Hadoop 與 Hadoop 分散式檔案系統 (HDFS)

趨勢科技研發實驗室

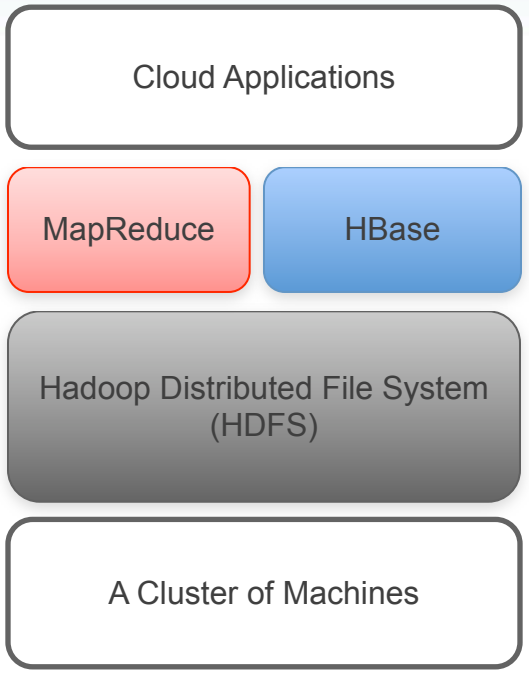
Public 2009/5/13

課程大綱

- Hadoop 簡介
- Hadoop 分散式檔案系統 (HDFS) 簡介
  - 架構
  - 管理
  - 用戶端介面
- 參考資料

Hadoop是什麼?

- Hadoop 是用來處理與保存大量資料的雲端運算平台
- Apache top-level 專案
- Hadoop 包含
  - 分散式檔案系統 (HDFS)
  - MapReduce 框架
- 使用 Java開發
- 用戶端則提供 C++/Java/Shell/ Command…等介面
- 執行於
  - Linux、Mac OS/X、Windows 和 Solaris
  - 一般商用等級的伺服器



Hadoop 簡史

- 2003 年 2 月
  - Google 撰寫第一個 MapReduce 程式庫
- 2003 年 10 月
  - Google 發表Goofle File System (GFS) 論文
- 2004 年 12 月
  - Google 發表 MapReduce 論文
- 2005 年 7 月
  - Doug Cutting 公佈 Nutch 開始採用全新的 MapReduce 實作
- 2006 年 2 月
  - Hadoop 程式碼從 Nutch 移至全新的 Lucene 子專案
- 2006 年 11 月
  - Google 發表 Bigtable 論文

Hadoop 簡史

- 2007 年 2 月
  - Mike Cafarella 發佈第一個 Hbase 程式碼
- 2007 年 4 月
  - Yahoo! 在 1000 個節點叢集上執行 Hadoop
- 2008 年 1 月
  - Hadoop 成為 Apache 頂層專案

Who use Hadoop?

- Yahoo!
  - Hadoop部署於2萬多台伺服器上，CPU數量超過10萬個。
- Google
  - 於校園中使用Hadoop推廣雲端運算相關的概念。
- Amazon
  - Amazon 使用Hadoop建置產品搜尋引擎的索引。
  - 每日處理數百萬個分析個案
- IBM
  - Blue Cloud 雲端運算叢集
- Trend Micro
  - 使用Hadoop來保存與並分由趨勢產品所傳送回來的大量可疑的病毒行為記錄檔。
- 更多使用Hadoop的公司…
  - http://wiki.apache.org/hadoop/PoweredBy

Hadoop 分散式檔案系統(HDFS)

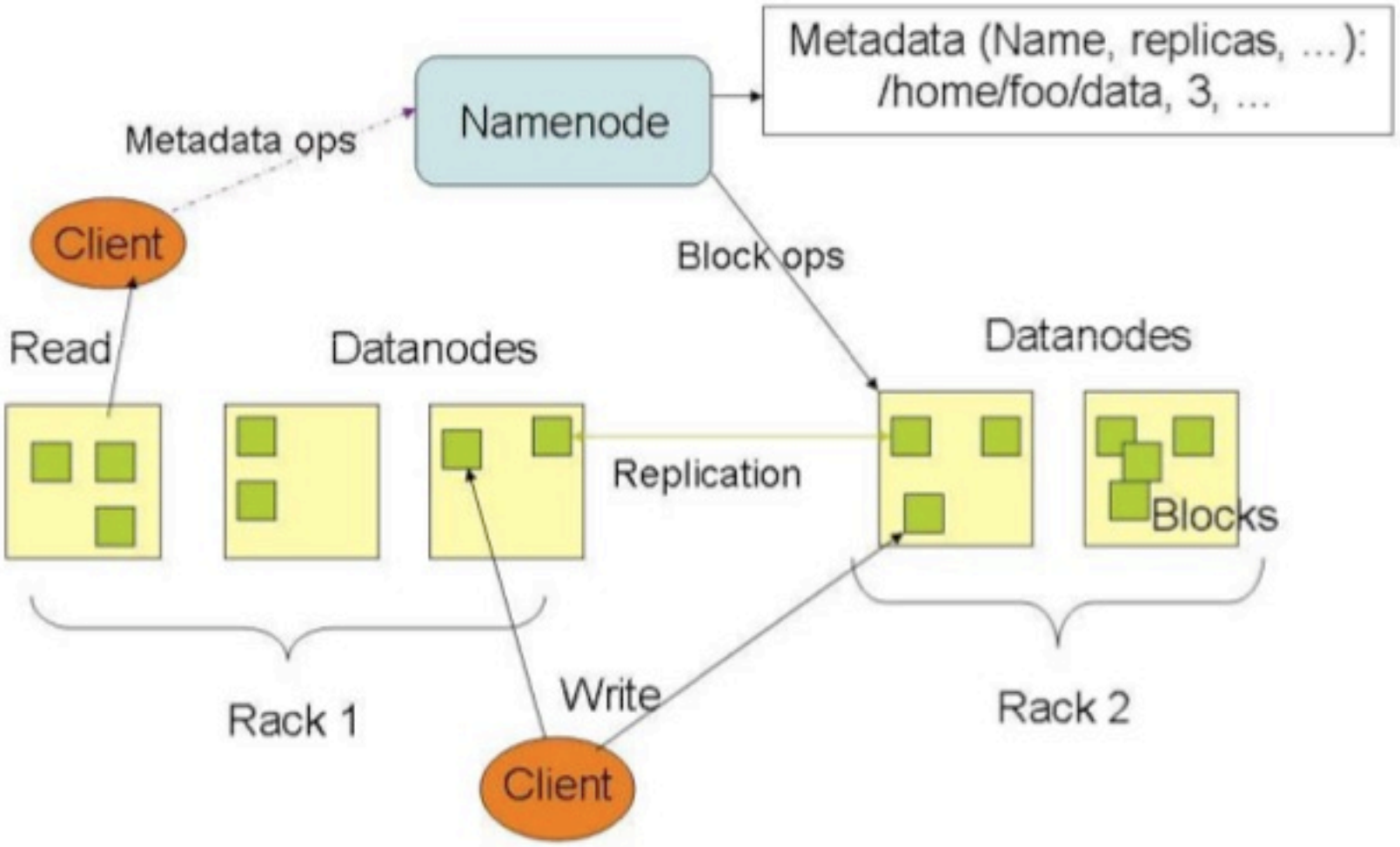
HDFS的設計理念

- 在分散式的儲存環境裏，提供單一的目錄系統 (Single Namespace)
- 超大型分散式檔案系統
  - 1 萬個節點；1 億個檔案；10 Peta Bytes的資料量
- 資料存取特性
  - Write-once-read-many 存取模式
  - 檔案一旦建立、寫入，就不允許修改
- 每個檔案被分割成許多區塊(block) 與異地備份
  - 每個區塊的大小通常為 128 MB
  - 系統會將每個區塊複製許多複本(replica)並分散儲存於不同的 資料節點(DataNode) 上

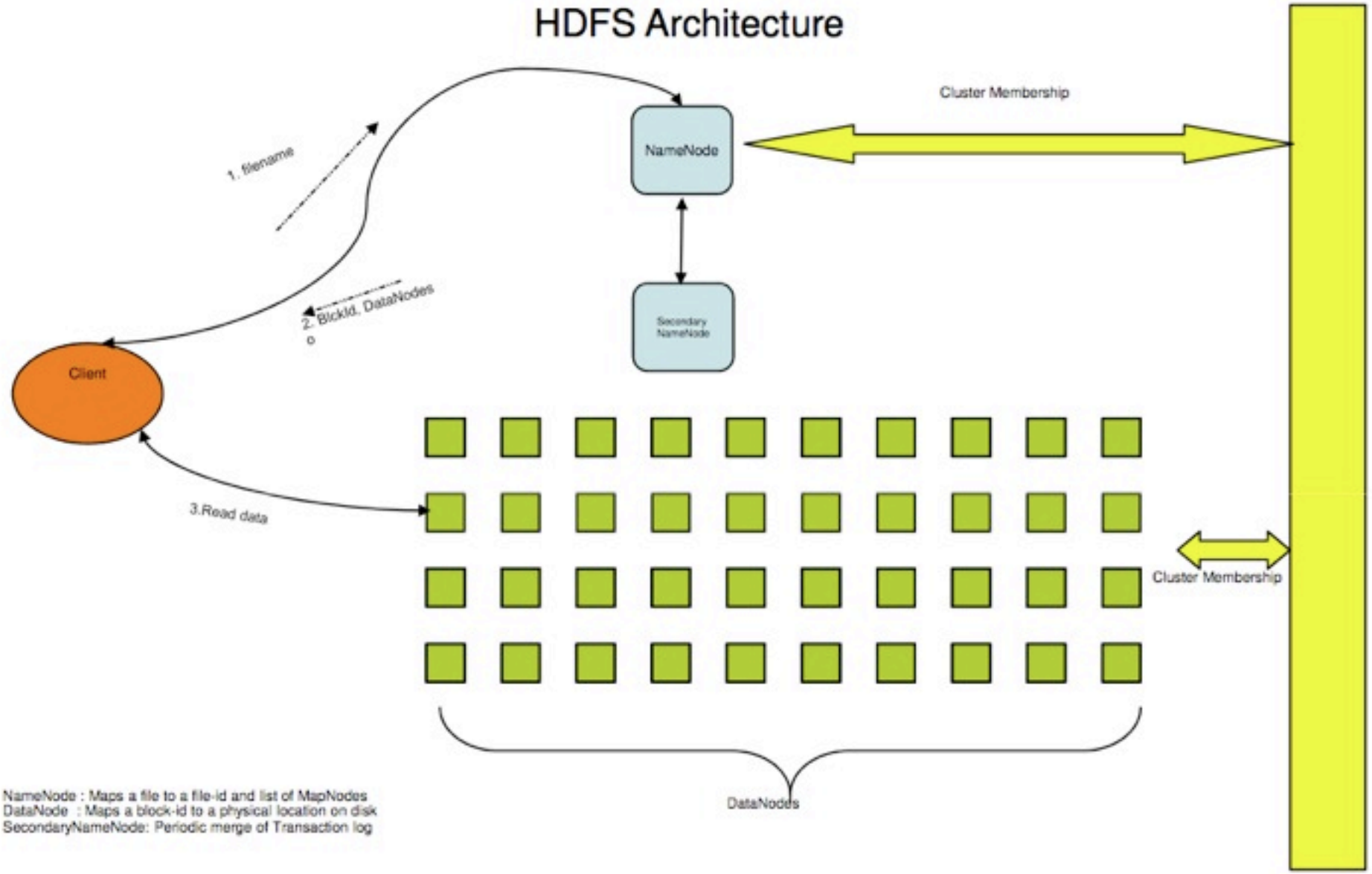
HDFS的設計理念

- 移動運算到資料端比移動資料到運算端來的成本低
  - 由於資料的位置資訊有被考慮，因此運算作業可以移至資料所在位置
- 檔案複本 (File replication)
  - 預設是每個檔案儲存 3 份.
  - 用戶端可自訂
- 採用一般伺服器
  - 採用複製資料以因應硬體的故障
  - 當偵測錯誤時，從複製的資料執行資料回復
- 串流式資料存取
  - 優先考慮大量資料存取行為，而非低延遲(low latency)資料存取行為。
  - 批次處理(Batch processing)最佳化

HDFS Architecture



HDFS Architecture



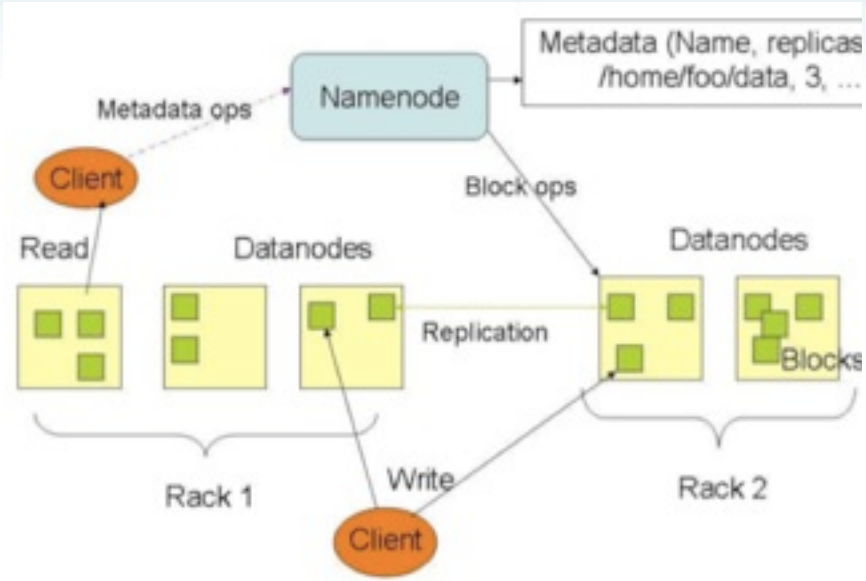
NameNode : Maps a file to a file-id and list of MapNodes
DataNode : Maps a block-id to a physical location on disk
SecondaryNameNode: Periodic merge of Transaction log

名稱節點 (NameNode)

- NameNode負責維護HDFS的檔案名稱空間 (File System Namespace)
  - 記錄檔案與其區塊(blocks)的對映關係
  - 記錄區塊(block)與區塊所在的 Data Node。
- Hadoop cluster的組態管理
- 檔案區塊的備份管理

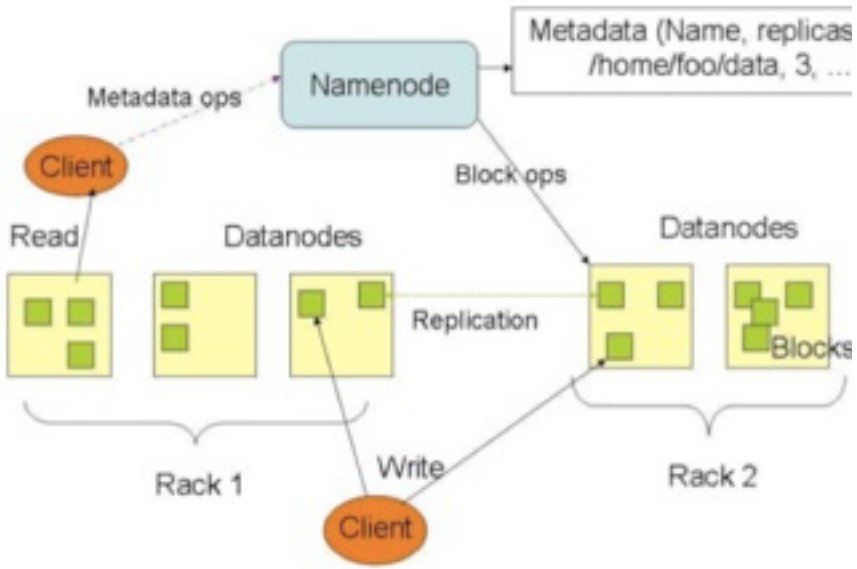
NameNode 中介資料 (Metadata)

- **Name node的Metadata儲存於記憶體中**
  - 完整的Metadata都位於主記憶體中
  - 無須任何的虛擬記憶體分頁的動作。
- **Metadata包含的資訊**
  - 檔案名稱 (files)
  - 檔案與其區塊(blocks)的對映關係
  - 區塊(block)與區塊所在的資料節點(Data Node)
  - 檔案的屬性
    - 例如: 建立時間(creation time), 複本數量 (replication factor)



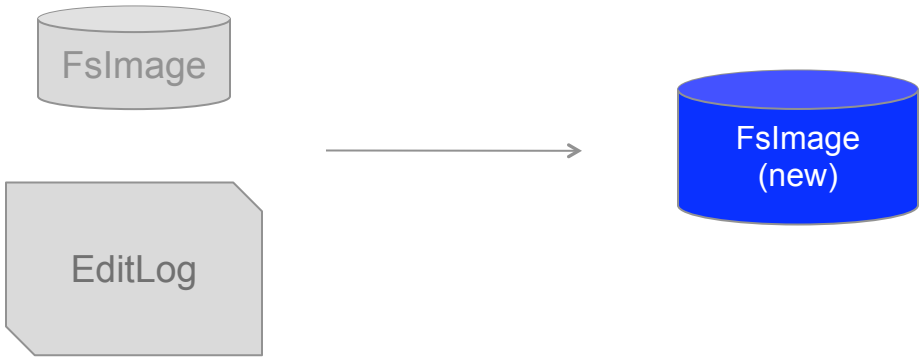
NameNode 中介資料 (Metadata)

- 交易日誌 (稱為 *EditLog*)
  - 記錄檔案建立、檔案刪除等動作。
- *FsImage*
  - **Name Node主要資料的完整映像檔。包含以下資訊**
    - 完整名稱空間 (Name Space)
    - 區塊 (Block) 與檔案 (File) 之間的對映
    - 檔案的屬性
  - NameNode 可以設定成維護多份 *FsImage* 與 *EditLog*
- **Checkpoint時機點**
  - 於 NameNode 啟動時執行
  - 從磁碟讀取 *FsImange* 與 *EditLog*，並將 *EditLog* 中的所有異動套用至讀取出來的*FsImange*



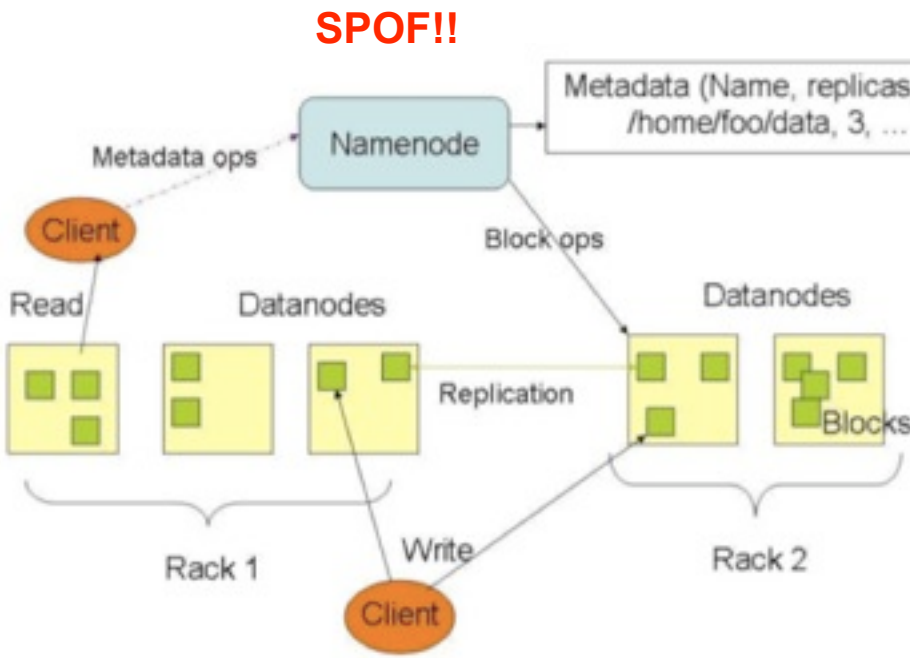
次要名稱節點 (Secondary NameNode)

- 將NameNode的 *FsImage* 與 *EditLog* 從 NameNode 複製到暫存目錄。
- 將 FSImage 與 *EditLog* 合併，並產生新的 *FSImage*
- 將新的 *FSImage* 上傳至 NameNode
  - NameNode 中的*EditLog* 則會被清除
- Secondary NameNode並非NameNode的備援(Fail over)。
  - Hadoop目前尚未支援Name Node的備援機制。



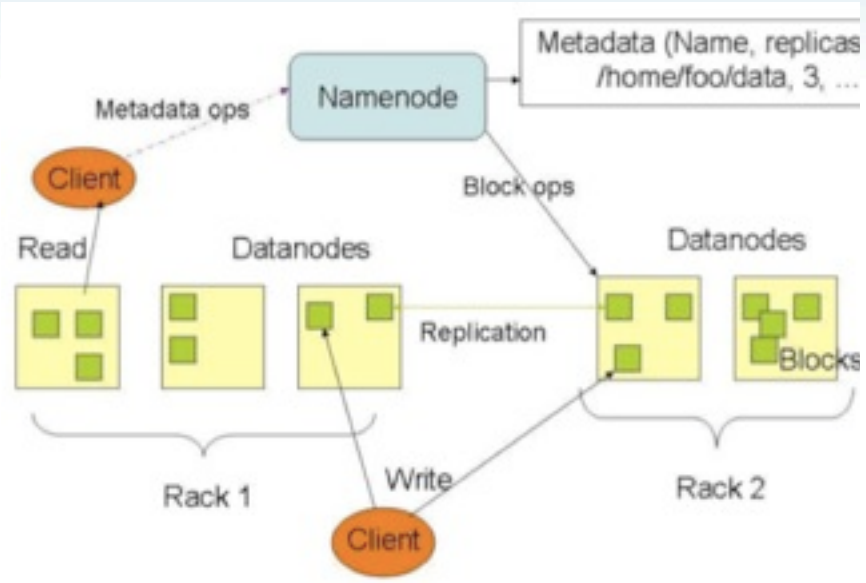
NameNode的異常

- 目前NameNode仍為一個 SPOF (single point of failure)
- 需要實作高可用性(High Availablity)的解決方案



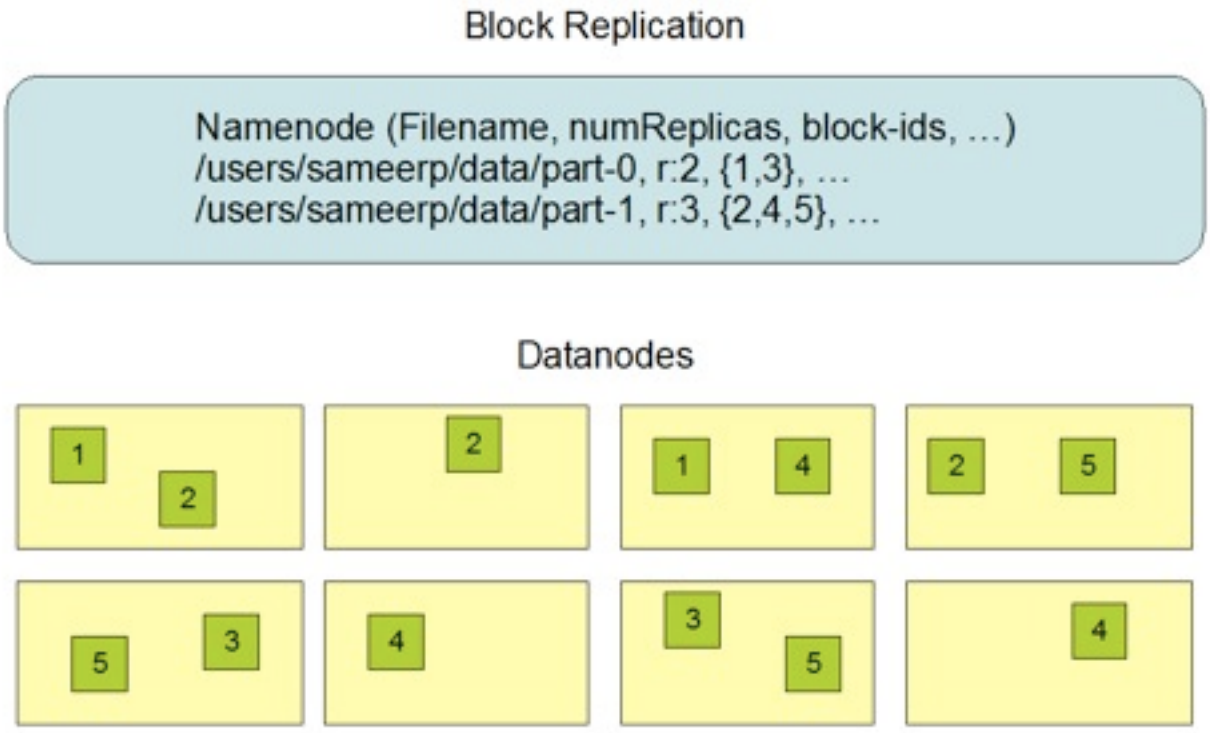
資料結點 (DataNode)

- 實際儲存檔案區塊(Blocks)的伺服器
  - 在本地端的檔案系統 (例如：ext3)中儲存真正的檔案資料
  - 記錄關於block的metadata
    - 例如：錯誤檢查碼(CRC), block與本地端檔案系統的位置的對映關係。
  - 提供資料與中介資料給用戶端
- Block狀態回報
  - 定期傳送現有Blocks的狀態給NameNode。
  - 若NameNode發現某個檔案的某個block的複本數量少於現有的備份設定時，NameNode會主動增加該block的複本。



HDFS – 備份機制 (Replication)

- 預設值為 3 份複製
- 可針對每個檔案設定區塊大小(*block size*)與複製因素(*replication factor*)
- 區塊放置演算法可參考機架的資訊(*rack- aware*)來進行放置最佳化.



Block Placement

- Policy (v0.19)
  - 在本端機架的本端節點上放置一份複本
  - 在本端機架的不同節點上放置第二份複本
  - 在遠端機架上放置第三份複本
  - 其他複本則隨機放置
- 用戶端會讀取位置最近的複本

Heartbeats

- DataNode 傳送Heartbeats給 NameNode
  - 每 3 秒傳送一次
- NameNode 使用Heartbeats來偵測 DataNode問題。

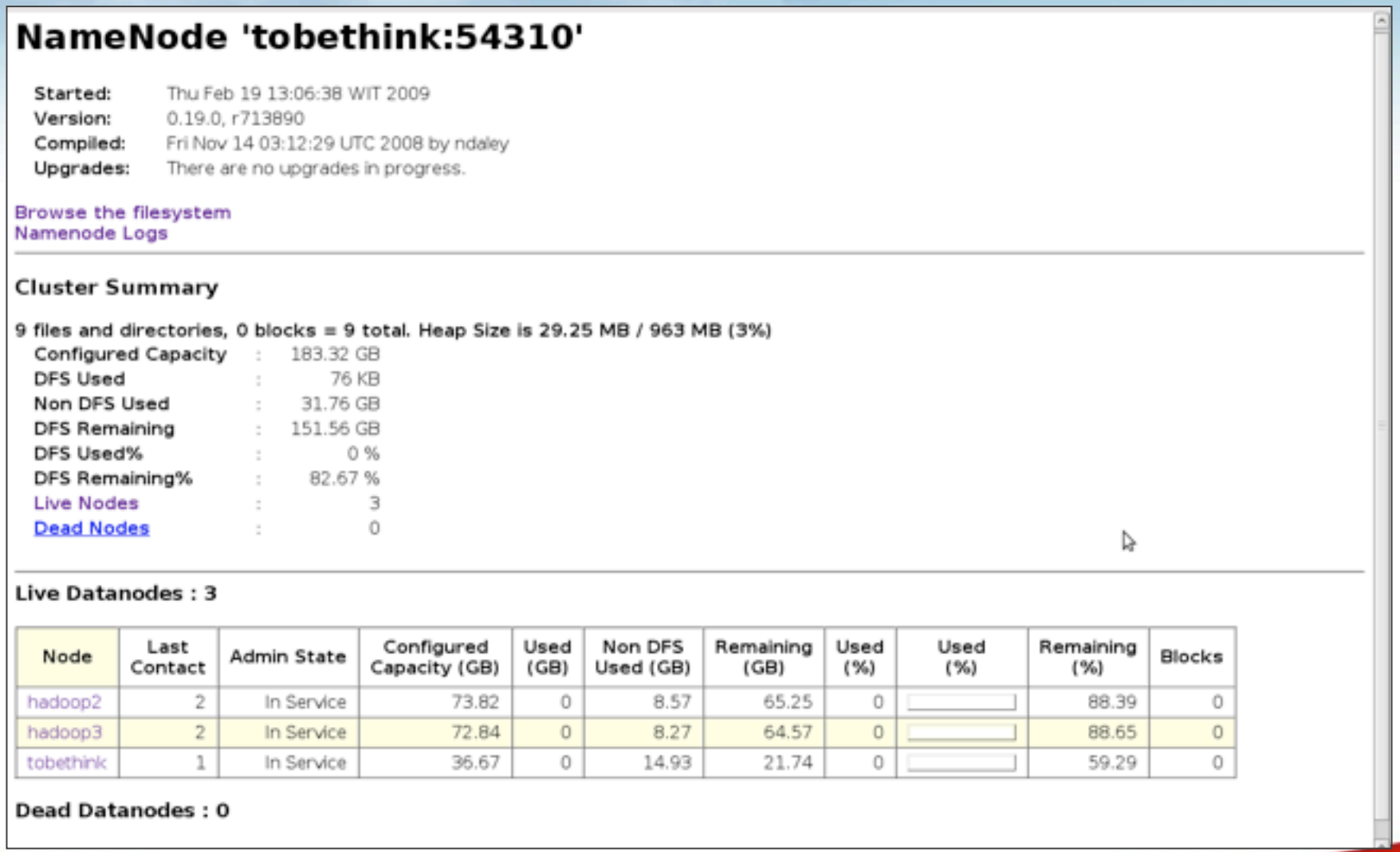
資料正確性 (Data Correctness)

- 使用Checksum來驗證資料
  - Cyclic Redundancy Check (CRC32)
- 檔案建立
  - 用戶端每隔 512 個位元組就會計算Checksum
  - DataNode 儲存Checksum的資訊
- 檔案存取時
  - 用戶端會同時擷取資料與Checksum
  - 如果驗證失敗，用戶端會嘗試使用其他複本

使用者介面 (User Interface)

- API
  - Java API
  - C language wrapper for the Java API is also avaiable
- POSIX like command
  - hadoop dfs -mkdir /foodir
  - hadoop dfs -cat /foodir/myfile.txt
  - hadoop dfs -rm /foodir myfile.txthadoop dfs -rm /foodir myfile.txt
- DFSAdmin
  - bin/hadoop dfsadmin –safemode
  - bin/hadoop dfsadmin –report
  - bin/hadoop dfsadmin -refreshNodes
- Web管理介面
  - http://host:port/dfshealth.jsp

Web管理介面



NameNode 'tobethink:54310'

Started: Thu Feb 19 13:06:38 WIT 2009
Version: 0.19.0, r713890
Compiled: Fri Nov 14 03:12:29 UTC 2008 by ndaley
Upgrades: There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

9 files and directories, 0 blocks = 9 total. Heap Size is 29.25 MB / 963 MB (3%)

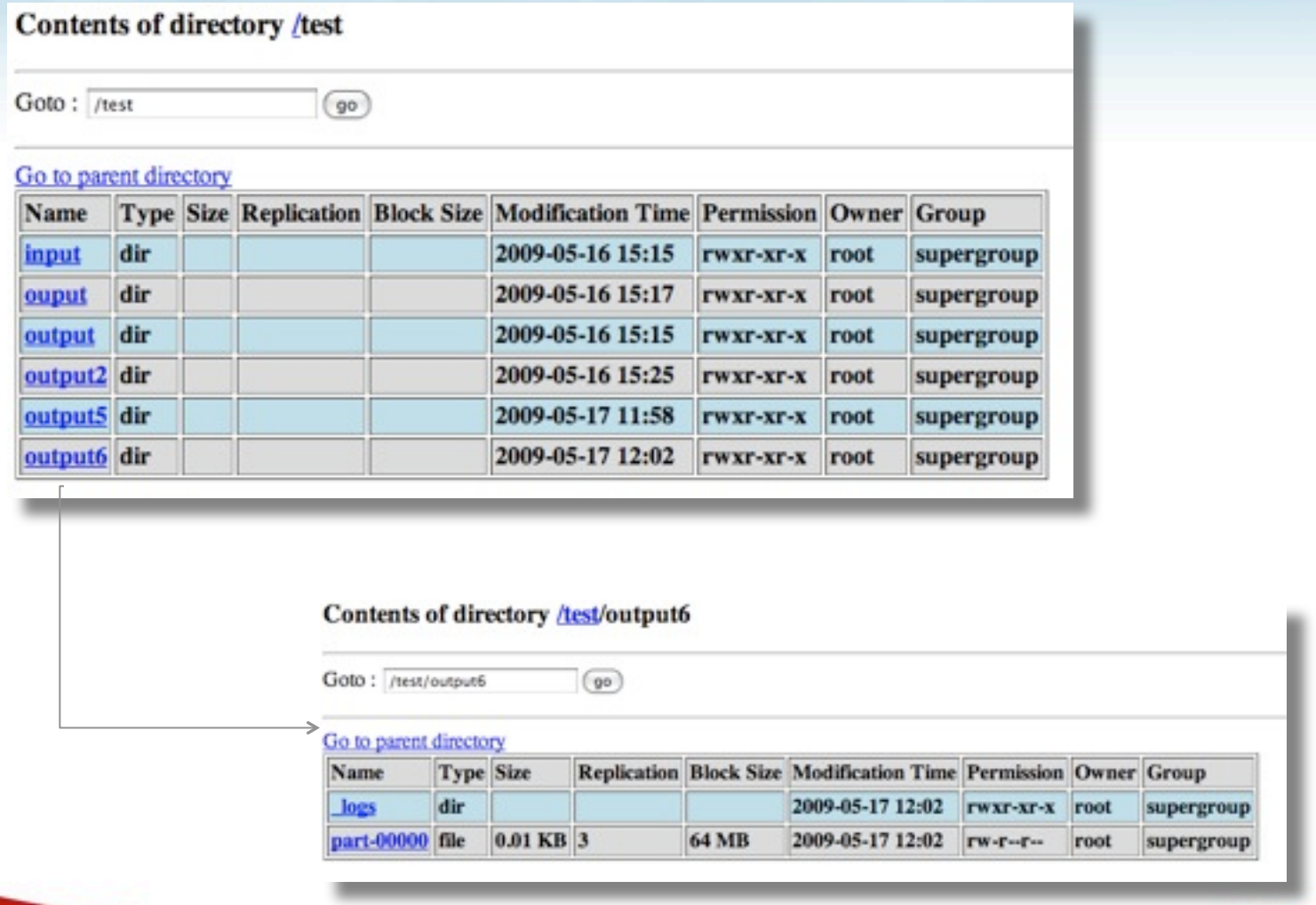
Configured Capacity	:	183.32 GB
DFS Used	:	76 KB
Non DFS Used	:	31.76 GB
DFS Remaining	:	151.56 GB
DFS Used%	:	0 %
DFS Remaining%	:	82.67 %
Live Nodes	:	3
Dead Nodes	:	0

Live Datanodes : 3

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
hadoop2	2	In Service	73.82	0	8.57	65.25	0		88.39	0
hadoop3	2	In Service	72.84	0	8.27	64.57	0		88.65	0
tobethink	1	In Service	36.67	0	14.93	21.74	0		59.29	0

Dead Datanodes : 0

Web管理介面
(http://172.16.203.136:50070)



Contents of directory /test

Goto : /test go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
input	dir				2009-05-16 15:15	rwxr-xr-x	root	supergroup
ouput	dir				2009-05-16 15:17	rwxr-xr-x	root	supergroup
output	dir				2009-05-16 15:15	rwxr-xr-x	root	supergroup
output2	dir				2009-05-16 15:25	rwxr-xr-x	root	supergroup
output5	dir				2009-05-17 11:58	rwxr-xr-x	root	supergroup
output6	dir				2009-05-17 12:02	rwxr-xr-x	root	supergroup

Contents of directory /test/output6

Goto : /test/output6 go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_logs	dir				2009-05-17 12:02	rwxr-xr-x	root	supergroup
part-00000	file	0.01 KB	3	64 MB	2009-05-17 12:02	rw-r--r--	root	supergroup

POSIX Like command

```
hadoop fs [-fs <local | file system URI>] [-conf <configuration file>]
    [-D <property=value>] [-ls <path>] [-lsr <path>] [-du <path>]
    [-dus <path>] [-mv <src> <dst>] [-cp <src> <dst>] [-rm <src>]
    [-rmr <src>] [-put <localsrc> <dst>] [-copyFromLocal <localsrc> <dst>]
    [-moveFromLocal <localsrc> <dst>] [-get <src> <localdst>]
    [-getmerge <src> <localdst> [addnl]] [-cat <src>]
    [-copyToLocal <src><localdst>] [-moveToLocal <src> <localdst>]
    [-mkdir <path>] [-report] [-setrep [-R] [-w] <rep> <path/file>]
    [-touchz <path>] [-test -[ezd] <path>] [-stat [format] <path>]
    [-tail [-f] <path>] [-text <path>]
    [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
    [-chown [-R] [OWNER][:[GROUP]] PATH...]
    [-chgrp [-R] GROUP PATH...]
    [-help [cmd]]
```

```
[root@172 bin]# ./hadoop dfs -mkdir /test/input10
[root@172 bin]# ./hadoop dfs -put ../conf/* /test/input10/
[root@172 bin]# ./hadoop dfs -ls /test/input10/
Found 13 items
-rw-r--r--   3 root supergroup       6275 2009-05-17 12:21 /test/input10/capacity-scheduler.xml
-rw-r--r--   3 root supergroup        535 2009-05-17 12:21 /test/input10/configuration.xsl
-rw-r--r--   3 root supergroup        270 2009-05-17 12:21 /test/input10/core-site.xml
-rw-r--r--   3 root supergroup       2296 2009-05-17 12:21 /test/input10/hadoop-env.sh
-rw-r--r--   3 root supergroup       1245 2009-05-17 12:21 /test/input10/hadoop-metrics.properties
-rw-r--r--   3 root supergroup       4190 2009-05-17 12:21 /test/input10/hadoop-policy.xml
-rw-r--r--   3 root supergroup        259 2009-05-17 12:21 /test/input10/hdfs-site.xml
-rw-r--r--   3 root supergroup       2815 2009-05-17 12:21 /test/input10/log4j.properties
-rw-r--r--   3 root supergroup        272 2009-05-17 12:21 /test/input10/mapred-site.xml
-rw-r--r--   3 root supergroup         10 2009-05-17 12:21 /test/input10/masters
-rw-r--r--   3 root supergroup         30 2009-05-17 12:21 /test/input10/slaves
-rw-r--r--   3 root supergroup       1243 2009-05-17 12:21 /test/input10/ssl-client.xml.example
-rw-r--r--   3 root supergroup       1195 2009-05-17 12:21 /test/input10/ssl-server.xml.example
[root@172 bin]# ./hadoop dfs -tail /test/input10/masters
localhost
[root@172 bin]#
```

Java API

```
URI uri = new URI("hdfs://namenode/");
FileSystem fs = FileSystem.get(uri, new Configuration());
Path file = new Path("answer");

DataOutputStream out = fs.create(file);
out.writeInt(42);
out.close();

DataInputStream in = fs.open(file);
System.out.println(in.readInt());
in.close();

fs.delete(file);
```

POSIX Like command

```
hadoop fs [-fs <local | file system URI>] [-conf <configuration file>]
        [-D <property=value>] [-ls <path>] [-lsr <path>] [-du <path>]
        [-dus <path>] [-mv <src> <dst>] [-cp <src> <dst>] [-rm <src>]
        [-rmr <src>] [-put <localsrc> <dst>] [-copyFromLocal <localsrc> <dst>]
        [-moveFromLocal <localsrc> <dst>] [-get <src> <localdst>]
        [-getmerge <src> <localdst> [addnl]] [-cat <src>]
        [-copyToLocal <src><localdst>] [-moveToLocal <src> <localdst>]
        [-mkdir <path>] [-report] [-setrep [-R] [-w] <rep> <path/file>]
        [-touchz <path>] [-test -[ezd] <path>] [-stat [format] <path>]
        [-tail [-f] <path>] [-text <path>]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-chgrp [-R] GROUP PATH...]
        [-help [cmd]]
```

參考資料

- Hadoop document and installation
  - http://hadoop.apache.org/
- Hadoop Wiki
  - http://wiki.apache.org/hadoop/
- Google File System Paper
  - http://labs.google.com/papers/gfs.html