



財團法人國家實驗研究院

國家高速網路與計算中心

NATIONAL CENTER FOR HIGH-PERFORMANCE COMPUTING

# Map Reduce Programming

王耀聰 陳威宇 楊順發

jazz@nchc.org.tw

waue@nchc.org.tw

shunfa@nchc.org.tw

國家高速網路與計算中心(NCHC)



自由軟體實驗室

# 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

# Program Prototype



# Class Mapper(0.18)

```
1  class MyMap extends MapReduceBase
   implements Mapper < INPUT KEY , INPUT VALUE , OUTPUT KEY , OUTPUT VALUE >
2  {
3  // 全域變數區
4  public void map ( INPUT KEY key, INPUT VALUE value,
                    OutputCollector< OUTPUT KEY , OUTPUT VALUE > output,
                    Reporter reporter) throws IOException
5  {
6  // 區域變數與程式邏輯區
7  output.collect( NewKey, NewValue);
8  }
9  }
```

# Class Mapper(0.20)

```
1  class MyMap extends MapReduceBase
   implements Mapper < INPUT KEY , INPUT VALUE , OUTPUT KEY , OUTPUT VALUE >
2  {
3  // 全域變數區
4  public void map ( INPUT KEY key, INPUT VALUE value, Context context)
   throws IOException, InterruptedException
   {
5      // 區域變數與程式邏輯區
6      context.write( NewKey, NewValue);
7  }
8  }
9
```

# Class Reducer(0.18)

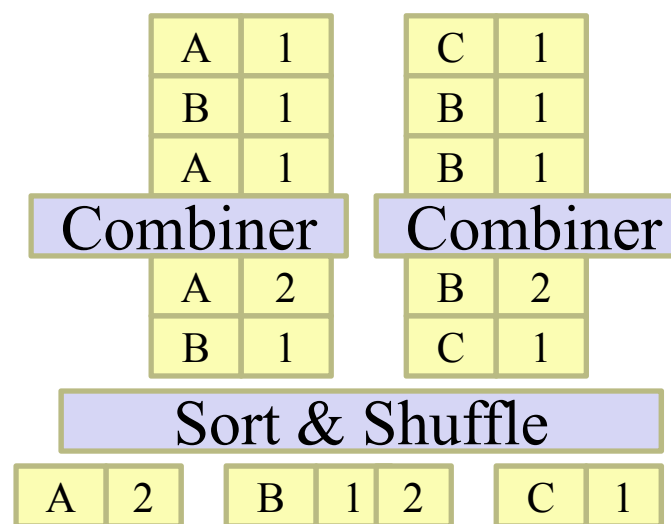
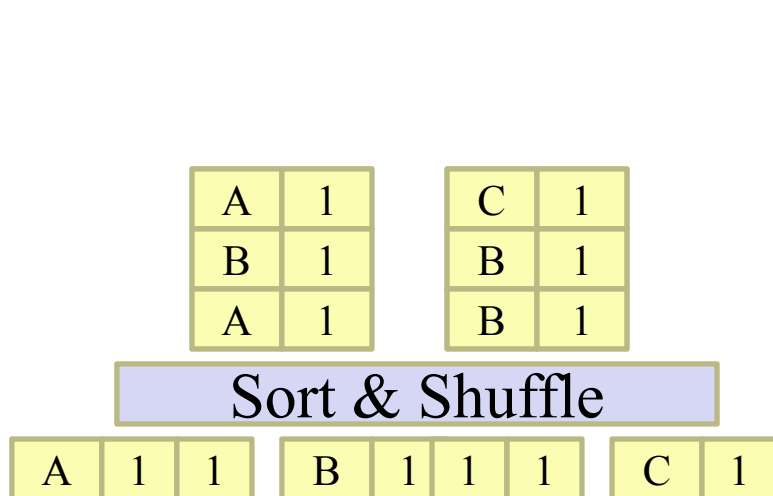
```
1  class MyRed extends MapReduceBase
   implements Reducer < INPUT KEY , INPUT VALUE , OUTPUT KEY , OUTPUT VALUE >
2  {
3  // 全域變數區
4  public void reduce ( INPUT KEY key, Iterator< INPUT VALUE > values,
   OutputCollector< OUTPUT KEY , OUTPUT VALUE > output,
   Reporter reporter) throws IOException
5  {
6  // 區域變數與程式邏輯區
7  output.collect( NewKey, NewValue);
8  }
9  }
```

# Class Reducer(0.20)

```
1 class MyRed extends MapReduceBase
  implements Reducer < INPUT KEY , INPUT VALUE , OUTPUT KEY , OUTPUT VALUE >
2 {
3   // 全域變數區
4   public void reduce ( INPUT KEY key, Iterator< INPUT VALUE > values,
      Context context) throws IOException, InterruptedException
      {
5     // 區域變數與程式邏輯區
6     output.collect( NewKey, NewValue);
7   }
8 }
9
```

# Class Combiner

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





# No Combiner

dog	1
cat	1
dog	1
dog	1

cat	1
bird	1
cat	1
dog	1

## Sort and Shuffle

dog	1	1	1	1	cat	1	1	1	bird	1
-----	---	---	---	---	-----	---	---	---	------	---

## Reduce class

### Reduce method (key, values)

```
for each v in values
    sum ← sum + v
    EMIT(key, sum)
```

# Added Combiner

dog	1
cat	1
dog	1
dog	1

Combine

cat	1
dog	3

cat	1
bird	1
cat	1
dog	1

Combine

bird	1
cat	2
dog	1

Sort and Shuffle

dog	3	1
-----	---	---

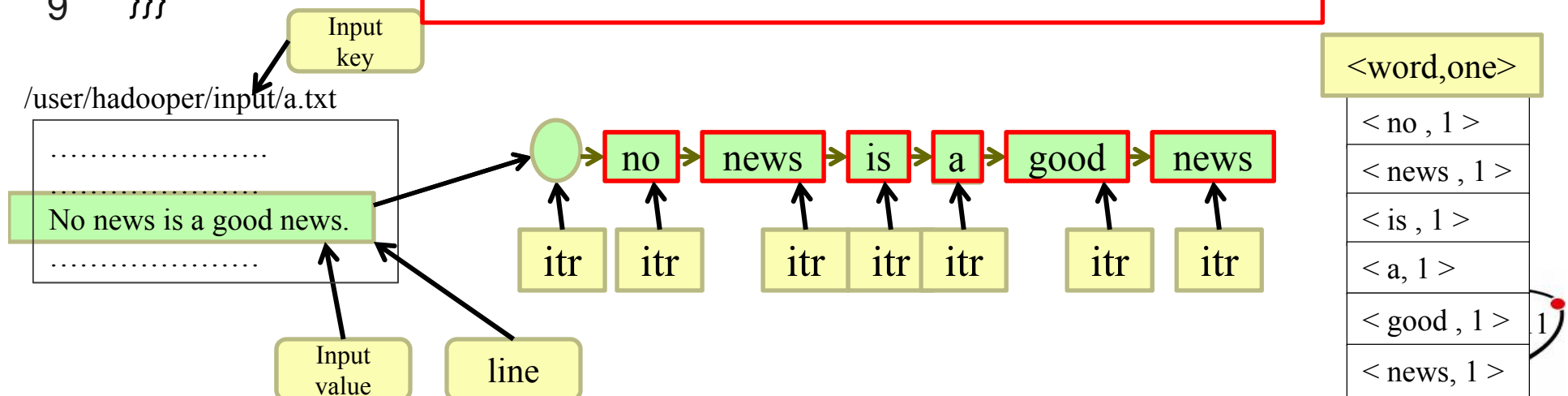
cat	1	2
-----	---	---

bird	1
------	---

## 範例 程式一

# Word Count Sample (1)

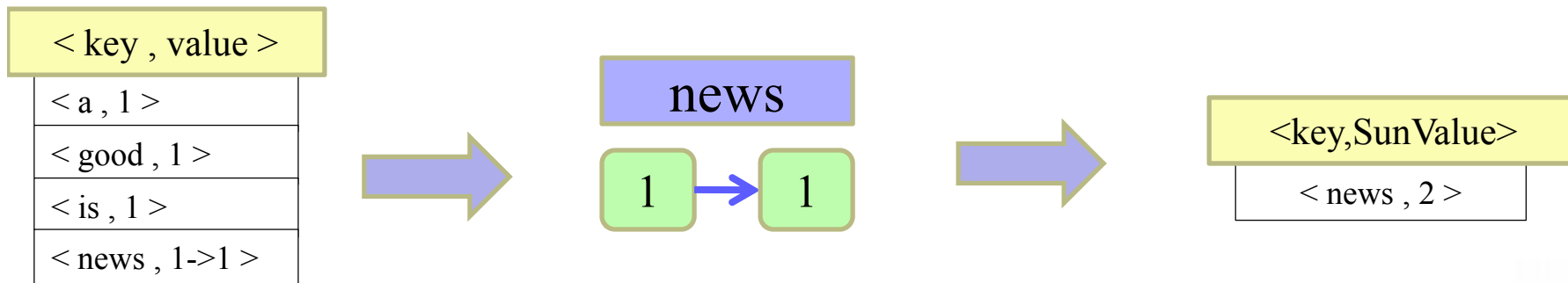
```
1 class MapClass extends MapReduceBase implements  
  Mapper<LongWritable, Text, Text, IntWritable> {  
2     private final static IntWritable one = new IntWritable(1);  
3     private Text word = new Text();  
4     public void map( LongWritable key, Text value,  
        Context context) throws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



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,  
        Context context)  
        throws IOException {  
4         int sum = 0;  
5         while (values.hasNext())  
6             sum += values.next().get();  
7         SumValue.set(sum);  
8         context.write(key, SumValue);  
    }  
}
```



## Word Count Sample (3)

```
Class WordCount{  
    main()
```

```
        JobConf conf = new JobConf(WordCount.class);  
        conf.setJobName("wordcount");  
        // set path  
        FileInputFormat.setInputPaths(new Path(args[0]));  
        FileOutputFormat.setOutputPath(new Path(args[1]));  
        // set map reduce  
        conf.setMapperClass(MapClass.class);  
        conf.setCombinerClass(Reduce.class);  
        conf.setReducerClass(ReduceClass.class);  
        // run  
        JobClient.runJob(conf);
```

```
}}
```

# 編譯與執行

## 1. 編譯

- `javac`  $\Delta$  `-classpath`  $\Delta$  `hadoop-*-core.jar`  $\Delta$  `-d`  $\Delta$  `MyJava`  $\Delta$   
`MyCode.java`

## 2. 封裝

- `jar`  $\Delta$  `-cvf`  $\Delta$  `MyJar.jar`  $\Delta$  `-C`  $\Delta$  `MyJava`  $\Delta$  `.`

## 3. 執行

- `bin/hadoop`  $\Delta$  `jar`  $\Delta$  `MyJar.jar`  $\Delta$  `MyCode`  $\Delta$  `HDFS_Input/`  
 $\Delta$  `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

## WordCount1 練習 (I)

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

## WordCount1 練習 (II)

1. 編輯 WordCount.java  
[http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/Hadoop\\_Lab6/WordCount.java?format=raw](http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/Hadoop_Lab6/WordCount.java?format=raw)
2. mkdir MyJava
3. javac -classpath hadoop-\*-core.jar -d MyJava  
WordCount.java
4. jar -cvf wordcount.jar -C MyJava .
5. bin/hadoop jar wordcount.jar WordCount input/ output/

- 
- 所在的執行目錄為Hadoop\_Home ( 因為hadoop-\*-core.jar )
  - javac編譯時需要classpath, 但hadoop jar時不用
  - wordcount.jar = 封裝後的編譯檔, 但執行時需告知class name
  - Hadoop進行運算時, 只有 input 檔要放到hdfs上, 以便hadoop分析運算; 執行檔 (wordcount.jar) 不需上傳, 也不需每個node都放, 程式的載入交由java處理



## 範例一 動手做

# WordCount1 練習(III)

```
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 arguments. 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

## WordCount1 練習 (IV)

```
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
```

# WordCount 進階版

- WordCount2

[http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/Hadoop\\_Lab6/WordCount2.java?format=raw](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 .`

## 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
```

## 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
```

# 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[])

## 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=hdbs://host:port/FilePath

非JAVA  
不可？

## Options without Java

- 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