

# 電腦使用資訊與課程網址

- 帳號：hadoop
- 密碼：hadoop
- <http://trac.nchc.org.tw/cloud>
- 第二天下午會教叢集環境設定，  
故建議一人用兩部電腦。



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

# 雲端運算基礎課程

王耀聰 陳威宇

[jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)

[waue@nchc.org.tw](mailto:waue@nchc.org.tw)

國家高速網路與計算中心 (NCHC)



自由軟體實驗室

# 課程大綱 ( 1 )

## 第一天

09:30~10:30	介紹課程 與 雲端運算簡介
10:20~10:30	休息
10:30~12:00	Hadoop 簡介
	實作： Hadoop 單機安裝
12:00~13:00	午餐
13:00~15:00	Hadoop Distributed File System 簡介
	實作： HDFS 指令操作練習
15:00~15:10	休息
15:10~16:30	Map Reduce 介紹 (4 <sup>th</sup> )
	實作： 執行 MapReduce 基本運算 (4 <sup>th</sup> )

# 課程大綱 ( 2 )

## 第二天

09:30~10:10	MapReduce 程式設計
10:10~10:20	實作： Hadoop 程式編譯與執行
10:20~10:30	休息
10:30~12:00	New ! Eclipse 與 Hadoop 的邂逅 (3 <sup>rd</sup> )
	New ! Nutch 搜尋引擎實做 (2 <sup>nd</sup> )
12:00~13:00	午餐
13:00~14:00	Hadoop 叢集安裝設定解析 (4 <sup>th</sup> )
14:00~15:00	實作： Hadoop 叢集環境操作 (4 <sup>th</sup> )
15:00~15:10	休息
15:10~16:00	Hadoop_DRBL 快速佈屬
16:00~16:30	實作： DRBL 快速佈屬 Hadoop 叢集

# It's Show Time

- 名稱
- 服務公司 / 就讀學校
- 報名原因
- 預期收穫



# 淺談雲端運算的新趨勢

## *The Trend of Cloud Computing*

**Jazz Wang**  
**Yao-Tsung Wang**  
**jazz@nchc.org.tw**



Powered by DRBL

# ***What is Cloud Computing?***

**何謂雲端運算？請用一句話說明！**

***Anytime* 隨時**

***Anywhere* 隨地**

***With Any Devices* 使用任何裝置**

***Accessing Services* 存取各種服務**

***Cloud Computing* =~ *Network Computing***

**雲端運算 =~ 網路運算**

More definition?

其他定義請參考：

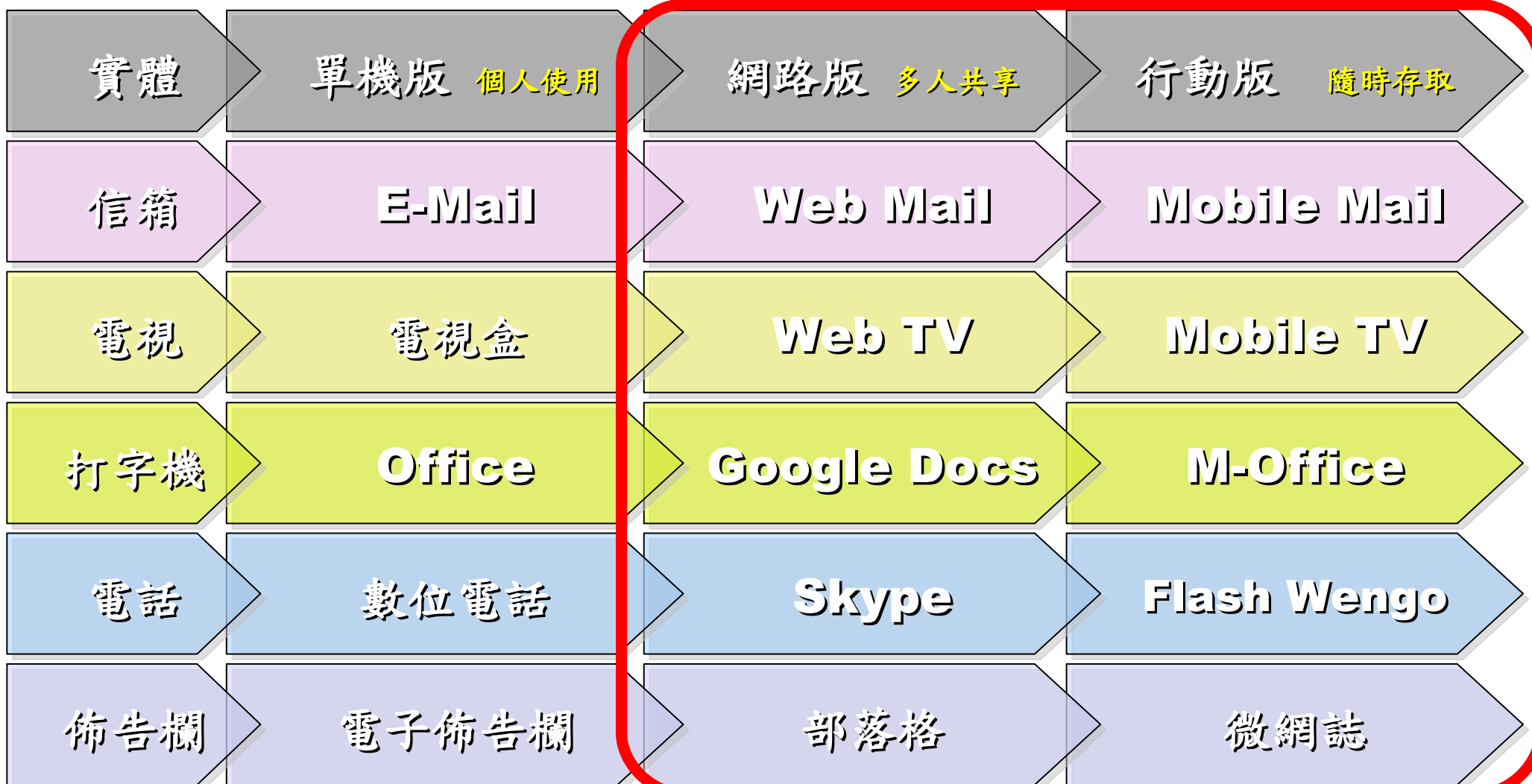
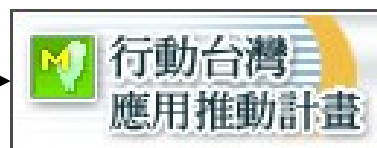
**NIST Notional**

**Definition of Cloud  
Computing**

# Evolution of Cloud Services

雲端服務只是軟體演化史的必然趨勢

數位化





***Rome wasn't built in a day !***

**羅馬不是一天造成的！**



圖片來源：<http://www.mjjq.com/pic/20070822/20070822234234402.jpg>

***When did the Cloud come ?!***

**這朵雲幾時飄過來的？！**

# ***Brief History of Computing (1/5)***



Source: <http://pinedakrch.files.wordpress.com/2007/07/>

***1960 PDP-1***

- 
- 
- 

***1965 PDP-7***

- 
- 
- 

***1969 1<sup>st</sup> Unix***

***Mainframe  
Super  
Computer***

***1977 Apple II***



***1981 IBM 1<sup>st</sup> PC 5150***



***Back to Year 1970s ...***





# ***Brief History of Computing (2/5)***



Source: <http://www.nhc.org.tw>

**Mainframe**  
**Super**  
**Computer**

**PC | Linux**  
**Cluster**  
**Parallel**

**1990 World Wide Web  
by CERN**

...

...

**1993 Web Browser  
Mosaic by NCSA**



**1991 CORBA**

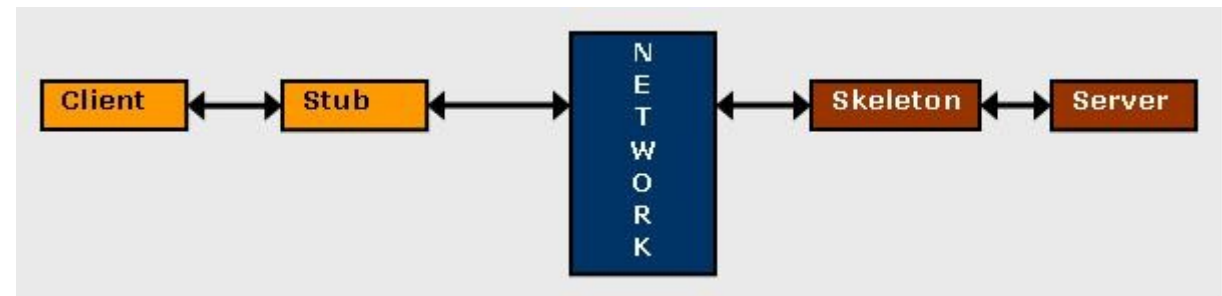
...

**Java RMI**

**Microsoft DCOM**

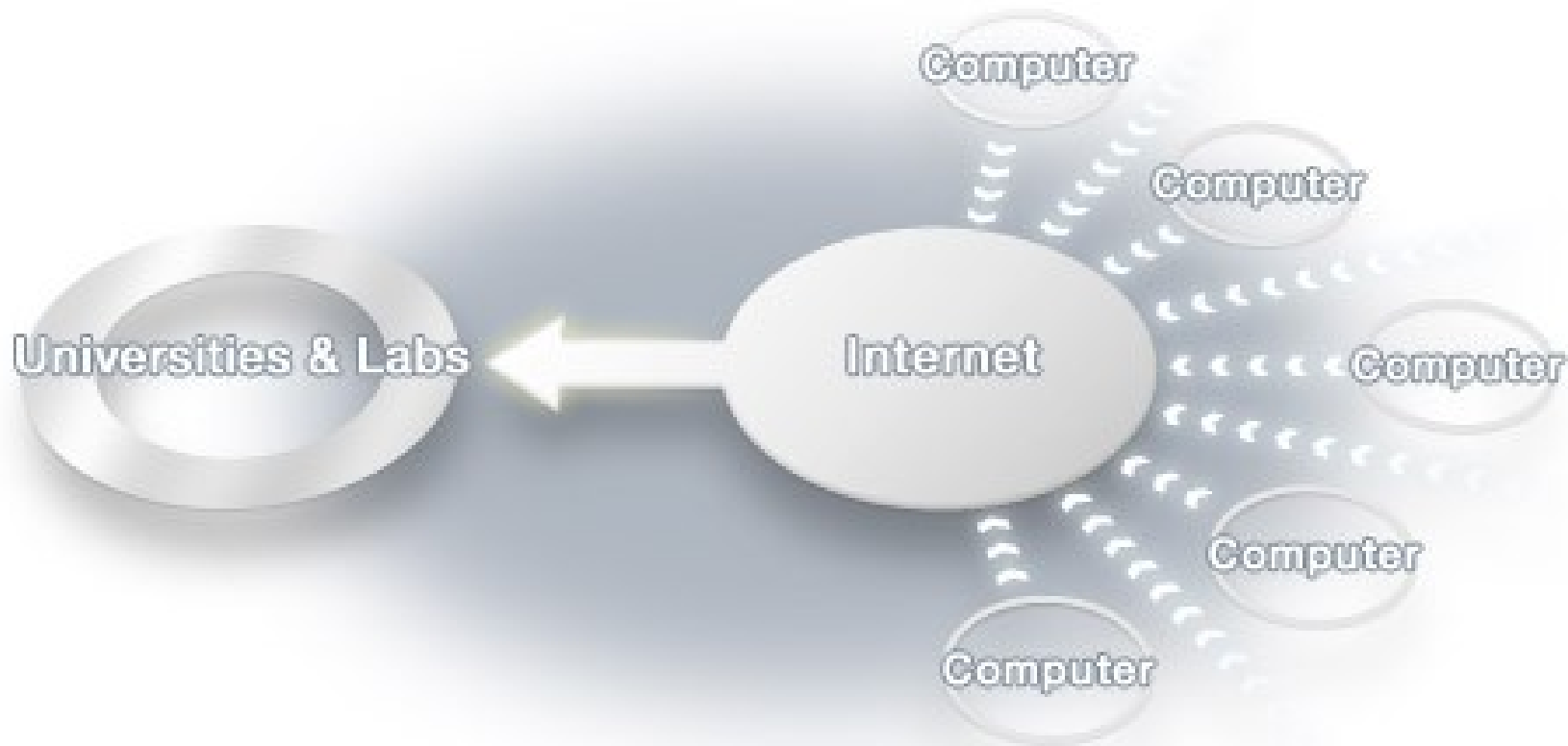
...

**Distributed Objects**



**Back to Year 1990s ...**

# ***Brief History of Computing (3/5)***



Source: <http://www.scei.co.jp/folding/en/dc.html>

**Mainframe**  
*Super  
Computer*

**PC | Linux**  
*Cluster  
Parallel*

**Internet**  
*Distributed  
Computing*

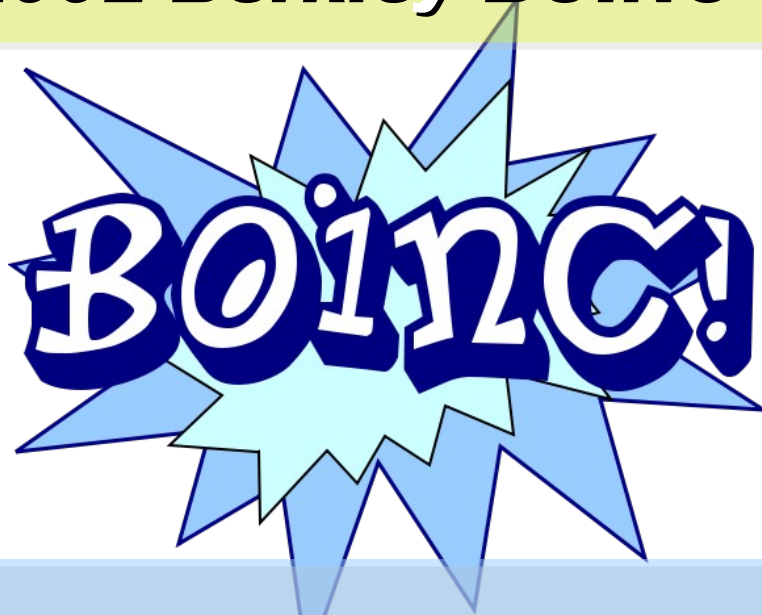
**1997 Volunteer Computing**  
**1999 SETI@HOME**



**2003 Globus Toolkit 2**



**2002 Berkley BOINC**



**2004 EGEE gLite**



**Back to Year 2000s ...**



# ***Brief History of Computing (4/5)***



Source: <http://gridcafe.web.cern.ch/gridcafe/whatisgrid/whatis.html>

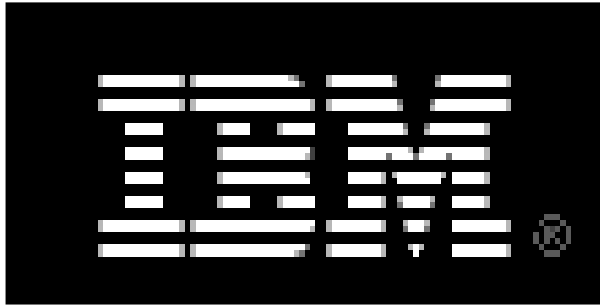
**Mainframe**  
*Super  
Computer*

**PC | Linux**  
*Cluster  
Parallel*

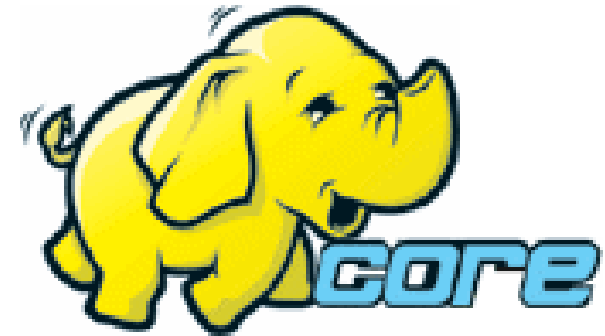
**Internet**  
*Distributed  
Computing*

**Virtual Org.**  
*Grid  
Computing*

**2001 Autonomic Computing**  
**IBM**



**2006 Apache Hadoop**



**2005 Utility Computing**  
**Amazon EC2 | S3**

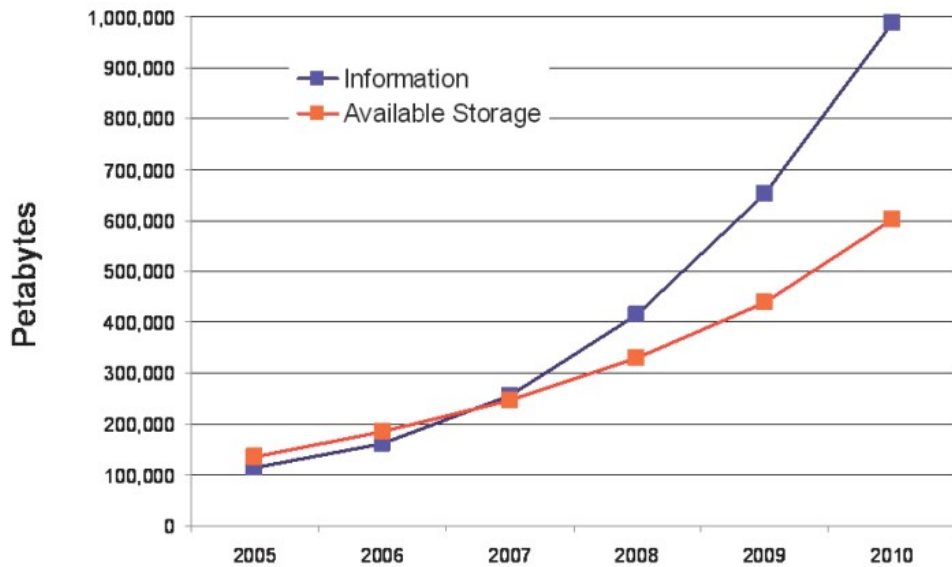


**2007 Cloud Computing**  
**Google + IBM**



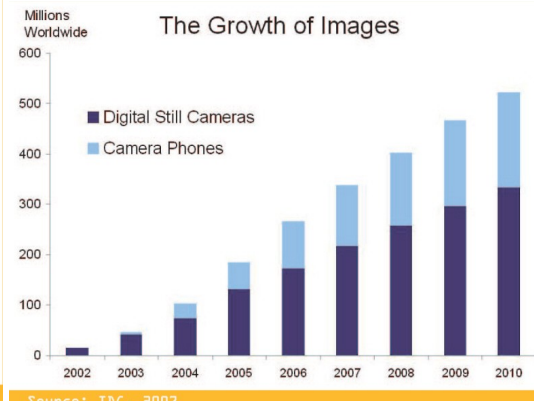
**Back to Year 2007 ...**

# Information Versus Available Storage



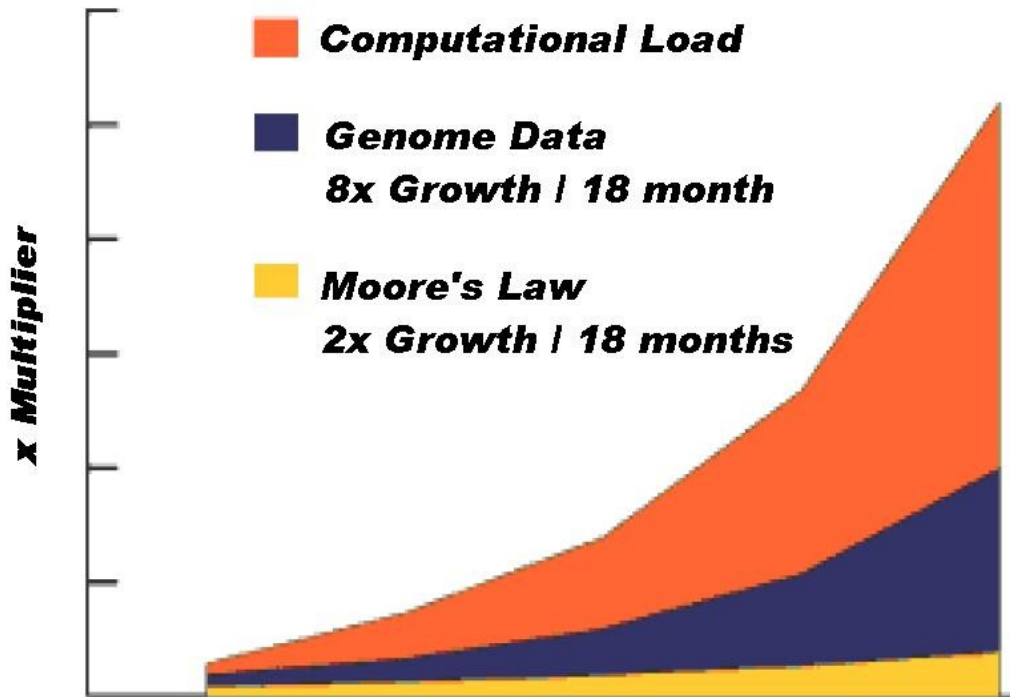
# 2007 Data Explore

**Top 1 : Human Genomics – 7000 PB / Year**  
**Top 2 : Digital Photos – 1000 PB+/ Year**  
**Top 3 : E-mail (no Spam) – 300 PB+ / Year**



Source: <http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf>  
 Source: IDC, 2007

Source: IDC, 2007

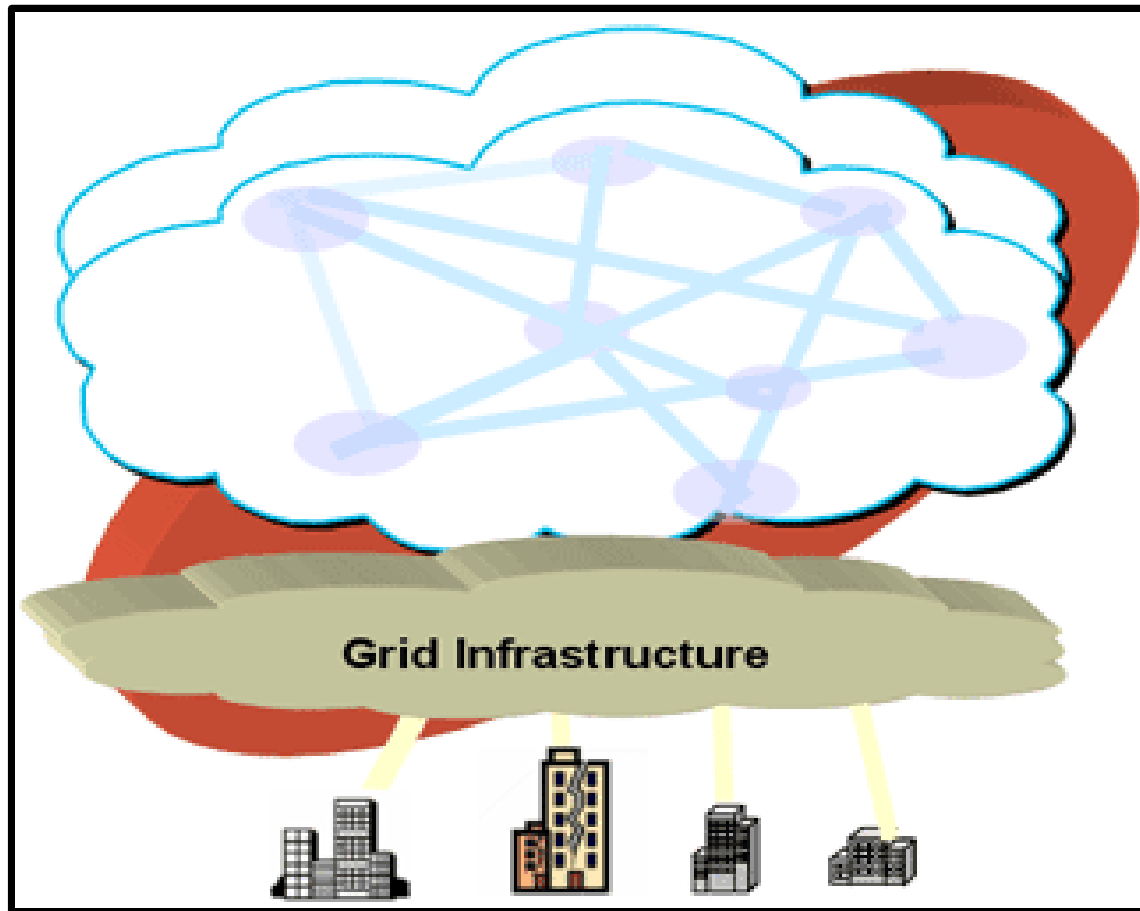


Particle Physics Large Hadron Collider (15PB)	Human Genomics (7000PB) 1GB / person 200PB+ captured 200% CAGR	World Wide Web (~1PB)	Wikipedia (10GB) 100% CAGR
Annual Email Traffic, no spam (300PB+)	Internet Archive (1PB+)	Estimated On-line RAM in Google (8PB)	Personal Digital Photos (1000PB+) 100% CAGR
200 of London's Traffic Cams (8TB/day)	2004 Walmart Transaction DB (500TB)	Typical Oil Company (350TB+)	Merck Bio Research DB (1.5TB/qtr)
UPMC Hospitals Imaging Data (500TB/yr)	MIT Babytalk Speech Experiment (1.4PB)	Terashake Earthquake Model of LA Basin (1PB)	One Day of Instant Messaging in 2002 (750GB)
Total digital data to be created this year <b>270,000PB</b> (IDC)			

Source: [http://lib.stanford.edu/files/see\\_pasig\\_dic.pdf](http://lib.stanford.edu/files/see_pasig_dic.pdf)

Phillip B. Gibbons, Data-Intensive Computing Symposium

# Brief History of Computing (5/5)



Source: <http://mmdays.com/2008/02/14/cloud-computing/>

**mainframe**  
super  
computer

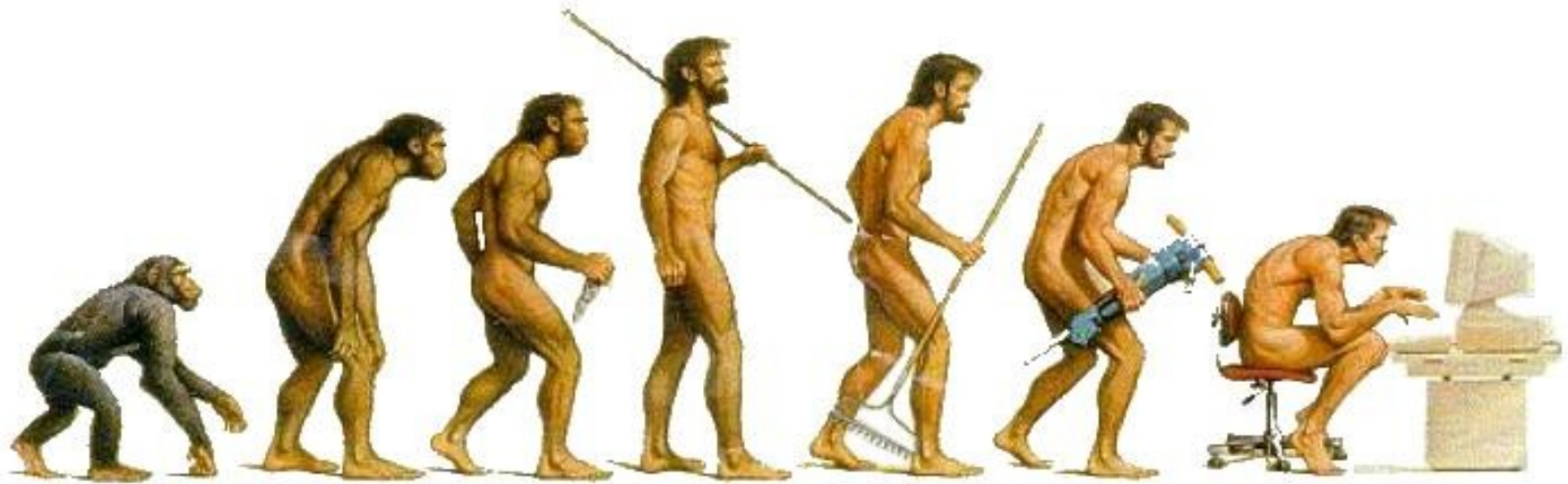
**PC | Linux**  
Cluster  
Parallel

**Internet**  
Distributed  
Computing

**Virtual Org.**  
Grid  
Computing

**Data Explode**  
Cloud  
Computing

# Evolution



(OR is it?)

***What can we learn from the past ?!***

**在這漫長的演化中，我們到底學到些什麼？！**



**Lesson #1: One cluster can't fit all !**

**教訓一：叢集的單一設定無法滿足所有需求！**

**Answer #1: Virtual Cluster 新服務：虛擬化叢集**

**Lesson #2: Grid for Heterogeneous Enterprise !**

**教訓二：格網運算該用在異業結盟的資源共享！**

**Answer #2: Peak Usage Time 尖峰用量發生時間點**

**Lesson #3: Extra cost to move data to Grid !**

**教訓三：資料搬運的網路與時間成本！**

**Answer #3: Total Cost of Ownership 總擁有成本**

**This is why Cloud Computing matters ?!**

**這就是為什麼雲端運算變得熱門?!**

# ***Trend #1: Data are moving to the Cloud***

**趨勢一：資料開始回歸集中管理**

***Access data anywhere anytime*** 為了隨時存取

***Reduce the risk of data lost*** 降低資料遺失風險

***Reduce data transfer cost*** 減少資料傳輸成本

***Enhance team collaboration*** 促進團隊協同合作

***How to store huge data ?!***

**如何儲存大量資料呢?!**

***Trend #2: Web become default Platform!***

**趨勢二：網頁變成預設開發平台**

***Open Standard*** 網頁是開放標準

***Open Implementation*** 實作不受壟斷

***Cross Platform*** 瀏覽器成為跨平台載具

***Web Application*** 網頁程式設計成為顯學

***Browser difference become entry barrier ?!***

**瀏覽器的差異造成新的技術門檻?!**



**Trend #3: HPC become a new industry**

**趨勢三：高速計算已悄悄變成新興產業**

**Parallel Computing** 平行運算的技能

**Distributed Computing** 分散運算的技能

**Multi-Core Programming** 多核心程式設計

**Processing Big Data** 處理大資料的技能

**Education and Training are needed !!**

**為了讓這些技能與產業接軌，亟需教育訓練！！**



***Flying to the Cloud ...***  
***or***  
***Falling to the Ground ...***

Source: [http://media.photobucket.com/image/falling%20ground/preeto\\_f10/falling.jpg](http://media.photobucket.com/image/falling%20ground/preeto_f10/falling.jpg)

該使用別人打造的雲端，還是自己打造專屬雲端呢？

# Types of Cloud Computing

## 雲端運算的三種型態

    
**Public Cloud**  
公用雲端

**Target Market**  
is **S.M.B.**  
主要客戶為  
中小企業

**Hybrid**  
**Cloud**

**Dynamic Resource Provisioning**  
**between public and private cloud**

私有雲端動態根據計算需求  
調用公用雲端的資源

以**大型企業**  
為主要客戶  
**Enterprise is**  
key market



私有雲端  
**Private Cloud**

# Types of Cloud Service Provider

## 雲端服務的市場區隔

### SaaS

Software as a Service

軟體即服務

### PaaS

Platform as a Service

平台即服務

### IaaS

Infrastructure as a Service

架構即服務



# *Everything as a Service* 啥米鬼都是一種服務

- AaaS Architecture as a Service
- BaaS Business as a Service
- CaaS Computing as a Service
- DaaS Data as a Service
- DBaaS Database as a Service
- EaaS Ethernet as a Service
- FaaS Frameworks as a Service
- GaaS Globalization or Governance as a Service
- HaaS Hardware as a Service
- IMaaS Information as a Service

## • **IaaS Infrastructure or Integration as a Service**

- IDaaS Identity as a Service
- LaaS Lending as a Service
- MaaS Mashups as a Service
- OaaS Organization or Operations as a Service

## • **SaaS Software or Storage as a Service**

## • **PaaS Platform as a Service**

- TaaS Technology or Testing as a Service
- VaaS Voice as a Service

*Customer-Oriented*

客戶導向

引用自：

[https://www.ibm.com/developerworks/mydeveloperworks/blogs/sbose/entry/gathering\\_clouds\\_of\\_xaas](https://www.ibm.com/developerworks/mydeveloperworks/blogs/sbose/entry/gathering_clouds_of_xaas)

## Public Cloud #1: **Amazon** 亞馬遜網路書店



- Amazon Web Service ( AWS )
- 虛擬伺服器： **Amazon EC2**
  - Small (Default) \$0.10 per hour \$0.125 per hour
  - All Data Transfer \$0.10 per GB
- 儲存服務： **Amazon S3**
  - \$0.150 per GB – first 50 TB / month of storage used
  - \$0.100 per GB – all data transfer in
  - \$0.01 per 1,000 PUT, COPY, POST, or LIST requests
- 觀念： **Paying for What You Use**

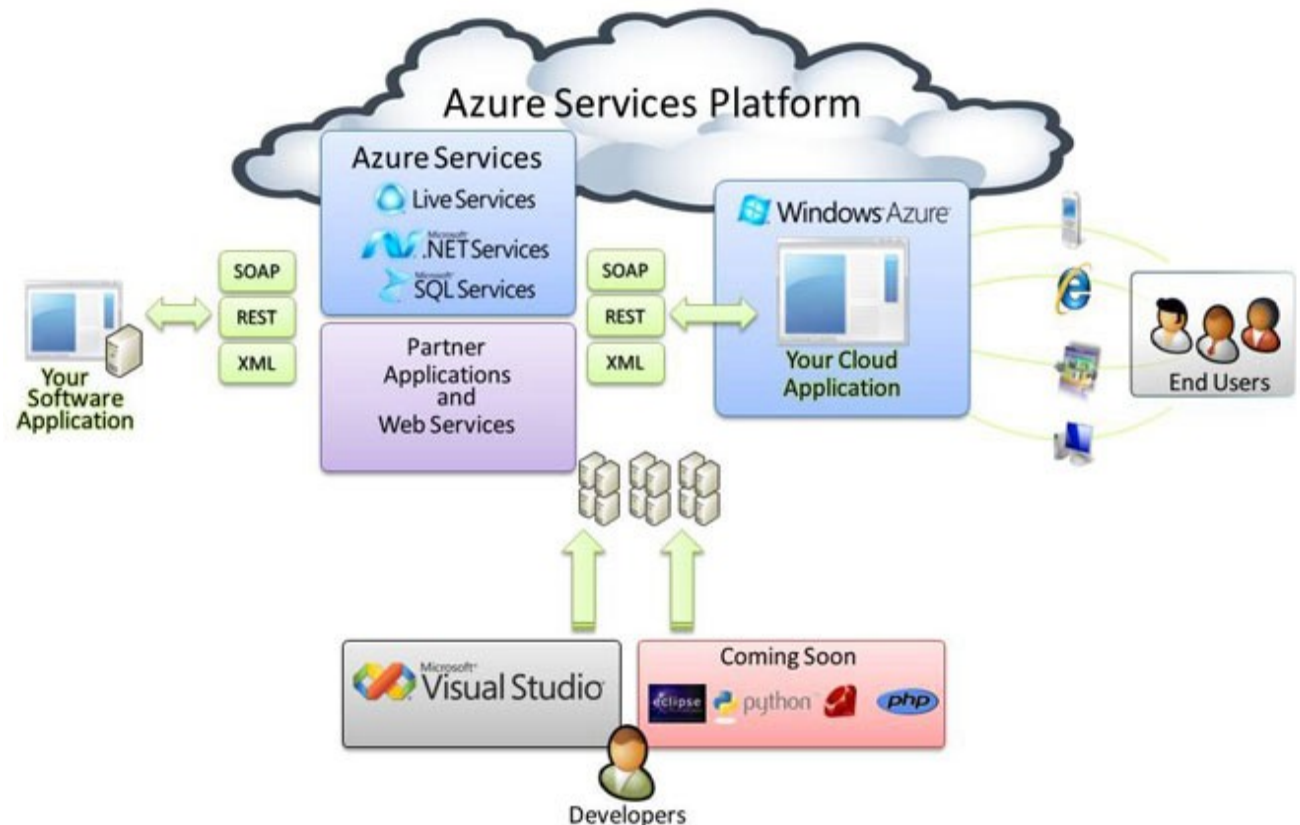
- Google App Engine (GAE)
- 讓開發者可自行建立網路應用程式於 Google 平台中。
- 提供：
  - 500MB of storage
  - up to 5 million page views a month
  - 10 applications per developer account
- 限制：
  - 程式設計語言：Python、Java





# Public Cloud #3: *Microsoft* 微軟

- Microsoft Azure 是一套雲端服務作業系統。
- 作為 Azure 服務平台的開發、服務代管及服務管理環境。
- 服務種類：
  - .Net services
  - SQL services
  - Live services





# Reference Cloud Architecture

## 雲端運算的參考架構

### 應用

Social Computing, Enterprise, ISV, ...

### 程式語言

Web 2.0 介面, Mashups, Workflows, ...

### 控制

Qos Negotiation, Admission Control, Pricing, SLA Management, Metering...

### 虛擬化

VM, VM management and Deployment

### 硬體設施

Infrastructure: Computer, Storage, Network

User-Level

User-Level  
Middleware

Core  
Middleware

System Level

SaaS  
PaaS  
IaaS

# Open Source for Private Cloud

## 建構私有雲端運算架構的自由軟體

### 應用

Social Computing, Enterprise, ISV, ...

eyeOS, Nutch, ICAS,  
X-RIME, ...

### 程式語言

Web 2.0 介面, Mashups, Workflows, ...

Hadoop (MapReduce),  
Sector/Sphere, AppScale

### 控制

Qos Negotiation, Admission Control,  
Pricing, SLA Management, Metering...

OpenNebula, Enomaly,  
Eucalyptus, OpenQRM, ...

### 虛擬化

VM, VM management and Deployment

Xen, KVM, VirtualBox,  
QEMU, OpenVZ, ...

### 硬體設施

Infrastructure: Computer, Storage,  
Network

## Open Cloud #1: *Eucalyptus*



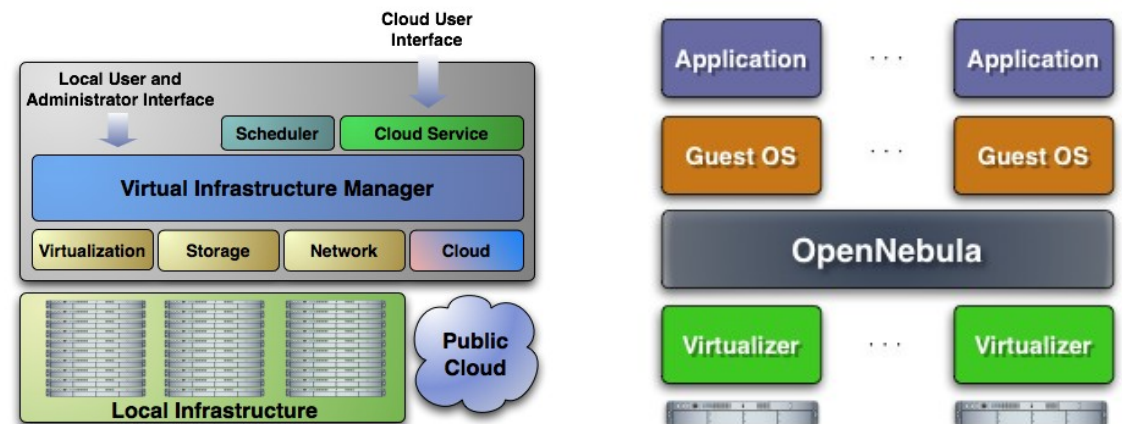
- <http://open.eucalyptus.com/>
- 原是加州大學聖塔芭芭拉分校 (UCSB) 的研究專案
- 目前已轉由 Eucalyptus System 這間公司負責維護
- 創立目的是讓使用者可以**打造自己的 EC2**
- 特色是相容於 Amazon EC2 既有的用戶端介面
- 優勢是 Ubuntu 9.04 已經收錄 Eucalyptus 的套件
- [Ubuntu Enterprise Cloud powered by Eucalyptus in 9.04](#)
- 目前有提供 Eucalyptus 的官方測試平台供註冊帳號
- 缺點：目前仍有部分操作需透過指令模式

# Open Cloud #2: *OpenNebula*

- <http://www.opennebula.org> OpenNebula.org
- 由歐洲研究學會 (European Union FP7 ) 贊助
- 將實體叢集轉換成具管理彈性的虛擬基礎設備
- 可管理**虛擬叢集**的**狀態、排程、遷徙 (migration)**
- 優勢是Ubuntu 9.04 已經收錄 OpenNebula 的套件
- 缺點：需下指令來進行虛擬機器的遷徙 (migration) 。



關於 OpenNebula 的更多資訊，請參考 <http://trac.nchc.org.tw/grid/wiki/OpenNEbula>



## Open Cloud #3: *Hadoop*

- <http://hadoop.apache.org>
- Hadoop 是 Apache Top Level 開發專案
- 目前主要由 Yahoo! 資助、開發與運用
- 創始者是 Doug Cutting，參考 Google Filesystem，以 Java 開發，提供 HDFS 與 MapReduce API。
- 2006 年使用在 Yahoo 內部服務中
- 已佈署於上千個節點。
- 處理 Petabyte 等級資料量。
- Facebook、Last.fm、Joost ... 等
- 著名網路服務均有採用 Hadoop。



- <http://sector.sourceforge.net/>
- 由美國資料探勘中心 (National Center for Data Mining) 研發的自由軟體專案。
- 採用 C/C++ 語言撰寫，因此效能較 Hadoop 更好。
- 提供「類似」Google File System 與 MapReduce 的機制
- 基於[UDT高效率網路協定](#)來加速資料傳輸效率
- [Open Cloud Consortium](#)的 [Open Cloud Testbed](#)，有提供測試環境，並開發了[MalStone效能評比軟體](#)。





**Questions?**

**Slides - <http://trac.nchc.org.tw/cloud>**

**Jazz Wang**  
**Yao-Tsung Wang**  
**jazz@nchc.org.tw**



Powered by DRBL

# What we learn today ?

## WHAT

隨時隨地用任何裝置存取各種服務！！

Accessing services with any device anytime anywhere!!

## WHO

亞馬遜、谷歌、微軟等！ 什麼都可以是服務 ~

Amazon, Google, Microsoft and more! Everything as a Service!

## WHEN

雲端運算是 2007 年繼格網運算之後的新趨勢！！

Cloud Computing become new trend since year 2007 !!

## WHY

資料集中、虛擬化、異業資源共享

Data-intensive, Virtualization, Heterogeneous

## HOW

採用自由軟體也能打造私有雲端

Hadoop, Sectore/Sphere, Eucalyptus, and more ....



# NCHC Cloud Computing Research Group

## 團隊小檔案：國網中心雲端運算研究小組

- 主要研究雲端運算的基礎架構組成元件
- <http://trac.nchc.org.tw/cloud>, <http://trac.nchc.org.tw/grid>
- 團隊成員：6名
  - 王耀聰 – drbl-xen / drbl-hadoop (~6 Years) 架構
  - 陳威宇 – Hadoop / NutchEz / ICAS (~3 Years) 應用
  - 郭文傑 – Xen / OpenNebula / Eucalyptus (~3 Years) 元件
  - 涂哲源 – Xen GPU / OpenMP / VirtualGL (~3 Years) 元件
  - 鄭宗碩 – Google App Engine (~2 Years) 新技術
  - 鄧偉華 – AMQP / OpenID (~2 Years) 新技術
- 定位：
  - 研發快速佈建軟體，提供實驗平台服務，開辦訓練課程育才
- 獨特性：
  - 基於企鵝龍 (DRBL)，可快速佈署雲端運算的叢集環境

# 更多相關的開放教材－生物叢集、GAE...

- 陽明生資所 97 年度暑期學分班 格網及平行運算 (實驗課程) <http://trac.nchc.org.tw/course/>
- 陽明生資所 98 年度暑期學分班 格網及平行運算 (實驗課程) <http://bio.classcloud.org>
- 雲端運算基礎課程 (一) Hadoop 簡介、安裝與範例實作 <http://www.classcloud.org/media/>
- 「Ruby on Rails 初學」電子書 by 鄭立竺 <http://nchcrails.blogspot.com>
- Google App Engine 電子書 by 鄭宗碩 <http://nchc-gae.blogspot.com/>
- More to come .....

陽明生資所98年度暑期學分班 格網及平行運算(實驗課程)

課程資訊

- 上課時間：2009/7/4(六),7/5(日),7/11(六) 9:10~17:30 3天，共計 18 個小時
- 上課地點：台北市北投區立農街二段155號 國立陽明大學 <=>地圖> 圖資大樓 <=>校園(P3)> R401 教室
- 講師：王耀聰、鄧偉華
- 報名網頁課程資訊
- 國網中心部份課程網站 - =><http://bio.classcloud.org> - 近期修改頁面

課程大綱

2009-07-04 (六)

- 投影片雙張一頁黑白列印版(1)

上午時段	課程內容	主講	投影片	實作步驟
09:10~09:30	課程大綱說明	王耀聰	Part-00	
09:30~10:30	第一次 Linux OS 安裝就上手 - 以 Ubuntu 9.04 安裝為例	鄧偉華	Part-01	
10:30~10:40	休息			
10:40~11:20	基本 Linux 操作 - 基礎指令	鄧偉華	Part-02	實作一
11:20~12:00	基本 Linux 操作 - 編輯器使用	鄧偉華	Part-03	實作二
下午時段	課程內容	主講	投影片	實作步驟
13:30~14:10	進階 Linux 操作(一) - SSH 遠端登入	王耀聰	Part-04	實作三
14:10~15:00	基本 Linux 程式設計 - Bash Shell Script 簡介	王耀聰	Part-05	

雲端運算基礎課程 (Hadoop簡介、安裝與範例實作)

投影片	實作步驟	課程錄影(桌面+錄音,HTML+SWF檔案)	課程錄音檔(MP3檔案)
<a href="#">介紹課程</a>		<a href="#">介紹課程</a>	<a href="#">介紹課程</a>
<a href="#">雲端運算簡介</a>		<a href="#">雲端運算的新趨勢</a>	<a href="#">雲端運算的新趨勢</a>
<a href="#">Hadoop 簡介</a>	<a href="#">實作一</a>	<a href="#">Hadoop 簡介</a>	<a href="#">Hadoop 簡介</a>
<a href="#">Hadoop 架構概述</a>		<a href="#">Hadoop 架構概述</a>	<a href="#">Hadoop 架構概述</a>
<a href="#">Hadoop Distributed File System 簡介</a>	<a href="#">實作二</a>	<a href="#">HDFS 簡介</a>	<a href="#">HDFS 簡介</a>
<a href="#">Map Reduce 介紹</a>	<a href="#">實作三</a>	<a href="#">Map Reduce 介紹</a>	<a href="#">Map Reduce 介紹</a>
<a href="#">Map Reduce 程式設計</a>	<a href="#">實作四</a>	<a href="#">Map Reduce 程式設計</a>	<a href="#">Map Reduce 程式設計</a>
<a href="#">進階 hadoop 程式開發(eclipse)</a>	<a href="#">實作五</a>	<a href="#">(1) Eclipse 安裝 (2) MapReduce Plugin 安裝設定 (3) Map Reduce 程式設計實例操作</a>	<a href="#">(1) Eclipse 安裝 (2) MapReduce Plugin 安裝設定 (3) Map Reduce 程式設計實例操作</a>
<a href="#">Hadoop 應用實例：搜尋引擎 Nutch 簡介</a>	<a href="#">實作六</a>	<a href="#">Nutch 簡介與 NutchEs 展示</a>	<a href="#">Nutch 簡介與 NutchEs 展示</a>
<a href="#">Hadoop 叢集安裝設定解析</a>		<a href="#">Hadoop 叢集設定解析</a>	<a href="#">Hadoop 叢集設定解析</a>
	<a href="#">實作七</a>	<a href="#">實作七：Hadoop 叢集安裝操作</a>	<a href="#">實作七：Hadoop 叢集安裝操作</a>
	<a href="#">實作八</a>	<a href="#">實作八：Hadoop 叢集進階操作</a>	<a href="#">實作八：Hadoop 叢集進階操作</a>
<a href="#">DRBL-Hadoop 快速佈局</a>	<a href="#">實作九</a>	<a href="#">當企鵝龍遇上小飛象</a>	<a href="#">當企鵝龍遇上小飛象</a>



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

---

# Hadoop 簡介

王耀聰 陳威宇

[jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)

[waue@nchc.org.tw](mailto:waue@nchc.org.tw)

國家高速網路與計算中心 (NCHC)



自由軟體實驗室



看了這麼多雲端服務

但……

是否有一套能夠  
開放給大家使用  
的雲端平台呢??



## The Other Open Source Projects:

<b>Eucalyptus</b>	University of California, Santa Barbara	<a href="http://open.eucalyptus.com">http://open.eucalyptus.com</a>
<b>Sector</b>	The National Center for Data Mining (NCDM)	<a href="http://sector.sf.net">http://sector.sf.net</a>
<b>Thrift</b>	Facebook	<a href="http://developers.facebook.com/thrift">http://developers.facebook.com/thrift</a>

什麼是  
Hadoop

# Hadoop ?

Hadoop is a software platform that lets one easily write and run applications that process vast amounts of data



什麼是  
Hadoop

# Hadoop

- 以 Java 開發
- 自由軟體
- 上千個節點
- Petabyte 等級的資料量
- 創始者 Doug Cutting
- 為 Apache 軟體基金會的 top level project

# 特色

- 巨量  
— 擁有儲存與處理大量資料的能力
- 經濟  
— 可以用在由一般 PC 所架設的叢集環境內
- 效率  
— 藉由平行分散檔案的處理以致得到快速的回應
- 可靠  
— 當某節點發生錯誤，系統能即時自動的取得備份資料以及佈署運算資源

## 起源：2002-2004

- Lucene
  - 用 Java 設計的高效能文件索引引擎 API
  - 索引文件中的每一字，讓搜尋的效率比傳統逐字比較還要高的多
- Nutch
  - nutch 是基於開放原始碼所開發的 web search
  - 利用 Lucene 函式庫開發

# 起源：Google 論文

- Google File System
  - SOSP 2003 : “The Google File System”
  - OSDI 2004 : “MapReduce : Simplified Data Processing on Large Cluster”
  - OSDI 2006 : “Bigtable: A Distributed Storage System for Structured Data”
- 可擴充的分散式檔案系統
- 大量的用戶提供總體性能較高的服務
- 對大量資訊進行存取的應用
- 運作在一般的普通主機上
- 提供錯誤容忍的能力

## 起源：2004~

- Dong Cutting 開始參考論文來實做
- Added DFS & MapReduce implement to Nutch
- Nutch 0.8 版之後，Hadoop 為獨立項目
- Yahoo 於 2006 年僱用 Dong Cutting 組隊專職開發
  - Team member = 14 (engineers, clusters, users, etc.)
- 2009 年 8 月 Dong Cutting 跳槽到 Cloudera

有誰  
在用

# 誰在用 Hadoop

- Yahoo 為最大的贊助商
- IBM 與 Google 在大學開授雲端課程的主要內容
- Hadoop on Amazon EC2/S3
- More...

- A9.com
- ADSDAQ by Contextweb
- EHarmony
- Facebook
- Fox Interactive Media

- IBM
- ImageShack
- ISI
- Joost
- Last.fm

- Powerset
- The New York Times
- Rackspace
- Veoh
- Metaweb



# Hadoop 於 yahoo 的運作資訊

年份	日期	節點數	耗時 (小時)
2006	四月	188	47.9
2006	五月	500	42
2006	十一月	20	1.8
2006	十一月	100	3.3
2006	十一月	500	5.2
2006	十一月	900	7.8
2007	七月	20	1.2
2007	七月	100	1.3
2007	七月	500	2

Sort benchmark, every nodes with terabytes data.

## Hadoop 於 Yahoo 的佈屬情形

資料標題：Yahoo! Launches World's Largest  
Hadoop Production Application

資料日期：February 19, 2008

Number of links between pages in the index	roughly 1 trillion links
Size of output	over 300 TB, compressed!
Number of cores used to run single Map-Reduce job	over 10,000
Raw disk used in the production cluster	over 5 Petabytes

# Hadoop 於 Yahoo 的佈屬情形

資料標題：Scaling Hadoop to 4000 nodes at Yahoo!

資料日期：September 30, 2008

Total Nodes	4000
Total cores	30000
Data	16PB

	500-node cluster		4000-node cluster	
	write	read	write	read
number of files	990	990	14,000	14,000
file size (MB)	320	320	360	360
total MB processes	316,800	316,800	5,040,000	5,040,000
tasks per node	2	2	4	4
avg. throughput (MB/s)	5.8	18	40	66

瞭解  
更多

## Hadoop 與 google 的對應

Develop Group	Google	Apache
Sponsor	Google	Yahoo, Amazon
Algorithm Method	MapReduce	MapReduce
Resource	open document	open source
File System	GFS	HDFS
Storage System (for structure data)	big-table	Hbase
Search Engine	Google	Nutch
OS	Linux	Linux / GPL

# 動手安裝囉！





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

---

# Hadoop Overview

王耀聰 陳威宇

[jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)

[waue@nchc.org.tw](mailto:waue@nchc.org.tw)

國家高速網路與計算中心 (NCHC)



自由軟體實驗室



# 作業系統的最核心！

儲存空間的資源管理



記憶體空間與  
行程分配



# 名詞

- Job
  - 任務
- Task
  - 工作
- JobTracker
  - 任務分派者
- TaskTracker
  - 工作的執行者
- Client
  - 發起任務的客戶端
- Map
  - 對應
- Reduce
  - 彙整



- Namenode
  - 名稱節點
- Datanode
  - 資料節點
- Namespace
  - 名稱空間
- Replication
  - 副本
- Blocks
  - 檔案區塊 (64M)
- Metadata
  - 屬性資料



# 管理資料

## Namenode

- Master
- 管理 HDFS 的名稱空間
- 控制對檔案的讀 / 寫
- 配置副本策略
- 對名稱空間作檢查及紀錄
- 只能有一個

## Datanode

- Workers
- 執行讀 / 寫動作
- 執行 Namenode 的副本策略
- 可多個

# 分派程序

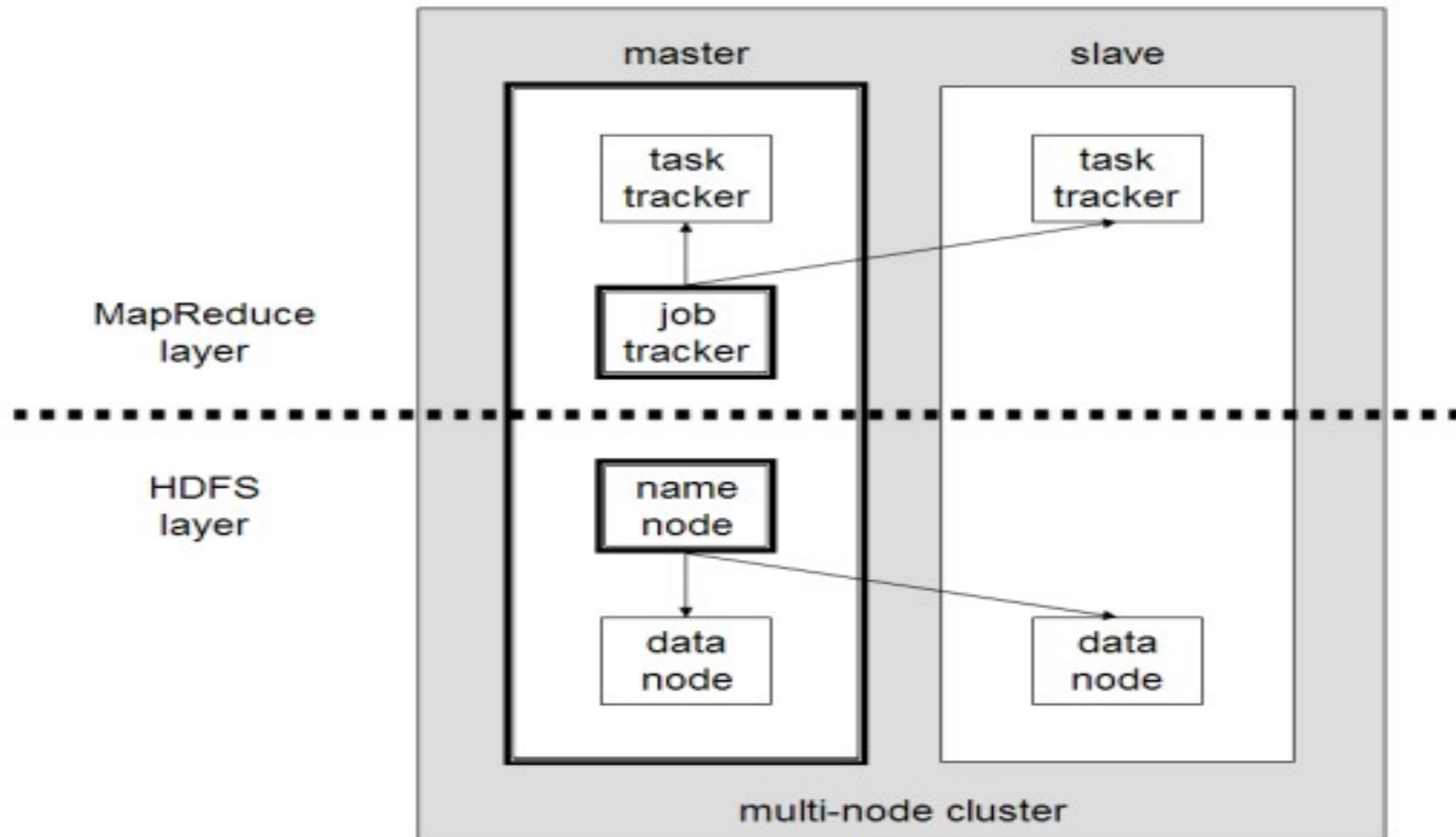
## Jobtracker

- Master
- 使用者發起工作
- 指派工作給 Tasktrackers
- 排程決策、工作分配、錯誤處理
- 只能有一個

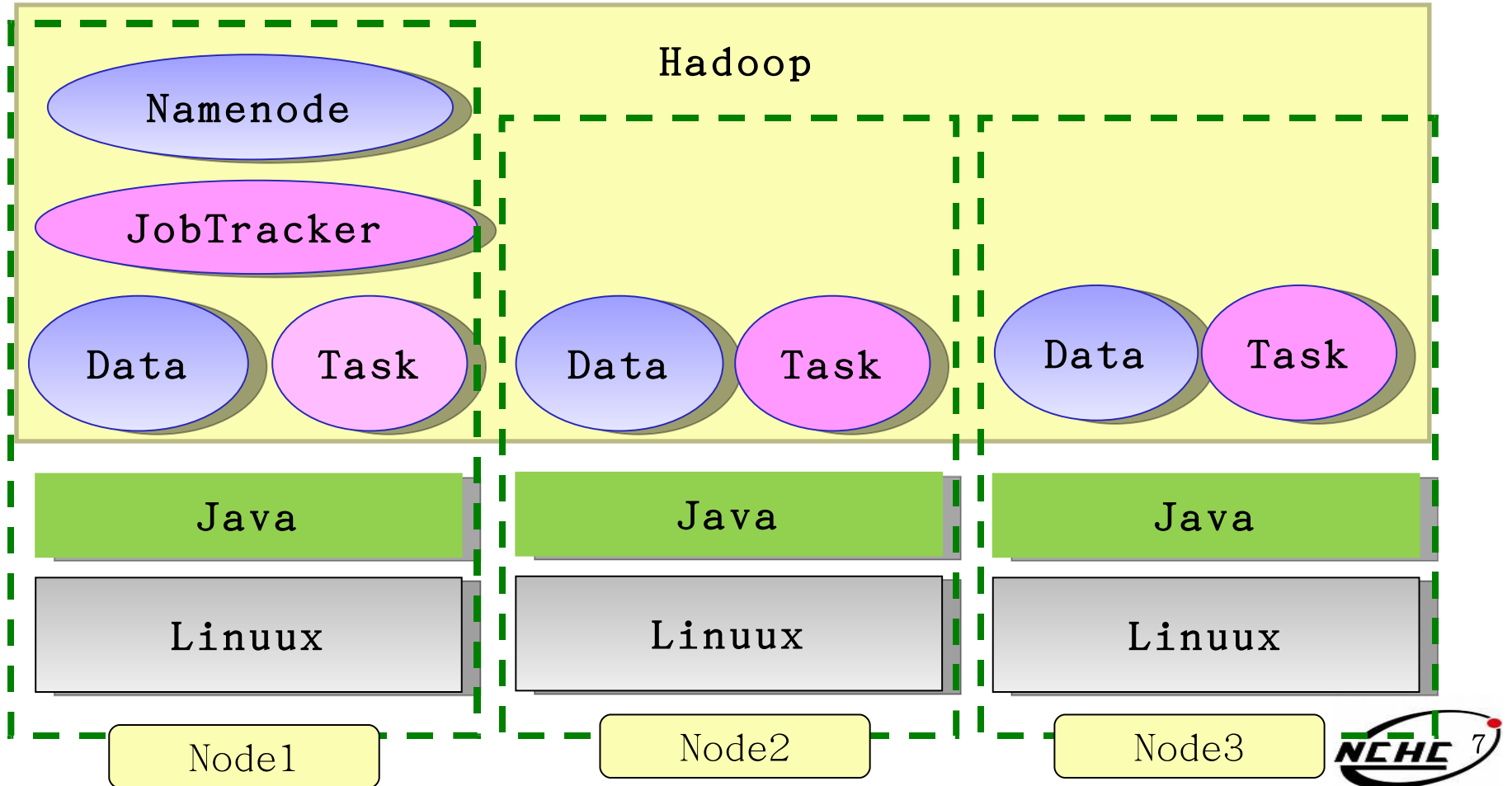
## Tasktrackers

- Workers
- 運作 Map 與 Reduce 的工作
- 管理儲存、回覆運算結果
- 可多個

# Hadoop 的各種身份

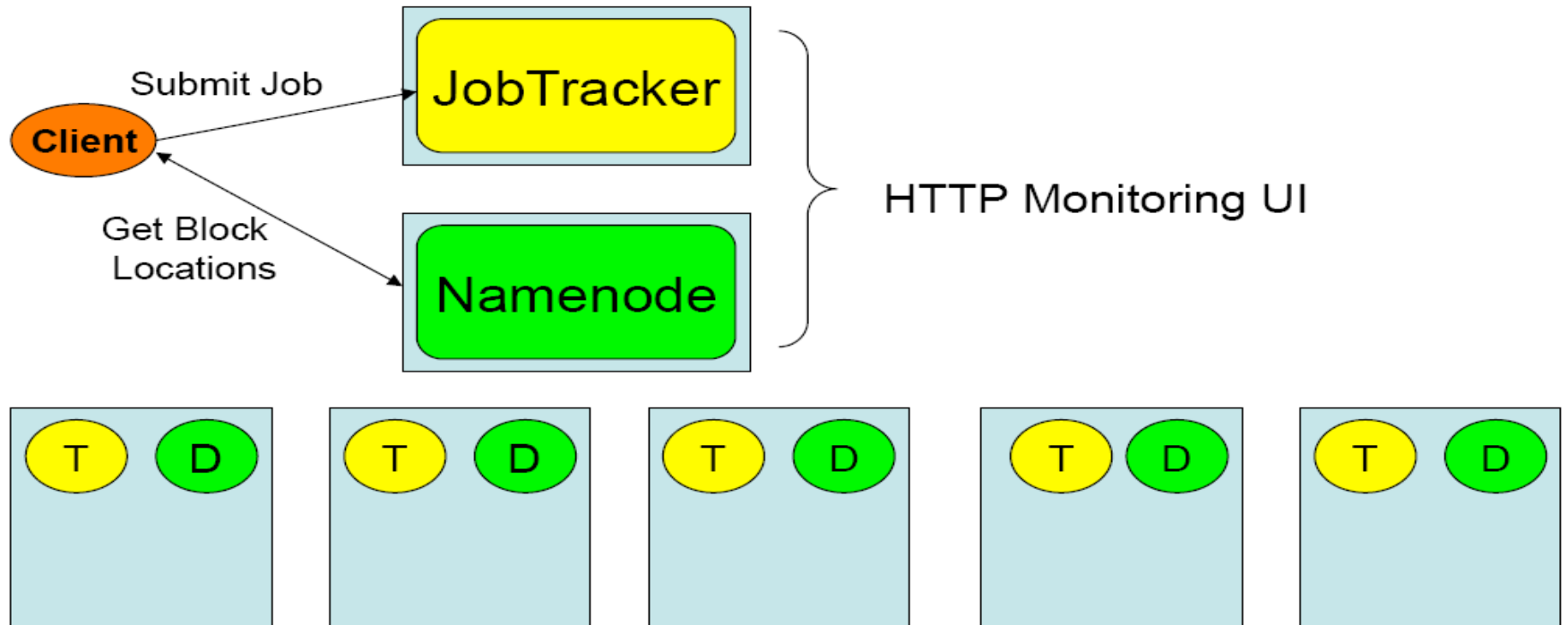


# Building Hadoop





# 不在雲裡的 Client



# Hadoop Package Topology

資料夾	說明
bin /	各執行檔：如 <code>start-all.sh</code> 、 <code>stop-all.sh</code> 、 <code>hadoop</code>
conf /	預設的設定檔目錄：設定環境變數 <code>hadoop-env.sh</code> 、各項參數 <code>hadoop-site.conf</code> 、工作節點 <code>slaves</code> 。 (可更改)
docs /	Hadoop api 與說明文件 (html & PDF)
contrib /	額外有用的功能套件，如：eclipse 的擴充外掛、Streaming 函式庫。
lib /	開發 hadoop 專案或編譯 hadoop 程式所需要的所有函式庫，如： <code>jetty</code> 、 <code>kfs</code> 。但主要的 hadoop 函式庫於 <code>hadoop_home</code>
src /	Hadoop 的原始碼。
build /	開發 Hadoop 編譯後的資料夾。需搭配 <code>ant</code> 程式與 <code>build.xml</code>
logs /	預設的日誌檔所在目錄。(可更改)

# 設定檔：hadoop-env.sh

- 設定 Linux 系統執行 Hadoop 的環境參數
  - export xxx=kkk
    - 將 kkk 這個值匯入到 xxx 參數中
  - # string...
    - 註解，通常用來描述下一行的動作內容

```
# The java implementation to use. Required.  
export JAVA_HOME=/usr/lib/jvm/java-6-sun  
export HADOOP_HOME=/opt/hadoop  
export HADOOP_LOG_DIR=$HADOOP_HOME/logs  
export HADOOP_SLAVES=$HADOOP_HOME/conf/slaves  
.....
```

# 設定檔：hadoop-site.xml

## <configuration>

```
<property>
  <name> fs.default.name</name>
  <value> hdfs://localhost:9000/</value>
  <description> ... </description>
</property>
```

```
<property>
  <name> mapred.job.tracker</name>
  <value> localhost:9001</value>
  <description>... </description>
</property>
```

```
<property>
  <name> hadoop.tmp.dir </name>
  <value> /tmp/hadoop/hadoop-$
    {user.name} </value>
  <description> </description>
</property>
```

```
<property>
  <name> mapred.map.tasks</name>
  <value> 1</value>
  <description> define mapred.map tasks
    to be number of slave hosts
  </description>
</property>
```

```
<property>
  <name> mapred.reduce.tasks</name>
  <value> 1</value>
  <description> define mapred.reduce
    tasks to be number of slave hosts
  </description>
</property>
```

```
<property>
  <name> dfs.replication</name>
  <value> 3</value>
</property>
```

</configuration>

# 設定檔：`hadoop-default.xml`

- Hadoop 預設參數

- 沒在 `hadoop.site.xml` 設定的話就會用此檔案的值

- 更多的介紹參數：

- [http://hadoop.apache.org/core/docs/current/cluster\\_setup.html#Configuring+the+Hadoop+Daemons](http://hadoop.apache.org/core/docs/current/cluster_setup.html#Configuring+the+Hadoop+Daemons)

## 設定檔： slaves

- 被此檔紀錄到的節點就會附有兩個身份：  
datanode & tasktracker
- 一行一個 hostname 或 ip

```
192.168.1.1
....
192.168.1.100
Pc101
....
Pc152
....
```

# 設定檔：masters

- 會被設定成 secondary namenode
- 可多個

192.168.1.1

....

Pc101

....



描述名稱	設定名稱	所在檔案
JAVA 安裝目錄	JAVA_HOME	hadoop-env.sh
HADOOP 家目錄	HADOOP_HOME	hadoop-env.sh
設定檔目錄	HADOOP_CONF_DIR	hadoop-env.sh
日誌檔產生目錄	HADOOP_LOG_DIR	hadoop-env.sh
HADOOP 工作目錄	hadoop.tmp.dir	hadoop-site.xml
JobTracker	mapred.job.tracker	hadoop-site.xml
Namenode	fs.default.name	hadoop-site.xml
TaskTracker	(hostname)	slaves
Datanode	(hostname)	slaves
第二 Namenode	(hostname)	masters
其他設定值	詳 hadoop-default.xml	hadoop-site.xml

# 重要執行程序

- 格式化

  - \$ bin/hadoop namenode -format

- 開始與結束

  - \$ bin/start-all.sh

  - \$ bin/start-dfs.sh

  - \$ bin/start-mapred.sh

  - \$ bin/stop-all.sh

  - \$ bin/stop-dfs.sh

  - \$ bin/stop-mapred.sh



# 重要指令程序

- 使用 hadoop 檔案系統指令

—\$ bin/hadoop fs —Instruction

- 使用 hadoop 運算功能

—\$ bin/hadoop jar XXX.jar  
Main\_Function ...



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



# Hadoop Distributed File System

# Outline

- HDFS 的定義 ?
- HDFS 的特色 ?
- HDFS 的架構 ?
- HDFS 運作方式 ?
- HDFS 如何達到其宣稱的好處 ?
- HDFS 功能 ?

# HDFS ?

- Hadoop Distributed File System
  - Hadoop：自由軟體專案，為實現 Google 的 MapReduce 架構
  - HDFS: Hadoop 專案中的檔案系統
- 實現類似 Google File System
  - GFS 是一個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，又可以提供容錯功能
  - 給大量的用戶提供總體性能較高的服務

# 設計目標 (1)

- 硬體錯誤容忍能力
  - 硬體錯誤是正常而非異常
  - 迅速地自動恢復
- 串流式的資料存取
  - 批次處理多於用戶交互處理
  - 高 **Throughput** > 低 Latency
- 大規模資料集
  - 支援 Perabytes 等級的磁碟空間

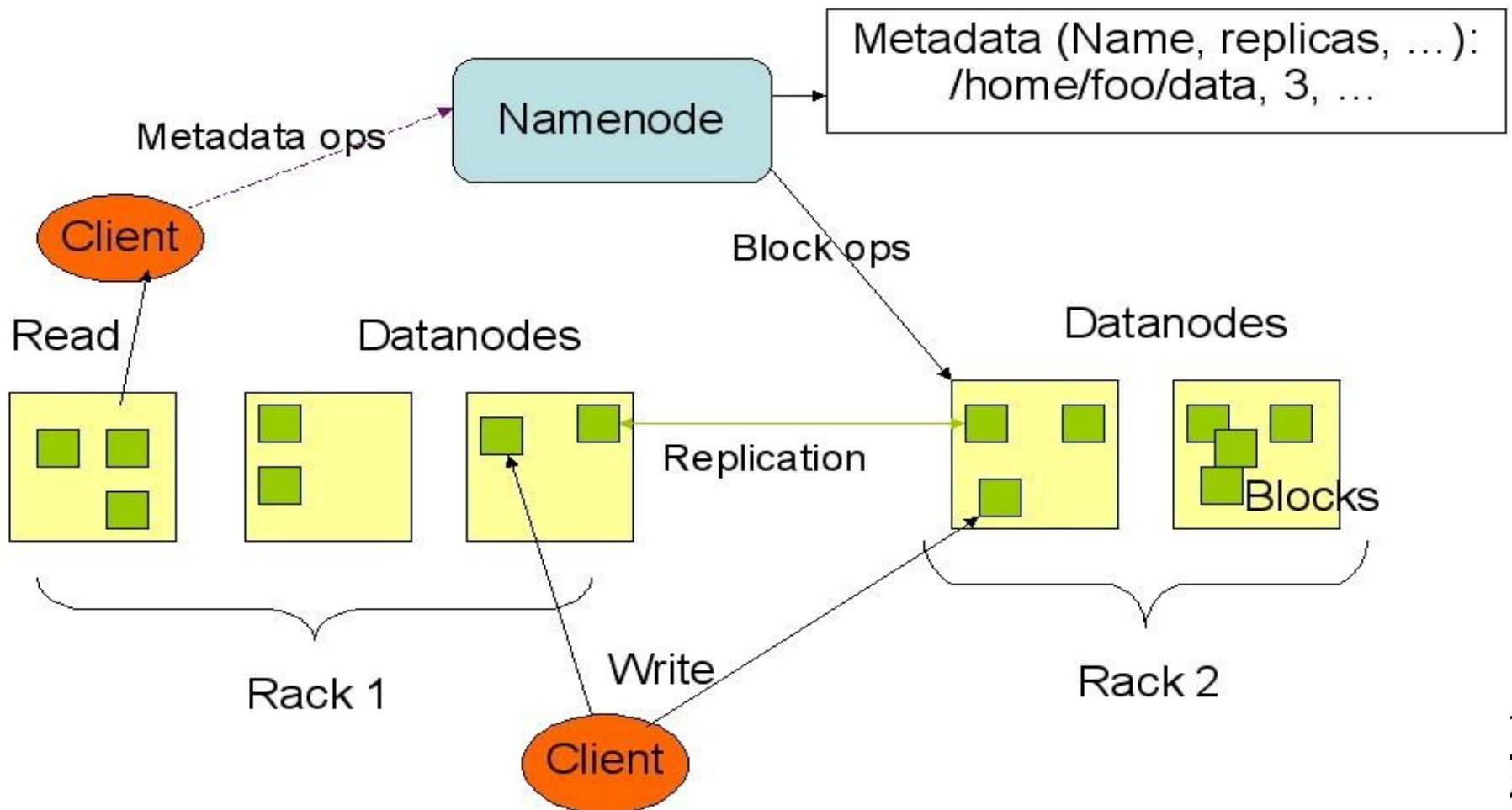


## 設計目標 (2)

- 一致性模型
  - 一次寫入，多次存取
  - 簡化一致性處理問題
- 在地運算
  - **移動到資料節點計算** > 移動資料過來計算
- 異質平台移植性
  - 即使硬體不同也可移植、擴充

# 管理資料

HDFS Architecture



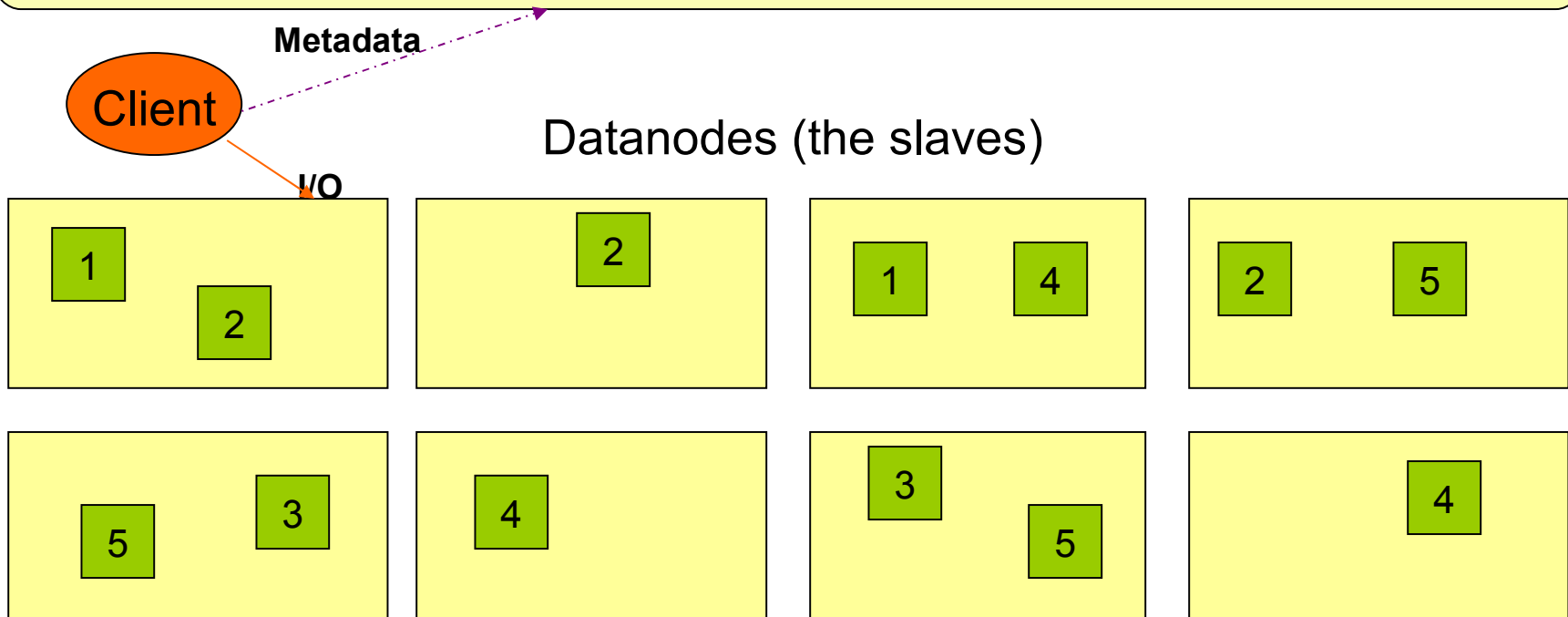
# HDFS 運作

Namenode (the master)

檔案路徑 - 副本數 , 由哪幾個 block 組成

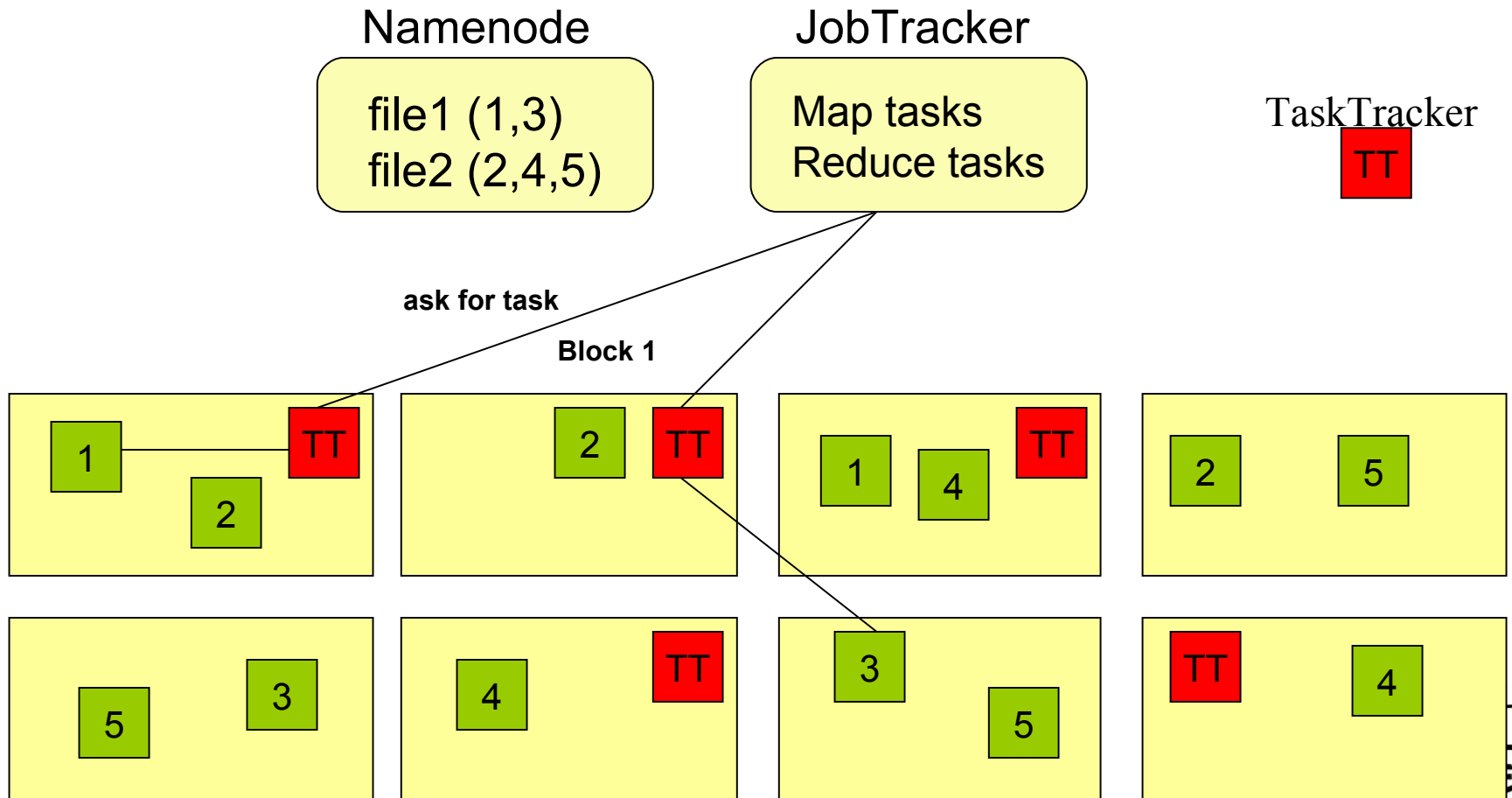
name:/users/joeYahoo/myFile - copies:2, blocks:{1,3}

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



# HDFS 運作

- 目的：提高系統的可靠性與讀取的效率
  - 可靠性：節點失效時讀取副本已維持正常運作
  - 讀取效率：分散讀取流量（但增加寫入時效能瓶頸）

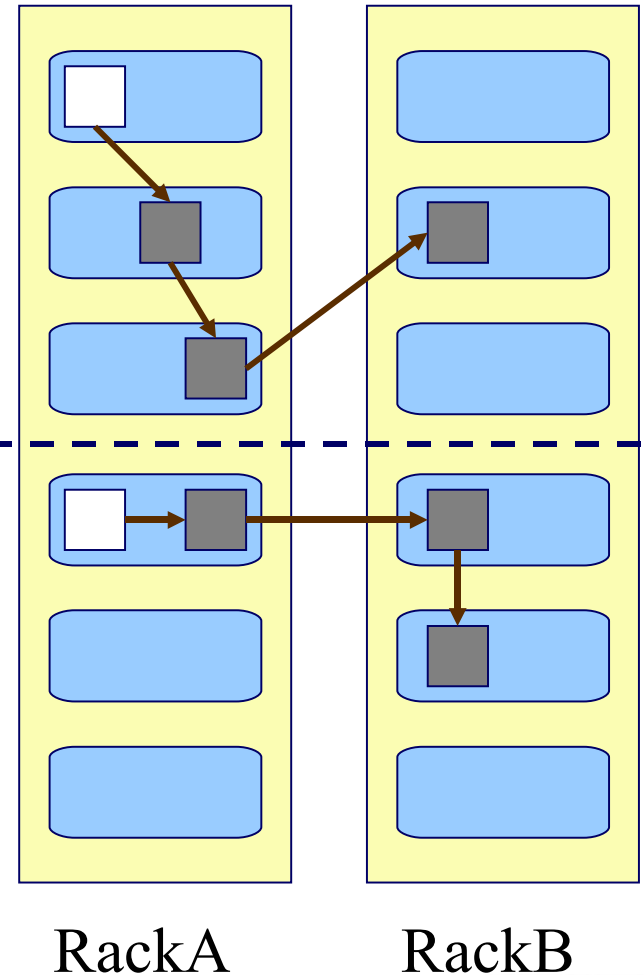


# HDFS 副本備份機制

- Original ~
  - First：同機架的不同節點
  - Second：同機架的另一節點
  - Third：不同機架另一節點
  - More：隨機挑選

---

- Hadoop 0.17 ~
  - 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



如何達成  
其好處？

# 可靠性機制

常見的  
三種  
錯誤  
狀況

資料崩毀

網路或  
資料節點  
失效

名稱節點  
錯誤

- 資料完整性
  - checked with CRC32
  - 用副本取代出錯資料
- Heartbeat
  - Datanode 定期向 Namenode 送 heartbeat
- Metadata
  - FSImage 、 Editlog 為核心印象檔及日誌檔
  - 多份儲存，當 NameNode 壞掉可手動復原

# 一致性與效能機制

- 檔案一致性機制
  - 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
  - 以 Block 為單位： 64M 為單位
  - 在 HDFS 上得檔案有可能大過一顆磁碟
  - 大區塊可提高存取效率
  - 區塊均勻散佈各節點以分散讀取流量



# HDFS 的功能

- 類 POXIS 指令
- 權限控管
- 超級用戶模式
- Web 瀏覽
- 用戶配額管理
- 分散式複製檔案

功能為何  
?

# POSIX Like

```
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]]
```



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 介紹



王耀聰 陳威宇

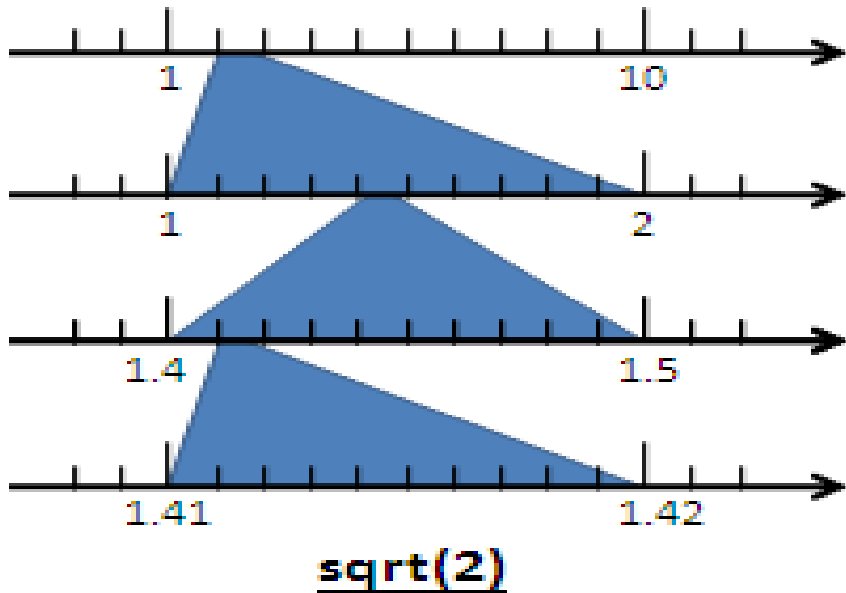
Jazz@nchc.org.tw

waue@nchc.org.tw

國家高速網路與計算中心  
(NCHC)

# Divide and Conquer

範例一：十分逼近法

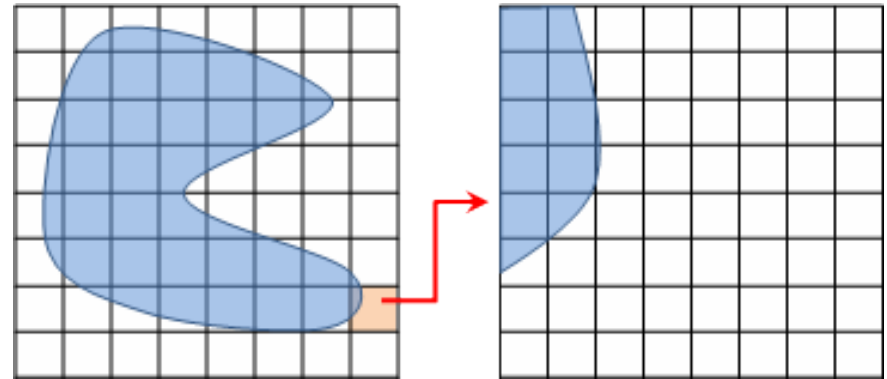


範例四：

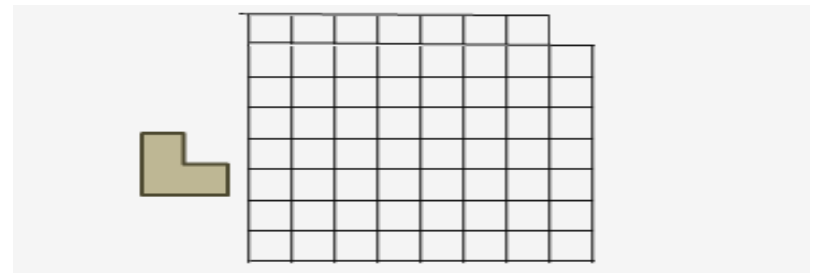
眼前有五階樓梯，每次可踏上一階或踏上兩階，那麼爬完五階共有幾種踏法？

Ex : (1,1,1,1,1) or (1,2,1,1)

範例二：方格法求面積



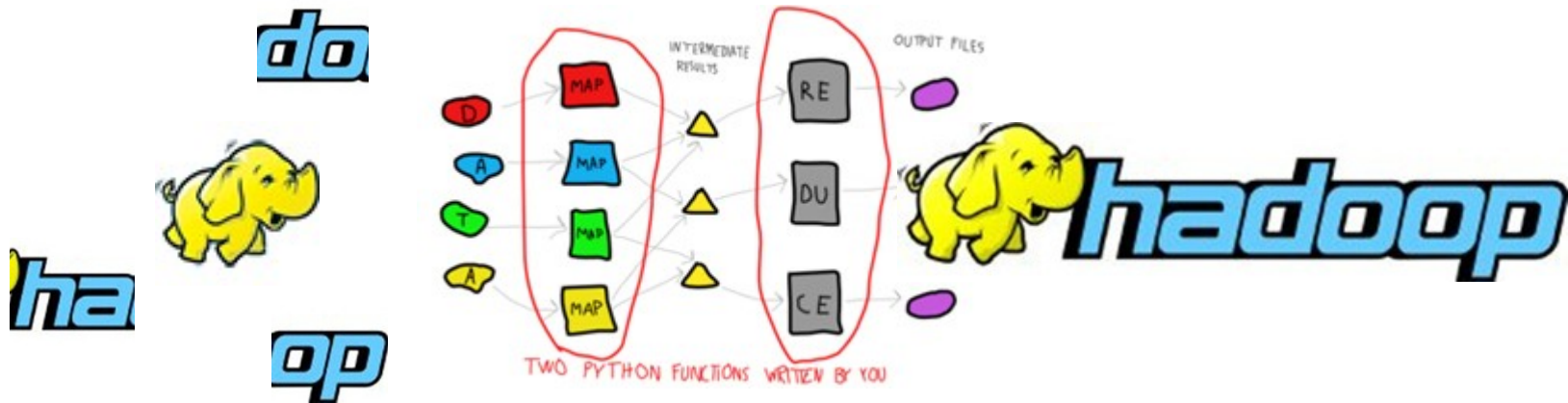
範例三：鋪滿 L 形磁磚



# Map Reduce 起源

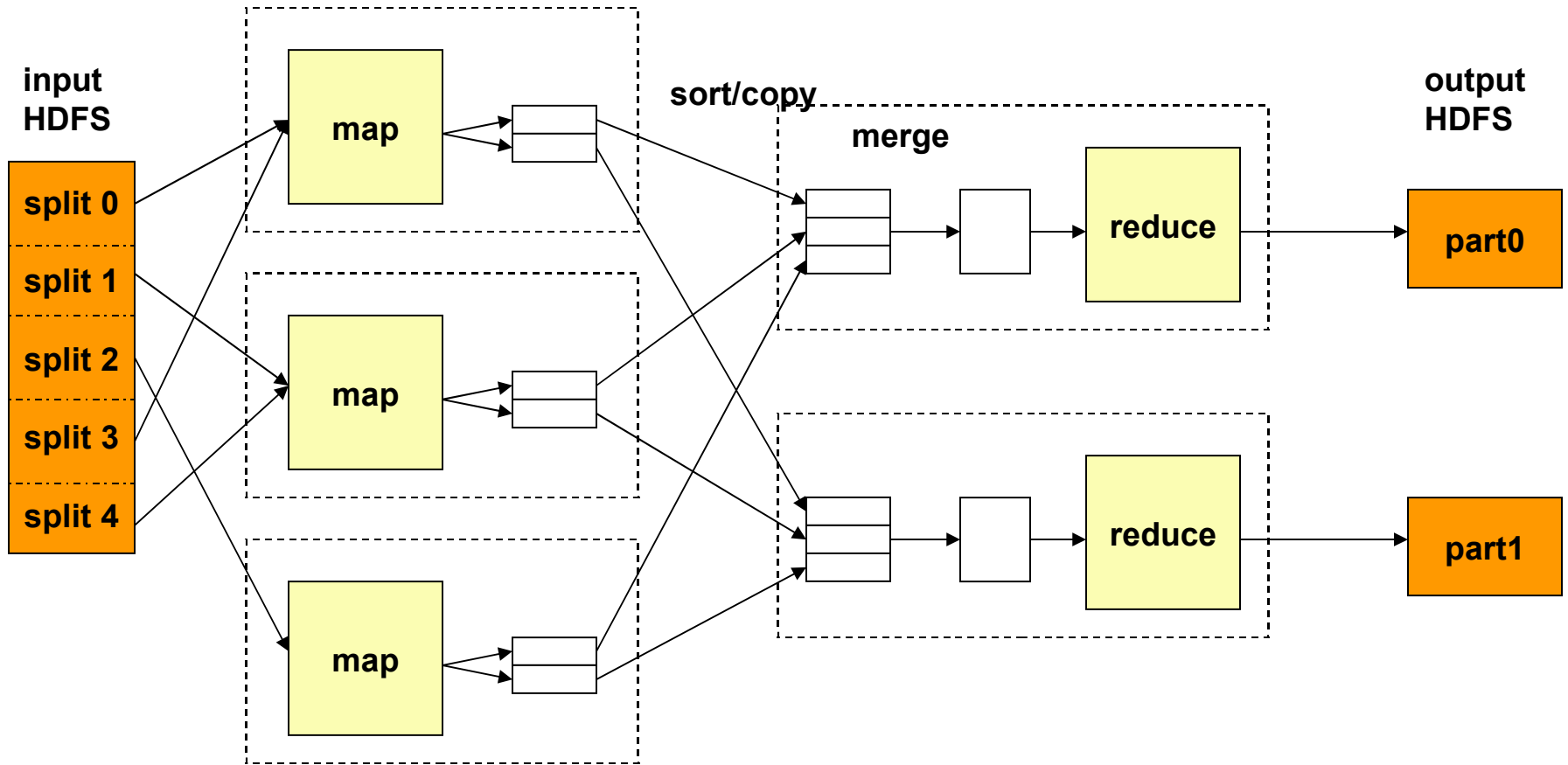
- Functional Programming : Map Reduce
  - map(...) :
    - [ 1,2,3,4 ] - (\*2) -> [ 2,4,6,8 ]
  - reduce(...):
    - [ 1,2,3,4 ] - (sum) -> 10
- 演算法 ( Algorithms ) :
  - Divide and Conquer
  - 分而治之
- 在程式設計的軟體架構內，適合使用在大規模數據的運算中

# 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。

# Hadoop-MapReduce 運作流程



JobTracker 跟 NameNode 取得需要運算的 blocks

JobTracker 選數個 TaskTracker 來作 Map 運算，產生些中間檔案

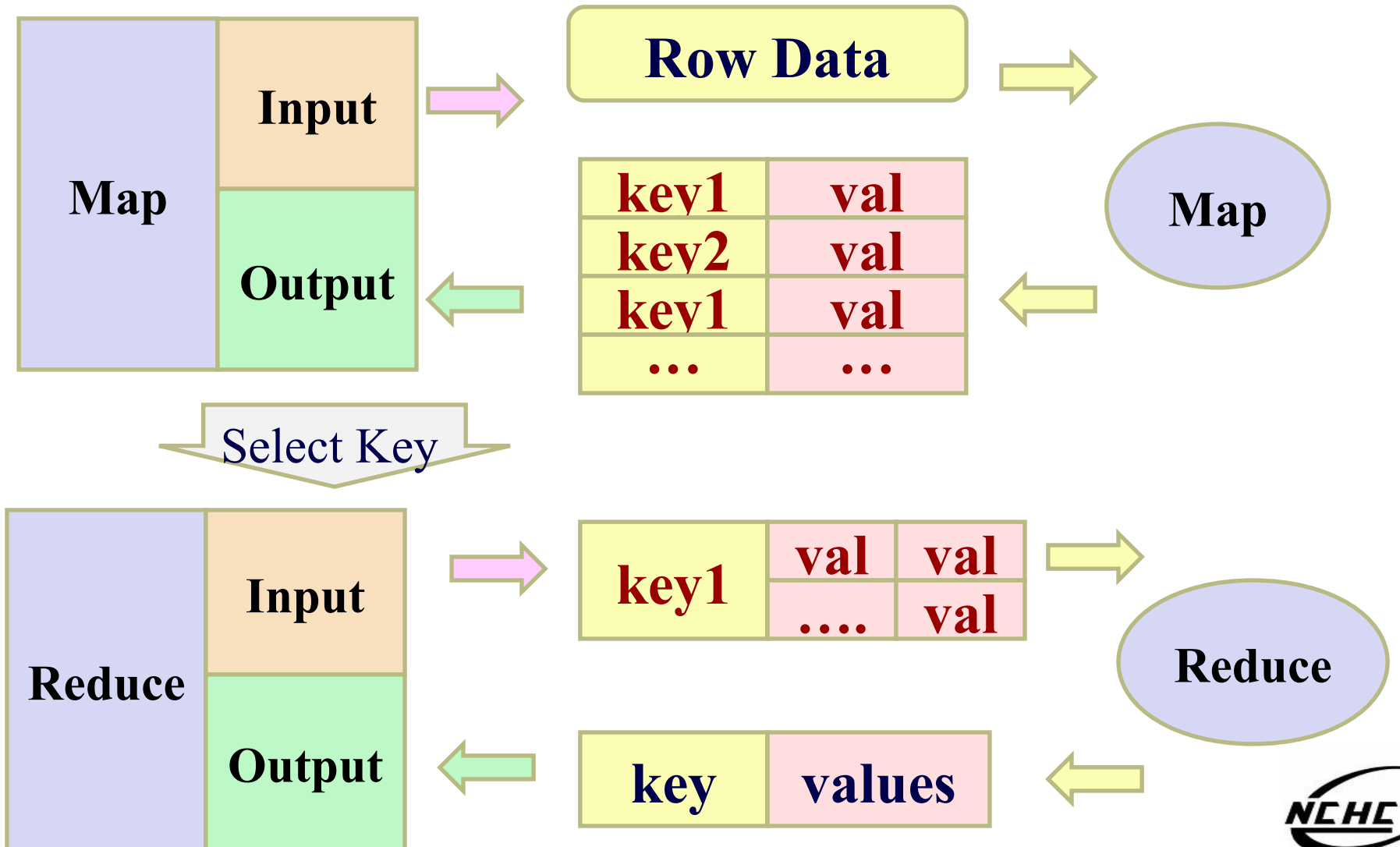
JobTracker 將中間檔案整合排序後，複製到需要的 TaskTracker 去

JobTracker 派遣 TaskTracker 作 reduce

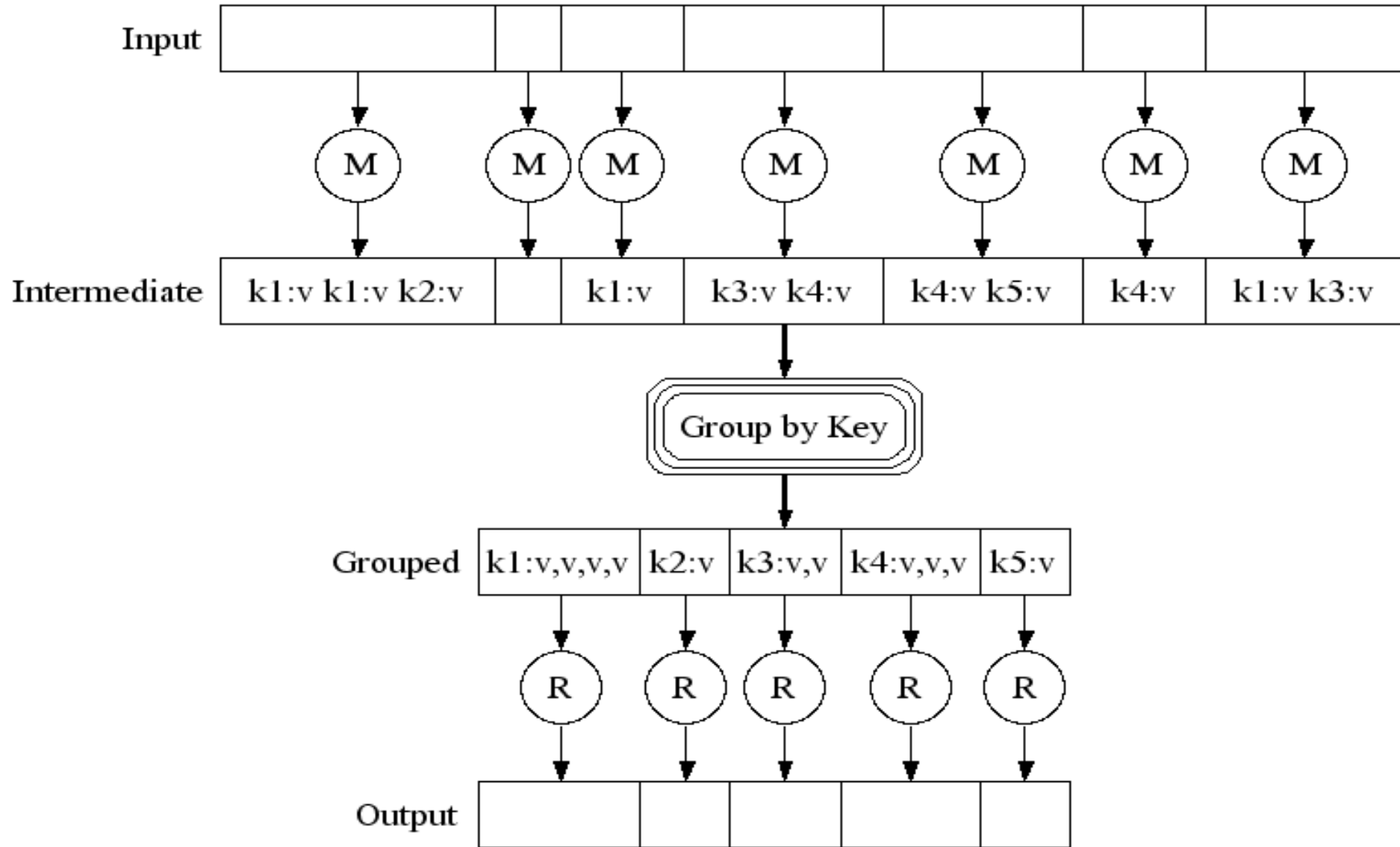
reduce 完後通知 JobTracker 與 Namenode 以產生 output



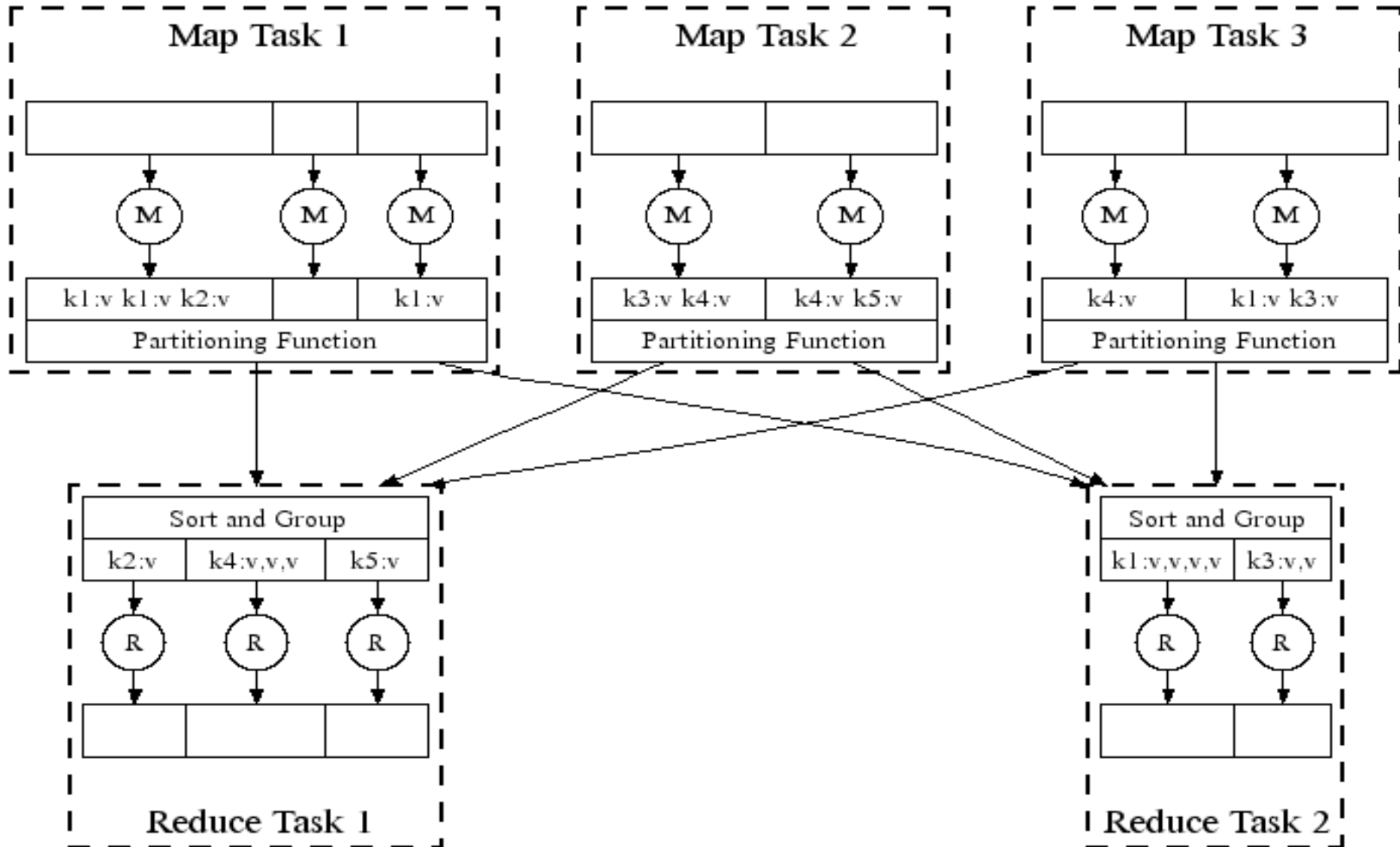
# MapReduce 與 $\langle \text{Key}, \text{Value} \rangle$



# MapReduce 圖解

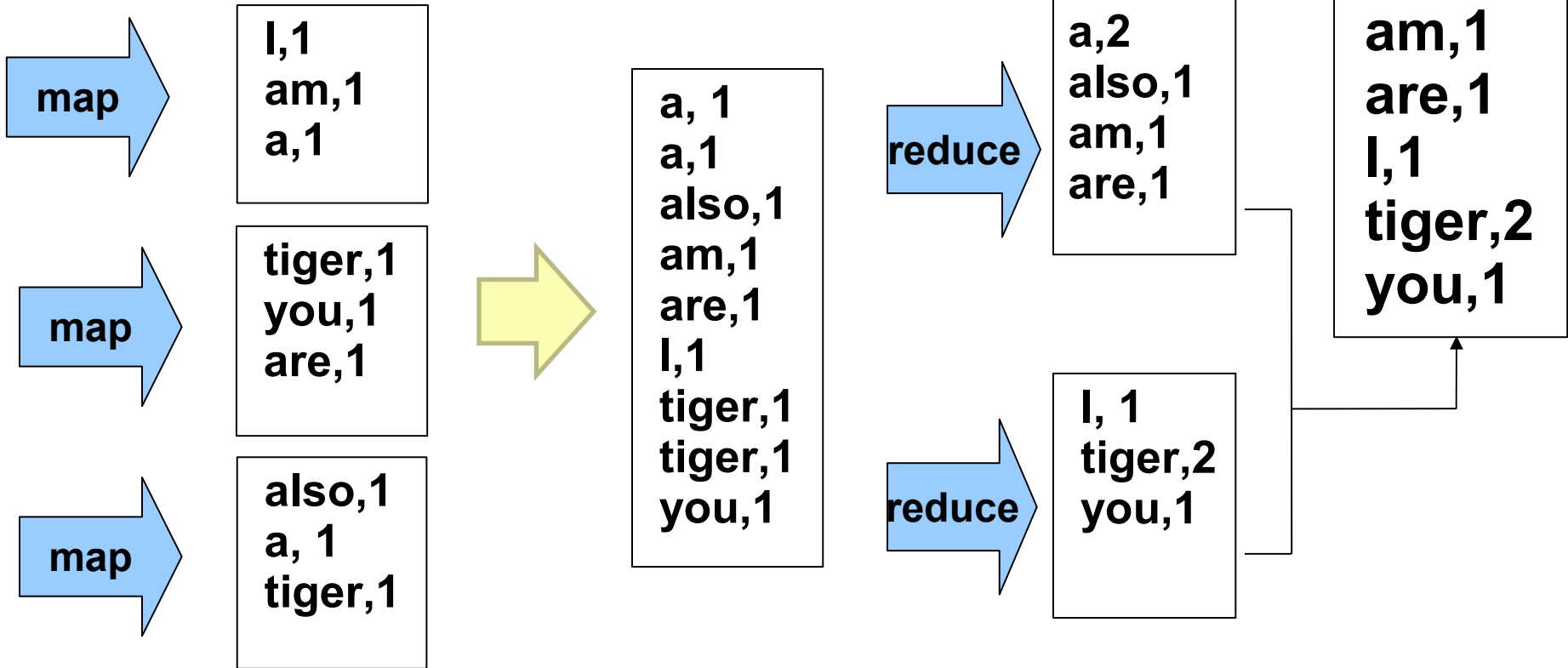


# MapReduce in Parallel



# 範例

I am a tiger, you are also a tiger



JobTracker 先選了三個 Tracker 做 map

Map 結束後，hadoop 進行中間資料的整理與排序

JobTracker 再選兩個 TaskTracker 作 reduce

# Hadoop 適用於 ..

- 大規模資料集
- 可拆解
  - Text tokenization
  - Indexing and Search
  - Data mining
  - machine learning
  - ...

- <http://www.dbms2.com/2008/08/26/known-applications-of-mapreduce/>
- <http://wiki.apache.org/hadoop/PoweredBy>

# Hadoop Applications (1)

- Adobe
  - use Hadoop and HBase in several areas from **social services** to structured data storage and processing for **internal use**.
- Adknowledge - Ad network
  - used to build the recommender system for **behavioral targeting**, plus other **clickstream analytics**
- Alibaba
  - processing **sorts of business data** dumped out of database and joining them together. These data will then be fed into **iSearch**, our vertical search engine.
- AOL
  - We use hadoop for variety of things ranging from **ETL style processing** and **statistics generation** to running advanced algorithms for doing **behavioral analysis**

# Hadoop Applications (2)

- **Baidu** - the leading Chinese language search engine
  - Hadoop used to analyze the **log of search and do some mining** work on web page database
- **Contextweb** - ADSDAQ Ad Exchange
  - use Hadoop to store ad serving log and use it as a source for **Ad optimizations/Analytics/reporting/machine learning**.
- **Detikcom** - Indonesia's largest news portal
  - use hadoop, pig and hbase to analyze **search log, generate Most View News,**
  - generate top **wordcloud**, and analyze all of our **logs**

# Hadoop Applications (3)

- DropFire
  - generate **Pig Latin** scripts that describe structural and semantic conversions between data contexts
  - use Hadoop to **execute these scripts** for production-level deployments
- Facebook
  - use Hadoop to store copies of internal log and dimension data sources
  - use it as a source for reporting/analytics and machine learning.
- Freestylers - Image retrieval engine
  - use Hadoop 影像處理
- Hosting Habitat
  - 取得所有 clients 的軟體資訊
  - 分析並告知 clients 未安裝或未更新的軟體



# Hadoop Applications (4)

- IBM
  - Blue Cloud Computing Clusters
- ICCS
  - 用 Hadoop and Nutch to crawl Blog posts 並分析之
- IIT, Hyderabad
  - We use hadoop 資訊檢索與提取
- Journey Dynamics
  - 用 Hadoop MapReduce 分析 billions of lines of GPS data 並產生交通路線資訊。
- Krugle
  - 用 Hadoop and Nutch 建構 原始碼搜尋引擎

# Hadoop Applications (5)

- **SEDNS - Security Enhanced DNS Group**
  - 收集全世界的 DNS 以探索網路分散式內容。
- **Technical analysis and Stock Research**
  - 分析股票資訊
- **University of Maryland**
  - 用 Hadoop 執行 machine translation, language modeling, bioinformatics, email analysis, and image processing 相關研究
- **University of Nebraska Lincoln, Research Computing Facility**
  - 用 Hadoop 跑約 200TB 的 CMS 經驗分析
  - 緊湊渺子線圈（ CMS ， Compact Muon Solenoid ）為瑞士歐洲核子研究組織 CERN 的大型強子對撞器計劃的兩大通用型粒子偵測器中的一個。

# Hadoop Applications (6)

- PARC
  - Used Hadoop to analyze Wikipedia conflicts
- Search Wikia
  - A project to help develop open source social search tools
- Yahoo!
  - Used to support research for Ad Systems and Web Search
  - 使用 Hadoop 平台來發現發送垃圾郵件的殭屍網絡
- 趨勢科技
  - 過濾像是釣魚網站或惡意連結的網頁內容

# 安裝設定補充說明

王耀聰 陳威宇

[jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)

[waue@nchc.org.tw](mailto:waue@nchc.org.tw)

國家高速網路與計算中心 (NCHC)



# Hadoop Package Topology

## 資料夾

## 說明

bin /	各執行檔：如 <code>start-all.sh</code> 、 <code>stop-all.sh</code> 、 <code>hadoop</code>
conf /	預設的設定檔目錄：設定環境變數 <code>hadoop-env.sh</code> 、各項參數 <code>hadoop-site.conf</code> 、工作節點 <code>slaves</code> 。 (可更改路徑)
docs /	Hadoop API 與說明文件 (html & PDF)
contrib /	額外有用的功能套件，如： <code>eclipse</code> 的擴充外掛、 <code>Streaming</code> 函式庫。
lib /	開發 <code>hadoop</code> 專案或編譯 <code>hadoop</code> 程式所需要的所有函式庫，如： <code>jetty</code> 、 <code>kfs</code> 。 但主要的 <code>hadoop</code> 函式庫於 <code>hadoop_home</code>
src /	Hadoop 的原始碼。
build /	開發 Hadoop 編譯後的資料夾。需搭配 <code>ant</code> 程式與 <code>build.xml</code>
logs /	預設的日誌檔所在目錄。 (可更改路徑)

# 設定檔：hadoop-env.sh

- 設定 Linux 系統執行 Hadoop 的環境參數
  - export xxx=kkk
    - 將 kkk 這個值匯入到 xxx 參數中
  - # string...
    - 註解，通常用來描述下一行的動作內容

```
# The java implementation to use. Required.  
export JAVA_HOME=/usr/lib/jvm/java-6-sun  
export HADOOP_HOME=/opt/hadoop  
export HADOOP_LOG_DIR=$HADOOP_HOME/logs  
export HADOOP_SLAVES=$HADOOP_HOME/conf/slaves  
.....
```

# 設定檔：hadoop-site.xml (0.18)

## <configuration>

```
<property>
  <name> fs.default.name</name>
  <value> hdfs://localhost:9000/</value>
  <description> ... </description>
</property>
```

```
<property>
  <name> mapred.job.tracker</name>
  <value> localhost:9001</value>
  <description>... </description>
</property>
```

```
<property>
  <name> hadoop.tmp.dir </name>
  <value> /tmp/hadoop/hadoop-$
    {user.name} </value>
  <description> </description>
</property>
```

```
<property>
  <name> mapred.map.tasks</name>
  <value> 1</value>
  <description> define mapred.map tasks to be
    number of slave hosts </description>
</property>
<property>
  <name> mapred.reduce.tasks</name>
  <value> 1</value>
  <description> define mapred.reduce tasks to be
    number of slave hosts </description>
</property>
<property>
  <name> dfs.replication</name>
  <value> 3</value>
</property>
```

## </configuration>

# 設定檔：hadoop-default.xml (0.18)

- Hadoop 預設參數
  - 沒在 `hadoop.site.xml` 設定的話就會用此檔案的值
  - 更多的介紹參數：[http://hadoop.apache.org/core/docs/current/cluster\\_setup.html#Configuring+the-](http://hadoop.apache.org/core/docs/current/cluster_setup.html#Configuring+the-)



# Hadoop 0.18 到 0.20 的轉變

***hadoop-site.xml***

***core-site.xml***

***mapreduce-core.xml***

***hdfs-site.xml***

***hadoop-site.xml***

***src/core/core-default.xml***

***src/mapred/mapred-default.xml***

***src/hdfs/hdfs-default.xml***

# 設定檔： core-site.xml (0.20)

## <configuration>

```
<property>  
  <name> fs.default.name</name>  
  <value> hdfs://localhost:9000/</value>  
  <description> ... </description>  
</property>
```

```
<property>  
  <name> hadoop.tmp.dir </name>  
  <value> /tmp/hadoop/hadoop-$  
    {user.name} </value>  
  <description> ... </description>  
</property>
```

## <configuration>

詳細 hadoop core 參數，

請參閱 <http://hadoop.apache.org/common/docs/current/core-default.html>

# 設定檔： mapreduce-site.xml (0.20)

## <configuration>

```
<property>
  <name> mapred.job.tracker</name>
  <value> localhost:9001</value>
  <description>... </description>
</property>
```

```
<property>
  <name> mapred.map.tasks</name>
  <value> 1</value>
  <description> ... </description>
</property>
```

```
<property>
  <name> mapred.reduce.tasks</name>
  <value> 1</value>
  <description> ... </description>
</property>
```

## </configuration>

詳細 hadoop mapreduce 參數，

請參閱 <http://hadoop.apache.org/common/docs/current/mapred-default.html>

# 設定檔： hdfs-site.xml (0.20)

**<configuration>**

```
<property>  
  <name> dfs.replication </name>  
  <value> 3</value>  
  <description>... </description>  
</property>
```

```
<property>  
  <name> dfs.permissions </name>  
  <value> false </value>  
  <description> ... </description>  
</property>
```

**</configuration>**

詳細 hadoop hdfs 參數，

請參閱 <http://hadoop.apache.org/common/docs/current/hdfs-default.html>



# 設定檔： slaves

- 給 start-all.sh , stop-all.sh 用
- 被此檔紀錄到的節點就會附有兩個身份：  
datanode & tasktracker
- 一行一個 hostname 或 ip

```
192.168.1.1  
....  
192.168.1.100  
Pc101  
....  
Pc152  
....
```

## 設定檔： masters

- 給 start-\*.sh , stop-\*.sh 用
- 會被設定成 secondary namenode
- 可多個

192.168.1.1

....

Pc101

....

描述名稱	設定名稱	所在檔案
JAVA 安裝目錄	JAVA_HOME	hadoop-env.sh
HADOOP 家目錄	HADOOP_HOME	hadoop-env.sh
設定檔目錄	HADOOP_CONF_DIR	hadoop-env.sh
日誌檔產生目錄	HADOOP_LOG_DIR	hadoop-env.sh
HADOOP 工作目錄	hadoop.tmp.dir	hadoop-site.xml
JobTracker	mapred.job.tracker	hadoop-site.xml
Namenode	fs.default.name	hadoop-site.xml
TaskTracker	(hostname)	slaves
Datanode	(hostname)	slaves
第二 Namenode	(hostname)	masters
其他設定值	詳可見 hadoop-default.xml	hadoop-site.xml

# 控制 Hadoop 的指令

- 格式化
  - \$ bin/hadoop \_ namenode \_ -format
- 全部開始 ( 透過 SSH )
  - \$ bin/start-all.sh
  - \$ bin/start-dfs.sh
  - \$ bin/start-mapred.sh
- 全部結束 ( 透過 SSH )
  - \$ bin/stop-all.sh
  - \$ bin/stop-dfs.sh
  - \$ bin/stop-mapred.sh
- 獨立啟動 / 關閉 ( 不會透過 SSH )
  - \$ bin/hadoop-daemon.sh [start/stop] namenode
  - \$ bin/hadoop-daemon.sh [start/stop] secondarynamenode
  - \$ bin/hadoop-daemon.sh [start/stop] datanode
  - \$ bin/hadoop-daemon.sh [start/stop] jobtracker
  - \$ bin/hadoop-daemon.sh [start/stop] tasktracker



# Hadoop 的操作與運算指令

- 使用 hadoop 檔案系統指令
  - \$ bin/hadoop fs -Instruction ...
- 使用 hadoop 運算功能
  - \$ bin/hadoop jar XXX.jar Main\_Function ...

# Hadoop 使用者指令

\$ bin/hadoop **△ 指令** △ 選項 △ 參數 △ ....

指令	用途	舉例
<b>fs</b>	對檔案系統進行操作	hadoop△ <b>fs</b> △-put△in△input
<b>jar</b>	啟動運算功能	hadoop△ <b>jar</b> △example.jar△wc△in△ out
<b>archive</b>	封裝 hdfs 上的資料	hadoop△ <b>archive</b> △foo.har△/dir △/user/hadoop
<b>distcp</b>	用於叢集間資料傳輸	hadoop△ <b>distcp</b> △hdfs://nn1:9000/aa △hdfs://nn2:9000/aa
<b>fsck</b>	hdfs 系統檢查工具	hadoop△ <b>fsck</b> △/aa△-files△-blocks △-locations
<b>job</b>	操作正運算中的程序	hadoop△ <b>job</b> △-kill△jobID
<b>version</b>	顯示版本	hadoop△ <b>version</b>

# Hadoop 管理者指令

\$ bin/hadoop  $\Delta$  指令  $\Delta$  選項  $\Delta$  參數  $\Delta$  ....

指令	用途	舉例
<b>balancer</b>	平衡 hdfs 覆載量	hadoop $\Delta$ <b>balancer</b>
<b>dfsadmin</b>	配額、安全模式 等管理員操作	hadoop $\Delta$ <b>dfsadmin</b> $\Delta$ -setQuota $\Delta$ 3 $\Delta$ /user1/
<b>namenode</b>	名稱節點操作	hadoop $\Delta$ <b>namenode</b> $\Delta$ -format

\$ bin/hadoop  $\Delta$  指令

<b>datanode</b>	成為資料節點	hadoop $\Delta$ <b>datanode</b>
<b>jobtracker</b>	成為工作分派者	hadoop $\Delta$ <b>jobtracker</b>
<b>tasktracker</b>	成為工作執行者	hadoop $\Delta$ <b>tasktracker</b>



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

Jazz@nchc.org.tw

waue@nchc.org.tw

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

# Program Prototype (v 0.18)



# Class Mapper

```
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 Reducer

```
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 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。
- 可不用設定交由 Hadoop 預設
- 也可不實做此程式，引用 Reducer
- 設定
  - `JobConf.setCombinerClass(Class)`

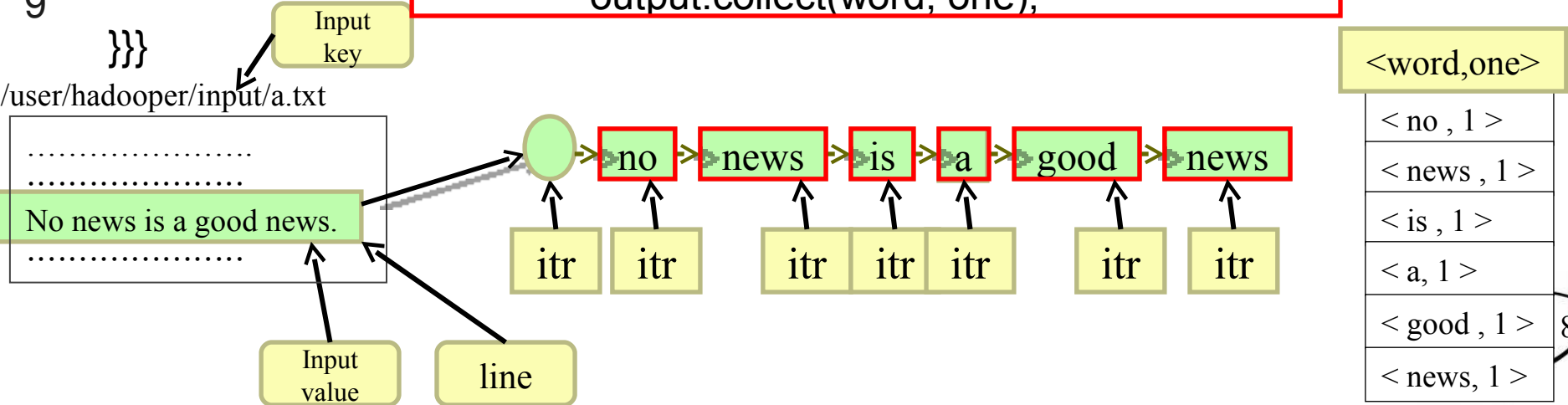
# Run Job

- `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。

# 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);  
        }  
    }  
}
```

```
String line = ((Text) value).toString();  
StringTokenizer itr = new StringTokenizer(line);  
while (itr.hasMoreTokens()) {  
    word.set(itr.nextToken());  
    output.collect(word, one);  
}
```

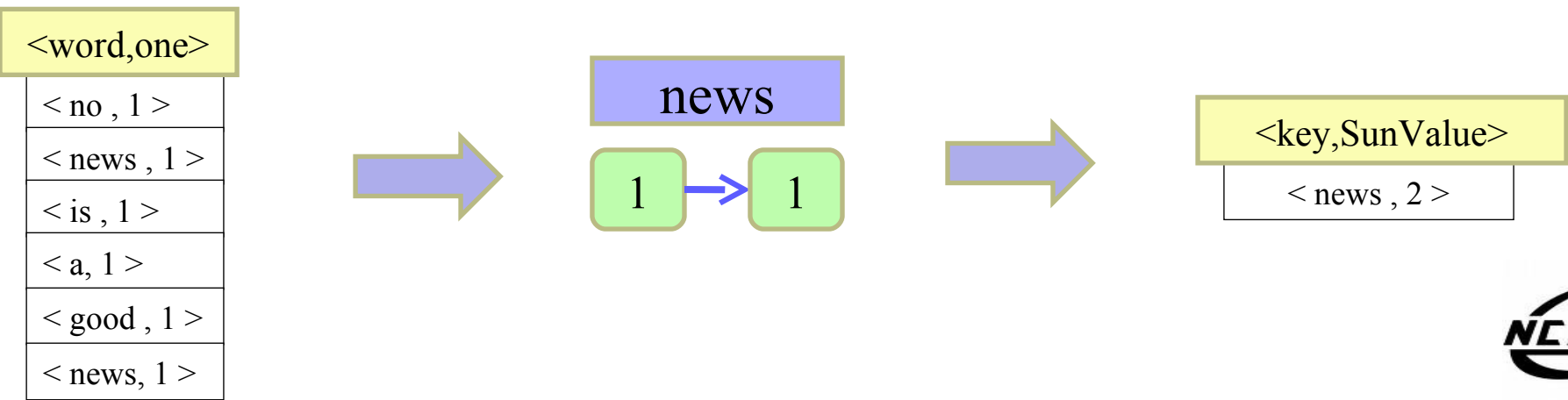


# Word Count Sample (2)

```

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);
    }}

```



# 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

- 編譯

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

- 封裝

- `jar -cvf MyJar.jar -C MyJava .`

- 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

- `MyJar.jar` = 封裝後的編譯檔

- 先放些文件檔到 HDFS 上的 input 目錄

- `./input`; `./output` = 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



# 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
```

# 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 = hdfs://host:port/FilePath

# 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





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

---

# Nutch 簡介

王耀聰 陳威宇

Jazz@nchc.org.tw

waue@nchc.org.tw

國家高速網路與計算中心 (NCHC)

# Outline

- What is Nutch
- Why Nutch
- Nutch's Details
- Let's go

# What's Nutch

- Nutch 是一個 open source ，以 Java 來實做的搜索引擎，它提供了架設自己的搜索引擎所需的全部工具。
- 利用 Lucene 為函式庫
- **架構於 Hadoop 之上**

# Nutch's goals

- 每個月抓取幾十億網頁
- 為這些網頁維護索引
- 對索引文件進行每秒上千次的搜索
- 提供高質量的搜索結果
- 以最小的成本運作

# Why Nutch ?

- 透明
  - Opensource ， 資訊不隱藏
- 擴充
  - 有各種函式庫應用於分析不同檔案
- 隱私
  - 可應用於搜尋專屬資料
- 客製化
  - 可以之為基礎設計自己的 data mining 工具

# Who use Nutch

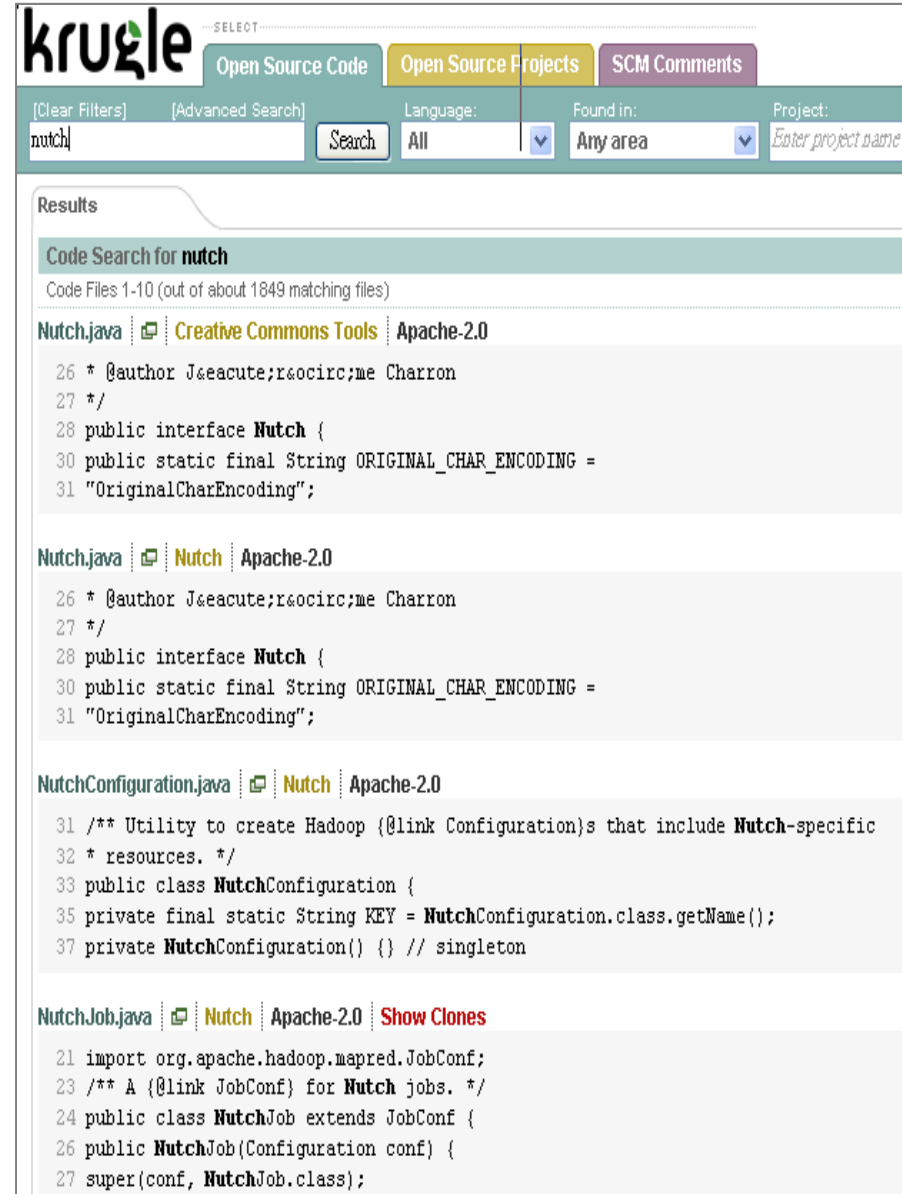
## Public search engines using Nutch

Please sort by name alphabetically

- [AskAboutOil](#) is a vertical search portal for the petroleum industry.
- [Baynote](#) provides free hosted Nutch search for businesses.
- [BeThere BeSquare](#) is an Event Search Engine for the San Francisco category and get details about events in 4 different views.
- [Bigsearch.ca](#) uses nutch open source software to deliver its search results.
- [BusyTonight](#): Search for any event in the United States, by key from original source Web sites.
- [Central Budapest Search](#) is a search engine for English language events.
- [Circuit Scout](#) is a search engine for electrical circuits.
- [Comtec Search](#) is a search engine for UK Tour Operator Pack.
- [Coder-Suche.de](#) searches for coding stuff like apis, documents in english.
- [Cornell University Library](#) is collaborating with the research group pages based on Nutch. The nutch-based search engine is near top level.
- [Creative Commons](#) is a search engine for creative commons images.
- [Dadi360](#) Use nutch search engine for providing search of Chinese.
- [Ecolhub Web Search](#) an E. coli specific search engine based on Nutch thereby reducing the number of spurious hits. Searches can be on more resources getting added.
- [Epivista](#) is a search engine of epilepsy related web sites.
- [eroscanner](#) is a search engine for german adult stuff. (Watching NSFW)

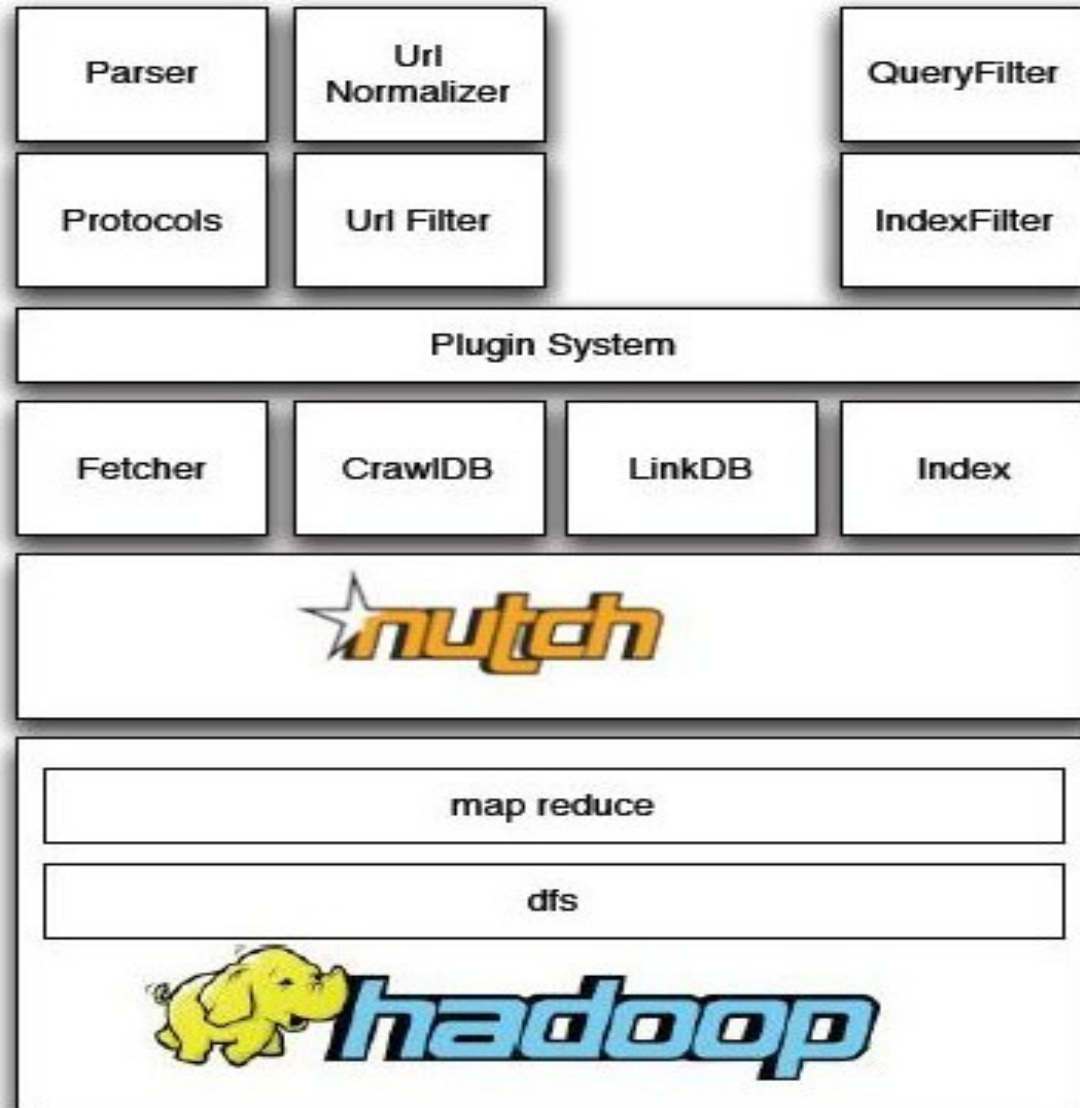
.....more

(<http://wiki.apache.org/nutch/PublicServers>)



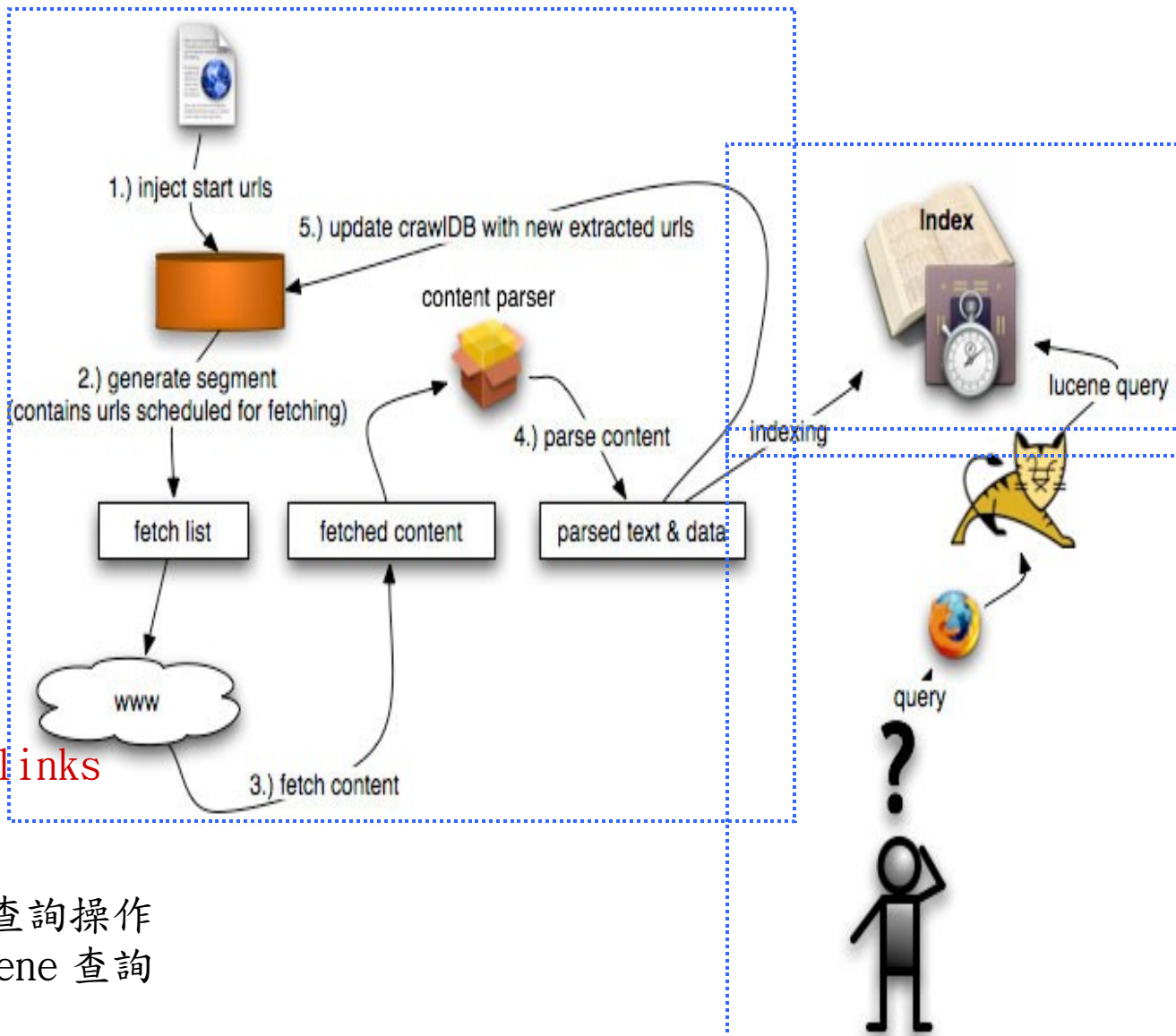
The screenshot shows the Krugle search engine interface. At the top, there are navigation tabs for "Open Source Code", "Open Source Projects", and "SCM Comments". Below these is a search bar with the text "nutch" entered. To the right of the search bar are buttons for "Search", "Language: All", "Found in: Any area", and "Project: Enter project name". Below the search bar, the results section is titled "Code Search for nutch" and shows "Code Files 1-10 (out of about 1849 matching files)". The first result is "Nutch.java" from "Creative Commons Tools" with "Apache-2.0" license, showing code lines 26-31. The second result is "Nutch.java" from "Nutch" with "Apache-2.0" license, also showing code lines 26-31. The third result is "NutchConfiguration.java" from "Nutch" with "Apache-2.0" license, showing code lines 31-37. The fourth result is "NutchJob.java" from "Nutch" with "Apache-2.0" license and a "Show Clones" link, showing code lines 21-27.

# 架構



# 運作流程

- 1) 建立初始 URL 集
- 2) 將 URL 集注入  
crawldb---inject
- 3) 根據 crawldb 建立抓取  
清單 ---generate
- 4) 執行抓取，獲取網頁內容  
---fetch
- 5) 用獲取到的頁面資訊更新  
crawldb---updatedb
- 6) 重複進行 3 ~ 5 的步驟  
，直到預先設定的抓取深度
- 7) 更新 linkdb ---invertlinks
- 8) 建立索引 ---index
- 9) 用戶通過用戶接口進行查詢操作
- 10) 將用戶查詢轉化為 lucene 查詢
- 11) 返回結果





# Plugin

- 修改 `conf/nutch-site.xml` 的 `plugin.includes` 屬性
- 在 `nutch` 基本功能之上擴充其功能
  - “`parse-xx`”：加入解析 `xx` 檔案類型的能力
  - “`protocol -xx`”：加入在此協定內的檔案也處理

**parse-text**

**parse-ext**

**parse-html**

**parse-js**

**parse-mp3**

**parse-zip**

**parse-rtf**

**parse-msword**

**parse-msexcel**

**parse-pdf**

**parse-rss**

**parse-oo**

**parse-swf**

**parse-mspowerpoint**

**protocol-file**

**protocol-ftp**

**protocol-http**

**protocol-httpclient**

# International

- 已有多國語言版可選，但若還要客製化…
- the page header
  - `src/web/include/language/header.xml`
- the "about" page
  - `src/web/pages/lang/about.xml`
- the "search" page
  - `src/web/pages/lang/search.xml`
- the "help" page
  - `src/web/pages/lang/help.xml`
- text for search results
  - `src/web/locale/org/nutch/jsp/search_lang.properties`

# No ! Nutch

- 告訴網頁機器人是否允許進入爬網
- 將 robots.txt 放在 web 上
- robots.txt

```
User-agent: Nutch  
Disallow: /
```

# Home Page



[About](#)

[FAQ](#)



[help](#)



[ca](#) | [de](#) | [en](#) | [es](#) | [fi](#) | [fr](#) | [hu](#) | [it](#) | [jp](#) | [ms](#) | [nl](#) | [pl](#) | [pt](#) | [sh](#) | [sr](#) | [sv](#) | [th](#) | [zh](#)

# Start

- **23 March 2009 - Apache Nutch 1.0 Released**

# Let's Go

# References..

- Nutch Website
  - <http://lucene.apache.org/nutch/>
- Nutch wiki
  - <http://wiki.apache.org/nutch/>
- Nutch API
  - <http://lucene.apache.org/nutch/apidocs-1.0/index>



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

# 設置 Hadoop 環境

王耀聰 陳威宇

Jazz@nchc.org.tw

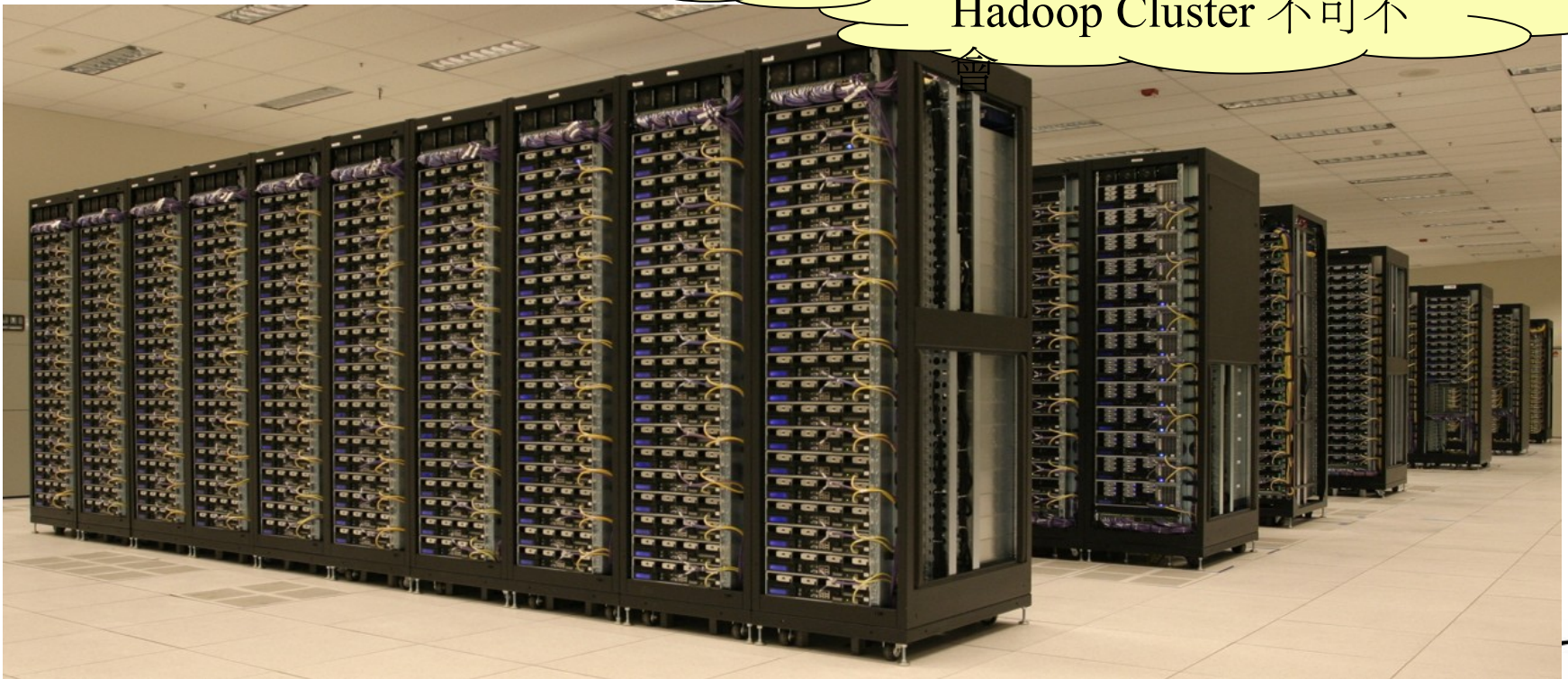
waue@nchc.org.tw

國家高速網路與計算中心 (NCHC)

# Yahoo's Hadoop Cluster

- ~10,000 machines running Hadoop in US
- The largest cluster is currently 2000 nodes
- Nearly 1 petabyte of user data (compressed, unreplicated)
- Running roughly 10,000 research jobs / week

Hadoop Cluster 不可不



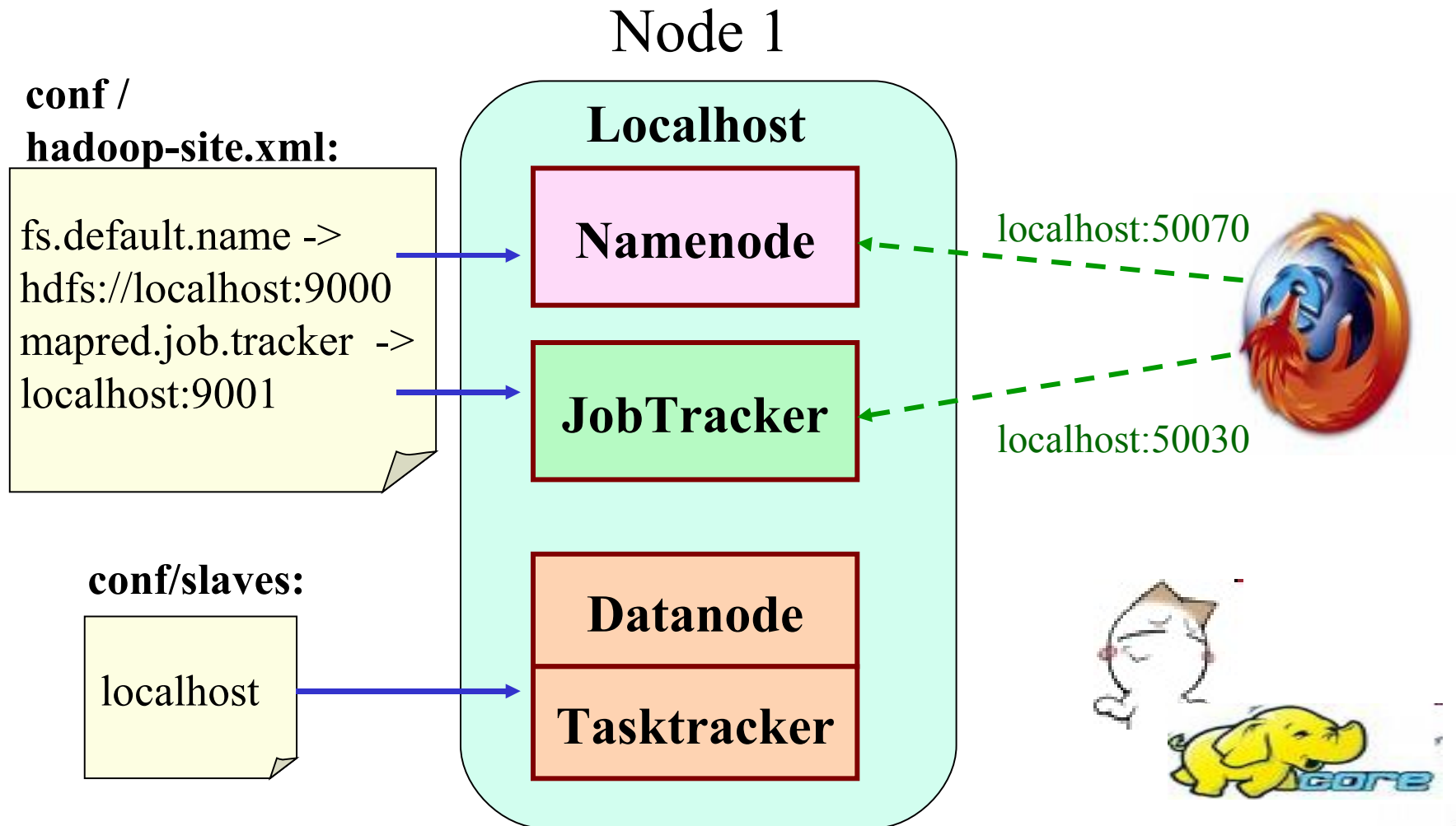


# Hadoop 單機設定與啟動

- step 1. 設定登入免密碼
- step 2. 安裝 java
- step 3. 下載安裝 Hadoop
- step 4.1 設定 `hadoop-env.sh`
  - `export JAVA_HOME=/usr/lib/jvm/java-6-sun`
- step 4.2 設定 `hadoop-site.xml`
  - 設定 Namenode-> `hdfs://localhost:9000`
  - 設定 Jobtracker -> `localhost:9001`
- step 5.1 格式化 HDFS
  - `bin/hadoop namenode -format`
- step 5.2 啟動 Hadoop
  - `bin/start-all.sh`
- step 6. 完成！檢查運作狀態
  - Job admin `http://localhost:50030/` HDFS `http://localhost:50070/`



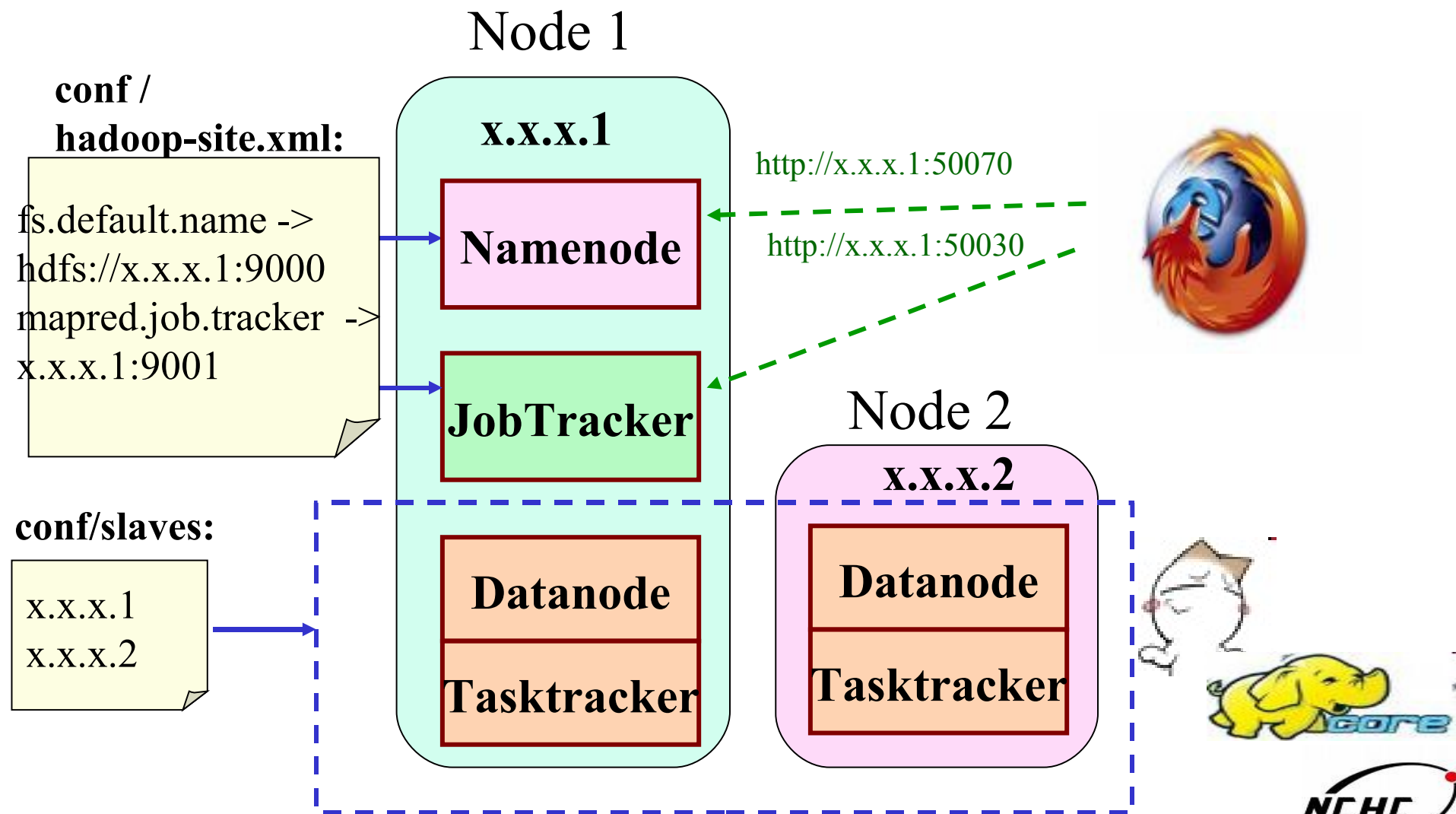
# Hadoop 單機環境示意圖



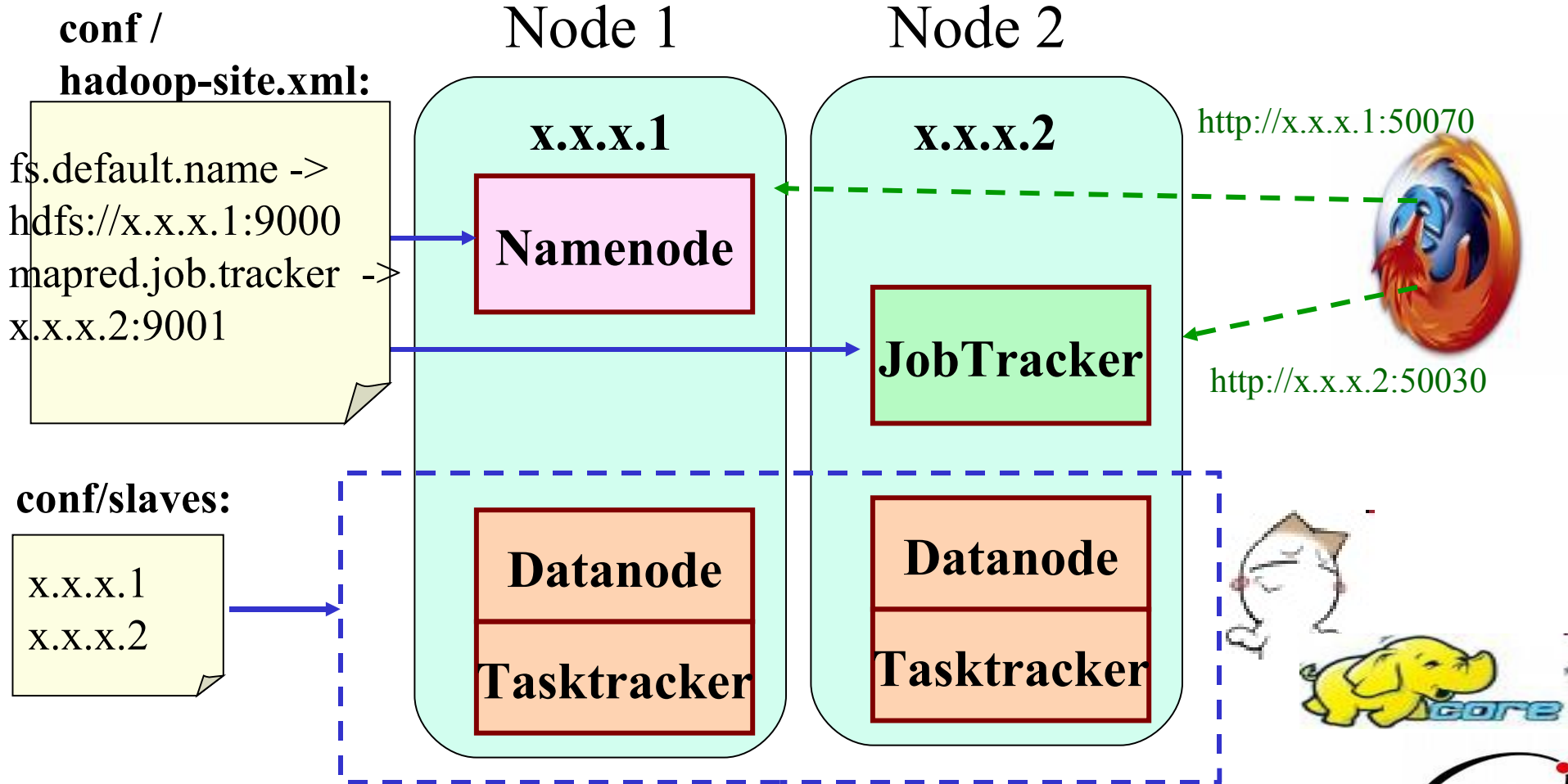
# Hadoop 叢集設定與啟動

- step 1. 設定登入免密碼
- step 2. 安裝 java
- step 3. 下載安裝 Hadoop
- step 4.1 設定 `hadoop-env.sh`
  - `export JAVA_HOME=/usr/lib/jvm/java-6-sun`
- step 4.2 設定 `hadoop-site.xml`
  - 設定 Namenode-> `hdfs://x.x.x.1:9000`
  - 設定 Jobtracker -> `x.x.x.2:9001`
- step 4.3 設定 slaves 檔
- step 4.4 將叢集內的電腦 Hadoop 都做一樣的配置
- step 5.1 格式化 HDFS
  - `bin/hadoop namenode -format`
- step 5.2 啟動 Hadoop
  - nodeN 執行：`bin/start-dfs.sh`；nodeJ 執行：`bin/start-mapred.sh`
- step 6. 完成！檢查運作狀態
  - Job admin `http://x.x.x.2:50030/` HDFS `http://x.x.x.1:50070/`

# 情況一



# 情況二



# 情況三

conf /  
hadoop.site.xml:

```
fs-default.name ->  
hdfs://x.x.x.1:9000  
mapred.job.tracker ->  
x.x.x.1:9001
```

Node 1

x.x.x.1

**Namenode**

http://x.x.x.1:50070

**JobTracker**

http://x.x.x.1:50030



conf/slaves:

```
x.x.x.2  
.....  
x.x.x.n
```

Node 2

x.x.x.2

**Datanode**

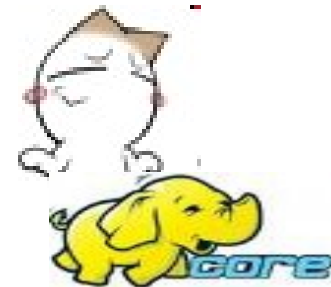
**Tasktracker**

Node N

x.x.x.n

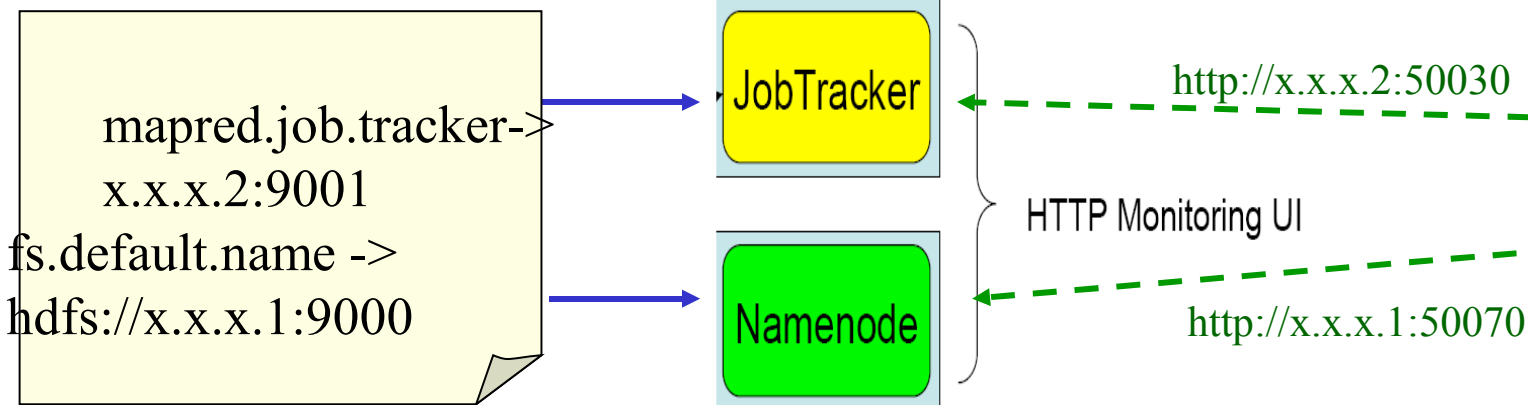
**Datanode**

**Tasktracker**



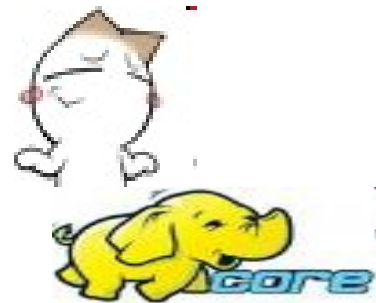
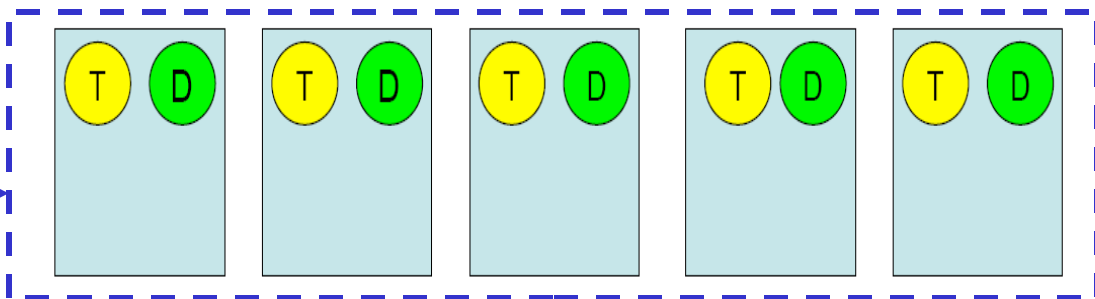
# 情況四

**conf /  
hadoop-site.xml:**



**conf/slaves:**

x.x.x.3  
.....  
x.x.x.n





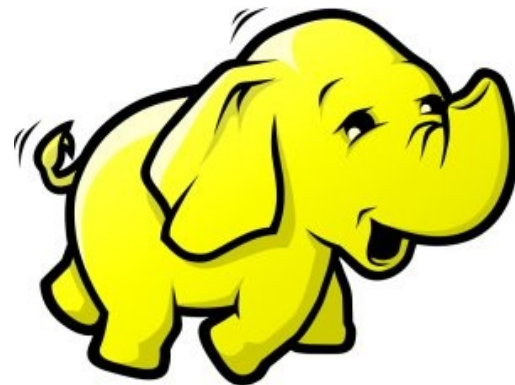
當企鵝龍遇上小飛象

***DRBL-Hadoop***

***Jazz Wang***

***Yao-Tsung Wang***

***jazz@nchc.org.tw***



Powered by **DRBL**



# ***Programmer v.s. System Admin.***



Source:  
<http://www.funnyjunksite.com/wp-content/uploads/2007/08/programmer.jpg>



Source:  
<http://www.sysadminday.com/images/people/136-3697.JPG>

# Agenda

## PART 1 :

**What is *Cluster Computing* ?**

**How to deploy PC cluster ?**

## PART 2 :

**What is *DRBL* and *Clonezilla* ?**

**Can *DRBL* help to *deploy Hadoop* ?**

## PART 3 :

**Live Demo of *DRBL Live*  
and *Clonezilla Live***



**PART 1 :**

# ***PC Cluster 101***

***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



Powered by **DRBL**



*At First, We have "4 + 1" PC Cluster*

*It'd better be*  
**2<sup>n</sup>**



*Manage*  
**Scheduler**

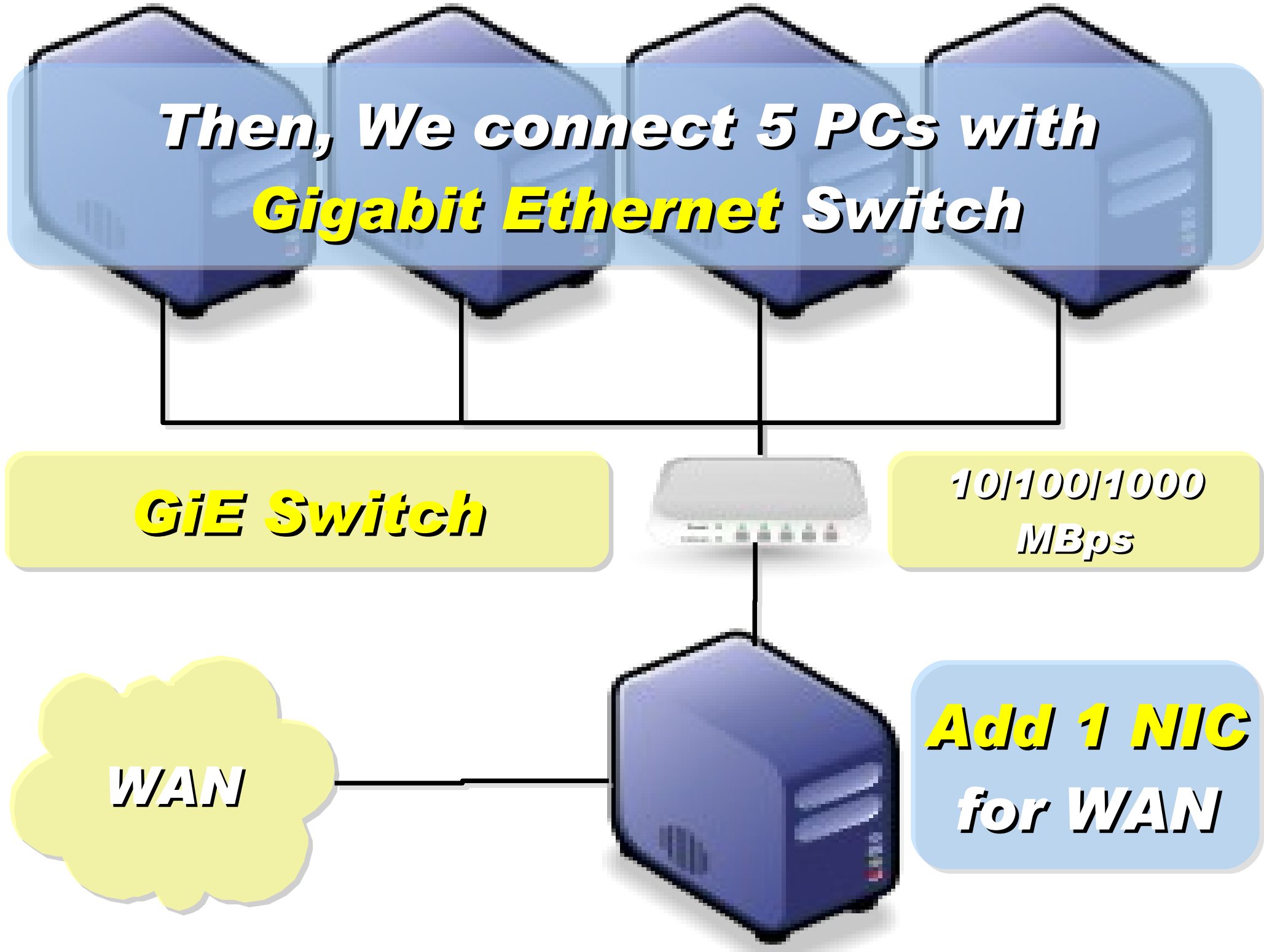
*Then, We connect 5 PCs with  
**Gigabit Ethernet Switch***

***GiE Switch***

*10/100/1000  
Mbps*

***WAN***

***Add 1 NIC  
for WAN***



**Compute Nodes**

**4 Compute Nodes will communicate via LAN Switch. Only Manage Node have Internet Access for Security!**

**WAN**

**Manage Node**



**Compute Nodes**

**Basic  
System  
Setup  
for  
Cluster**

*Messaging*

**MPICH**

*Account Mgmt.*

**SSHD**

**NIS**

**YP**

**GCC**

**GNU Libc**

**Bash**

**Perl**



**Kernel Module**

**Linux Kernel**

**Boot Loader**

On **Manage Node**,

We need to install **Scheduler** and **Network File System** for sharing Files with **Compute Node**

*Job Mgmt.*

**OpenPBS**

*File Sharing*

**NFS**

**Extra**

*Messaging*

**MPICH**

**GCC**

**Bash**

**Perl**

*Account Mgmt.*

**SSHD**

**NIS**

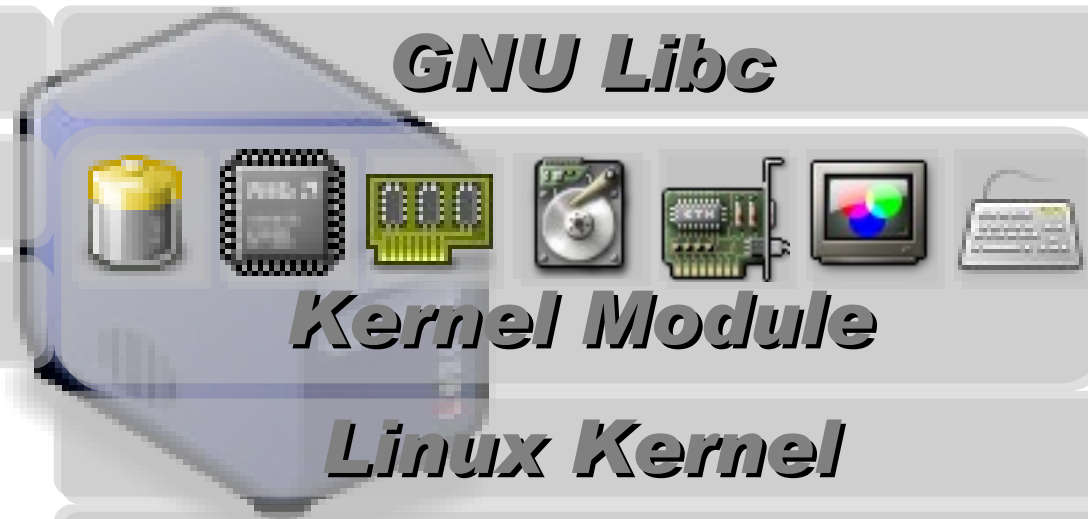
**YP**

**GNU Libc**

**Kernel Module**

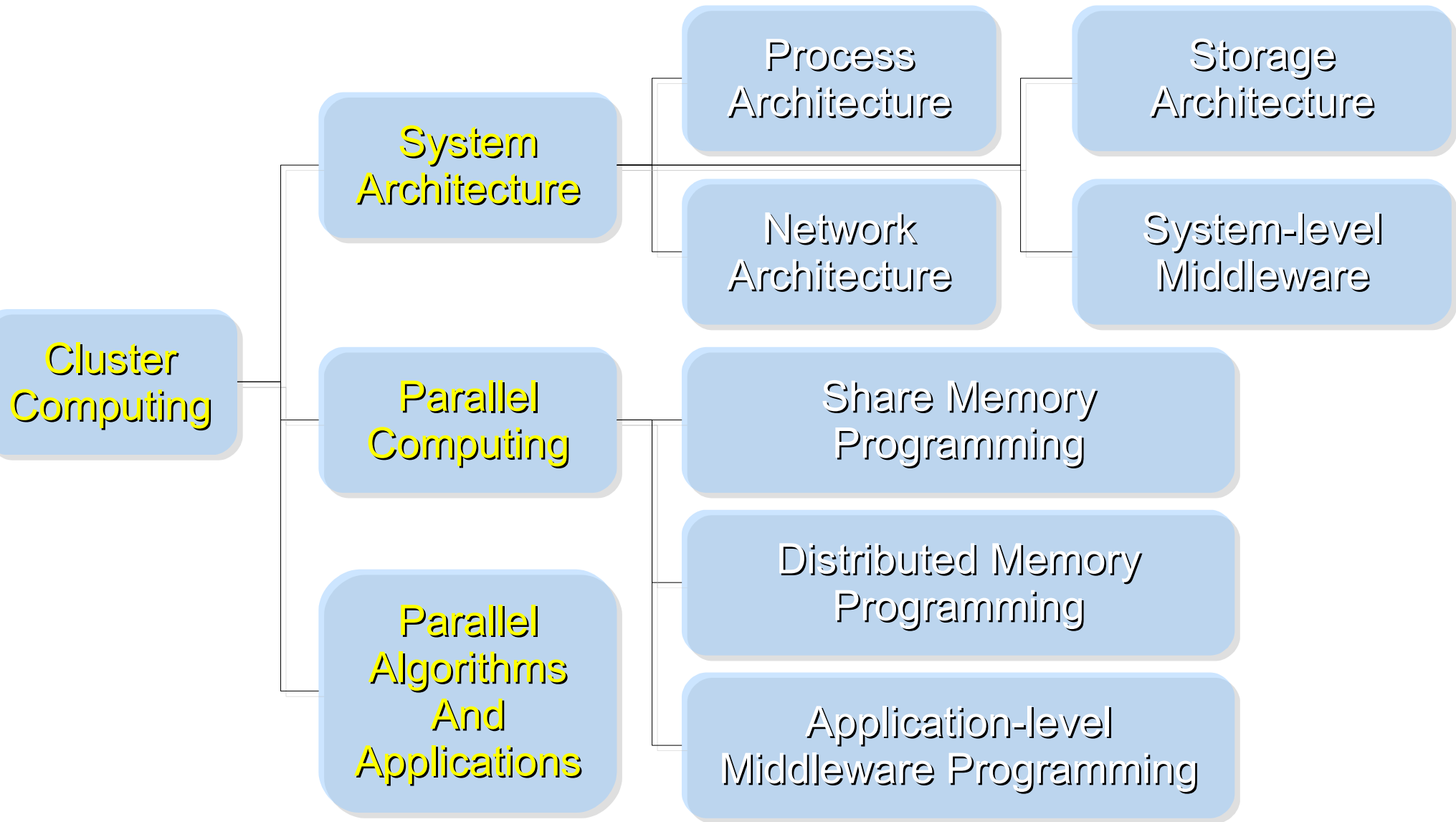
**Linux Kernel**

**Boot Loader**





# Research topics about PC Cluster



# **Challenges of Cluster Computing**

- **Hardware**

- **Ethernet Speed | PC Density**
- **Power | Cooling | Heat**
- **Network and Storage Architecture**

- **Software**

- **Job Scheduler ( Cluster level )**
- **Account Management**
- **File Sharing | Package Management**

- **Limitation**

- **Shared Memory**
- **Global Memory Management**

# ***Common Method to deploy Cluster***



**1. Setup one  
Template  
machine**

**2. Cloning  
to  
multiple  
machine**



**3. Configure  
Settings**



**4. Install  
Job  
Scheduler**



**5. Running  
Benchmark**

# **Challenges of Common Method**

**Add New User Account ?**

**Upgrade Software ?**

**How to share user data ?**

**Configuration Synchronization**

# How to deploy 4000+ Nodes ????

資料標題：Scaling Hadoop to 4000 nodes at Yahoo!

資料日期：September 30, 2008

<b>Total Nodes</b>	<b>4000</b>
<b>Total cores</b>	<b>30000</b>
<b>Data</b>	<b>16PB</b>

	<b>500-node cluster</b>		<b>4000-node cluster</b>	
	<b>write</b>	<b>read</b>	<b>write</b>	<b>read</b>
<b>number of files</b>	990	990	14,000	14,000
<b>file size (MB)</b>	320	320	360	360
<b>total MB processes</b>	316,800	316,800	5,040,000	5,040,000
<b>tasks per node</b>	2	2	4	4
<b>avg. throughput (MB/s)</b>	<b>5.8</b>	<b>18</b>	<b>40</b>	<b>66</b>

# ***Advanced Methods to deploy Cluster***

- ***SSI ( Single System Image )***
  - ***Multiple PCs as Single Computing Resources***
  - ***Image-based***
    - ***homogeneous***
    - ***ex. SystemImager, OSCAR, Kadeploy***
  - ***Package-based***
    - ***heterogeneous***
    - ***easy update and modify packages***
    - ***ex. FAI, DRBL***
- ***Other deploy tools***
  - ***Rocks : RPM only***
  - ***cfengine : configuration engine***

# ***Comparison of Cluster Deploy Tools***

	<i><b>Distribution</b></i>	<i><b>Support Diskless/ Sysmless</b></i>	<i><b>Type</b></i>	<i><b>Node configuration tools</b></i>	<i><b>Cluster management tools</b></i>	<i><b>Database installation</b></i>
<i><b>System Imager</b></i>	<i><b>ALL</b></i>	<i><b>Yes</b></i>	<i><b>Image</b></i>	<i><b>Yes</b></i>	<i><b>No</b></i>	<i><b>No</b></i>
<i><b>OSCAR</b></i>	<i><b>RPM- based</b></i>	<i><b>Yes</b></i>	<i><b>Image</b></i>	<i><b>Yes</b></i>	<i><b>Yes</b></i>	<i><b>No</b></i>
<i><b>Kadeploy</b></i>	<i><b>ALL</b></i>	<i><b>No</b></i>	<i><b>Image</b></i>	<i><b>Yes</b></i>	<i><b>Yes</b></i>	<i><b>Yes</b></i>
<i><b>Kadeploy</b></i>	<i><b>ALL</b></i>	<i><b>No</b></i>	<i><b>Image</b></i>	<i><b>Yes</b></i>	<i><b>Yes</b></i>	<i><b>Yes</b></i>
<i><b>FAI</b></i>	<i><b>Debian- Based</b></i>	<i><b>Yes</b></i>	<i><b>Package</b></i>	<i><b>Yes</b></i>	<i><b>No</b></i>	<i><b>No</b></i>



**PART 2-1 :**

# ***Hadoop Deployment Tool***

***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



Powered by **DRBL**





- Make Hadoop deployment *agile*
- Integrate with dynamic cluster deployments

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevel/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevel/slides/deploying_hadoop_with_smartfrog.pdf)

# SmartFrog - HPLabs' CM tool

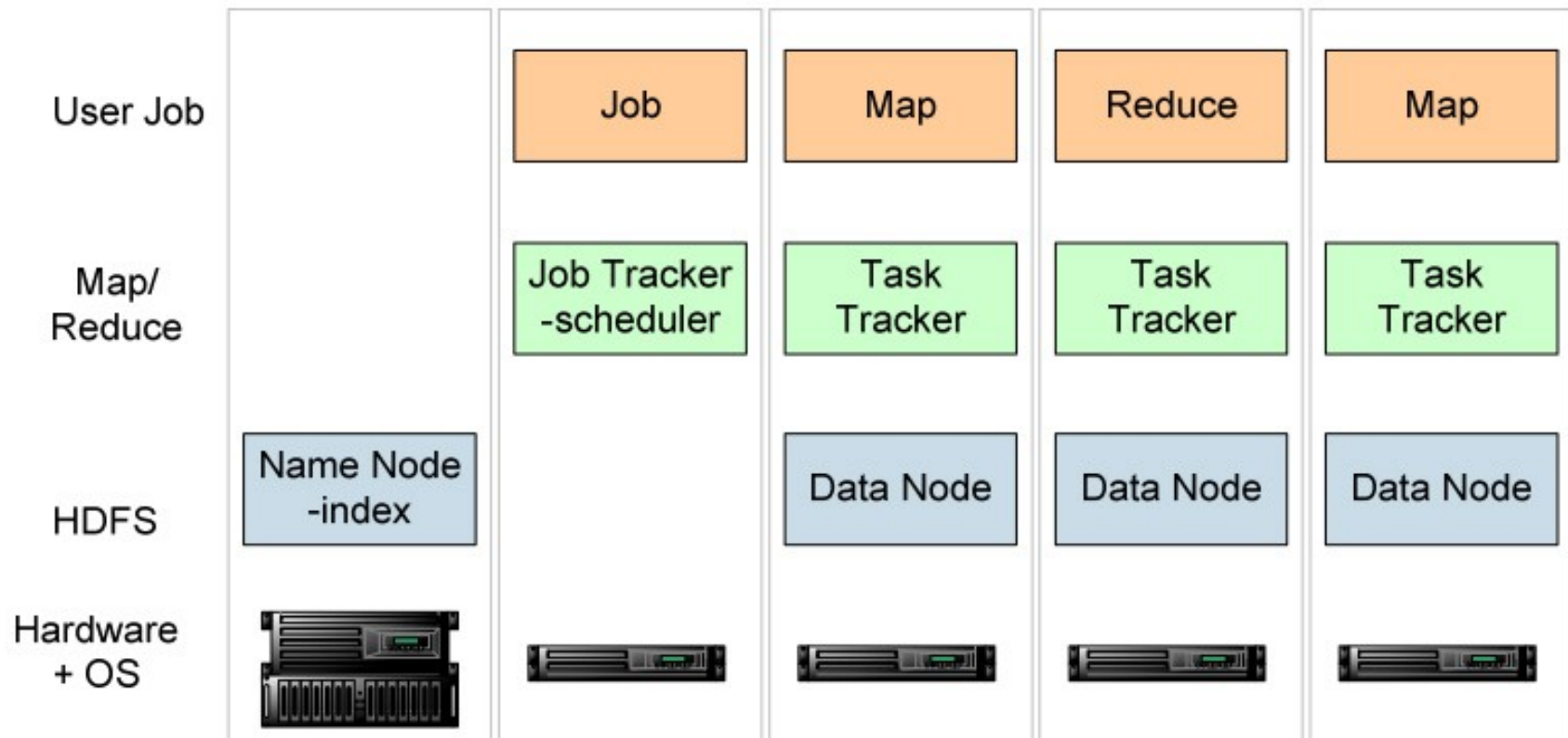
- Language for describing systems to deploy  
—everything from datacentres to test cases
  - Runtime to create *components* from the model
  - Components have a lifecycle
  - LGPL Licensed, Java 5+
- <http://smartfrog.org/>

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevel/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevel/slides/deploying_hadoop_with_smartfrog.pdf)



# Basic problem: deploying Hadoop



*one namenode, 1+ Job Tracker, many data nodes and task trackers*

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevel/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevel/slides/deploying_hadoop_with_smartfrog.pdf)

# The hand-managed cluster

- Manual install onto machines
- SCP/FTP in Hadoop zip
- copy out hadoop-site.xml and other files
- edit /etc/hosts, /etc/rc5.d, SSH keys ...
- Installation scales  $O(N)$
- Maintenance, debugging scales worse

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)



# The locked-down cluster

- PXE Preboot of OS images
- RedHat Kickstart to serve up (see [instalinux.com](http://instalinux.com))
- Maybe: LDAP to manage state, or custom RPMs

## Requires:

uniform images, central LDAP service, good ops team, stable configurations, home-rolled RPMs

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevel/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevel/slides/deploying_hadoop_with_smartfrog.pdf)



# CM-tool managed cluster

## Configuration Management tools

- State Driven: observe system state, push it back into the desired state
- Workflow: apply a sequence of operations to change a machine's state
- Centralized: central DB in charge
- Decentralized: machines look after themselves

CM tools are the only way to manage big clusters

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevel/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevel/slides/deploying_hadoop_with_smartfrog.pdf)

12 June 2006





# Model the system in the SmartFrog language

```
TwoNodeHDFS extends OneNodeHDFS {  
  
    localDataDir2 extends TempDirwithCleanup {  
  
    }  
  
    datanode2 extends datanode {  
        dataDirectories [LAZY localDataDir2];  
        dfs.datanode.https.address "https://localhost:0";  
    }  
}
```

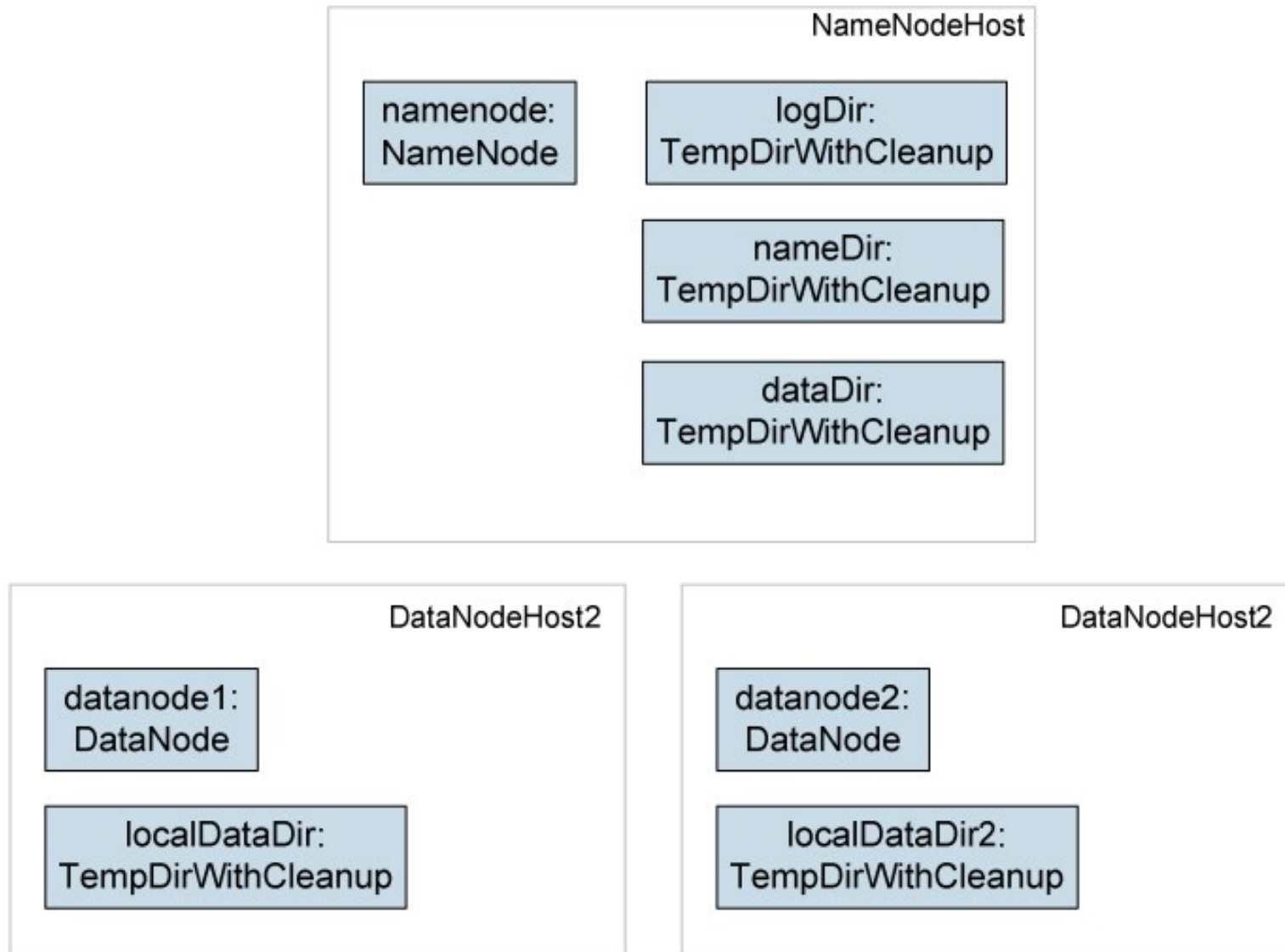
Inheritance, cross-referencing, templating

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)



# The runtime deploys the model



Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevel/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevel/slides/deploying_hadoop_with_smartfrog.pdf)



# Steps to deployability

1. Configure Hadoop from an SmartFrog description
2. Write components for the Hadoop nodes
3. Write the functional tests
4. Add *workflow* components to work with the filesystem; submit jobs
5. Get the tests to pass

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)





**PART 2-2 :**

# 企鵝龍與再生龍

***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



Powered by **DRBL**

# 何謂企鵝龍 DRBL ??

- **Diskless Remote Boot in Linux**

- 網路是便宜的，人的時間才是昂貴的。
- 企鵝龍簡單來說就是.....
  - 用網路線取代硬碟排線
  - 所有學生的電腦都透過網路连接到一台伺服器主機



**Diskfull  
PC**



=



+



+



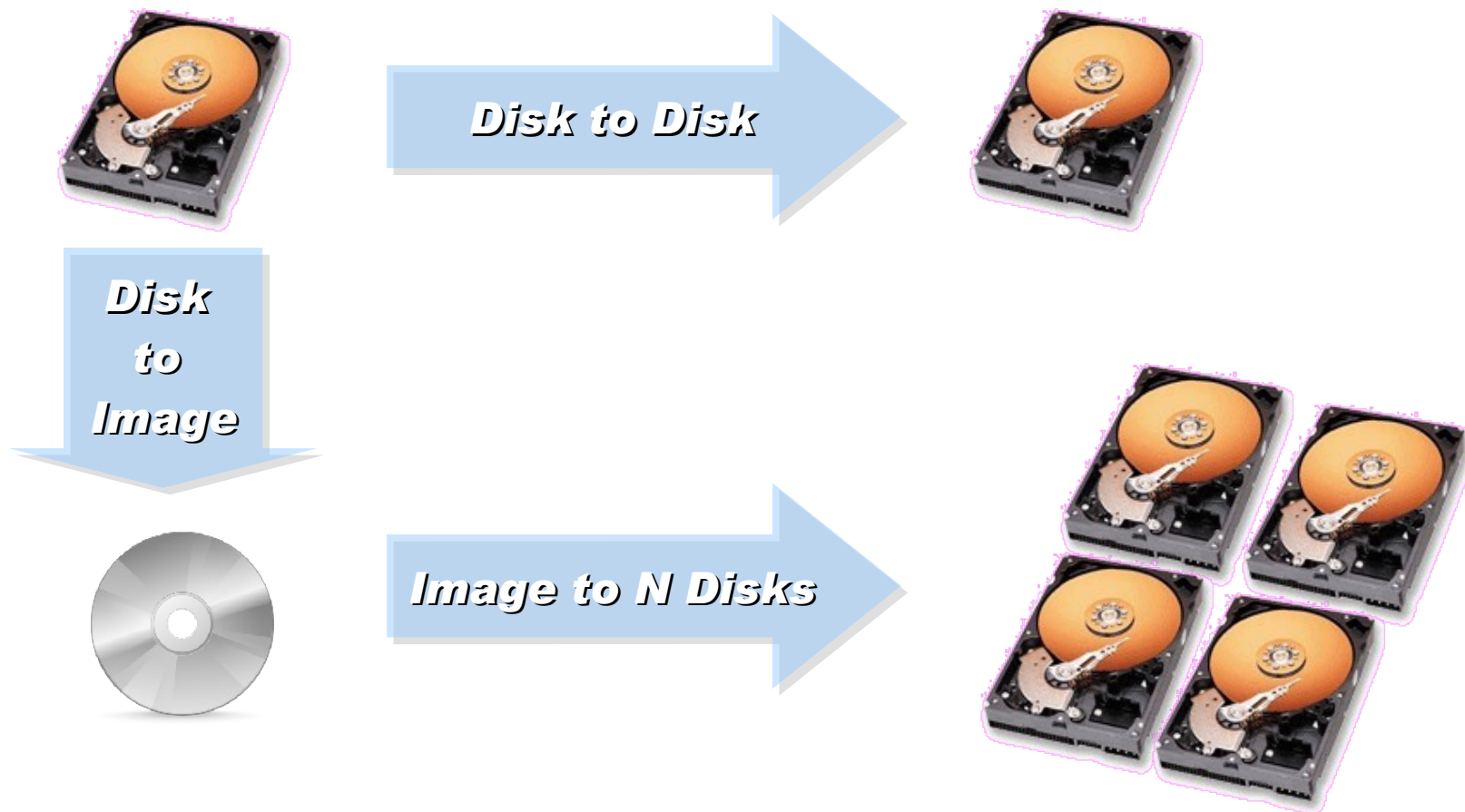
**Diskless  
PC**



Server

# 何謂再生龍 Clonezilla ??

- **Clone** (複製) + **zilla** = **Clonezilla** (再生龍)
- 裸機備分還原工具
- **Norton Ghost** 的自由軟體版替代方案



# 降低資訊教育管理成本

需要「化繁為簡」的解決方案！



一般國內小學的電腦教室

## ☑ 人力、時間成本高

教師 1 人維護管理多組設備  
教學同時分派或收集作業

## ☑ 設備維護成本高

需分別處理設定 ( 每班約 40 台 )  
如：電腦中毒、環境設定  
系統操作問題、開關機、  
備份還原等



# 平衡商業軟體與知識教育

知識和軟體都需要讓孩子「帶著走」！



## ✓ 商業軟體授權高成本

在校學習，也需回家複習  
學校每台（平均）2 萬  
學生家用（平均）4 萬

## ✓ 知識與法治的學習

教育知識，也需教育尊重  
尊重智財權觀念

# 國網中心自由軟體開發

## 多元化資訊教學的新選擇！

以個人叢集電腦 (PC Cluster) 經驗發展 DRBL&Clonezilla



### 企鵝龍 DRBL

(Diskless Remote Boot in Linux)

適合將整個電腦教室轉換  
成純自由軟體環境



### 再生龍 Clonezilla

適用完整系統備份、裸機  
還原或災難復原

## 是自由！不是免費…

分送、修改、存取、使用軟體的自由。免費是附加價值。

# 企鵝龍 DRBL 與再生龍 Clonezilla

## 電腦教室管理的新利器！

■ 以每班 40 台電腦為估算單位

DRBL&Clonezilla	未使用	使用
管理簡化	分別管理40台	管理 <b>1台</b> 伺服器
硬體設備成本	每台都需配備周邊硬體	伺服器控制，節約每台學生機之周邊硬體
軟體授權成本	40台:3000*40= <b>120,000</b> (MS Windows授權1台電腦之授權費NT\$3,000)	軟體授權 <b>NT\$0</b>
合法複製、分享	需負擔授權費	複製合法 <b>NT\$0</b>
多元化電腦教學	不同系統無法並存	<b>Linux 與MS Windows可並存</b>





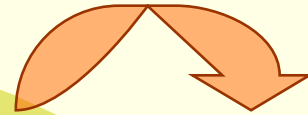
高速計算研究  
資料儲存備援

教育單位採用 DRBL

降低管理維護成本  
帶動自由軟體使用  
節樽軟體授權成本 (估計)

**NT. 98,595,000 元**

以某商業獨家軟體每機 3000 元授權費計，  
每班 35 台電腦 (3000\*35\*939)

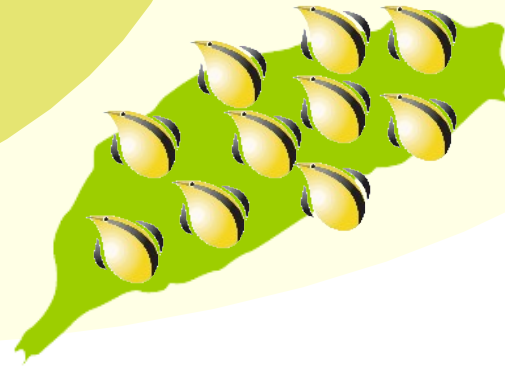


擴至全國各單位

節省龐大軟體授權費

降低台灣盜版率

提升台灣形象





**PART 1-3 :**

# 企鵝龍的開機原理

***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



Powered by **DRBL**

**1st, We install Base System of **GNU/Linux** on **Management Node**. You can choose:**

**Redhat, Fedora, CentOS, Mandriva, Ubuntu, Debian, ...**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**

*2nd, We install **DRBL package** and  
configure it as **DRBL Server**.*

*There are lots of service needed:  
**SSHD, DHCPD, TFTP, NFS Server,  
NIS Server, YP Server ...***

*Network Booting*

*Account Mgmt.*

**NFS**

**TFTP**

**DHCPD**

**SSHD**

**NIS**

**YP**

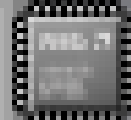
**Perl**

**Bash**

**GNU Libc**

**DRBL Server**

*based on existing  
Open Source and  
keep Hacking!*



**Kernel Module**

**Linux Kernel**

**Boot Loader**

After running **“drblsrv -i”** & **“drblpush -i”**, there will be **pxelinux**, **vmlinux-pex**, **initrd-pxe** in **TFTPROOT**, and different **configuration files** for each Compute Node in **NFSROOT**

**NFS**

**TFTPD**

**DHCPD**

**SSHD**

**NIS**

**YP**

**Config. Files**

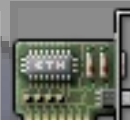
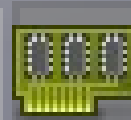
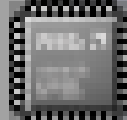
*Ex. hostname*

**initrd-pxe**

**vmlinux-pxe**

**pxelinux**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**

**3nd, We enable *PXE* function in *BIOS* configuration.**

***BIOS PXE***

***BIOS PXE***

***BIOS PXE***

***BIOS PXE***

***NFS***

***TFTPD***

***DHCPD***

***SSHD***

***NIS***

***YP***

***Config. Files***

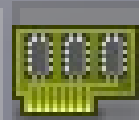
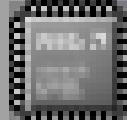
***Ex. hostname***

***initrd-pxe***

***vmlinuz-pxe***

***pxelinux***

***GNU Libc***



***Kernel Module***

***Linux Kernel***

***Boot Loader***

**While Booting, *PXE* will query IP address from *DHCPD*.**

***BIOS PXE***

***BIOS PXE***

***BIOS PXE***

***BIOS PXE***

***NFS***

***TFTPD***

***DHCPD***

***SSHD***

***NIS***

***YP***

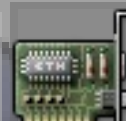
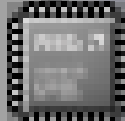
***Config. Files***  
***Ex. hostname***

***initrd-pxe***

***vmlinuz-pxe***

***pxelinux***

***GNU Libc***



***Kernel Module***

***Linux Kernel***

***Boot Loader***

**While Booting, *PXE* will query IP address from *DHCPD*.**

***IP 1***

***IP 2***

***IP 3***

***IP 4***

***NFS***

***TFTPD***

***DHCPD***

***SSHD***

***NIS***

***YP***

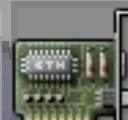
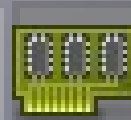
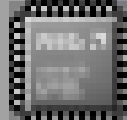
***Config. Files***  
***Ex. hostname***

***initrd-pxe***

***vmlinuz-pxe***

***pxelinux***

***GNU Libc***



***Kernel Module***

***Linux Kernel***

***Boot Loader***



**After PXE get its IP address, it will download booting files from **TFTPD**.**

**IP 1**

**IP 2**

**IP 3**

**IP 4**

**NFS**

**TFTPD**

**DHCPD**

**SSHD**

**NIS**

**YP**

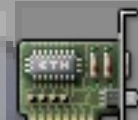
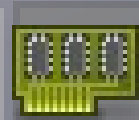
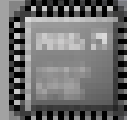
**Config. Files**  
**Ex. hostname**

**initrd-pxe**

**vmlinuz-pxe**

**pxelinux**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**



**NFS**   **TFTPD**   **DHCPD**   **SSHD**   **NIS**   **YP**

**Config. Files**  
**Ex. hostname**

**initrd-pxe**

**vmlinuz-pxe**

**pxelinux**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**

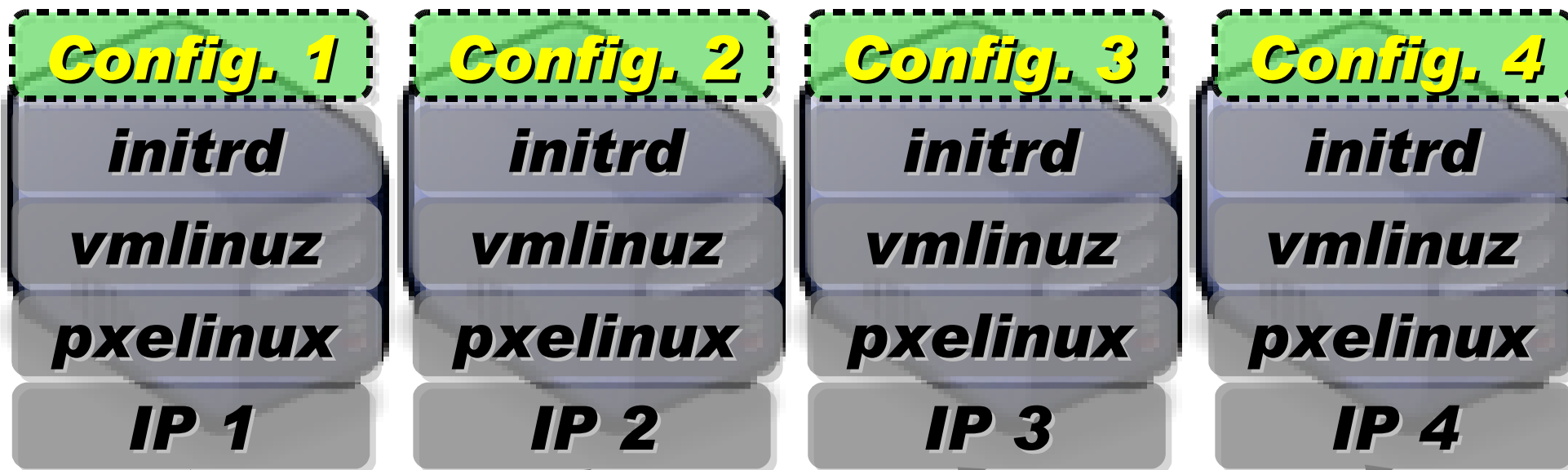


- NFS**
- TFTPD
- DHCPD
- SSHD
- NIS
- YP

Config. Files      GNU Libc

**After downloading booting files, scripts in *initrd-pxe* will config **NFSROOT** for each Compute Node.**

pxelinux      Boot Loader



**NFS**   **TFTPD**   **DHCPD**   **SSHD**   **NIS**   **YP**

**Config. Files**  
*Ex. hostname*

**initrd-pxe**

**vmlinuz-pxe**

**pxelinux**

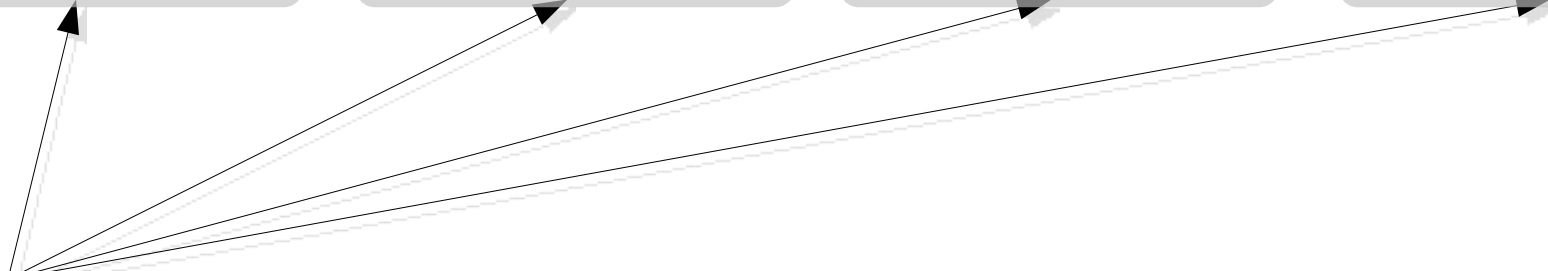
**GNU Libc**



**Kernel Module**

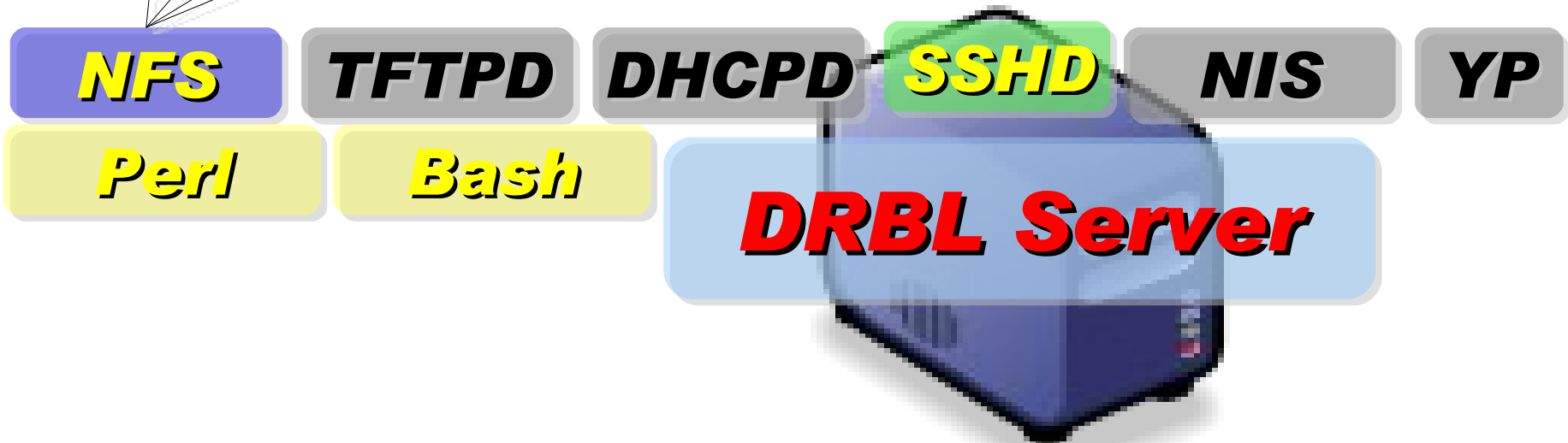
**Linux Kernel**

**Boot Loader**





**Applications and Services will also  
deployed to each Compute Node  
via **NFS** ....**





*With the help of **NIS** and **YP**,  
You can login each Compute Node  
with the **Same ID | PASSWORD**  
stored in **DRBL Server!***

**SSH Client**



**DRBL Server**

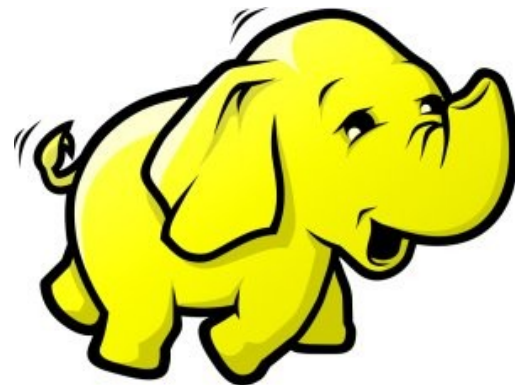




**PART 2 -1:**

# 當企鵝龍遇上小飛象

***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



Powered by **DRBL**

# 使用 DRBL 佈署 Hadoop

- 仍在開發中，待整理套件
- **drbl-hadoop** – 掛載本機硬碟給 **HDFS** 用

```
svn co http://trac.nchc.org.tw/pub/grid/drbl-hadoop
```

- **hadoop-register** – 註冊網站與 **ssh applet**

```
svn co http://trac.nchc.org.tw/pub/cloud/hadoop-register
```



root / **drbl-hadoop-0.1**

Name ▲
↑ ../
📄 drbl-hadoop
📄 drbl-hadoop-mount-disk



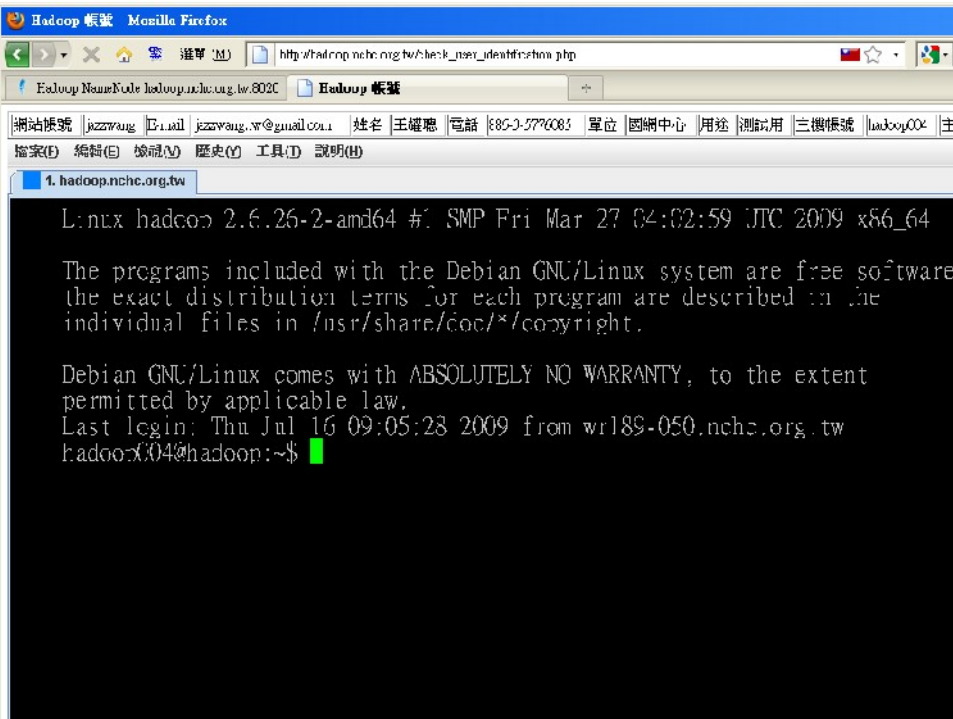
root / **hadoop-register**

Name ▲	Size	Rev	Age	Last
↑ ../				
▶ 📁 etc		<b>103</b>	4 weeks	wa
📄 adduser.php	1.3 kB	<b>85</b>	6 weeks	wa
📄 check_activate_code.php	2.2 kB	<b>85</b>	6 weeks	wa
📄 check_mount_status.php		<b>55</b>	7 weeks	wa



# 關於 [hadoop.nchc.org.tw](http://hadoop.nchc.org.tw)

- **DRBL Server - 1 台 (hadoop)** , 加大 **/home** 與 **/tftpboot** 空間。
- **DRBL Client - 19 台 (hadoop101~hadoop119)**
- 使用 **Cloudera** 的 **Debian** 套件
- 使用 **drbl-hadoop** 的設定跟 **init.d script** 來協助部署
- 使用 **hadoop-register** 來提供使用者註冊與 **ssh applet** 介面



```
L:ux hadcoo 2.6.26-2-amd64 # SMP Fri Mar 27 04:02:59 UTC 2009 x86_64

The programs included with the Debian GNU/Linux system are free software
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 16 09:05:23 2009 from wr189-050.nchc.org.tw
hadoop:x04@hadoop:~$
```



### hadoop Hadoop Map/Reduce Administration

State: RUNNING|  
Started: Sun Jul 19 22:48:19 EDT 2009  
Version: 0.18.3-4cloudera0.3.0, r  
Compiled: Fri May 29 23:29:49 UTC 2009 by root  
Identifier: 200907192248

#### Cluster Summary

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task
0	0	711	19	38	38

#### Running Jobs

Running Jobs

# Lesson Learn

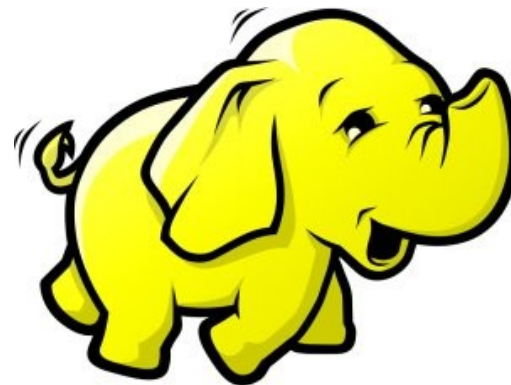
- **Cloudera** 套件的好處：使用 **init.d script** 來啟動關閉
  - **name node, data node, job tracker, task tracker**
- 建立大量帳號：
  - 可透過 **DRBL** 內建指令完成 **/opt/drbl/sbin/drbl-useradd**
- 使用者預設 **HDFS** 家目錄
  - 跑迴圈切換使用者，下 **hadoop fs -mkdir tmp**
- 設定使用者 **HDFS** 權限
  - 跑迴圈切換使用者，下 **hadoop dfs -chown \$(id) /usr/\$(id)**
- **HDFS** 會使用 **/var/lib/hadoop/cache/hadoop/dfs**
- **MapReduce** 會使用 **/var/lib/hadoop/cache/hadoop/mapred**



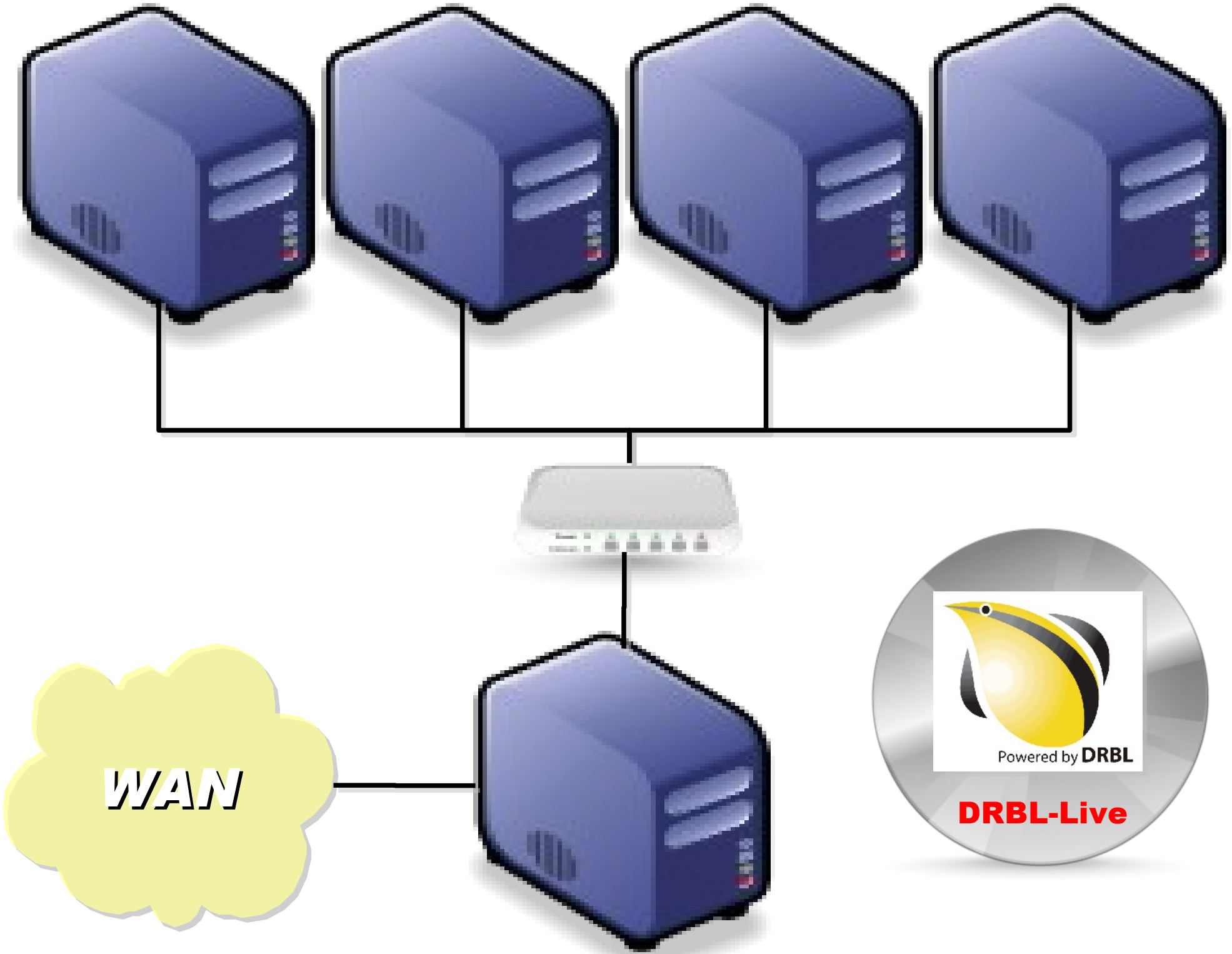
**PART 2 -2:**

# ***Live Demo***

***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



Powered by **DRBL**



# ***Demo with DRBL-Live CD***

***1. Boot Server with DRBL-Live CD***

***<http://free.nchc.org.tw/drbl-live/stable/>***

***2. Download DRBL-Hadoop Script***

***<http://classcloud.org/drbl-hadoop-live.sh>***

***<http://classcloud.org/drbl-hadoop-live-run.sh>***

***3. Follow the steps***

***<http://classcloud.org/drbl-hadoop>***



***Questions?***

***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



Powered by **DRBL**



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

---

# 結語

王耀聰 陳威宇

[Jazz@nchc.org.tw](mailto:Jazz@nchc.org.tw)

[waue@nchc.org.tw](mailto:waue@nchc.org.tw)

國家高速網路與計算中心 (NCHC)

# Overview

- 重點回顧
- 其他專案介紹
- 文獻參考

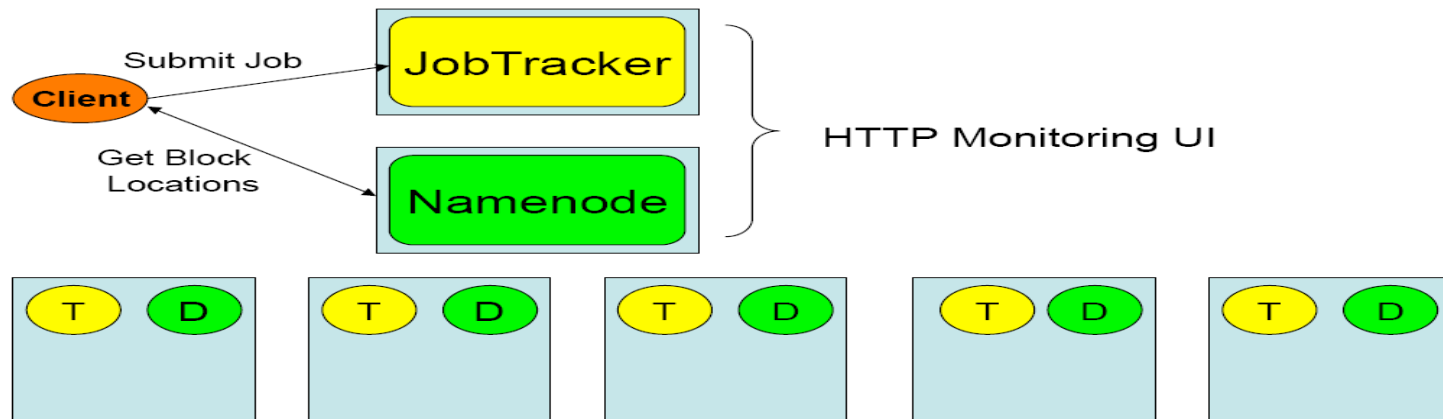


# 雲端運算

- 什麼是雲端運算？
  - 將使用者的命令透過介面，交由網路上眾多伺服器所組成的龐大系統運算之後，再把處理結果回傳給用戶
  - 雲 = ~ 網路
- 網路運算的演化
  - 叢集 -> 平行分散 -> 格網 -> 雲端
- 服務型態
  - SaaS, Paas, Iaas
- 特色
  - 經濟、簡單、可擴充…

# Hadoop

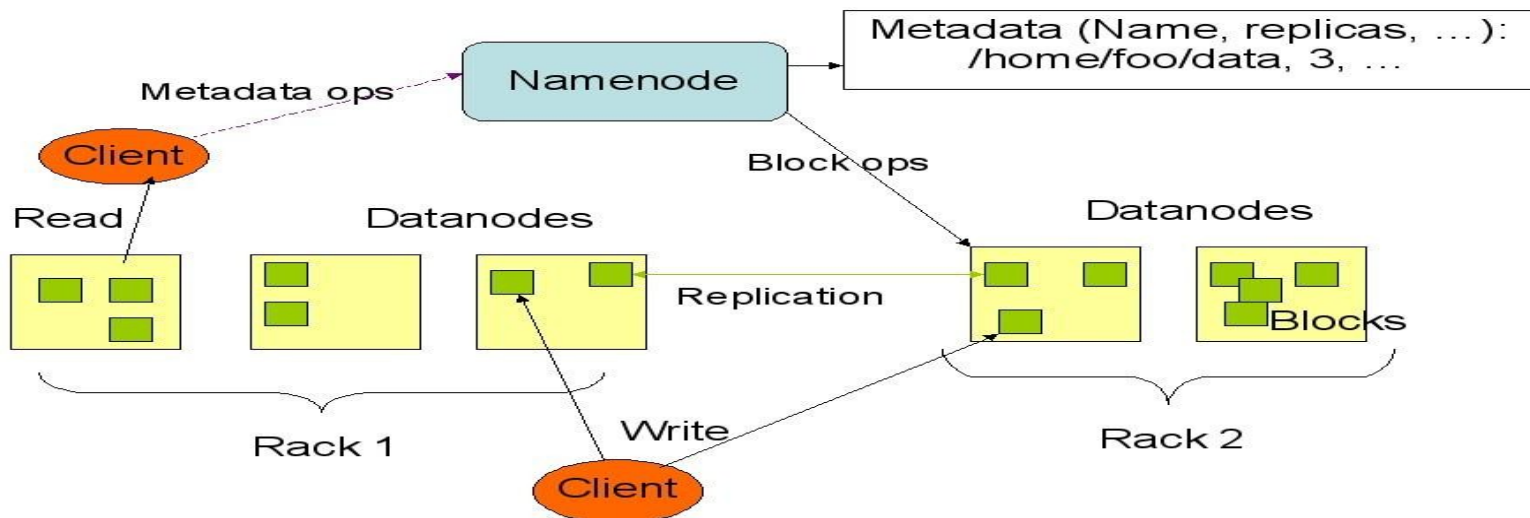
- 什麼是 Hadoop ?
  - 借鑑了 google 在分佈式系統上設計的分散式計算平台
  - 由 java 實做的自由軟體
  - 包含了 HDFS、MapReduce
- 特色
  - 巨量、經濟、高效率、可靠、持續更新



# Hadoop Distributed File System

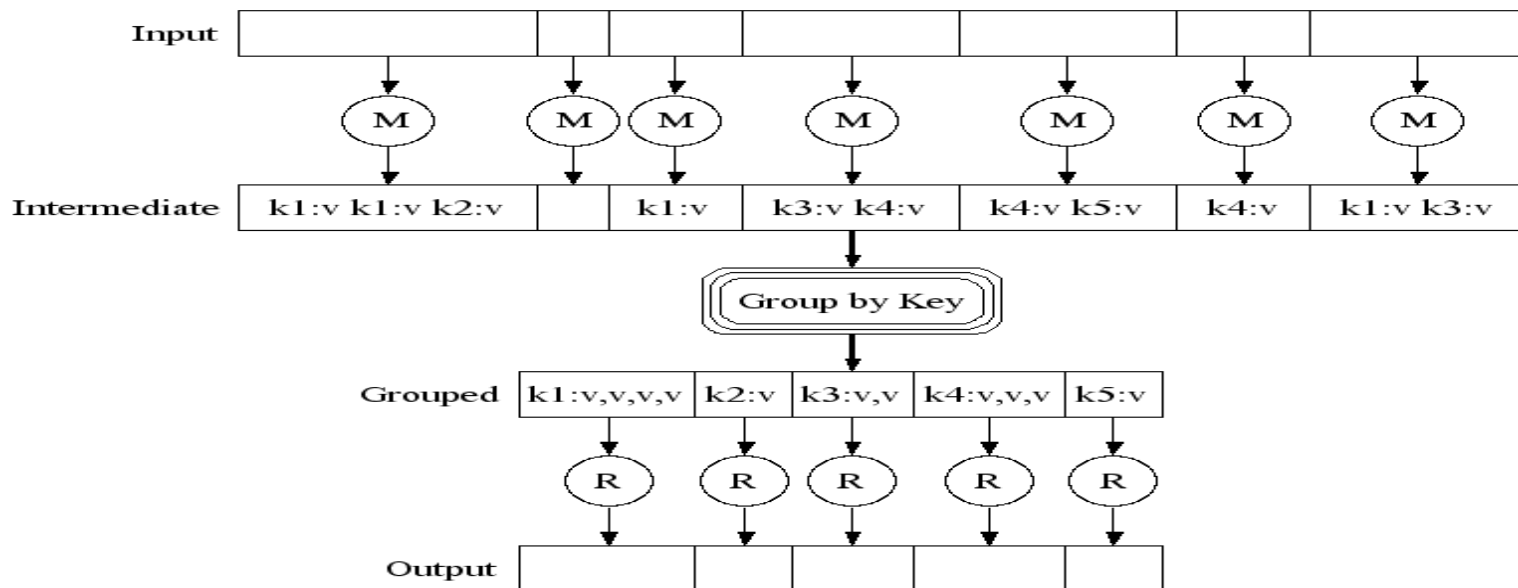
- 什麼是 HDFS ?
  - Hadoop 專案中的檔案系統
- 特色
  - 錯誤容忍、**高 Throughput**、大規模資料集、一次多次讀、在地運算、異質平台移植

HDFS Architecture



# MapReduce

- 什麼是 MapReduce
  - Map 將每個資料視為一個 key，並作  $\langle \text{key}, \text{value} \rangle$  的配對，Reduce 再統合所有的 Map 結果做出  $\langle \text{key}, \text{list}(\text{value}) \rangle$
- 運作方法



# Hadoop 安裝設定

安裝

- step 1. 設定登入免密碼
- step 2. 安裝 java
- step 3. 下載安裝 Hadoop

Java-6-sun

Hadoop-core

設定

- step 4.1 設定 hadoop-env.sh
  - export JAVA\_HOME=/usr/lib/jvm/java-6-sun
- step 4.2 設定 hadoop-site.xml
  - 設定 Namenode-> hdfs://x.x.x.1:9000
  - 設定 Jobtracker -> x.x.x.2:9001
- step 4.3 設定 slaves 檔
- step 4.4 將叢集內的電腦 Hadoop 都做一樣的配置

hadoop-env.sh

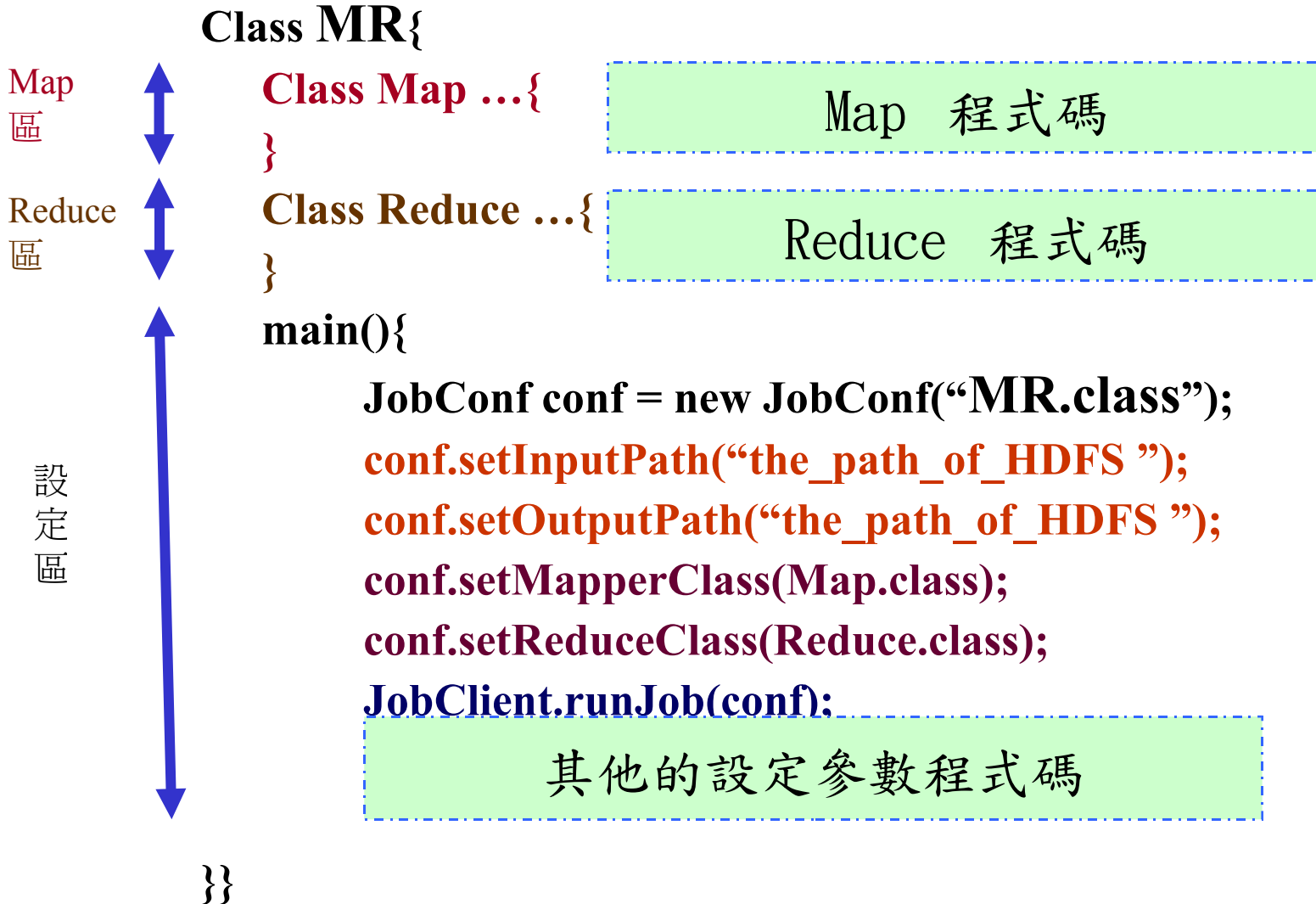
hadoop-site.xml

slaves

執行

- step 5.1 格式化 HDFS
  - bin/hadoop namenode -format
- step 5.2 啟動 Hadoop
  - nodeN 執行： bin/start-dfs.sh ; nodeJ 執行： bin/start-mapred.sh
- step 6. 完成！檢查運作狀態
  - Job admin <http://x.x.x.2:50030/> HDFS <http://x.x.x.1:50070/>

# 程式基本寫法



# 編譯程式基本步驟

## 1. 編譯

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

## 1. 封裝

- `jar -cvf MyJar.jar -C MyJava .`

## 1. 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
  - Myjar.jar = 封裝後的編譯檔

- 先放些文件檔到 HDFS 上的 input 目錄

- ./input; ./output = hdfs 的輸入、輸出目錄

# 其他相關專案

- HBase ( <http://hadoop.apache.org/hbase/> )



- 用 Hadoop 為基礎的雲端資料庫

- Nutch ( <http://lucene.apache.org/nutch/> )



- 以 Hadoop 為基礎的搜尋引擎

- Pig ( <http://hadoop.apache.org/pig/> )

- 一個可用在 Hadoop 上的平台，提供一個全新語言 (Pig Latin) 以簡化撰寫分析的程式



- Disco ( <http://discoproject.org/> )

- Nokia 所研發的 MapReduce 架構，用 erlang 實現 massive data - minimal code 者可以 python 驅動，類似 Hadoop 的自由軟體專案。

disco

massive data - minimal code



# 文獻參考

- Hadoop 官方網站
  - <http://hadoop.apache.org/core/>
- Hadoop API
  - <http://hadoop.apache.org/core/docs/r0.18.3/api/index.html>
- Hadoop Taiwan User Group
  - <http://www.hadoop.tw/>
- 中文 Hadoop 手冊
  - <http://cn.hadoop.org/doc/index.html>
- 維基百科
  - <http://en.wikipedia.org/wiki/Hadoop>
  - <http://zh.wikipedia.org/wiki/Hadoop>

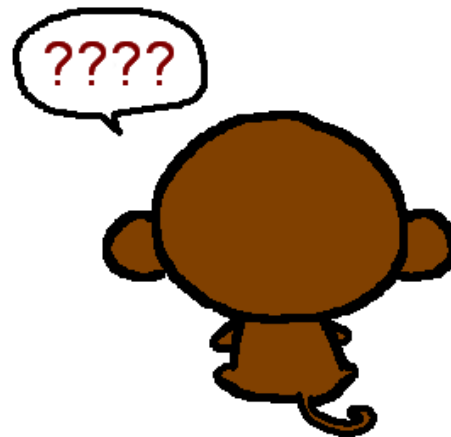


財團法人國家實驗研究院

國家高速網路與計算中心

NATIONAL CENTER FOR HIGH-PERFORMANCE COMPUTING

# Question ?



[http://chinese.storylands.org/1329magic\\_al/story1c15.php](http://chinese.storylands.org/1329magic_al/story1c15.php)



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

---

**Thank You !**



<http://miumiu516.pixnet.net/album/photo/94262410>