Map Reduce 介紹

王耀聰 陳威宇
Jazz@nchc.org.tw
waue@nchc.org.tw

2008. 04 . 27-28

國家高速網路與計算中心(NCHC)

Outline

- Why should we learn this ?
- What is MapReduce ?
- Where does it fix ?
- What is its benefit ?
- How does it work ?
- Must be in Java ?

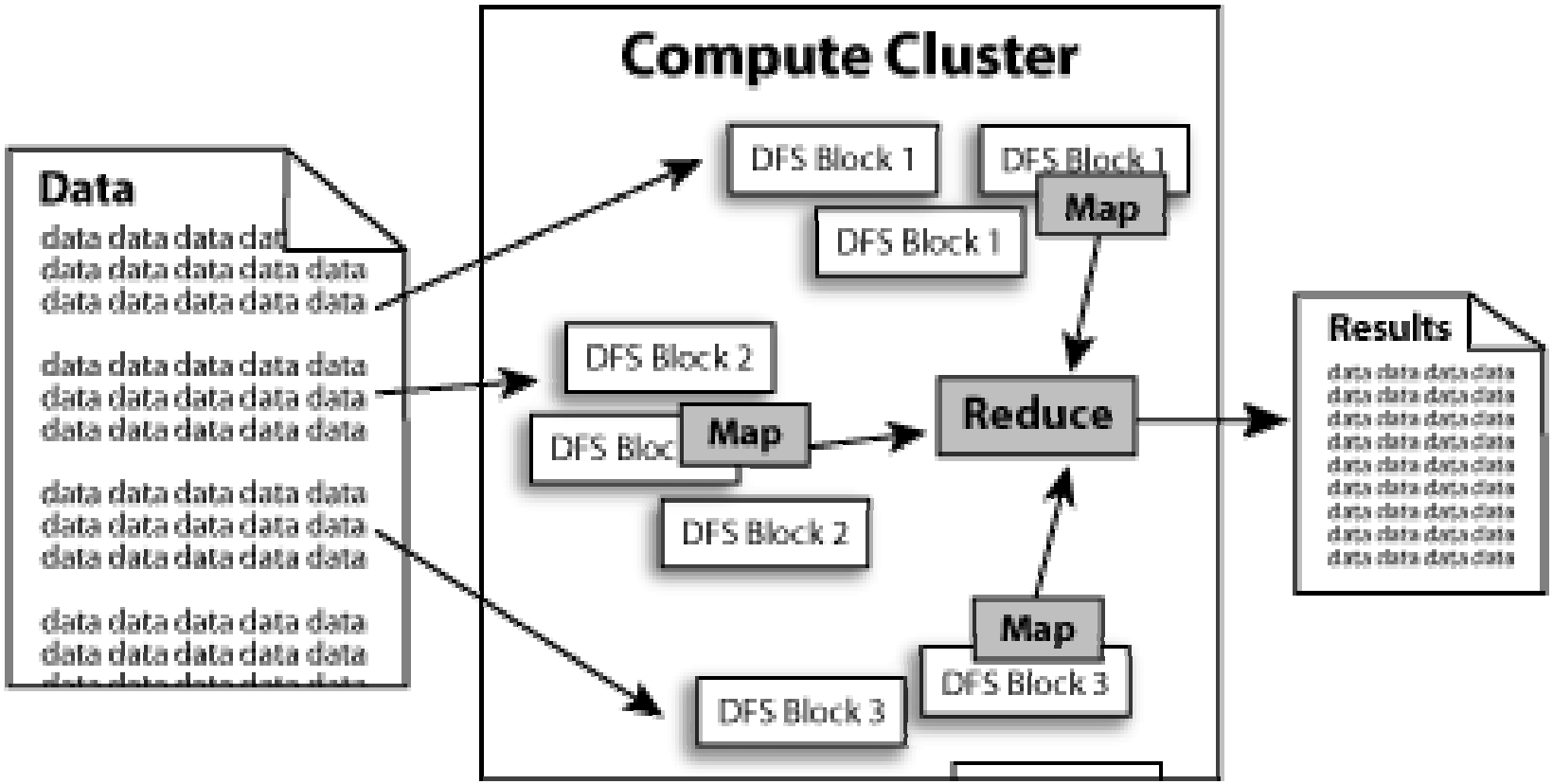
Why should we learn this ?

目的

知己知彼百戰百勝

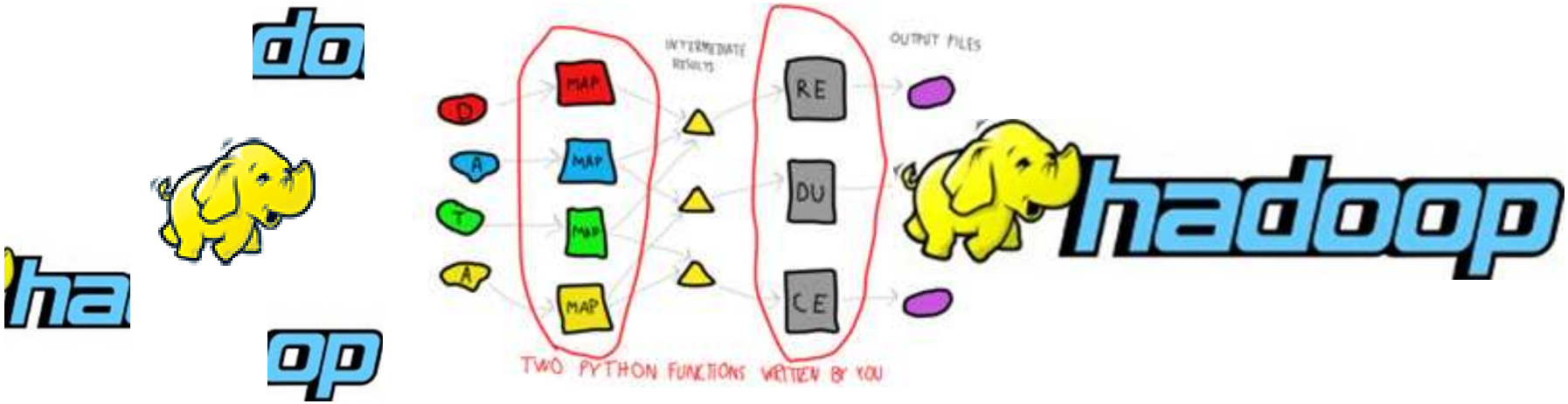
太歲頭上動土

Google 原生定義



MapReduce is a framework for computing certain kinds of distributable problems using a large number of computers (nodes), collectively referred to as a cluster.

Hadoop MapReduce定義



Hadoop Map/Reduce是一個易於使用的軟體平台，以MapReduce為基礎的應用程序，能夠運作在由上千台PC所組成的大型叢集上，並以一種可靠容錯的方式平行處理上P級別的資料集。

MapReduce由來

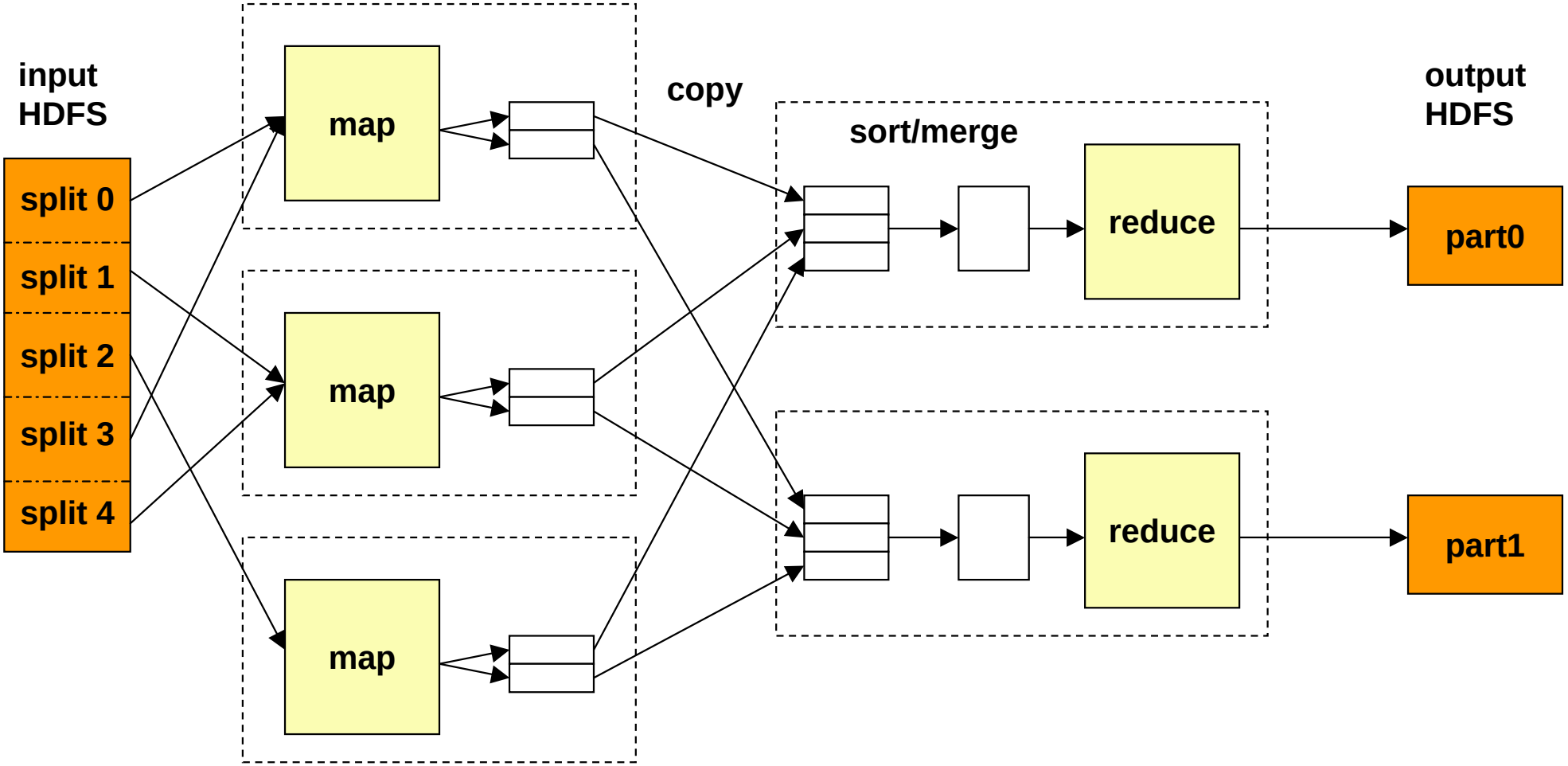
- Functional Programming : Map Reduce
  - map(...) :
    - [1,2,3,4] – (*2) -> [2,4,6,8]
  - reduce(...):
    - [1,2,3,4] - (sum) -> 10
  - 對應演算法中的Divide and conquer
  - 將問題分解成很多個小問題之後，再做總和
- 首先被Google引用到程式設計的軟體架構內，使用在大規模數據的運算中

應用範圍

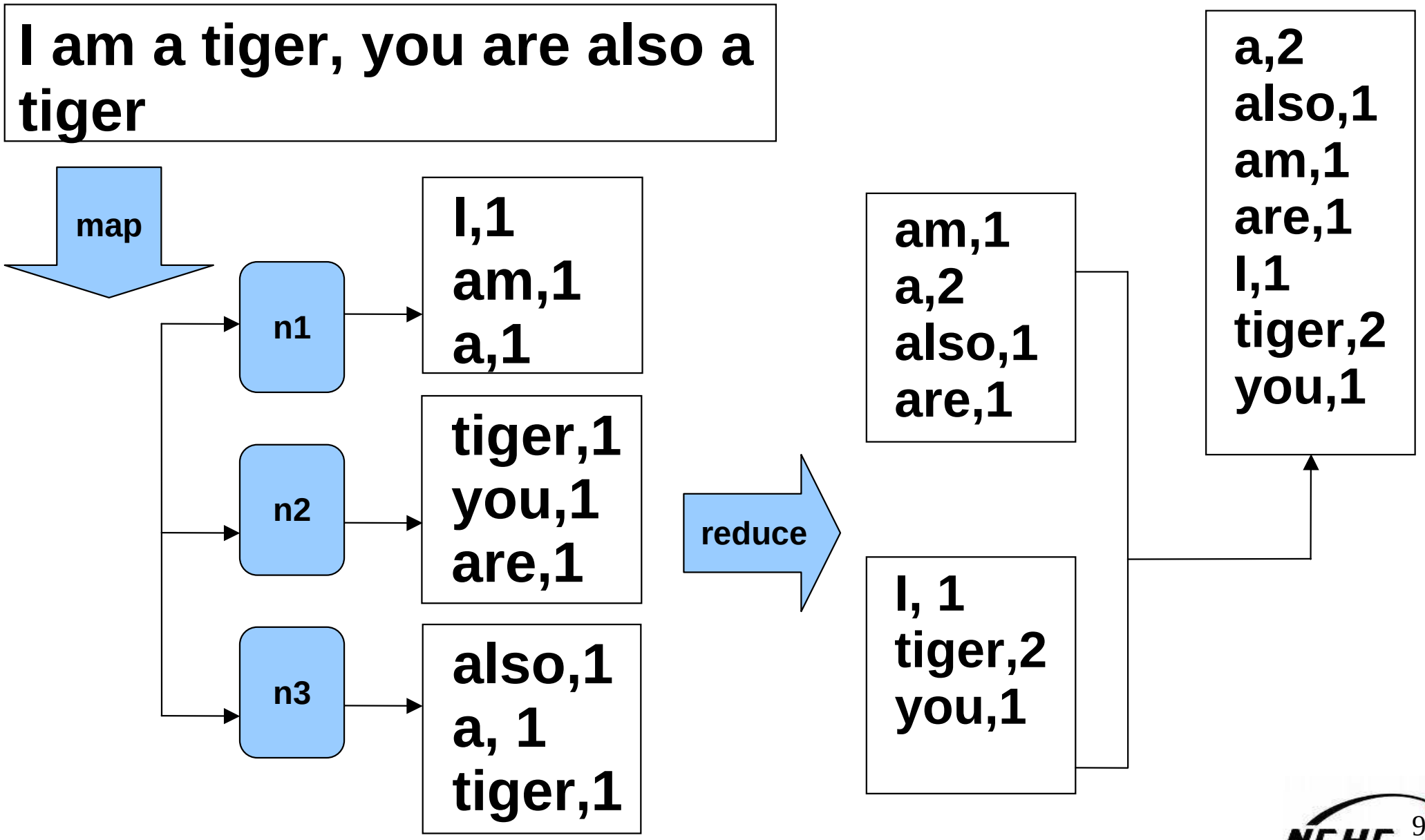
- Text tokenization
- Indexing and Search
- Data mining
- machine learning
- …

http://www.dbms2.com/2008/08/26/known-applications-of-mapreduce/

MapReduce 運作流程



範例


I am a tiger, you are also a tiger
map
n1
n2
n3
I,1
am,1
a,1
tiger,1
you,1
are,1
also,1
a, 1
tiger,1
reduce
am,1
a,2
also,1
are,1
I, 1
tiger,2
you,1
a,2
also,1
am,1
are,1
I,1
tiger,2
you,1

Must be in Java ?

Streaming & Pipes

- 雖然Hadoop框架是用Java實作，但Map/Reduce應用程序則不一定要用 Java來寫
- Hadoop Streaming ：
  - 執行作業的工具，使用者可以用其他語言（如：PHP）套用到Hadoop的mapper和reducer
- Hadoop Pipes：C++ API

Map Reduce Programming

王耀聰 陳威宇
Jazz@nchc.org.tw
waue@nchc.org.tw

2008. 04 . 27-28

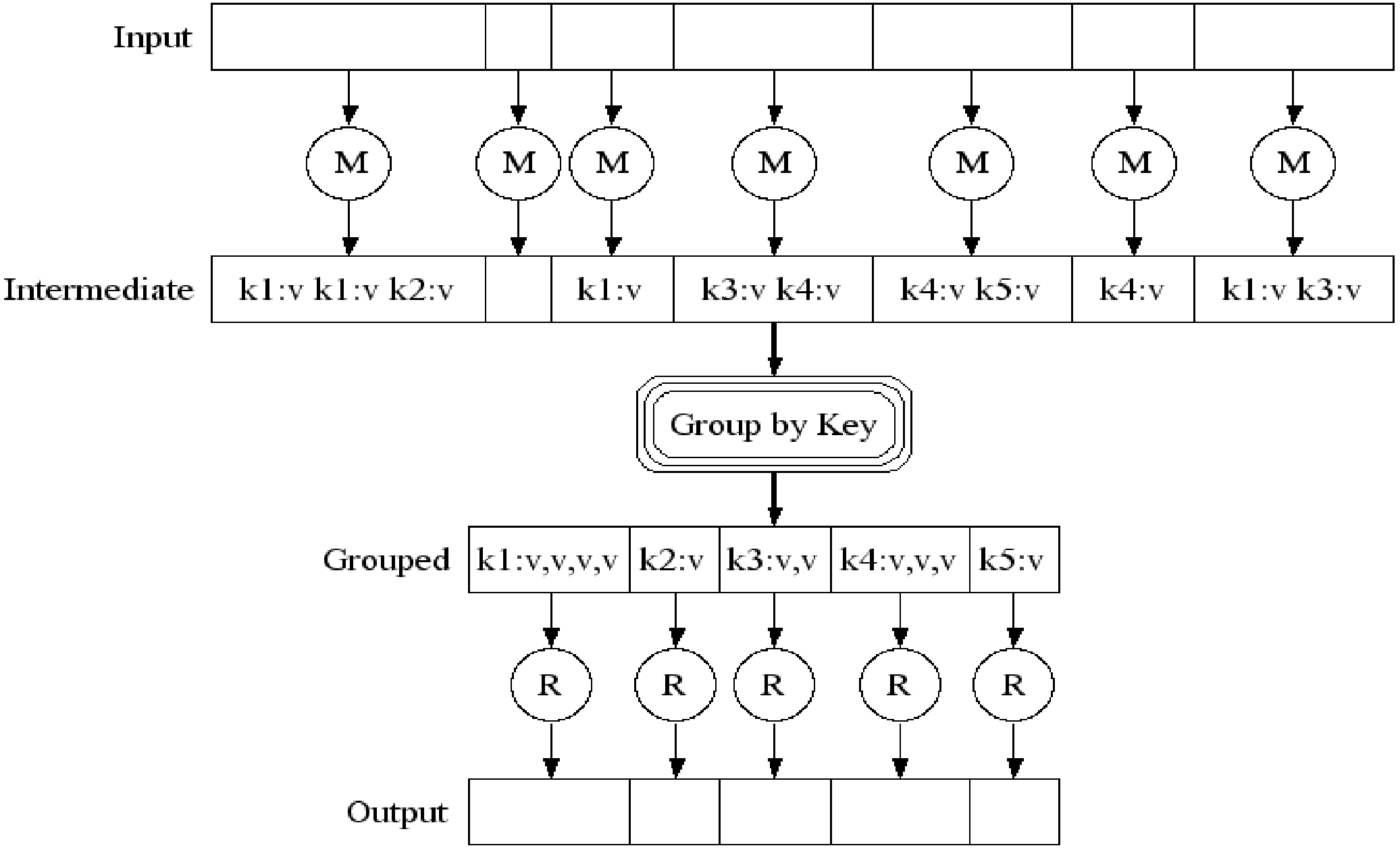
國家高速網路與計算中心(NCHC)

Outline

- 概念
- 程式基本框架及執行步驟方法
- 範例一:
  - Hadoop 的 Hello World => Word Count
  - 說明
  - 動手做
- 範例二:
  - 進階版=> Word Count 2
  - 說明
  - 動手做

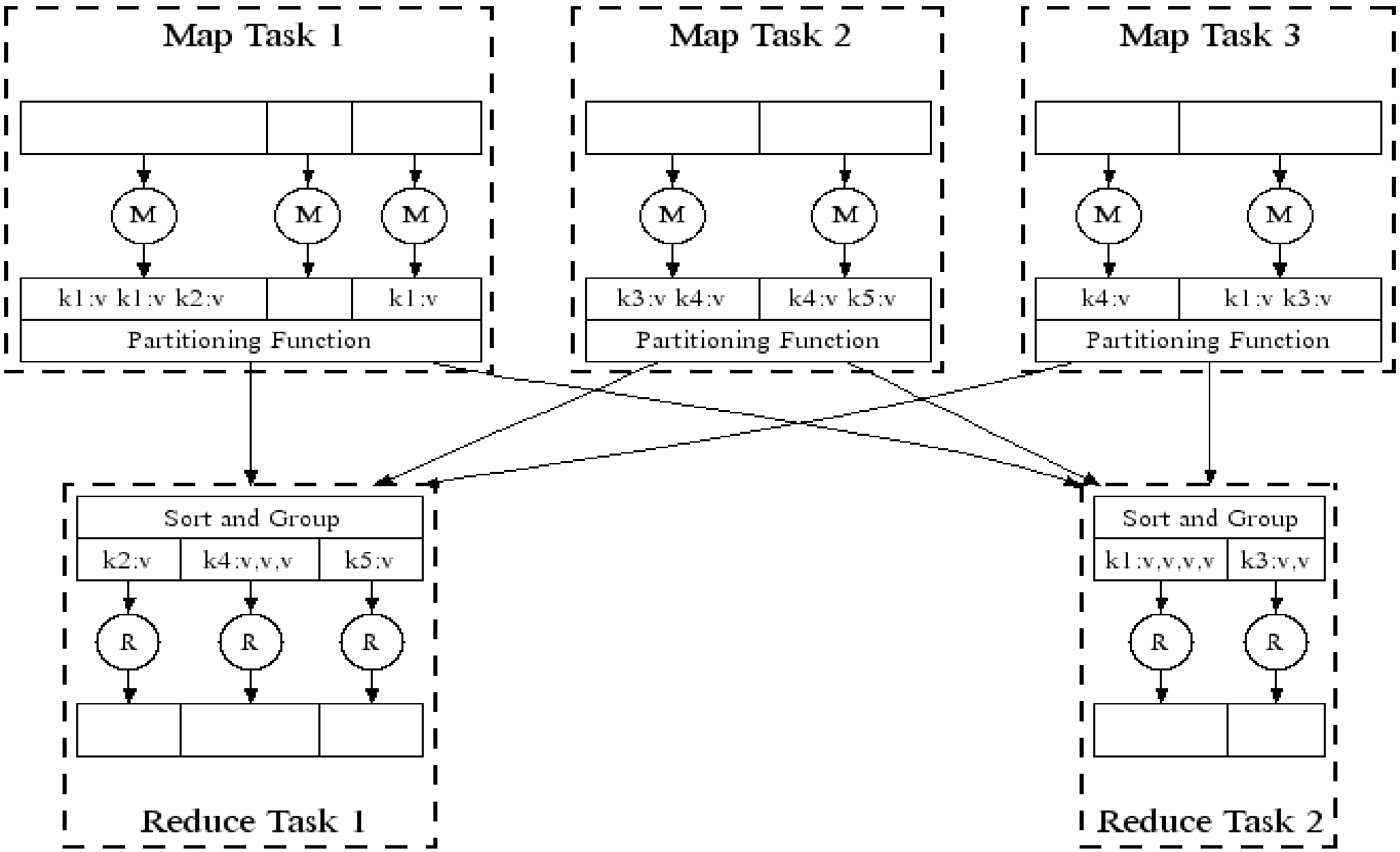
概念

MapReduce 圖解



概念

MapReduce in Parallel



程式基本框架

Program Prototype

```
Class MR{
    Class Map …{
    }
    Class Reduce …{
    }
    main(){
        JobConf conf = new JobConf( “MR.class” );
        conf.setInputPath( “the_path_of_HDFS ” );
        conf.setOutputPath( “the_path_of_HDFS ” );
        conf.setMapperClass(Map.class);
        conf.setReduceClass(Reduce.class);
        其他的設定參數程式碼
        JobClient.runJob(conf);
}}
```

Map 程式碼

Reduce 程式碼

Map 區

Reduce 區

設定區

執行流程
基本步驟

Process Prototype

1. 編譯
   - javac -classpath hadoop-*-core.jar -d MyJava MyCode.java
2. 封裝
   - jar -cvf MyJar.jar -C MyJava .
3. 執行
   - bin/hadoop jar MyJar.jar MyCode HDFS_Input/ HDFS_Output/

- 所在的執行目錄為Hadoop_Home
- ./MyJava = 編譯後程式碼目錄
- My jar. jar = 封裝後的編譯檔
- 先放些文件檔到HDFS上的input目錄
- ./input; ./ouput = hdfs的輸入、輸出目錄

範例程式一

Word Count Sample (1)

```
1   class MapClass extends MapReduceBase implements
    Mapper<LongWritable, Text, Text, IntWritable> {
2           private final static IntWritable one = new IntWritable(1);
3           private Text word = new Text();
4           public void map( LongWritable key, Text value,
             OutputCollector<Text, IntWritable> output, Reporter
                      reporter) throws IOException {
5                   String line = ((Text) value).toString();
6                   StringTokenizer itr = new StringTokenizer(line);
7                   while (itr.hasMoreTokens()) {
8                           word.set(itr.nextToken());
9                           output.collect(word, one);
    }}}
```

範例程式一

Word Count Sample (2)

```
1  class ReduceClass extends MapReduceBase implements
   Reducer< Text, IntWritable, Text, IntWritable> {
2          IntWritable SumValue = new IntWritable();
3          public void reduce( Text key, Iterator<IntWritable> values,
           OutputCollector<Text, IntWritable> output, Reporter reporter)
           throws IOException {
4                  int sum = 0;
5                  while (values.hasNext())
6                          sum += values.next().get();
7                  SumValue.set(sum);
8                  output.collect(key, SumValue);
   }}
```

範例程式一

Word Count Sample (3)

```
Class WordCount{
 main()
        JobConf conf = new JobConf(WordCount.class);
        conf.setJobName("wordcount");
        // set path
        conf.setInputPath(new Path(args[0]));
        conf.setOutputPath(new Path(args[1]));
        // set map reduce
        conf.setOutputKeyClass(Text.class);  // set every word as key
        conf.setOutputValueClass(IntWritable.class); // set 1 as value
        conf.setMapperClass(MapClass.class);
        conf.setCombinerClass(Reduce.class);
        conf.setReducerClass(ReduceClass.class);
        onf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);
        // run
        JobClient.runJob(conf);
}}
```

範例一說明

核心 Mapper

- <key/value > 的映射集合
- 設定
  - conf.setMapperClass(MapClass.class);
- 每次map的輸入
  - map (WritableComparable, Writable, OutputCollector, Reporter)
- map完後的輸出
  - OutputCollector.collect (WritableComparable,Writable)

範例一說明

核心Combiner

- 指定一個combiner，它負責對中間過程的輸出進行本地的聚集，這會有助於降低從Mapper 到 Reducer數據傳輸量。
- 設定
  - JobConf.setCombinerClass(Class)

範例一說明

核心Reducer

- 將Map送來的<key/value > ,對每個key作value的整合
- 輸入: <key, (list of values)>
  - Reduce (WritableComparable, Iterator, OutputCollector, Reporter)
- 輸出
  - OutputCollector.collect(WritableComparable, Writable)
- 若沒有Reduce要執行,可以不編寫

範例一說明

配置JobConf

- Hadoop程式架構內主要的執行設定類別
- 指定Mapper、Combiner、 Partitioner、Reducer、InputFormat和OutputFormat 的類別為何
- 指定輸入文件
  - setInputPaths(JobConf, Path...) / addInputPath(JobConf, Path))
- 指定輸出文件
  - setOutputPath(Path)
- debug script
  - setMapDebugScript(String) / setReduceDebugScript(String)
- 最多的嘗試次數
  - setMaxMapAttempts(int) / setMaxReduceAttempts(int))
- 容許任務失敗的百分比
  - setMaxMapTaskFailuresPercent(int) / setMaxReduceTaskFailuresPercent(int)
- ……

範例一
說明

任務執行

- runJob(JobConf)：
  - 提交作業，僅當作業完成時返回。
- submitJob(JobConf)：
  - 只提交作業，之後需要你輪詢它返回的 RunningJob 句柄的狀態，並根據情況調度。
- JobConf.setJobEndNotificationURI(String)：
  - 設置一個作業完成通知，可避免輪詢。

範例一
動手做

WordCount練習 (前置)

1. cd $HADOOP_HOME
2. bin/hadoop dfs -mkdir input
3. echo "I like NCHC Cloud Course." > input1
4. echo "I like nchc Cloud Course, and we enjoy this crouse." > input2
5. bin/hadoop dfs -put input1 input
6. bin/hadoop dfs -put input2 input
7. bin/hadoop dfs -ls input

```
waue@vPro:/opt/hadoop$ bin/hadoop dfs -ls input
Found 2 items
-rw-r--r--   1 waue supergroup         26 2009-03-22 12:15 /user/waue/input/input1
-rw-r--r--   1 waue supergroup         52 2009-03-22 12:15 /user/waue/input/input2
waue@vPro:/opt/hadoop$
```

8. 編輯WordCount.java
   http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/Hadoop_Lab6/WordCount.java?format=raw
9. mkdir MyJava

範例一
動手做

WordCount練習 (執行)

1. 編譯
   - javac -classpath hadoop-*-core.jar -d MyJava WordCount.java
2. 封裝
   - jar -cvf wordcount.jar -C MyJava .
3. 執行
   - bin/hadoop jar wordcount.jar WordCount input/ output/

- 所在的執行目錄為Hadoop_Home
- ./MyJava = 編譯後程式碼目錄
- wordcount.jar = 封裝後的編譯檔
- 先放些文件檔到HDFS上的input目錄
- ./input; ./ouput = hdfs的輸入、輸出目錄

範例一
動手做

WordCount練習 (執行)

```
waue@vPro:/opt/hadoop$ mkdir MyJava
waue@vPro:/opt/hadoop$ javac -classpath hadoop-*-core.jar -d MyJava WordCount.java
waue@vPro:/opt/hadoop$ jar -cvf wordcount.jar -C MyJava .
新增 manifest
新增：WordCount.class (讀=1516)(寫=740)(壓縮 51%)
新增：WordCount$Reduce.class (讀=1591)(寫=642)(壓縮 59%)
新增：WordCount$Map.class (讀=1918)(寫=795)(壓縮 58%)
waue@vPro:/opt/hadoop$ bin/hadoop jar wordcount.jar WordCount input/ output/
09/03/22 11:39:01 WARN mapred.JobClient: Use GenericOptionsParser for parsing the argu
ments. Applications should implement Tool for the same.
09/03/22 11:39:01 INFO mapred.FileInputFormat: Total input paths to process : 1
09/03/22 11:39:01 INFO mapred.FileInputFormat: Total input paths to process : 1
09/03/22 11:39:02 INFO mapred.JobClient: Running job: job_200903201526_0007
09/03/22 11:39:03 INFO mapred.JobClient:  map 0% reduce 0%
09/03/22 11:39:08 INFO mapred.JobClient:  map 100% reduce 0%
09/03/22 11:39:15 INFO mapred.JobClient: Job complete: job_200903201526_0007
09/03/22 11:39:15 INFO mapred.JobClient: Counters: 16
09/03/22 11:39:15 INFO mapred.JobClient:   File Systems
09/03/22 11:39:15 INFO mapred.JobClient:     HDFS bytes read=320950
09/03/22 11:39:15 INFO mapred.JobClient:     HDFS bytes written=130568
09/03/22 11:39:15 INFO mapred.JobClient:     Local bytes read=168448
09/03/22 11:39:15 INFO mapred.JobClient:     Local bytes written=336932
09/03/22 11:39:15 INFO mapred.JobClient:   Job Counters
09/03/22 11:39:15 INFO mapred.JobClient:     Launched reduce tasks=1
```

範例一
動手做

WordCount練習 (結果)

```
waue@vPro:/opt/hadoop$ bin/hadoop dfs -cat output/part-00000
Cloud   2
Course, 1
Course. 1
I       2
NCHC    1
and     1
course. 1
enjoy   1
like    2
nchc    1
this    1
we      1
```

範例二
動手做

WordCount 進階版

- WordCount2

http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/Hadoop_Lab6/WordCount2.java?format=raw

- 功能
  - 不計標點符號
  - 不管大小寫
- 步驟 (接續 WordCount 的環境)
  1. echo "\." >pattern.txt && echo "\," >>pattern.txt
  2. bin/hadoop dfs -put pattern.txt ./
  3. mkdir MyJava2
  4. javac -classpath hadoop-*-core.jar -d MyJava2 WordCount2.java
  5. jar -cvf wordcount2.jar -C MyJava2 .

範例二
動手做

不計標點符號

- 執行
  - bin/hadoop jar wordcount2.jar WordCount2 input output2 -skip pattern.txt dfs -cat output2/part-00000

```
waue@vPro:/opt/hadoop$ bin/hadoop dfs -cat output2/part-00000
Cloud   2
Course  2
I       2
NCHC    1
and     1
course  1
enjoy   1
like    2
nchc    1
this    1
we      1
```

範例二
動手做

不管大小寫

- 執行
  - bin/hadoop jar wordcount2.jar WordCount2 -Dwordcount.case.sensitive=false input output3 -skip pattern.txt

```
waue@vPro:/opt/hadoop$ bin/hadoop dfs -cat output3/part-00000
and     1
cloud   2
course  3
enjoy   1
i       2
like    2
nchc    2
this    1
we      1
```

範例二
補充說明

Tool

- 處理Hadoop命令執行的選項

  -conf <configuration file>

  -D <property=value>

  -fs <local|namenode:port>

  -jt <local|jobtracker:port>

- 透過介面交由程式處理
  - ToolRunner.run(Tool, String[])

範例二
補充說明

DistributedCache

- 設定特定有應用到相關的、超大檔案、或只用來參考卻不加入到分析目錄的檔案
  - 如pattern.txt檔
- DistributedCache.addCacheFile(URI,conf)
  - URI＝hdfs://host:port/FilePath