MapReduce介紹

觀念與程式設計

趨勢科技研發實驗室

Public 2009/5/13

MapReduce 由來

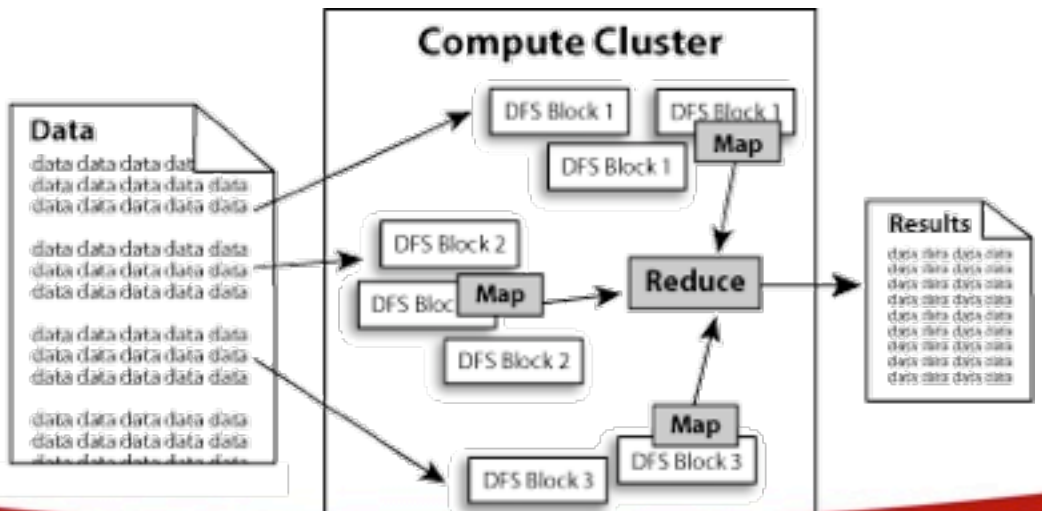
- Google 需要進行大規模資料處理
- Google 工程師發現，處大量資料時會面臨某些共同課題
  - 需要使用許多機器協同計算
  - 處理輸入資料時有兩項基本作業：**Map** 與 **Reduce**。
- Google 的概念是受到函數編程當中 ***map*** 與 ***reduce*** 函數的啟發
- 函數編程： MapReduce
  - Map
    - [1,2,3,4] – (*2) → [2,3,6,8]
  - Reduce
    - [1,2,3,4] – (sum) → 10

– 分而治之、個別擊破 **(Divide and Conquer)** 典範

Copyright 2009 - Trend Micro Inc.

MapReduce概觀

- MapReduce 是由 Google 所引進的軟體框架，目的是對電腦叢集上的大型資料集執行分散式運算。讓使用者可以把心力放在定義Map和Reduce函數。MapReduce框架會協調機器資源配置並處理的程式輸入、輸入與執行。
- 使用者指定
  - *Map* 函數，此函數的輸入是"一個" key/value 序對 ，輸出則為另"一組" intermediate key/value 序對組。
  - *Reduce* 函數，此函數負責針對相同的 intermediate key 合併其所有相關聯的 intermediate values。並產生輸出結果的key/value序對
- 執行細節交由 MapReduce框架處理。



Copyright 2009 - Trend Micro Inc.

應用領域

- 範例
  - 大型資料處理，例如：搜尋、索引製作與排序
  - 大型資料集的資料採礦與機器學習
  - 大型網站的網站存取日誌分析

- 更多應用

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

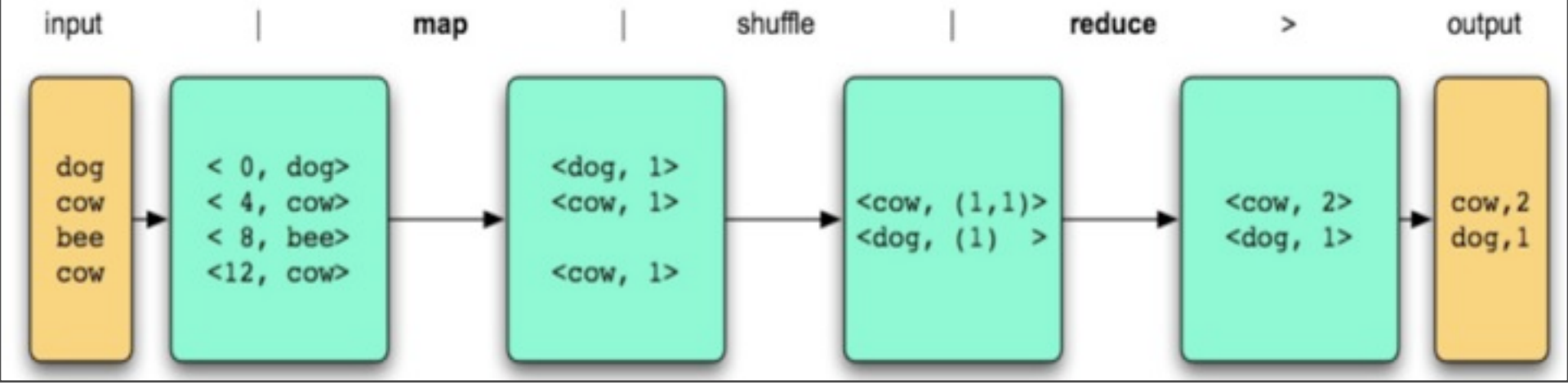
Copyright 2009 - Trend Micro Inc.

MapReduce 程式設計模型

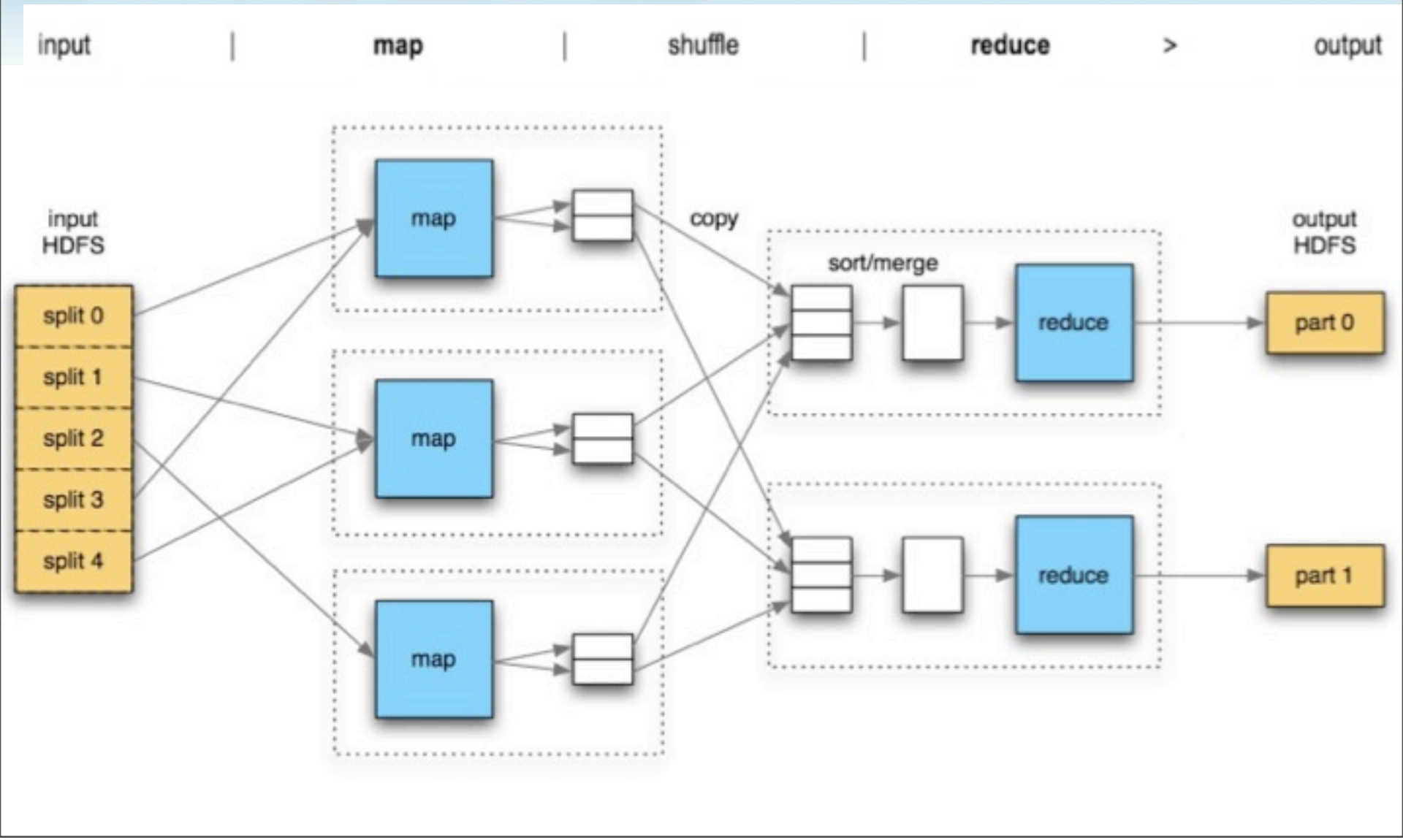
- 使用者定義兩項函數
  - map (K1, V1) → list(K2, V2)
  - reduce (K2, list(V2)) → list(K3, V3)

- grep 範例
  - Map: (offset, line) → [(match, 1)]
  - Reduce: (match, [1, 1, ...]) → [(match, n)]

- MapReduce 邏輯流程:



MapReduce 實際運作流程



Word Count 範例

字數問題：
試想如何在大量的文件集合中，計算每個word的出現次數。

- Map：(offset, line) → [(word1, 1), (word2, 1), …]
- Reduce：(word, [1, 1, …]) → [(word, n)]

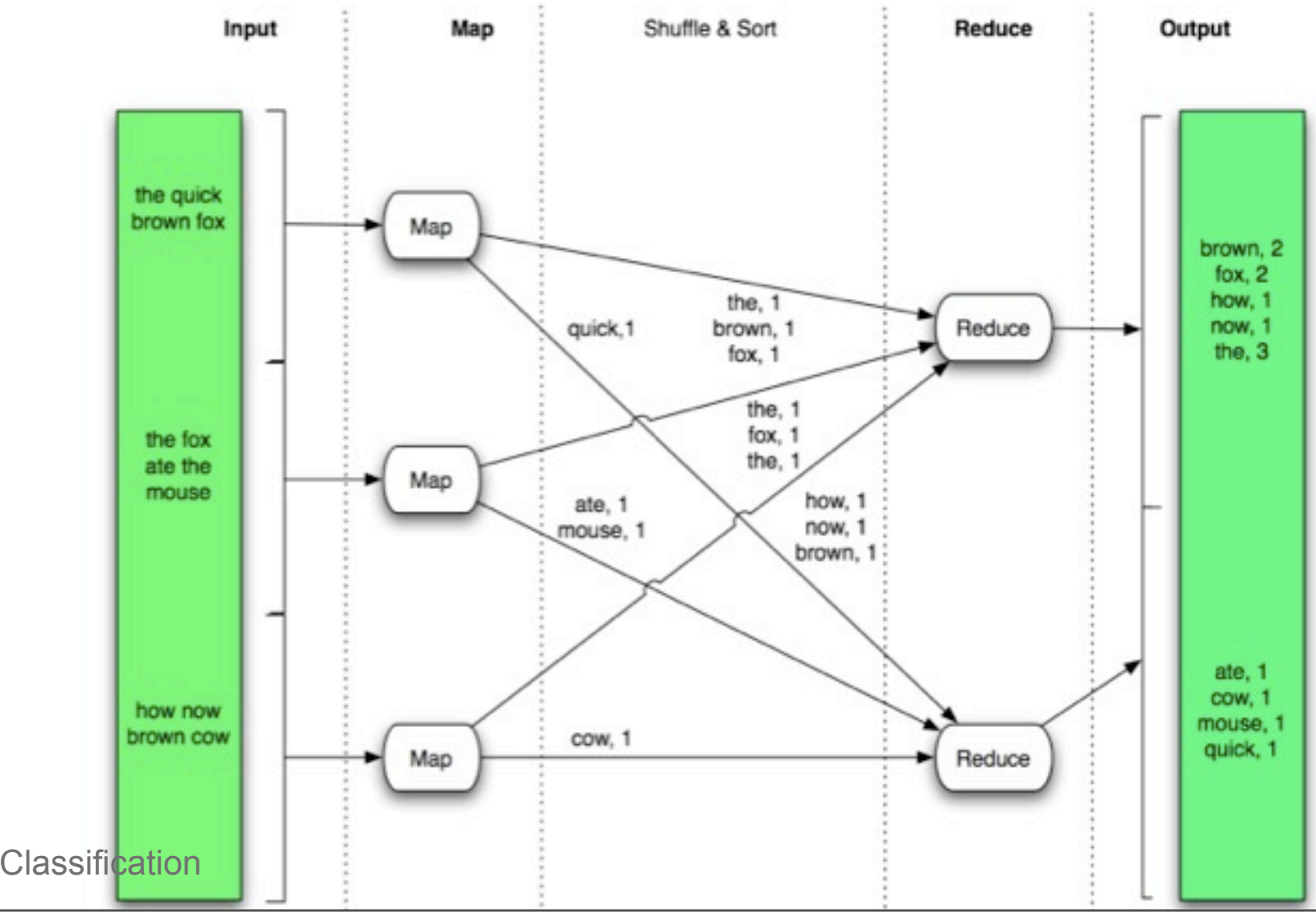
Map, Reduce函式虛擬碼

```
map(String key, String value):
  // key: document name
  // value: document contents
  for each word w in value:
    EmitIntermediate(w, "1");
```

```
reduce(String key, Iterator values):
  // key: a word
  // values: a list of counts
  int result = 0;
  for each v in values:
    result += ParseInt(v);
  Emit(AsString(result));
```

Copyright 2009 - Trend Micro Inc.

Word Count 實際運作流程



更多 MapReduce 範例

- 分散式擷取 (Distributed Grep)
  - 大量文字中找尋符合指定樣式 (pattern) 的資料行
- 分散式排序 (Distributed Sort)
  - 為大量的值進行排序
- 計算 URL 的存取頻率 (Count of URL Access Frequency)
  - 計算 Web 要求日誌當中的 URL 存取頻率，特別是在日誌檔規模龐大與日誌檔數目眾多的情況下

Copyright 2009 - Trend Micro Inc.

MapReduce 程式設計簡介

Copyright 2007 - Trend Micro Inc.

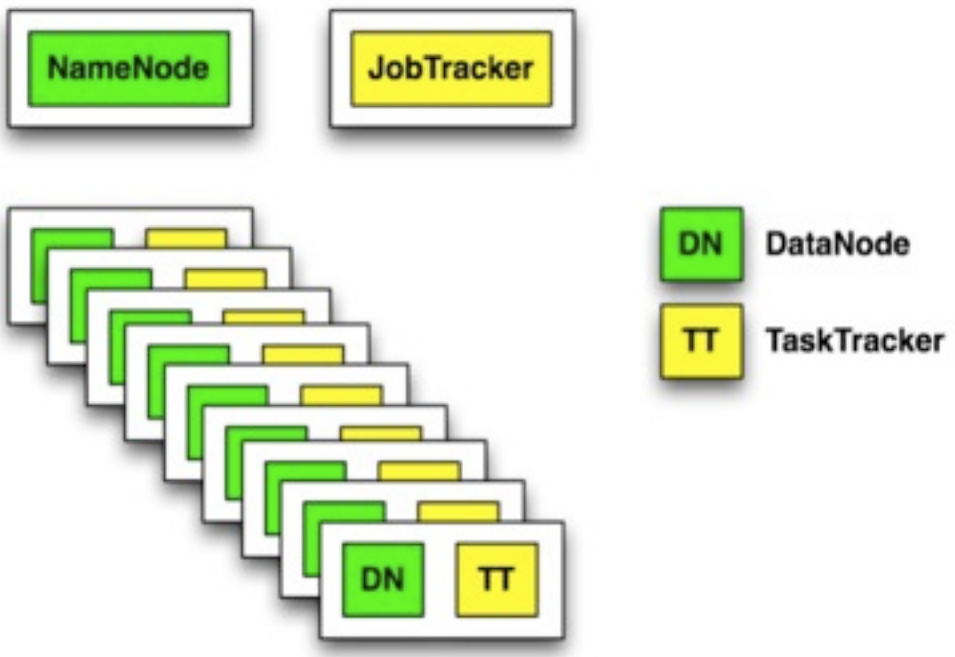
Hadoop 的 MapReduce

- Apache Hadoop 實作 Google 的 MapReduce
  - 提供開放原始碼 MapReduce 架構
  - 以 Java 做為原生語言
  - 以 Hadoop 分散式檔案系統 (HDFS) 做為資料儲存系統
- Yahoo! 是主要的贊助者
- Google, Yahoo!, IBM, Amazon 等等 都使用 Hadoop
- 趨勢科技 (Trend Micro) 運用 Hadoop MapReduce 在大量可疑的電腦行為記錄中進行分析，並找出解決方案

Copyright 2009 - Trend Micro Inc.

Hadoop MapReduce 架構圖

- Map/Reduce framework的系統服務
  - JobTracker
  - TaskTracker
- JobTracker
  - Job的排程作業
  - 監控Job的狀況。若發生問題，JobTracker會重新傳送發生問題的Job.
- TaskTrackers
  - 實作執行Job



Classification
Copyright 2009 - Trend Micro Inc.

一個Hadoop MapReduce程式的基本框架

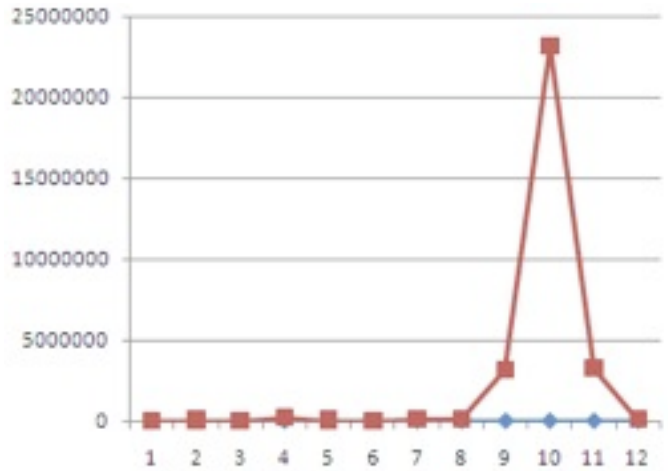

```
class MyJob {
        class Map {              // 定義 Map 程式碼
        }
        class Reduce {           // 定義 Reduce 程式碼
        }

        }
        main() {
                // 設定 job 的相關參數
                 JobConf conf = new JobConf("MyJob.class");
                  conf.setInputPath(…);
                  conf.setOutputPath(…);
                  conf.setMapperClass(Map.class);
                  conf.setReduceClass(Reduce.class)
                // 執行Job
                  JobClient.runJob(conf);
        }
}
```

待解決問題簡介

- 背景
  - 趨勢科技研究部門每天都會收到大量的使用者回饋日誌
  - 每個日誌都包含趨勢科技產品所偵測到的安全威脅相關事件之資訊
  - 研發部門的同仁運用採礦、關聯法與機器學習等方式分析回饋日誌以偵測更多新病毒
  - 因為日誌資料量非常大，無法用既有的儲存系統來儲存。因此研究部門使用 HDFS 來儲存日誌資料。並使用MapReduce進行研究分析的工作。
- 問題
  - 計算每個小時的回饋日誌數量的分佈情況
  - 日誌格式
    - 每一行為一筆記錄
    - 每一筆記錄的格式如下

行數, GUID, 時間,偵測模組, 病毒行為
1, 123, 131231231, VSAPI, open file
2, 456, 123123123, VSAPI, connect internet



Copyright 2007 - Trend Micro Inc.

定義 Map 函數

- 實作 Mapper 介面並且定義 map() 方法
- Map : (K1, V1) → list(K2, V2)

map(WritableComparable, Writable,
OutputCollector, Reporter)

- 每一對 input ，框架都會呼叫一次 map() 函數以處理之
- OutputCollector 使用 collect() method 來收集立即結果

OutputCollector.collect(WritableComparable,Writable)

Copyright 2007 - Trend Micro Inc.

定義 Map 函數


```
class MapClass extends MapReduceBase

implements Mapper<LongWritable, Text, Text, IntWritable> {

        private final static IntWritable one = new IntWritable(1);

        private Text hour = new Text();

        public void map( LongWritable key, Text value,
OutputCollector<Text,IntWritable> output, Reporter reporter) throws
IOException {

              String line = ((Text) value).toString();

                String[] token = line.split(",");

                String timestamp = token[1];

                Calendar c = Calendar.getInstance();

                c.setTimeInMillis(Long.parseLong(timestamp));

                Integer h = c.get(Calendar.HOUR);

                hour.set(h.toString());

                output.collect(hour, one)

}}}
```

定義 *Reduce* 函數

- 實作 Reducer 介面，並定義 reduce() method
- Reduce : (K2, list(V2)) → list(K3, V3)

```
reduce (WritableComparable, Iterator,
        OutputCollector, Reporter)
```

- OutputCollector 使用 collect() method 來收集最終結果

```
OutputCollector.collect( WritableComparable,Writable )
```

Copyright 2007 - Trend Micro Inc.

定義 *Reduce* 函數


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

Copyright 2007 - Trend Micro Inc.

設定與執行您的工作

- JobConf 類別用於設定
  - Mapper、Reducer、Inputformat、OutputFormat、Combiler Petitioner 等等
  - 輸入路徑
  - 輸出路徑
  - 其他的工作配置
    - map 與 reduce 作業的失敗百分比
    - 發生錯誤時的重試數目

- 執行您的工作時，JobClient 類別會以 JobConf 做為配置

```
JobClient.runJob(conf);
JobClient.submitJob(conf);
JobClient.setJobEndNotificationURI(URI);
```

Copyright 2007 - Trend Micro Inc.

Main Function


```
Class MyJob{
public static void main(String[] args) {
        JobConf conf = new JobConf(MyJob.class);
        conf.setJobName(”Caculate feedback log time distribution");
        // 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);
}}
```

Copyright 2007 - Trend Micro Inc.

編譯與執行

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

Copyright 2007 - Trend Micro Inc.

執行過程

- bin/hadoop jar MyJob.jar MyJob input/ output/

```
09/05/16 17:39:58 INFO mapred.FileInputFormat: Total input paths to process : 13
09/05/16 17:39:58 INFO mapred.JobClient: Running job: job_200905161736_0001
09/05/16 17:39:59 INFO mapred.JobClient:  map 0% reduce 0%
09/05/16 17:40:09 INFO mapred.JobClient:  map 7% reduce 0%
09/05/16 17:40:19 INFO mapred.JobClient:  map 23% reduce 0%
09/05/16 17:40:37 INFO mapred.JobClient:  map 38% reduce 0%
09/05/16 17:40:39 INFO mapred.JobClient:  map 53% reduce 0%
09/05/16 17:40:40 INFO mapred.JobClient:  map 61% reduce 0%
09/05/16 17:40:42 INFO mapred.JobClient:  map 61% reduce 7%
09/05/16 17:40:43 INFO mapred.JobClient:  map 76% reduce 7%
09/05/16 17:40:45 INFO mapred.JobClient:  map 92% reduce 7%
09/05/16 17:40:46 INFO mapred.JobClient:  map 100% reduce 7%
09/05/16 17:40:48 INFO mapred.JobClient:  map 100% reduce 12%
09/05/16 17:40:58 INFO mapred.JobClient:  map 100% reduce 100%
09/05/16 17:41:00 INFO mapred.JobClient: Job complete: job_200905161736_0001
09/05/16 17:41:00 INFO mapred.JobClient: Counters: 19
09/05/16 17:41:00 INFO mapred.JobClient:   Job Counters
09/05/16 17:41:00 INFO mapred.JobClient:     Launched reduce tasks=1
09/05/16 17:41:00 INFO mapred.JobClient:     Rack-local map tasks=5
09/05/16 17:41:00 INFO mapred.JobClient:     Launched map tasks=13
09/05/16 17:41:00 INFO mapred.JobClient:     Data-local map tasks=8
09/05/16 17:41:00 INFO mapred.JobClient:   FileSystemCounters
09/05/16 17:41:00 INFO mapred.JobClient:     FILE_BYTES_READ=177
09/05/16 17:41:00 INFO mapred.JobClient:     HDFS_BYTES_READ=20635
09/05/16 17:41:00 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=842
09/05/16 17:41:00 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=111
09/05/16 17:41:00 INFO mapred.JobClient:   Map-Reduce Framework
09/05/16 17:41:00 INFO mapred.JobClient:     Reduce input groups=1
09/05/16 17:41:00 INFO mapred.JobClient:     Combine output records=9
09/05/16 17:41:00 INFO mapred.JobClient:     Map input records=613
09/05/16 17:41:00 INFO mapred.JobClient:     Reduce shuffle bytes=249
09/05/16 17:41:00 INFO mapred.JobClient:     Reduce output records=1
09/05/16 17:41:00 INFO mapred.JobClient:     Spilled Records=18
09/05/16 17:41:00 INFO mapred.JobClient:     Map output bytes=1564
09/05/16 17:41:00 INFO mapred.JobClient:     Map input bytes=20635
09/05/16 17:41:00 INFO mapred.JobClient:     Combine input records=92
09/05/16 17:41:00 INFO mapred.JobClient:     Map output records=92
09/05/16 17:41:00 INFO mapred.JobClient:     Reduce input records=9
```

Classification

Copyright 2009 - Trend Micro Inc.

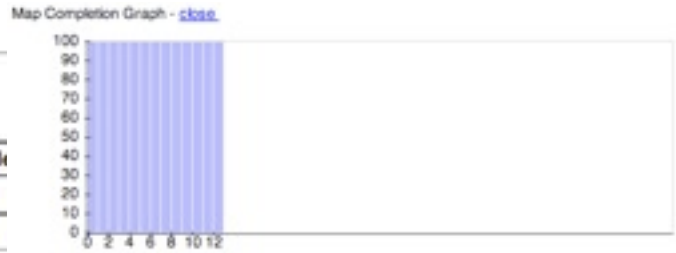
執行過程與結果可以用Web Console來檢視

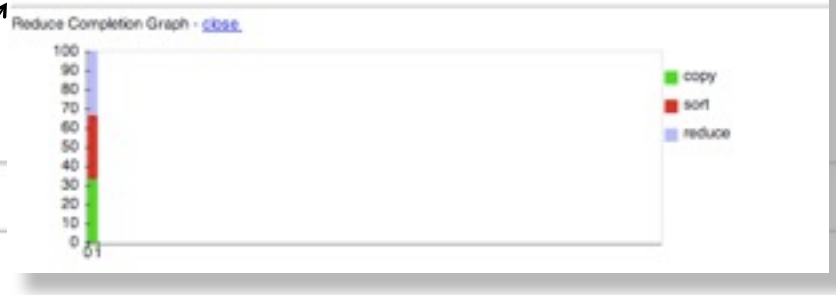
http://172.16.203.132:50030/

State: RUNNING
Started: Sat May 16 17:36:12 CST 2009
Version: 0.20.0, r763504
Compiled: Thu Apr 9 05:18:40 UTC 2009 by ndaley
Identifier: 200905161736

Cluster Summary (Heap Size is 4.94 MB/198.5 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node
0	1	4	2	4	4	4.00





Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_200905161736_0004	NORMAL	root	grep-search	100.00%	13	13	25.64%	1	0	NA

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_200905161736_0001	NORMAL	root	grep-search	100.00%	13	13	100.00%	1	1	NA
job_200905161736_0002	NORMAL	root	grep-sort	100.00%	1	1	100.00%	1	1	NA
job_200905161736_0003	NORMAL	root	grep-search	100.00%	13	13	100.00%	1	1	NA

Failed Jobs

none

Local Logs

Hadoop MapReduce補充

- Mapper 有多少個?
  - Mapper的數量由Input的大小決定。Input資料被Hadoop切割成幾份就會有多少個Mapper。
  - 可以用JobConf物件的***setNumMapTasks(int)***來提示***Hadoop***關於***Mapper***的數量。但決定權仍由***Hadoop***控制。

- Reducer有多少個?
  - 可以用JobConf物件的***JobConf.setNumReduceTasks(int)***來直接控制***Reducer***的數量。決定權可由使用者控制 。
  - 增加Reducer的數量會讓系統的負荷增加，但增加Reducer數量也可以讓整個MapReduce的Map與Reduce的工作平衡有好的成效，同時也增加容錯性。

Copyright 2009 - Trend Micro Inc.

Non-Java Interface

- Hadoop Pipes
  - 為 MapReduce 提供 C++ API 與程式庫
  - C++ 應用程式會以 java 作業子程序的身分執行
- Hadoop Streaming
  - 可以讓使用者在MapReduce 中用外部執行檔來定義其工作

Copyright 2007 - Trend Micro Inc.

參考資料

- **Google MapReduce 論文**
  - http://labs.google.com/papers/mapreduce.html
- **Google： 一週內搞定 MapReduce**
  - http://code.google.com/edu/submissions/mapreduce/listing.html
- **Google： 叢集運算與 MapReduce**
  - http://code.google.com/edu/submissions/mapreduce-minilecture/listing.html
- **Hadoop 專案官方網站**
  - http://hadoop.apache.org/core/

Copyright 2007 - Trend Micro Inc.

參考資料

- 適用於 **Eclipse** 外掛程式的 **MapReduce** 工具 **(IBM** 提供**)**
  - 可為 Eclipse 平台提供 Hadoop 支援的穩固外掛程式
  - 網址
    - http://code.google.com/edu/parallel/tools/hadoopvm/hadoop-eclipse-plugin.jar
- **Hadoop** 虛擬映像檔 **(Google** 提供**)**
  - 其 VMware 映像檔包含預先配置的 Hadoop 單一節點實例， 可提供和完整叢集完全相同的介面，但不會耗用任何資源， 適合想要探索平台的教育人員，以及想獨立工作的學生使用。 下列下載項目與 VMware 播放程式鏈結，將指向 Google 外部網站
  - 網址
    - http://code.google.com/edu/parallel/tools/hadoopvm/index.html

Copyright 2007 - Trend Micro Inc.