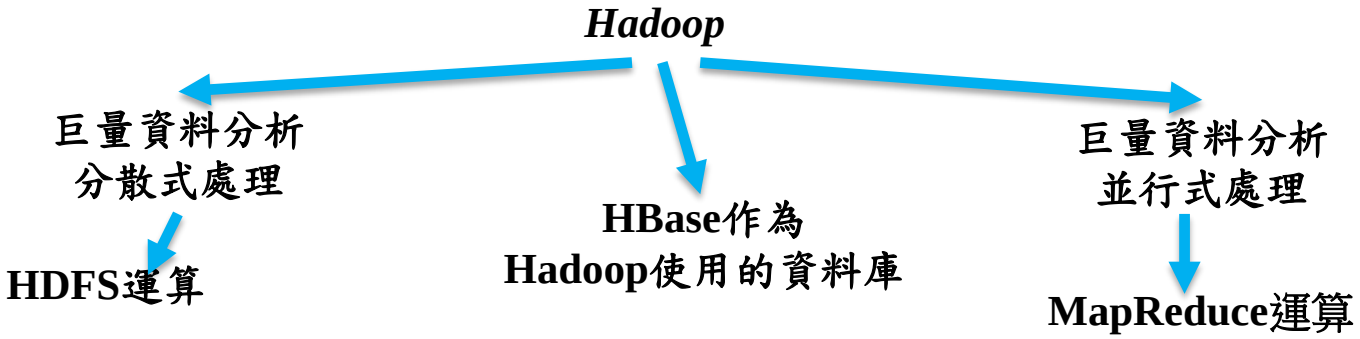
HADOOP DISTRIBUTED FILE SYSTEM HDFS

國立臺北科技大學資訊工程系
郭忠義

Hadoop

- Hadoop是Apache軟體基金會開放原始碼計劃
  - 以java寫成，提供巨量資料的分散式運算環境
- Hadoop架構由Google發表的BigTable及Google File System等概念實做，跟Google雲端運算架構相似。
  - Hadoop MapReduce如同Google MapReduce，提供分散式運算環境
  - Hadoop Distributed File System如同Google File System，提供大量儲存空間、HBase是一個類似 BigTable 的分散式資料庫 ，方便提供整合的雲端服務。



Hadoop

- Google File System
  - 可擴充的分散式檔案系統
  - 設計目的在於給大量用戶提供總體性能較高的服務
  - 適用於分散式、對大量資訊進行存取的應用
  - 可運作在一般的普通主機，提供錯誤容忍的能力
- The Google File System發表於SOSP' 03 October，並將設計概念公開

表一 Hadoop 與 Google 架構比較

Google	Hadoop
MapReduce	Hadoop MapReduce
GFS	HDFS
BigTable	HBase

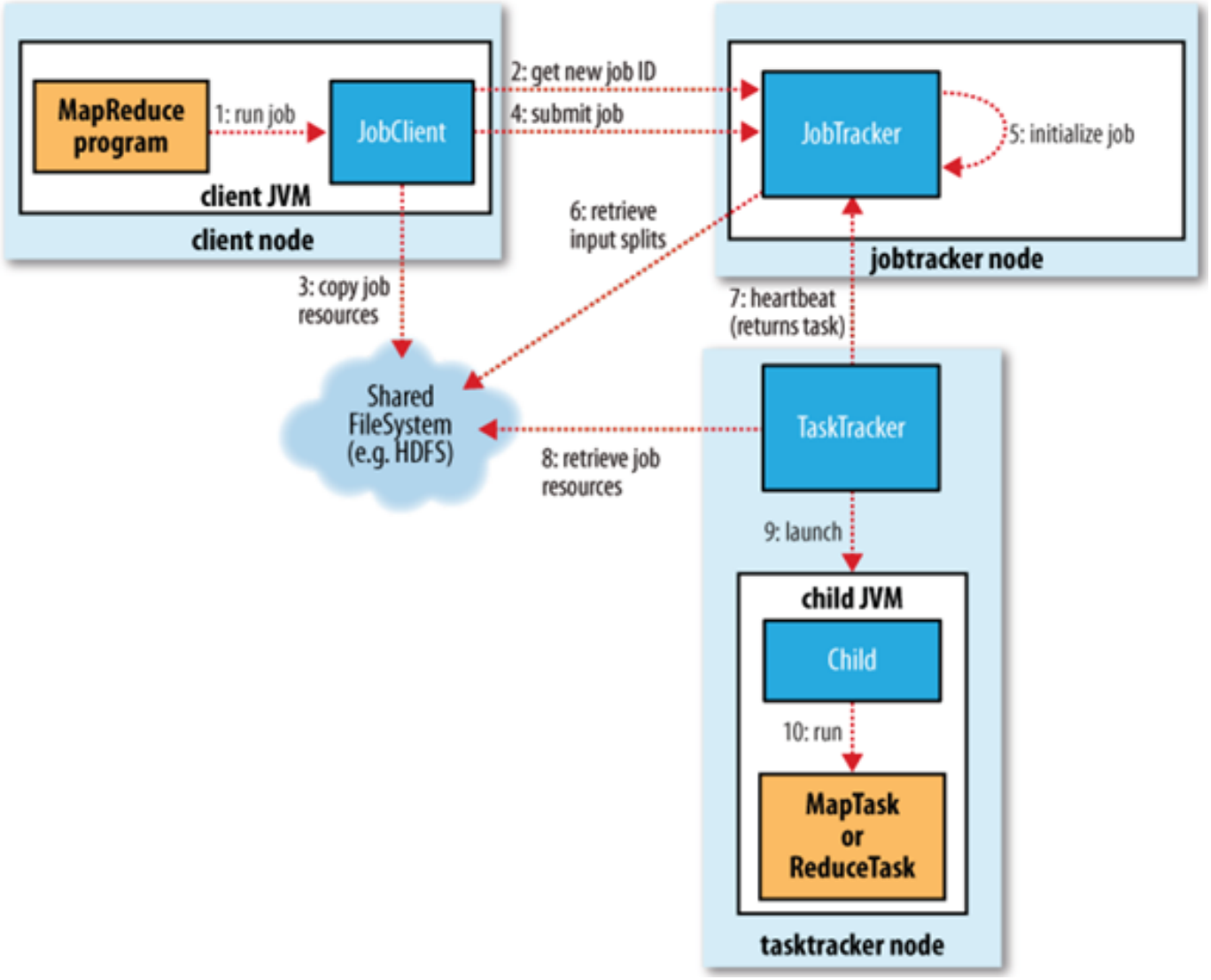
MapReduce Framework

- MapReduce是分散式程式框架，讓開發者簡單撰寫程式，利用大量運算資源，加速處理龐大的資料量。
- 一個MapReduce運算分成兩個部份—Map和Reduce
  - 大量資料在運算開始，被系統轉換成一組組 (key, value) 的序對並自動切割成許多部份
  - 分別傳給不同的Mapper處理，Mapper處理完後要將運算結果整理成一組組 (key, value)序對，再傳給Reducer整合所有Mapper結果，最後將整體結果輸出
- NameNode
  - HDFS file system 的中心區塊
- DataNode
  - HDFS儲存資料的地方

Hadoop術語

- Job 工作任務
- Task 由Job分解出的小工作
- JobTracker 工作任務分派者
- TaskTracker 小工作任務執行者
- Client 發起任務的客戶端
- Map 應對 Reduce 總和
- NameNode 名稱節點
- DataNode 資料節點
- Replication 資料檔案副本
- Block 檔案區塊 64M
- Metadata 屬性資料

MapReduce Framework



Hadoop Distributed File System

- 實現類似Google File System 分散式檔案系統
- 易於擴充的分散式檔案系統，目的為對大量資料進行分析
- 運作於廉價的普通硬體上，提供容錯功能
- 給大量使用者提供總體性能高的服務
- Hadoop系統中大量資料和運算產生暫存檔案，都存放HDFS。
  - 將分散的儲存資源整合成一個具容錯能力、高效率且超大容量的儲存環境

HDFS 特色

- 硬體錯誤容忍能力 Fault Tolerance
  - 硬體錯誤是正常而非異常
  - 自動恢復或故障排除
- 串流式的資料存取 Streaming data access
  - 批次處理多於用戶交互處理
  - 高Throughput而非低Latency
- 大規模資料集 Large data sets and files
  - 支援Petabytes等級的磁碟空間
- 一致性模型 Coherency Model
  - 一次寫入，多次存取 Write-once-read-many
  - 簡化一致性處理問題 This assumption simplifies coherency

HDFS 特色

- 本地運算 Data Locality
  - 到資料的節點上計算 > 將資料從遠端複製過來計算
- 異質平台移植性 Heterogeneous
  - 即使硬體不同也可移植、擴充

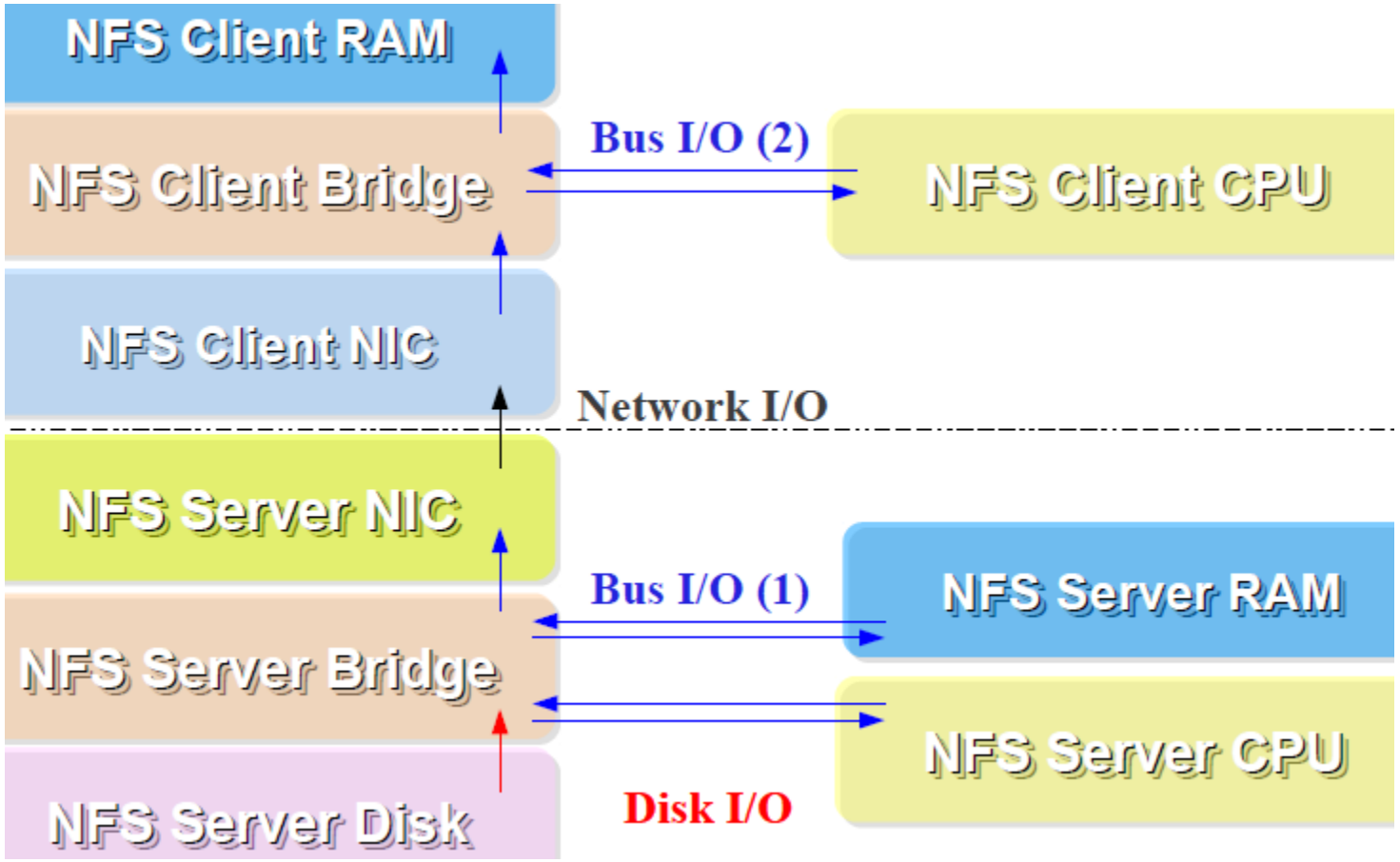
Hadoop Distributed File System

- HDFS是master/slave架構，由Name node及data nodes組成
  - Name node負責檔案系統中各檔案屬性權限等資訊的管理及儲存。
  - Name node需紀錄每一份檔案存放的位置，當有存取檔案需求，協調Data node回應；有節點損壞時，Name node會自動進行資料搬遷和複製。
  - Data node由數以百計節點擔任，一個資料檔被切割成數個較小的區塊 儲存在不同data node上，每一個區塊會有數份副本存放在不同節點，當其中一個節點損壞時，檔案系統中的資料還能保存無缺。

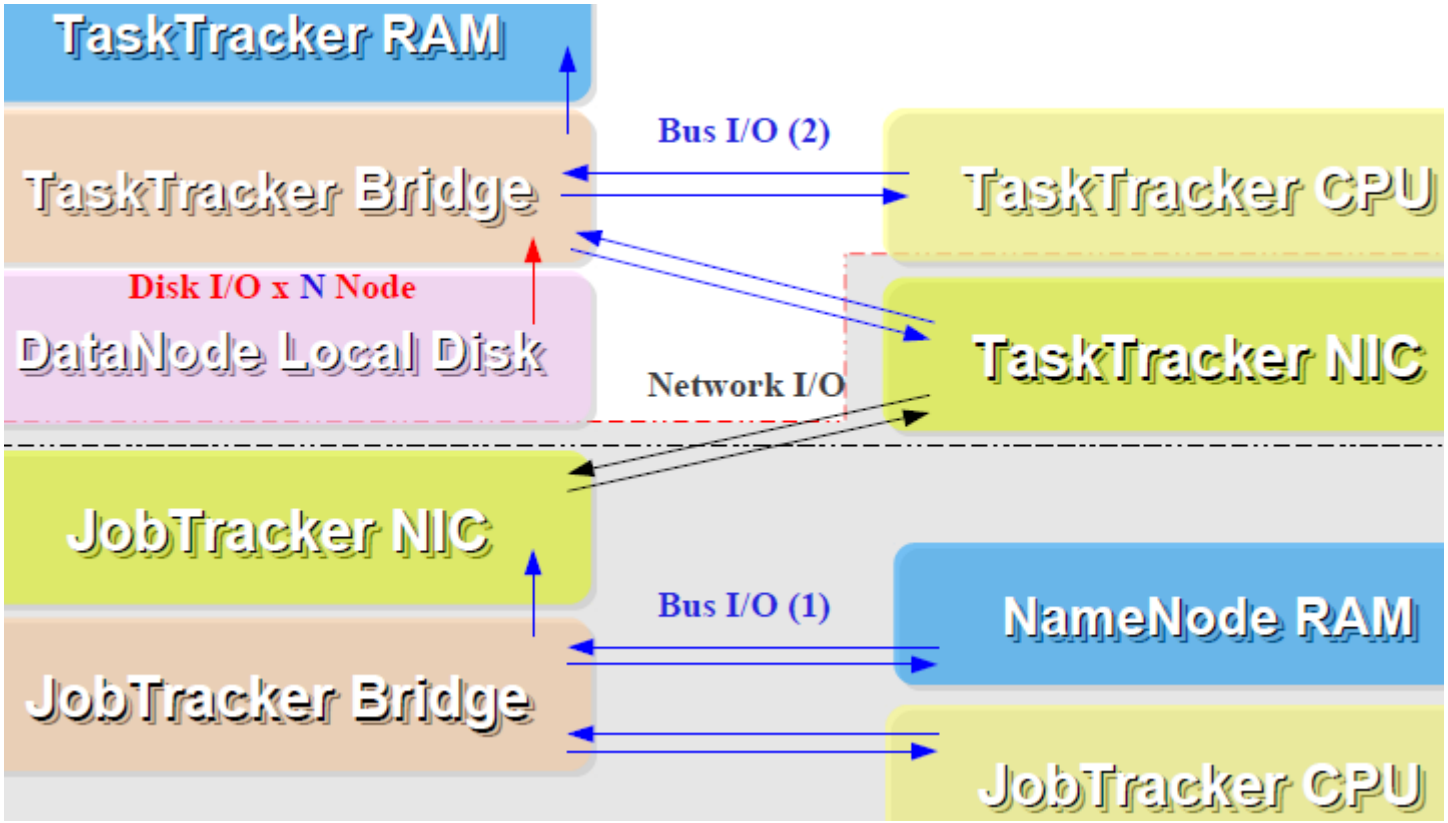
Hadoop Distributed File System

- HDFS沒有整合進Linux kernel
  - 透過Hadoop的dfs shell進行檔案操作。
  - Hadoop下的系統都與HDFS整合，做為資料儲存備份及分享的媒介。
- MapReduce在系統分配運算工作時，會將運算工作分配到存放有運算資料的節點上進行，減少大量資料透過網路傳輸的時間。

使用 NFS 平行運算

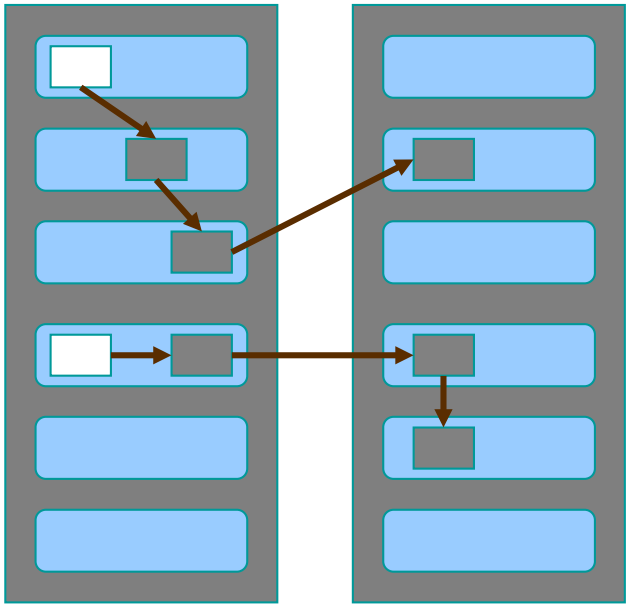

NFS Client RAM
NFS Client Bridge
Bus I/O (2)
NFS Client CPU
NFS Client NIC
Network I/O
NFS Server NIC
Bus I/O (1)
NFS Server RAM
NFS Server Bridge
NFS Server CPU
NFS Server Disk
Disk I/O

使用 NFS 平行運算


TaskTracker RAM
Bus I/O (2)
TaskTracker Bridge
TaskTracker CPU
Disk I/O x N Node
DataNode Local Disk
TaskTracker NIC
Network I/O
JobTracker NIC
Bus I/O (1)
NameNode RAM
JobTracker Bridge
JobTracker CPU

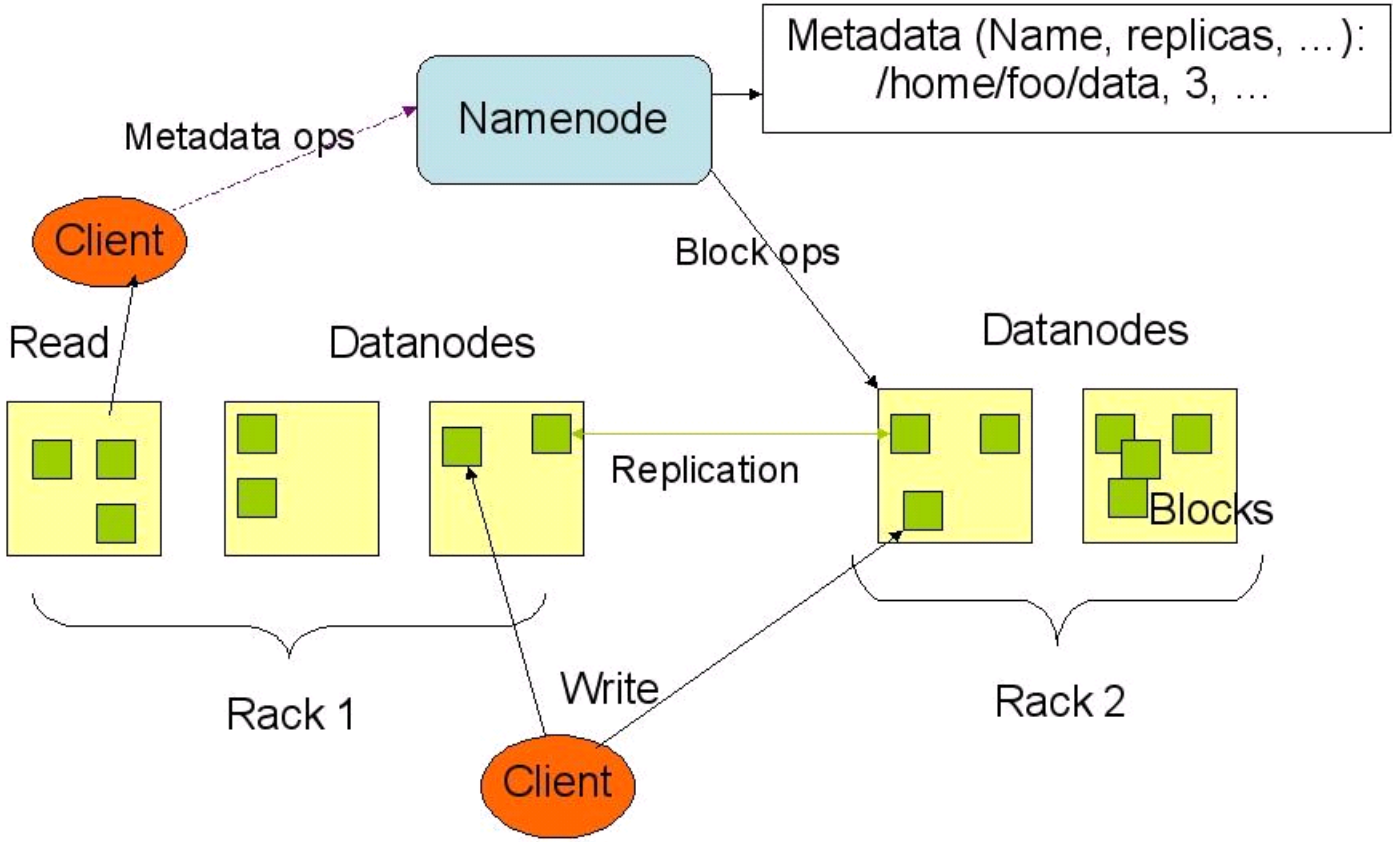
HDFS 副本備份機制

- First : 同Client的節點上
- Second : 不同機架中的節點上
- Third : 同第二個副本的機架中的另一個節點上
- More : 隨機挑選

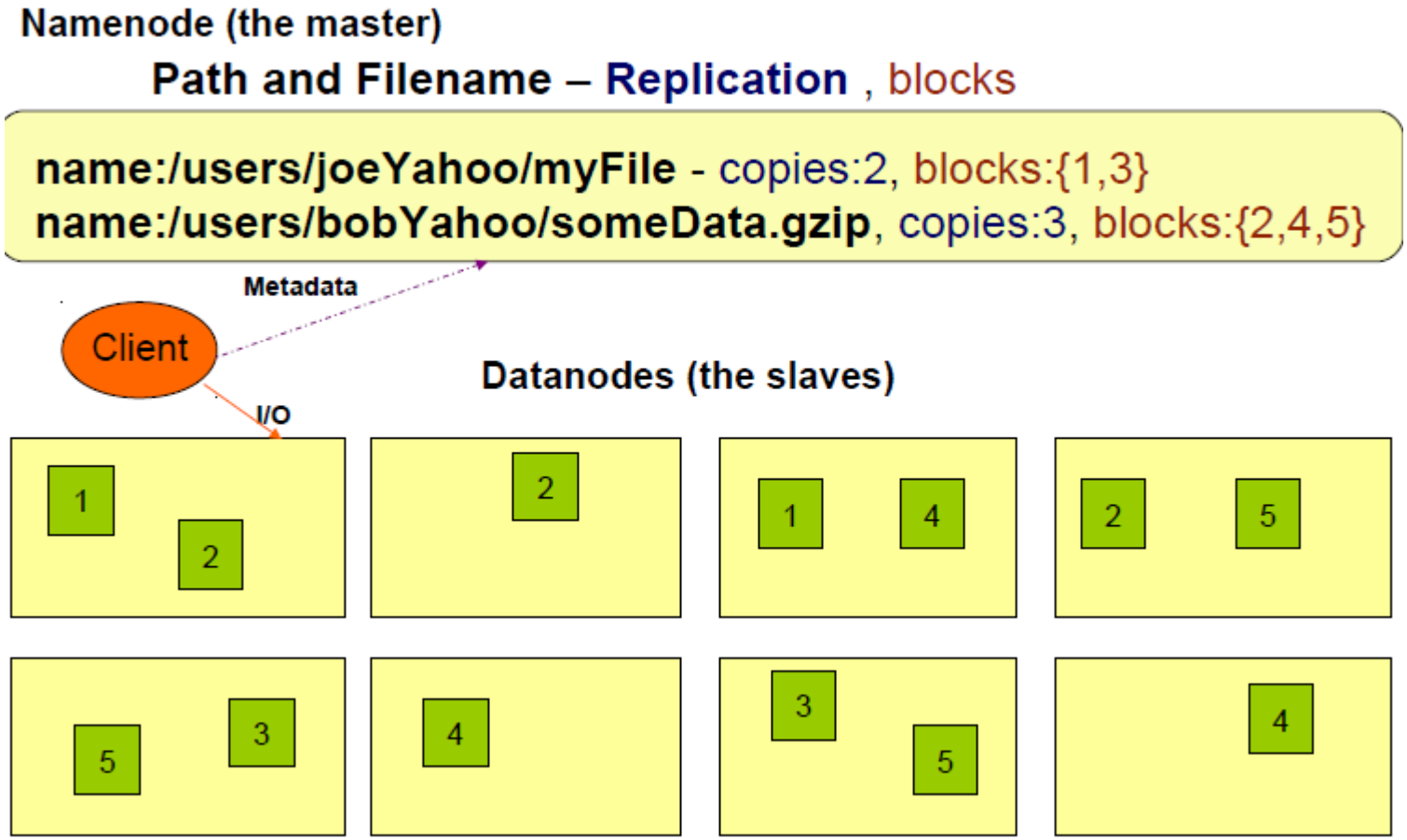


HDFS資料管理

name:/users/bobYahoo/someData.gzip, copies:3, blocks:{2,4,5}

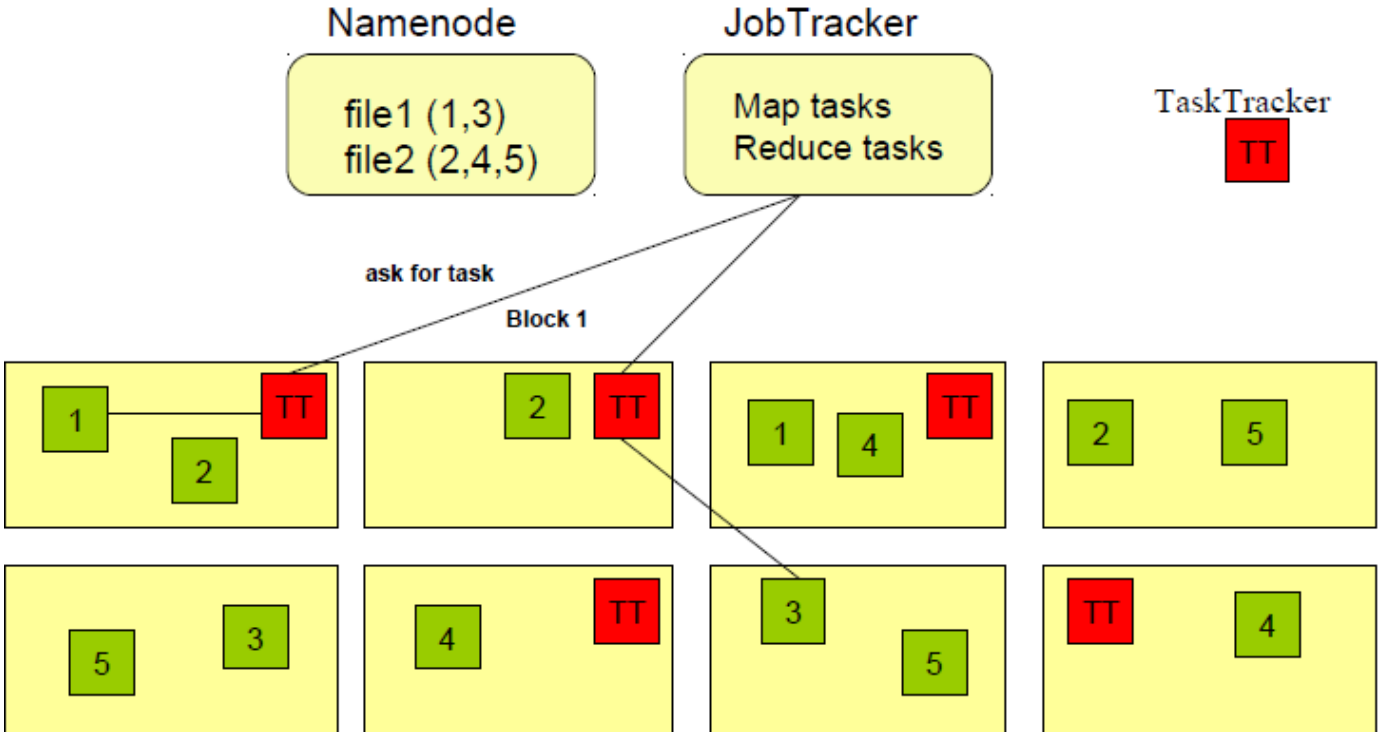


HDFS運作


Namenode (the master)
Path and Filename – Replication , blocks
name:/users/joeYahoo/myFile - copies:2, blocks:{1,3}
name:/users/bobYahoo/someData.gzip, copies:3, blocks:{2,4,5}
Metadata
Client
I/O
Datanodes (the slaves)
1
2
2
1
4
2
5
5
3
4
3
5
4

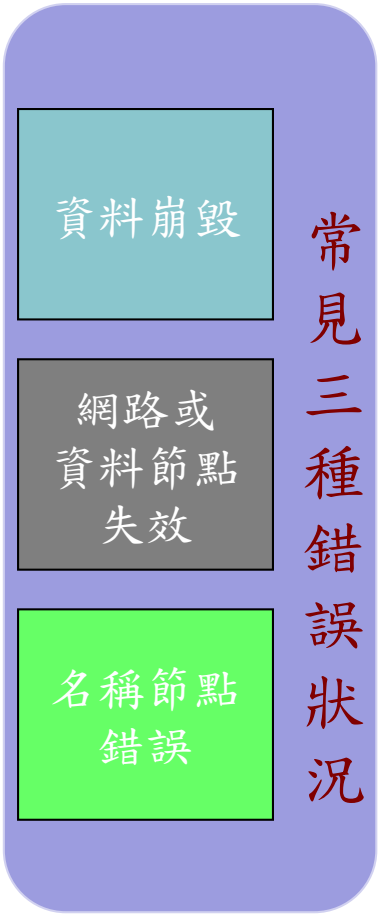
HDFS在地運算

- 強化可靠性和讀取頻寬
  - robustness：任何失效發生就讀取副本 replication
  - High read bandwidth：分散式讀取，但增加寫入瓶頸



HDFS容錯機制

- 資料完整性 Data integrity
  - checked with CRC32
  - 用副本取代出錯資料
- Heartbeat: Datanode send heartbeat to Namenode
- Metadata
  - FSImage、Editlog為核心印象檔及日誌檔
  - 多份儲存，當名稱節點故障時可以手動復原



一致性與效能機制

- 檔案一致性機制 Coherency model of files
  - 刪除檔案＼新增寫入檔案＼讀取檔案皆由Namenode負責
- 巨量空間及效能機制 Large Data Set and Performance
  - 預設每個區塊大小以64MB為單位
  - 大區塊可提高存取效率
  - 檔案有可能大過一顆磁碟
  - 區塊均勻散佈各節點以分散讀取流量

與POSIX 相似的操作指令

- hadoop fs

```
[-ls <path>]
[-lsr <path>]
[-du <path>]
[-dus <path>]
[-count[-q] <path>]
[-mv <src> <dst>]
[-cp <src> <dst>]
[-rm <path>]
[-rmr <path>]
[-expunge]
[-put <localsrc> ... <dst>]
[-copyFromLocal <localsrc> ... <dst>]
[-moveFromLocal <localsrc> ... <dst>]
[-get [-ignoreCrc] [-crc] <src> <localdst>]
[-getmerge <src> <localdst> [addnl]]
[-cat <src>]
[-text <src>]
[-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>]
[-moveToLocal [-crc] <src> <localdst>]
[-mkdir <path>]
[-setrep [-R] [-w] <rep> <path/file>]
[-touchz <path>]
[-test -[ezd] <path>]
[-stat [format] <path>]
[-tail [-f] <file>]
[-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
[-chown [-R] [OWNER][:[GROUP]] PATH...]
[-chgrp [-R] GROUP PATH...]
[-help [cmd]]
```