1、Introduction

1.1、What is Hadoop

談到 Hadoop 就不得不提到 Lucene 和 Nutch。首先，Lucene 並不是一個應用程序，而是提供了一個純 Java 的高性能全文索引引擎工具包，它可以方便的嵌入到各種實際應用中實現全文搜索/索引功能。

Nutch 是一個應用程序，是一個以 Lucene 爲基礎實現的搜索引擎應用，Lucene 爲 nutch 提供了文本搜索和索引的 API，Nutch 不光有搜索的功能，還有數據抓取的功能。在 nutch0.8.0 版本之前，Hadoop 還屬於 Nutch 的一部分，而從 nutch0.8.0 開始，將其中實現的 HDFS 和 MapReduce 剝離出來成立一個新的開源碼專案，這就是 Hadoop，而 nutch0.8.0 版本較之以前的 Nutch 在架構上有了根本性的變化，那就是完全構建在 Hadoop 的基礎之上了。

Hadoop 實現了 Google 的 GFS 和 MapReduce 演算法，使得 Hadoop 成爲了一個分散式的計算平台。其實，Hadoop 並不僅僅是一個用於存儲的分散式文件存取系統，而是被設計來用在由數台計算機組成的大型叢集上執行分散式應用的框架。

1.2、HDFS

HDFS 即 Hadoop Distributed File System (Hadoop 分散式文件系統)， HDFS 具有高容錯性，並且可以被部署在低價的硬體設備之上。HDFS 很適合應用在有大量資料處理需求的地方，並且提供了對數據讀寫的高 throughput。HDFS 是一個 master/slave 的結構，就通常的部署來說，在 master 上只執行一個 Namenode，而在每一個 slave 上執行一個 Datanode。

HDFS 支援傳統檔案系統的結構，在操作上就如同現有的一些檔案系統一樣很類似，比如產生和刪除一個文件，把一個文件從一個目錄移到另一個目錄，重命名..等等操作。Namenode 管理著整個分散式檔案系統，對檔案系統的操作（如建立、刪除文件和資料夾）都是通過 Namenode 來控制。如下圖一所示即爲 HDFS 的結構。

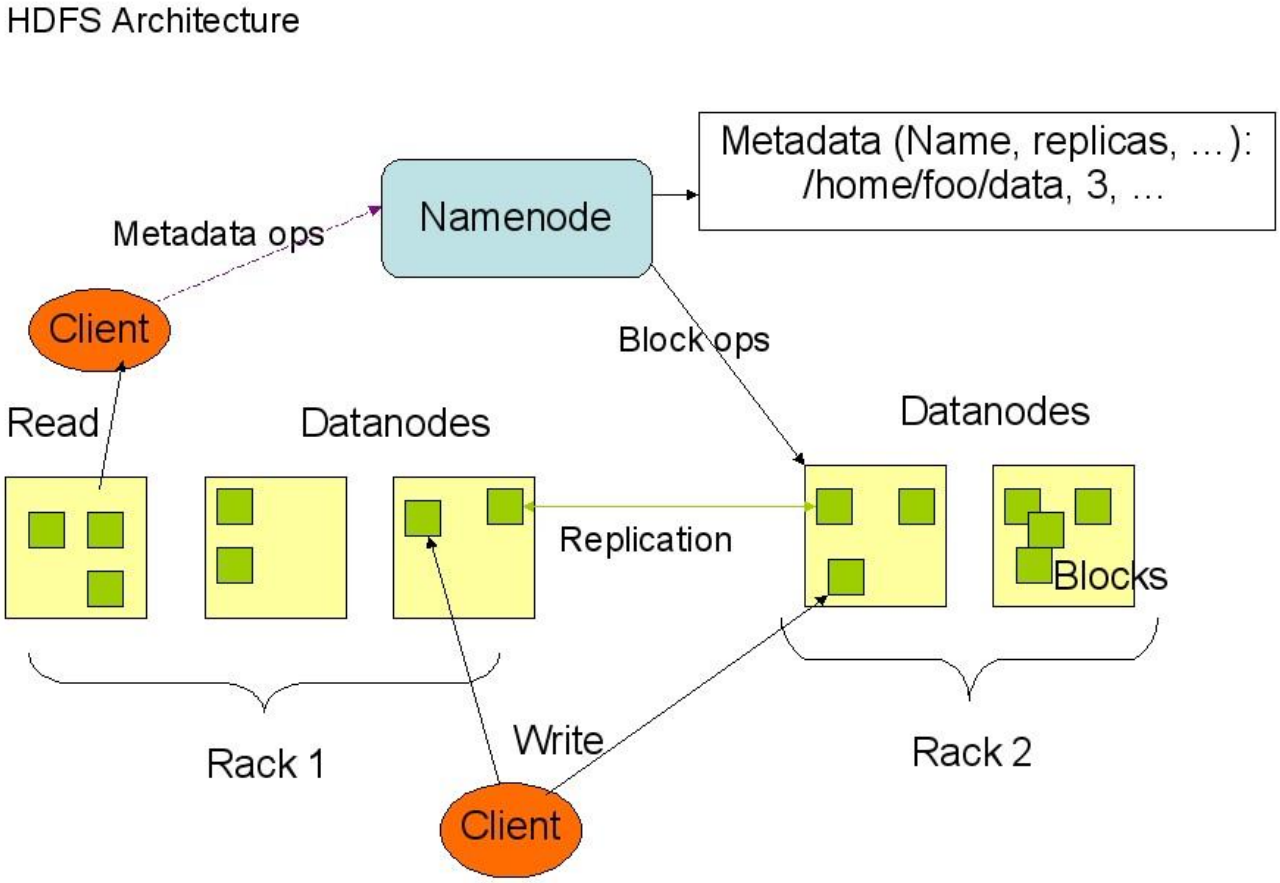

圖 1.2-1、HDFS 結構

從上面的圖中可以看出，Namenode、Datanode、Client 之間的通信都是建立在 TCP/IP 的基礎之上的。當 Client 要執行一個寫入操作的時候，指令並不是馬上就發送到 Namenode，而是 Client 首先得在本機的臨時資料夾中暫存這些資料，當臨時資料夾中的資料達到了設定的 Block 值（默認是 64MB）時，Client 便會通知 Namenode，此時 Namenode 便會回應 Client 的 RPC 請求，將資料插入檔案系統中並且在 Datanode 中找到一塊存放該資料的 block，同時將該 Datanode 及相對應的資料訊息告訴 Client，Client 便會將這些本機臨時資料夾中的資料寫入指定的 Datanode。

HDFS 採取了副本策略，其目的是爲了提高系統的可靠性及可用性。HDFS 的副本放置策略是存放三個副本，一個放在本節點上，一個放在同一機架中的另一個節點上，還有一個副本則是放在另一個不同的機架中的其中一個節點上。

2、*Hadoop* Prerequisites

2.1、Sun Java 6

以我們的實驗環境爲例：

作業系統	Debian GNU/Linux 5.0 (lenny)
核心(Kernel)	02/6/26

步驟一： 先安裝 DRBL 金鑰

執行指令：

$ wget -q http://drbl.nchc.org.tw/GPG-KEY-DRBL -O- | sudo apt-key add -

說明：待出現 OK 訊息後即完成匯入金鑰的動作。

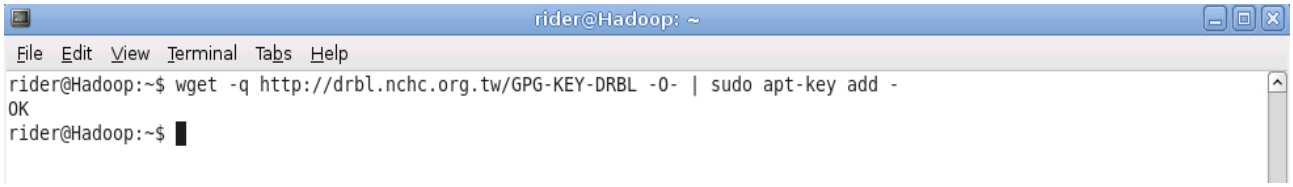

圖 2.1-1：DRBL 金鑰安裝

步驟二： 編輯套件庫來源 (新增 contrib , non-free 套件庫)

執行指令：

$ sudo /etc/apt/sources.list

$ sudo apt-get update

說明：新增套件庫來幫助快速安裝 Sun Java 6 。

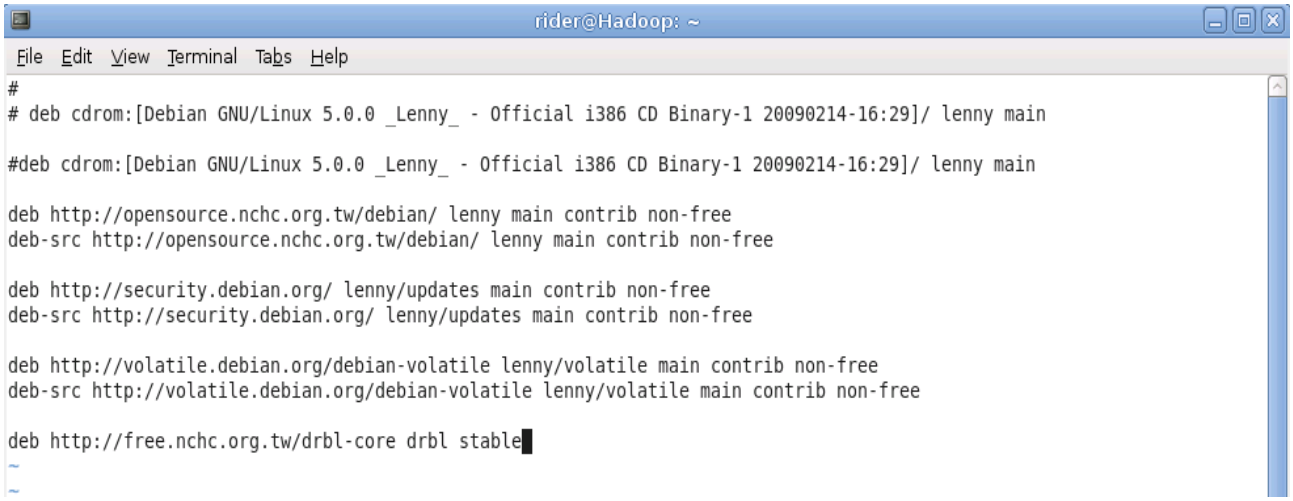

圖 2.1-2： 新增套件庫來源

步驟三： 安裝 Sun Java 6 等相關套件

執行指令：

$ sudo apt-get install sun-java6-bin sun-java6-jre sun-java6-jdk

說明：依照安裝指示即可順利完成 Sun Java 6 的安裝。

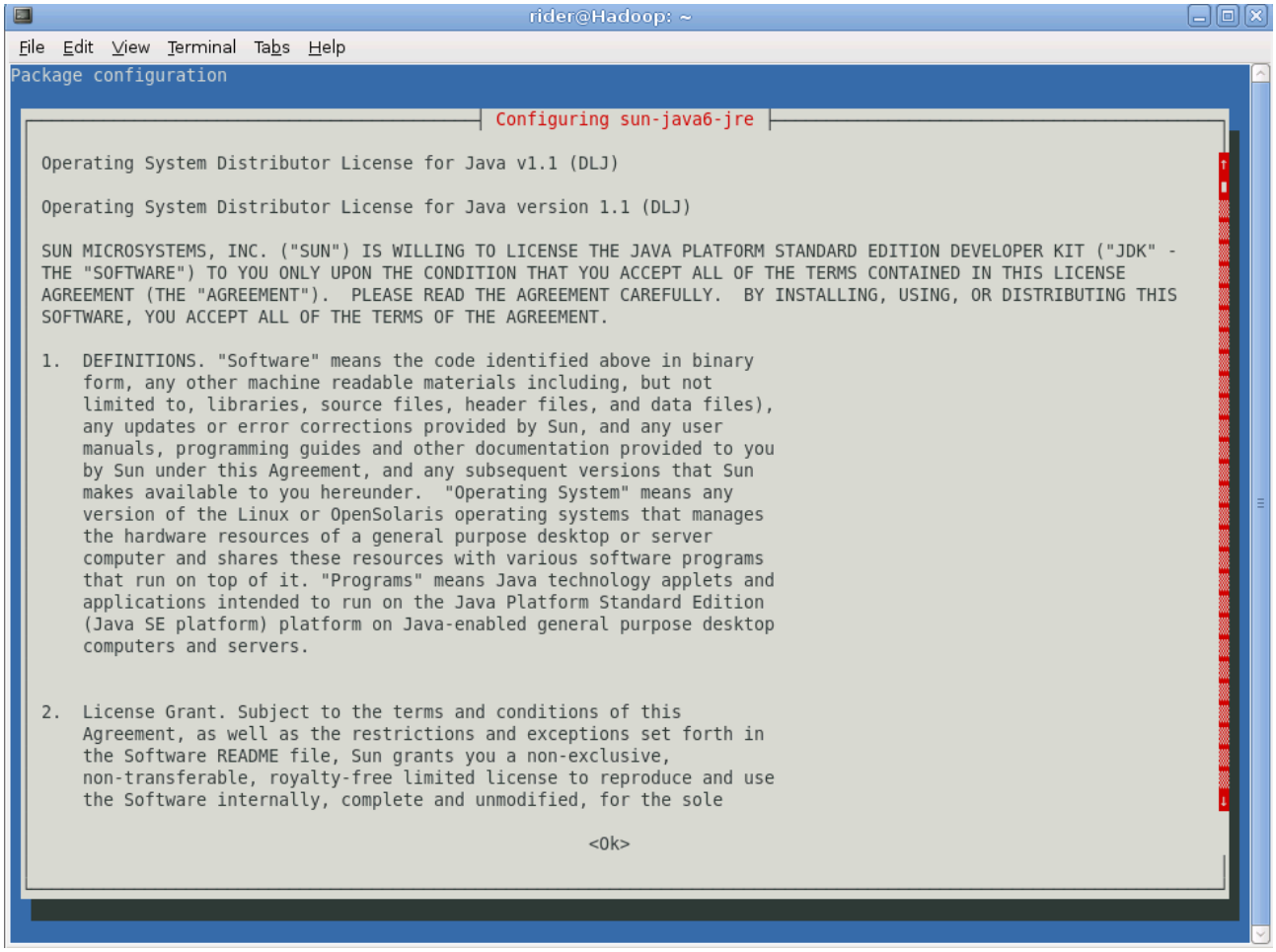

圖 2.1-3： Sun-Java6-jre 安裝畫面(1)

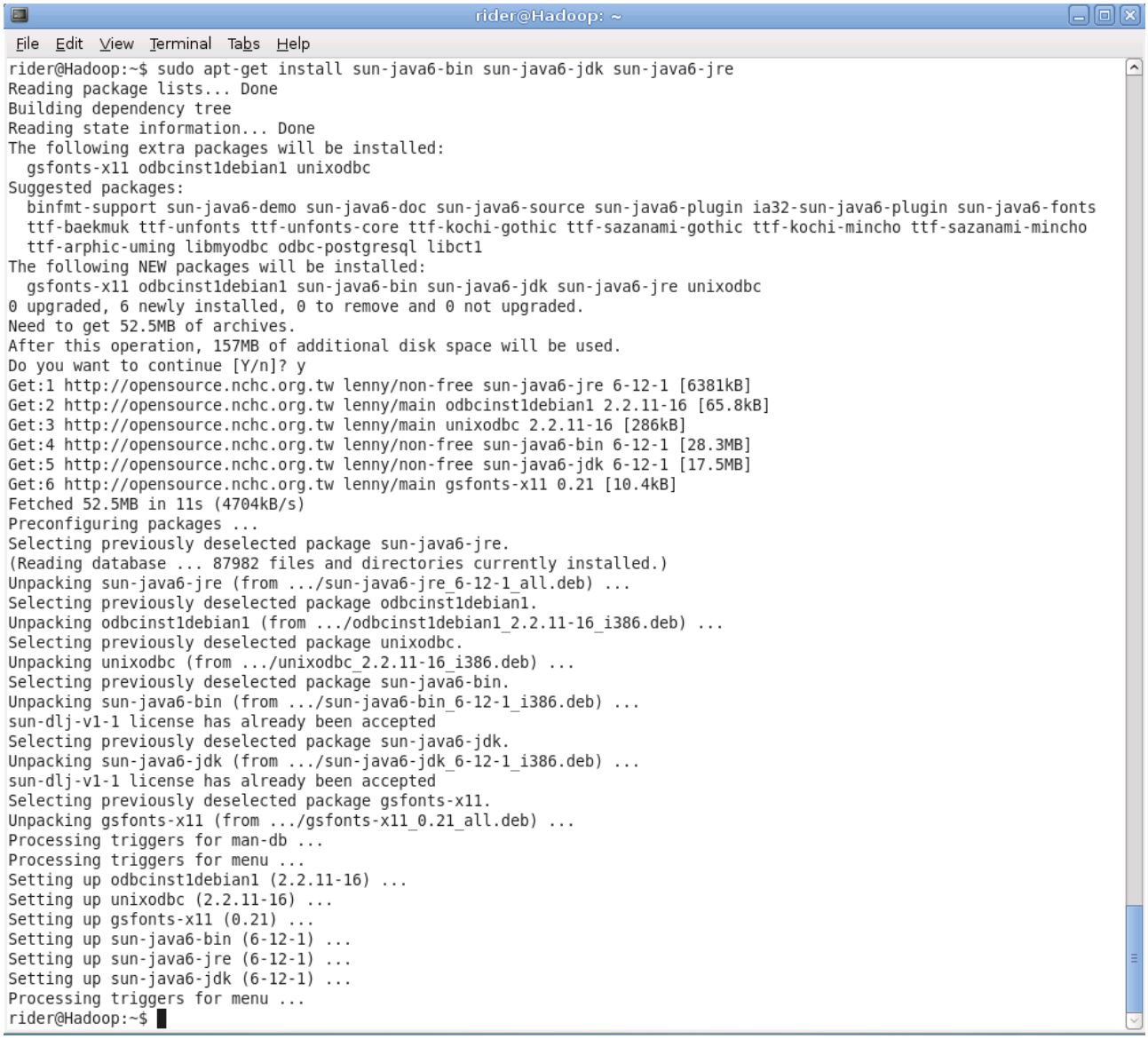

圖 2.1-4： Sun-Java6-jre 安裝畫面(2)

步驟四： 設定 JAVA_HOME 的環境變數

執行指令：

$ echo "export JAVA_HOME=/usr/lib/jvm/java-6-sun" >> ~/.bash_profile

$ source ~/.bash_profile

說明：可以透過 export 指令來查看 JAVA6 的環境變數是否有設定完成。

2.2、SSH

步驟一： 安裝 OpenSSH Server 套件

執行指令：

$ sudo apt-get install openssh-server

說明：Hadoop 是透過 ssh 登入的方式來管理每個 node，也可順便提升使用者或管理員往後使用與管理的方便性。

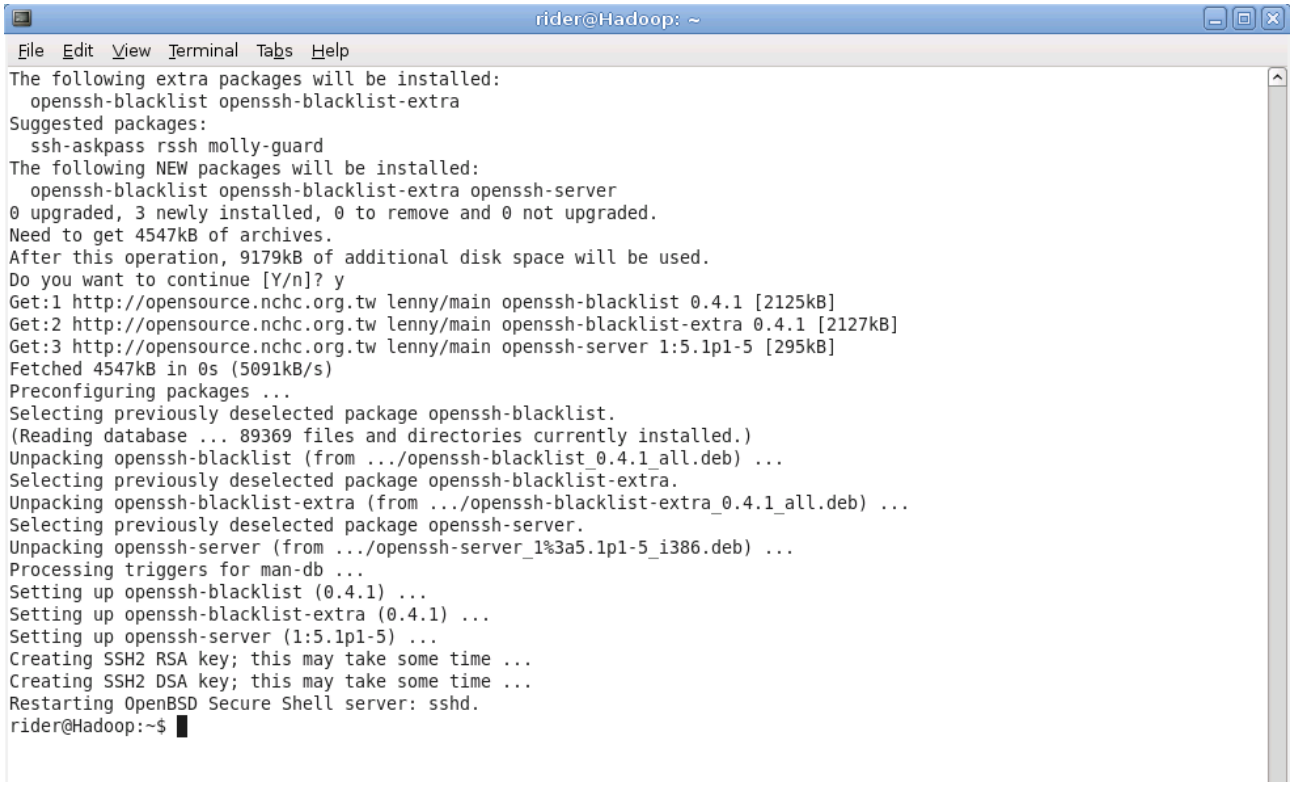

圖 2.2-1： OpenSSH 安裝畫面

步驟二： 產生 SSH key

執行指令：

$ ssh-keygen -t rsa

$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

說明：藉由產生 SSH key 來方便以後新增或移除主機的管理方便性。以後只需修改 (新增或刪除 SSH key) “authorized_keys”便可輕鬆管理各節點。

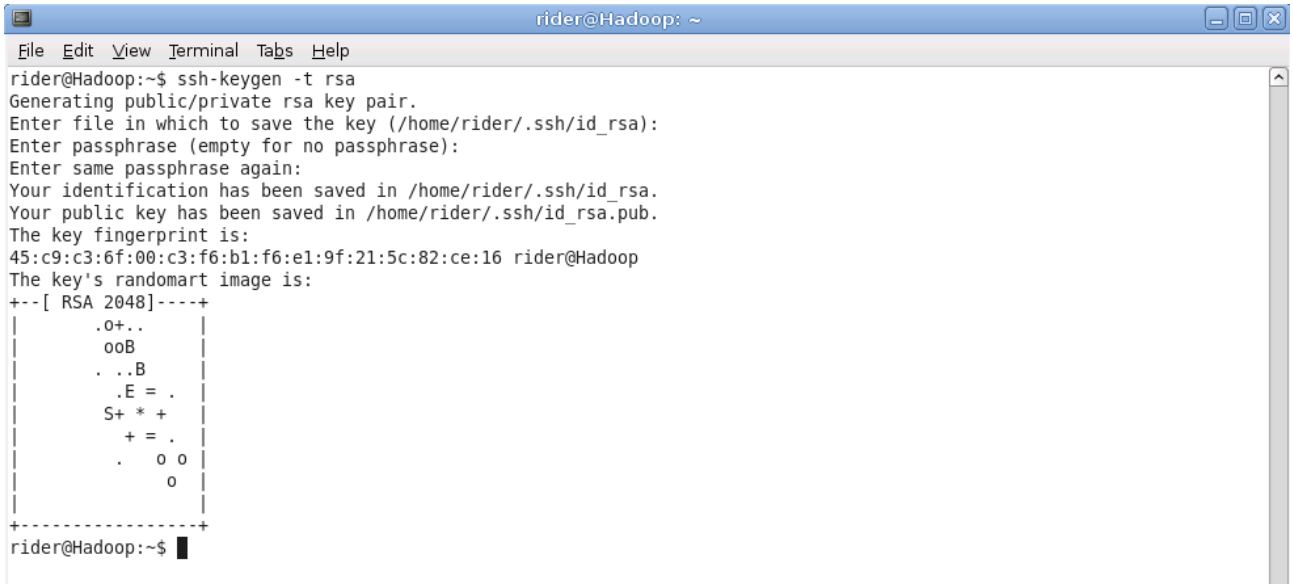

圖 2.2-2： 產生 SSH key

2.3、Disabling Ipv6

步驟一： 編輯禁用模組清單

執行指令：

$ sudo vim /etc/modprobe.d/blacklist

新增：

```
blacklist ipv6
```

編輯完成後存檔離開。

說明：停用 ipv6，使得 ipv6 模組在開機時不會載入。

3、 Hadoop Installation

3.1、Hadoop Installation HOWTO

步驟一： 下載 Hadoop 安裝檔

執行指令：

$ wget **http://ftp.mirror.tw/pub/apache/hadoop/core/hadoop-0.18.3/hadoop-0.18.3.tar.gz**

步驟二： 將 Hadoop 安裝檔解壓縮到欲安裝的目錄

執行指令：

$ sudo tar zxvf hadoop-0.18.3.tar.gz -C /opt

$ sudo ln -sf /opt/hadoop-0.18.3 /opt/hadoop

說明：將 Hadoop 解壓縮到安裝目錄並建立連結。

步驟三： 編輯 Hadoop 環境變數

執行指令：

$ sudo vim /opt/hadoop-0.18.3/conf/hadoop-env.sh

新增設定內容如下：

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun
export HADOOP_HOME=/opt/hadoop-0.18.3
export HADOOP_CONF_DIR=$HADOOP_HOME/conf
```

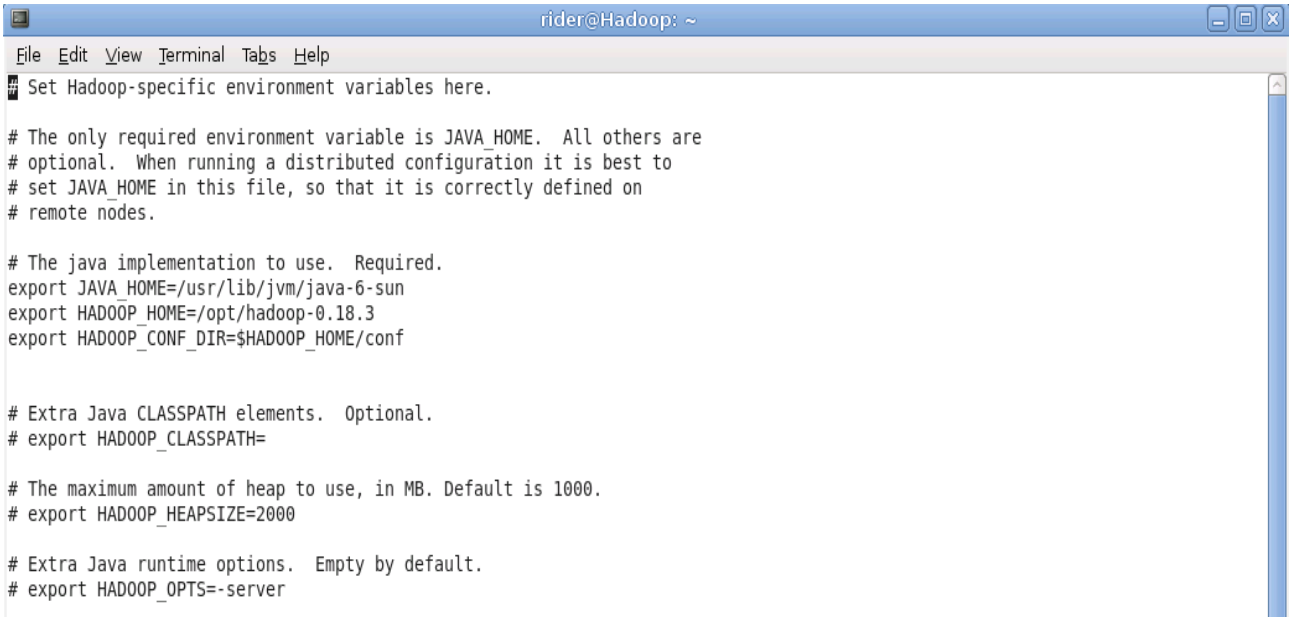

圖 3.1-1： Hadoop 環境變數設定

步驟四： 編輯 Hadoop Site 設定檔

執行指令：

$ sudo vim /opt/hadoop-0.18.3/conf/hadoop-site.xml

新增設定內容如下：

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. →

<configuration>
  <property>
      <name>fs.default.name</name>
      <value>hdfs://localhost:9000/</value>
      <description>
        The name of the default file system. Either the literal string
        "local" or a host:port for NDFS.
      </description>
  </property>
  <property>
      <name>mapred.job.tracker</name>
      <value>hdfs://localhost:9001</value>
      <description>
        The host and port that the MapReduce job tracker runs at. If
        "local", then jobs are run in-process as a single map and
        reduce task.
      </description>
```

```
  </property>
</configuration>
```

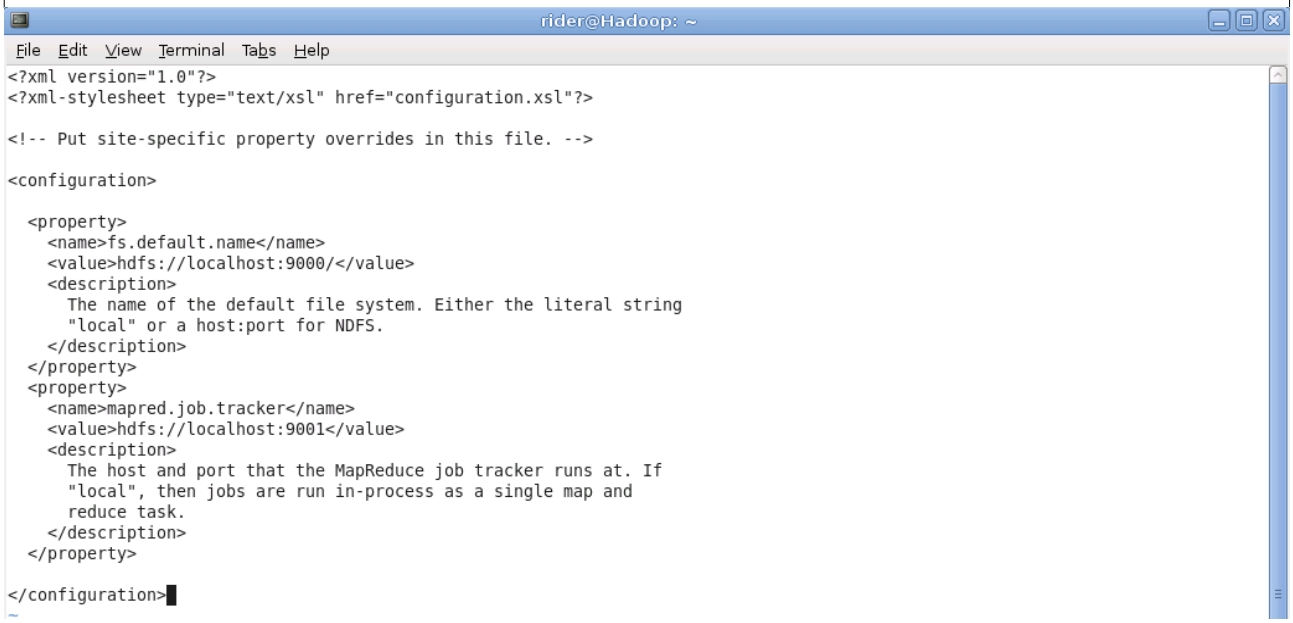

圖 3.1-2： Hadoop Site 設定檔畫面

步驟五： Formatting Hadoop Namenode

執行指令：

$ /opt/hadoop-0.18.3/bin/hadoop namenode -format

對應顯示之訊息應如下：

```
rider@Hadoop:~$ /opt/hadoop-0.18.3/bin/hadoop namenode -format
09/02/18 22:53:57 INFO dfs.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Hadoop/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 0.18.3
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/core/branches/branch-0.18 -r 736250; compiled by 'ndaley' on Thu Jan 22 23:12:08 UTC 2009
************************************************************/
09/02/18 22:53:58 INFO fs.FSNamesystem: fsOwner=rider,rider,dialout,cdrom,floppy,audio,video,plugdev,netdev,powerdev
09/02/18 22:53:58 INFO fs.FSNamesystem: supergroup=supergroup
09/02/18 22:53:58 INFO fs.FSNamesystem: isPermissionEnabled=true
09/02/18 22:53:58 INFO dfs.Storage: Image file of size 79 saved in 0 seconds.
09/02/18 22:53:58 INFO dfs.Storage: Storage directory /tmp/hadoop-rider/dfs/name has been successfully formatted.
09/02/18 22:53:58 INFO dfs.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at Hadoop/127.0.1.1
************************************************************/
```

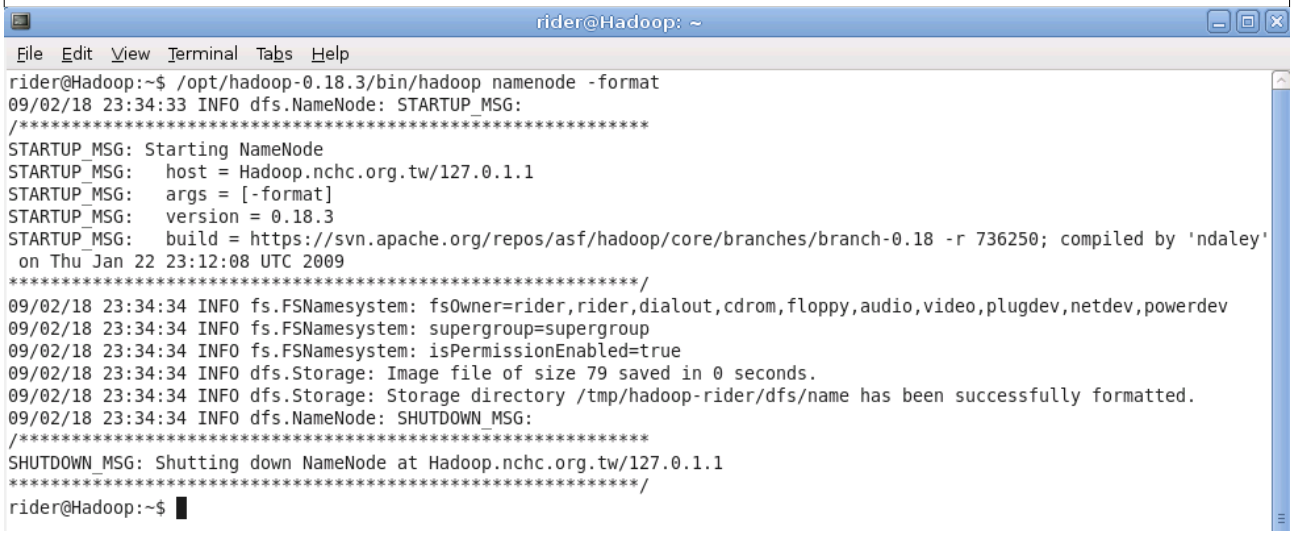

圖 3.1-3： Formatting Hadoop Namenode

步驟六： 啓動 Hadoop

執行指令：

$ sudo /opt/hadoop-0.18.3/bin/start-all.sh

對應顯示之訊息應如下：

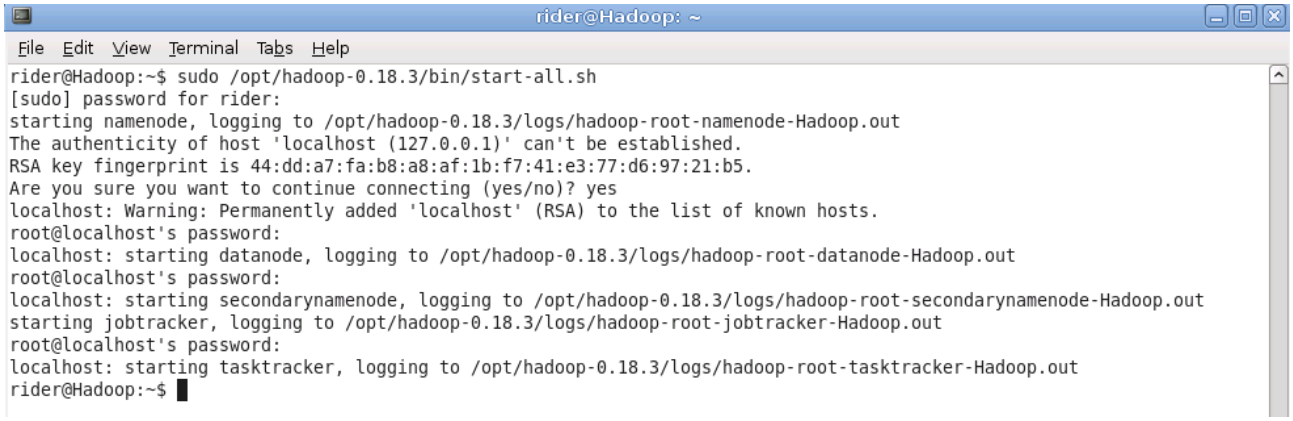

圖 3.1-4： Hadoop 啓動時的認證畫面

說明：Hadoop Server 上便會啓動 namenode, datanode, jobtracker, tasktracker

步驟七： 開啓 Hadoop Web Interface 來檢視啓動狀態

http://localhost:50030/ - web UI for MapReduce job tracker(s)

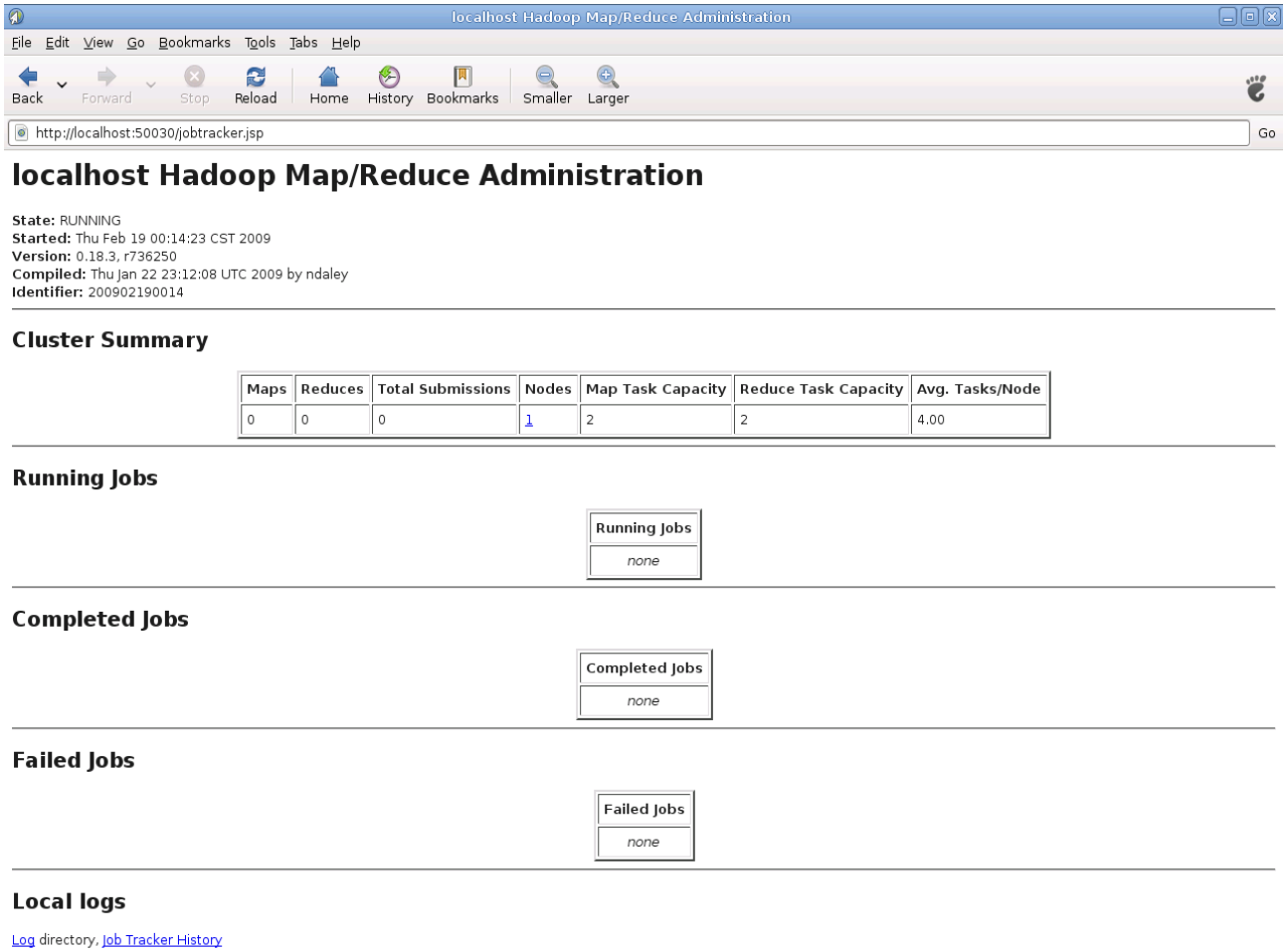

圖 3.1-5： web UI for MapReduce job tracker(s)

http://localhost:50060/ - web UI for task tracker(s)

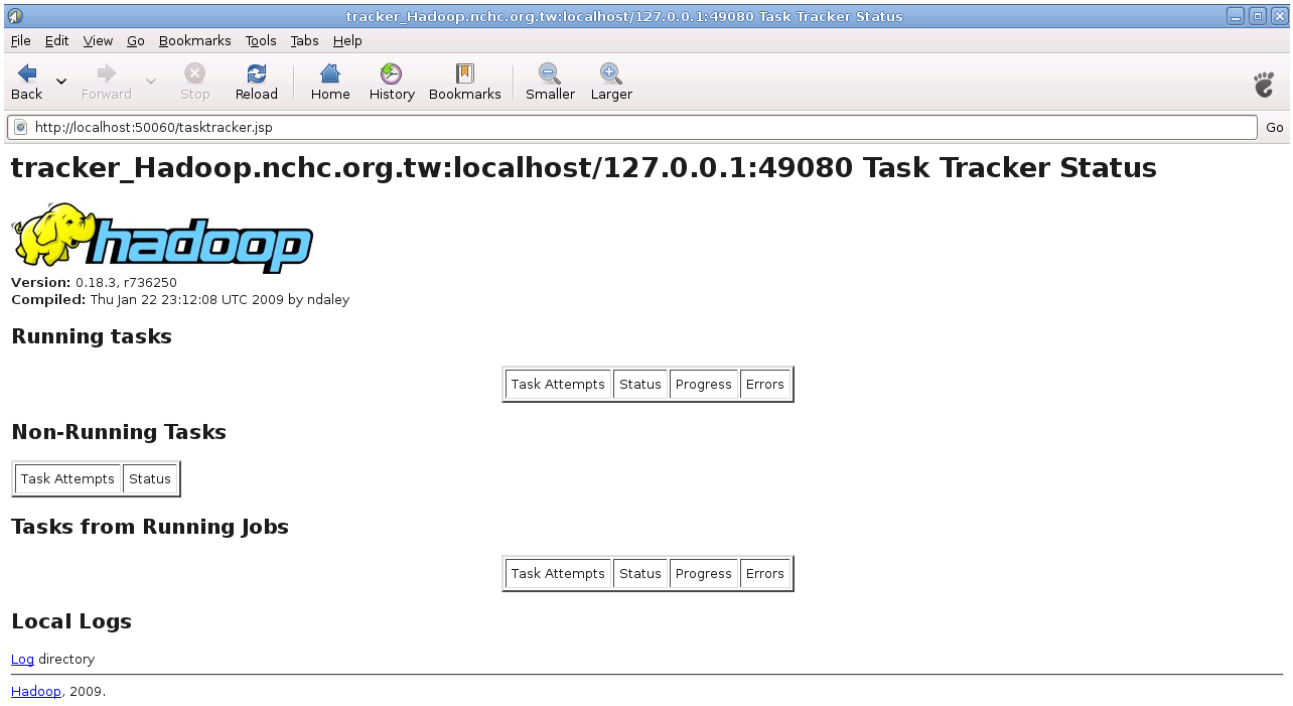

圖 3.1-6： web UI for task tracker(s)

http://localhost:50070/ - web UI for HDFS name node(s)

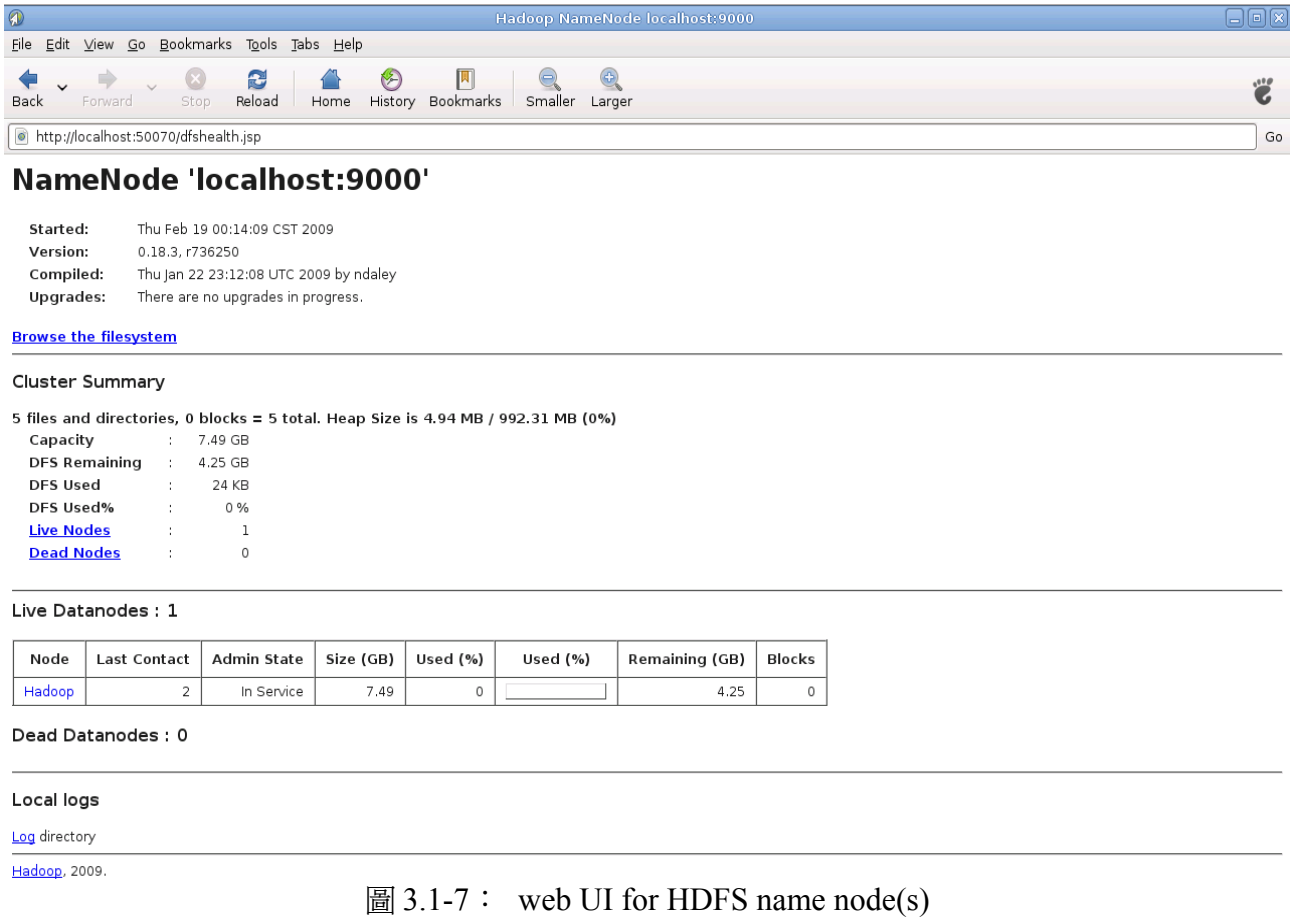

圖 3.1-7： web UI for HDFS name node(s)

說明：主要的初始化設定是參照 /opt/hadoop-0.18.3/conf/hadoop-default.xml 該設定檔，而此網頁介面也提供了許多 Hadoop cluster 簡明扼要的資訊，對於使用與管理上皆相當方便。

3.2、Hadoop 安裝目錄說明

環境說明：

目錄	說明
/hadoop	hadoop 家目錄
/hadoop/conf	hadoop 設定檔目錄
/hadoop/bin	hadoop 執行檔目錄
/hadoop/input	我們自己設定的來源文件檔目錄
/home/gtd	設定 gtd 爲 hadoop 的管理者，並且在每個 node 都要有 gtd 這個 user 才可以，/hodoop 下所有的資料也都要讓 gtd 有完全的讀寫權限。

表 3-1： Hadoop 安裝目錄說明表

3.3、Hadoop 基本測試 - Running a MapReduce job

步驟一：使用 “WordCount.java” example 來當測試範例

說明：此範例是去讀數份文字檔(電子書)，並且去計算文件中各文字出現的頻繁程度(次數)。此範例在 Hadoop 安裝完成後 Examples 資料夾便有提供。

步驟二：下載範例電子書

說明：我們從提供免費電子書的網站(**http://www.gutenberg.org/wiki/Main_Page**)隨意下載了三份供實驗範例的電子書到本機端，格式皆爲 us-ascii 編碼的 text 檔。

步驟三：將本機端的電子書複製到我們的 Hadoop HDFS

執行指令：

$ mkdir /tmp/gutenberg

$ mv *.txt /tmp/gutenberg

$ sudo /opt/hadoop/bin/start-all.sh

$ sudo /opt/hadoop/bin/hadoop dfs -copyFromLocal /tmp/gutenberg/ gutenberg

$ sudo /opt/hadoop/bin/hadoop dfs -ls

$ sudo /opt/hadoop/bin/hadoop dfs -ls gutenberg

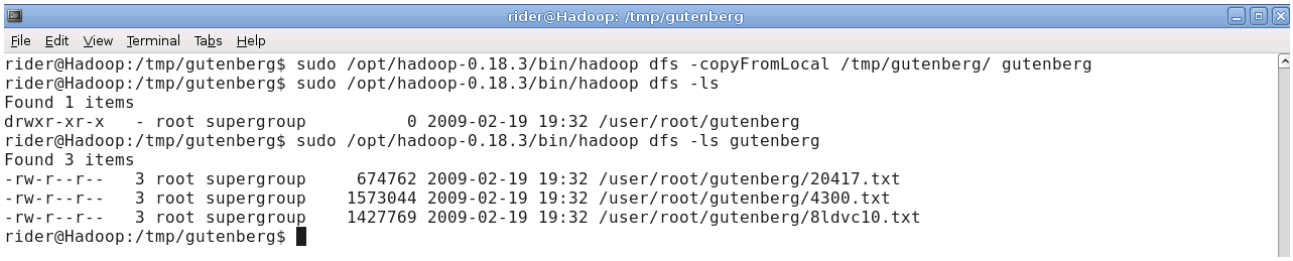

圖 3.3-1： 上傳測試電子書到 HDFS 並且檢查是否有上傳成功

說明：將測試用的電子書從本機端資料夾上傳到 Hadoop 的 HDFS 上面去。

步驟四：開始執行測試 Running MapReduce job

執行指令：

/opt/hadoop/$ sudo ./bin/hadoop jar hadoop-0.18.3-examples.jar wordcount gutenberg gutenberg-output

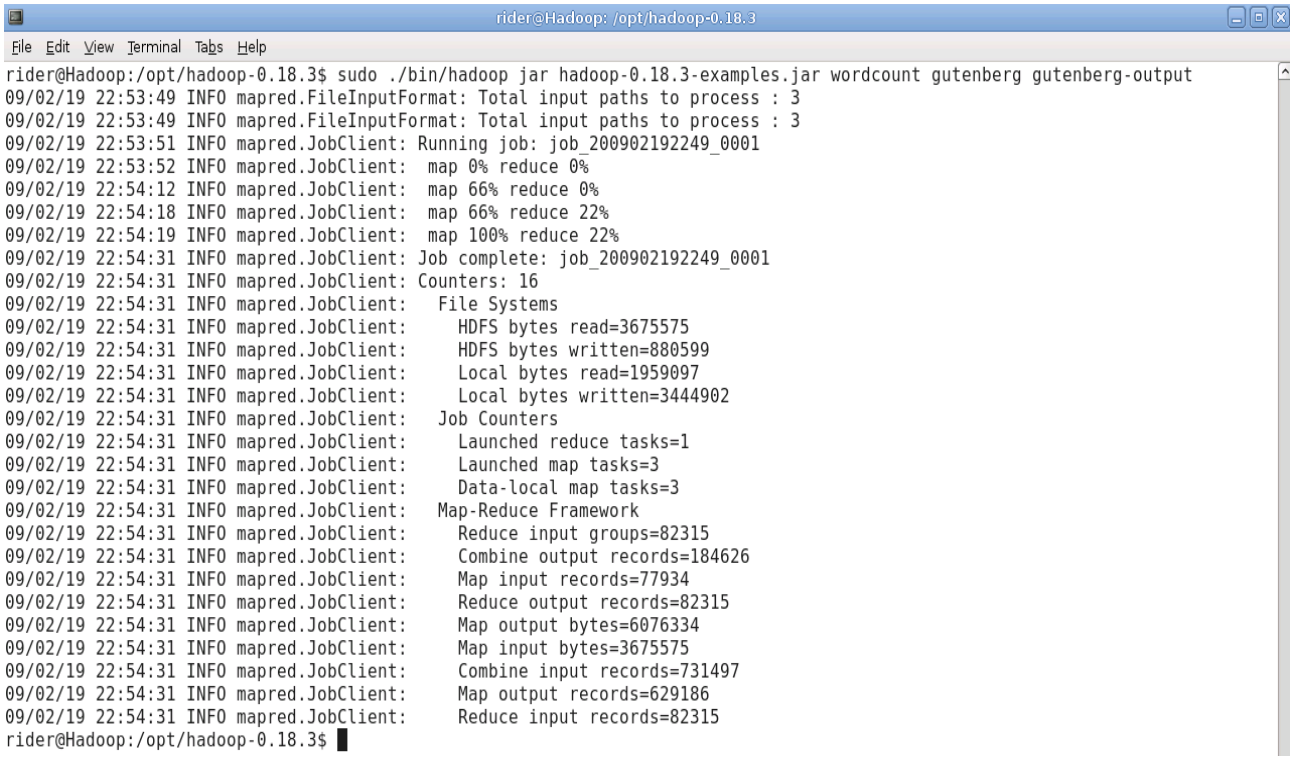

圖 3.3-2： 執行 MapReduce 測試

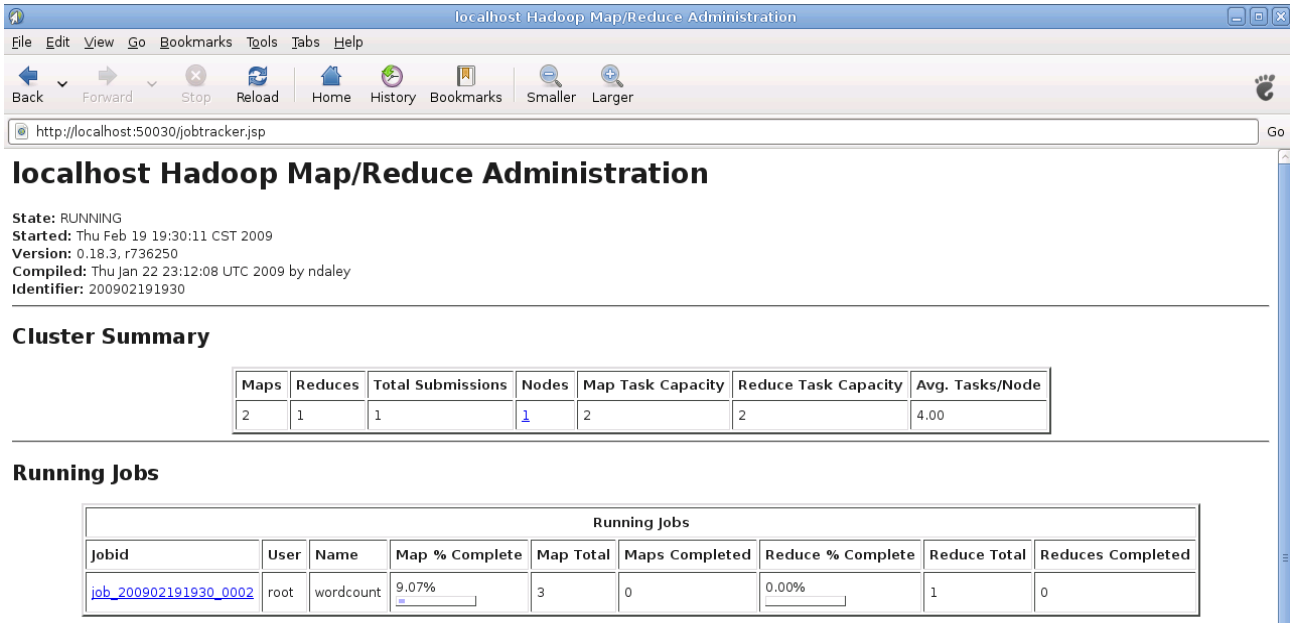

圖 3.3-3： 執行 MapReduce 時可以透過 Web 介面來掌握工作處理情形

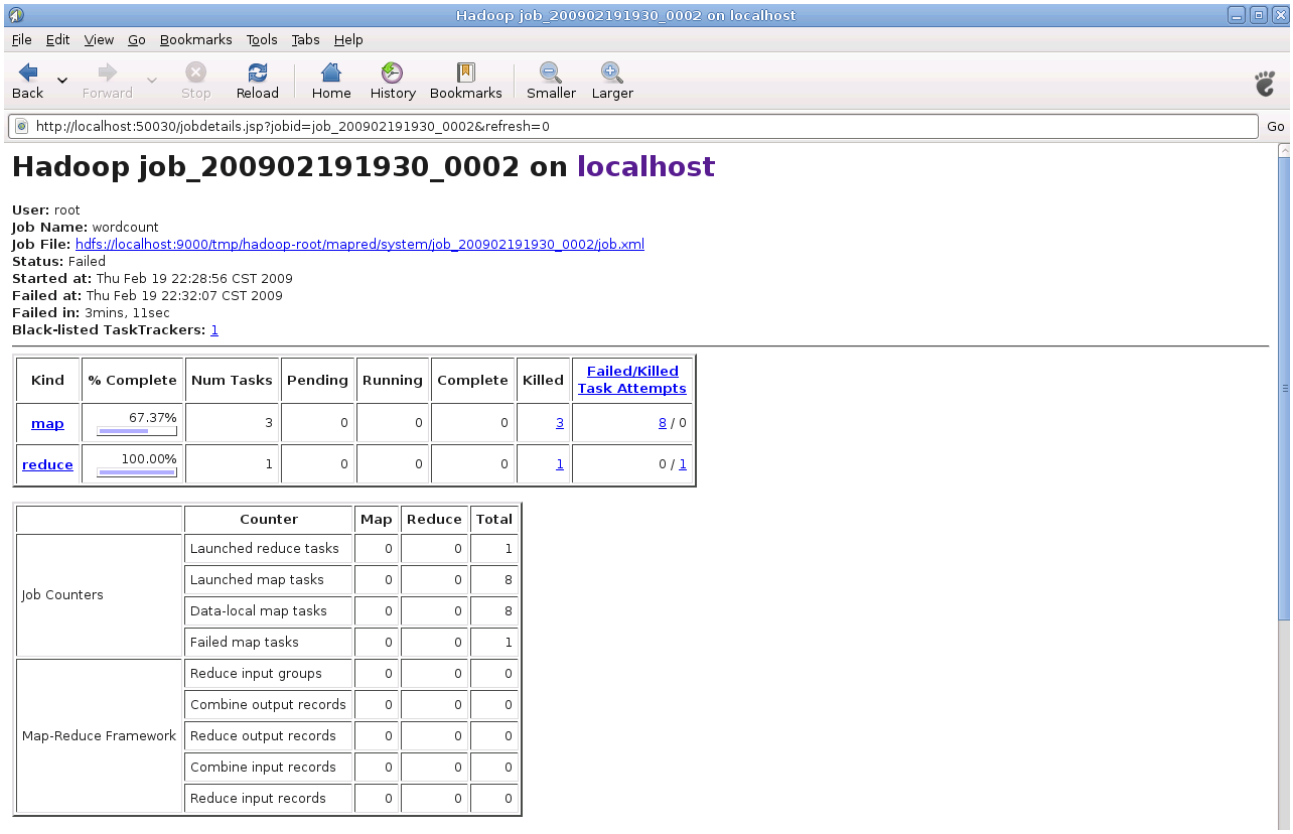

圖 3.3-4： 執行 MapReduce 時可以透過 Web 介面來掌握處理情形

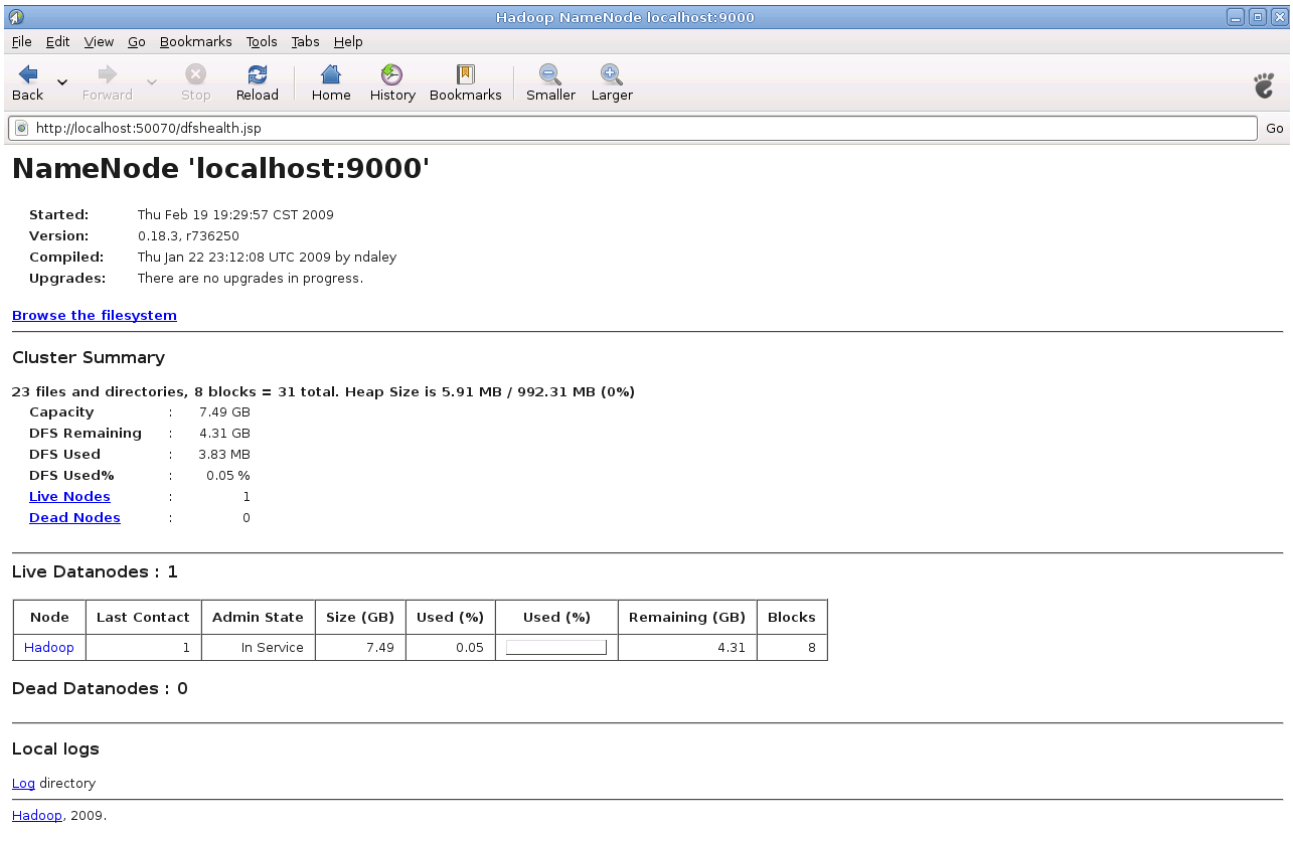

圖 3.3-5： 執行 MapReduce 時可以透過 Web 介面來掌握處理情形

說明：在上傳測試用電子書完成後，我們便可以開始執行 MapReduce 測試。

步驟五：檢視執行完後的結果

執行指令：

/opt/hadoop/$ sudo ./bin/hadoop dfs -ls

/opt/hadoop/$ sudo ./bin/hadoop/dfs -ls gutenberg-output

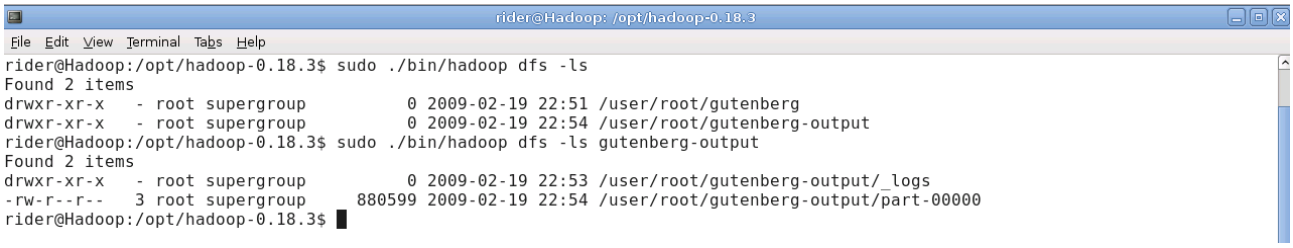

圖 3.3-6： 檢視執行完後的結果

/opt/hadoop/$ sudo ./bin/hadoop dfs -cat gutenberg-output/part-00000

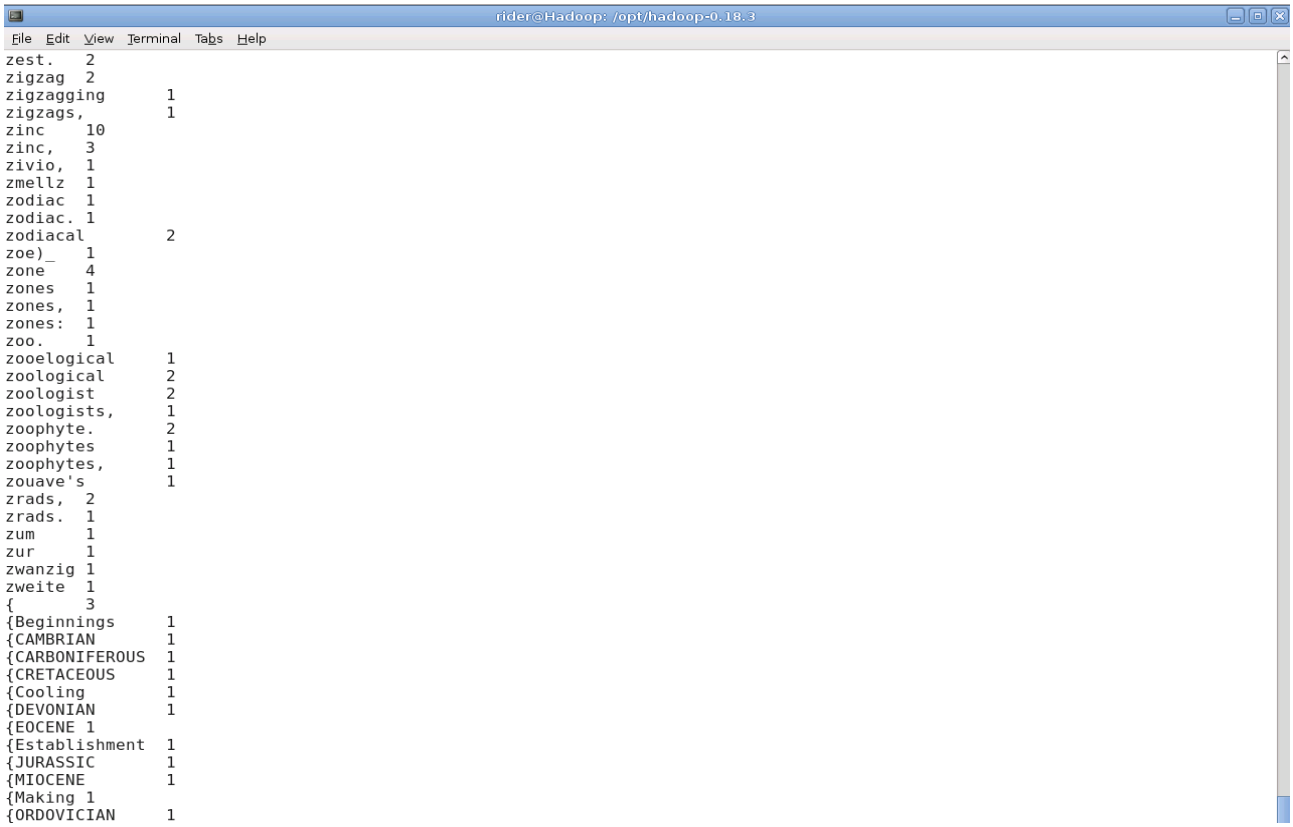

圖 3.3-7： 檢視執行完後的結果

說明：在 MapReduce 完成工作後，便可以檢視輸出結果與紀錄檔。

步驟六：將執行結果從 HDFS 取回本機端

執行指令：

/opt/hadoop/$ sudo mkdir /tmp/gutenberg-output

/opt/hadoop/$ sudo .bin/hadoop dfs -copyToLocal gutenberg-output/part-00000 /tmp/gutenberg-output

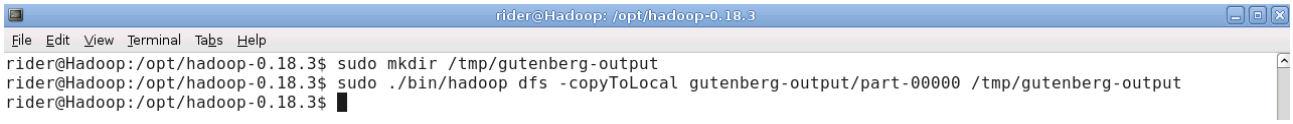

圖 3.3-8： 在本機端建資料夾並將完成的結果傳回本機端

/opt/hadoop/ $ head -n40 /tmp/gutenberg-output/part-00000

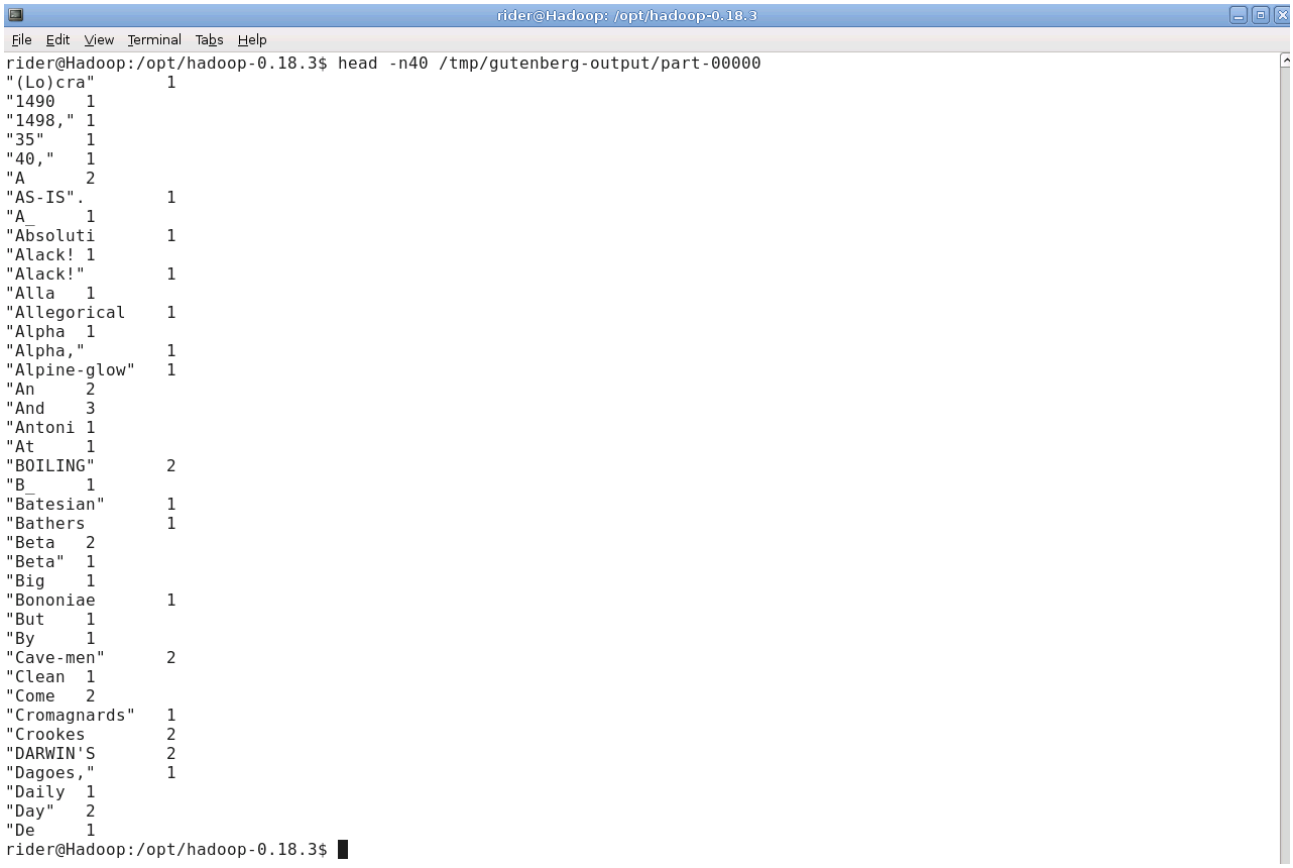

圖 3.3-9： 在本機端檢視的測試結果

說明：在本機端建立新資料夾並將結果從 HDFS 傳回本機端以供檢視測試結果，以確認 MapReduce 確實有完成我們指派的工作，而 Hadoop 基本測試也差不多完成了。

Reference:

1. **NCHC GTD Trac: Hadoop Hands-on Labs**
   https://trac.nchc.org.tw/cloud/wiki/HadoopWorkshopHandsOn

2. **Michael G.Noll Running Hadoop On Ubuntu Linux (Single-Node Cluster)**
   http://www.michael-noll.com/wiki/Running_Hadoop_On_Ubuntu_Linux_(Single-Node_Cluster)