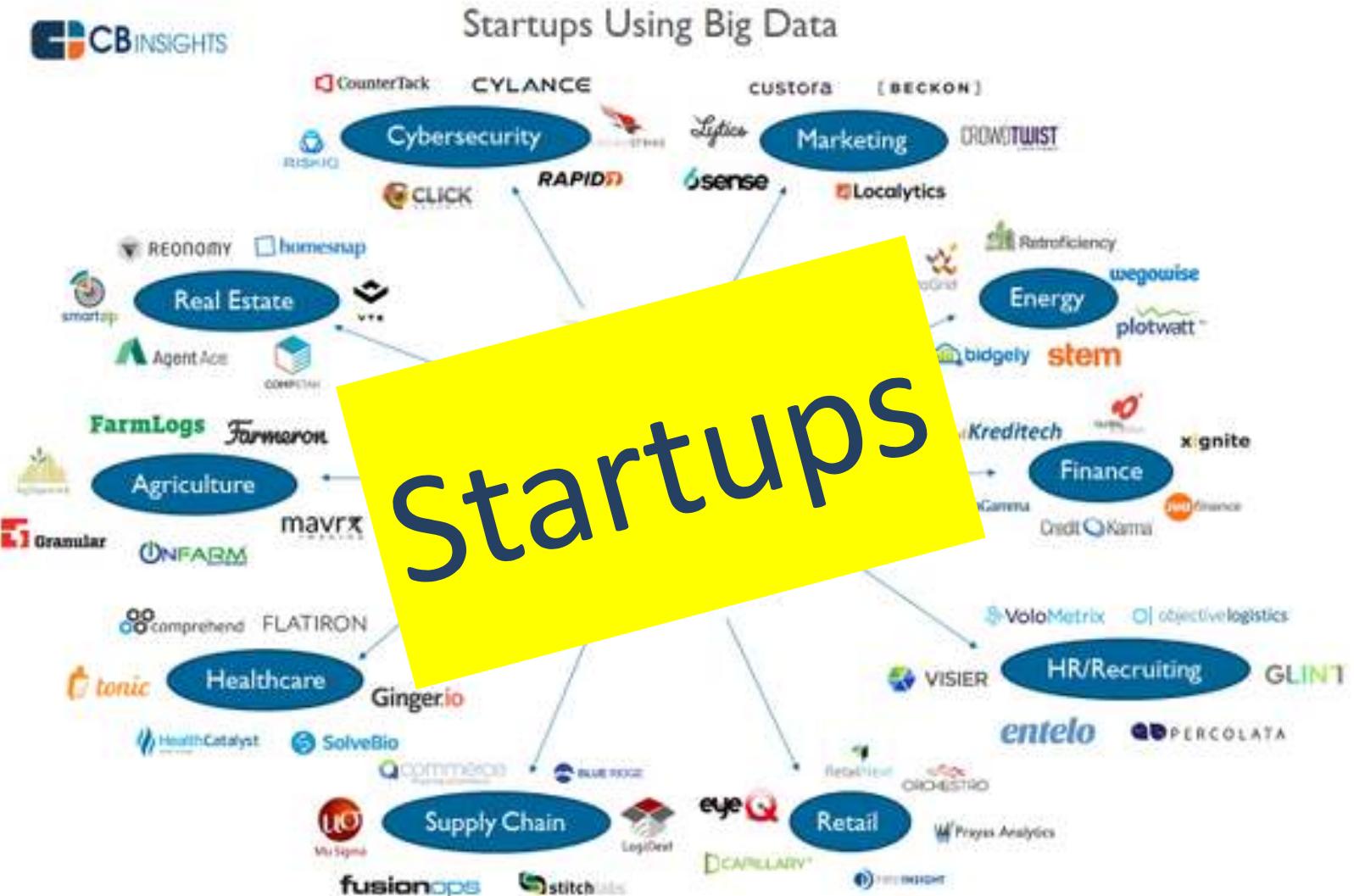
# Introduction to Big Data 5Vs



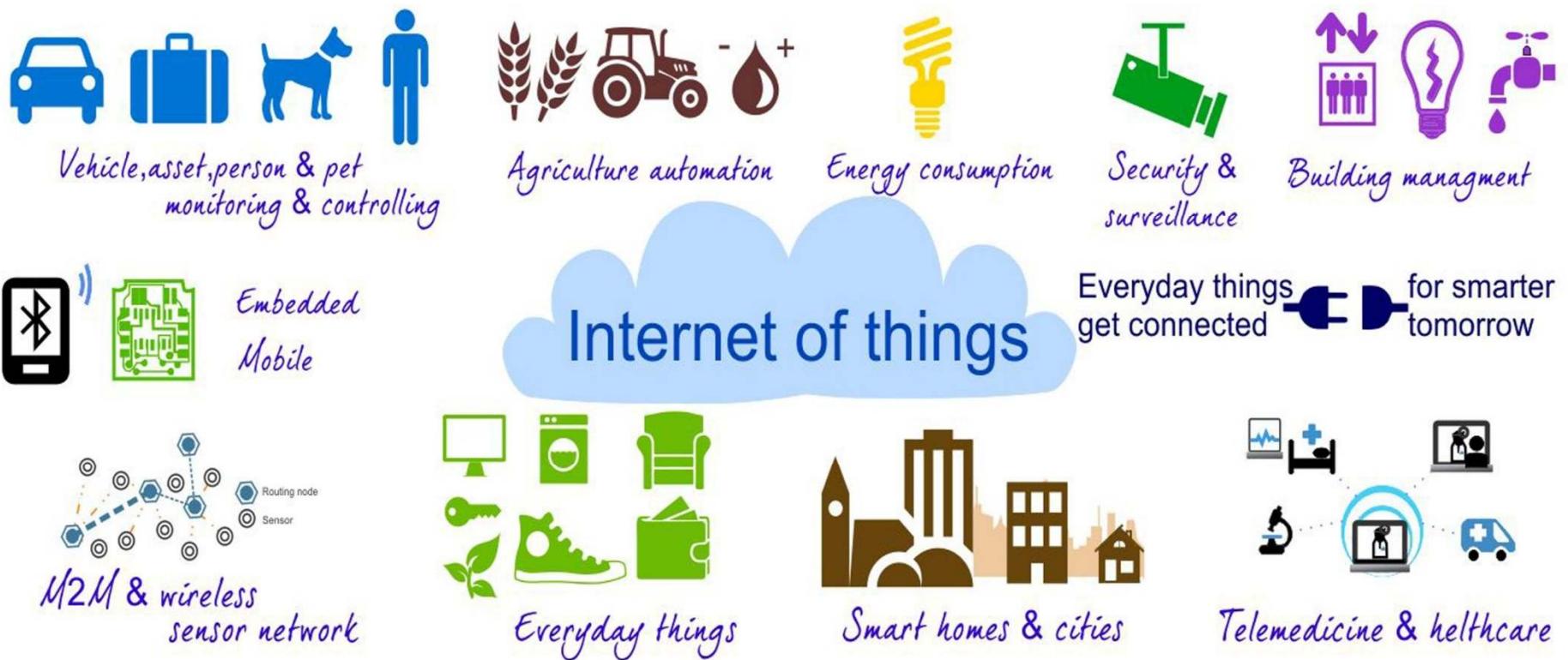
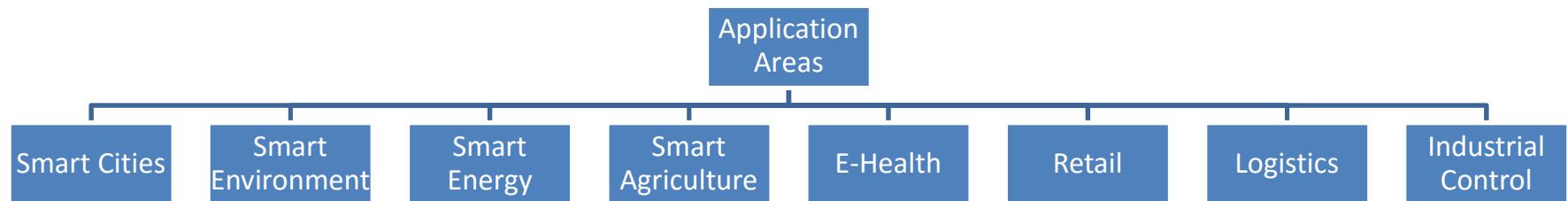
luizricardo.org



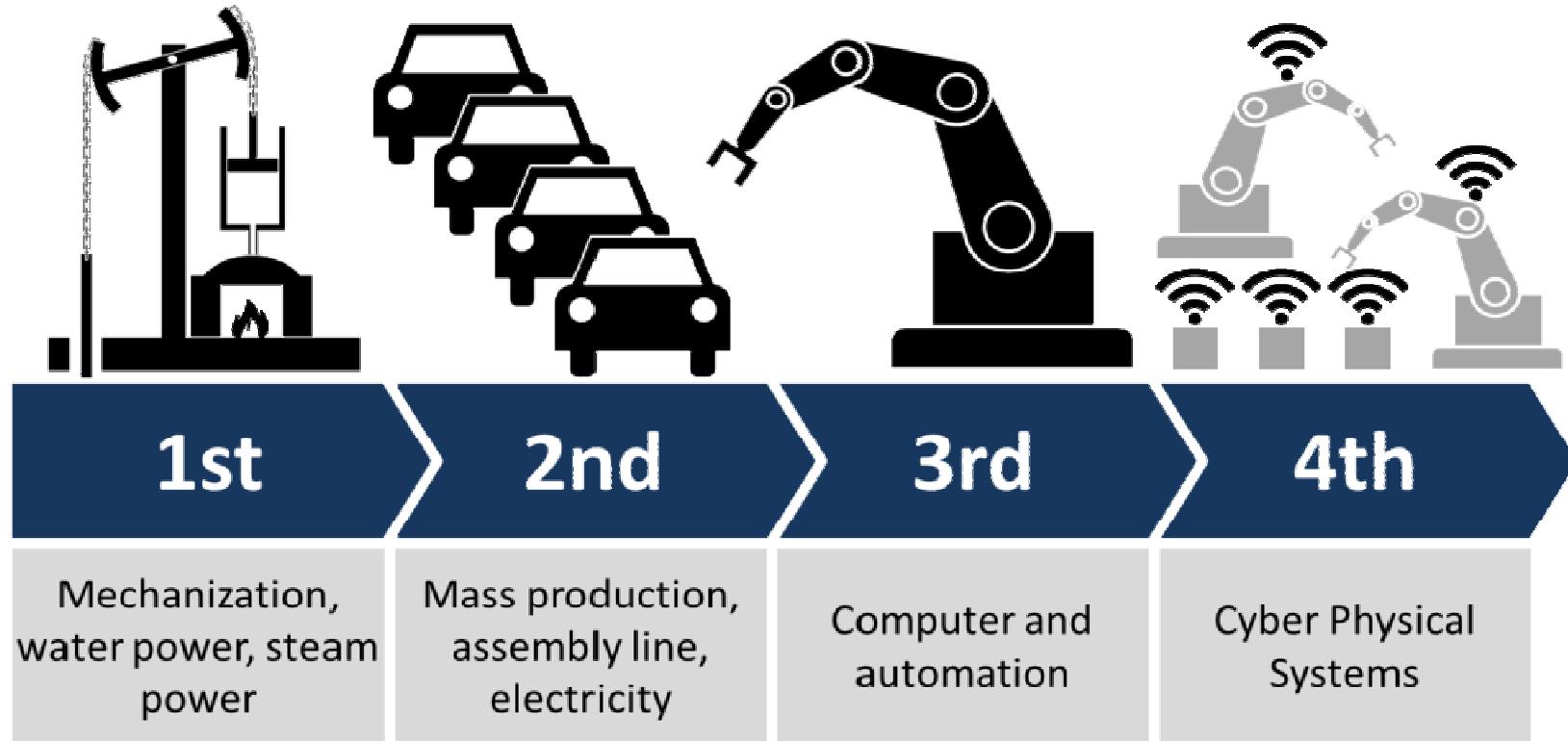
# Startups using Big Data



# Internet of Things

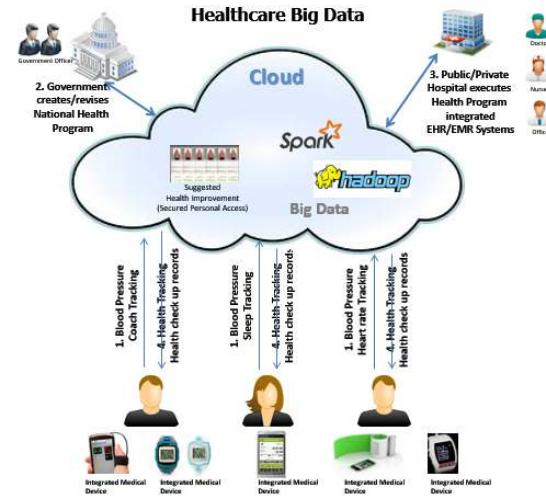


# Industry 4.0



## The impacts of industry 4.0:-

- 1. Services Innovation**
- 2. Business Models**
- 3. Manned and Unmanned**
- 4. Digital Security**



# Big Data Example Cases

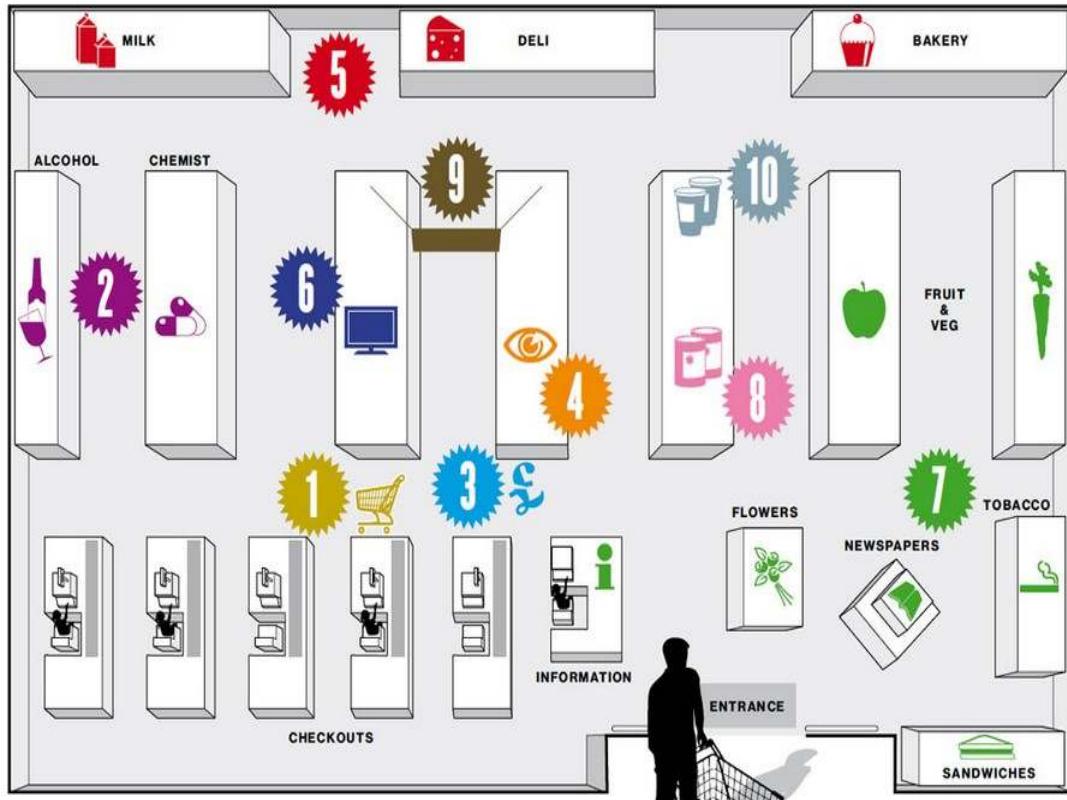
อ.ดนัยรัฐ ธนาบดีธรรมจารี

Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

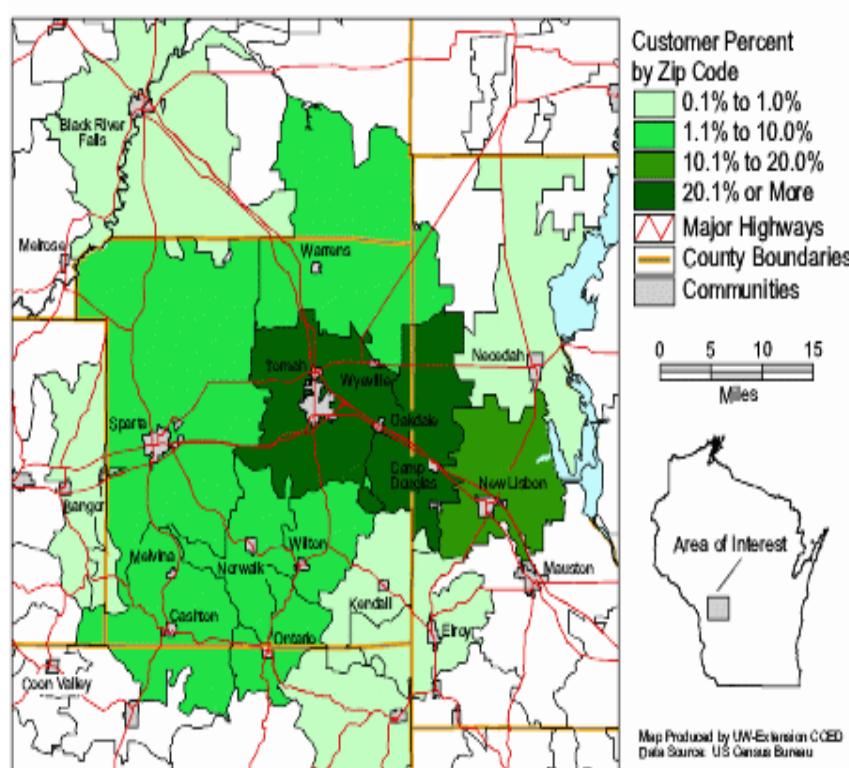
# Retails Store Use Cases



1. **Golden zones**
2. **More expensive items with bigger margins**
3. **Endcaps**
4. **Buy level**
5. **Traffic builder**
6. **Action alley**
7. **Front of shop**
8. **Signpost brands**
9. **Hanging signs and shelf signs**
10. **Range reduction**  
Less can be more. Average household uses 300 products in a year

# Retails Store Use Cases

## Customer Density



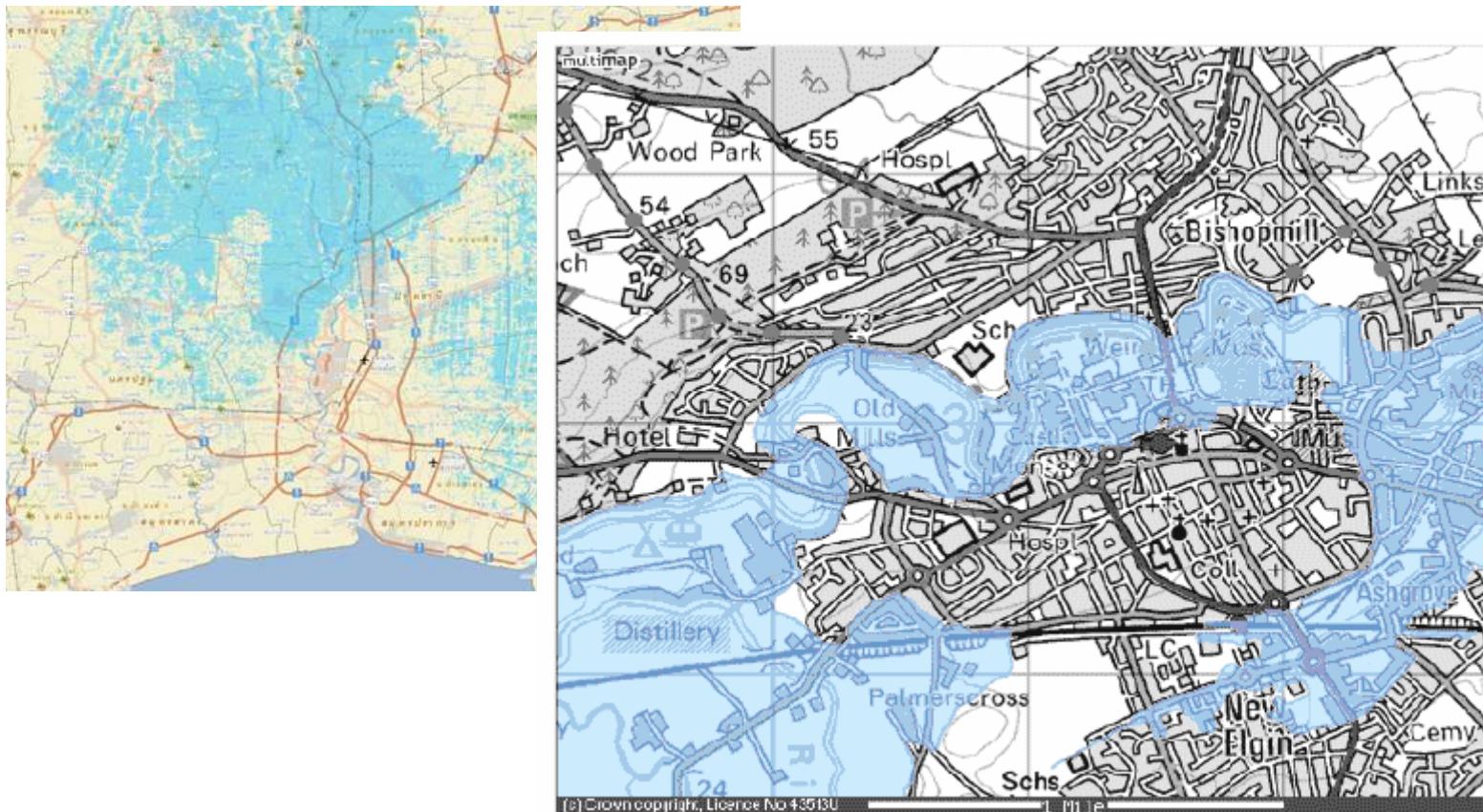
## Store Layout Profits



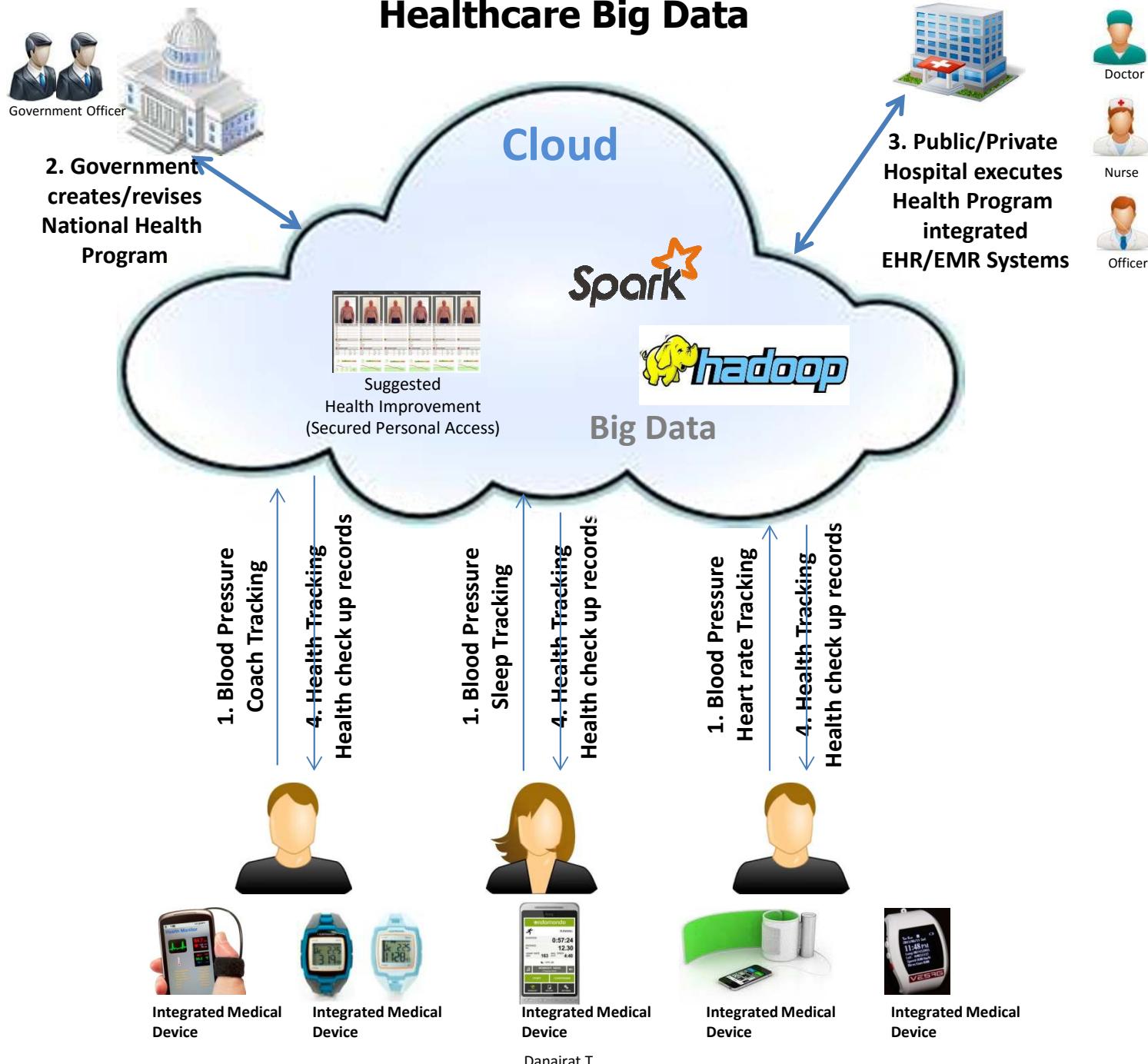
[independent.co.uk](http://independent.co.uk)

# การเตือนภัยพิบัติ

Sensor ปริมาณน้ำฝน น้ำในเขื่อน ปริมาตรเขื่อน, แสดงแนวโน้มสภาวะน้ำท่วม  
Nearly Real-Time



# Healthcare Big Data



# Opportunity and Market Outlook

SECTOR	EXAMPLE APPLICATIONS	MAJOR DRIVER
Smart buildings	Automated monitoring of heating, ventilation and cooling	Reduced energy costs
Smart cities	Street lights that dim when roads are empty	Cost savings
Automotive	Emergency calling and accident alerts	
Leisure	Leisure vehicle and boat tracking	Safety and security
Consumer electronics	Connected satellite navigation devices to monitor traffic jams	Production innovation
Health	Remote monitoring of patients and personal health monitoring	Cheaper, home-based care
Utilities	Smart meters and energy demand response	Regulatory requirement
Transportation and logistics	Fleet optimization and supply-chain tracking and tracing	Cost savings
Retail	Wireless payments	Retail innovation
Manufacturing	Predictive maintenance through improved system monitoring	Reduced maintenance costs
Construction	Monitoring usage of equipment to improve efficiency and cut fuel usage	Cost savings
Agriculture and extraction	Remote monitoring of farm or mining operations and equipment	Proactive maintenance
Emergency services and national security	Disaster response and critical infrastructure protection	Faster response times

Source: Machina Research



# Apache Software Foundation?

อ.ดนัยรัตน์ ธนาบดีธรรมจารี

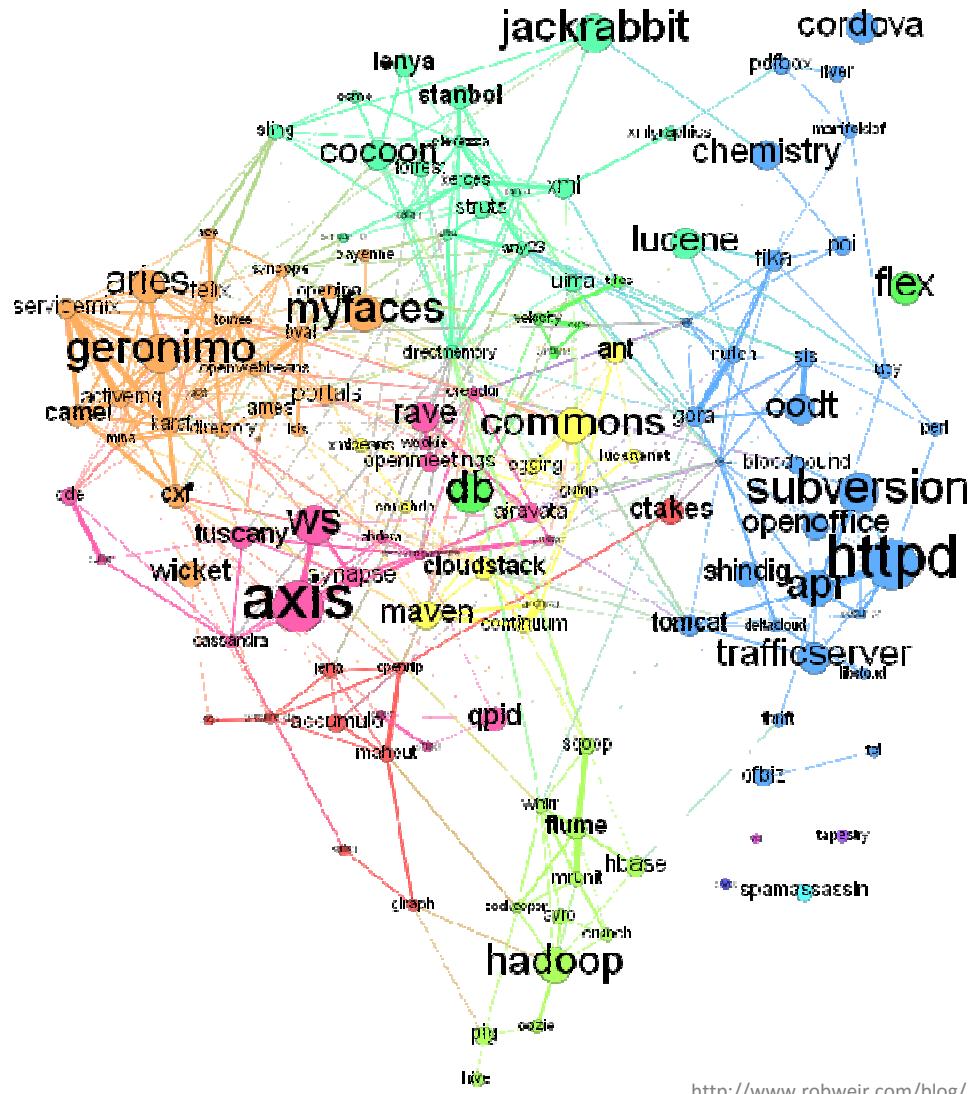
Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

# Apache Software Foundation

Established in 1999, the ASF is a US 501(c)(3) charitable organization, funded by individual donations and corporate sponsors. Our all-volunteer board oversees more than 350 leading Open Source projects, including Apache HTTP Server -- the world's most popular Web server software.



<http://www.robweir.com/blog/2013/05/mapping-apache.html>

# Apache Open Source Project



Over **300** Open Source Projects  
Over **25** different programming languages

<http://www.slideshare.net/WCGWorld/matt-franklin-apache-software-geekfest>

# Apache Open Source Project



Over **300** Open Source Projects  
Over **25** different programming languages

<http://www.slideshare.net/WCGWorld/matt-franklin-apache-software-geekfest>



# Apache Hadoop

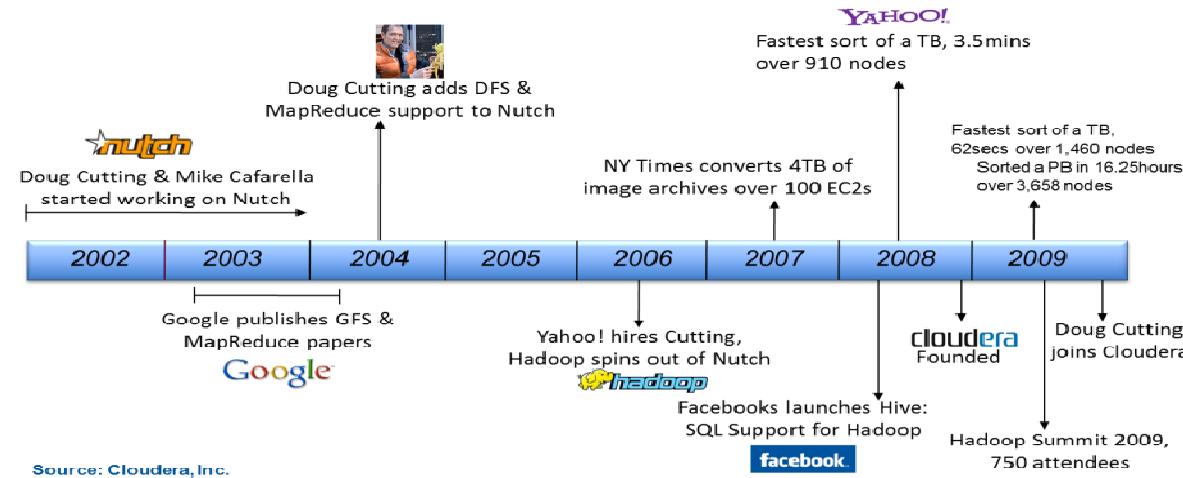
อ.ดนัยรัตน์ ธนาบดีธรรมจารี

Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

# Hadoop History



- **2003:** Google publishes GFS and MapReduce papers
- **2004:** Cutting adds DFS & MapReduce support to Nutch
- **2006:** Yahoo! hires Cutting
- **2007:** NY Times converts 4TB of archives over 100 EC2s
- **2008:** Yahoo does fastest sort of a TB, 3.5mins over 910 nodes
- **2009:** First Hadoop Summit

[apache.org/hadoop/](http://apache.org/hadoop/)

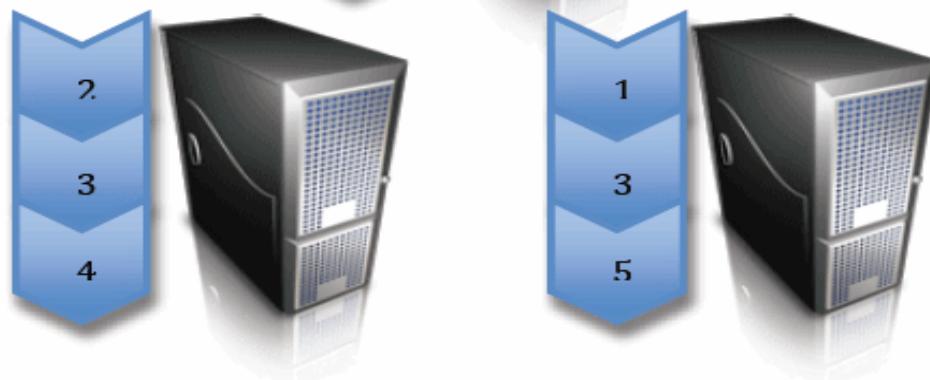
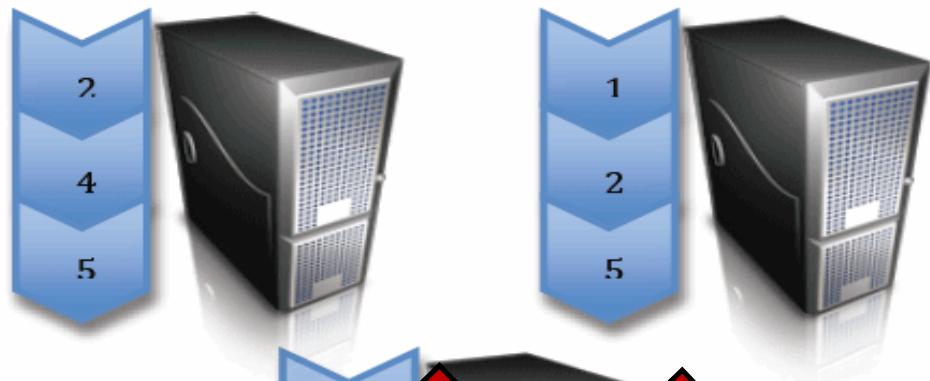
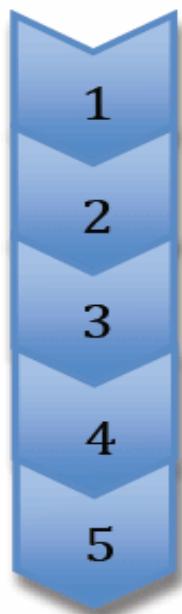
# What is Hadoop?



- Am apache open source **scalable fault-tolerant distributed system for data storage and processing**:-
  - **HDFS**: Self-Healing High-Bandwidth Clustered Storage
  - **YARN**: Resources Management
  - **MapReduce**: Fault-Tolerant Distributed Processing
  - **Common**: Common utilities for Hadoop modules.
- Support both structured and complex data
- Large ecosystem eg. Spark, Cassandra, Ambari, ...

# HDFS: Hadoop Distributed File System

Block Size = 64MB  
Replication Factor = 3

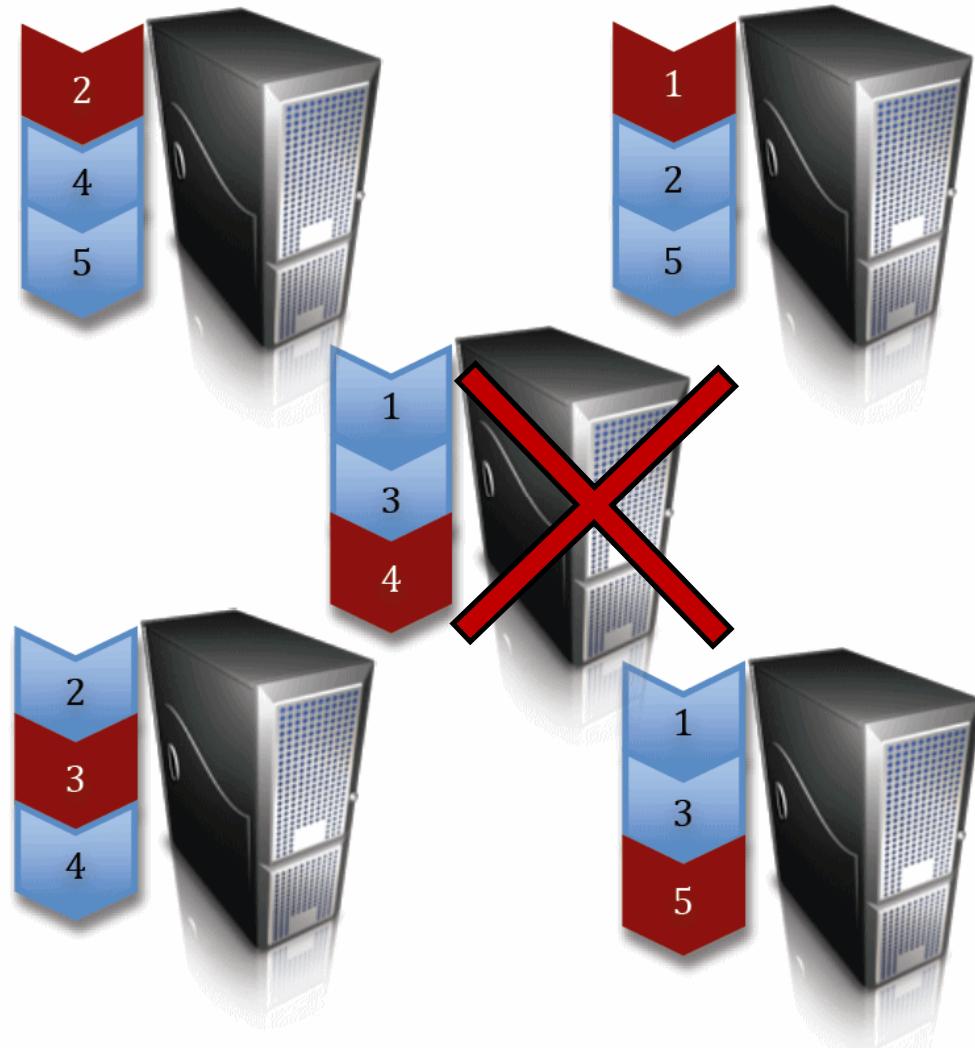


Cost/GB is a few  
¢/month vs \$/month

[apache.org/hadoop/](http://apache.org/hadoop/)

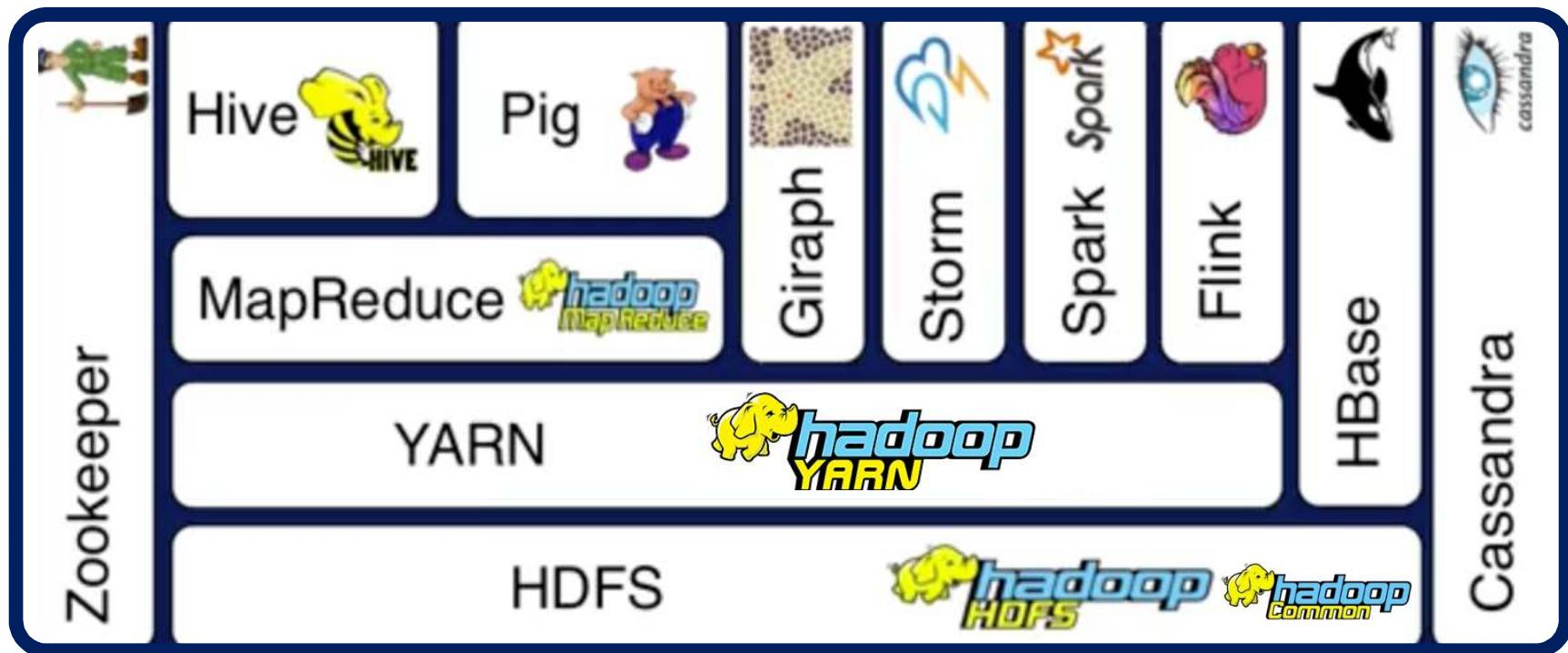
# MapReduce: Distributed Processing

*Hadoop takes advantage of HDFS' data distribution strategy to push work out to many nodes in a cluster. This allows analyses to run in parallel and eliminates the bottlenecks imposed by monolithic storage systems.*



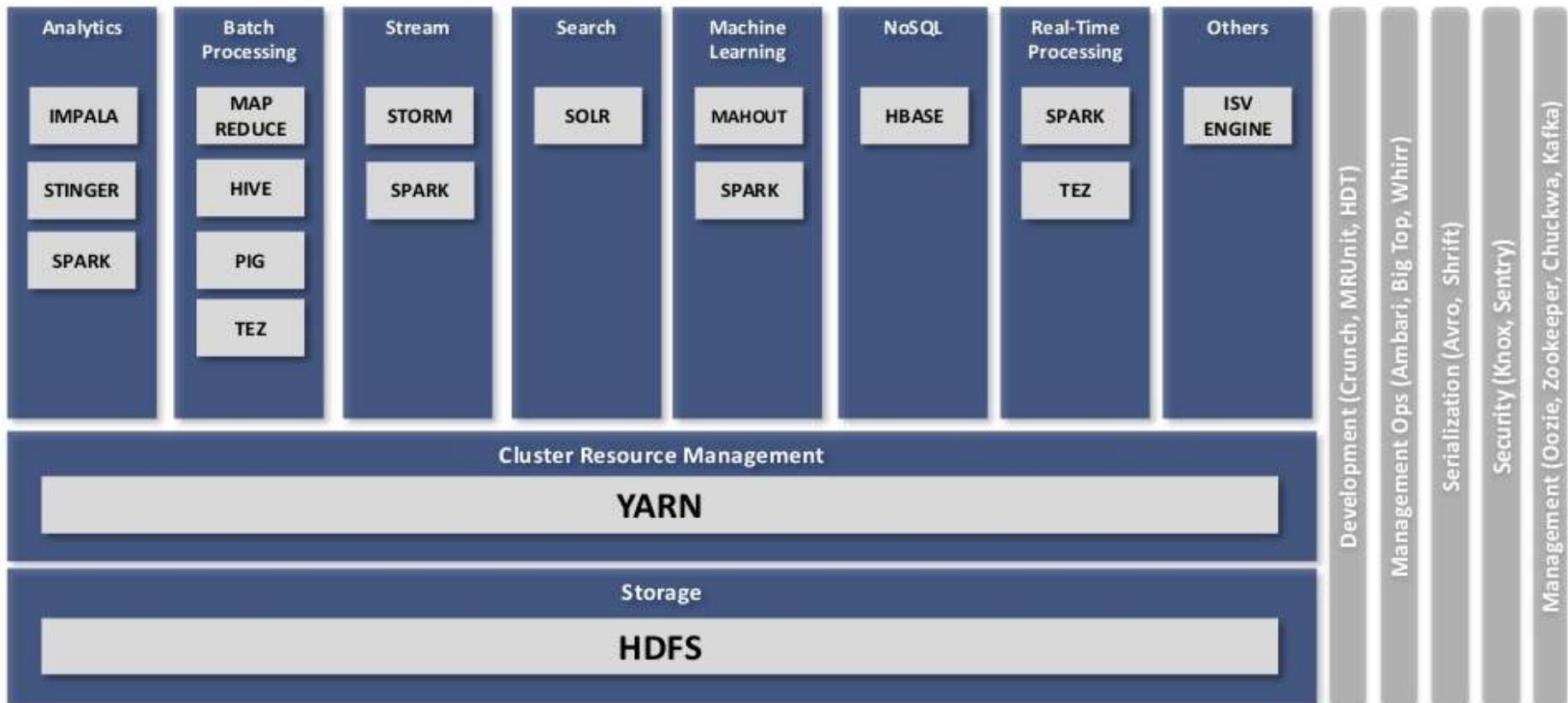
[apache.org/hadoop/](http://apache.org/hadoop/)

# Hadoop 2 Ecosystem



<http://3v4n4r34l.blogspot.com/2016/04/coursera-ucsd-big-data-specialization.html>

# Hadoop 2 Ecosystem

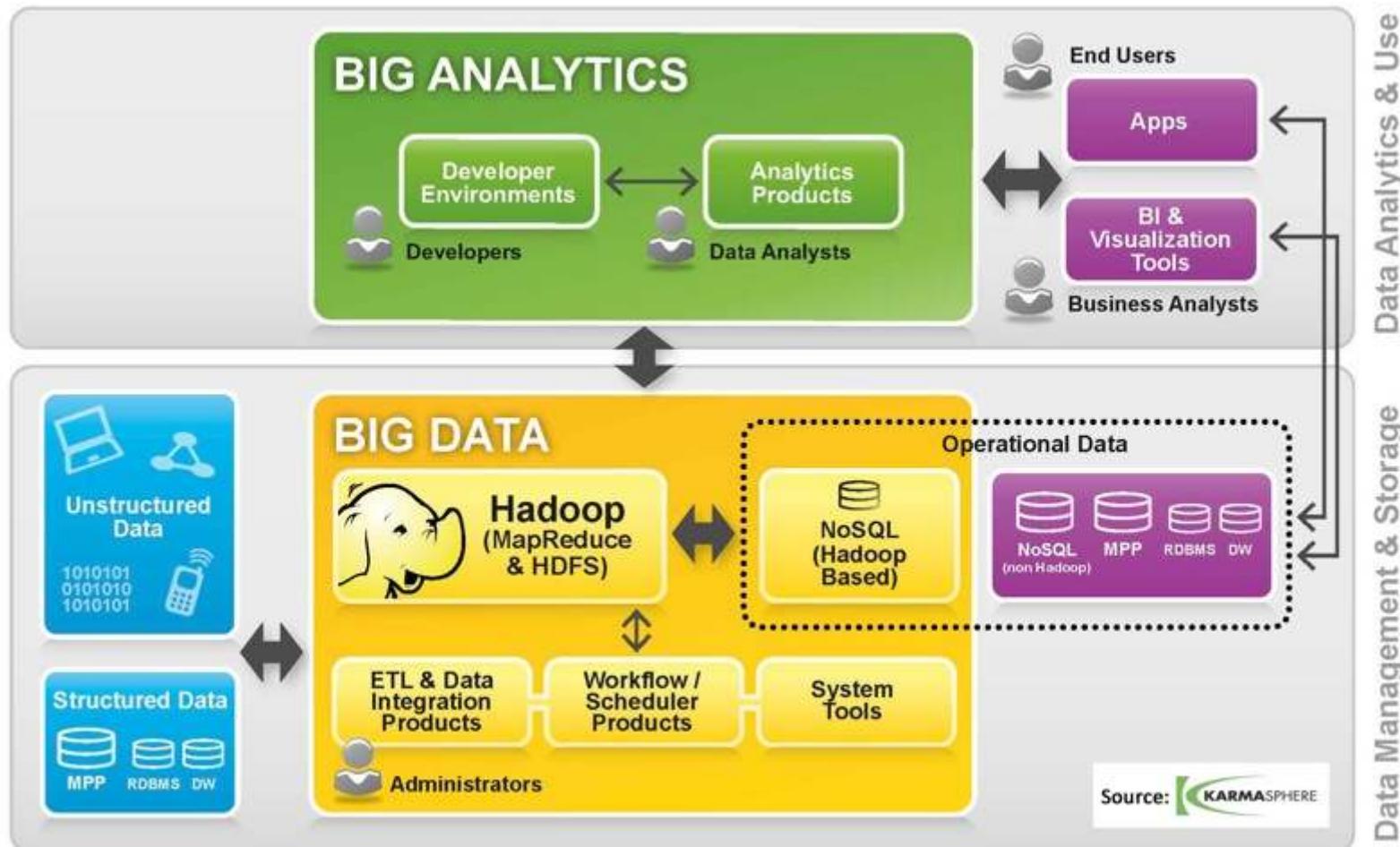


# The Evolving Hadoop Landscape

Components		Description	Components		Description
		Data Mining/machine learning tools used against Hadoop data to detect patterns and trends		BigTop	Packaging services for Hadoop projects to ease testing and deployment
	Pig	Scripting language for analyzing large datasets. Compiles to MapReduce jobs		Hbase	A non-relational, distributed database that runs on top of HDFS
MapReduce (Hadoop 1.0)	YARN (Hadoop 2.0)	Programming model for processing large data sets. YARN performs overall resource mgmt			Schema-based data serialization system using RPC calls
		A workflow scheduler tool to manage Hadoop MapReduce jobs			Indexing and search tools for data stored in HDFS for Hadoop
		Enable SQL for Hadoop data: Sqoop - Data transfer between Hadoop and structured datastores. HIVE - data warehouse for Hadoop. Drill - open source, low latency SQL query engine for Hadoop and NoSQL.			Collect, aggregate, and move streaming data from multiple sources into Hadoop
					AppDev tool for Hadoop apps combining batch, streaming, and interactive analytics
	ZooKeeper	Coordination of config. data, naming and synchronization of Hadoop projects			Monitoring & Management of Hadoop clusters and nodes

# Big Data Platform & Big Data Analytics

## Hadoop Technology



# Hands On Labs

## Installing Apache Hadoop 2.7.2

อ.ดนัยรัตน์ ธนบดีธรรมจารี

Line ID: Danairat

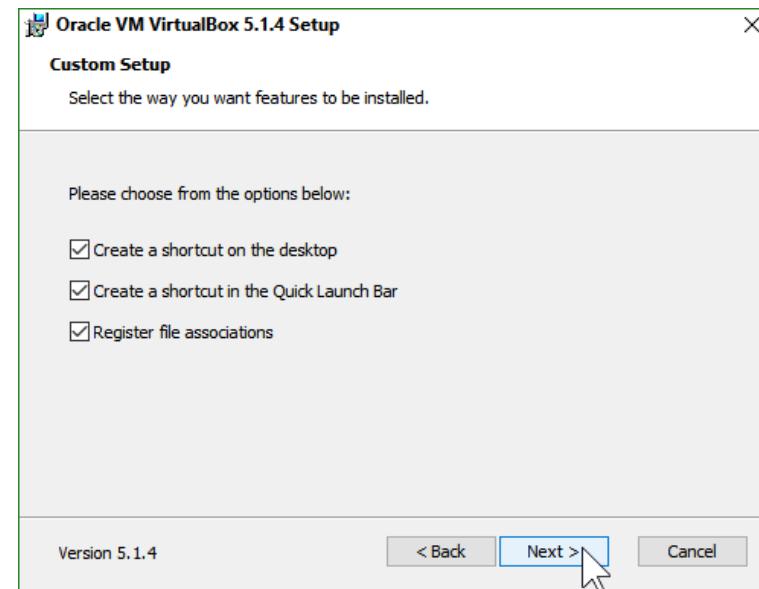
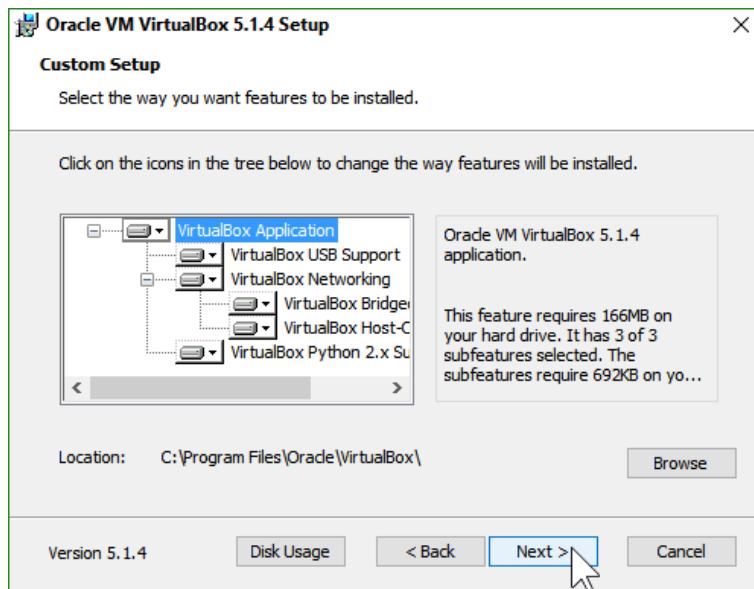
FB: Danairat Thanabodithammachari

+668-1559-1446

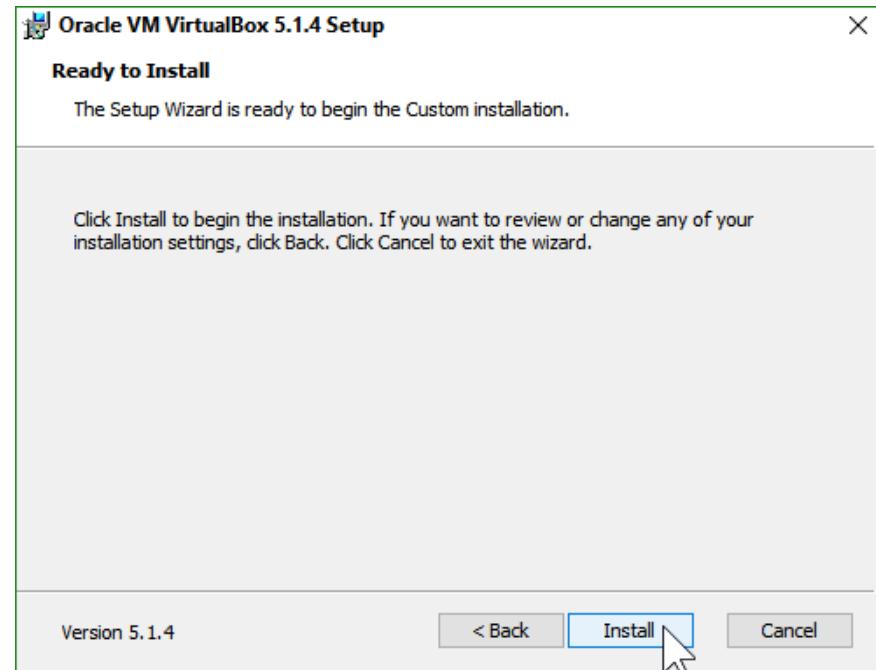
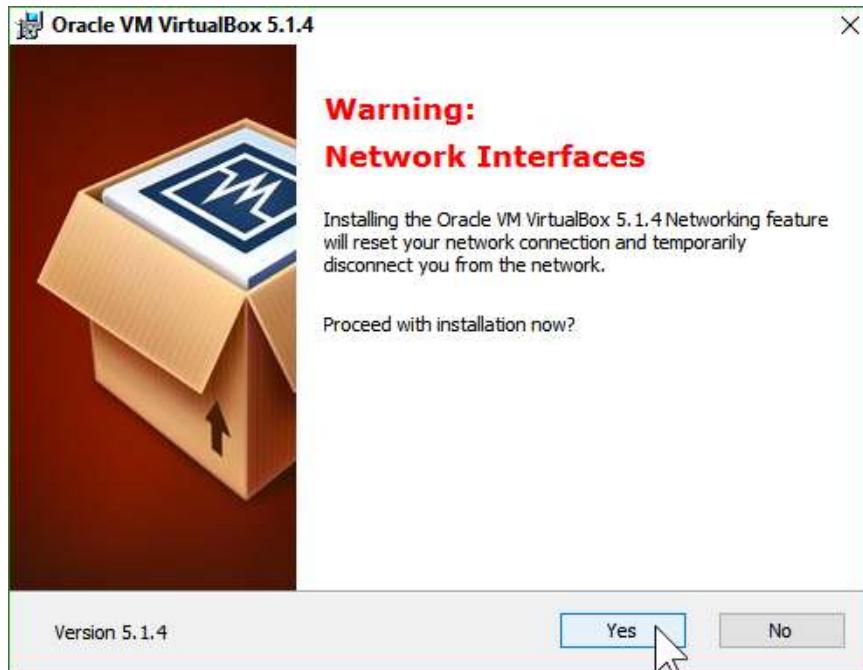
# 1. Install Virtual Box



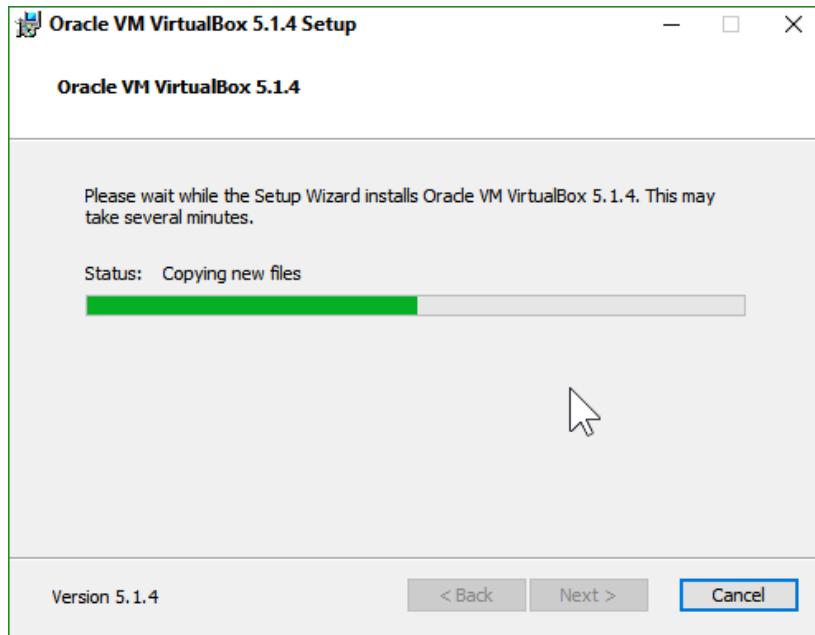
# 1. Install Virtual Box



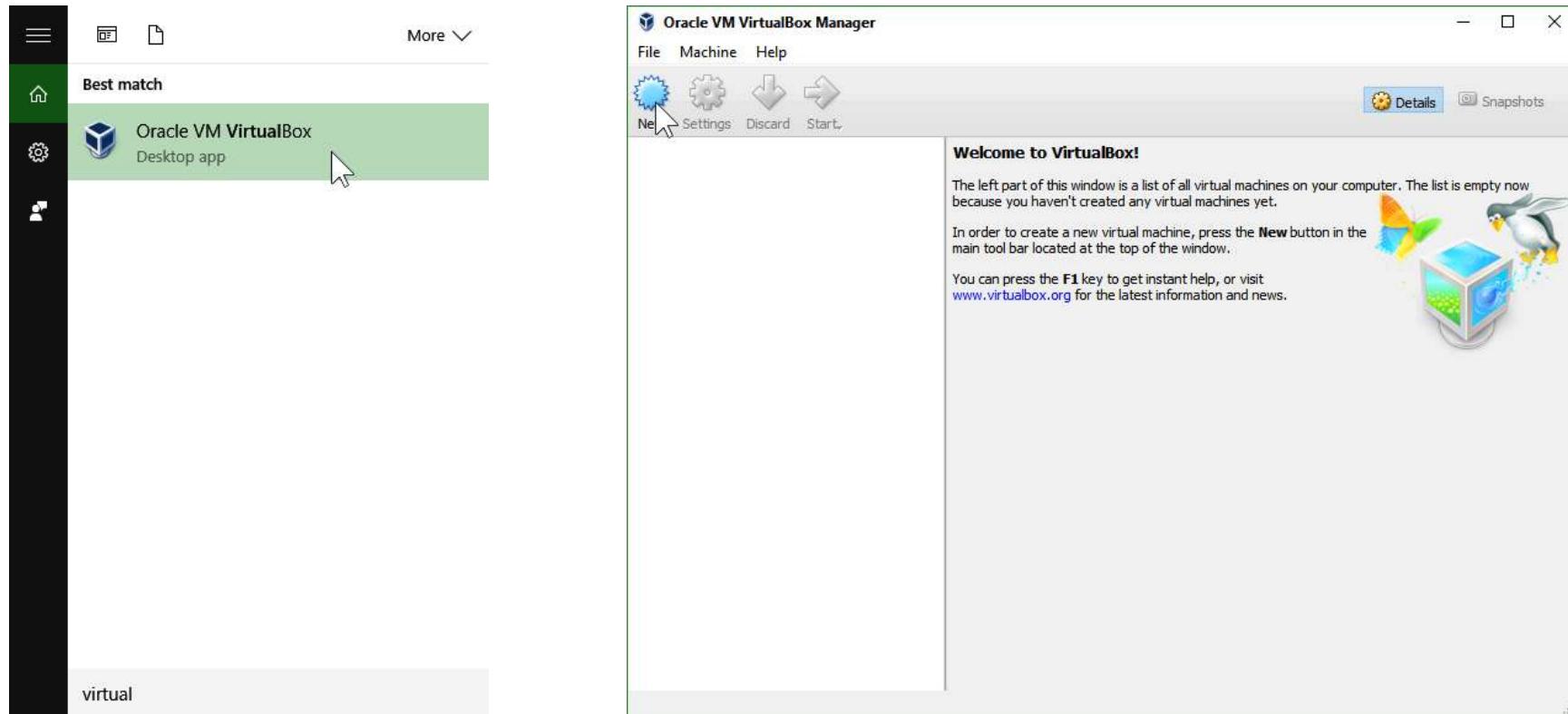
# 1. Install Virtual Box



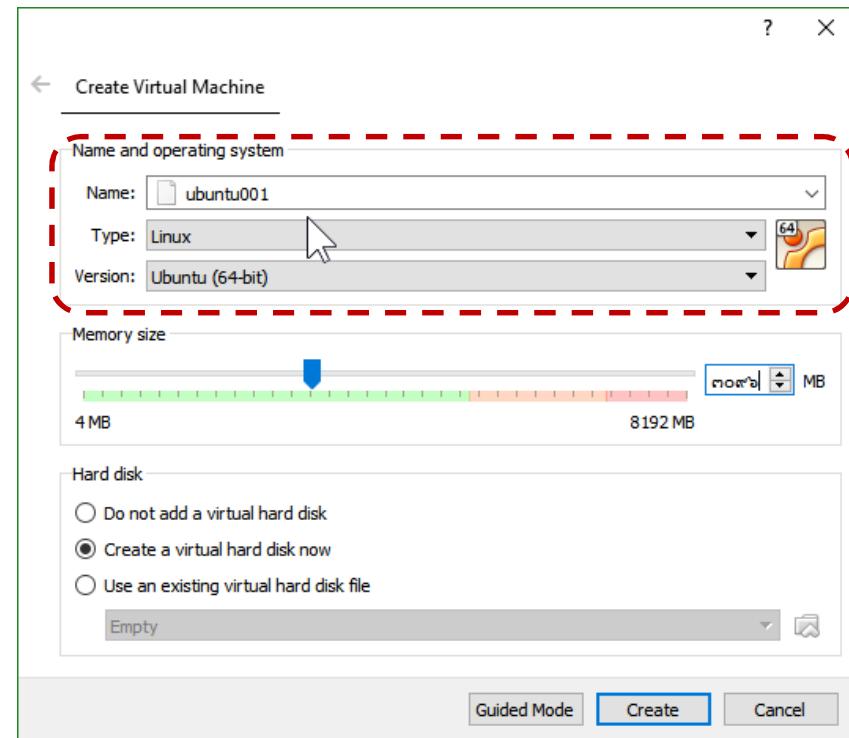
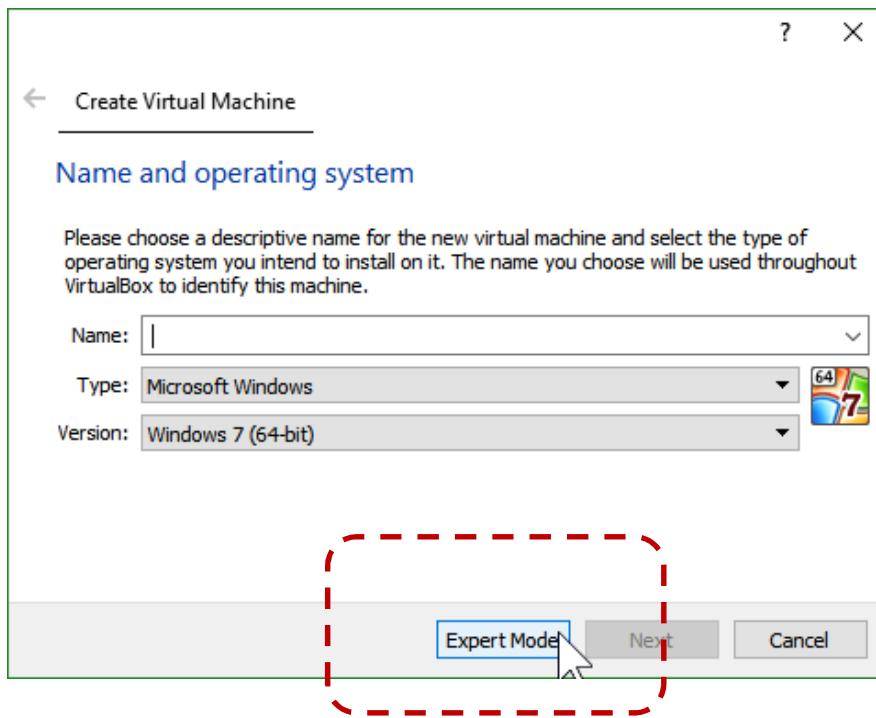
# 1. Install Virtual Box



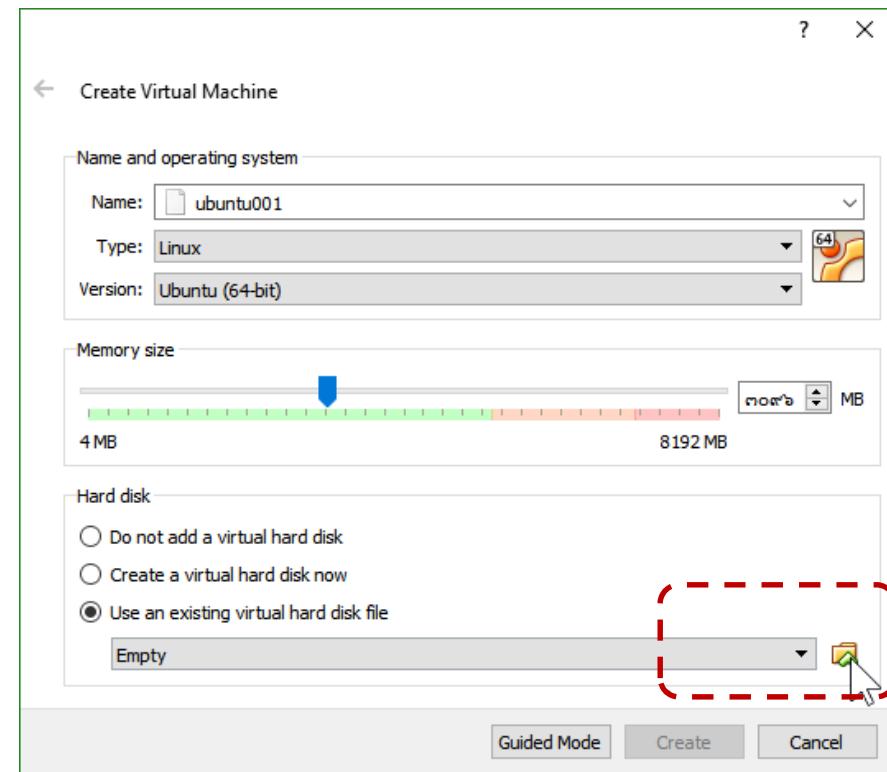
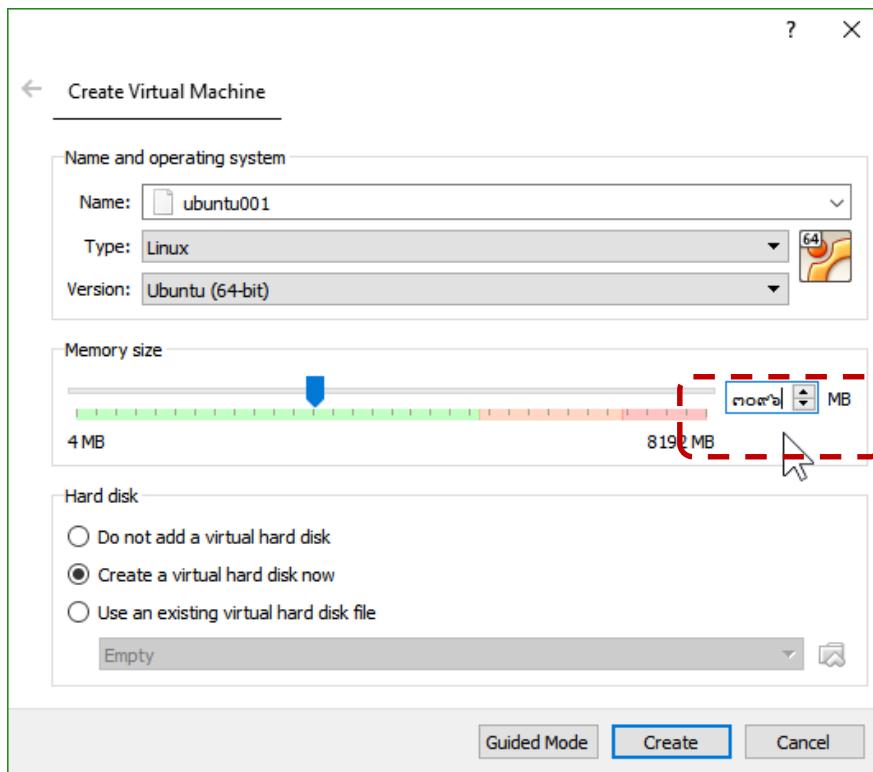
## 2. Start Ubuntu Linux Server



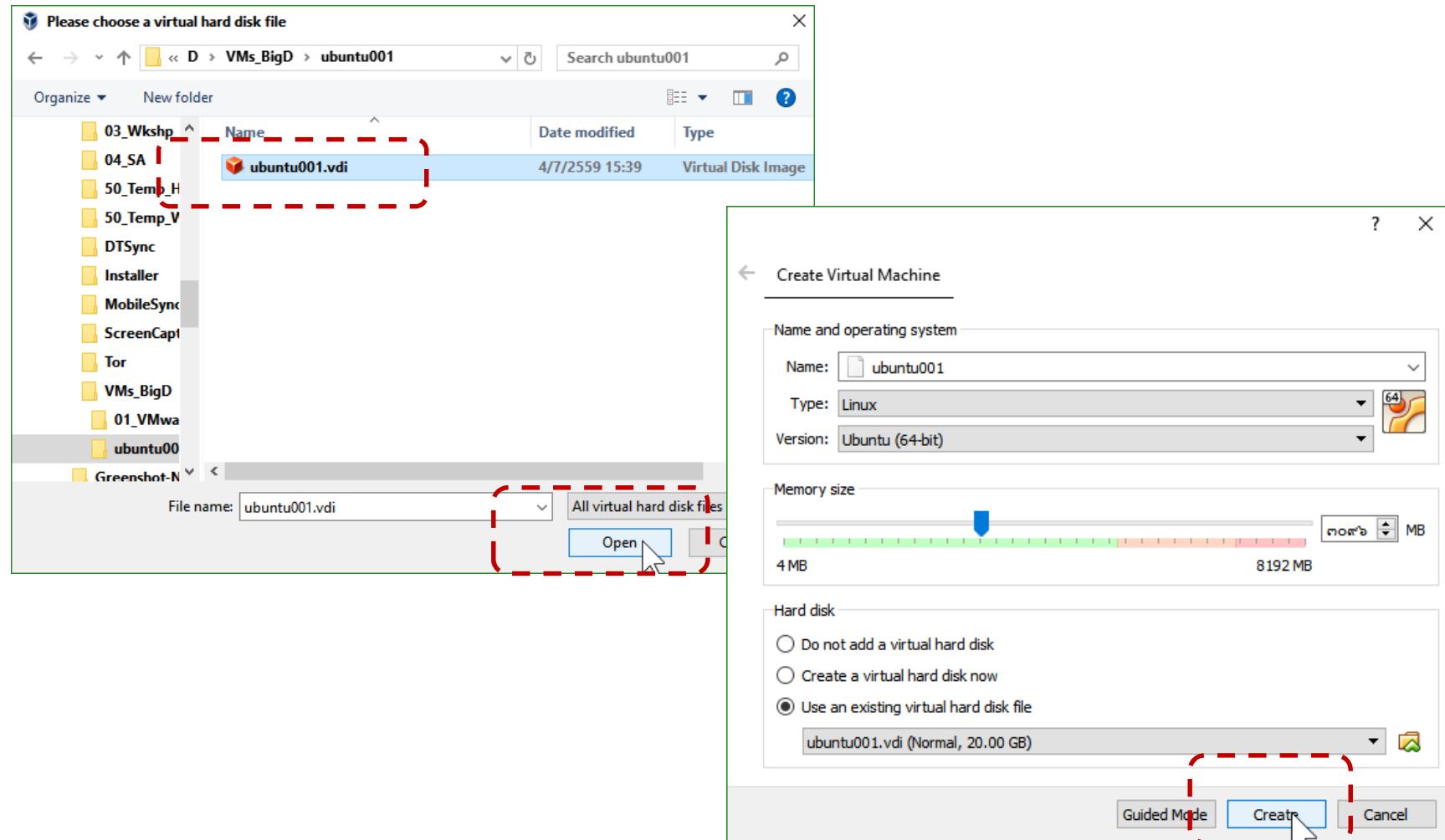
## 2. Start Ubuntu Linux Server



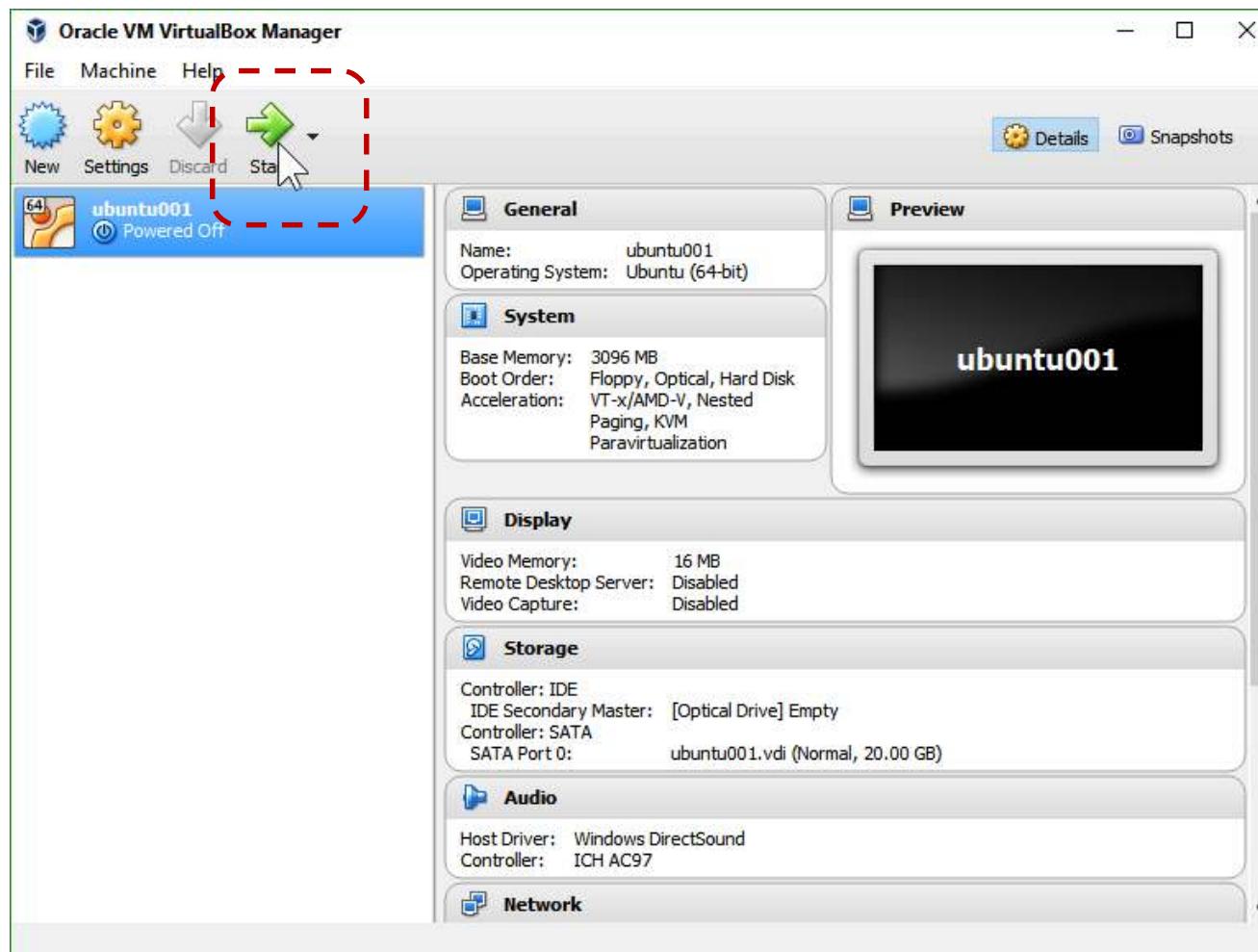
## 2. Start Ubuntu Linux Server



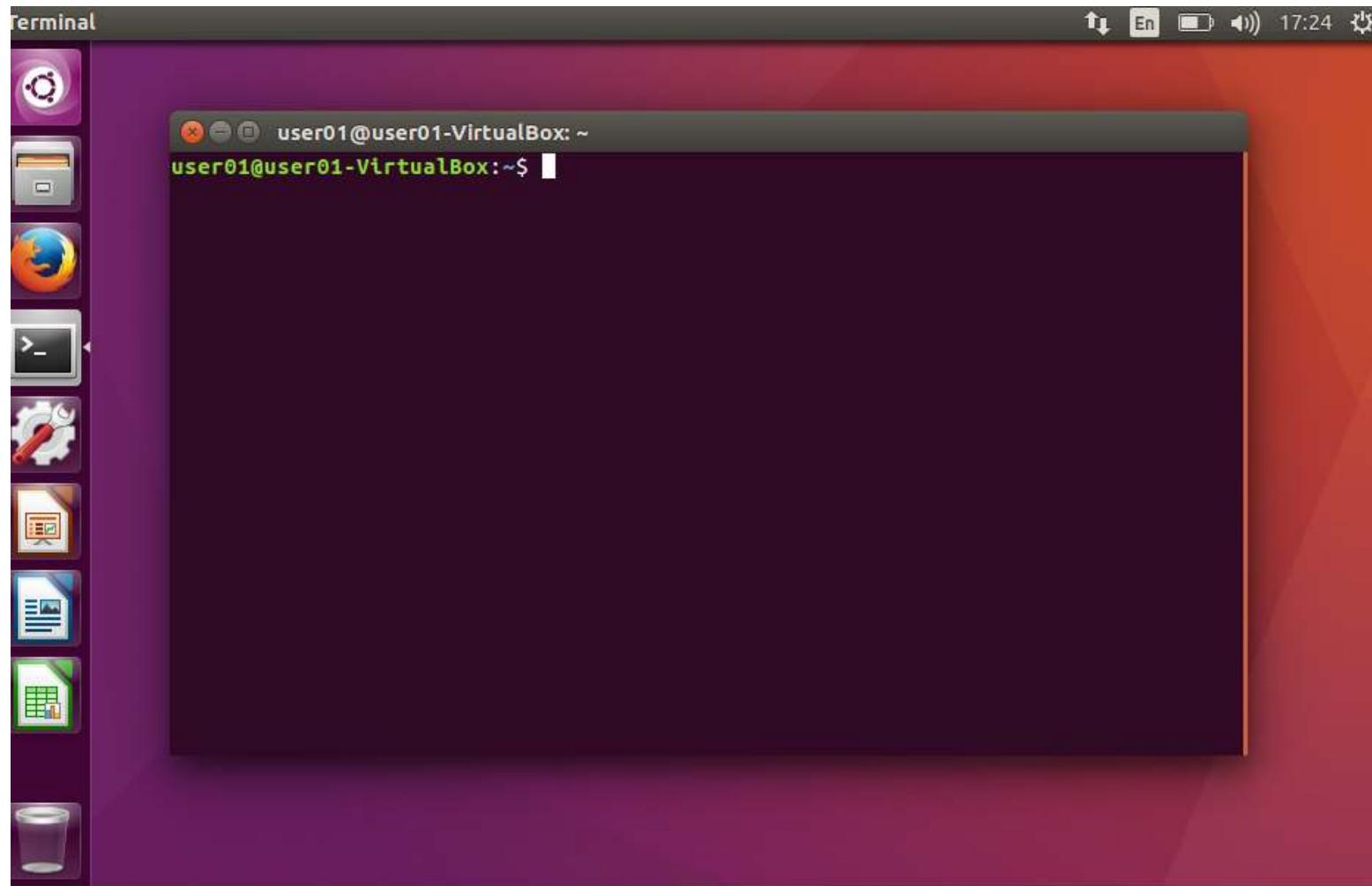
## 2. Start Ubuntu Linux Server



## 2. Start Ubuntu Linux Server

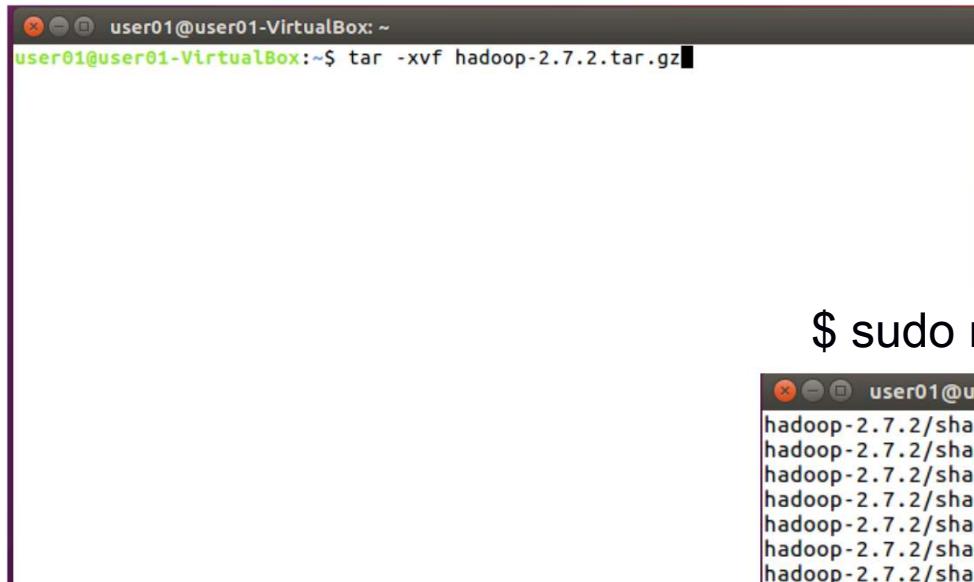


# 2. Start Ubuntu Linux Server



### 3. Setup Apache Hadoop 2.7.2

```
$ cd  
$ tar -xvf hadoop-2.7.2.tar.gz
```

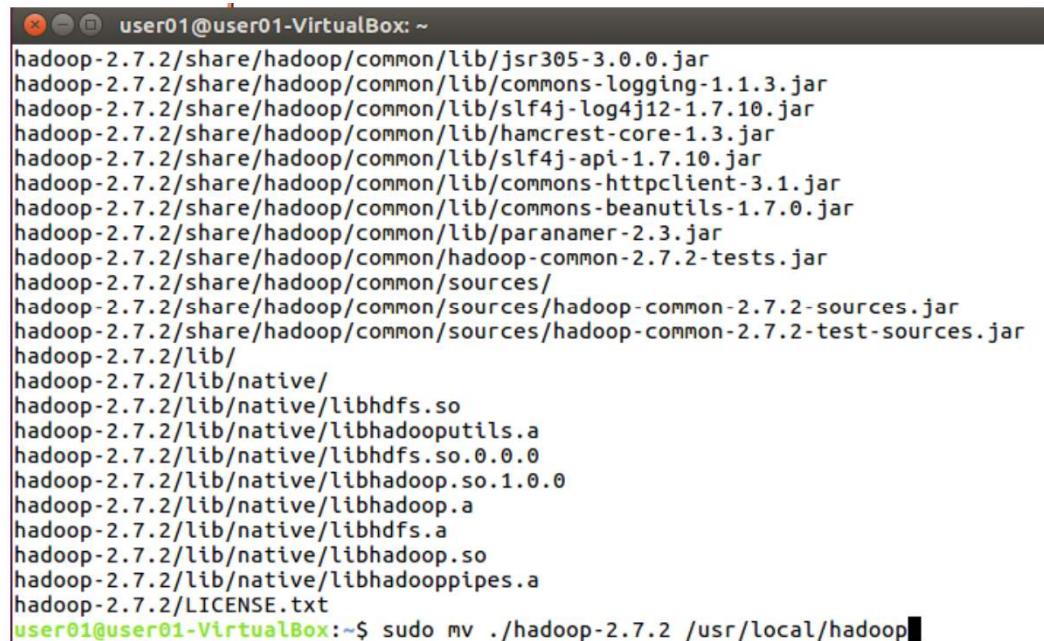


```
user01@user01-VirtualBox: ~  
user01@user01-VirtualBox:~$ tar -xvf hadoop-2.7.2.tar.gz
```

มีจุดที่นี่

มีเว้นวรคที่นี่

```
$ sudo mv ./hadoop-2.7.2 /usr/local/hadoop
```



```
user01@user01-VirtualBox: ~  
hadoop-2.7.2/share/hadoop/common/lib/jsr305-3.0.0.jar  
hadoop-2.7.2/share/hadoop/common/lib/commons-logging-1.1.3.jar  
hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar  
hadoop-2.7.2/share/hadoop/common/lib/hamcrest-core-1.3.jar  
hadoop-2.7.2/share/hadoop/common/lib/slf4j-api-1.7.10.jar  
hadoop-2.7.2/share/hadoop/common/lib/commons-httpclient-3.1.jar  
hadoop-2.7.2/share/hadoop/common/lib/commons-beanutils-1.7.0.jar  
hadoop-2.7.2/share/hadoop/common/lib/paranamer-2.3.jar  
hadoop-2.7.2/share/hadoop/common/hadoop-common-2.7.2-tests.jar  
hadoop-2.7.2/share/hadoop/common/sources/  
hadoop-2.7.2/share/hadoop/common/sources/hadoop-common-2.7.2-sources.jar  
hadoop-2.7.2/share/hadoop/common/sources/hadoop-common-2.7.2-test-sources.jar  
hadoop-2.7.2/lib/  
hadoop-2.7.2/lib/native/  
hadoop-2.7.2/lib/native/libhdfs.so  
hadoop-2.7.2/lib/native/libhadooputils.a  
hadoop-2.7.2/lib/native/libhdfs.so.0.0.0  
hadoop-2.7.2/lib/native/libhadoop.so.1.0.0  
hadoop-2.7.2/lib/native/libhadoop.a  
hadoop-2.7.2/lib/native/libhdfs.a  
hadoop-2.7.2/lib/native/libhadoop.so  
hadoop-2.7.2/lib/native/libhadooppipes.a  
hadoop-2.7.2/LICENSE.txt  
user01@user01-VirtualBox:~$ sudo mv ./hadoop-2.7.2 /usr/local/hadoop
```

### 3. Setup Apache Hadoop 2.7.2

```
$ sudo nano ~/.bashrc
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/  
export PATH=$PATH:$JAVA_HOME/bin  
  
export HADOOP_HOME=/usr/local/hadoop  
export PATH=$PATH:$HADOOP_HOME/bin  
export PATH=$PATH:$HADOOP_HOME/sbin
```

Ctrl + x and save file to save with exit the editor

# 1. Setup Apache Hadoop 2.7.2

```
$ source ~/.bashrc  
  
$ sudo mkdir /var/log/hadoop  
  
$ sudo chown -R user01:user01 /var/log/hadoop  
  
$ cd /usr/local/hadoop/etc/hadoop  
  
$ nano hadoop-env.sh
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop
GNU nano 2.5.3          File: hadoop-env.sh          Modified

# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

#-----# The java implementation to use.-----#
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

#-----# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol. Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements. Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Linter ^L Go To Line
```

# 3. Setup Apache Hadoop 2.7.2

```
$ nano yarn-env.sh
```

```
YARN_LOG_DIR=/var/log/hadoop
```

```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop
GNU nano 2.5.3          File: yarn-env.sh          Modified

# Specify the JVM options to be used when starting the NodeManager.
# These options will be appended to the options specified as YARN_OPTS
# and therefore may override any similar flags set in YARN_OPTS
#export YARN_NODEMANAGER_OPTS=

# so that filenames w/ spaces are handled correctly in loops below
IFS=


# default log directory & file
if [ "$YARN_LOG_DIR" = "" ]; then
    YARN_LOG_DIR=/var/log/hadoop
fi
if [ "$YARN_LOGFILE" = "" ]; then
    YARN_LOGFILE='yarn.log'
fi

# default policy file for service-level authorization
if [ "$YARN_POLICYFILE" = "" ]; then

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit     ^R Read File  ^L Replace   ^U Uncut Text  ^T To Upper  ^A Go To Line
```

# 3. Setup Apache Hadoop 2.7.2

```
$ ifconfig
```

```
user01@user01-VirtualBox: /usr/local/hadoop
user01@user01-VirtualBox: /usr/local/hadoop$ ifconfig
enp0s3    Link encap:Ethernet HWaddr 08:00:27:ec:9c:f2
           inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
             inet6 addr: fe80::280:3162:b5b8:edb2/64 Scope:Link
               UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
               RX packets:32 errors:0 dropped:0 overruns:0 frame:0
               TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
               collisions:0 txqueuelen:1000
               RX bytes:8557 (8.5 KB) TX bytes:16385 (16.3 KB)

lo        Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
             inet6 addr: ::1/128 Scope:Host
               UP LOOPBACK RUNNING MTU:65536 Metric:1
               RX packets:58 errors:0 dropped:0 overruns:0 frame:0
               TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
               collisions:0 txqueuelen:1
               RX bytes:4320 (4.3 KB) TX bytes:4320 (4.3 KB)

user01@user01-VirtualBox: /usr/local/hadoop$
```

วิธีดู IP Address ของเครื่องเรา

### 3. Setup Apache Hadoop 2.7.2

```
$ nano core-site.xml
```

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://10.0.2.15:9000</value>
  </property>
</configuration>
```

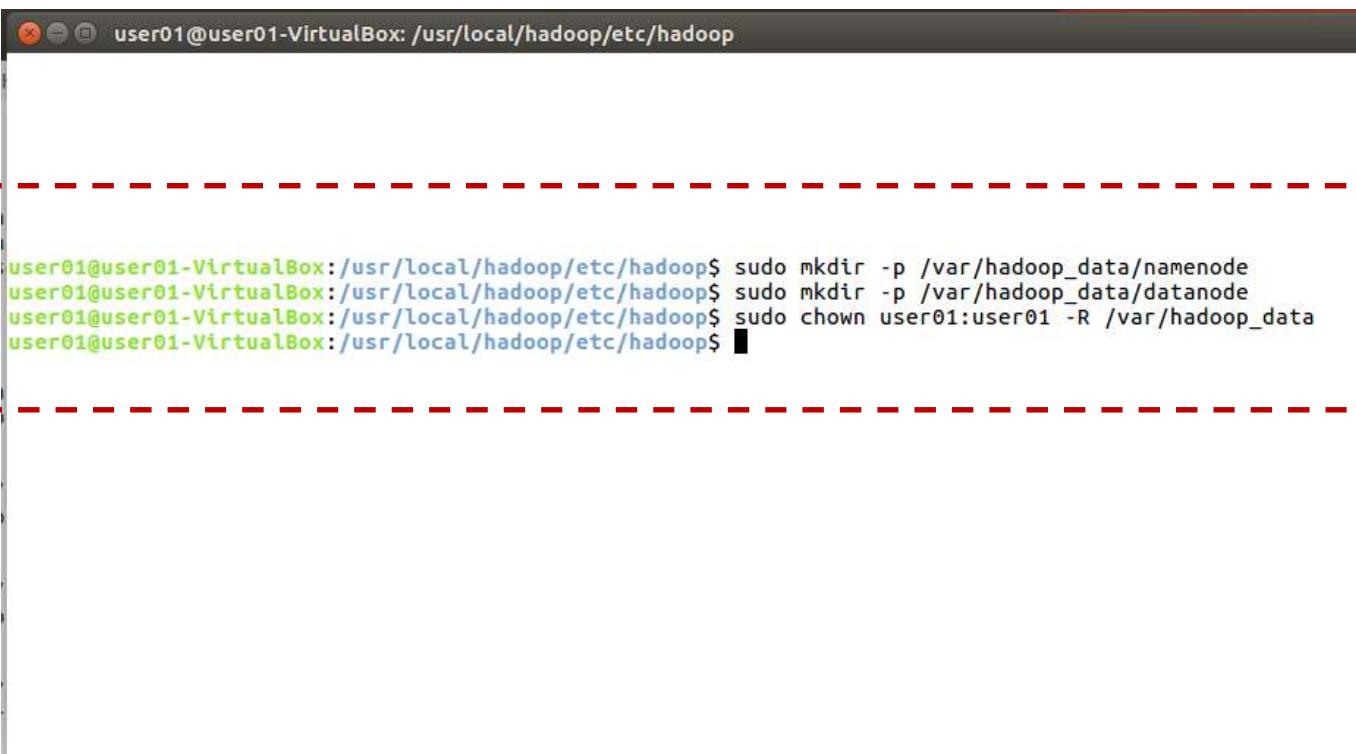
```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop
GNU nano 2.5.3          File: core-site.xml          Modified
```

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://10.0.2.15:9000</value>
  </property>
</configuration>
```

```
G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
X Exit      ^R Read File  ^I Replace   ^U Uncut Text  ^T To Spell  ^L Go To Line
```

### 3. Setup Apache Hadoop 2.7.2

```
$ sudo mkdir -p /var/hadoop_data/namenode  
$ sudo mkdir -p /var/hadoop_data/datanode  
$ sudo chown user01:user01 -R /var/hadoop_data
```



The screenshot shows a terminal window titled "user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop". The window contains a red dashed rectangular box highlighting the command history. The commands listed are:

```
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo mkdir -p /var/hadoop_data/namenode  
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo mkdir -p /var/hadoop_data/datanode  
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo chown user01:user01 -R /var/hadoop_data  
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ █
```

### 3. Setup Apache Hadoop 2.7.2

```
$ nano hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/var/hadoop_data/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/var/hadoop_data/datanode</value>
  </property>
</configuration>
```

# 3. Setup Apache Hadoop 2.7.2

\$ nano yarn-site.xml

```
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>10.0.2.15</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>10.0.2.15:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>10.0.2.15:8031</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>10.0.2.15:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>10.0.2.15:8033</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>10.0.2.15:8088</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

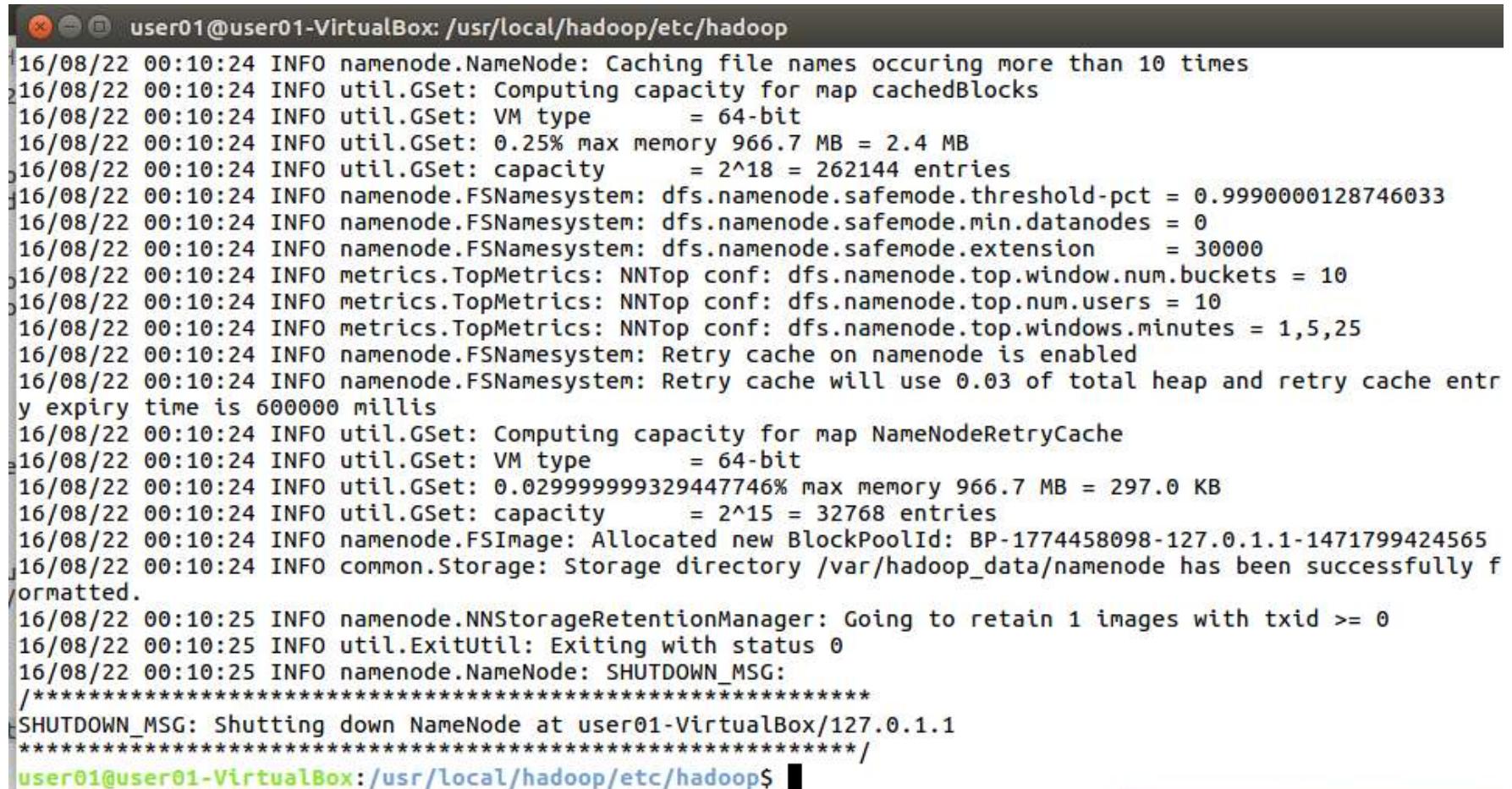
### 3. Setup Apache Hadoop 2.7.2

```
$ cp mapred-site.xml.template mapred-site.xml  
$ nano mapred-site.xml
```

```
<configuration>  
  <property>  
    <name>mapreduce.framework.name</name>  
    <value>yarn</value>  
  </property>  
</configuration>
```

# 3. Setup Apache Hadoop 2.7.2

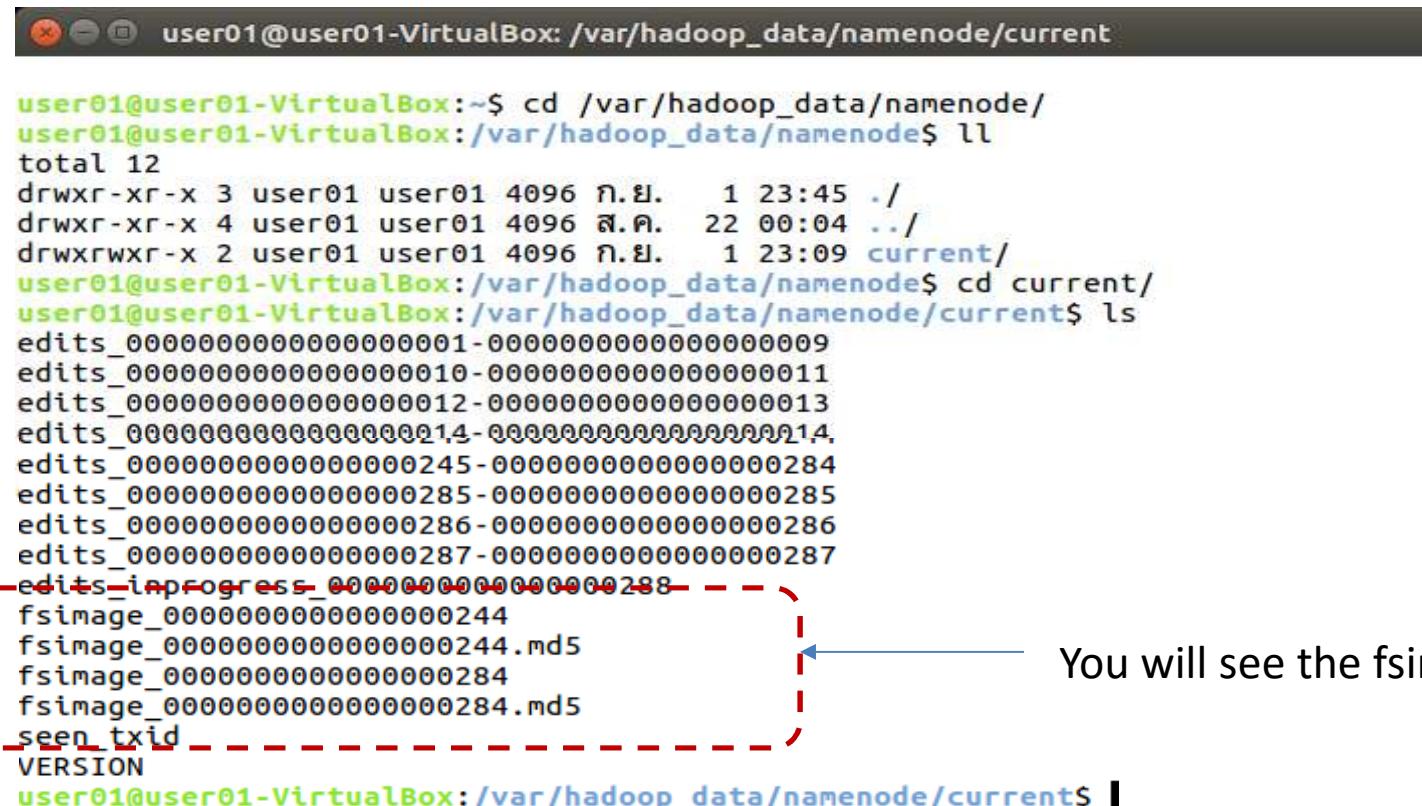
```
$ hdfs namenode -format
```



```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop
16/08/22 00:10:24 INFO namenode.NameNode: Caching file names occurring more than 10 times
16/08/22 00:10:24 INFO util.GSet: Computing capacity for map cachedBlocks
16/08/22 00:10:24 INFO util.GSet: VM type      = 64-bit
16/08/22 00:10:24 INFO util.GSet: 0.25% max memory 966.7 MB = 2.4 MB
16/08/22 00:10:24 INFO util.GSet: capacity      = 2^18 = 262144 entries
16/08/22 00:10:24 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
16/08/22 00:10:24 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
16/08/22 00:10:24 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension      = 30000
16/08/22 00:10:24 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
16/08/22 00:10:24 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
16/08/22 00:10:24 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
16/08/22 00:10:24 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
16/08/22 00:10:24 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
16/08/22 00:10:24 INFO util.GSet: Computing capacity for map NameNodeRetryCache
16/08/22 00:10:24 INFO util.GSet: VM type      = 64-bit
16/08/22 00:10:24 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 297.0 KB
16/08/22 00:10:24 INFO util.GSet: capacity      = 2^15 = 32768 entries
16/08/22 00:10:24 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1774458098-127.0.1.1-1471799424565
16/08/22 00:10:24 INFO common.Storage: Storage directory /var/hadoop_data/namenode has been successfully formatted.
16/08/22 00:10:25 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
16/08/22 00:10:25 INFO util.ExitUtil: Exiting with status 0
16/08/22 00:10:25 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at user01-VirtualBox/127.0.1.1
*****user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$
```

# 3. Setup Apache Hadoop 2.7.2

```
$ cd /var/hadoop_data/namenode/  
$ ll  
$ cd current/  
$ ls
```



```
user01@user01-VirtualBox:~$ cd /var/hadoop_data/namenode/  
user01@user01-VirtualBox:/var/hadoop_data/namenode$ ll  
total 12  
drwxr-xr-x 3 user01 user01 4096 ก.ย. 1 23:45 ./  
drwxr-xr-x 4 user01 user01 4096 ส.ค. 22 00:04 ../  
drwxrwxr-x 2 user01 user01 4096 ก.ย. 1 23:09 current/  
user01@user01-VirtualBox:/var/hadoop_data/namenode$ cd current/  
user01@user01-VirtualBox:/var/hadoop_data/namenode/current$ ls  
edits_0000000000000001-0000000000000009  
edits_0000000000000010-0000000000000011  
edits_0000000000000012-0000000000000013  
edits_0000000000000014-0000000000000014.  
edits_0000000000000245-0000000000000284  
edits_0000000000000285-0000000000000285  
edits_0000000000000286-0000000000000286  
edits_0000000000000287-0000000000000287  
- edits_inprogress_0000000000000288 -  
fsimage_000000000000000244  
fsimage_000000000000000244.md5  
fsimage_000000000000000284  
fsimage_000000000000000284.md5  
seen_txid  
VERSION  
user01@user01-VirtualBox:/var/hadoop_data/namenode/current$ |
```

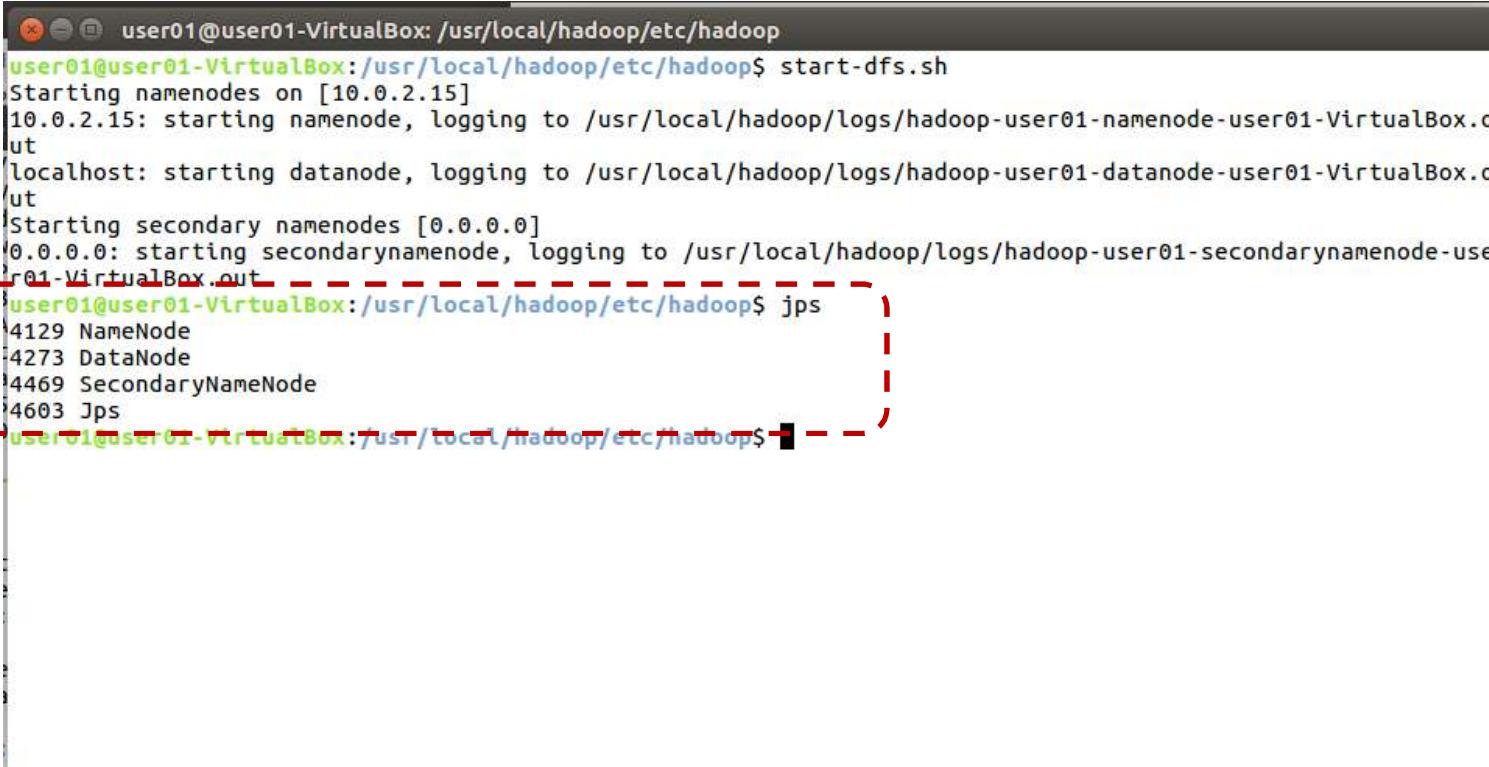
You will see the fsimage here

# 3. Setup Apache Hadoop 2.7.2

```
$ start-dfs.sh
```

ຕອນ yes

```
$ jps
```



```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [10.0.2.15]
10.0.2.15: starting namenode, logging to /usr/local/hadoop/logs/hadoop-user01-namenode-user01-VirtualBox.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-user01-datanode-user01-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-user01-secondarynamenode-user01-VirtualBox.out
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ jps
4129 NameNode
4273 DataNode
4469 SecondaryNameNode
4603 Jps
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$
```

# 3. Setup Apache Hadoop 2.7.2

```
$ start-yarn.sh
```

```
$ jps
```

```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [10.0.2.15]
10.0.2.15: starting namenode, logging to /usr/local/hadoop/logs/hadoop-user01-namenode-user01-VirtualBox.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-user01-datanode-user01-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-user01-secondarynamenode-user01-VirtualBox.out
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ jps
4129 NameNode
4273 DataNode
4469 SecondaryNameNode
4603 Jps
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /var/log/hadoop/yarn-user01-resourcemanager-user01-VirtualBox.out
localhost: starting nodemanager, logging to /var/log/hadoop/yarn-user01-nodemanager-user01-VirtualBox.out
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ jps
4129 NameNode
4273 DataNode
5073 Jps
4469 SecondaryNameNode
4665 ResourceManager
4783 NodeManager
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$
```

# 3. Setup Apache Hadoop 2.7.2

URL: <http://127.0.0.1:50070>

The screenshot shows a web browser window titled "Namenode information". The address bar displays the URL "127.0.0.1:50070/dfshealth.html#tab-overview". The top navigation bar includes links for "Hadoop", "Overview", "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities". Below the navigation bar, the main content area is titled "Overview '10.0.2.15:9000' (active)". A table provides detailed cluster information:

<b>Started:</b>	Mon Aug 22 00:16:00 ICT 2016
<b>Version:</b>	2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41
<b>Compiled:</b>	2016-01-26T00:08Z by jenkins from (detached from b165c4f)
<b>Cluster ID:</b>	CID-0c5235c0-9280-45f6-9f58-8fbab6ed8cd6
<b>Block Pool ID:</b>	BP-1774458098-127.0.1.1-1471799424565

Below the table, the "Summary" section contains the following status information:

- Security is off.
- Safemode is off.
- 1 files and directories. 0 blocks = 1 total filesystem object(s).



# Apache Hadoop HDFS

อ.ดนัยรัตน์ ธนาบดีธรรมจารี

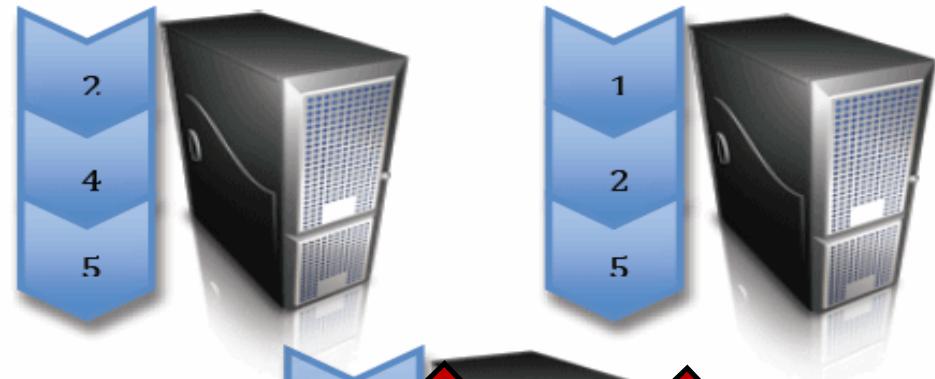
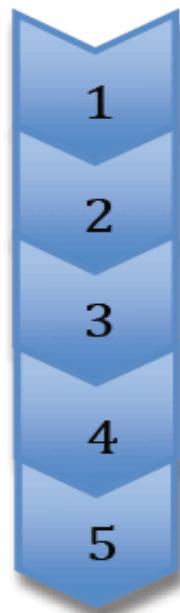
Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

# HDFS: Hadoop Distributed File System

Block Size =  
64MB/128MB/256MB  
Replication Factor = 3

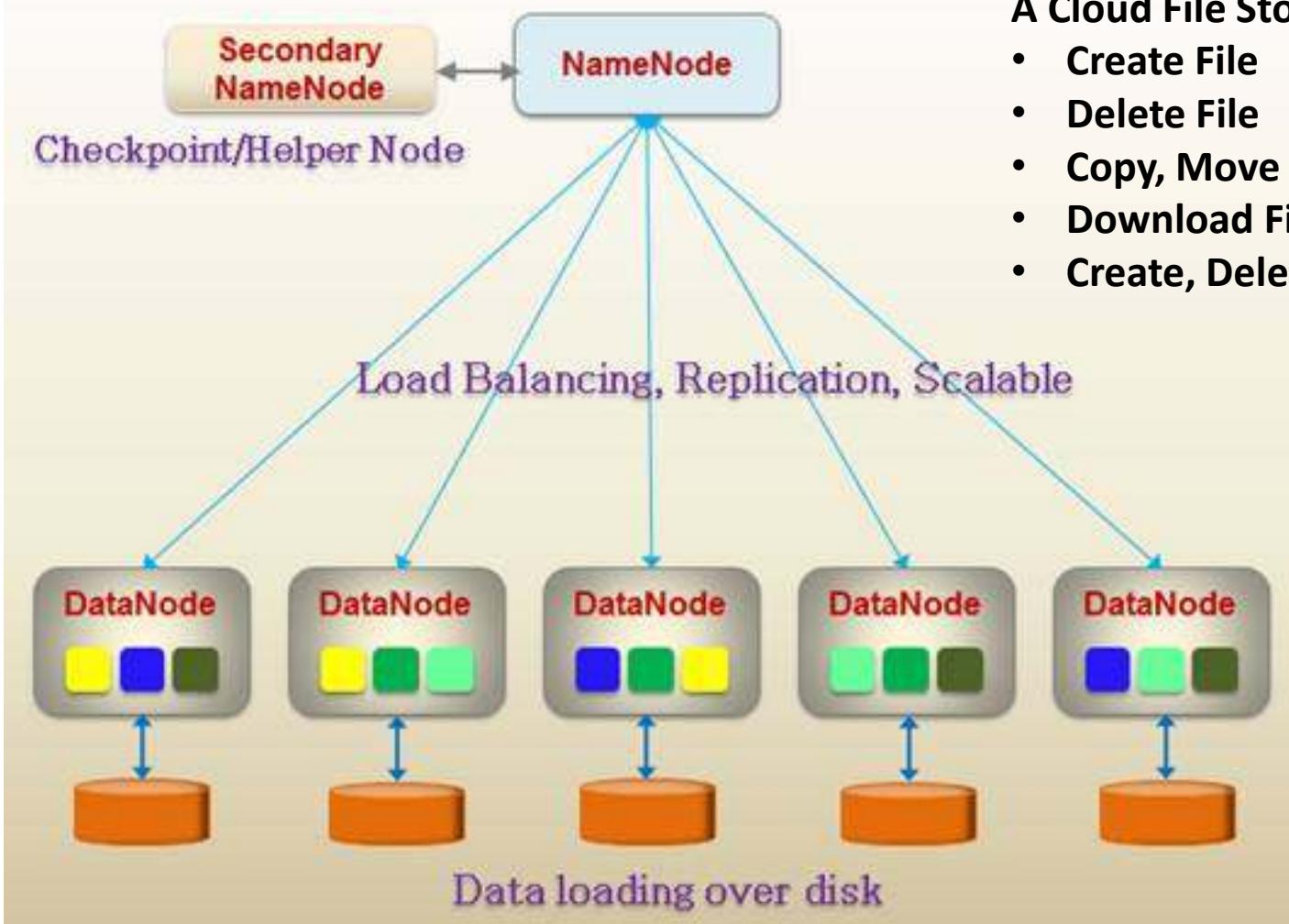


Cost/GB is a few  
¢/month vs \$/month

[apache.org/hadoop/](http://apache.org/hadoop/)

# HDFS

## HDFS Architecture



### A Cloud File Storage:-

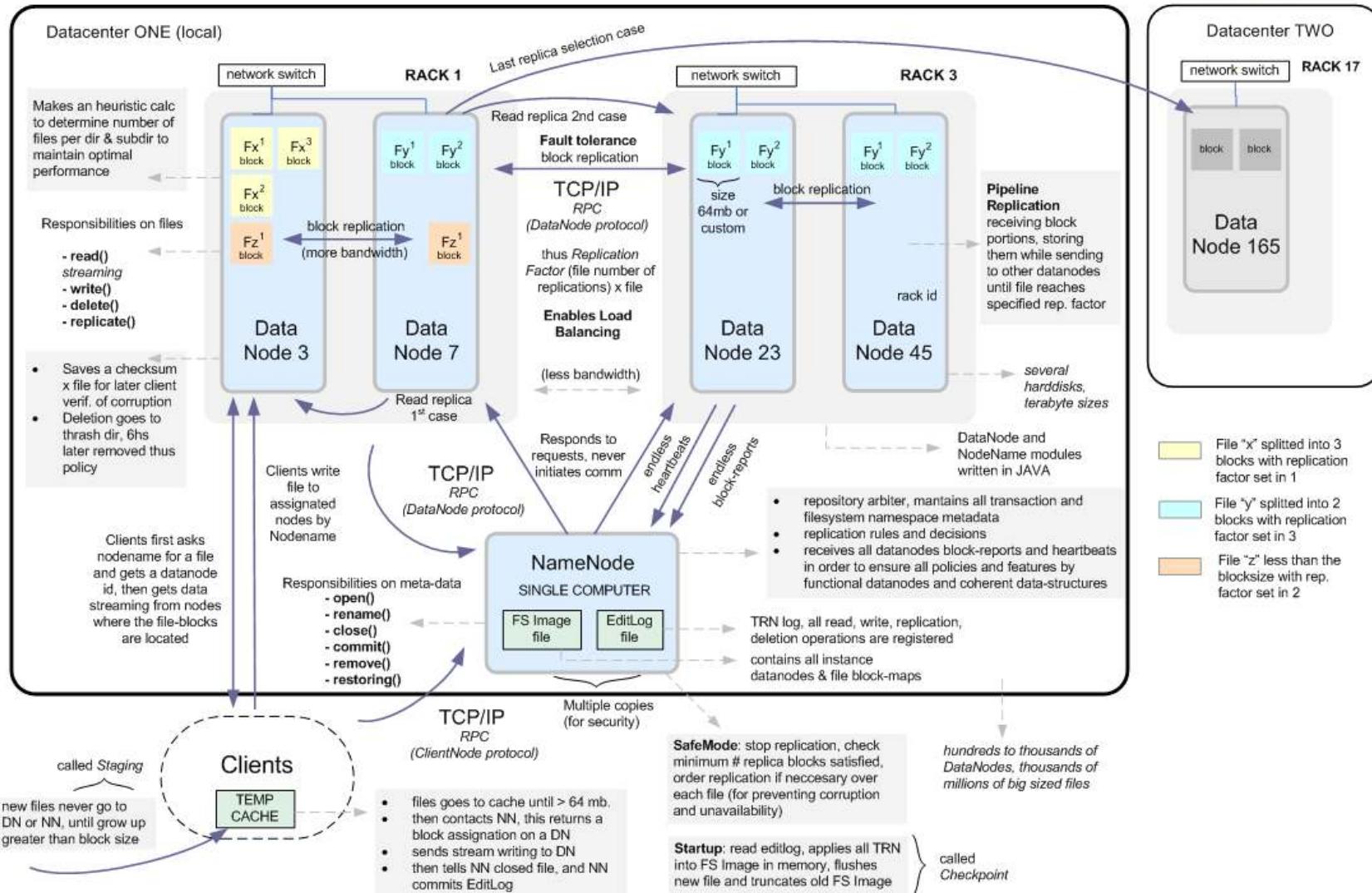
- Create File
- Delete File
- Copy, Move File
- Download File
- Create, Delete Directory

<http://www.developer.com/db/getting-familiarized-with-the-hadoop-distribution-file-system.html>

# Hadoop HDFS Architecture

## Hadoop Distributed Filesystem Architecture

For more information visit me at  
[www.Hadooping.wordpress.com](http://www.Hadooping.wordpress.com)



# 4. Hands on HDFS

```
$ hdfs dfs -mkdir /inputs  
$ hdfs dfs -mkdir /outputs
```

```
$ hdfs dfs -copyFromLocal ~/sales_data_jan2009.csv /inputs/
```

```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop
```

```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop$ hdfs dfs -copyFromLocal ~/sales_data_jan2009.csv /inputs/
```

```
$ hdfs dfs -ls /inputs
```

```
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop$ hdfs dfs -ls /inputs/  
Found 1 items  
-rw-r--r--    1 user01 supergroup      123637 2016-08-22 00:22 /inputs/sales_data_jan2009.csv  
user01@user01-VirtualBox: /usr/local/hadoop/etc/hadoop$
```

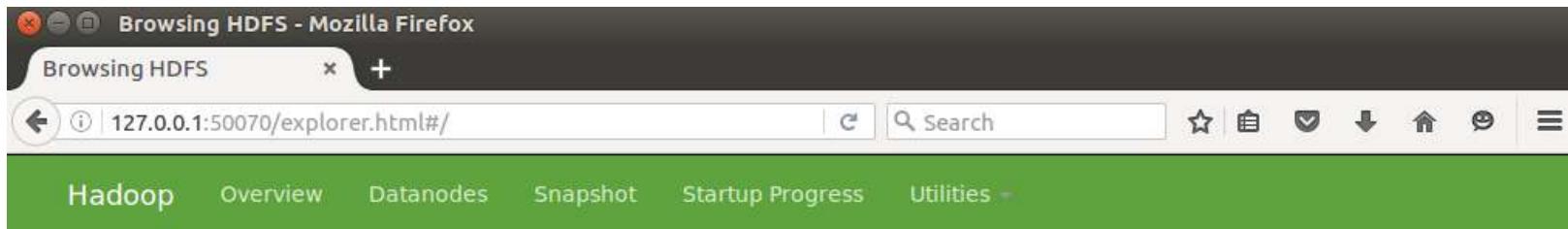
# 4. Hands on HDFS

The screenshot shows a Mozilla Firefox browser window titled "Namenode information - Mozilla Firefox". The address bar displays the URL "127.0.0.1:50070/dfshealth.html#tab-overview". The main content area is titled "Overview '10.0.2.15:9000' (active)". A red dashed circle highlights the "Utilities" dropdown menu in the top navigation bar. A tooltip from this menu points to "Browse the file system" and "Logs". Below the navigation bar, there is a table with the following data:

<b>Started:</b>	Mon Aug 22 00:16:00 ICT 2016
<b>Version:</b>	2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41
<b>Compiled:</b>	2016-01-26T00:08Z by jenkins from (detached from b165c4f)
<b>Cluster ID:</b>	CID-0c5235c0-9280-45f6-9f58-8fbab6ed8cd6
<b>Block Pool ID:</b>	BP-1774458098-127.0.1.1-1471799424565

At the bottom left of the content area, there is a link: "127.0.0.1:50070/explorer.html".

# 4. Hands on HDFS



## Browse Directory

Browse Directory							Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	user01	supergroup	0 B	22/8/2559 00:22:01	0	0 B	<a href="#">inputs</a>
drwxr-xr-x	user01	supergroup	0 B	22/8/2559 00:19:39	0	0 B	<a href="#">outputs</a>

Hadoop, 2015.

# 4. Hands on HDFS

Browsing HDFS - Mozilla Firefox

Browsing HDFS

127.0.0.1:50070/explorer.html#/inputs

Search

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

## Browse Directory

/inputs Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	user01	supergroup	120.74 KB	22/8/2559 00:22:01	1	128 MB	<a href="#">sales_data_jan2009.csv</a>

Hadoop, 2015.

# 4. Hands on HDFS

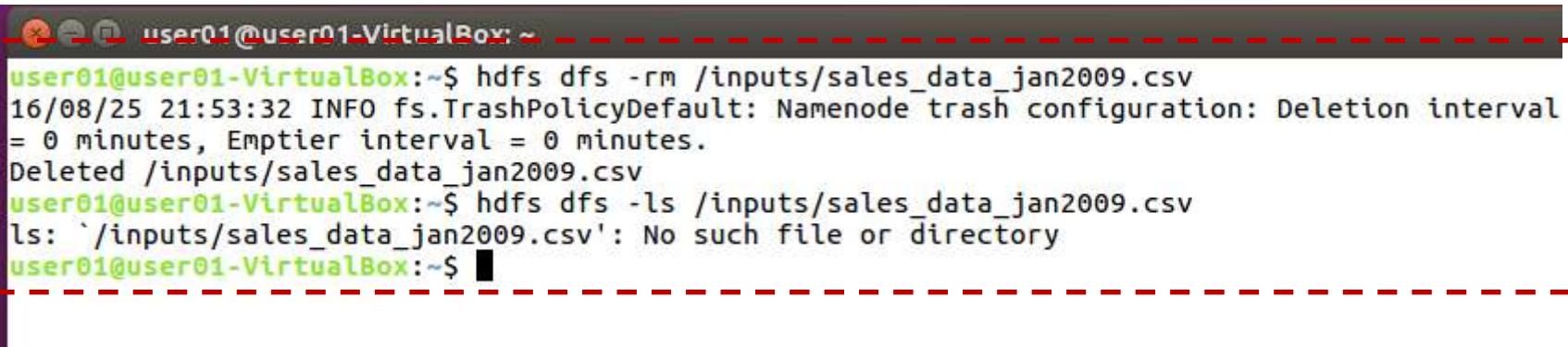
```
$ hdfs dfs -cat /inputs/sales_data_jan2009.csv
```



```
user01@user01-VirtualBox:~$ hdfs dfs -cat /inputs/sales_data_jan2009.csv
Transaction_date,Product,Price,Payment_Type,Name,City,State,Country,Account_Created,Last_Login
1/2/09 6:17,Product1,1200,Mastercard,carolina,Basildon,England,United Kingdom,1/2/09 6:00,1/2/
1/2/09 4:53,Product1,1200,Visa,Betina,Parkville ,MO,United States,1/2/09 4:4
1/2/09 13:08,Product1,1200,Mastercard,Federica e Andrea,Astoria ,OR,United
1/3/09 14:44,Product1,1200,Visa,Gouya,Echuca,Victoria,Australia,9/25/05 21:13,1/3/09 14:22,-36
1/4/09 12:56,Product2,3600,Visa,Gerd W ,Cahaba Heights ,AL,United States,11/15/08
1/4/09 13:19,Product1,1200,Visa,LAURENCE,Mickleton ,NJ,United States,9/24/08
1/4/09 20:11,Product1,1200,Mastercard,Fleur,Peoria ,IL,United States,1/3/
1/2/09 20:09,Product1,1200,Mastercard,adam,Martin ,TN,United States,1/2/0
1/4/09 13:17,Product1,1200,Mastercard,Renee Elisabeth,Tel Aviv,Tel Aviv,Israel,1/4/09 13:03,1/
1/4/09 14:11,Product1,1200,Visa,Aidan,Chatou,Ile-de-France,France,6/3/08 4:22,1/5/09 1:17,48.8
1/5/09 2:42,Product1,1200,Diners,Stacy,New York ,NY,United States,1/5/09 2:
1/5/09 5:39,Product1,1200,Amex,Heidi,Eindhoven,Noord-Brabant,Netherlands,1/5/09 4:55,1/5/09 8:
1/2/09 9:16,Product1,1200,Mastercard,Sean ,Shavano Park ,TX,United States,1/2/0
1/5/09 10:08,Product1,1200,Visa,Georgia,Eagle ,ID,United States,11/11/08
1/2/09 14:18,Product1,1200,Visa,Richard,Riverside ,NJ,United States,12/9/08
1/4/09 1:05,Product1,1200,Diners,Leanne,Julianstown,Meath,Ireland,1/4/09 0:00,1/5/09 13:36,53.
1/5/09 11:37,Product1,1200,Visa,Janet,Ottawa,Ontario,Canada,1/5/09 9:35,1/5/09 19:24,45.416666
1/6/09 5:02,Product1,1200,Diners,barbara,Hyderabad,Andhra Pradesh,India,1/6/09 2:41,1/6/09 7:5
1/6/09 7:45,Product2,3600,Visa,Sabine,London,England,United Kingdom,1/6/09 7:00,1/6/09 9:17,51
1/2/09 7:35,Product1,1200,Diners,Hani,Salt Lake City ,UT,United States,12/30/08 5
1/6/09 12:56,Product1,1200,Visa,Jeremy,Manchester,England,United Kingdom,1/6/09 10:58,1/6/09 1
1/1/09 11:05,Product1,1200,Diners,Janis,Ballynora,Cork,Ireland,12/10/07 12:37,1/7/09 1:52,51.8
1/5/09 4:10,Product1,1200,Mastercard,Nicola,Roodepoort,Gauteng,South Africa,1/5/09 2:33,1/7/09
1/6/09 7:18,Product1,1200,Visa,asuman,Chula Vista ,CA,United States,1/6/09 7:0
1/2/09 1:11,Product1,1200,Mastercard,Lena,Kuopio,Ita-Suomen Laani,Finland,12/31/08 2:48,1/7/09
```

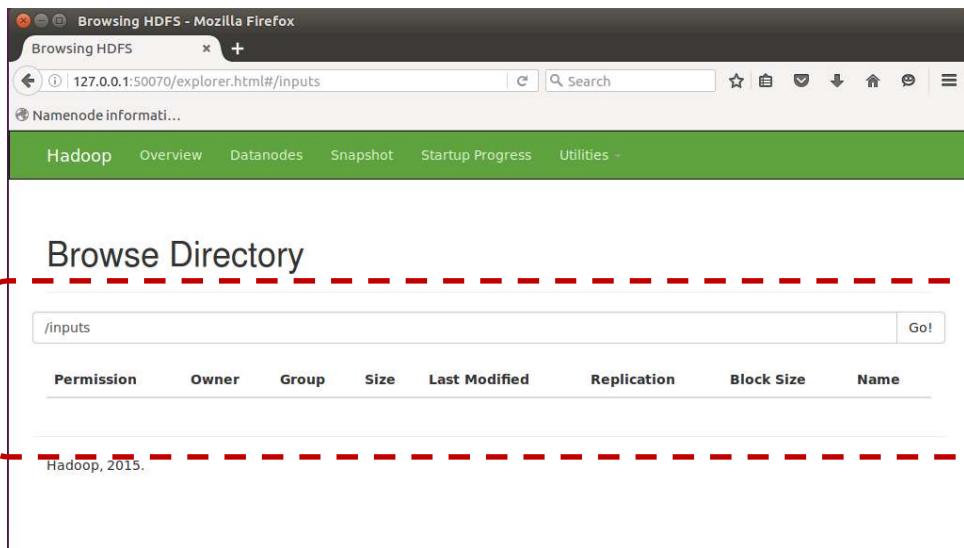
# 4. Hands on HDFS

```
$ hdfs dfs -rm /inputs/sales_data_jan2009.csv
```



A terminal window titled "user01@user01-VirtualBox:~\$". It shows the command `hdfs dfs -rm /inputs/sales_data_jan2009.csv` being run, followed by its execution log. The log indicates the file was deleted successfully, and then attempts to list it again fail with a "No such file or directory" error.

```
user01@user01-VirtualBox:~$ hdfs dfs -rm /inputs/sales_data_jan2009.csv
16/08/25 21:53:32 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /inputs/sales_data_jan2009.csv
user01@user01-VirtualBox:~$ hdfs dfs -ls /inputs/sales_data_jan2009.csv
ls: `/inputs/sales_data_jan2009.csv': No such file or directory
user01@user01-VirtualBox:~$
```



A screenshot of a web browser window titled "Browsing HDFS - Mozilla Firefox". The address bar shows "127.0.0.1:50070/explorer.html#/inputs". The page header includes "Namenode information" and a navigation menu with links to "Hadoop", "Overview", "Datanodes", "Snapshot", "Startup Progress", and "Utilities". The main content area is titled "Browse Directory" and contains a search bar with the path "/inputs" and a "Go!" button. Below the search bar is a table header with columns: "Permission", "Owner", "Group", "Size", "Last Modified", "Replication", "Block Size", and "Name". At the bottom of the page, there is a footer note: "Hadoop, 2015.".

# 4. Hands on HDFS

```
$ hdfs dfs -copyFromLocal ./sales_data_jan2009.csv /inputs/sales_data_jan2009.csv
```

```
$ hdfs dfs -ls /inputs/
```

```
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ hdfs dfs -ls /inputs/
Found 1 items
-rw-r--r-- 1 user01 supergroup 123637 2016-08-22 00:22 /inputs/sales_data_jan2009.csv
user01@user01-VirtualBox:/usr/local/hadoop/etc/hadoop$ █
```

# 4. Hands on HDFS

(Optional) Review linux file system

```
user01@user01-VirtualBox:~$ cd /var/hadoop_data/datanode/
user01@user01-VirtualBox:/var/hadoop_data/datanode$ ls
current
user01@user01-VirtualBox:/var/hadoop_data/datanode$ cd current/
user01@user01-VirtualBox:/var/hadoop_data/datanode/current$ ls
BP-1774458098-127.0.1.1-1471799424565 VERSION
user01@user01-VirtualBox:/var/hadoop_data/datanode/current$ cd BP-1774458098-127.0.1.1-1471799
424565/
user01@user01-VirtualBox:/var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-14717994245
65$ ls
current  scanner.cursor  tmp
user01@user01-VirtualBox:/var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-14717994245
65$ cd current/
user01@user01-VirtualBox:/var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-14717994245
65/current$ ls
dfsUsed  finalized  rbw  VERSION
user01@user01-VirtualBox:/var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-14717994245
65/current$ █
```



# Apache Hadoop YARN and MRv2

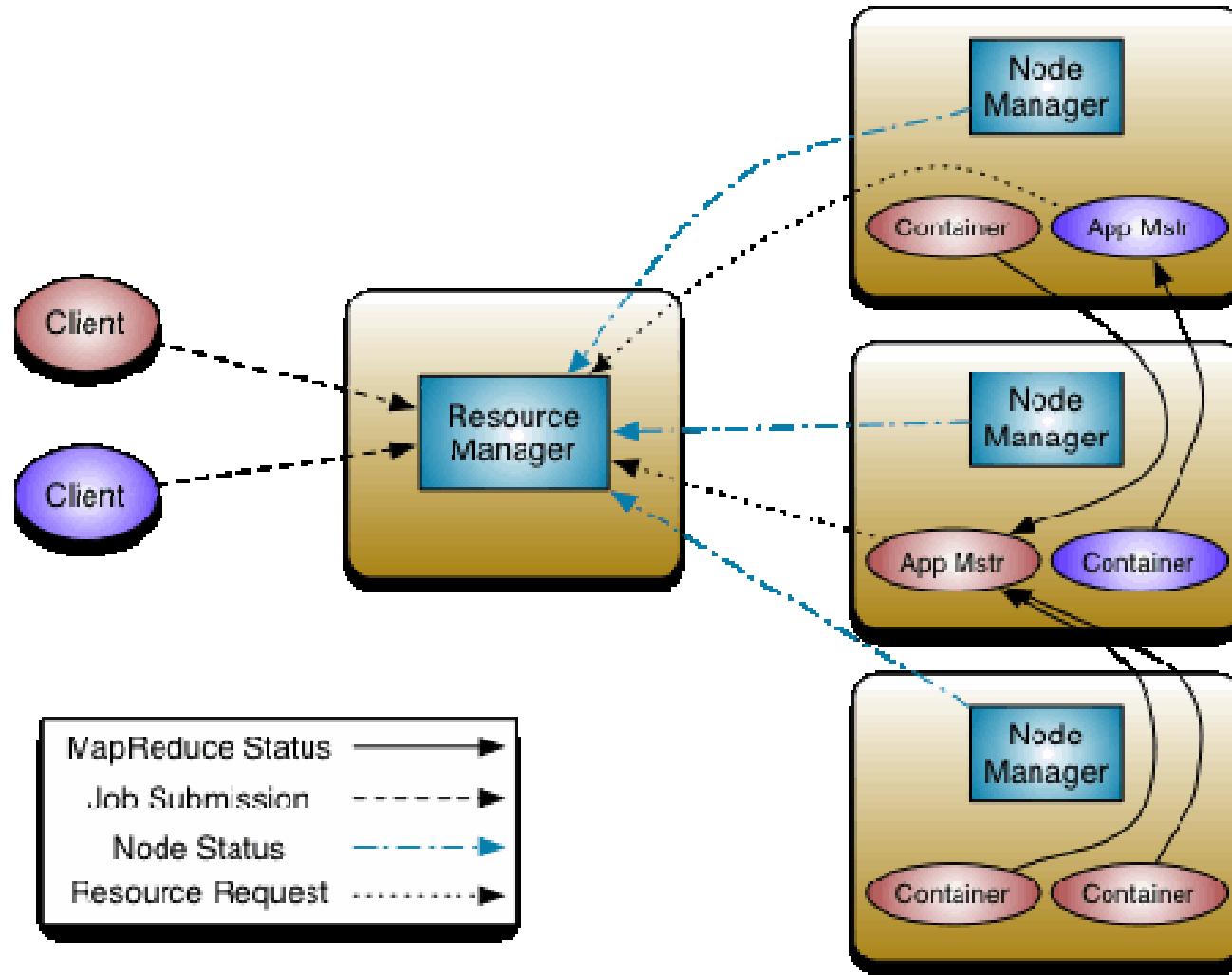
อ.ดนัยรัตน์ ธนบดีธรรมจารี

Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

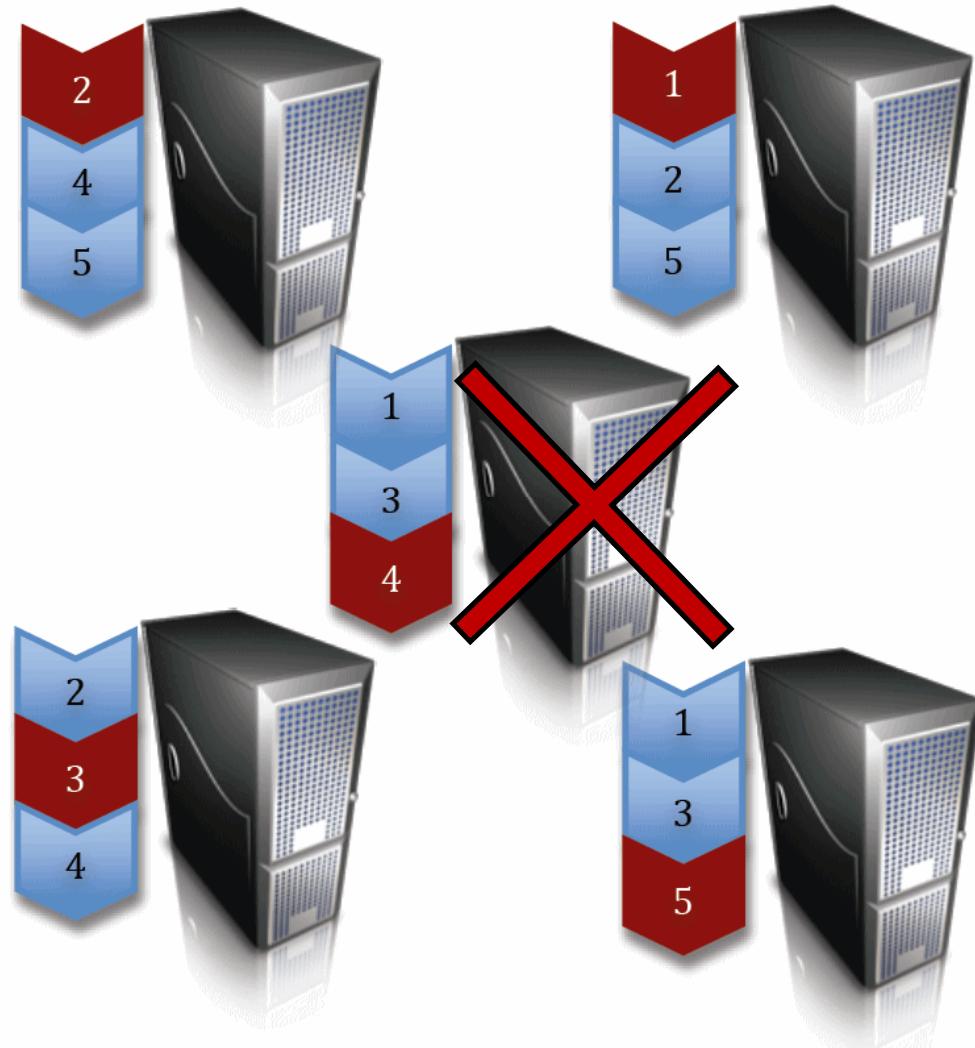
# YARN: Yet Another Resource Negotiator



MRv2 maintains API compatibility with previous stable release (hadoop-1.x). This means that all Map-Reduce jobs should still run unchanged on top of MRv2 with just a recompile.

# MapReduce: Distributed Processing

*Hadoop takes advantage of HDFS' data distribution strategy to push work out to many nodes in a cluster. This allows analyses to run in parallel and eliminates the bottlenecks imposed by monolithic storage systems.*



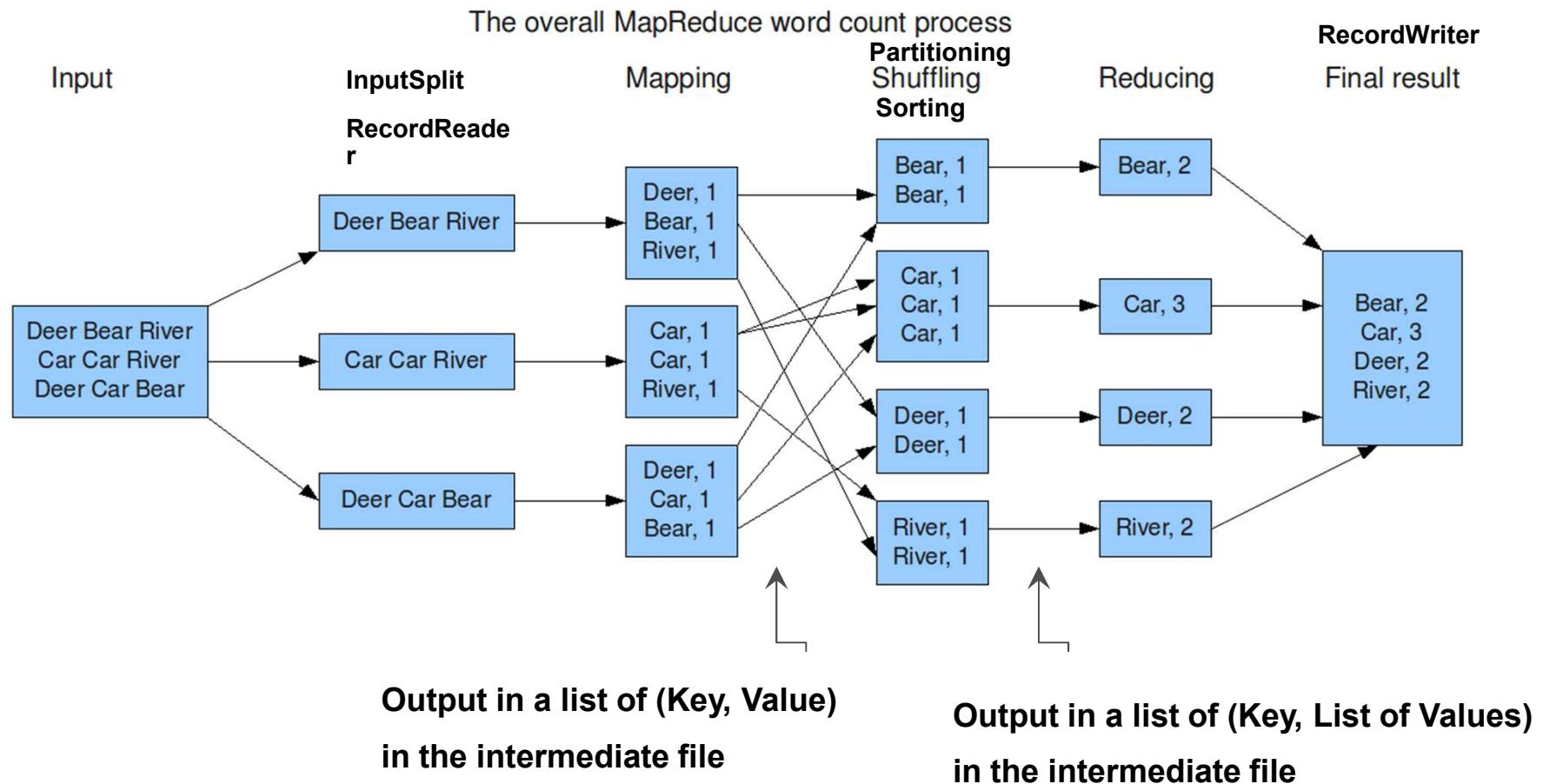
[apache.org/hadoop/](http://apache.org/hadoop/)

# MapReduce Framework

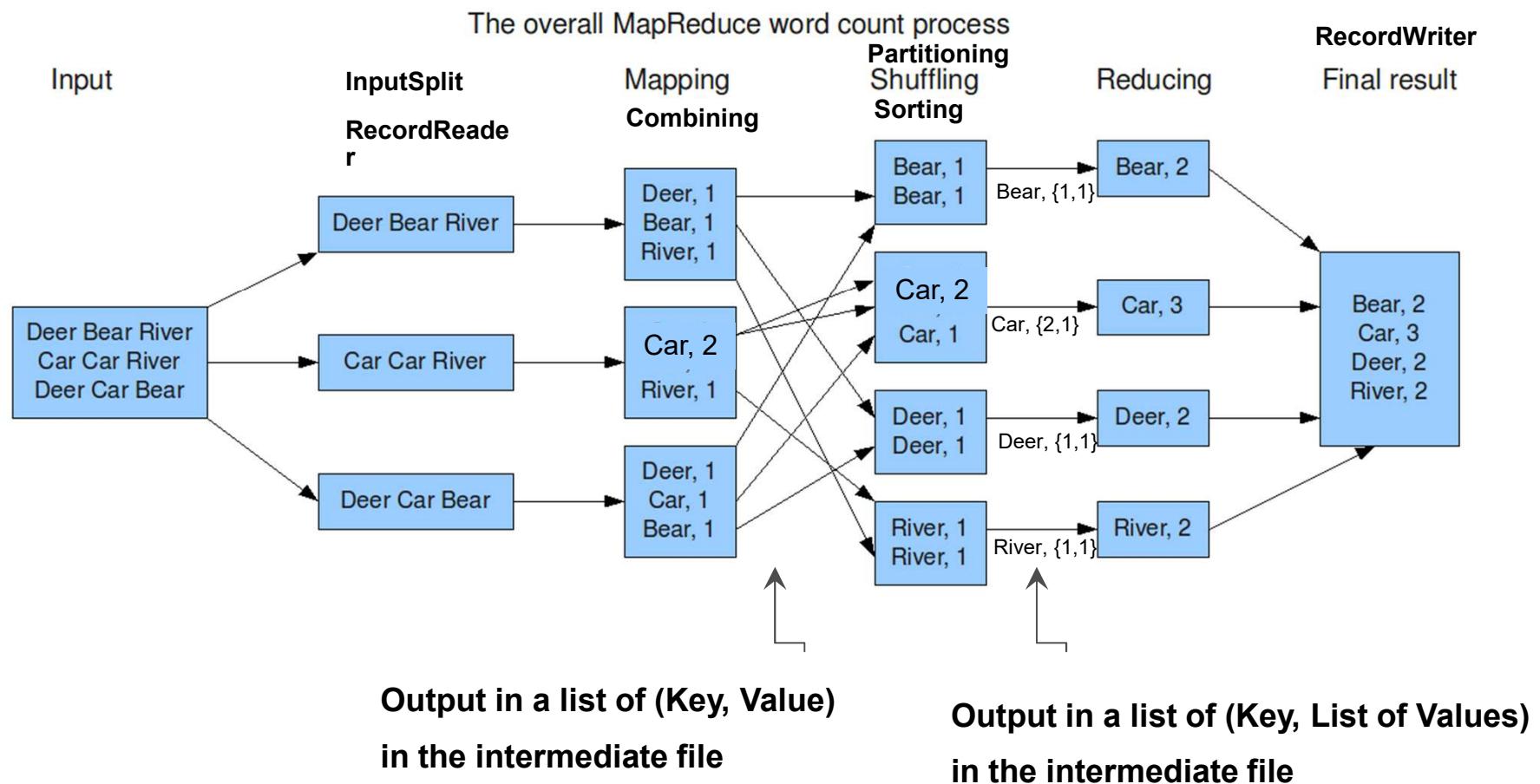
**map:**  $(K_1, V_1) \rightarrow \text{list}(K_2, V_2)$

**reduce:**  $(K_2, \text{list}(V_2)) \rightarrow \text{list}(K_3, V_3)$

# How does the MapReduce work?



# How does the MapReduce work?



# MapReduce Processing – The Data flow

- 1. InputFormat, InputSplits, RecordReader**
- 2. Mapper - your focus is here**
- 3. Partition, Shuffle & Sort**
- 4. Reducer - your focus is here**
- 5. OutputFormat, RecordWriter**

# InputFormat

InputFormat:	Description:	Key:	Value:
TextInputFormat	Default format; reads lines of text files	The byte offset of the line	The line contents
KeyValueInputFormat	Parses lines into key, val pairs	Everything up to the first tab character	The remainder of the line
SequenceFileInputFormat	A Hadoop-specific high-performance binary format	user-defined	user-defined

# InputSplit

An InputSplit describes a unit of work that comprises a single *map task*.

InputSplit presents a byte-oriented view of the input.

You can control this value by setting the mapred.min.split.size parameter in core-site.xml, or by overriding the parameter in the JobConf object used to submit a particular MapReduce job.

# RecordReader

[RecordReader](#) reads <key, value> pairs from an InputSplit.

Typically the RecordReader converts the byte-oriented view of the input, provided by the InputSplit, and presents a record-oriented to the Mapper

# Mapper

**Mapper:** The Mapper performs the user-defined logic to the input a key, value and emits (key, value) pair(s) which are forwarded to the Reducers.

## Partition, Shuffle & Sort

After the first map tasks have completed, the nodes may still be performing several more map tasks each. But they also begin exchanging the intermediate outputs from the map tasks to where they are required by the reducers.

Partitioner controls the partitioning of map-outputs to assign to reduce task . he total number of partitions is the same as the number of reduce tasks for the job

The set of intermediate keys on a single node is automatically sorted by internal Hadoop before they are presented to the Reducer

This process of moving map outputs to the reducers is known as shuffling.

# Reducer

This is an instance of user-provided code that performs read each key, iterator of values in the partition assigned. The *OutputCollector* object in Reducer phase has a method named collect() which will collect a (key, value) output.

## OutputFormat, Record Writer

**OutputFormat** governs the writing format in **OutputCollector** and **RecordWriter** writes output into HDFS.

<b>TextOutputFormat</b>	Default; writes lines in "key \t value" form
<b>SequenceFileOutputFormat</b>	Writes binary files suitable for reading into subsequent MapReduce jobs
<b>NullOutputFormat</b>	generates no output files

# 5. Hands on MRv2

```
$ cd  
$ nano WordCount.java
```

```
import java.io.IOException;  
import java.util.StringTokenizer;  
  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.Mapper;  
import org.apache.hadoop.mapreduce.Reducer;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
  
public class WordCount {  
  
    public static class TokenizerMapper  
        extends Mapper<Object, Text, Text, IntWritable>{  
  
        private final static IntWritable one = new IntWritable(1);  
        private Text word = new Text();  
  
        public void map(Object key, Text value, Context context  
                       ) throws IOException, InterruptedException {  
            StringTokenizer itr = new StringTokenizer(value.toString());  
            while (itr.hasMoreTokens()) {  
                word.set(itr.nextToken());  
                context.write(word, one);  
            }  
        }  
    }  
  
    public static class IntSumReducer  
        extends Reducer<Text,IntWritable,Text,IntWritable> {  
        private IntWritable result = new IntWritable();  
  
        public void reduce(Text key, Iterable<IntWritable> values,  
                          Context context  
                          ) throws IOException, InterruptedException {  
            int sum = 0;  
            for (IntWritable val : values) {  
                sum += val.get();  
            }  
            result.set(sum);  
            context.write(key, result);  
        }  
    }  
  
    public static void main(String[] args) throws Exception {  
        Configuration conf = new Configuration();  
        Job job = Job.getInstance(conf, "word count");  
        job.setJarByClass(WordCount.class);  
        job.setMapperClass(TokenizerMapper.class);  
        job.setCombinerClass(IntSumReducer.class);  
        job.setReducerClass(IntSumReducer.class);  
        job.setOutputKeyClass(Text.class);  
        job.setOutputValueClass(IntWritable.class);  
        FileInputFormat.addInputPath(job, new Path(args[0]));  
        FileOutputFormat.setOutputPath(job, new Path(args[1]));  
        System.exit(job.waitForCompletion(true) ? 0 : 1);  
    }  
}
```

# 5. Hands on MRv2

Compile .java to .class

```
$ mkdir wordcount_classes
```

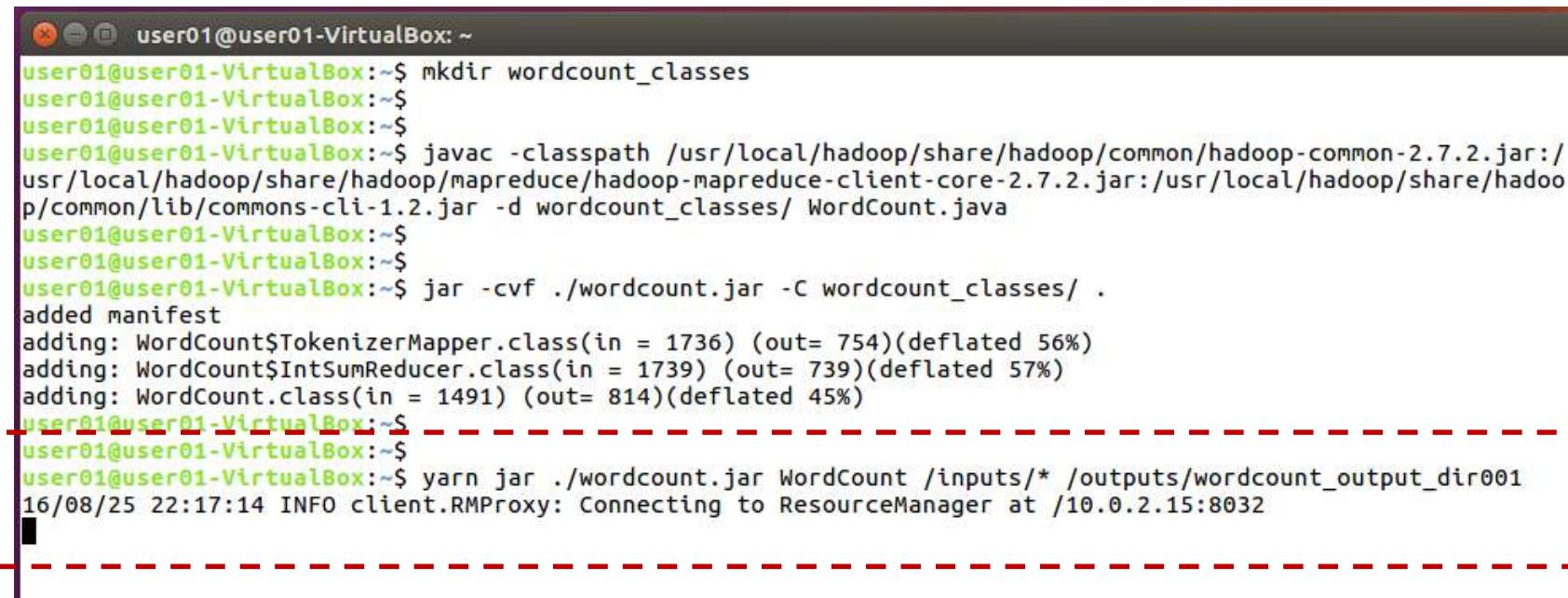
```
$ javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-  
2.7.2.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-  
core-2.7.2.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -  
d wordcount_classes/ WordCount.java
```

```
$ jar -cvf ./wordcount.jar -C wordcount_classes/ •
```

# 5. Hands on MRv2

Execute YARN

```
$ yarn jar ./wordcount.jar WordCount /inputs/* /outputs/wordcount_output_dir001
```



```
user01@user01-VirtualBox:~$ mkdir wordcount_classes
user01@user01-VirtualBox:~$ 
user01@user01-VirtualBox:~$ javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.7.2.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.7.2.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d wordcount_classes/ WordCount.java
user01@user01-VirtualBox:~$ 
user01@user01-VirtualBox:~$ jar -cvf ./wordcount.jar -C wordcount_classes/ .
added manifest
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 754)(deflated 56%)
adding: WordCount$IntSumReducer.class(in = 1739) (out= 739)(deflated 57%)
adding: WordCount.class(in = 1491) (out= 814)(deflated 45%)
user01@user01-VirtualBox:~$ 
user01@user01-VirtualBox:~$ yarn jar ./wordcount.jar WordCount /inputs/* /outputs/wordcount_output_dir001
16/08/25 22:17:14 INFO client.RMProxy: Connecting to ResourceManager at /10.0.2.15:8032
```

# 5. Hands on MRv2

Open another terminal to check Yarn Processes are running  
\$ jps

The image shows two terminal windows side-by-side. The left terminal window displays the output of a Hadoop word count job. It includes logs from the MapReduce proxy, InputFormat, JobSubmitter, and Job classes, detailing the submission and execution of the job. The right terminal window shows the results of a 'jps' command, listing various YARN components: YarnChild, RunJar, NameNode, Jps, NodeManager, DataNode, MRAppMaster, ResourceManager, YarnChild, and SecondaryNameNode.

```
user01@user01-VirtualBox:~$ yarn jar ./wordcount.jar WordCount /inputs/* /output
s/wordcount_output_dir004
16/09/22 14:54:16 INFO client.RMProxy: Connecting to
15:8032
16/09/22 14:54:27 WARN mapreduce.JobResourceUploader: parsing not performed. Implement the Tool interface
n with ToolRunner to remedy this.
16/09/22 14:54:28 INFO input.FileInputFormat: Total
16/09/22 14:54:28 INFO mapreduce.JobSubmitter: numb
16/09/22 14:54:28 INFO mapreduce.JobSubmitter: Subm
74525942644_0002
16/09/22 14:54:29 INFO impl.YarnClientImpl: Submitt
74525942644_0002
16/09/22 14:54:29 INFO mapreduce.Job: The url to t
:8088/proxy/application_1474525942644_0002/
16/09/22 14:54:29 INFO mapreduce.Job: Running job:
16/09/22 14:54:43 INFO mapreduce.Job: Job job_14745
mode : false
16/09/22 14:54:43 INFO mapreduce.Job: map 0% reduc
16/09/22 14:54:55 INFO mapreduce.Job: map 100% red
16/09/22 14:55:06 INFO mapreduce.Job: map 100% red
```

```
user01@user01-VirtualBox:~$ jps
5568 YarnChild -----
5329 RunJar
3297 NameNode
5586 Jps
3922 NodeManager
3427 DataNode
5445 MRAppMaster -----
3799 ResourceManager -----
5560 YarnChild -----
3630 SecondaryNameNode
```

# 5. Hands on MRv2

```
user01@user01-VirtualBox: ~
    Combine input records=5548
    Combine output records=3731
    Reduce input groups=3731
    Reduce shuffle bytes=120297
    Reduce input records=3731
    Reduce output records=3731
    Spilled Records=7462
    Shuffled Maps =1
    Failed Shuffles=0
    Merged Map outputs=1
    GC time elapsed (ms)=258
    CPU time spent (ms)=1930
    Physical memory (bytes) snapshot=315682816
    Virtual memory (bytes) snapshot=3810422784
    Total committed heap usage (bytes)=202379264
  Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=123637
  File Output Format Counters
    Bytes Written=105427
user01@user01-VirtualBox:~$ █
```

## 5. Hands on MRv2

## Review the results from Map Reduce Command

```
$ hdfs dfs -ls /outputs
```

```
$ hdfs dfs -ls /outputs/wordcount_output_dir001
```

```
$ hdfs dfs -cat /outputs/wordcount_output_dir001/part-r-00000
```

```
user01@user01-VirtualBox:~$ hdfs dfs -cat /outputs/wordcount_output_dir001/part-r-00000
'Adlyah,Al      1
(Bruxelles),Belgium,1/11/09      1
(Bruxelles),Belgium,1/24/09      1
(Bruxelles),Belgium,1/31/08      1
(Bruxelles),Belgium,12/18/08     1
(Bruxelles),Belgium,6/30/08      1
(Bruxelles),Belgium,7/13/08      1
,7500,Mastercard,Amanda,Shreveport      1
,AK,United      5
,AL,United      1
,AR,United      3
,AZ,United      10
,Anchorage      1
,Arlington      1
,Ashburn        1
,Ashford,England,United 1
,Auckland,Auckland,New  1
,Austin 1
,Ballyneety,Limerick,Ireland,1/7/09      1
,Basingstoke,England,United      1
,Beachwood      1
,Bluffton       1
,Borja,Bohol,Philippines,1/17/09     1
,Brooklyn       1
```

# 5. Hands on MRv2

Review Linux File System

```
$ cd /var/hadoop_data/namenode/current/  
$ ls -l
```

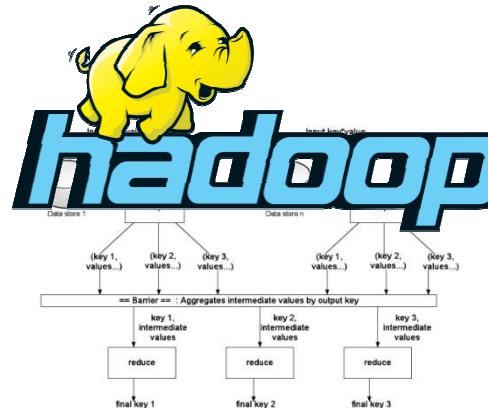
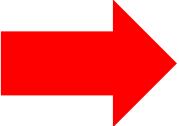
```
$ cd /var/hadoop_data/datanode/current/  
$ ls -l
```

# 5. Hands on MRv2

```
user01@user01-VirtualBox: /var/hadoop_data/namenode/current$  
user01@user01-VirtualBox: /var/hadoop_data/namenode/current$  
user01@user01-VirtualBox: /var/hadoop_data/namenode/current$ cd /var/hadoop_data/namenode/current/  
user01@user01-VirtualBox: /var/hadoop_data/namenode/current$ ll  
total 3128  
drwxrwxr-x 2 user01 user01 4096 斯.គ. 25 22:31 ./  
drwxr-xr-x 3 user01 user01 4096 斯.គ. 25 21:37 ../  
-rw-rw-r-- 1 user01 user01 1048576 斯.គ. 22 00:22 edits_00000000000000001-00000000000000009  
-rw-rw-r-- 1 user01 user01 42 斯.គ. 22 10:55 edits_00000000000000010-00000000000000011  
-rw-rw-r-- 1 user01 user01 42 斯.គ. 22 11:55 edits_00000000000000012-00000000000000013  
-rw-rw-r-- 1 user01 user01 1048576 斯.គ. 22 11:55 edits_00000000000000014-00000000000000014  
-rw-rw-r-- 1 user01 user01 12521 斯.គ. 25 22:30 edits_00000000000000015-00000000000000014  
-rw-rw-r-- 1 user01 user01 1048576 斯.គ. 25 22:41 edits_inprogress_00000000000000015  
-rw-rw-r-- 1 user01 user01 581 斯.គ. 25 21:37 fsimage_00000000000000014  
-rw-rw-r-- 1 user01 user01 62 斯.គ. 25 21:37 fsimage_00000000000000014.md5  
-rw-rw-r-- 1 user01 user01 1682 斯.គ. 25 22:31 fsimage_00000000000000014  
-rw-rw-r-- 1 user01 user01 62 斯.គ. 25 22:31 fsimage_00000000000000014.md5  
-rw-rw-r-- 1 user01 user01 4 斯.គ. 25 22:30 seen_txid  
-rw-rw-r-- 1 user01 user01 202 斯.គ. 25 21:37 VERSION  
user01@user01-VirtualBox: /var/hadoop_data/namenode/current$ █
```

# 5. Hands on MRv2

```
user01@user01-VirtualBox: /var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-1471799424565/current/finalize
total 12
drwxrwxr-x 3 user01 user01 4096 ສ.ຄ. 22 00:22 .
drwxrwxr-x 3 user01 user01 4096 ສ.ຄ. 22 00:22 ..
drwxrwxr-x 2 user01 user01 4096 ສ.ຄ. 25 22:41 subdir0/
user01@user01-VirtualBox:/var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-1471799424565/current/finalized/subdir0$ cd subdir0/
user01@user01-VirtualBox:/var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-1471799424565/current/finalized/subdir0$ ll
total 688
drwxrwxr-x 2 user01 user01 4096 ສ.ຄ. 25 22:41 .
drwxrwxr-x 3 user01 user01 4096 ສ.ຄ. 22 00:22 ..
-rw-rw-r-- 1 user01 user01 123637 ສ.ຄ. 25 21:57 blk_1073741826
-rw-rw-r-- 1 user01 user01 975 ສ.ຄ. 25 21:57 blk_1073741826_1002.meta
-rw-rw-r-- 1 user01 user01 105427 ສ.ຄ. 25 22:18 blk_1073741833
-rw-rw-r-- 1 user01 user01 831 ສ.ຄ. 25 22:18 blk_1073741833_1009.meta
-rw-rw-r-- 1 user01 user01 351 ສ.ຄ. 25 22:18 blk_1073741834
-rw-rw-r-- 1 user01 user01 11 ສ.ຄ. 25 22:18 blk_1073741834_1010.meta
-rw-rw-r-- 1 user01 user01 33867 ສ.ຄ. 25 22:18 blk_1073741835
-rw-rw-r-- 1 user01 user01 275 ສ.ຄ. 25 22:18 blk_1073741835_1011.meta
-rw-rw-r-- 1 user01 user01 115668 ສ.ຄ. 25 22:18 blk_1073741836
-rw-rw-r-- 1 user01 user01 911 ສ.ຄ. 25 22:18 blk_1073741836_1012.meta
-rw-rw-r-- 1 user01 user01 105427 ສ.ຄ. 25 22:41 blk_1073741843
-rw-rw-r-- 1 user01 user01 831 ສ.ຄ. 25 22:41 blk_1073741843_1019.meta
-rw-rw-r-- 1 user01 user01 349 ສ.ຄ. 25 22:41 blk_1073741844
-rw-rw-r-- 1 user01 user01 11 ສ.ຄ. 25 22:41 blk_1073741844_1020.meta
-rw-rw-r-- 1 user01 user01 33823 ສ.ຄ. 25 22:41 blk_1073741845
-rw-rw-r-- 1 user01 user01 275 ສ.ຄ. 25 22:41 blk_1073741845_1021.meta
-rw-rw-r-- 1 user01 user01 115668 ສ.ຄ. 25 22:41 blk_1073741846
-rw-rw-r-- 1 user01 user01 911 ສ.ຄ. 25 22:41 blk_1073741846_1022.meta
user01@user01-VirtualBox:/var/hadoop_data/datanode/current/BP-1774458098-127.0.1.1-1471799424565/current/finalized/subdir0$ cd ..
```



# Very Large Server Log files

# MRv2 in Data Lake

# Analyzing Large Server Log Files with Hadoop MapReduce

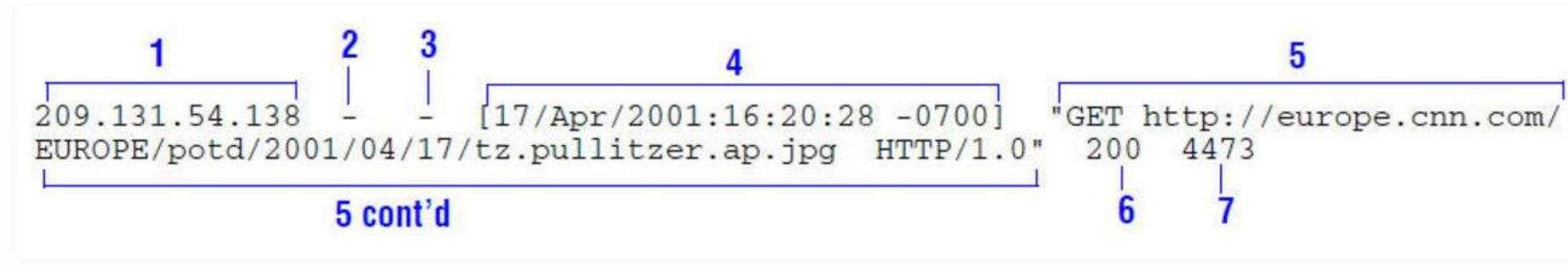
## ວ.ດນ້ຍຮັກ ຮນບດີໂຮມຈາກ

# Line ID: Danairat

# **FB: Danairat Thanabodithammachari**

**+668-1559-1446**

# Understand Server Log



No.	Field	Description
1	chi	The IP address of the client's host machine.
2	-	This hyphen (-) is always present in Netscape log entries.
3	caun	The authenticated client username. A hyphen (-) means no authentication was required.
4	cqtn	The date and time of the client request, enclosed in brackets.
5	cqtx	The request line, enclosed in quotes.
6	pssc	The proxy response status code (HTTP reply code).
7	pscl	The length of the Traffic Server response to the client in bytes.

# Put HTTP log to HDFS

hdfs dfs -put server.log /user/inputs

Input server log

```
alpine.prodigy.com - [01/AU/1995:00:00:23 -9490] "GET /shuttle/missions/etc/TI/sts-71-patch-seall.gif HTTP/1.0" 200 1205
ppp-mia-39.shadow.net - [01/AU/1995:00:00:25 -9490] "GET /images/kscimagemedium.gif HTTP/1.0" 200 5868
dial22.lloyd.com - [01/AU/1995:00:00:26 -9490] "GET /history/missions/sts-71/patchseall.gif HTTP/1.0" 200 8176
piemontry.prodigy.com - [01/AU/1995:00:00:28 -9490] "GET /history/missions/sts-71/patchseall.gif HTTP/1.0" 200 10176
205.188.154.54 - [01/AU/1995:00:00:40 -9490] "GET /images/KSC-logoseall.gif HTTP/1.0" 200 786
ix-12-01.ix.netcom.com - [01/AU/1995:00:00:41 -9490] "GET /history/missions/sts-71/patchseall.gif HTTP/1.0" 200 3898
ppp-mia-39.shadow.net - [01/AU/1995:00:00:41 -9490] "GET /images/KSC-logoseall.gif HTTP/1.0" 200 786
piemontry.prodigy.com - [01/AU/1995:00:00:41 -9490] "GET /history/missions/sts-71/patchseall.gif HTTP/1.0" 200 12053
205.188.154.54 - [01/AU/1995:00:00:41 -9490] "GET /images/KSC-logoseall.gif HTTP/1.0" 200 12053
ppp-mia-39.shadow.net - [01/AU/1995:00:00:41 -9490] "GET /images/USPS-logoseall.gif HTTP/1.0" 200 648
ppp-mia-39.shadow.net - [01/AU/1995:00:00:41 -9490] "GET /images/USPS-logoseall.gif HTTP/1.0" 200 648
ix-12-01.ix.netcom.com - [01/AU/1995:00:00:44 -9490] "GET /huttle/countdown/count.gif HTTP/1.0" 200 4010
ppgprod1.prodigy.com - [01/AU/1995:00:00:44 -9490] "GET /shuttle/missions/etc/TI/sts-71-patch-seall.gif HTTP/1.0" 200 12940
piemontry.prodigy.com - [01/AU/1995:00:00:54 -9490] "GET /shuttle/missions/etc/TI/sts-71-patch-seall.gif HTTP/1.0" 200 12054
schwager-clients.net - [01/AU/1995:00:00:54 -9490] "GET /huttle/missions/sts-71/patchseall.gif HTTP/1.0" 200 8152
piemontry.prodigy.com - [01/AU/1995:00:00:59 -9490] "GET /history/missions/sts-71/patchseall.gif HTTP/1.0" 200 77163
199.72.81.95 - [01/AU/1995:00:00:59 -9490] "GET /history/HTTP/1.0" 200 2082
port22.annex2.netlink.com - [01/AU/1995:00:01:00 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 1882
port22.annex2.netlink.com - [01/AU/1995:00:01:01 -9490] "GET /software/Windows/Windows.gif HTTP/1.0" 200 25218
port22.annex2.netlink.com - [01/AU/1995:00:01:04 -9490] "GET /software/Windows/Windows.gif HTTP/1.0" 200 25218
edt-012.coopserve.com - [01/AU/1995:00:01:05 -9490] "GET /huttle/technology/images/98/e-mail.gif HTTP/1.0" 200 42732
205.188.154.54 - [01/AU/1995:00:01:05 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 1882
ix-12-01.ix.netcom.com - [01/AU/1995:00:01:05 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 1882
205.188.154.54 - [01/AU/1995:00:01:05 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 1882
dial22.lloyd.com - [01/AU/1995:00:01:12 -9490] "GET /news/sci/spec/shuttle/archive/sci-specs-shuttle-22-apr-1995-90.txt HTTP/1.0" 404 -
205.188.154.54 - [01/AU/1995:00:01:12 -9490] "GET /news/sci/spec/shuttle/archive/sci-specs-shuttle-22-apr-1995-90.txt HTTP/1.0" 404 -
piemontry.prodigy.com - [01/AU/1995:00:01:13 -9490] "GET /shuttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 55665
remktz-coopserve1.ad.ad - [01/AU/1995:00:01:14 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 12059
port22.annex2.netlink.com - [01/AU/1995:00:01:15 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 12059
ix-12-01.ix.netcom.com - [01/AU/1995:00:01:15 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 12059
205.188.154.54 - [01/AU/1995:00:01:15 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 12059
205.188.154.54 - [01/AU/1995:00:01:19 -9490] "GET /shuttle/missions/etc/TI/sts-71/patchseall.gif HTTP/1.0" 200 1124
piemontry.prodigy.com - [01/AU/1995:00:01:19 -9490] "GET /huttle/countdown/visde/linecountse.gif HTTP/1.0" 200 76712
piemontry.prodigy.com - [01/AU/1995:00:01:19 -9490] "GET /huttle/countdown/visde/linecountse.gif HTTP/1.0" 200 76712
alpinet.gab.com - [01/AU/1995:00:01:20 -9490] "GET /huttle/resources/orbiters/endeavour.html HTTP/1.0" 200 6168
ix-12-01.ix.netcom.com - [01/AU/1995:00:01:21 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 12057
port22.annex2.netlink.com - [01/AU/1995:00:01:21 -9490] "GET /images/KSC-logoseall.gif HTTP/1.0" 200 12057
port22.annex2.netlink.com - [01/AU/1995:00:01:21 -9490] "GET /images/KSC-logoseall.gif HTTP/1.0" 200 12057
remktz-coopserve1.ad.ad - [01/AU/1995:00:01:27 -9490] "GET /huttle/missions/etc/TI/sts-71-patchseall.gif HTTP/1.0" 200 363
lin097.tcdirect.net - [01/AU/1995:00:01:27 -9490] "GET /huttle/countdown/HTTP/1.0" 200 3886
205.188.154.54 - [01/AU/1995:00:01:27 -9490] "GET /huttle/missions/etc/TI/sts-71/patchseall.gif HTTP/1.0" 200 10991
lin097.tcdirect.net - [01/AU/1995:00:01:31 -9490] "GET /images/MSD-logoseall.gif HTTP/1.0" 200 786
```

## Count by Host Name Results

dial22.lloyd.com	61716
ix-0rl2-01.ix.netcom.com	3985
net-1-141.eden.com	34029
ppp-mia-30.shadow.net	14089
unicomp6.unicomp.net	46285
waters-gw.starway.net.au	6723

# Write MR program

```
$ vi LogWritable.java  
$ vi LogProcessorMap.java  
$ vi LogProcessorReduce.java  
$ vi LogProcessor.java
```

# Write MR program - LogWritable.java

```
import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.WritableComparable;

public class LogWritable implements WritableComparable<LogWritable> {

    private Text userIP, timestamp, request;
    private IntWritable responseSize, status;

    public LogWritable() {
        this.userIP = new Text();
        this.timestamp = new Text();
        this.request = new Text();
        this.responseSize = new IntWritable();
        this.status = new IntWritable();
    }

    public void set (String userIP, String timestamp, String request, int bytes, int status)
    {
        this.userIP.set(userIP);
        this.timestamp.set(timestamp);
        this.request.set(request);
        this.responseSize.set(bytes);
        this.status.set(status);
    }

    @Override
    public void readFields(DataInput in) throws IOException {
        userIP.readFields(in);
        timestamp.readFields(in);
        request.readFields(in);
        responseSize.readFields(in);
        status.readFields(in);
    }
}
```

# Write MR program - LogWritable.java (Cont.)

```
    @Override
    public void write(DataOutput out) throws IOException {
        userIP.write(out);
        timestamp.write(out);
        request.write(out);
        responseSize.write(out);
        status.write(out);
    }

    @Override
    public int compareTo(LogWritable o) {
        if (userIP.compareTo(o.userIP) == 0) {
            return
        timestamp.compareTo(o.timestamp);
        } else
            return
        userIP.compareTo(o.userIP);
    }

    public int hashCode()
    {
        return userIP.hashCode();
    }

    public Text getUserIP() {
        return userIP;
    }

    public Text getTimestamp() {
        return timestamp;
    }

    public Text getRequest() {
        return request;
    }

    public IntWritable getResponseSize() {
        return responseSize;
    }

    public IntWritable getStatus() {
        return status;
    }
}
```

# Write MR program - LogProcessorMap.java

```
package logprocess;
import java.io.IOException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class LogProcessorMap extends Mapper<Object, Text, Text, LogWritable > {
    LogWritable outValue = new LogWritable();
    Text outKey = new Text();

    public void map(Object key, Text value, Context context)
            throws IOException, InterruptedException {
        String logEntryPattern = "^(\\S+) (\\S+) (\\S+) \\[(\\w:/)+\\s[+\\-]\\d{4}\\] \"(.?)\" (\\d{3}) (\\d+)";
        Pattern p = Pattern.compile(logEntryPattern);
        Matcher matcher = p.matcher(value.toString());
        if (!matcher.matches()) {
            System.err.println("Bad Record : "+value);
            return;
        }

        String userIP = matcher.group(1);
        String timestamp = matcher.group(4);
        String request = matcher.group(5);
        int status = Integer.parseInt(matcher.group(6));
        int bytes = Integer.parseInt(matcher.group(7));

        outKey.set(userIP);
        outValue.set(userIP, timestamp, request,
                    bytes,status);
        context.write(outKey,outValue);
    }
}
```

## Write MR program - LogProcessorReduce.java (cont.)

```
package logprocess;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class LogProcessorReduce extends
        Reducer<Text,LogWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<LogWritable> values,
                      Context context) throws IOException, InterruptedException {
        int sum = 0;
        for (LogWritable val : values) {
            sum += val.getResponseSize().get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

# Write MR program - LogProcessor.java

```
package logprocess;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class LogProcessor extends Configured implements Tool {

    public static void main(String[] args) throws Exception {
        int res = ToolRunner.run(new Configuration(), new LogProcessor(), args);
        System.exit(res);
    }

    @Override
    public int run(String[] args) throws Exception {
        if (args.length < 3) {
            System.err.println("Usage: <input_path> <output_path> <num_reduce_tasks>");
            System.exit(-1);
        }
    }
}
```

# Write MR program - LogProcessor.java (cont.)

```
/* input parameters */
String inputPath = args[0];
String outputPath = args[1];
int numReduce = Integer.parseInt(args[2]);

Job job = Job.getInstance(getConf(), "log-analysis");

job.setJarByClass(LogProcessor.class);
job.setMapperClass(LogProcessorMap.class);
job.setReducerClass(LogProcessorReduce.class);
job.setNumReduceTasks(numReduce);

System.out.println("set output format");
job.setMapOutputKeyClass(Text.class);
job.setMapOutputValueClass(LogWritable.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

System.out.println("set input path");

FileInputFormat.setInputPaths(job, new Path(inputPath));
FileOutputFormat.setOutputPath(job, new Path(outputPath));

int exitStatus = job.waitForCompletion(true) ? 0 : 1;

return exitStatus;
```

}

# Compile, Package, Run MapReduce

```
mkdir mr_log_classes
```

```
javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d mr_log_classes ./*.java -Xlint
```

```
jar -cvf ./logprocessor.jar -C mr_log_classes/ .
```

```
yarn jar logprocessor.jar logprocess.LogProcessor /user/inputs/* /user/cloudera/logprocessor001 1
```

```
hdfs dfs -cat /user/cloudera/logprocessor001/part-r-00000
```

```
-bash-4.1$ hdfs dfs -cat /user/cloudera/logprocessor011/part-r-00000
199.120.110.21  9977
199.72.81.55    6245
burger.letters.com      0
dial22.lloyd.com       61716
ix-orl2-01.ix.netcom.com 3985
net-1-141.eden.com     34029
ppp-mia-30.shadow.net  14089
unicomp6.unicomp.net   46285
waters-gw.starway.net.au 6723
-bash-4.1$
```



# Apache Spark

อ.ดนัยรัฐ ธนาบดีธรรมจารี

Line ID: Danairat

FB: Danairat Thanabodithammachari

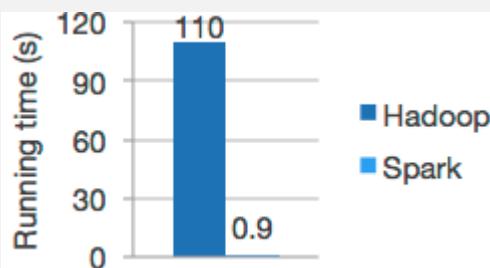
+668-1559-1446

# Apache Spark History

[Apache® Spark™](#) is a powerful open source processing engine built around speed, ease of use, and sophisticated analytics. It was originally developed at UC Berkeley in 2009. Since its release,



Apache Spark™ is a fast and general engine for large-scale data processing.



Internet powerhouses such as Yahoo, Baidu, and Tencent, have eagerly deployed Spark at massive scale of data on clusters of over 8,000 nodes



It has quickly become the largest open source community in big data, with over 1000 contributors from 250+ organizations.

<http://spark.apache.org/docs/>

# Who is Databricks?

Formed by the **creators of Apache Spark** and Shark, Databricks is working to greatly expand these open source projects and transform big data analysis in the process. We're deeply committed to keeping all work on these systems open source.

Databricks **provided a hosted service to run Spark**, Databricks Cloud, and partner to support Apache Spark with other Hadoop and big data companies.



Data Science made easy, from ingest to production. Powered by Apache<sup>®</sup> Spark.<sup>™</sup>

<https://databricks.com/>

# What is Spark?

Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including [Spark SQL](#) for SQL and structured data processing, [MLlib](#) for machine learning, [GraphX](#) for graph processing, and [Spark Streaming](#).

Spark  
SQL

Spark  
Streaming

MLlib  
(machine  
learning)

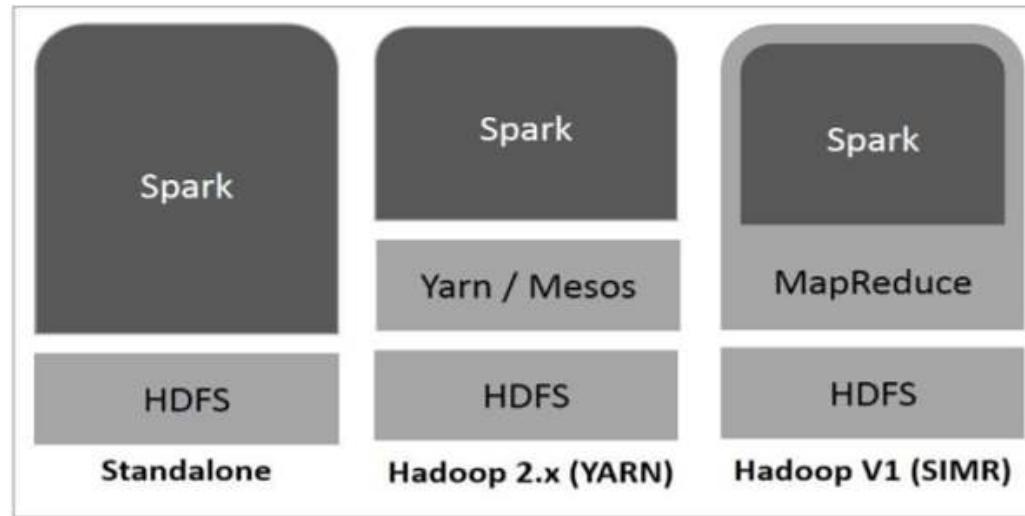
GraphX  
(graph)

Apache Spark

<http://spark.apache.org/docs/>

Danairat T.

# Spark and Hadoop Deployment



**Standalone** – Spark Standalone deployment means Spark occupies the place on top of HDFS(Hadoop Distributed File System) and space is allocated for HDFS, explicitly. Here, Spark and MapReduce will run side by side to cover all spark jobs on cluster.

**Hadoop Yarn** – Hadoop Yarn deployment means, simply, spark runs on Yarn without any pre-installation or root access required. It helps to integrate Spark into Hadoop ecosystem or Hadoop stack. It allows other components to run on top of stack.

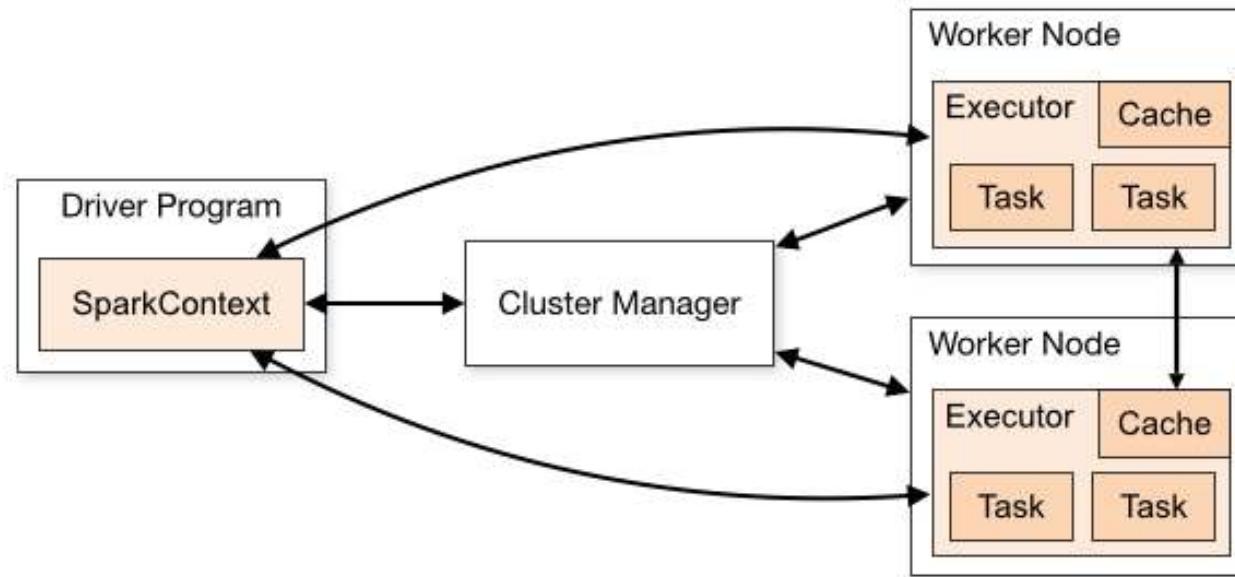
**Spark in MapReduce (SIMR)** – Spark in MapReduce is used to launch spark job in addition to standalone deployment. With SIMR, user can start Spark and uses its shell without any administrative access.

# Launching Spark on a Cluster

The Spark cluster mode overview explains the key concepts in running on a cluster. Spark can run both by itself, or over several existing cluster managers. It currently provides several options for deployment:

- Standalone Deploy Mode: simplest way to deploy Spark on a private cluster
- Apache Mesos
- Hadoop YARN

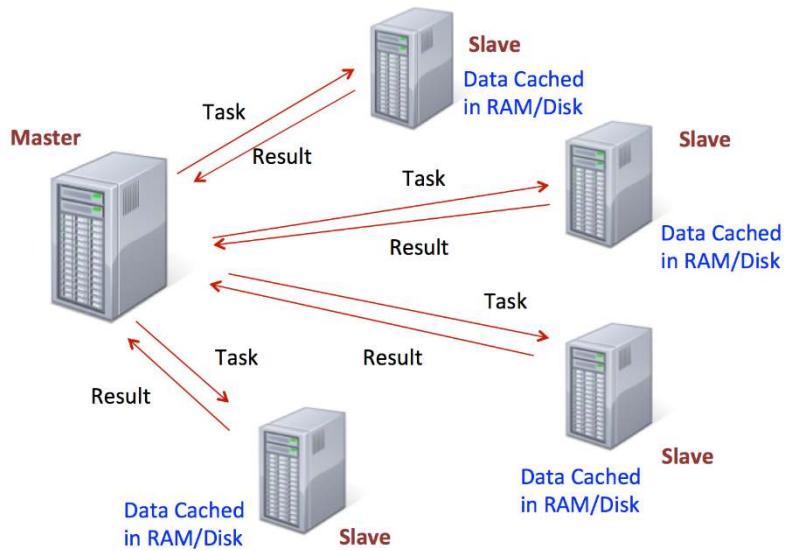
# Spark Cluster Mode Overview



Specifically, to run on a cluster, the `SparkContext` can connect to several types of *cluster managers* (either Spark's own standalone cluster manager, Mesos or YARN), which allocate resources across applications. Once connected, Spark acquires *executors* on nodes in the cluster, which are processes that run computations and store data for your application. Next, it sends your application code (defined by JAR or Python files passed to `SparkContext`) to the executors. Finally, `SparkContext` sends *tasks* to the executors to run.

# Spark 2.0 Programming

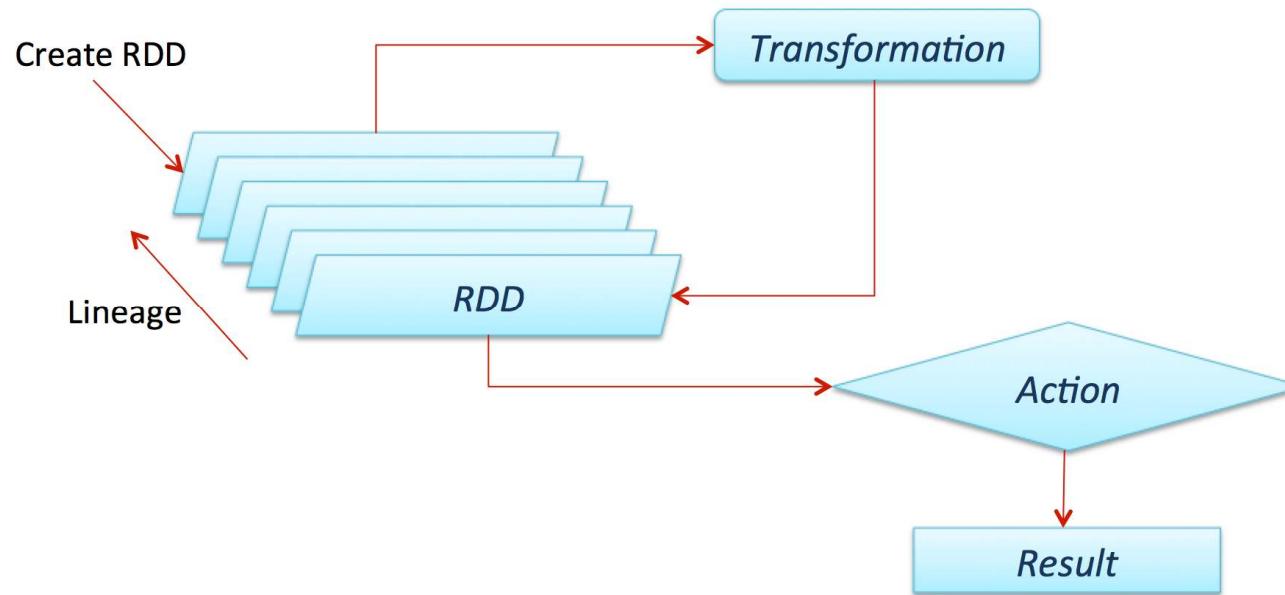
## How does Spark execute a job



A second abstraction in Spark is shared variables that can be used in parallel operations. By default, when Spark runs a function in parallel as a set of tasks on different nodes, it ships a copy of each variable used in the function to each task. Sometimes, a variable needs to be shared across tasks, or between tasks and the driver program. Spark supports two types of shared variables: broadcast variables, which can be used to cache a value in memory on all nodes, and accumulators, which are variables that are only “added” to, such as counters and sums.

# Spark 2.0 Programming

At a high level, every Spark application consists of a driver program that runs the user's main function and executes various parallel operations on a cluster. The main abstraction Spark provides is a resilient distributed dataset (RDD), which is **a collection of elements partitioned across the nodes of the cluster** that can be operated on in parallel. RDDs are created by starting with a file in the Hadoop file system (or others).. Finally, RDDs automatically recover from node failures.



# Transformations

The following table lists some of the common transformations supported by Spark.

Refer to the RDD API doc ([Scala](#), [Java](#), [Python](#), [R](#)) and pair RDD functions doc ([Scala](#), [Java](#)) for details.

Transformation	Meaning
<b>map(func)</b>	Return a new distributed dataset formed by passing each element of the source through a function func.
<b>filter(func)</b>	Return a new dataset formed by selecting those elements of the source on which func returns true.
<b>flatMap(func)</b>	Similar to map, but each input item can be mapped to 0 or more output items (so func should return a Seq rather than a single item).
<b>mapPartitions(func)</b>	Similar to map, but runs separately on each partition (block) of the RDD, so func must be of type Iterator<T> => Iterator<U> when running on an RDD of type T.
<b>mapPartitionsWithIndex(func)</b>	Similar to mapPartitions, but also provides func with an integer value representing the index of the partition, so func must be of type (Int, Iterator<T>) => Iterator<U> when running on an RDD of type T.
<b>sample(withReplacement, fraction, seed)</b>	Sample a fraction fraction of the data, with or without replacement, using a given random number generator seed.

# Transformations (cont.)

Transformation	Meaning
<b>union(otherDataset)</b>	Return a new dataset that contains the union of the elements in the source dataset and the argument.
<b>intersection(otherDataset)</b>	Return a new RDD that contains the intersection of elements in the source dataset and the argument.
<b>distinct([numTasks])</b>	Return a new dataset that contains the distinct elements of the source dataset.
<b>groupByKey([numTasks])</b>	<p>When called on a dataset of (K, V) pairs, returns a dataset of (K, Iterable&lt;V&gt;) pairs.</p> <p>Note: If you are grouping in order to perform an aggregation (such as a sum or average) over each key, using reduceByKey or aggregateByKey will yield much better performance.</p> <p>Note: By default, the level of parallelism in the output depends on the number of partitions of the parent RDD. You can pass an optional numTasks argument to set a different number of tasks.</p>
<b>reduceByKey(func, [numTasks])</b>	When called on a dataset of (K, V) pairs, returns a dataset of (K, V) pairs where the values for each key are aggregated using the given reduce function func, which must be of type (V,V) => V. Like in groupByKey, the number of reduce tasks is configurable through an optional second argument.
<b>aggregateByKey(zeroValue)(seqOp, combOp, [numTasks])</b>	When called on a dataset of (K, V) pairs, returns a dataset of (K, U) pairs where the values for each key are aggregated using the given combine functions and a neutral "zero" value. Allows an aggregated value type that is different than the input value type, while avoiding unnecessary allocations. Like in groupByKey, the number of reduce tasks is configurable through an optional second argument.
<b>sortByKey([ascending], [numTasks])</b>	When called on a dataset of (K, V) pairs where K implements Ordered, returns a dataset of (K, V) pairs sorted by keys in ascending or descending order, as specified in the boolean ascending argument.

# Transformations (cont.)

Transformation	Meaning
<b>join(otherDataset, [numTasks])</b>	When called on datasets of type (K, V) and (K, W), returns a dataset of (K, (V, W)) pairs with all pairs of elements for each key. Outer joins are supported through leftOuterJoin, rightOuterJoin, and fullOuterJoin.
<b>cogroup(otherDataset, [numTasks])</b>	When called on datasets of type (K, V) and (K, W), returns a dataset of (K, (Iterable<V>, Iterable<W>)) tuples. This operation is also called groupWith.
<b>cartesian(otherDataset)</b>	When called on datasets of types T and U, returns a dataset of (T, U) pairs (all pairs of elements).
<b>pipe(command, [envVars])</b>	Pipe each partition of the RDD through a shell command, e.g. a Perl or bash script. RDD elements are written to the process's stdin and lines output to its stdout are returned as an RDD of strings.
<b>coalesce(numPartitions)</b>	Decrease the number of partitions in the RDD to numPartitions. Useful for running operations more efficiently after filtering down a large dataset.
<b>repartition(numPartitions)</b>	Reshuffle the data in the RDD randomly to create either more or fewer partitions and balance it across them. This always shuffles all data over the network.
<b>repartitionAndSortWithinPartitions(partitioner)</b>	Repartition the RDD according to the given partitioner and, within each resulting partition, sort records by their keys. This is more efficient than calling repartition and then sorting within each partition because it can push the sorting down into the shuffle machinery.

# Actions

The following table lists some of the common actions supported by Spark. Refer to the RDD API doc ([Scala](#), [Java](#), [Python](#), [R](#)) and pair RDD functions doc ([Scala](#), [Java](#)) for details.

Action	Meaning
<b>reduce(func)</b>	Aggregate the elements of the dataset using a function func (which takes two arguments and returns one). The function should be commutative and associative so that it can be computed correctly in parallel.
<b>collect()</b>	Return all the elements of the dataset as an array at the driver program. This is usually useful after a filter or other operation that returns a sufficiently small subset of the data.
<b>count()</b>	Return the number of elements in the dataset.
<b>first()</b>	Return the first element of the dataset (similar to take(1)).
<b>take(n)</b>	Return an array with the first n elements of the dataset.
<b>takeSample(withReplacement, num, [seed])</b>	Return an array with a random sample of num elements of the dataset, with or without replacement, optionally pre-specifying a random number generator seed.
<b>takeOrdered(n, [ordering])</b>	Return the first n elements of the RDD using either their natural order or a custom comparator.

# Actions (Cont.)

Action	Meaning
<b>saveAsTextFile(path)</b>	Write the elements of the dataset as a text file (or set of text files) in a given directory in the local filesystem, HDFS or any other Hadoop-supported file system. Spark will call <code>toString</code> on each element to convert it to a line of text in the file.
<b>saveAsSequenceFile(path) (Java and Scala)</b>	Write the elements of the dataset as a Hadoop SequenceFile in a given path in the local filesystem, HDFS or any other Hadoop-supported file system. This is available on RDDs of key-value pairs that implement Hadoop's <code>Writable</code> interface. In Scala, it is also available on types that are implicitly convertible to <code>Writable</code> (Spark includes conversions for basic types like <code>Int</code> , <code>Double</code> , <code>String</code> , etc).
<b>saveAsObjectFile(path) (Java and Scala)</b>	Write the elements of the dataset in a simple format using Java serialization, which can then be loaded using <code>SparkContext.objectFile()</code> .
<b>countByKey()</b>	Only available on RDDs of type <code>(K, V)</code> . Returns a hashmap of <code>(K, Int)</code> pairs with the count of each key.
<b>foreach(func)</b>	Run a function <code>func</code> on each element of the dataset. This is usually done for side effects such as updating an <a href="#">Accumulator</a> or interacting with external storage systems. Note: modifying variables other than Accumulators outside of the <code>foreach()</code> may result in undefined behavior. See <a href="#">Understanding closures</a> for more details.

# 6. Installing Spark

```
$ tar -xvf spark-2.0.0-bin-hadoop2.7.tgz
```

```
user01@user01-VirtualBox: ~  
user01@user01-VirtualBox:~$ tar -xvf spark-2.0.0-bin-hadoop2.7.tgz
```

```
user01@user01-VirtualBox: ~  
spark-2.0.0-bin-hadoop2.7/jars/stax-api-1.0.1.jar  
spark-2.0.0-bin-hadoop2.7/jars/javax.annotation-api-1.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/curator-framework-2.6.0.jar  
spark-2.0.0-bin-hadoop2.7/jars/janino-2.7.8.jar  
spark-2.0.0-bin-hadoop2.7/jars/commons-io-2.4.jar  
spark-2.0.0-bin-hadoop2.7/jars/avro ipc-1.7.7.jar  
spark-2.0.0-bin-hadoop2.7/jars/hive-beeline-1.2.1.spark2.jar  
spark-2.0.0-bin-hadoop2.7/jars/hadoop-yarn-api-2.7.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/jackson-annotations-2.6.5.jar  
spark-2.0.0-bin-hadoop2.7/jars/jersey-media-jaxb-2.22.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/jackson-mapper-asl-1.9.13.jar  
spark-2.0.0-bin-hadoop2.7/jars/jackson-module-paranamer-2.6.5.jar  
spark-2.0.0-bin-hadoop2.7/jars/aopalliance-1.0.jar  
spark-2.0.0-bin-hadoop2.7/jars/super-csv-2.2.0.jar  
spark-2.0.0-bin-hadoop2.7/jars/antlr4-runtime-4.5.3.jar  
spark-2.0.0-bin-hadoop2.7/jars/jpm-1.1.jar  
spark-2.0.0-bin-hadoop2.7/jars/javassist-3.18.1-GA.jar  
spark-2.0.0-bin-hadoop2.7/jars/bcprov-jdk15on-1.51.jar  
spark-2.0.0-bin-hadoop2.7/jars/spark-hive-thriftserver_2.11-2.0.0.jar  
spark-2.0.0-bin-hadoop2.7/jars/javax.ws.rs-api-2.0.1.jar  
spark-2.0.0-bin-hadoop2.7/jars/hadoop-client-2.7.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/metrics-graphite-3.1.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/jackson-xc-1.9.13.jar  
spark-2.0.0-bin-hadoop2.7/jars/lz4-1.3.0.jar  
spark-2.0.0-bin-hadoop2.7/jars/core-1.1.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/antlr-2.7.7.jar  
spark-2.0.0-bin-hadoop2.7/jars/hadoop-annotations-2.7.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/hadoop-common-2.7.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/mx4j-3.0.2.jar  
spark-2.0.0-bin-hadoop2.7/jars/spark-launcher_2.11-2.0.0.jar  
spark-2.0.0-bin-hadoop2.7/jars/commons-logging-1.1.3.jar  
spark-2.0.0-bin-hadoop2.7/jars/commons-beanutils-core-1.8.0.jar  
spark-2.0.0-bin-hadoop2.7/jars/spark-core_2.11-2.0.0.jar
```

# 6. Installing Spark

```
$ sudo mv spark-2.0.0-bin-hadoop2.7 /usr/local/spark
```

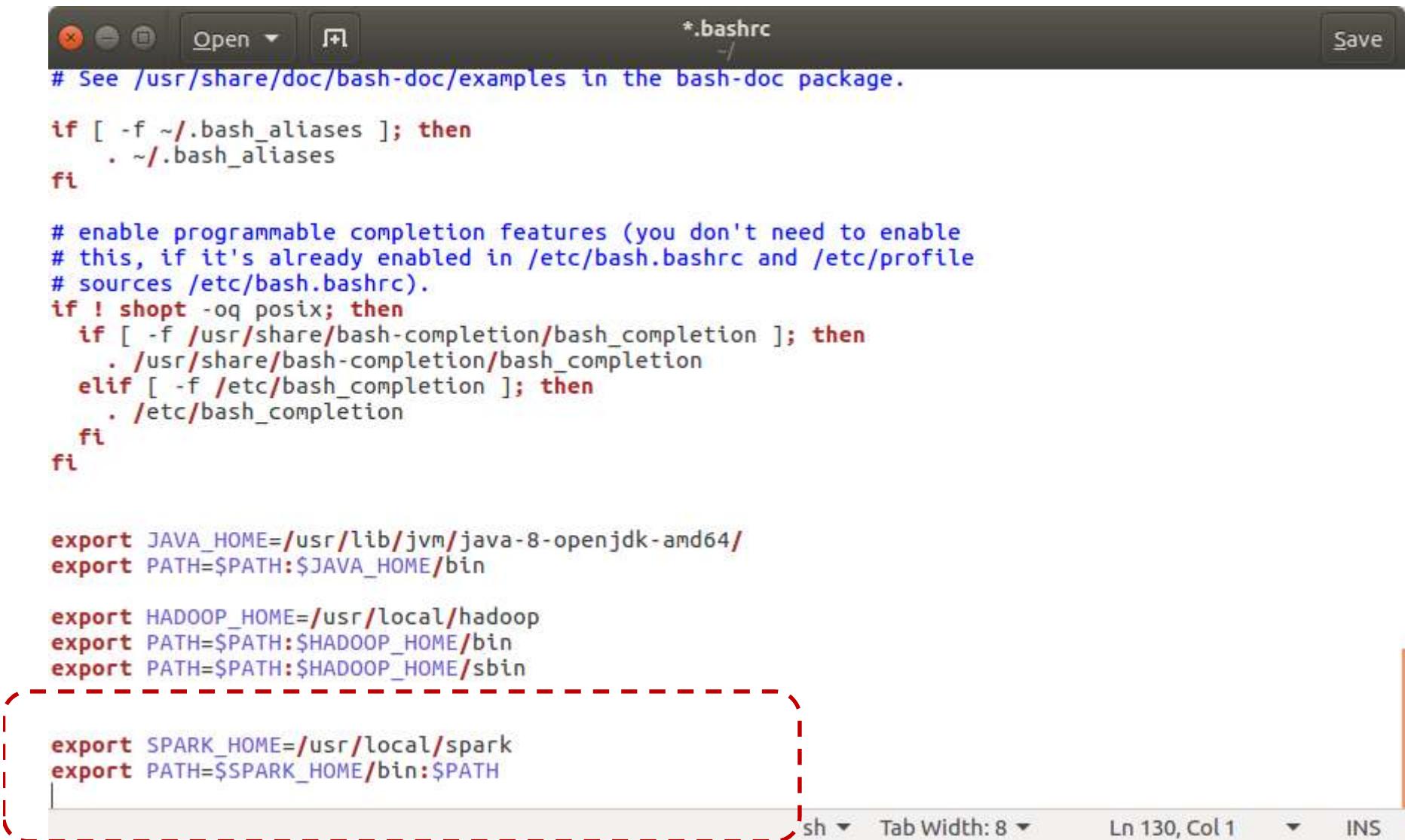


```
user01@user01-VirtualBox: ~
user01@user01-VirtualBox:~$ sudo mv spark-2.0.0-bin-hadoop2.7 /usr/local/spark
[sudo] password for user01: ■
```

```
$ cd
$ nano ~/.bashrc
```

```
...
export SPARK_HOME=/usr/local/spark
export PATH=$SPARK_HOME/bin:$PATH
```

# 6. Installing Spark



```
*.bashrc
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -q posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

export SPARK_HOME=/usr/local/spark
export PATH=$SPARK_HOME/bin:$PATH
```

# 6. Installing Spark

```
$source ~/.bashrc
```

```
$ cd /usr/local/spark/conf  
$ cp spark-env.sh.template spark-env.sh  
$ nano /usr/local/spark/conf/spark-env.sh
```

```
...
```

```
HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop/
```

*Notes: To be done in Master only, 5.In slaves file in /usr/local/spark/conf, add the slaves hostname [If slaves file is not found, copy slaves.template to slaves file]*

# 6. Installing Spark

The screenshot shows a terminal window titled "user01@user01-VirtualBox: /usr/local/spark/conf". The file being edited is "spark-env.sh". The terminal title bar also displays "GNU nano 2.5.3", "File: ./spark-env.sh", and "Modified". The text in the editor is the configuration script for Spark, detailing various environment variables and their descriptions. A red dashed box highlights the line "HADOOP\_CONF\_DIR=/usr/local/hadoop/etc/hadoop/".

```
# - SPARK_MASTER_PORT / SPARK_MASTER_WEBUI_PORT, to use non-default ports for the master
# - SPARK_MASTER_OPTS, to set config properties only for the master (e.g. "-Dx=y")
# - SPARK_WORKER_CORES, to set the number of cores to use on this machine
# - SPARK_WORKER_MEMORY, to set how much total memory workers have to give executors (e.g. 1000m$)
# - SPARK_WORKER_PORT / SPARK_WORKER_WEBUI_PORT, to use non-default ports for the worker
# - SPARK_WORKER_INSTANCES, to set the number of worker processes per node
# - SPARK_WORKER_DIR, to set the working directory of worker processes
# - SPARK_WORKER_OPTS, to set config properties only for the worker (e.g. "-Dx=y")
# - SPARK_DAEMON_MEMORY, to allocate to the master, worker and history server themselves (defaul$)
# - SPARK_HISTORY_OPTS, to set config properties only for the history server (e.g. "-Dx=y")
# - SPARK_SHUFFLE_OPTS, to set config properties only for the external shuffle service (e.g. "-D$)
# - SPARK_DAEMON_JAVA_OPTS, to set config properties for all daemons (e.g. "-Dx=y")
# - SPARK_PUBLIC_DNS, to set the public dns name of the master or workers

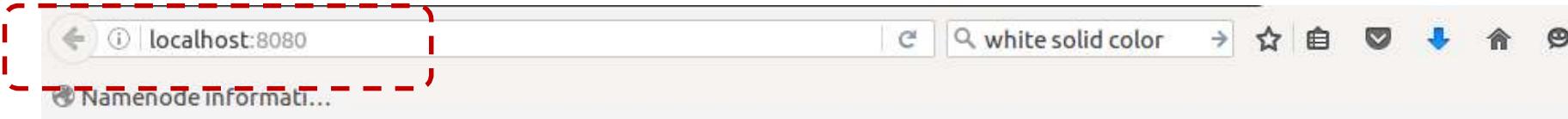
# Generic options for the daemons used in the standalone deploy mode
# - SPARK_CONF_DIR      Alternate conf dir. (Default: ${SPARK_HOME}/conf)
# - SPARK_LOG_DIR       Where log files are stored. (Default: ${SPARK_HOME}/logs)
# - SPARK_PID_DIR       Where the pid file is stored. (Default: /tmp)
# - SPARK_IDENT_STRING  A string representing this instance of spark. (Default: $USER)
# - SPARK_NICENESS      The scheduling priority for daemons. (Default: 0)

HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop/
```

^G Get Help    ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos  
^X Exit    ^R Read File    ^\ Replace    ^U Uncut Text    ^T To Linter    ^\_ Go To Line

# 6. Installing Spark

```
$ /usr/local/spark/sbin/start-all.sh  
http://localhost:8080
```



The screenshot shows a web browser displaying the Apache Spark master interface. The address bar shows "localhost:8080". Below it, a link "Namenode information..." is highlighted with a red dashed box. The main content area has a header "Spark Master at spark://user01-VirtualBox:7077". It provides various metrics: URL (spark://user01-VirtualBox:7077), REST URL (spark://user01-VirtualBox:6066 (cluster mode)), Alive Workers (1), Cores in use (1 Total, 0 Used), Memory in use (1986.0 MB Total, 0.0 B Used), Applications (0 Running, 0 Completed), Drivers (0 Running, 0 Completed), and Status (ALIVE). The "Workers" section lists one worker with ID "worker-20160828131158-10.0.2.15-44692", Address "10.0.2.15:44692", State "ALIVE", Cores "1 (0 Used)", and Memory "1986.0 MB (0.0 B Used)". The "Running Applications" section is currently empty.

Apache Spark 2.0.0

## Spark Master at spark://user01-VirtualBox:7077

URL: spark://user01-VirtualBox:7077  
REST URL: spark://user01-VirtualBox:6066 (cluster mode)

Alive Workers: 1

Cores in use: 1 Total, 0 Used

Memory in use: 1986.0 MB Total, 0.0 B Used

Applications: 0 [Running](#), 0 [Completed](#)

Drivers: 0 Running, 0 Completed

Status: ALIVE

### Workers

Worker Id	Address	State	Cores	Memory
worker-20160828131158-10.0.2.15-44692	10.0.2.15:44692	ALIVE	1 (0 Used)	1986.0 MB (0.0 B Used)

### Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration

# 7. Test Submit Application to Spark

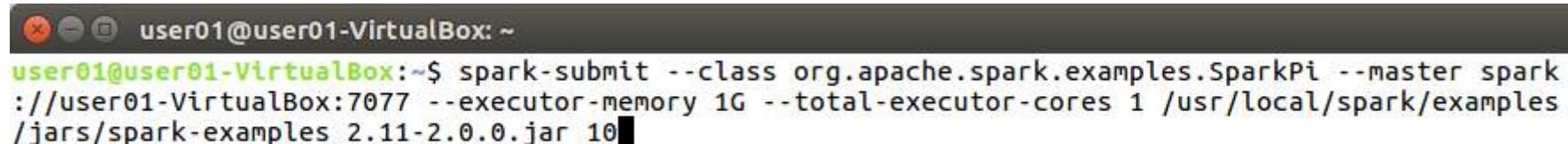
```
$ spark-submit --class org.apache.spark.examples.SparkPi --master spark://user01-VirtualBox:7077 --executor-memory 1G --total-executor-cores 1 /usr/local/spark/examples/jars/spark-examples_2.11-2.0.0.jar 10
```

–class: The entry point for your application.

–master: The master URL for the cluster.

–executor-memory: Specify memory to be allocated for the application

–total-executor-cores: Specify no of cpu cores to be allocated for the application



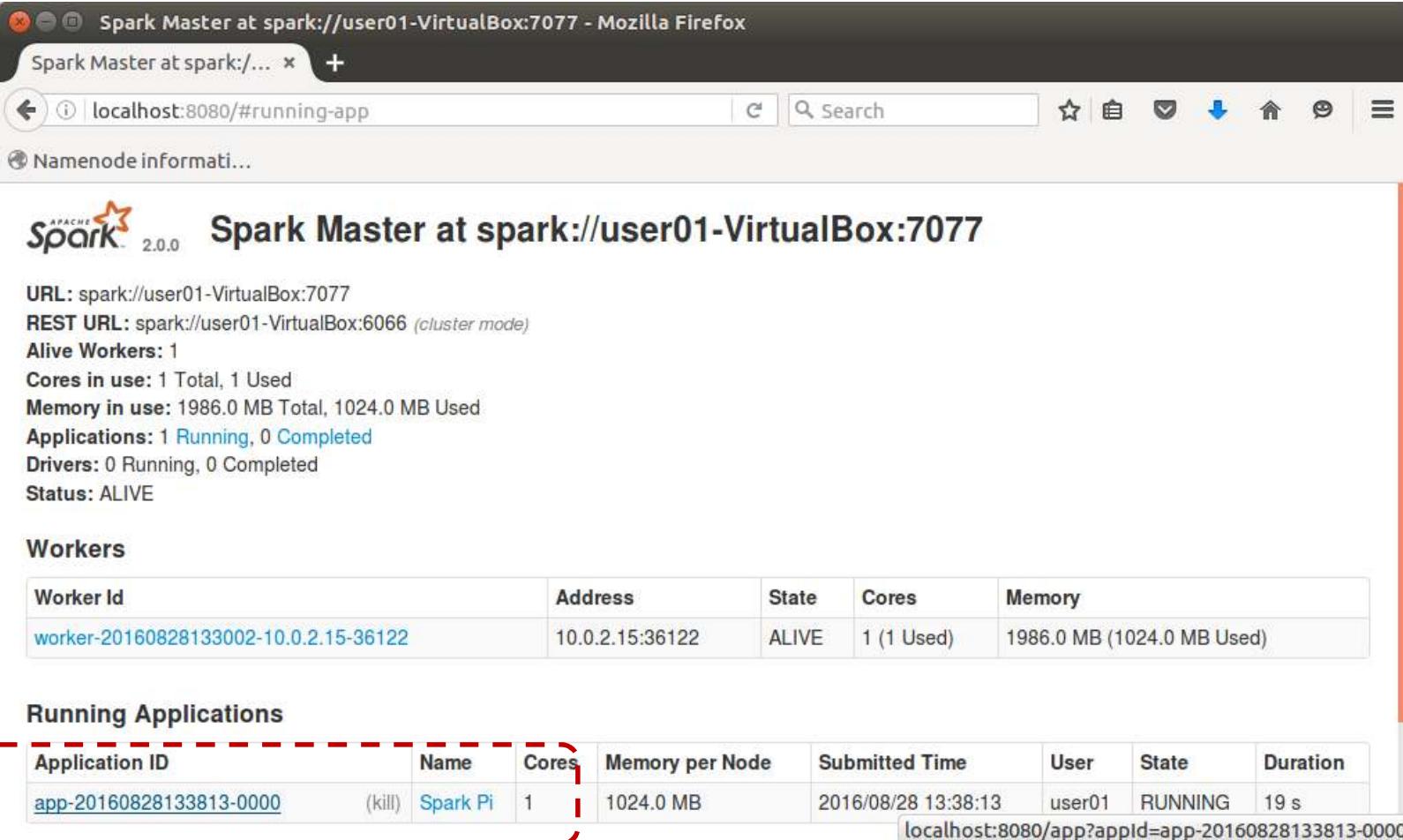
```
user01@user01-VirtualBox:~$ spark-submit --class org.apache.spark.examples.SparkPi --master spark://user01-VirtualBox:7077 --executor-memory 1G --total-executor-cores 1 /usr/local/spark/examples/jars/spark-examples_2.11-2.0.0.jar 10
```

# 7. Test Submit Application to Spark

```
user01@user01-VirtualBox: ~
orange/rdd,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@64beebb7{/st
orange/rdd/json,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@7813cb11{/en
vironment,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@bcec031{/env
ironment/json,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@21005f6c{/ex
ecutors,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@32f0fba8{/ex
ecutors/json,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@545de5a4{/ex
ecutors/threadDump,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@29ef6856{/ex
ecutors/threadDump/json,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@ab7a938{/sta
tic,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@3faf2e7d{/n
ull,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@4648ce9{/api
,null,AVAILABLE}
16/08/28 13:38:11 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@569bf9eb{/st
ages/stage/kill,null,AVAILABLE}
16/08/28 13:38:12 INFO server.ServerConnector: Started ServerConnector@4d9c69e2{HTTP/1.1}{0.0.0.0
:4040}
16/08/28 13:38:12 INFO server.Server: Started @5635ms
16/08/28 13:38:12 INFO util.Utils: Successfully started service 'SparkUI' on port 4040.
16/08/28 13:38:12 INFO ui.SparkUI: Bound SparkUI to 0.0.0.0, and started at http://10.0.2.15:4040
16/08/28 13:38:12 INFO spark.SparkContext: Added JAR file:/usr/local/spark/examples/jars/spark-ex
amples_2.11-2.0.0.jar at spark://10.0.2.15:43959/jars/spark-examples_2.11-2.0.0.jar with timestamp
p 1472366292200
16/08/28 13:38:12 INFO client.StandaloneAppClient$ClientEndpoint: Connecting to master spark://us
er01-VirtualBox:7077...
```

# 7. Test Submit Application to Spark

http://10.0.2.15:8080/



Spark Master at spark://user01-VirtualBox:7077 - Mozilla Firefox  
Spark Master at spark://... +  
localhost:8080/#running-app Search  
Namenode information

**Apache Spark 2.0.0** Spark Master at spark://user01-VirtualBox:7077

**URL:** spark://user01-VirtualBox:7077  
**REST URL:** spark://user01-VirtualBox:6066 (*cluster mode*)  
**Alive Workers:** 1  
**Cores in use:** 1 Total, 1 Used  
**Memory in use:** 1986.0 MB Total, 1024.0 MB Used  
**Applications:** 1 [Running](#), 0 [Completed](#)  
**Drivers:** 0 Running, 0 Completed  
**Status:** ALIVE

**Workers**

Worker Id	Address	State	Cores	Memory
worker-20160828133002-10.0.2.15-36122	10.0.2.15:36122	ALIVE	1 (1 Used)	1986.0 MB (1024.0 MB Used)

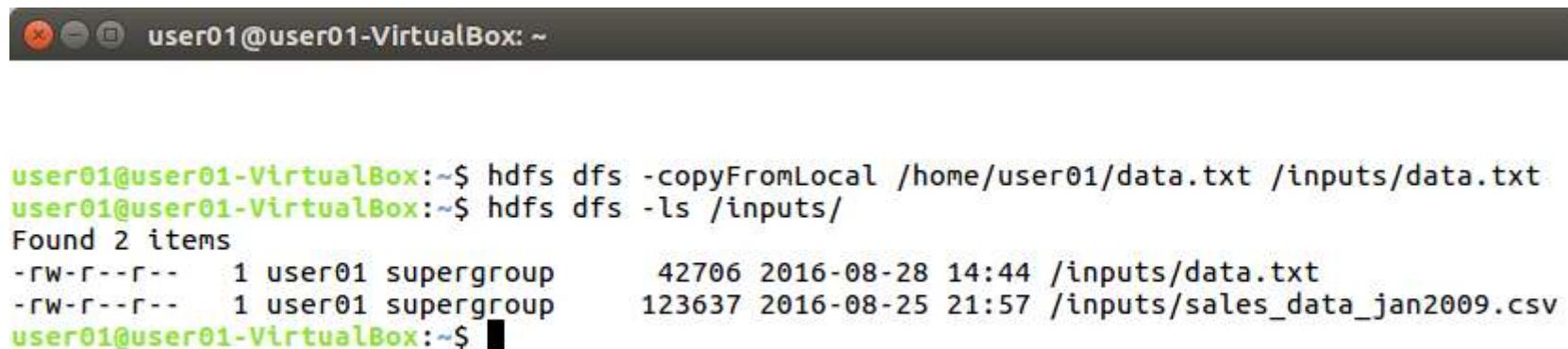
**Running Applications**

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
<a href="#">app-20160828133813-0000</a>	(kill) Spark Pi	1	1024.0 MB	2016/08/28 13:38:13	user01	RUNNING	19 s

localhost:8080/app?appId=app-20160828133813-0000

# 8. Test Core Spark RDD on HDFS Data

```
$ hdfs dfs -copyFromLocal /home/user01/data.txt /inputs/data.txt  
$ hdfs dfs -ls /inputs/
```



A screenshot of a terminal window titled "user01@user01-VirtualBox: ~". The window contains the following command-line session:

```
user01@user01-VirtualBox:~$ hdfs dfs -copyFromLocal /home/user01/data.txt /inputs/data.txt  
user01@user01-VirtualBox:~$ hdfs dfs -ls /inputs/  
Found 2 items  
-rw-r--r--    1 user01 supergroup      42706 2016-08-28 14:44 /inputs/data.txt  
-rw-r--r--    1 user01 supergroup    123637 2016-08-25 21:57 /inputs/sales_data_jan2009.csv  
user01@user01-VirtualBox:~$ █
```

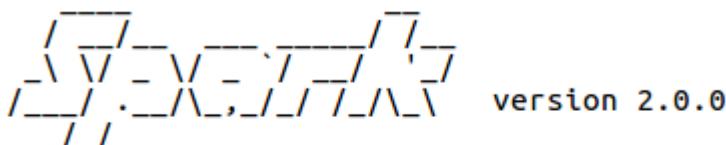
## 8. Test Core Spark RDD on HDFS Data

```
$ spark-shell spark://10.0.2.15:7077/
```

# 8. Test Core Spark RDD on HDFS Data

## WordCount Processing

```
scala> var hFile = sc.textFile("hdfs://10.0.2.15:9000/inputs/data.txt")
scala> val wc = hFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
```



```
Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_91)
Type in expressions to have them evaluated.
Type :help for more information.
```

```
scala> var hFile = sc.textFile("hdfs://10.0.2.15:9000/inputs/data.txt")
hFile: org.apache.spark.rdd.RDD[String] = hdfs://10.0.2.15:9000/inputs/data.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> val wc = hFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
wc: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:26

scala> ■
```

Note: sc is the object of SparkContext

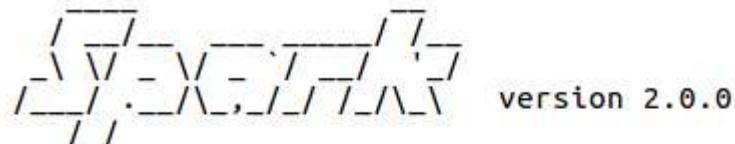
# 8. Test Core Spark RDD on HDFS Data

display first 10 results

```
scala> wc.take(10)
```

```
scala>
```

```
wc.saveAsTextFile("hdfs://10.0.2.15:9000/outputs/spark_output_dir001")
scala> :q
```



```
Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_91)
```

```
Type in expressions to have them evaluated.
```

```
Type :help for more information.
```

```
scala> var hFile = sc.textFile("hdfs://10.0.2.15:9000/inputs/data.txt")
hFile: org.apache.spark.rdd.RDD[String] = hdfs://10.0.2.15:9000/inputs/data.txt MapPartitionsRDD[1] at textFile at <console>:24
```

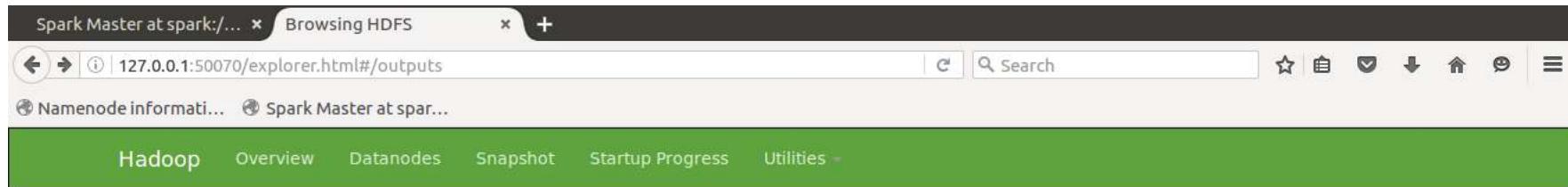
```
scala> val wc = hFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
wc: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:26
```

```
scala> wc.take(10)
res0: Array[(String, Int)] = Array((hippopotamus-sized,1), (favour,1), (1881,,1), (someone,1), (young,2), (surgery,1), (House,2), (Glyptodon,,1), (ship,,1), (urging,1))
```

```
scala> wc.saveAsTextFile("hdfs://10.0.2.15:9000/outputs/spark_output_dir001")
```

```
scala> ■
```

# 8. Test Core Spark RDD on HDFS Data



## Browse Directory

/outputs

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	user01	supergroup	0 B	28/8/2559 15:03:17	0	0 B	<a href="#">spark_output_dir001</a>
drwxr-xr-x	user01	supergroup	0 B	25/8/2559 22:18:40	0	0 B	<a href="#">wordcount_output_dir001</a>
drwxr-xr-x	user01	supergroup	0 B	25/8/2559 22:41:37	0	0 B	<a href="#">wordcount_output_dir002</a>

Hadoop, 2015.

# 8. Test Core Spark RDD on HDFS Data

Spark Master at spark:/... x Browsing HDFS x +

127.0.0.1:50070/explorer.html#/outputs/spark\_output\_dir001

Namenode information Spark Master at spar...

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

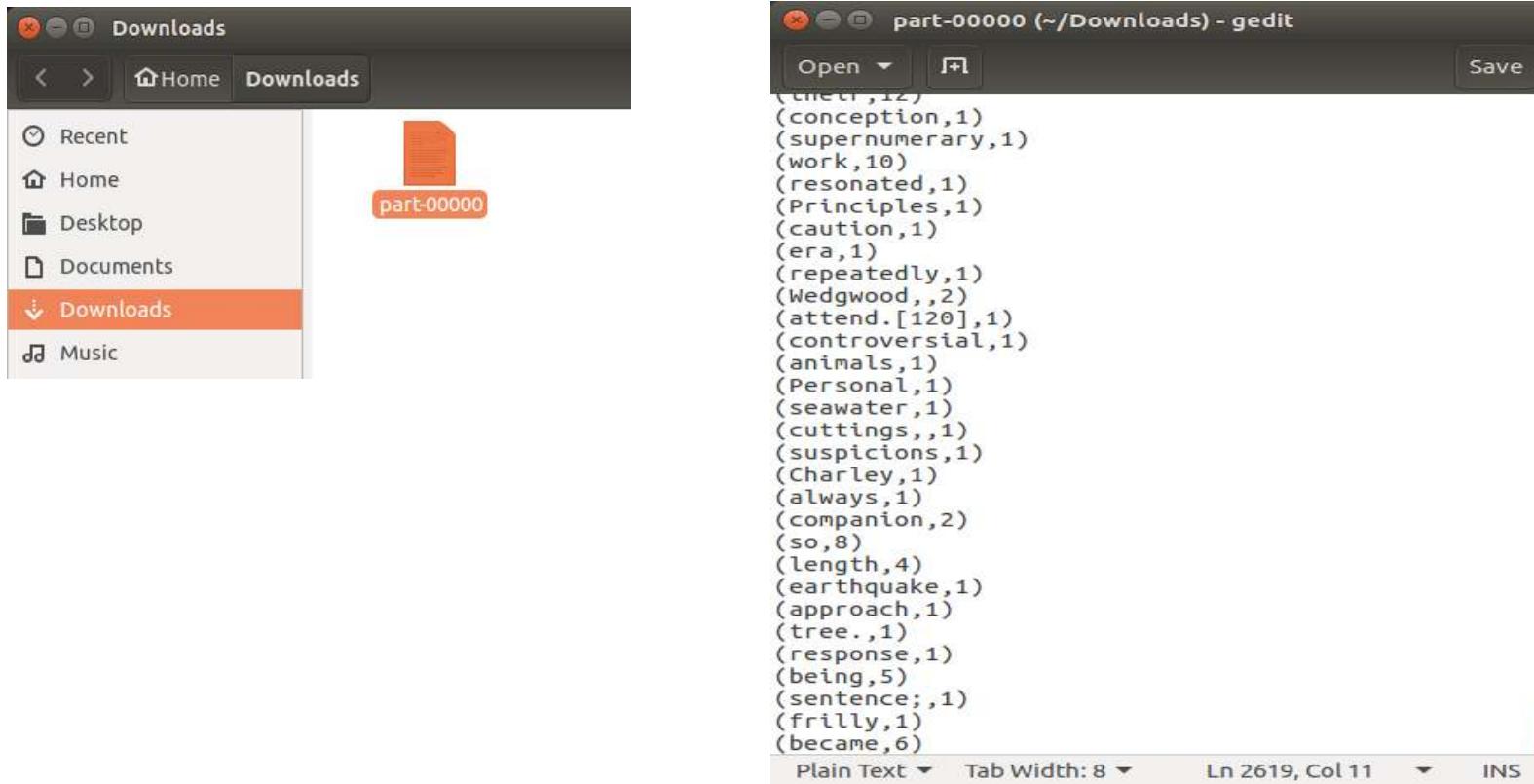
## Browse Directory

/outputs/spark\_output\_dir001 Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rW-r--r--	user01	supergroup	0 B	28/8/2559 15:03:17	1	128 MB	_SUCCESS
-rw-r--r--	user01	supergroup	31.83 KB	28/8/2559 15:03:17	1	128 MB	<a href="#">part-00000</a>

Hadoop, 2015.

# 8. Test Core Spark RDD on HDFS Data



```
$ hdfs dfs -ls /outputs/spark_output_dir001/
```

```
$ hdfs dfs -ls /outputs/spark_output_dir001/
```

## 8. Test Core Spark RDD on HDFS Data

 version 2.0.0

Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0\_91)  
Type in expressions to have them evaluated.  
Type :help for more information.

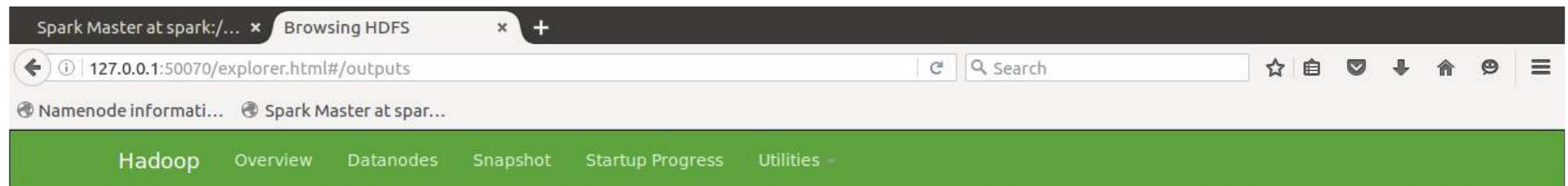
```
scala> var hFile = sc.textFile("hdfs://10.0.2.15:9000/inputs/*")
hFile: org.apache.spark.rdd.RDD[String] = hdfs://10.0.2.15:9000/inputs/* MapPartitionsRDD[1] at t
extFile at <console>:24

scala> val wc = hFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
wc: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:26

scala> wc.take(10)
res0: Array[(String, Int)] = Array((someone,1), (6:24,2/23/09,1), (House,3), (8:23,Product1,1200,
Mastercard,Sarah,Floyds,1), (11:35,39.96111,-82.99889,1), (20:11,1/13/09,1), (order,2), (7:06,47.
45,18.9666667,1), (18:48,40.5945,-74.6244,1), (pigeon,1))

scala> wc.saveAsTextFile("hdfs://10.0.2.15:9000/outputs/spark_output_dir002")
scala>
```

# 8. Test Core Spark RDD on HDFS Data



The screenshot shows a web browser window titled "Spark Master at spark:/... x Browsing HDFS x +". The address bar contains "127.0.0.1:50070/explorer.html#/outputs". The page header includes links for "Namenode informati...", "Spark Master at spar...", "Hadoop", "Overview", "Datanodes", "Snapshot", "Startup Progress", and "Utilities". Below the header is a search bar and a toolbar with icons for refresh, search, and navigation.

## Browse Directory

/outputs Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	user01	supergroup	0 B	28/8/2559 15:03:17	0	0 B	<a href="#">spark_output_dir001</a>
drwxr-xr-x	user01	supergroup	0 B	28/8/2559 15:23:23	0	0 B	<a href="#">spark_output_dir002</a>
drwxr-xr-x	user01	supergroup	0 B	25/8/2559 22:18:40	0	0 B	<a href="#">wordcount_output_dir001</a>
drwxr-xr-x	user01	supergroup	0 B	25/8/2559 22:41:37	0	0 B	<a href="#">wordcount_output_dir002</a>

Hadoop, 2015.

# 8. Test Core Spark RDD on HDFS Data

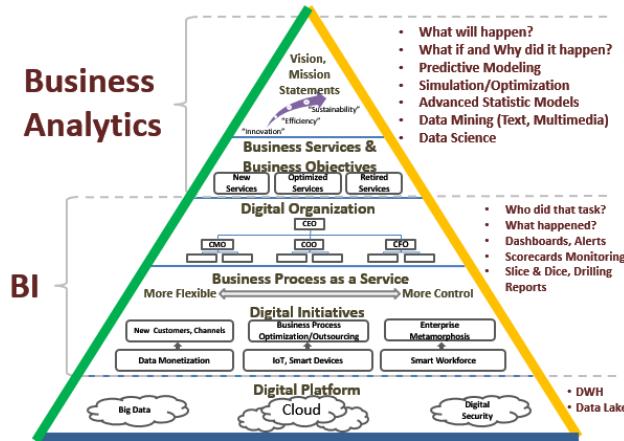
The screenshot shows a web-based HDFS browser interface. The title bar says "Spark Master at spark:/... x Browsing HDFS x +". The address bar shows the URL "127.0.0.1:50070/explorer.html#/outputs/spark\_output\_dir002". Below the address bar are links for "Namenode informati..." and "Spark Master at spar...". The main menu bar has a green background with white text, containing links for "Hadoop", "Overview", "Datanodes", "Snapshot", "Startup Progress", and "Utilities".

## Browse Directory

/outputs/spark\_output\_dir002

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	user01	supergroup	0 B	28/8/2559 15:23:23	1	128 MB	SUCCESS
-rw-r--r--	user01	supergroup	71.4 KB	28/8/2559 15:23:23	1	128 MB	part-00000
-rw-r--r--	user01	supergroup	70.6 KB	28/8/2559 15:23:23	1	128 MB	part-00001

Hadoop, 2015.



# Business Analytics vs. Business Intelligence

อ.ดนัยรัตน์ ธนบดีธรรมจารี

Line ID: Danairat

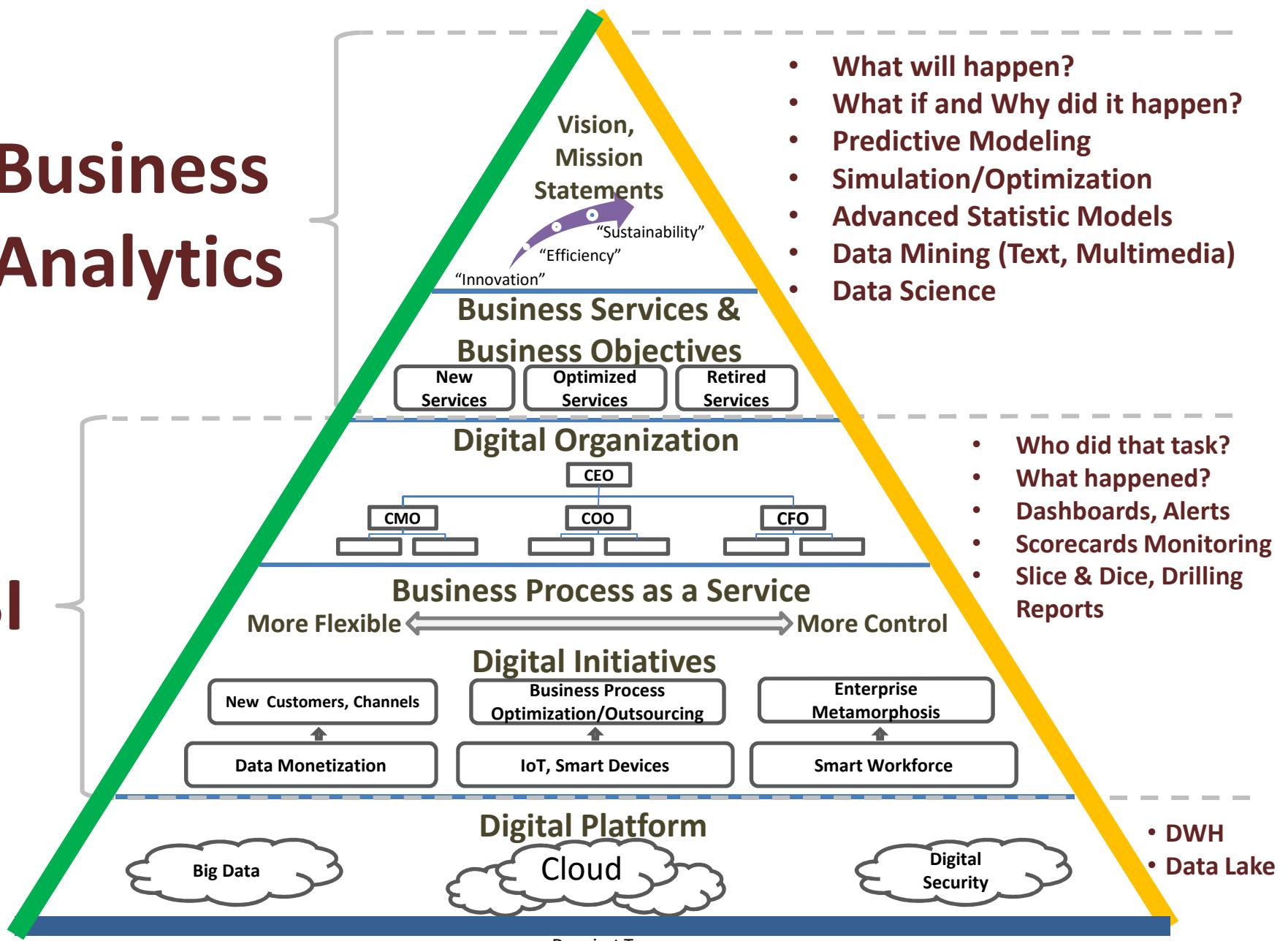
FB: Danairat Thanabodithammachari

+668-1559-1446

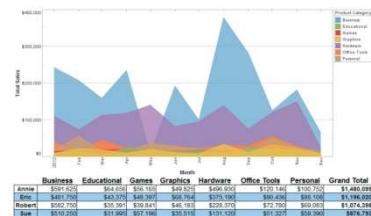
# Business Analytics and Business Intelligence

## Business Analytics

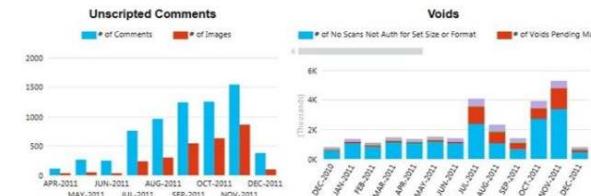
BI



# What is Business Intelligence (BI)?



Performance Dashboard

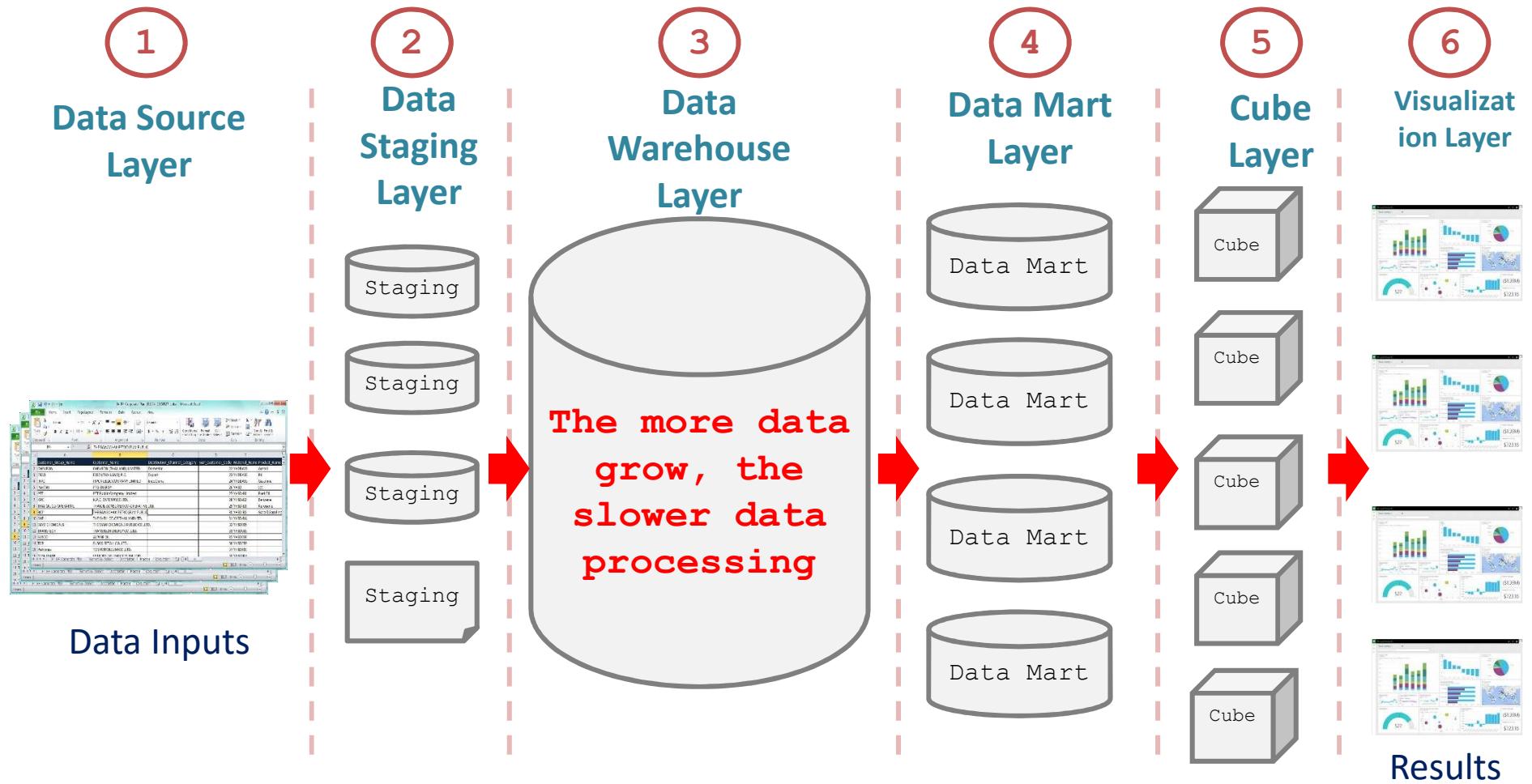


Business intelligence systems are used to maintain, optimize and streamline current operations. BI improves and maintains operational efficiency and helps businesses **increase organizational productivity**. Business intelligence software confers many benefits, notably **powerful reporting and data analysis** capabilities. Using BI's rich visualization mechanisms, managers are able to generate intuitive, readable reports that contain relevant, actionable data.

*Popular business intelligence solutions include; SAP BusinessObjects, QlikView, IBM Cognos, Microstrategy, etc.*

<https://selecthub.com/business-intelligence/business-intelligence-vs-business-analytics/>

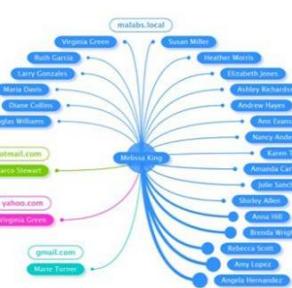
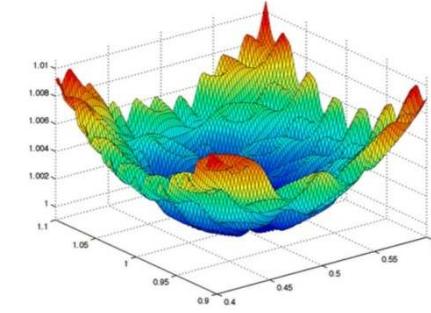
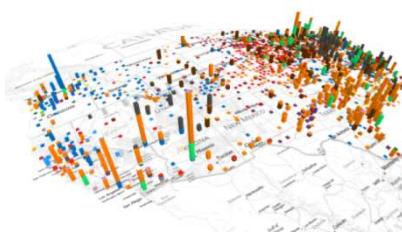
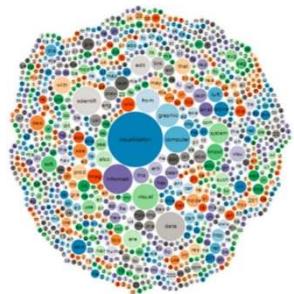
# Traditional Data Warehouse



## Top Concerns from Traditional Data Warehouse Architecture

1. A lot of data duplication lead to cost of data store/storage issue
2. Very slow of data processing and need to restart/roll back the job if any failed
3. Data security issue due to keep data too many copies and various formats

# What is Business Analytics (BA)?



Like business intelligence, BA collects and analyzes data, employs **predictive analytics** and generates **richly visualized reports**, helping identify and address an organization's weak points. That's where similarities end. Business analytics software is used to explore and analyze historical and current data. It utilizes **statistical analysis, data mining** and quantitative analysis to identify past business trends.

*Popular business analytics solutions include; SAP Business Analytics Suite, Pentaho BA, Birst BI and Tableau Blg Data Analytics.*

<https://selecthub.com/business-intelligence/business-intelligence-vs-business-analytics/>

# Data Lake

## HOW DO DATA LAKES WORK?

The concept can be compared to a water body, a lake, where water flows in, filling up a reservoir and flows out.

### STRUCTURED DATA

1. Information in rows and columns
2. Easily ordered and processed with data mining tools

1

The incoming flow represents multiple raw data archives ranging from emails, spreadsheets, social media content, etc.



### UNSTRUCTURED DATA

1. Raw, unorganized data
2. Emails
3. PDF files
4. Images, video and audio
5. Social media tools

2

The reservoir of water is a dataset, where you run analytics on all the data.



3

The outflow of water is the analyzed data.

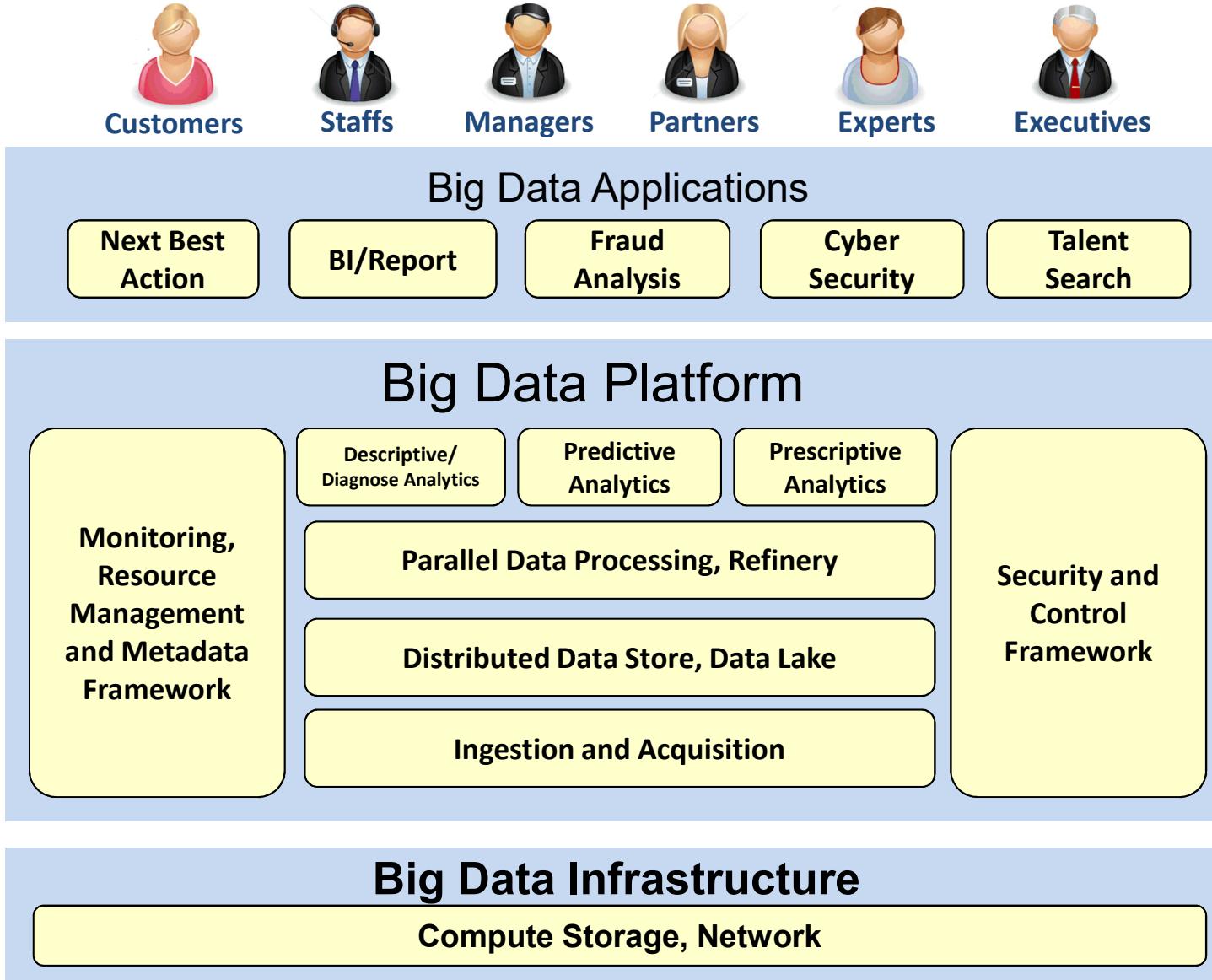
4

Through this process, you are able to "swim" through all the data quickly to gain key business insights.

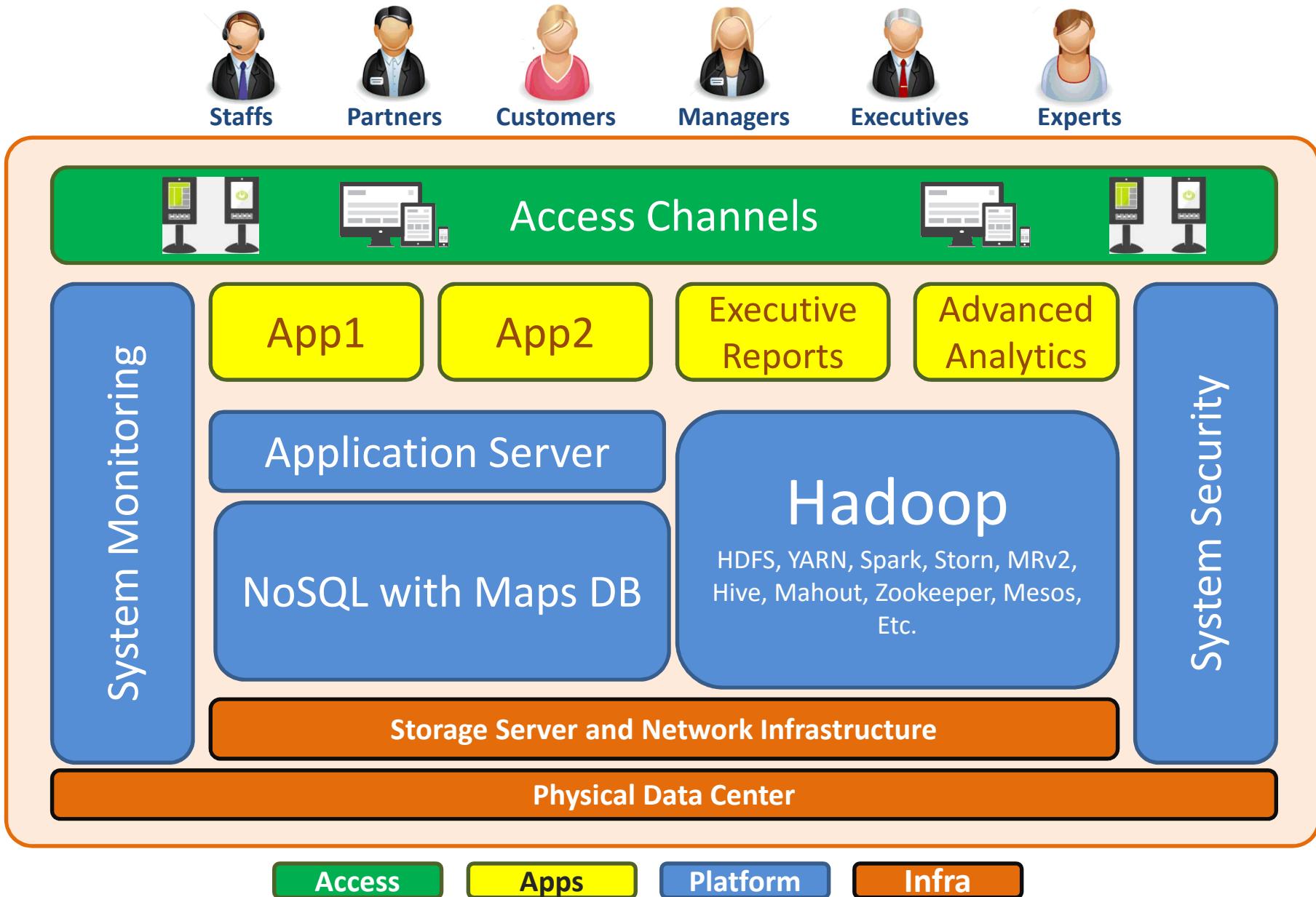


[www.digitalnewsasia.com](http://www.digitalnewsasia.com)

# Big Data for Business Analytics Platform

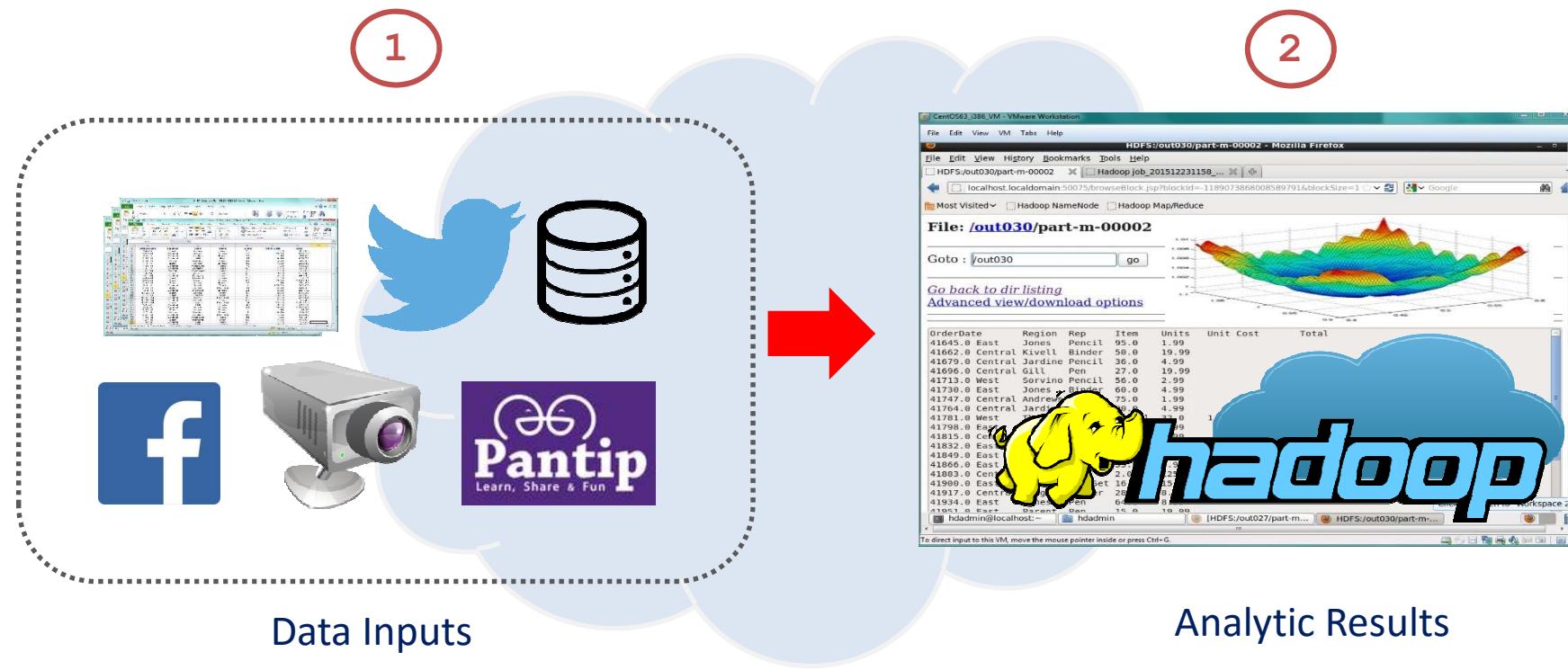


# Example of Big Data Deployment Architecture



# Core Hadoop processing

NO data staging transformation and NO data move required!!

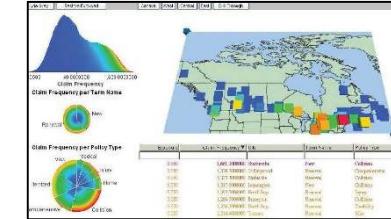
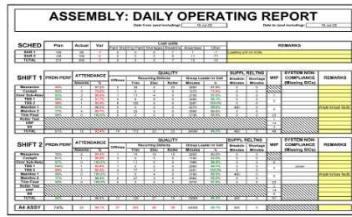


## Top Benefits

1. Cloud and IoT ready architecture roadmap
2. No data duplication with reduce cost of data store/storage
3. Fast data processing and all processing are built-in fault tolerant
4. Align with unify data architecture and data governance
5. Less steps of data processing comparing with traditional DWH

## The Effort Investment:-

1. Learn core Hadoop



# Big Data Analytics

## ວ.ດນຍຮ້ສ ຮນບດීතຣມຈາກී

## Line ID: Danairat

**FB: Danairat Thanabodithammachari**

+668-1559-1446

# Descriptive/Diagnostic analytics

Descriptive/Diagnostic analytics answers the question, "**What happened in the business?**" It looks at data and information to describe the current business situation in a way that trends, patterns and exceptions become apparent. This takes the form of reports, dashboards, MIS, etc.

ASSEMBLY: DAILY OPERATING REPORT										
			Lost units			REMARKS				
SCHED	Plan	Actual	Var	Plant Stop	Plant Shortages	Breakdown	Awareness	Other		
SHR 1	208	198	-7	0	0	2	1	-1		
SHR 2	108	108	0	0	0	0	11	-11		
<b>TOTAL</b>	<b>316</b>	<b>206</b>	<b>-11</b>	<b>0</b>	<b>0</b>	<b>7</b>	<b>12</b>	<b>-12</b>		

SHIFT 1	PRDN PERF	ATTENDANCE		QUALITY			SUPPL RELTNS		WIP	SYSTEM NON-COMPLIANCE (Missing SICs)	REMARKS
		Officers	Absents %	Recurring Defects	Group Leader in Cell	Minutes	%	Breakin			
Machine 1	98%	0	99.4%	1	0	0	0	2910	7145	0	
Cocktail	99%	0	99.4%	1	1	4	0	2769	8145	0	
Door Sub-Assy	99%	0	99.4%	1	0	0	0	3656	9515	0	
TBS 1	99%	0	99.4%	1	0	0	0	3217	8830	0	
TBS 2	99%	1	99.8%	0	102	5	0	3217	10830	0	
Machine 2	99%	1	99.7%	0	0	0	0	4276	8830	400	
Machine 3	99%	1	99.7%	0	0	0	0	3056	9515	0	
Machine 4	99%	1	99.7%	0	0	0	0	3056	9515	0	
Total Final	99%	0	99.4%	0	0	0	0	2715	8475	0	
Racker Test										1	
HHP										14	
BS										0	
<b>TOTAL</b>	<b>99%</b>	<b>15</b>	<b>99.4%</b>	<b>14</b>	<b>170</b>	<b>23</b>	<b>23</b>	<b>24290</b>	<b>8475</b>	<b>400</b>	<b>0</b>

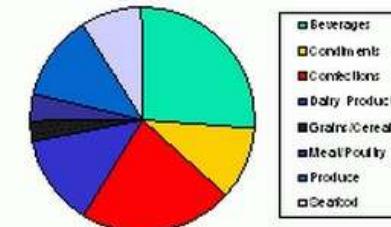
SHIFT 2	PRDN PERF	ATTENDANCE		QUALITY			SUPPL RELTNS		WIP	SYSTEM NON-COMPLIANCE (Missing SICs)	REMARKS
		Officers	Absents %	Recurring Defects	Group Leader in Cell	Minutes	%	Breakin			
Machine 1	99%	0	99.4%	1	0	0	0	2910	8015	0	
Cocktail	99%	0	99.4%	3	0	2	0	2769	9145	0	
Door Sub-Assy	99%	0	99.4%	1	11	3	0	3656	9515	0	
TBS 1	99%	1	99.4%	1	0	2	0	2656	8615	0	
TBS 2	99%	2	99.4%	4	17	3	0	3411	10830	0	
Machine 2	99%	0	99.4%	1	27	4	0	2656	9515	400	
Machine 3	99%	1	99.4%	0	27	3	0	2656	9515	0	
Total Final	99%	0	99.4%	0	0	0	0	2019	8745	0	
Racker Test										1	
HHP										14	
BS										0	
<b>TOTAL</b>	<b>99%</b>	<b>7</b>	<b>99.4%</b>	<b>13</b>	<b>120</b>	<b>27</b>	<b>18</b>	<b>18688</b>	<b>8615</b>	<b>400</b>	<b>0</b>

<b>A4 ASSY</b>	<b>74%</b>	<b>32</b>	<b>94.7%</b>	<b>27</b>	<b>301</b>	<b>38</b>	<b>43088</b>	<b>95.1%</b>	<b>920</b>	<b>0</b>
----------------	------------	-----------	--------------	-----------	------------	-----------	--------------	--------------	------------	----------

Sales by Categories

Category Name	Quantity	Amount
Beverages	925	\$27,761.57
Condiments	378	\$10,773.27
Confections	890	\$2,877.18
Dairy Products	581	\$13,685.32
Grains/Cereals	189	\$3,325.40
Meat/Poultry	92	\$4,083.66
Produce	351	\$13,031.20
Seafood	669	\$9,316.54
<b>Total</b>	<b>4,065</b>	<b>\$104,854.14</b>



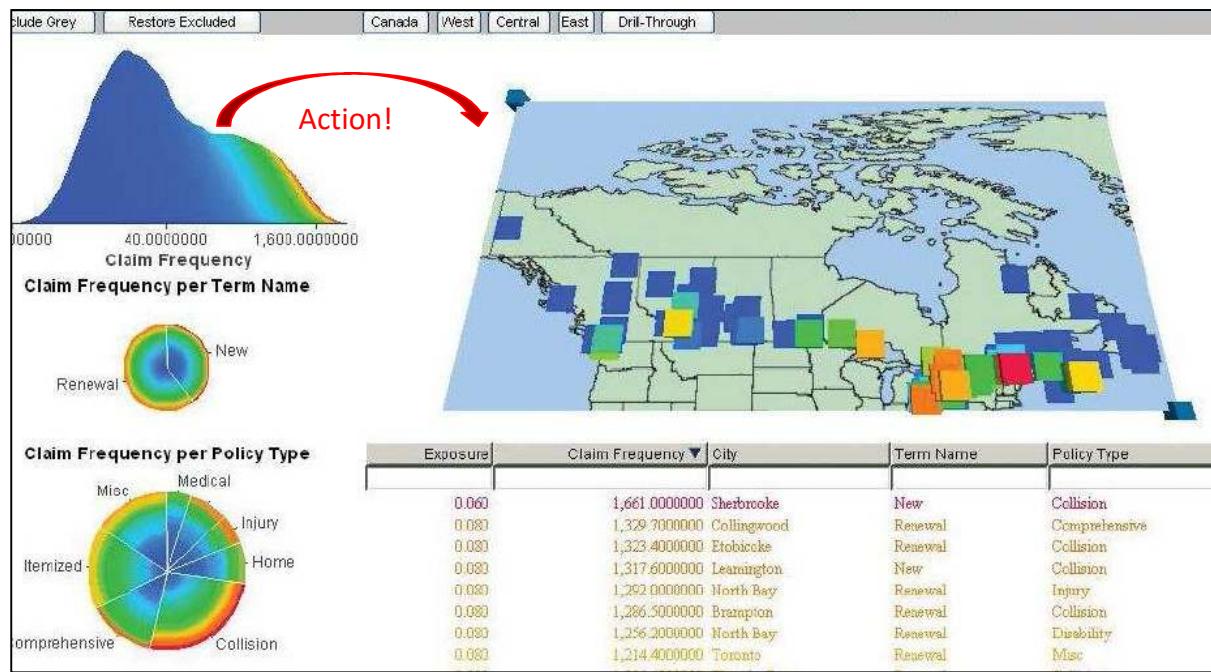
# Predictive analytics

Predictive analytics answers the question, "What is likely to happen in the future?" Here data modeling and forecasting are used to determine future possibilities



# Prescriptive analytics

Prescriptive analytics is the combination of the above to provide answers to the "So what?" and the "Now what?" For example, **what should a business do** to retain key customers? How can businesses improve their supply chain to enhance service levels while reducing costs?





- |  |   |   |   |
|--|---|---|---|
| <ol style="list-style-type: none"> <li>1. Identify Targeted Users</li> <li>2. Identify Target Opportunities / key Measurements</li> <li>3. Identify Data Sources/Types</li> <li>4. Identify Data Capturing Approaches</li> <li>5. Identify Data Processing and Visualization Planning</li> <li>6. Identify Big Data Platform</li> <li>7. Identify Security</li> <li>8. Identify Governance and Change Control for Operations</li> <li>9. Identify Team Structure</li> <li>10. Identify Phasing, Budget and Cost</li> </ol> | <ol style="list-style-type: none"> <li>1. Develop Use Cases</li> <li>2. Develop Requirements Definition</li> <li>3. Develop Big Data Solution Framework</li> <li>4. Develop the Development and Test Environment</li> <li>5. Develop Data Capture</li> <li>6. Develop Analytics</li> <li>7. Integrate Visualization</li> <li>8. Manage Assets and Configurations</li> </ol> | <ol style="list-style-type: none"> <li>1. Monitor Big Data Platform Availability, Utilization and Capacity Planning</li> <li>2. Manage Operation Tasks (Admin. Scripts, Commands), Data Capturing System, Upgrading or Patching</li> <li>3. Manage Service Requests and Incidents</li> <li>4. System admin. Training</li> <li>5. Helpdesk Training</li> <li>6. End-User Training (Analytics Results)</li> </ol> | <ol style="list-style-type: none"> <li>1. Adoption Rates for each analytics results</li> <li>2. No. of Missing Analytic Results</li> <li>3. No. of Missing Data</li> <li>4. Lost hours per month</li> <li>5. Avg. of each Analytic Result Response Time</li> <li>6. No. of Technology System Failure per month</li> <li>7. Evaluate Activity Conformance with Policies</li> </ol> |
|--|---|---|---|

# Big Data Project Life Cycle Management

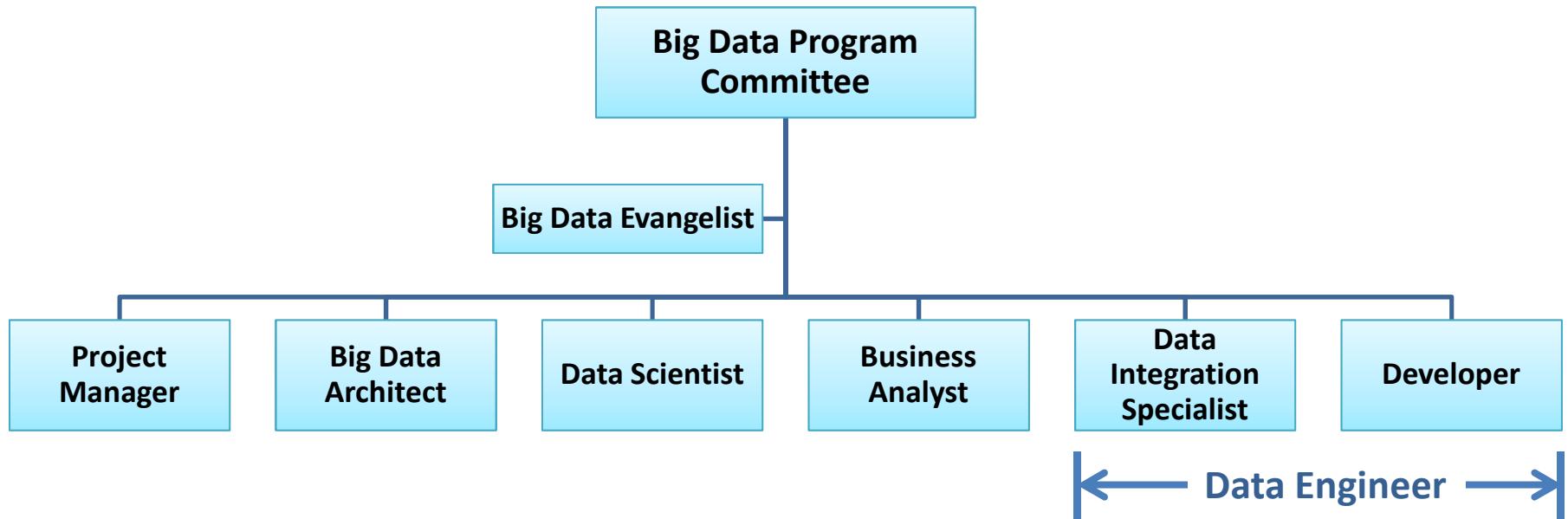
อ.ดเนียรัฐ ธนาบดีธรรมจารี

Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

# Big Data People and Team Structure



# Big Data Team Structure

No.	Roles	Description
1	<b>Big Data Program Committee</b>	The Team to develop Big Data initiative and deliver solution results
2	<b>Big Data Evangelist</b>	The business evangelist must have a combination of good communication and presentation skills and deep contextual business knowledge, as well as a clear understanding of technology in general and big data techniques.
3	<b>Project Manager</b>	The project manager “owns” the development schedule and is expected to ensure that the right architects, designers, and developers are brought into the project at the right times.
4	<b>Big Data Architect</b>	The person who has background in parallel and distributed computing architecture. This person is knowledgeable about fundamental performance “gotchas” that will impede the speed, scalability, and extensibility of any application requiring massive data volumes.

# Big Data Team Structure

No.	Roles	Description
5	<b>Data Scientist</b>	The data scientist combines knowledge of computer science, the use of high-performance applications, and statistics, economics, mathematics, and probabilistic analysis skills.
6	<b>Business Analyst</b>	The person who engages with the business process owners and solicits their needs and expectations. Business analysts who are able to effectively translate business expectations into specific data analysis results.
7	<b>Data Integration Specialist</b>	The person who has experience in data extraction, transformation, loading, and data transformations in preparation for cleansing and delivery to target systems. Seek people with experience with data federation and virtualization, data quality, and metadata analysis.
8	<b>Application Developer</b>	The technical resources with the right set of skills for programming and testing parallel and distributed applications.

# Big Data Project Life Cycle



- |   |   |  |  |
|---|---|--|--|
| 1. Identify Targeted Users                              | 1. Develop Use Cases                            | 1. Monitor Big Data Platform Availability, Utilization and Capacity Planning                       | 1. Adoption Rates for each analytics results   |
| 2. Identify Target Opportunities / Key Measurements     | 2. Develop Requirements Definition              | 2. Manage Operation Tasks (Admin. Scripts, Commands), Data Capturing System, Upgrading or Patching | 2. No. of Missing Analytic Results             |
| 3. Identify Data Sources/Types                          | 3. Develop Big Data Solution Framework          | 3. Manage Service Requests and Incidents   | 3. No. of Missing Data                         |
| 4. Identify Data Capturing Approaches                   | 4. Develop the Development and Test Environment | 4. System admin. Training  | 4. Lost hours per month                        |
| 5. Identify Data Processing and Visualization Planning  | 5. Develop Data Capture                         | 5. Helpdesk Training   | 5. Avg. of each Analytic Result Response Time  |
| 6. Identify Big Data Platform                           | 6. Develop Analytics                            | 6. End-User Training (Analytic Results)  | 6. No. of Technology System Failure per month  |
| 7. Identify Security                                    | 7. Integrate Visualization                      |  | 7. Evaluate Activity Conformance with Policies |
| 8. Identify Governance and Change Control for Operation | 8. Manage Assets and Configurations             |  |  |
| 9. Identify Team Structure                              |   |  |  |
| 10. Identify Phasing, Budget and Cost                   |   |  |  |

# Key Activities, People and Deliverables

No.	Phases	Activities	People	Deliverables
1	Planning	Identify Targeted Users	Big Data Program Committee	Big Data Discovery Worksheet
2	Planning	Identify Target Opportunities	Big Data Program Committee	Big Data Discovery Worksheet
3	Planning	Identify Team Structure	Big Data Program Committee	Team Organization Chart
4	Planning	Identify Data Sources/Types	Big Data Architect, Data Scientist, Data Integration Specialist	Data Sources Types Information
5	Planning	Identify Data Capturing Approaches	Data Integration Specialist, Data Scientist	Data Capturing Information
6	Planning	Identify Data Processing and Visualization Planning	Business Analyst, Big Data Architect, Data Scientist, Developer	Data Processing Framework and User Interface Summary
7	Planning	Identify Big Data Platform	Big Data Architect, Project Manager	Big Data Platform Summary
8	Planning	Identify Security	Corporate Information Security, Big Data Architect, Project Manager	Security Scope Summary
9	Planning	Identify Governance and Change Control for Operation	Internal Control Team, Corporate Information Security, Big Data Architect, Project Manager	Governance, RACI, Change Procedures Summary
10	Planning	Identify Phasing Budget and Cost	CIO, CFO, Project Manager, Business Analyst	Project Investment Summary

# Key Activities, People and Deliverables

No.	Phases	Activities	People	Deliverables
1	Development	Develop Use Cases	Business Users, Business Analyst, Big Data Architect, Big Data Evangelist	Use Cases Summary
2	Development	Develop Requirements Definition	Business Users, Business Analyst, Big Data Architect	Requirements Summary
3	Development	Develop Big Data Solution Framework	Big Data Architect	Solution Component Diagram
4	Development	Develop the Development and Test Environment	Big Data Architect, Data Integration Specialist, Developer	Development and Test Environment
5	Development	Develop Data Capture	Data Integration Specialist, Developer	Data Capturing Module
6	Development	Develop Analytics	Data Integration Specialist, Developer	Data Analytic Module
7	Development	Integrate Visualization	Data Integration Specialist, Developer	User Interface and Visualization Results
8	Development	Manage Assets and Configurations	Project Manager, Big Data Architect, Corporate Information Security, Head of IT Operation	Assets Inventory and Configurations Change Procedure

Agile Methodology may apply in Big Data Development Phase.

# Key Activities, People and Deliverables

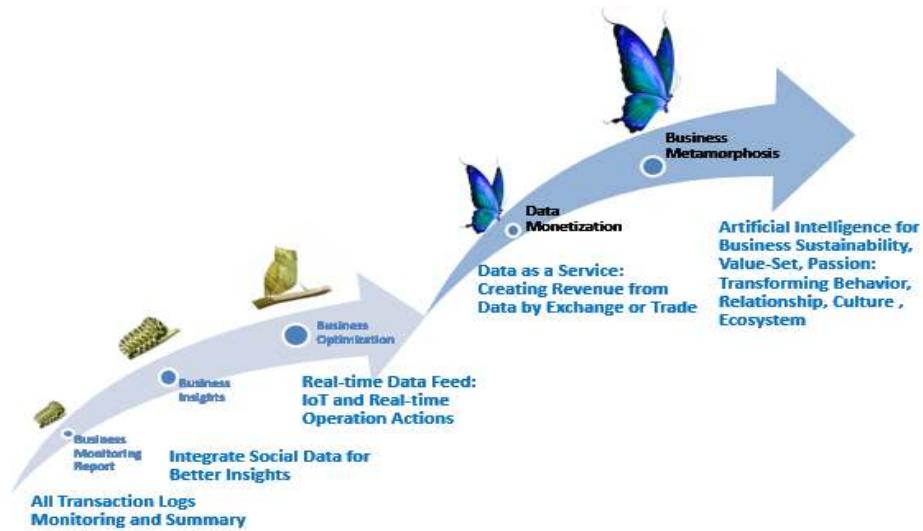
No.	Phases	Activities	People	Deliverables
1	Operation and Support	Monitor Big Data Platform Availability, Utilization and Capacity Planning	IT Operation Team	Availability, Utilization and Capacity Planning Report
2	Operation and Support	Manage Operation Tasks (Admin. Scripts, Commands), Data Capturing System, Upgrading or Patching	IT Operation Team, Big Data Architect	Schedule or Ad-Hoc Operation Activities
3	Operation and Support	Manage Service Requests and Incidents	IT Operation Team	Service Requests and Incidents Procedures
4	Operation and Support	System Administration Training	Big Data Architect, Data Integration Specialist, Developer, IT Administration, IT Operation	System Administration and Operation Training Activity
5	Operation and Support	Helpdesk Training	IT Administration, IT Operation, IT Helpdesk	Helpdesk Training Activity
6	Operation and Support	End-User Training (Analytic Results)	Business Analyst, Business Users	End-User Training Activity

# Key Activities, People and Deliverables

No.	Phases	Activities	People	Deliverables
1	Evaluation	Create Adoption Rates for each analytics Results	IT Operation	Percent of user adoption
2	Evaluation	Create No. of Missing Analytic Results	Big Data Program Committee	No. of Missing Analytics Report
3	Evaluation	Create No. of Missing Data Results	Big Data Program Committee	No. of Missing Data Report
4	Evaluation	Create Lost hours per month Results	Big Data Architect, Data Scientist, Data Integration Specialist	Lost hours per month Report
5	Evaluation	Create Avg. of each Analytic Processing and Response Time Results	Data Integration Specialist, Data Scientist	Analytic Processing and Response Time Report
6	Evaluation	Create No. of Technology System Failure per month Results	Business Analyst, Big Data Architect, Data Scientist, Developer	Technology System Failure per month Report
7	Evaluation	Evaluate Activity Conformance with Policies	Big Data Architect, Project Manager	Change Control Log Report

# Big Data Governance Worksheet

No.	Master/ Transactional/ Summary Data	Data Name	Owner	Used by Critical Business Service (Y/N)	Volume (MB/GB/TB)	Varieties of Types (Text, XML, JSON, Image, Sound, VDO, etc.)	Velocity (How fast data change in minutes)	Change Control (Y/N), Change Procedure	Current Issues



# Big Data Maturity Model

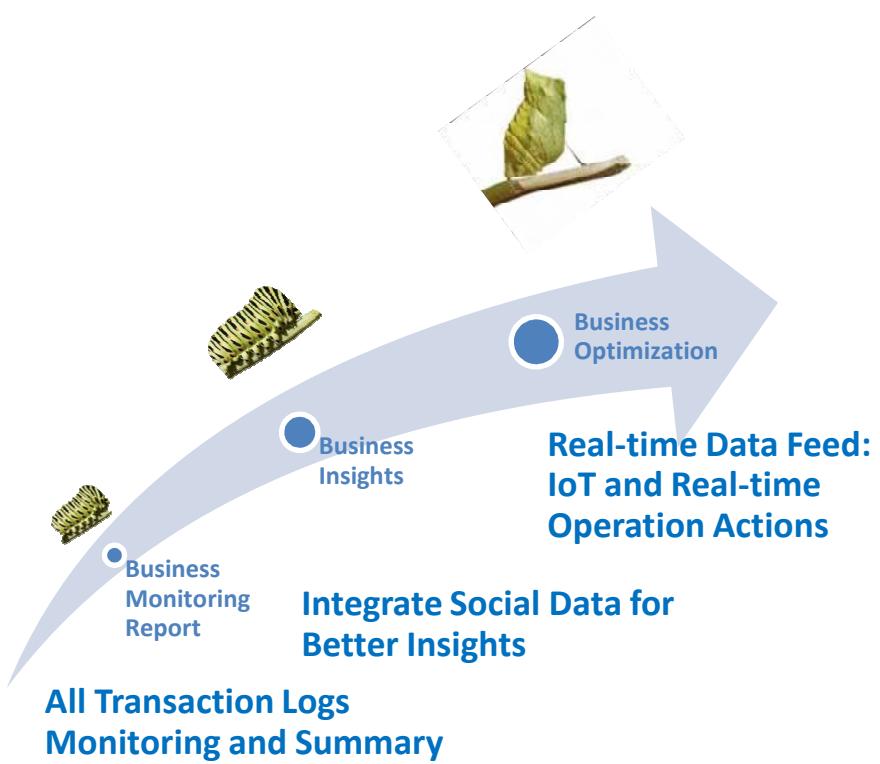
อ.ดนัยรัตน์ ธนบดีธรรมจารี

Line ID: Danairat

FB: Danairat Thanabodithammachari

+668-1559-1446

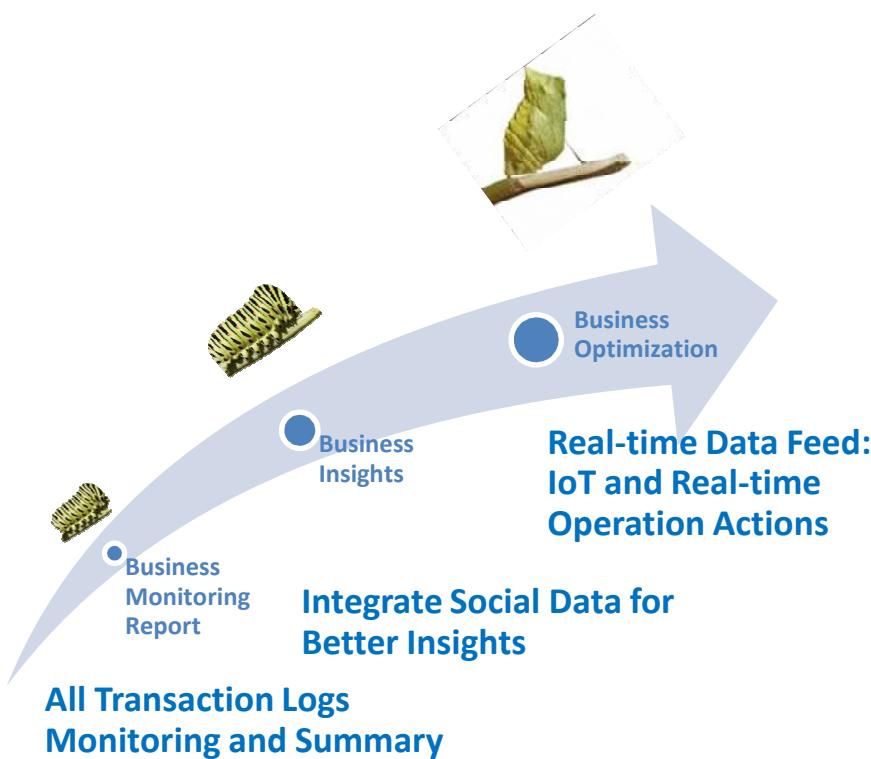
# Big Data Maturity Model



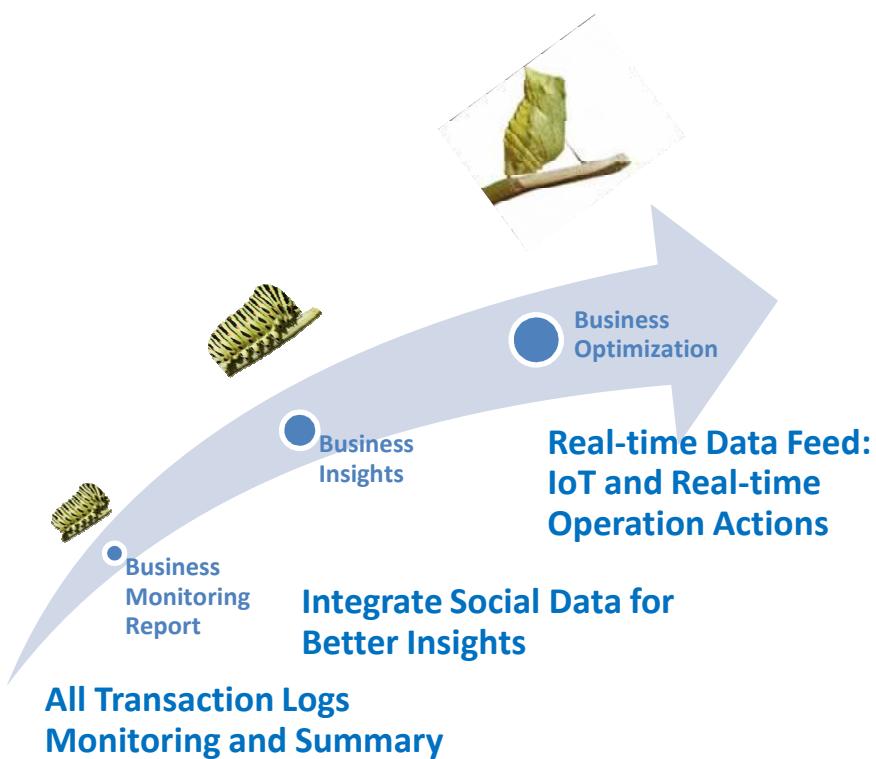
# Big Data Maturity Model

Business Monitoring Report

Mine all the transactional data at the lowest levels of detail



# Big Data Maturity Model



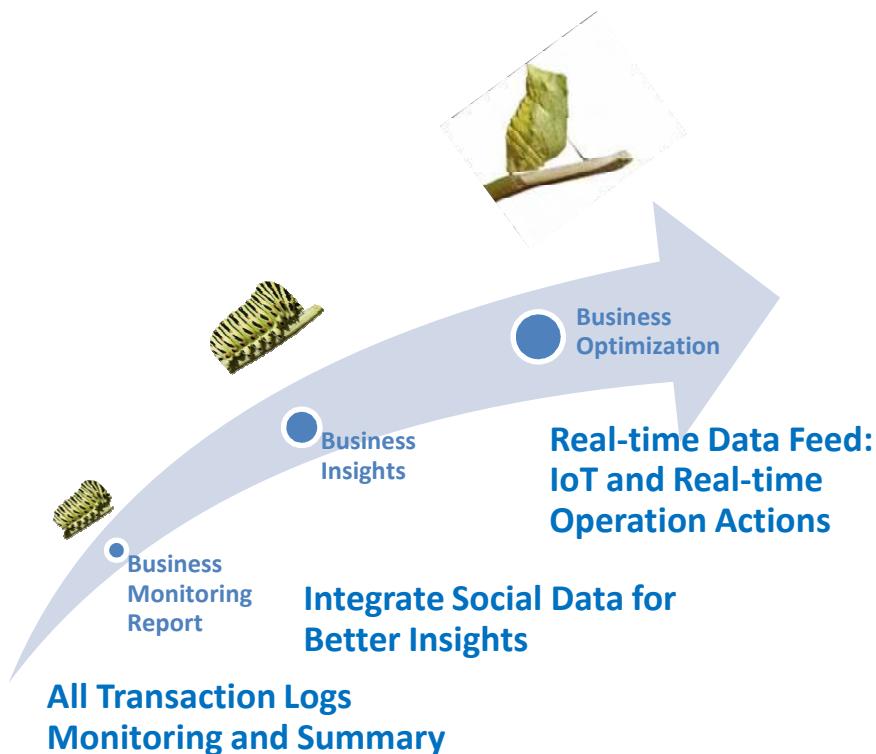
## Business Monitoring Report

Mine all the transactional data at the lowest levels of detail

## Business Insights

Integrate unstructured data with detailed structured (transactional) data to provide new metrics and new dimensions against which to monitor and optimize key business processes.

# Big Data Maturity Model



## Business Monitoring Report

Mine all the transactional data at the lowest levels of detail

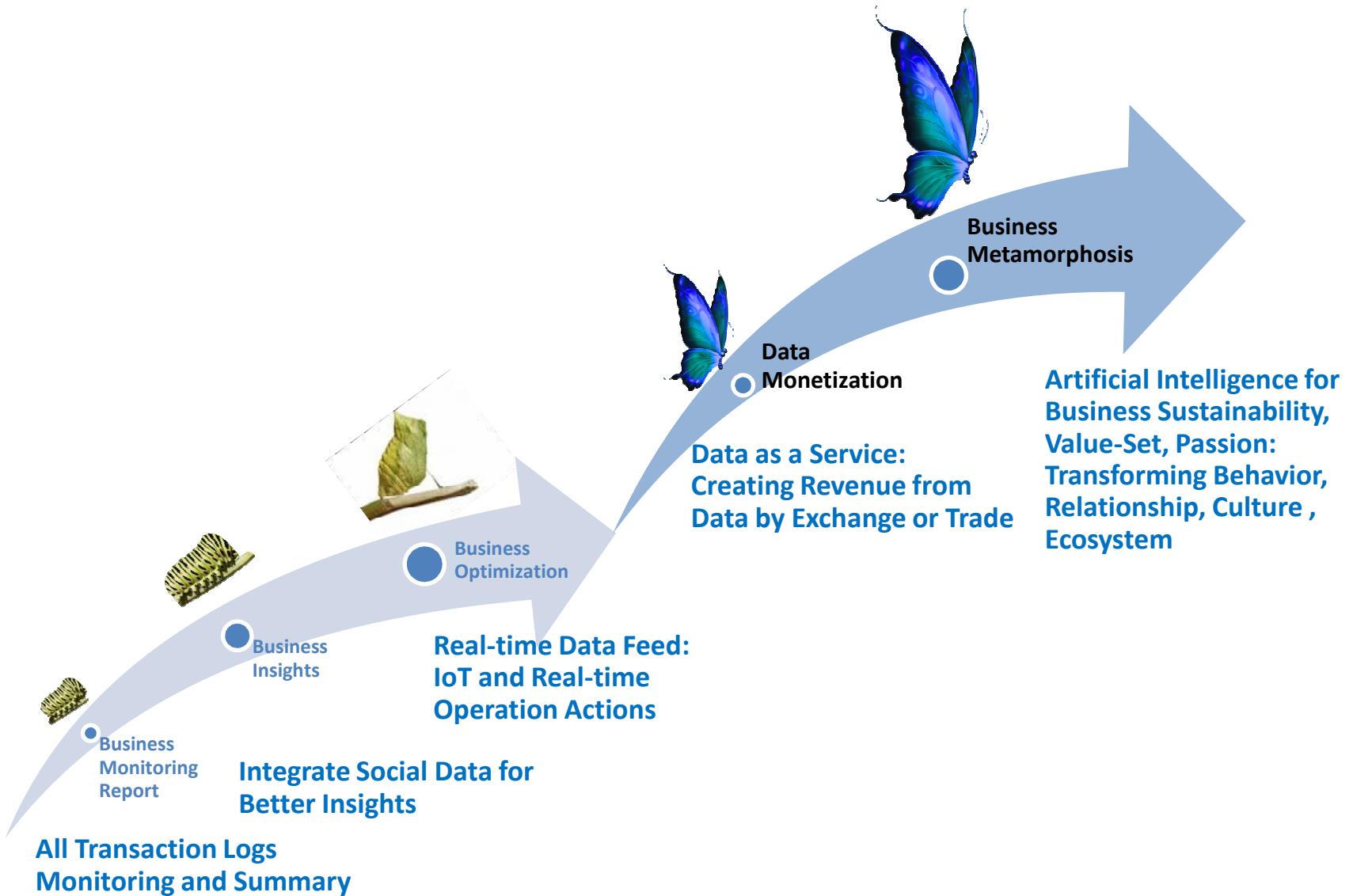
## Business Insights

Integrate unstructured data with detailed structured (transactional) data to provide new metrics and new dimensions against which to monitor and optimize key business processes.

## Business Optimization

Leverage real-time (or low-latency) data feeds to accelerate the organization's ability to identify and act upon business and market opportunities in a timely manner.

# Big Data Maturity Model

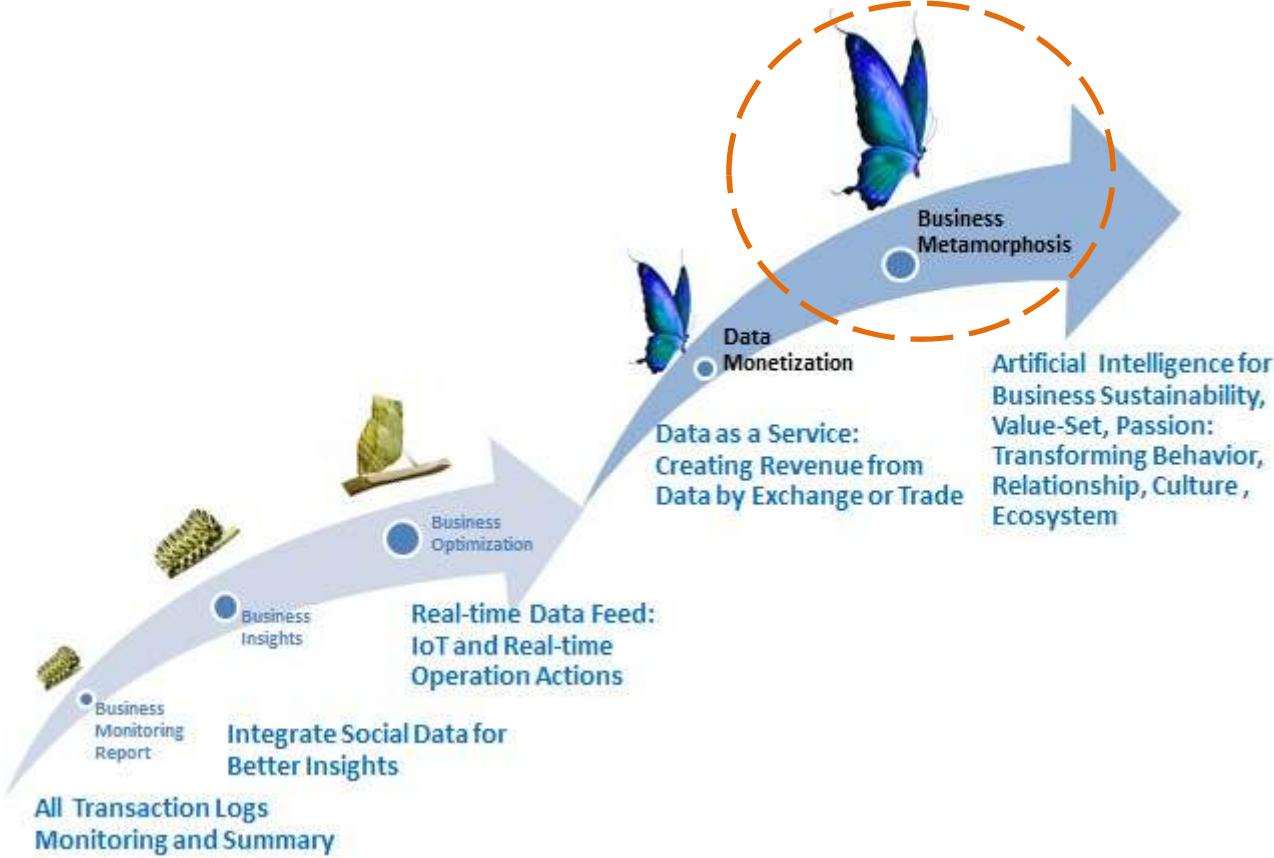


# Big Data Maturity Model



Integrate **predictive analytics** into your key business processes to uncover insights buried in the massive volumes of detailed structured and unstructured data. (Note: having business users slice and dice the data to uncover insights worked fine when dealing with gigabytes of data, but doesn't work when dealing with terabytes and petabytes of data.)

# Big Data Maturity Model



Driving new business models, new processes, more meaningful business interactions, innovation, improved and faster decision making, and a more agile organization

A digital ecosystem is a business community of organizations and individuals transacting across a distributed, adaptive, open, social, technical system with collaboration, transparency, constant evolution, self-organization, scalability and sustainability.

# Thank you.

Together we can!

อ.ดนัยรัฐ มนบดีธรรมจารี  
+668-1559-1446 Line ID: danairat  
FB: <https://www.facebook.com/tdanairat>