

# Course Information 課程資訊



- 講師介紹：
  - 國網中心 王耀聰 副研究員 / 交大電控碩士
  - [jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)
- 所有投影片、參考資料與操作步驟均在網路上
  - 由於雲端資訊變動太快，愛護地球，請減少不必要之講義列印。
- 礙於缺乏實機操作環境，故以影片展示與單機操作為主
  - 若有興趣實機操作，請參考國網中心雲端運算課程錄影
  - <http://trac.nchc.org.tw/cloud>
  - <http://www.classcloud.org/media>
  - <http://www.screentoaster.com/user?username=jazzwang>
- 若需要實驗環境，可至國網中心雲端運算實驗叢集申請帳號
  - <http://hadoop.nchc.org.tw>
- Hadoop 相關問題討論：
  - <http://forum.hadoop.tw>



# 雲端運算的關鍵技術概述

Part 1 : Overview Core Technologies of Cloud Computing

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



Powered by DRBL



什麼是雲端運算啊？可以個簡單的定義嗎？

**What is Cloud Computing ?**

雲端運算怎麼聽起來要買一些新硬體、新軟體啊？

**Is it about buying NEW Hardware and Software?**



雲端運算可能只是拿來振興經濟的幌子吧？

**Is it a trap to another bubble economy ?**



我聽你們在那裡講五四三.....

**Cloud Computing is as simple as 5..4..3..2..1...**



# National Definition of Cloud Computing

## 美國國家標準局 NIST 給雲端運算所下的定義

### 5 Characteristics

五大基礎特徵

### 4 Deployment Models

四個佈署模型

### 3 Service Models

三個服務模式

#### 1. On-demand self-service.

隨需自助服務

#### 2. Broad network access

隨時隨地用任何網路裝置存取

#### 3. Resource pooling

多人共享資源池

#### 4. Rapid elasticity

快速重新佈署靈活度

#### 5. Measured Service

可被監控與量測的服務

# 4 Deployment Models of Cloud Computing

## 雲端運算的四種佈署模型

Public Cloud  
公用雲端



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

**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

Hybrid  
Cloud

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

Community Cloud  
社群雲端

**Academia** 學術為主



私有雲端  
Private Cloud

# 3 Service Models of Cloud Computing

## 雲端運算的三種服務模式

### IaaS

Infrastructure as a Service

架構即服務

### PaaS

Platform as a Service

平台即服務

### SaaS

Software as a Service

軟體即服務



# 2 R&D directions : Cloud or Device

兩大研究方向：你該選「雲」還是「端」？



# One key spirit of Cloud Computing

用一句話說明雲端運算！服務才是王道！

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

Key spirit of Cloud ~

形成服務才是重點！！

Everything as a Service !!



# What are the trend of next 10 years ?

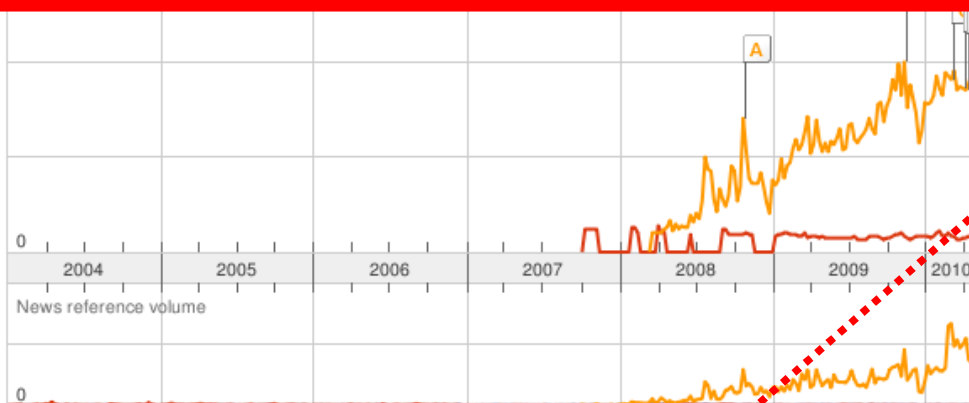
什麼是下個十年的熱門技能？

● distributed computin... ● grid computing ● cloud computing

[Sign in](#) to see and export additional Tren

All regions  All years

Search Volume index



Rank by  cloud computing

## Regions

1. [India](#)
2. [Singapore](#)
3. [South Korea](#)
4. [Hong Kong](#)
5. [Taiwan](#)
6. [Ireland](#)

## Cities

1. Bangalore, India
2. Mahape, India
3. Mumbai, India
4. Chennai, India
5. San Jose, CA, USA
6. Delhi, India

## Regions

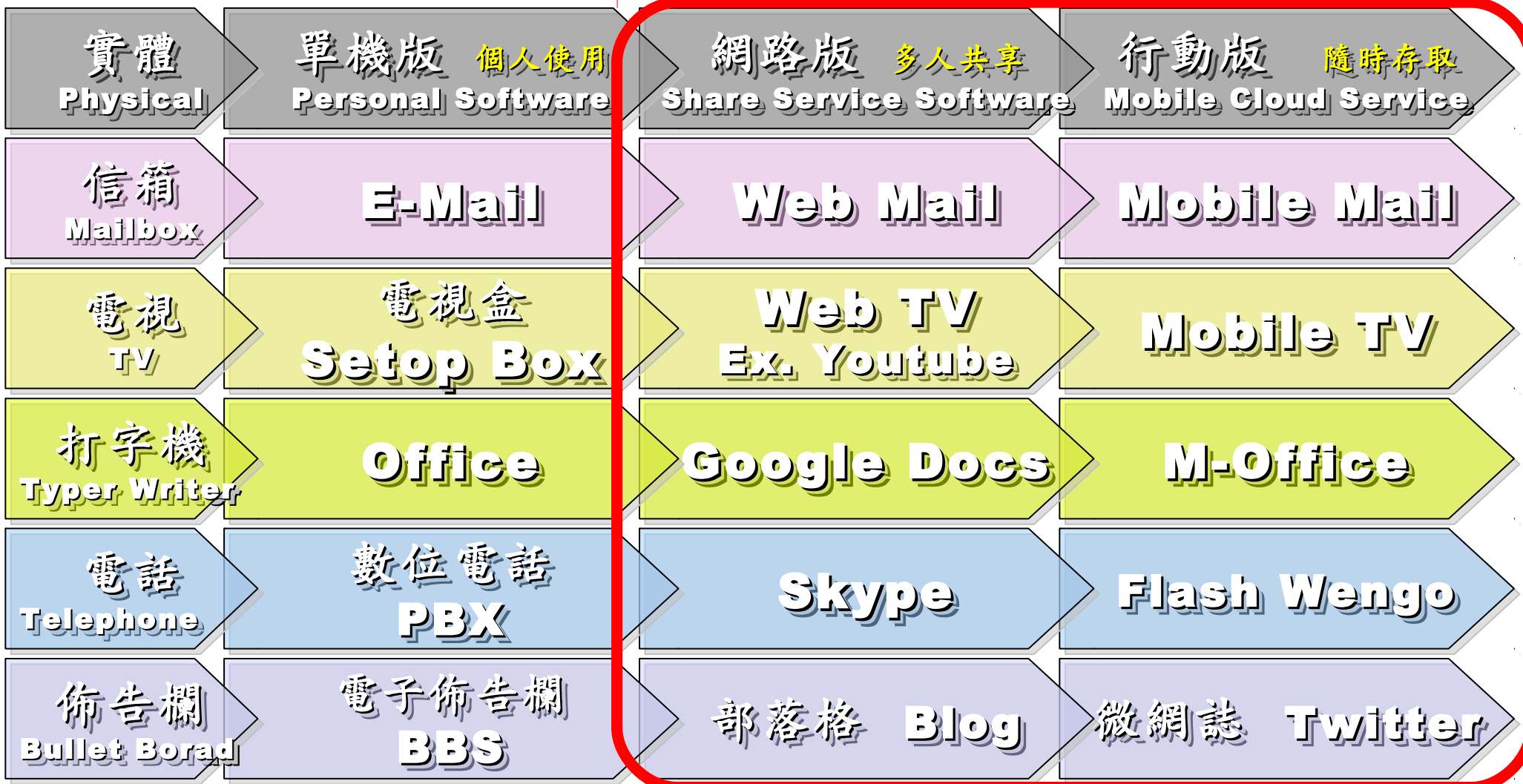
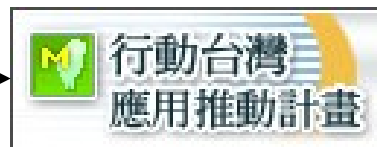
1. [India](#)
2. [Singapore](#)
3. [South Korea](#)
4. [Hong Kong](#)
5. [Taiwan](#)
6. [Ireland](#)

似乎亞洲國家特別熱愛雲端?! Too Hot in Asia ?!

# Evolution of Cloud Services

雲端服務只是軟體演化史的必然趨勢

數位化



# Key Driving Forces of Cloud Computing

## 雲端運算的關鍵驅動力

端

隨需  
行動服務

**Mobile Service**

用任何連網裝置  
都可以存取資料

租賃取代買斷  
動態隨需付費

降低  
經營成本

**Cost Down**

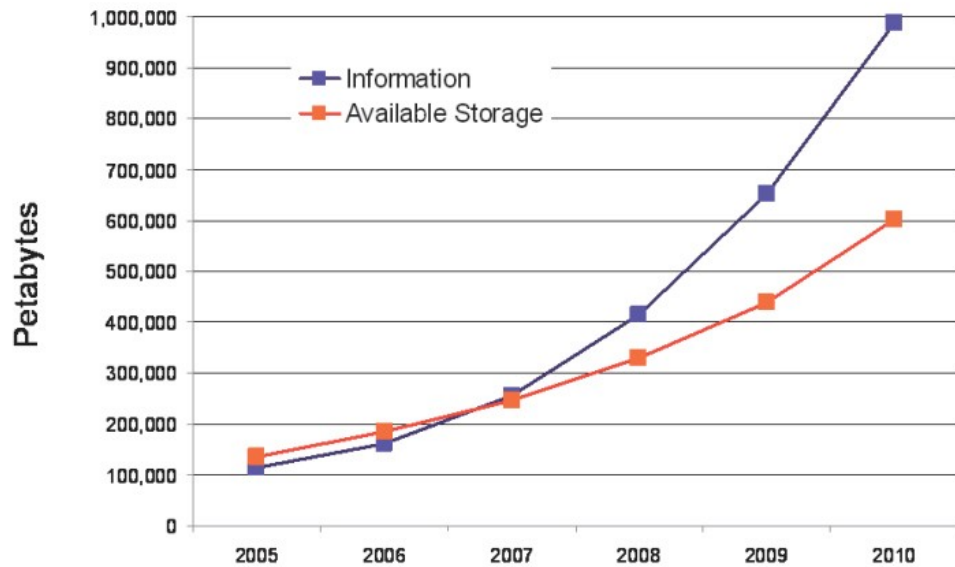
資料往雲擺  
減少資料傳輸

因應  
資料爆炸

**Data Explore**

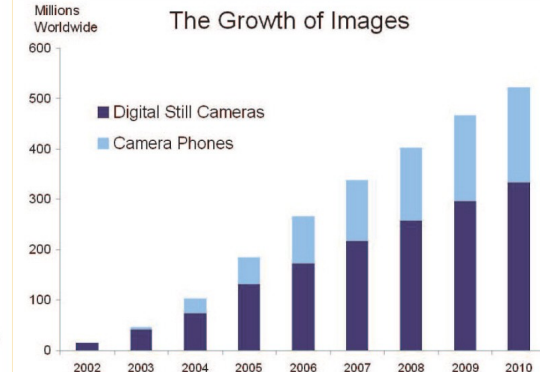
雲

# Information Versus Available Storage



# 2007 Data Explore

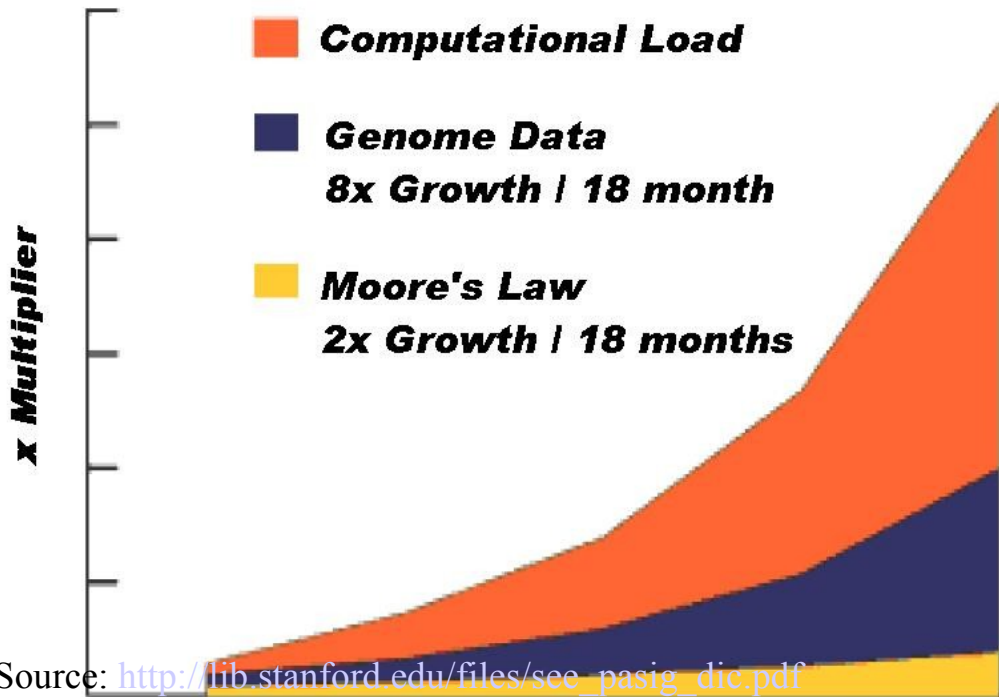
**Top 1 : Human Genomics - 7000 PB / Year**  
**Top 2 : Digital Photos - 1000 PB+ / Year**  
**Top 3 : E-mail (no Spam) - 300 PB+ / Year**



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

Source: IDC, 2007

Source: IDC, 2007



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

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)			

Phillip B. Gibbons, Data-Intensive Computing Symposium

**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

**Cost Down is the Key Drive !!**

**降低營運成本才是企業導入雲端運算的關鍵考量！！**

# How can we build Cloud Services ??

觀察雲端關鍵提供者，找尋打造雲端服務的模式

Public Cloud  
公用雲端

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



可敬的對手  
Reference Model

Hybrid  
Cloud

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

Community Cloud  
社群雲端

Academia **學術**為主



**私有雲端**  
Private Cloud



- Amazon Web Service ( AWS )
- 虛擬伺服器：**Amazon EC2**
  - Small (Default) \$0.085 per hour(L) - \$0.12 per hour(W)
  - All Data Transfer \$0.15 per GB
- 儲存服務：**Amazon S3**
  - \$0.15 per GB – first 50 TB / month of storage used
  - \$0.15 per GB – all data transfer in
  - \$0.01 per 1,000 PUT, COPY, POST, or LIST requests
- 觀念：**Paying for What You Use**

參考來源：  
<http://eblog.cisnet.org.tw/post/Cloud-Computing.aspx>  
<http://aws.amazon.com/ec2/pricing/>  
<http://aws.typepad.com/aws/2010/02/aws-data-transfer-prices-reduced.html>  
<http://aws.amazon.com/s3/#pricing>

# Amazon Web Services 亞馬遜的網頁服務

## E-Commerce

Fulfillment Web Service (FWS)

## Support

AWS Premium Support

## Database

SimpleDB, Rational DB Service

## Payment / Billing

Amazon DevPay

## Compute

EC2, Elastic MapReduce (EMR)

## Messaging

Simple Queue / Notification Service

## Network

VPC, Elastic Load Balance

## Web Traffic

Alexa Web Info / Top Sites

## Storage

S3, Elastic Block Storage (EBS)

## Content Delivery

Amazon CloudFront



# Reference Cloud Architecture

## 雲端運算的參考架構

### 應用軟體 Application

Social Computing, Enterprise, ISV,....

### 程式語言 Programming

Web 2.0 介面, Mashups, Workflows, ...

### 控制管理 Control

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

### 虛擬化 Virtualization

VM, VM management and Deployment

### 硬體設施 Hardware

Infrastructure: Computer, Storage, Network

User-Level

User-Level  
Middleware

Core  
Middleware

System Level

SaaS  
Saas  
SaaS

# Open Source to build Cloud Service

## 建構雲端服務的 自由軟體

### 應用軟體 Application

Social Computing, Enterprise, ISV, ...

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

### 程式語言 Programming

Web 2.0 介面, Mashups, Workflows, ...

Hadoop (MapReduce),  
Sector/Sphere, AppScale

### 控制管理 Control

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

OpenNebula, Enomaly,  
Eucalyptus, OpenQRM, ...

### 虛擬化 Virtualization

VM, VM management and Deployment

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

### 硬體設施 Hardware

Infrastructure: Computer, Storage,  
Network

# Building IaaS with Open Source

## 用自由軟體打造 IaaS 服務

應用軟體 Application  
Social Computing, Enterprise, ISV, ...

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

程式語言 Programming  
Web 2.0 介面, Mashups, Workflows, ...

Hadoop (MapReduce),  
Sector/Sphere, AppScale

控制管理 Control  
Qos Negotiation, Admission Control,  
Pricing, SLA Management, Metering...

OpenNebula, Enomaly,  
Eucalyptus, OpenQRM, ...

虛擬化 Virtualization  
VM, VM management and Deployment

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

硬體設施 Hardware  
Infrastructure: Computer, Storage,  
Network



# 深入解析雲端虛擬化技術

Part 2 : Deep Dive into Virtualization Technologies

**Jazz Wang**

**Yao-Tsung Wang**

**jazz@nchc.org.tw**



Powered by DRBL

# CIO 2010 : Virtualization, Cloud and Web 2.0

虛擬化技術是 2010 年 CIO 資訊長認為最重要的資訊投資

CIO strategic technologies reflect increased interest in “lighter-weight” solutions

CIO technologies

Ranking of technologies CIOs selected as one of their top 5 priorities in 2010

Ranking	2010		2009	2008	2007
Virtualization	1	↑	3	3	5
Cloud computing	2	↑	16	*	*
Web 2.0	3	↑	15	15	*
Networking, voice and data communications	4	↑	6	7	4
Business intelligence (BI)	5	↓	1	1	1
Mobile technologies	6	↑	12	12	11
Data/document management and storage	7	↑	10	9	9
Service-oriented applications and architecture	8	↑	9	10	7
Security technologies	9	↓	8	5	6
IT management	10		*	*	*
Enterprise applications	11	↓	2	2	2

\* New question for that year

Source: *Gartner Executive Programs : “ Leading in Times of Transition: The 2010 CIO Agenda ”*

# What is Virtualization ??

## 虛擬化技術有哪些呢??

Application Virtualization 應用程式虛擬化

Desktop Virtualization  
Client Virtualization 桌面虛擬化

Presentation Virtualization 顯示虛擬化

OS-level Virtualization 作業系統虛擬化

Network Virtualization 網路虛擬化

Storage Virtualization 儲存虛擬化

資料庫虛擬化

資料虛擬化

Database Virtualization

Data Virtualization

# Open Source for Virtualization

## 虛擬化技術對應的自由軟體

Application Virtualization  
應用程式虛擬化

Ex. VMWare ThinApp

Desktop Virtualization  
桌面虛擬化

Redhat SPICE

Presentation Virtualization  
顯示虛擬化

VNC, FreeNX

OS-level Virtualization  
作業系統虛擬化

Xen, KVM, OpenVZ

Network Virtualization  
網路虛擬化

OpenFlow vSwitch

Storage Virtualization  
儲存虛擬化

Lessfs, SDFS

# NIST Mapping of Cloud Technologies

## 美國國家標準局的定義主要鎖定虛擬化技術

**5. Measured Service**  
可被監控與量測的服務

Monitoring / AAA  
狀態監控與認證收費機制

**1. On-demand self-service.**  
隨需自助服務

VM Management Tool  
虛擬機器管理平台

**2. Broad network access**  
隨時隨地用任何網路裝置存取

OS-level Virtualization  
作業系統虛擬化

**3. Resource pooling**  
多人共享資源池

Network Virtualization  
網路虛擬化

**4. Rapid elasticity**  
快速重新佈署靈活度

Storage Virtualization  
儲存虛擬化



# NIST Mapping of Open Source Cloud

## 美國國家標準局的定義對應自由軟體技術

Monitoring / AAA  
狀態監控與認證收費機制

Ganglia, Nagios  
/ OpenID, SAML

VM Management Tool  
虛擬機器管理平台

OpenNebula, Eucalyptus

OS-level Virtualization  
作業系統虛擬化

Xen, KVM, OpenVZ

Network Virtualization  
網路虛擬化

OpenFlow vSwitch

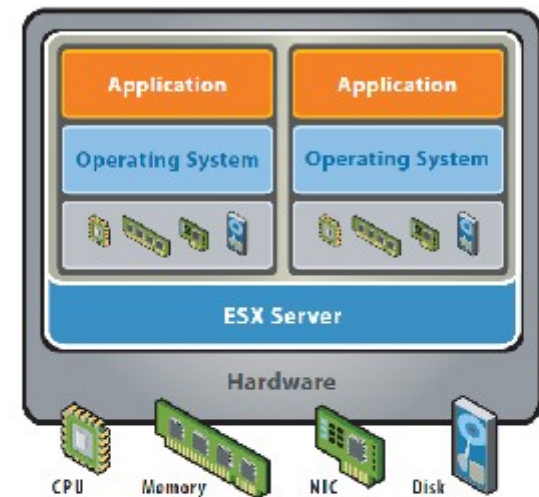
Storage Virtualization  
儲存虛擬化

Lessfs, SDFS, ZFS

# Adjust #1: Virtualization

## 企業導入雲端參考策略 (1) Virtualization

- 導入虛擬化的效益高於雲端運算
- 導入評量指標：
  - 總硬體使用率 (100% CPU 使用 ?)
  - 總電源開銷、空調、管理人力
- 商業解決方案：
  - **VMWare** ESXi / vSphere
  - **Microsoft** Hyper-V
  - **Citrix** XenServer
- 預期效益：
  - 減少伺服器採購成本與營運成本
  - (Server Consolidation)
  - 增加管理彈性與災害復原機制
  - (Ex. 異常斷電造成的服務修復)



VMware ESX Server virtualizes server storage and networking, allowing multiple applications to run in virtual machines on the same physical server.



Windows Server™ 2008  
Hyper-V™



# How to Evaluate the need of Virtualization ?

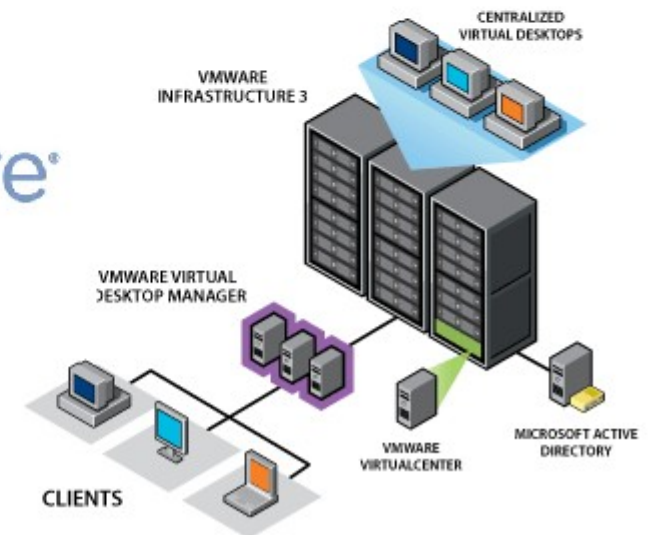
## 如何評估對虛擬化的需求??

- 關鍵驅動因素
  - 預算刪減？節約能源？提高現有硬體使用率？
- 導入評估工具
  - Microsoft Assessment and Planning (MAP) Toolkit for Hyper-V
  - Hyper-V 評估工具：收集 Server 使用情況，產生評估報告等
  - Open Source: MRTG, Ganglia, Nagios, ....
  - 蒐集 CPU 用量、記憶體用量、尖峰負載發生時間、批次排程時間
- 隱含額外成本
  - 若想獲得虛擬化帶來的好處（災害復原與動態負載平衡），需要額外建置共享的儲存設施。

# Thin Client and VDI for next IT procurement ?

## 未來改採購精簡型電腦與桌面虛擬化??

- 桌面虛擬化的需求越來越高??
- 導入評量指標：
  - 總電源開銷、空調、管理人力
  - 既存共用儲存設施
  - 資料機敏性高，不容外洩
- 商業解決方案：
  - **VMWare** Viewer 4
  - **Microsoft** Hyper-V
  - **Citrix** XenDesktop
- 預期效益：
  - 降低電力損耗 (需評估)
  - 減少資料外洩 (搭配資安政策)



# Another alternative of Virtualization

虛擬化番外篇：改用 **Terminal** + 無碟架構

用綠色自由軟體

**GREEN OSS**

打造綠色工廠

**GREEN FACTORY**

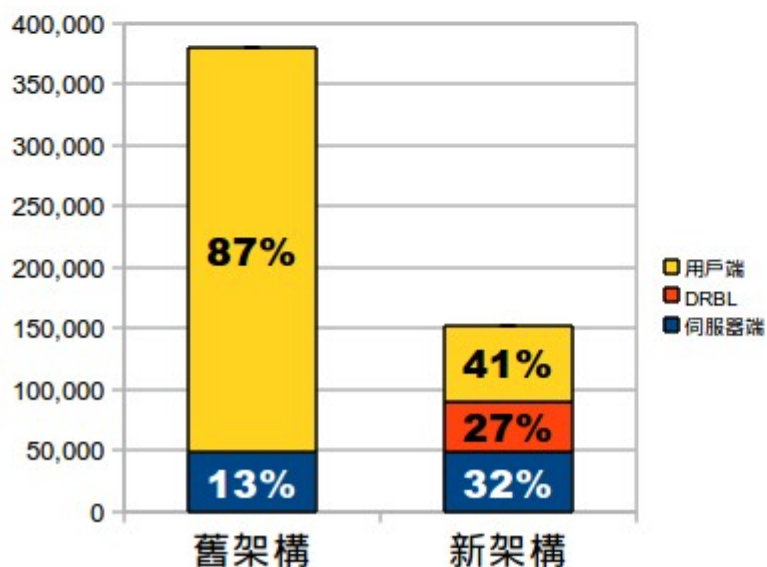
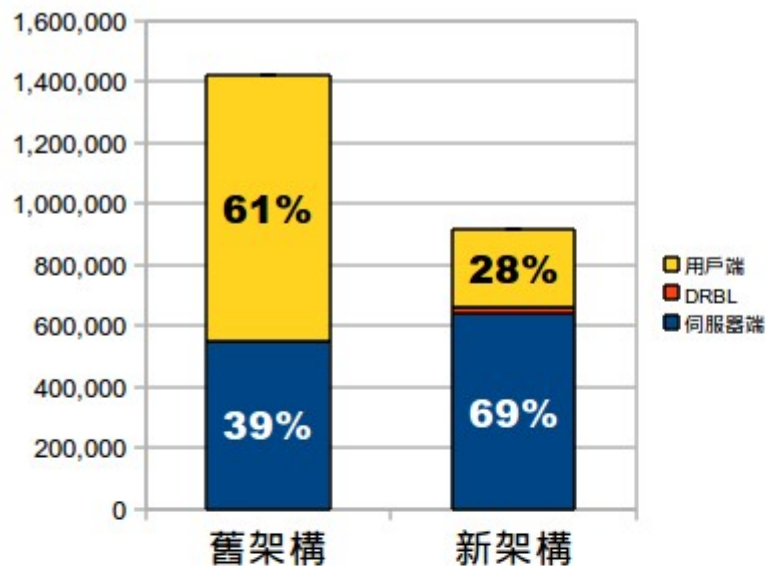
關於企鵝龍 (DRBL) 在金屬加工業的節能案例分享，請參考

[http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/09-10-14/TSMC/08-08-17\\_DRBL\\_Green\\_Computing\\_v2.pdf](http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/09-10-14/TSMC/08-08-17_DRBL_Green_Computing_v2.pdf)

# 修改理念 = 減少不必要的開支與能源消耗

- 降低硬體建置開支
  - 不用 **200W** 的桌上型電腦 (有風扇→易故障)
  - 改用 **20W** 的精簡型電腦 (無風扇→耐油氣)
  - **不裝硬碟** (少一個零件也可減少用電)
- 降低軟體授權開支
  - 購買 **Windows** 作業系統與 **Office** 授權只爲了相當簡單的報工作業並不划算
  - 改用 **Terminal 遠端桌面連線** 模式
- 降低 **MIS** 系統管理人員負擔
  - 故障時直接拿備品精簡電腦替換→**免安裝設定**

# 新舊架構成本比較 (總建置成本 / 年度營運成本)



舊架構	
伺服器端	550,000
一般用戶端	870,000
新架構	
伺服器端	638,000
DRBL	25,000
一般用戶端	255,000
合計金額	
舊架構	1,420,000
新架構	918,000
節省金額	
	502,000

舊架構	
伺服器端	48,198
一般用戶端	331,920
新架構	
伺服器端	48,198
DRBL	41,628
一般用戶端	62,442
合計金額	
舊架構	380,118
新架構	152,268
節省金額	
	227,850

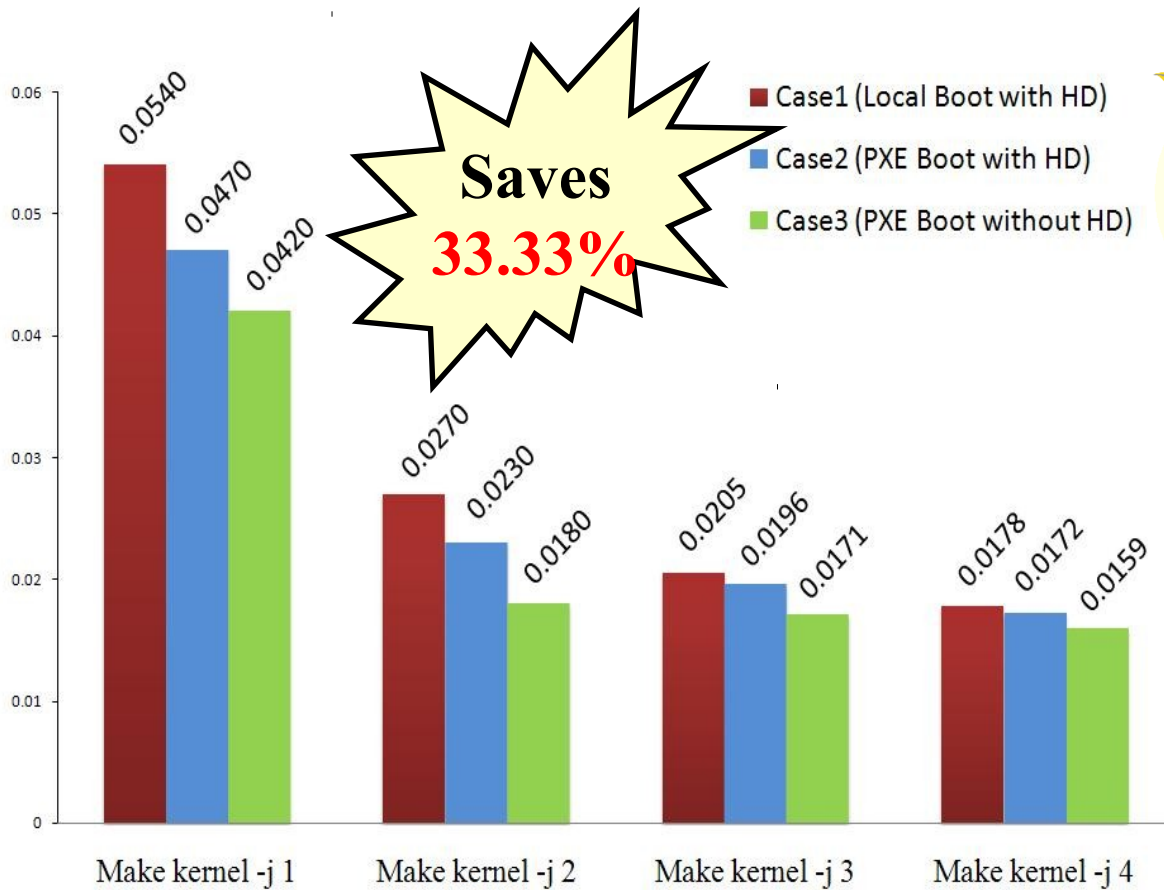
節省 **50 萬**  
總建置成本

省 **17 萬** 硬體  
省 **33 萬** 軟體

節省 **22 萬**  
年度營運成本

省 **3 萬** 電費 (60%)  
省 **19 萬** 工資 (57%)

# 若無很高的檔案讀寫需求，多善用 RAM Disk，無碟架構不僅省電最高達 33%，效能還比較好



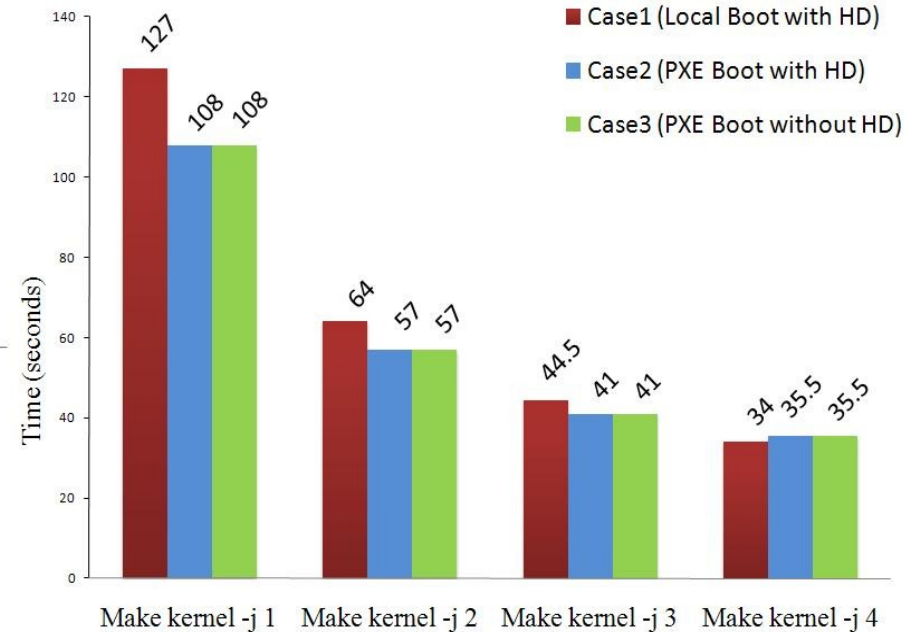
企鵝龍 **DRBL**  
(Diskless Remote Boot in Linux)

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



關於企鵝龍與再生龍的更多資訊，請參考

<http://drbl.nchc.org.tw> , <http://clonezilla.nchc.org.tw>

關於 E2CC 的更多資訊，請參考

<http://trac.nchc.org.tw/grid/raw-attachment/wiki/deliverable09/1003anav.pdf>



# Adjust #2: Storage Virtualization

## 企業導入雲端參考策略(2) 儲存虛擬化

- **資料整合為跨單位整合的第一步 !!**
- 導入評量指標：
  - 資料所有權 / 散佈權 / 重覆比例
  - 資料成長量 = ?? PB/Year
  - 現有資料副本個數 / 災害復原能力
- 商業硬體方案：EMC、NetApp
- 商業軟體方案：IBM GPFS
- 自由軟體方案：
  - Lustre、ZFS、GlusterFS...
- 預期效益：
  - **減少資料重覆 (Data Deduplication)**
  - 提供異地副本備份、**災害復原機制**



# Building IaaS with Open Source

## 用自由軟體打造 IaaS 服務

應用軟體 Application  
Social Computing, Enterprise, ISV, ...

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

程式語言 Programming  
Web 2.0 介面, Mashups, Workflows, ...

Hadoop (MapReduce),  
Sector/Sphere, AppScale

控制管理 Control  
Qos Negotiation, Admission Control,  
Pricing, SLA Management, Metering...

OpenNebula, Enomaly,  
Eucalyptus, OpenQRM, ...

虛擬化 Virtualization  
VM, VM management and Deployment

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

硬體設施 Hardware  
Infrastructure: Computer, Storage,  
Network

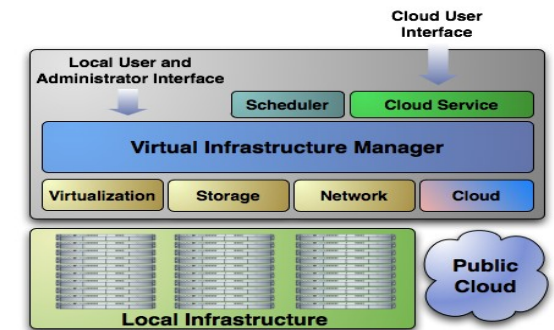
- 原是加州大學聖塔芭芭拉分校 (UCSB) 的研究專案
- **It was a research project of UCSB, USA**
- 目前已轉由 Eucalyptus System 這間公司負責維護
- **Now Eucalyptus System provide technical supports.**
- 創立目的是讓使用者可以**打造自己的 EC2**
- **It designed to help user to build their own Amazon EC2**
- 特色是相容於 Amazon EC2 既有的用戶端介面
- **Its feature is compatible with existing EC2 client.**
- 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
- **You can register trail account at <http://open.eucalyptus.com/>**
- 缺點：目前仍有部分操作需透過指令模式
- **Cons : you might need to type commands in some case**



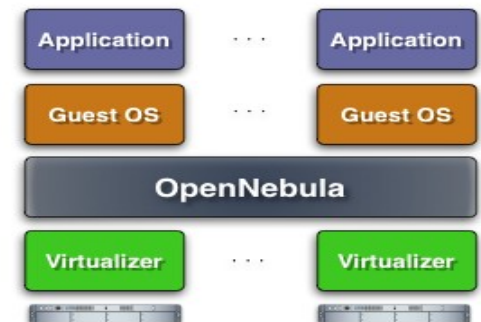
# Eucalyptus

- <http://www.opennebula.org>
- 由歐洲研究學會 (European Union FP7 ) 贊助
- **Sponsor by European Union FP7**
- 將實體叢集轉換成具管理彈性的虛擬基礎設備
- Turn Physical Cluster into Virtual Cluster
- 可管理**虛擬叢集**的狀態、排程、遷徙 (migration)
- **manage status, scheduling and migration of virtual cluster**
- [Ubuntu 9.04 provide package of opennebula](#)
- 缺點：需下指令來進行虛擬機器的遷徙 (migration) 。
- **Cons** : You need to type commands to check or migration

OpenNebula.org



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



# Virtualization for Embedded Device ?

虛擬化只用在「雲」嗎？那「端」呢？



# Motivation for Embedded Virtualization

## 嵌入式軟體採用虛擬化技術的三大動機

多核心  
Multiple  
Cores

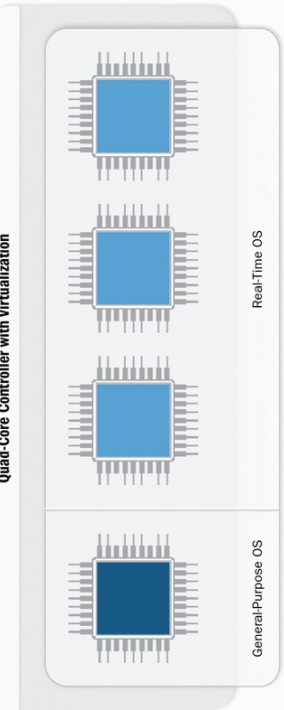
co-existence of different OS environment  
on the same platform

讓同一個硬體平台上共存不同的作業系統

isolating critical components  
from an untrusted OS environment  
在不受信任的作業系統上建立獨立的管制區

an indirection level for remote control  
of OS environments

為作業系統建立一個間接的遠端遙控機制



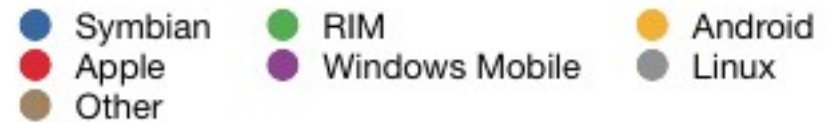
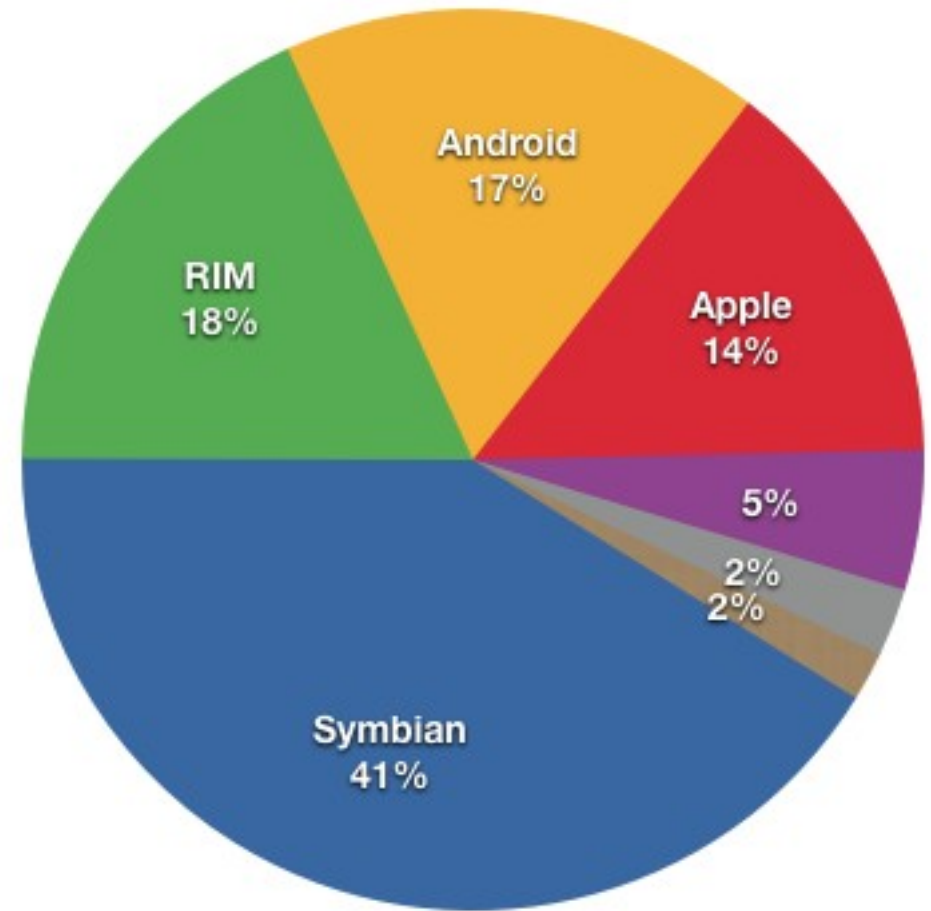
參考來源：**Embedded systems virtualization: Consider a Hypervisor**

<http://www.eetimes.com/design/automotive-design/4016811/Embedded-systems-virtualization-Consider-a-Hypervisor>

參考來源：**10 questions to ask when choosing a virtualization solution**

<http://www.eetimes.com/design/eda-design/4006406/10-questions-to-ask-when-choosing-a-virtualization-solution>

# Application : Key drivers of Multiple OS in one device 應用程式是在同一個裝置跑多作業系統的關鍵驅動力



圖片來源：[http://www.iphonic.tv/iphone\\_screen\\_amaze.png](http://www.iphonic.tv/iphone_screen_amaze.png)

圖片來源：[http://en.wikipedia.org/wiki/File:Smartphone\\_share\\_current.png](http://en.wikipedia.org/wiki/File:Smartphone_share_current.png)

參考來源：**iPhone - Playing mame4iphone with a WiiMote**

<http://www.youtube.com/watch?v=2AqgIWPnr1c>

參考來源：**Android running on iPhone!**

<http://linuxoniphone.blogspot.com/2010/04/ive-been-working-on-this-quietly-in.html>

<http://www.youtube.com/watch?v=5yO2KQHkt4A>

參考來源：**Android on Motorola Rokr E6**

<http://www.youtube.com/watch?v=-QAWd4ljV3g>

參考來源：**Dual Boot Windows Mobile and Android**

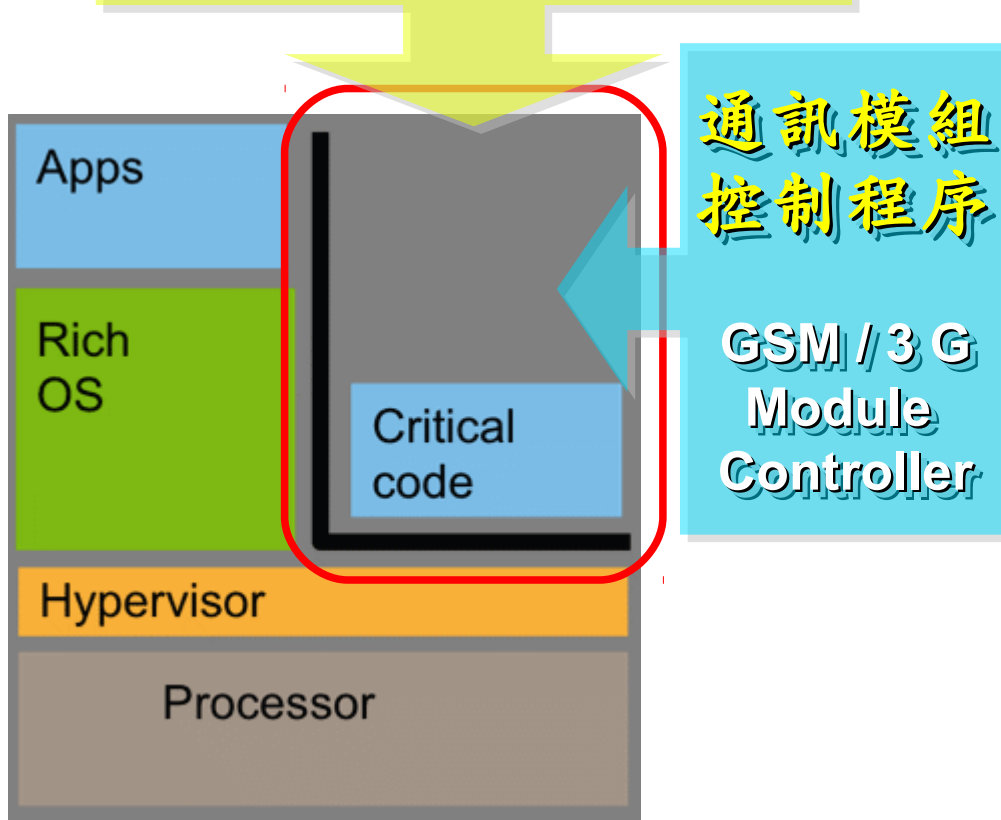
<http://www.redmondpie.com/how-to-dual-boot-windows-mobile-and-android-on-windows-phone-9140407/>

<http://www.youtube.com/watch?v=Nvj4ObHmxCI>

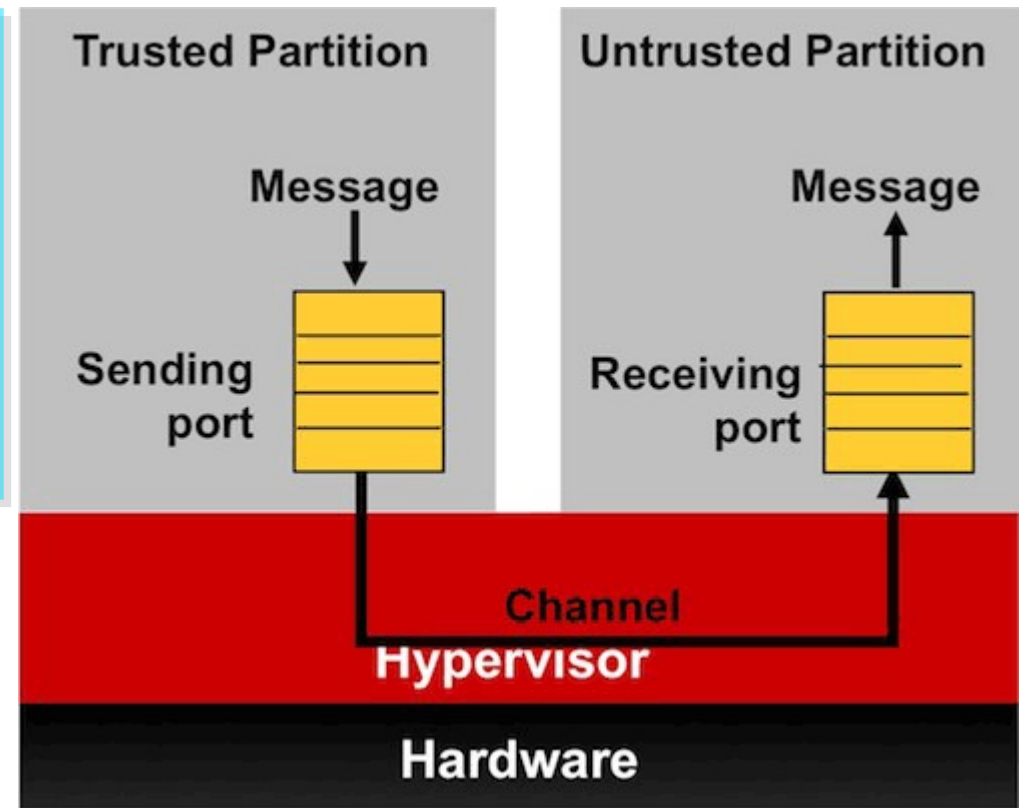
# Virtualization for Embedded Security

## 爲了安全性，必須建立獨立的管制區

DRM 音樂授權  
Digital Rights Management



區域間用訊息傳遞方式溝通  
Communicate with Message Channel



參考來源：**Embedded systems virtualization: Consider a Hypervisor**

<http://www.eetimes.com/design/automotive-design/4016811/Embedded-systems-virtualization-Consider-a-Hypervisor>

參考來源：**Securing Smart Grid Devices - Using Virtualization to Protect the Grid**

<http://embeddedinnovator.com/securing-smart-grid-devices>

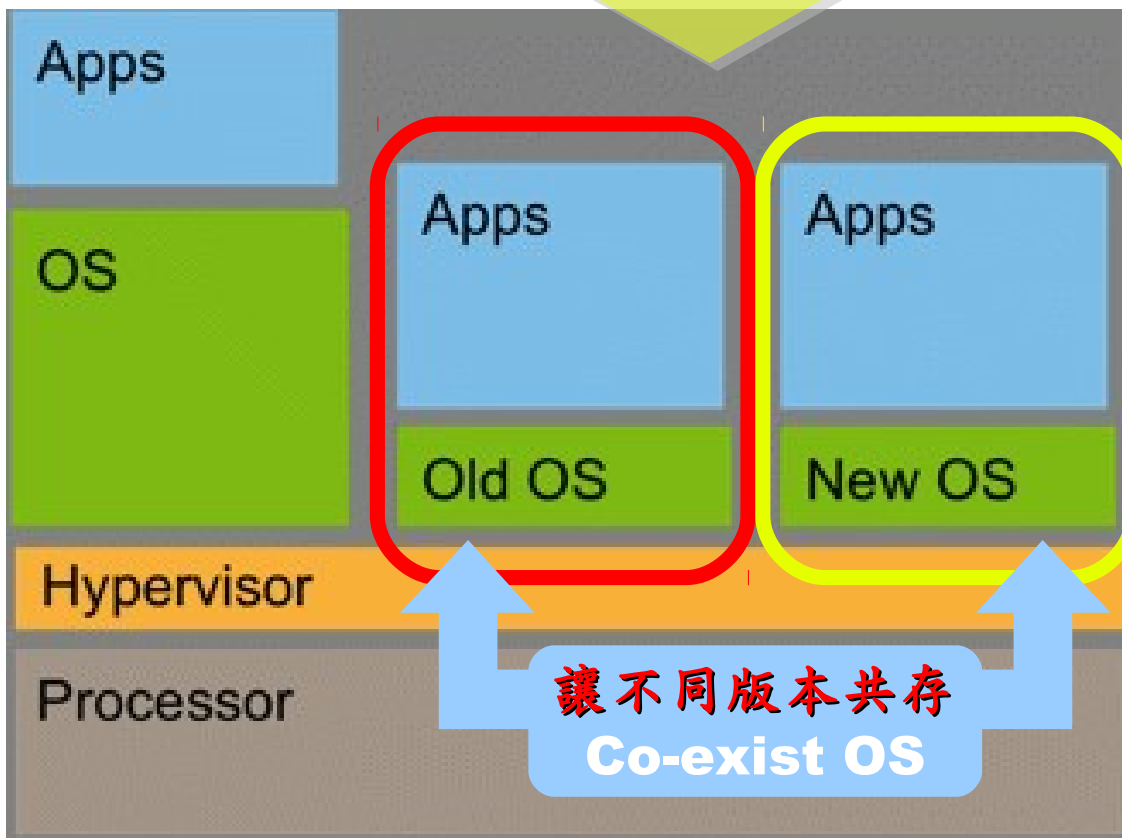


# Upgrade and Data Security : Key drivers of remote control

## 有助於作業系統升級並保護手機被偷時的資料安全

FOTA 韌體升級管制  
firmware-over-the-air

遠距資料清除  
Remote control to wipe data



參考來源：**Embedded systems virtualization: Consider a Hypervisor**

<http://www.eetimes.com/design/automotive-design/4016811/Embedded-systems-virtualization-Consider-a-Hypervisor>

# Open Source for Embedded Virtualization

## 嵌入式虛擬化技術的自由軟體

### Embedded XEN on ARM platforms

<http://embeddedxen.sf.net>

sourceforge FIND AND DEVELOP OPEN SOURCE SOFTWARE

[Find Software](#) [Develop](#) [Create Project](#) [Blog](#) [Site Support](#) [About](#)

[SourceForge.net](#) > [Find Software](#) > Embedded XEN on ARM platforms



Embedded XEN on ARM platforms Alpha by rossierd

[Summary](#) [Files](#) [Support](#) [Develop](#)

Requirement	KVM	Xen	Embedded Hypervisor
Embedded Processor Support	no	yes	yes
Real-time	no	no	yes
Fast communication	no	no	yes
Footprint	>10MB	16MB	<64kB
Code size	100s kLoC	100s kLoC	10 kLoC

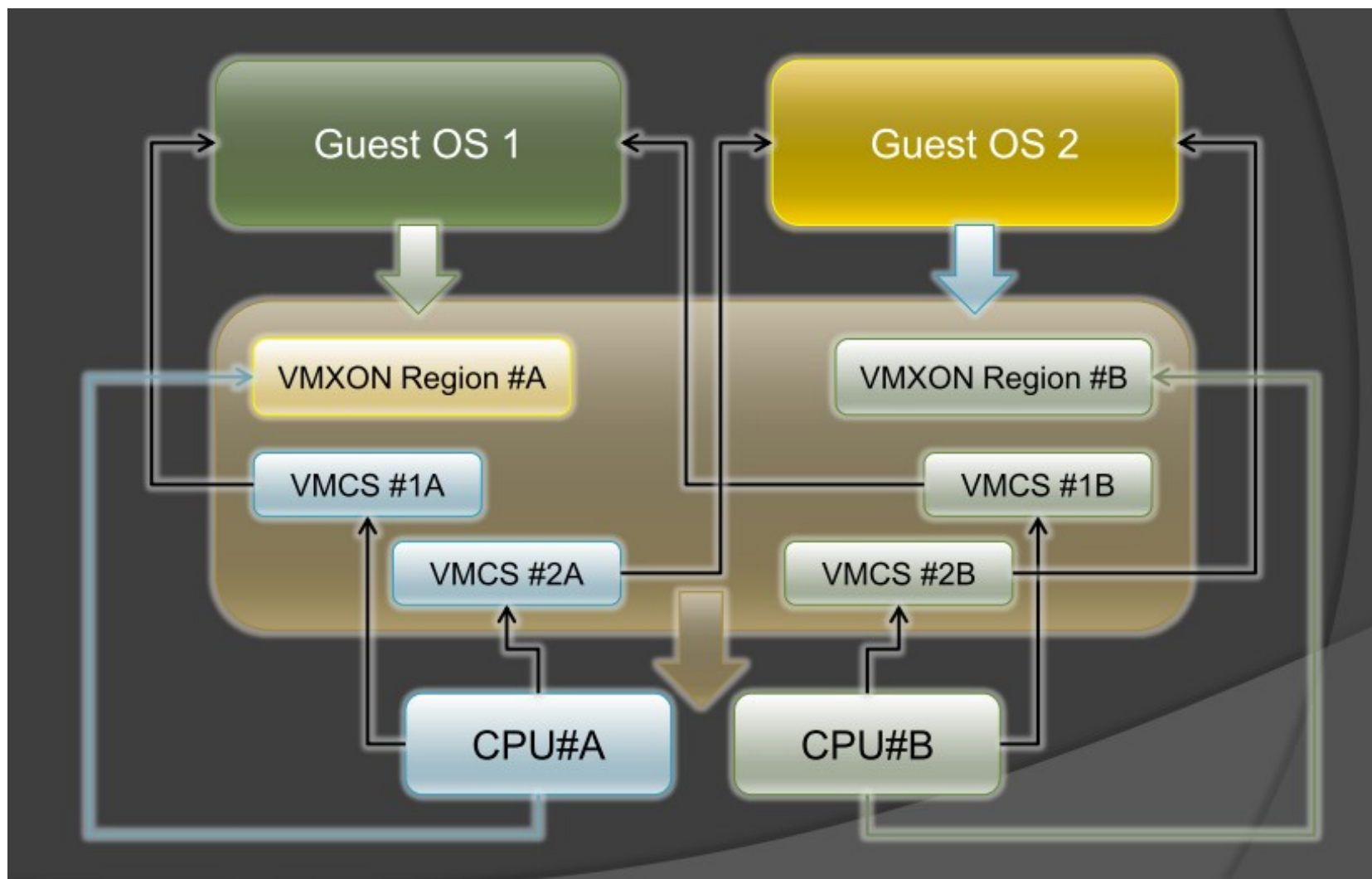
image with  
bedded

EDIT

[v project details](#)

# 虛擬化衍生的新興資安問題

透過虛擬機器，竊取鍵盤輸入、植入後門 .....



圖片來源： Hacks in Taiwan Conference 2010

[http://www.hitcon.org/hit2010/download/6\\_New%20Battlefield%20For%20Malware%20Game.pdf](http://www.hitcon.org/hit2010/download/6_New%20Battlefield%20For%20Malware%20Game.pdf)

王大寶 & PK / Hypervisor - New Battlefield For Malware Game 虛擬機 - 惡意程式攻防的新戰場



# 深入解析雲端大量資料分析技術

Part 3 : Deep Dive into Data Science Technologies

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



Powered by DRBL

- Google App Engine (GAE)
- 讓開發者可自行建立網路應用程式於 Google 平台之上。
- 提供：
  - 500MB of storage
  - up to 5 million page views a month
  - 10 applications per developer account
- 限制：
  - 程式設計語言只能用 Python 或 Java
- 計費標準：
  - 連出頻寬 \$0.12 美元/GB, 連入頻寬 \$0.10 美元/GB
  - CPU 時間 \$0.10 美元/時
  - 儲存的資料 \$0.15 美元/GB-每月
  - 電子郵件收件者 \$0.0001 美元/每個收件者



# Three Core Technology of Google ....

## Google 的三大關鍵技術 ....

- Google 在一些會議分享他們的三大關鍵技術
- Google shared their design of web-search engine
  - SOSP 2003 :
    - “The Google File System”
    - <http://labs.google.com/papers/gfs.html>
  - OSDI 2004 :
    - “MapReduce : Simplified Data Processing on Large Cluster”
    - <http://labs.google.com/papers/mapreduce.html>
  - OSDI 2006 :
    - “Bigtable: A Distributed Storage System for Structured Data”
    - <http://labs.google.com/papers/bigtable-osdi06.pdf>



# Open Source Mapping of Google Core Technologies

## Google 三大關鍵技術對應的自由軟體

### BigTable

A huge key-value datastore

HBase, Hypertable  
Cassandra, ....

### MapReduce

To parallel process data

Hadoop MapReduce API  
Sphere MapReduce API, ...

### Google File System

To store petabytes of data

Hadoop Distributed File System (HDFS)  
Sector Distributed File System

更多不同語言的 MapReduce API 實作：

<http://trac.nchc.org.tw/grid/intertrac/wiki%3Ajazz/09-04-14%23MapReduce>

其他值得觀察的分散式檔案系統：

- IBM GPFS - <http://www-03.ibm.com/systems/software/gpfs/>
- Lustre - <http://www.lustre.org/>
- Ceph - <http://ceph.newdream.net/>

# Building PaaS with Open Source

## 用自由軟體打造 PaaS 雲端服務

應用軟體 Application  
Social Computing, Enterprise, ISV, ...

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

程式語言 Programming  
Web 2.0 介面, Mashups, Workflows, ...

Hadoop (MapReduce),  
Sector/Sphere, AppScale

控制管理 Control  
Qos Negotiation, Admission Control,  
Pricing, SLA Management, Metering...

OpenNebula, Enomaly,  
Eucalyptus, OpenQRM, ...

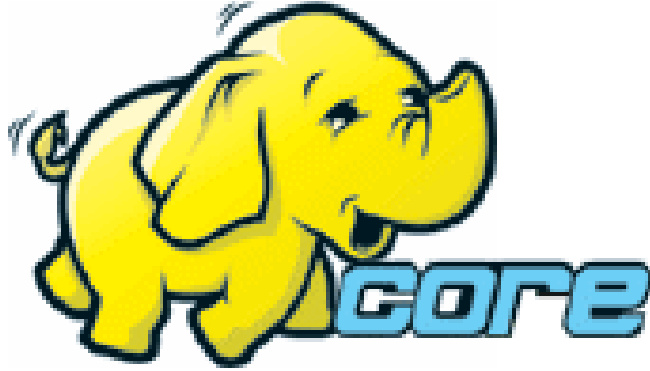
虛擬化 Virtualization  
VM, VM management and Deployment

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

硬體設施 Hardware  
Infrastructure: Computer, Storage, Network



# Hadoop

- <http://hadoop.apache.org>
  - Hadoop 是 Apache Top Level 開發專案
  - **Hadoop is Apache Top Level Project**
  - 目前主要由 Yahoo! 資助、開發與運用
  - **Major sponsor is Yahoo!**
  - 創始者是 Doug Cutting，參考 Google Filesystem
  - **Developed by Doug Cutting, Reference from Google Filesystem**
  - 以 Java 開發，提供 HDFS 與 MapReduce API。
  - **Written by Java, it provides HDFS and MapReduce API**
  - 2006 年使用在 Yahoo 內部服務中
  - **Used in Yahoo since year 2006**
  - 已佈署於上千個節點。
  - **It had been deploy to 4000+ nodes in Yahoo**
  - 處理 Petabyte 等級資料量。
  - **Design to process dataset in Petabyte**
- 
- Facebook、Last.fm  
、Joost are also  
powered by Hadoop**

# Sector / Sphere

- <http://sector.sourceforge.net/>
- 由美國資料探勘中心研發的自由軟體專案。
- **Developed by National Center for Data Mining, USA**
- 採用 C/C++ 語言撰寫，因此效能較 Hadoop 更好。
- **Written by C/C++, so performance is better than Hadoop**
- 提供「類似」Google File System 與 MapReduce 的機制
- **Provide file system similar to Google File System and MapReduce API**
- 基於UDT高效率網路協定來加速資料傳輸效率
- **Based on UDT which enhance the network performance**
- Open Cloud Testbed有提供測試環境，並開發Ma1Stone效能評比軟體
- **Open Cloud Consortium provide Open Cloud Testbed and develop Ma1Stone toolkit for benchmark**

The logo for Sector-Sphere, featuring the text "Sector-Sphere" in a bold, orange, sans-serif font. The text is set against a blue background with a white, cloudy sky pattern.

National Center for Data Mining  
University of Illinois at Chicago



Open Data Group

<http://www.opendatagroup.com/>

# What is Hadoop ?

用一句話解釋 **Hadoop** 是什麼??

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

*Hadoop* 是一個讓使用者簡易撰寫並執行處理海量資料應用程式的軟體平台。

亦可以想像成一個處理海量資料的生產線，只須學會定義 **map** 跟 **reduce** 工作站該做哪些事情。

# Two Key Elements of Operating System

## 作業系統兩大關鍵組成元素

Scheduler  
程序排程

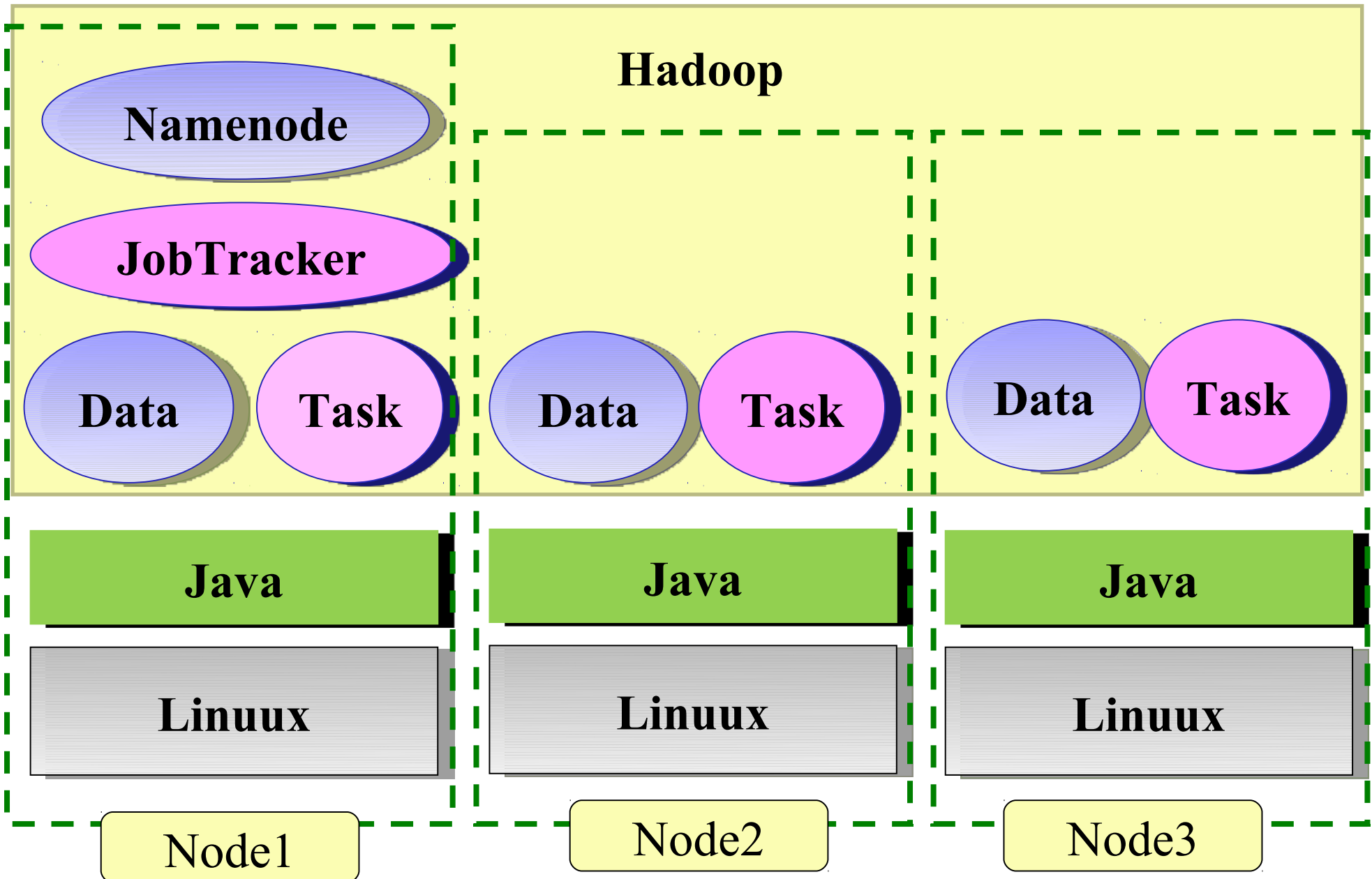


File System  
檔案系統



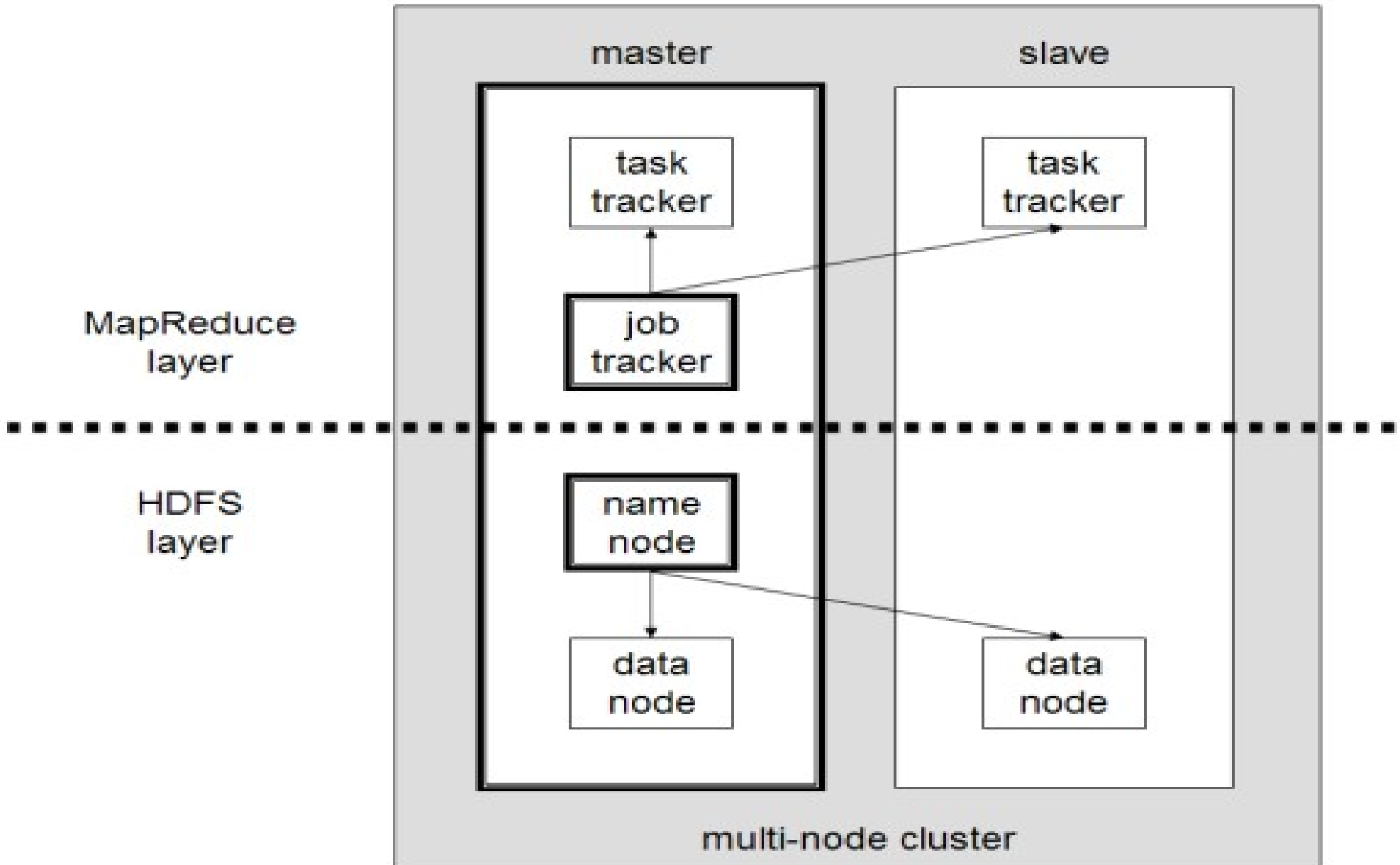
# Distributed Operating System of Hadoop

**Hadoop** 建構成一個分散式作業系統



# Different Roles of Hadoop Architecture

## *Hadoop* 軟體架構中的不同角色



# Two Key Roles of HDFS

## HDFS 軟體架構的兩種關鍵角色

### 名稱節點 **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

### 資料節點 **DataNode**

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

# Two Key Roles of Job Scheduler

## 程序排程的兩種關鍵角色

### JobTracker

- **Master Node**
- 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

### TaskTracker

- **Worker Nodes**
- 運作 Map/Reduce 的工作
- 管理儲存、回覆運算結果
- 可多個

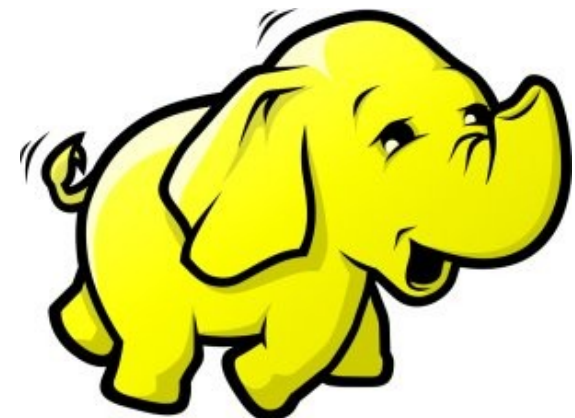




# HDFS 簡介

*Introduction to Hadoop Distributed File System*

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



# ***What is HDFS ??***

## ***什麼是 HDFS ??***

- **Hadoop Distributed File System**

- 實現類似 Google File System 分散式檔案系統
- Reference from Google File System.
- 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
- **A scalable distributed file system for large data analysis .**
- 運作於廉價的普通硬體上，又可以提供容錯功能
- **based on commodity hardware with high fault-tolerant.**
- 給大量的用戶提供總體性能較高的服務
- **It have better overall performance to serve large amount of users.**

# *Features of HDFS ...*

## **HDFS 的特色是 ...**

- **硬體錯誤容忍能力 Fault Tolerance**
  - 硬體錯誤是正常而非異常
  - Failure is the norm rather than exception
  - 自動恢復或故障排除
  - automatic recovery or report failure
- **串流式的資料存取 Streaming data access**
  - 批次處理多於用戶交互處理
  - Batch processing rather than interactive user access.
  - 高 Throughput 而非低 Latency
  - High aggregate data bandwidth (throughput)

# Features of HDFS ...

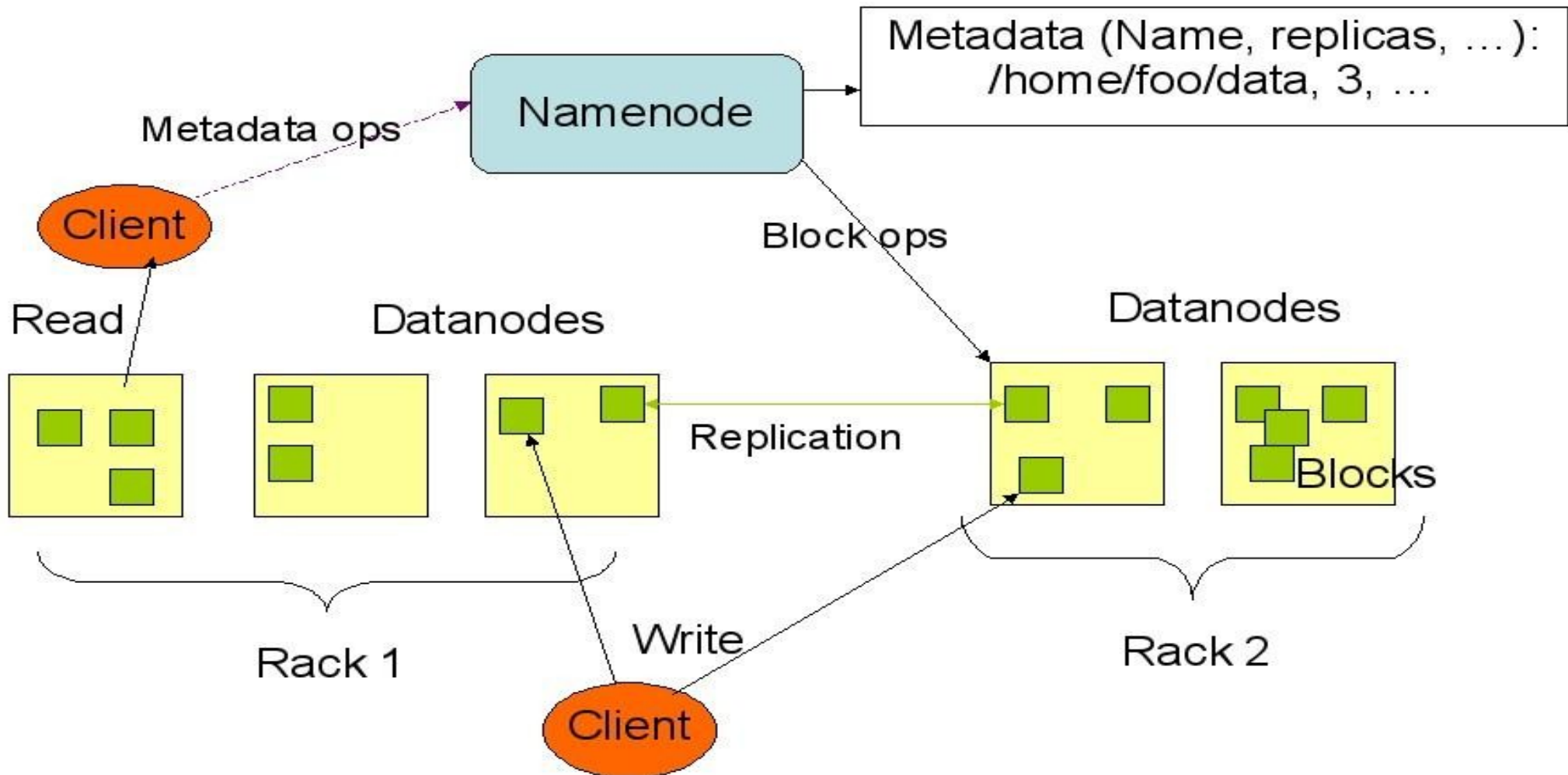
## HDFS 的特色是 ...

- **大規模資料集 Large data sets and files**
  - 支援 Petabytes 等級的磁碟空間
  - Support Petabytes size
- **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
- **在地運算 Data Locality**
  - 到資料的節點上計算 > 將資料從遠端複製過來計算
  - “move compute to data” > “move data to compute”
- **異質平台移植性 Heterogeneous**
  - 即使硬體不同也可移植、擴充
  - HDFS could be deployed on different hardware

# How HDFS manage data ...

## HDFS 如何管理資料 ...

### HDFS Architecture



# How does HDFS work ...

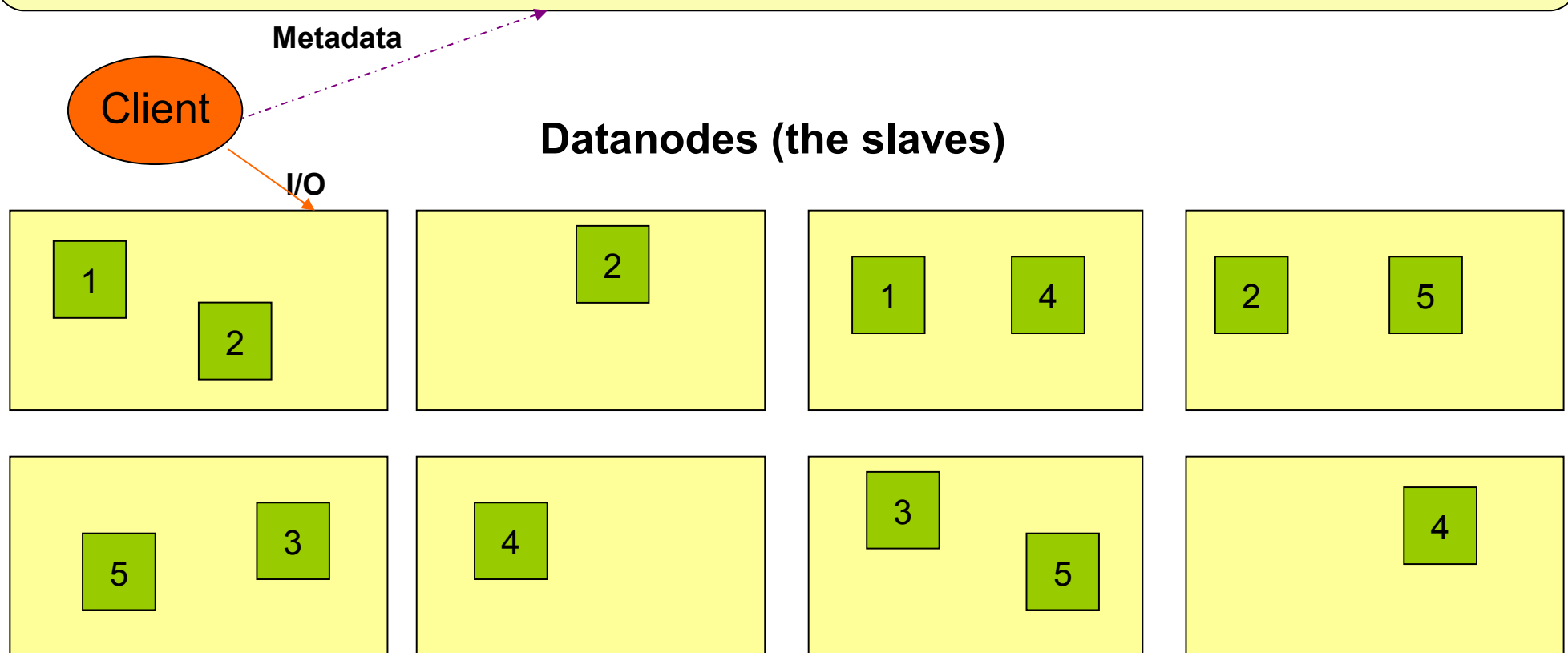
## HDFS 如何運作 ...

Namenode (the master)

Path and Filename – Replication , blocks

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

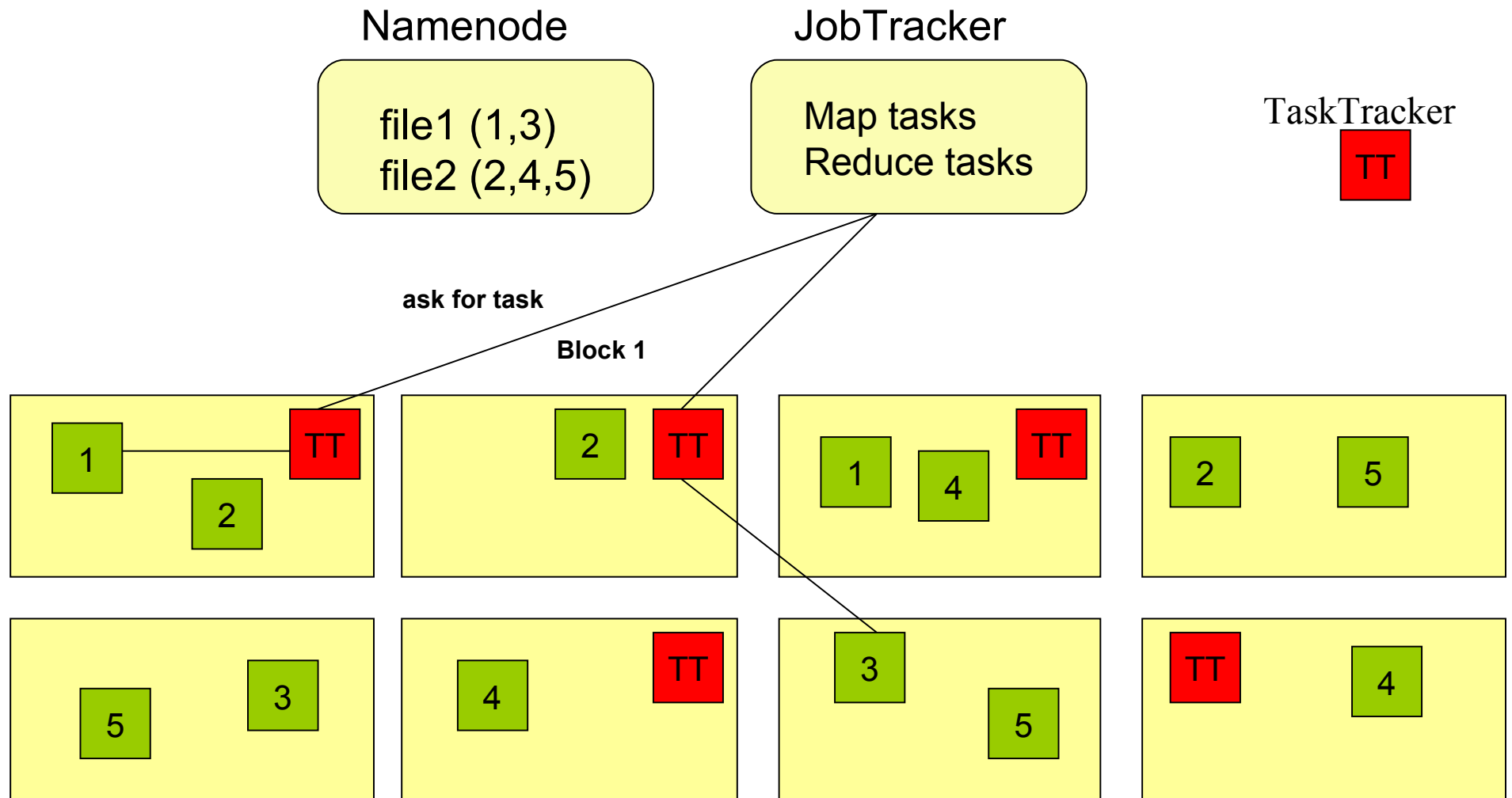
name:/users/bobYahoo/someData.gzip, copies:3, blocks:{2,4,5}



# About Data Locality ...

## HDFS 如何達成在地運算 ...

- Increase reliability and read bandwidth
  - robustness : read replication while found any failure
  - High read bandwidth : distribute read ( but increase write bottleneck )



# About Fault Tolerance ...

## HDFS 如何達成容錯機制 ...

資料崩毀  
Data Corrupt

網路或資料  
節點失效  
Network Fault  
DataNode Fault

名稱節點錯誤  
NameNode Fault

- 資料完整性 Data integrity
  - checked with CRC32
  - 用副本取代出錯資料
  - Replace corrupt block with replication one
- Heartbeat
  - Datanode send **heartbeat** to Namenode
- Metadata
  - FSImage、Editlog 為核心印象檔及日誌檔
  - FSImage – core file system mapping image
  - Editlog – like. SQL transaction log
  - 多份儲存，當名稱節點故障時可以手動復原
  - Multiple backups of FSImage and Editlog
  - Manually recovery while NameNode Fault



# Coherency Model and Performance of HDFS

## HDFS 的一致性機制與效能 ...

- **檔案一致性機制 Coherency model of files**
  - 刪除檔案 \ 新增寫入檔案 \ 讀取檔案皆由名稱節點負責
  - NameNode handle the operation of write, read and delete.
- **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
  - By default, the block size is 64MB
  - 大區塊可提高存取效率
  - Bigger block size will enhance read performance
  - 檔案有可能大過一顆磁碟
  - Single file stored on HDFS might be larger than single physical disk of DataNode.
  - 區塊均勻散佈各節點以分散讀取流量
  - Fully distributed blocks increase throughput of reading.

# POSIX like HDFS commands

## 與 **POSIX** 相似的操作指令 ...

```
jazz@hadoop:~$ hadoop fs
Usage: java FsShell
    [-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]]
```



**Questions?**

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

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



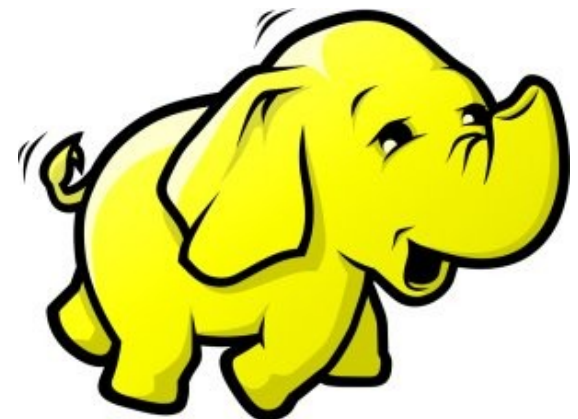
Powered by **DRBL**



# MapReduce 簡介

*Introduction to MapReduce*

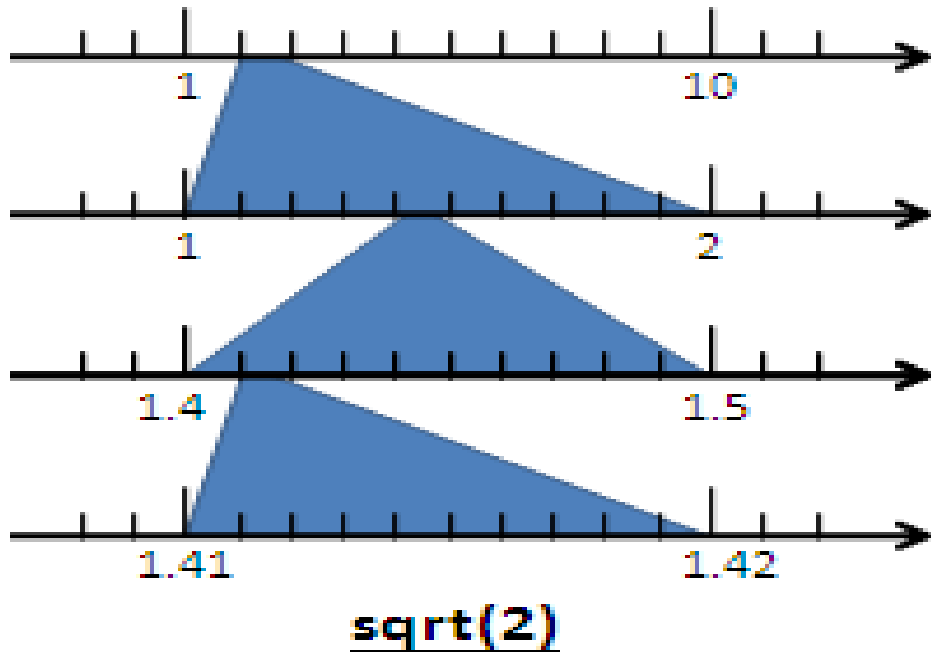
***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



# Divide and Conquer Algorithms

## 分而治之演算法

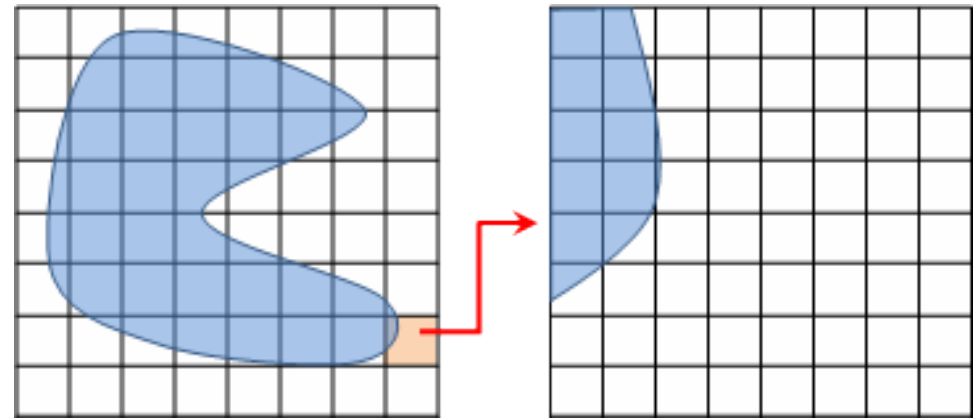
Example 1:



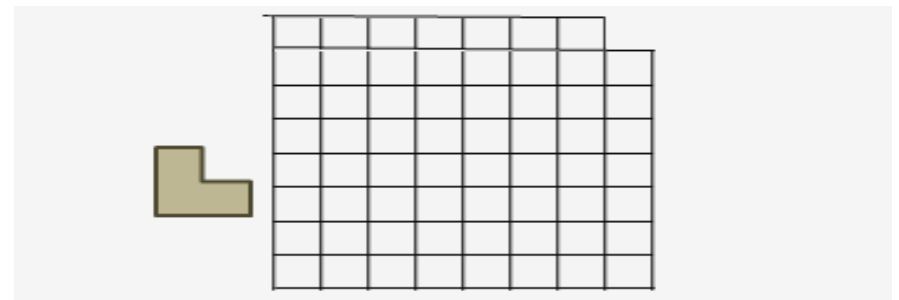
Example 4: The way to climb 5 steps stair within 2 steps each time. 眼前有五階樓梯，每次可踏上一階或踏上兩階，那麼爬完五階共有幾種踏法？

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

Example 2:



Example 3:



# What is MapReduce ??

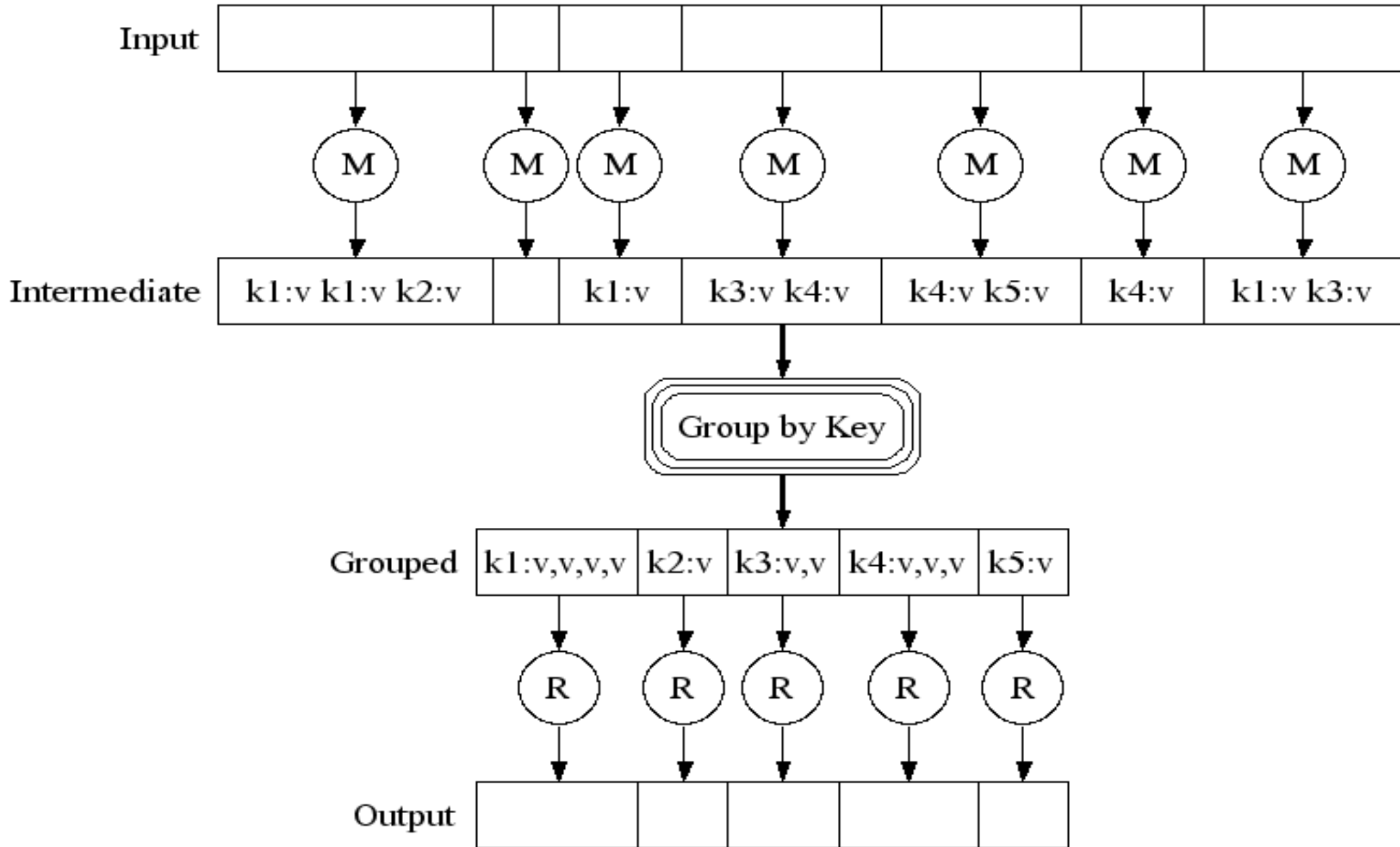
## 什麼是 MapReduce ??

- MapReduce 是 Google 申請的軟體專利，主要用來處理大量資料
- MapReduce is a **patented** software framework introduced by **Google** to support distributed computing on large data sets on clusters of computers.
- 啟發自函數編程中常用的 map 與 reduce 函數。
- The framework is inspired by **map** and **reduce** functions commonly used in **functional programming**, although their purpose in the MapReduce framework is not the same as their original forms
  - Map(...):  $N \rightarrow N$ 
    - Ex. [ 1,2,3,4 ] – (**\*2**) -> [ 2,4,6,8 ]
  - Reduce(...):  $N \rightarrow 1$ 
    - [ 1,2,3,4 ] - (**sum**) -> 10
- **Logical view of MapReduce**
  - **Map(k1,v1) -> list(k2,v2)**
  - **Reduce(k2, list (v2)) -> list(v3)**

Source: <http://en.wikipedia.org/wiki/MapReduce>

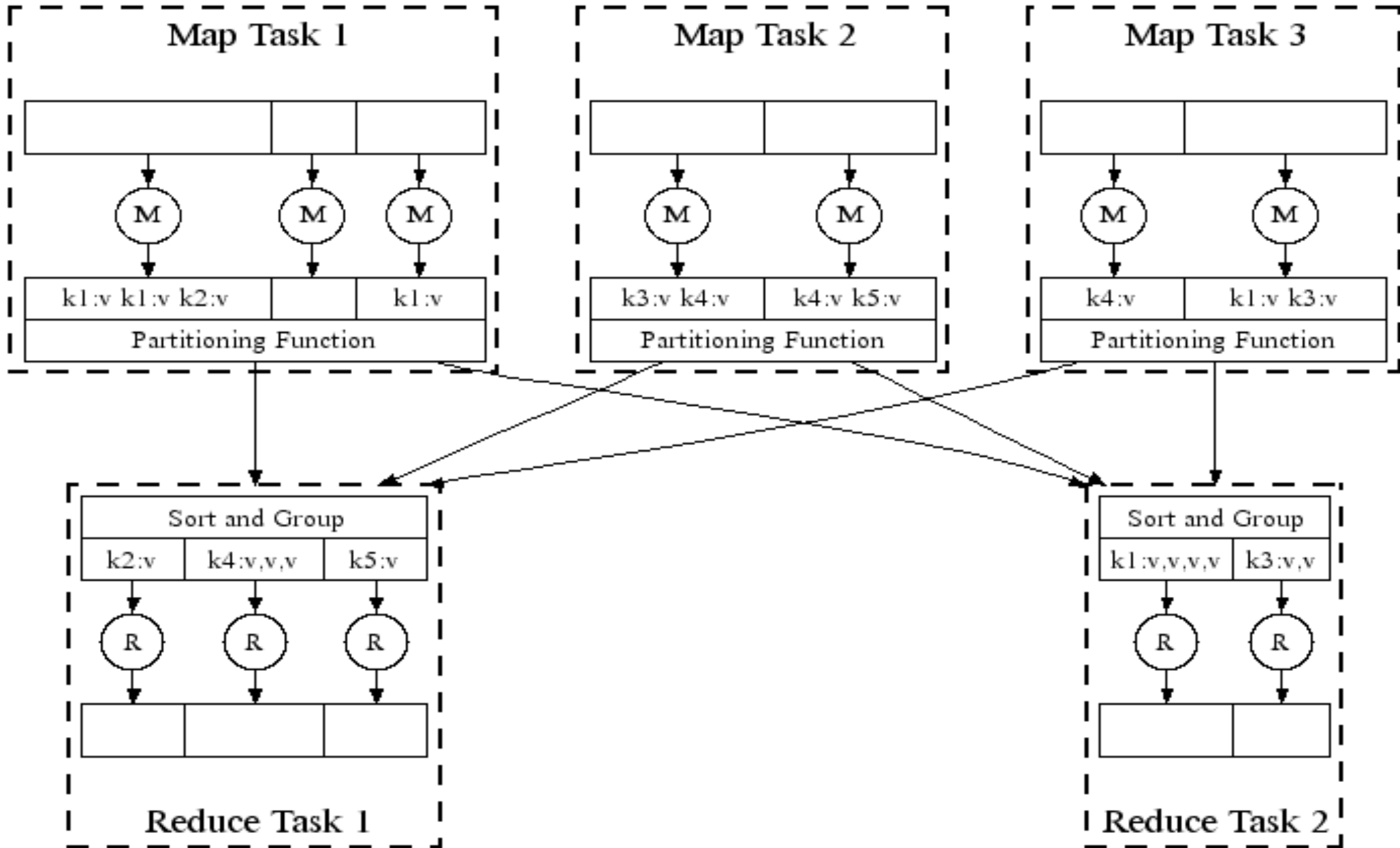
# Google's MapReduce Diagram

## Google 的 MapReduce 圖解



# Google's MapReduce in Parallel

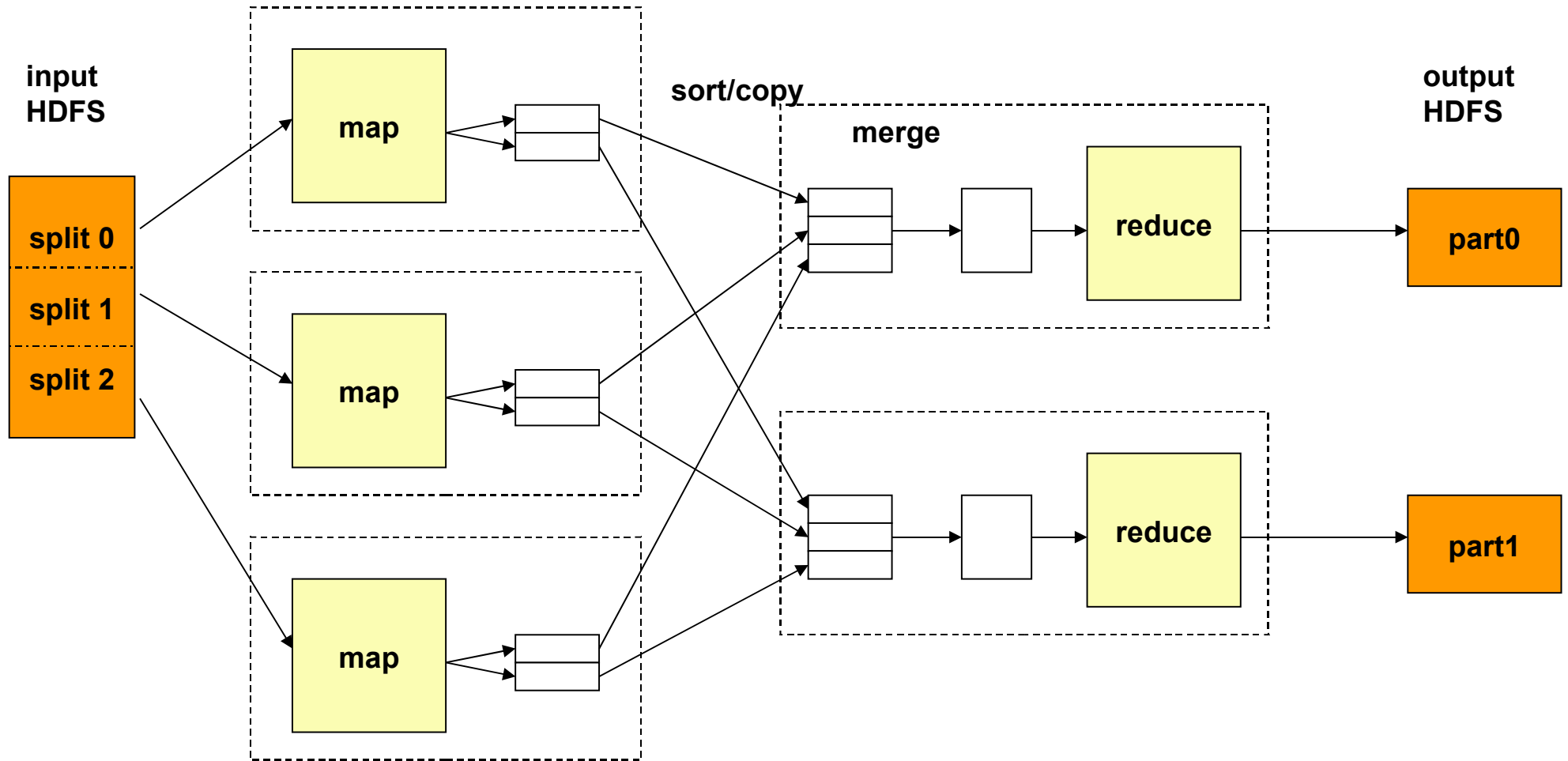
## Google 的 MapReduce 平行版圖解





# How does MapReduce work in Hadoop

## 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

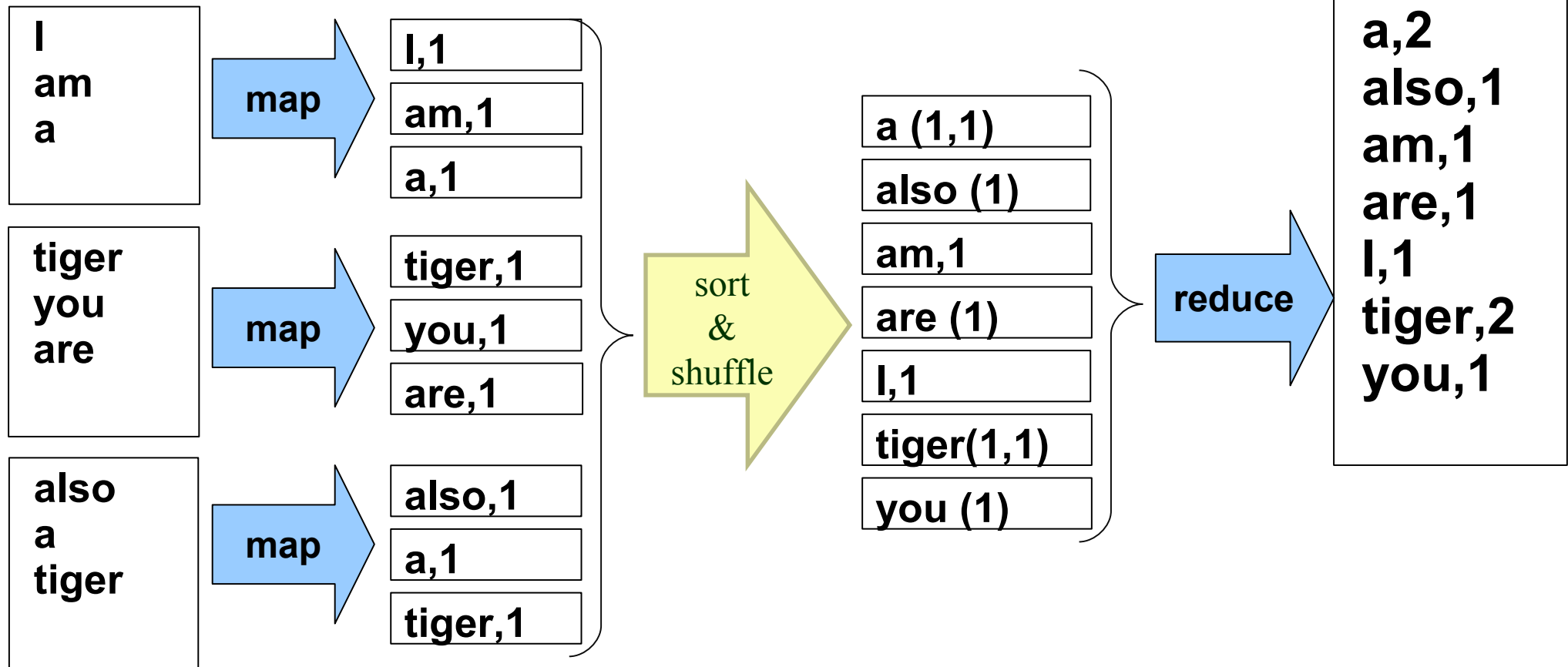
JobTracker 派遣 TaskTracker 作 reduce

reduce 完後通知 JobTracker 與 Namenode 以產生 output

# MapReduce by Example (1)

## MapReduce 運作實例 (1)

I am a tiger, you are also a tiger



JobTracker 先選了三個 Tracker 做 map

Map 結束後，hadoop 進行中間資料的重組與排序

JobTracker 再選一個 TaskTracker 作 reduce

# MapReduce by Example (2)

## MapReduce 運作實例 (2)

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \begin{bmatrix} \text{sqrt}(a + b) \\ \text{sqrt}(c + d) \end{bmatrix}$

$\begin{bmatrix} 1.0 & 0.0 & 3.0 \\ 3.2 & 0.8 & 32.0 \\ 1.0 & 14.0 & 1.0 \end{bmatrix} \rightarrow ?$

Input File

```
0 0 1.0 // A[0][1] = 1.0
0 1 0.0 // A[0][1] = 0.0
0 2 3.0 // A[0][2] = 3.0
1 0 3.2 // A[1][0] = 3.2
1 1 0.8 // A[1][1] = 0.8
```

map

```
(0, 1.0)
(0, 0.0)
(0, 3.0)
(1, 3.2)
(1, 0.8)
```

```
1 2 32.0 // A[1][2] = 32.0
2 0 1.0 // A[2][0] = 1.0
2 1 14.0 // A[2][1] = 14.0
2 2 1.0 // A[2][2] = 1.0
```

map

```
(1, 32.0)
(2, 1.0)
(2, 14.0)
(2, 1.0)
```

sort /  
merge

```
(0, {1.0, 0.0, 3.0})
(1, {3.2, 0.8, 32.0})
(2, {1.0, 14.0, 1.0})
```

reduce

```
(0, sqrt(1.0 + 0.0 + 3.0))
(1, sqrt(3.2 + 0.8 + 32.0))
(2, sqrt(1.0 + 14.0 + 1.0))
```

# *MapReduce is suitable to ....*

## **MapReduce** 合適用於 ....

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

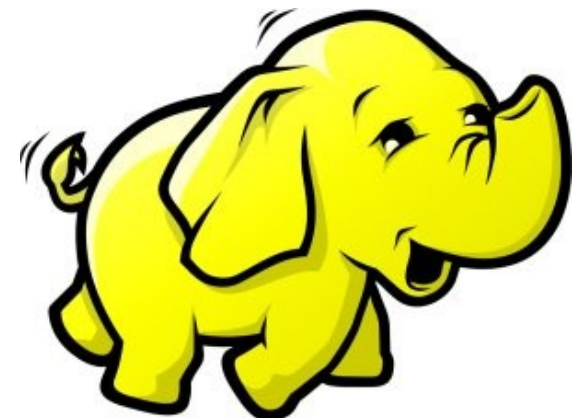
- <http://www.dbms2.com/2008/08/26/known-applications-of-mapreduce/>
- <http://wiki.apache.org/hadoop/PoweredBy>



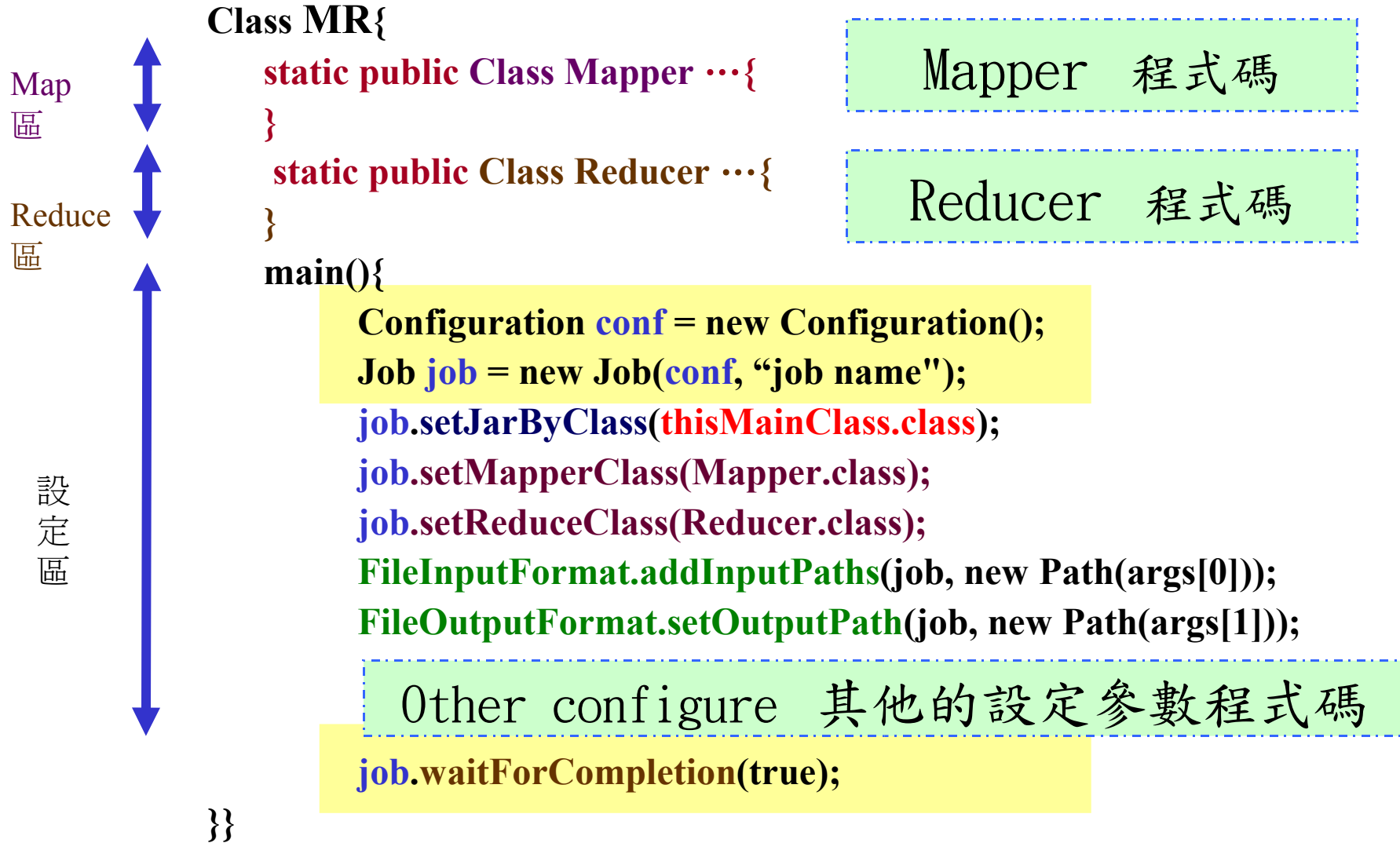
# MapReduce 程式設計入門

*MapReduce Programing 101*

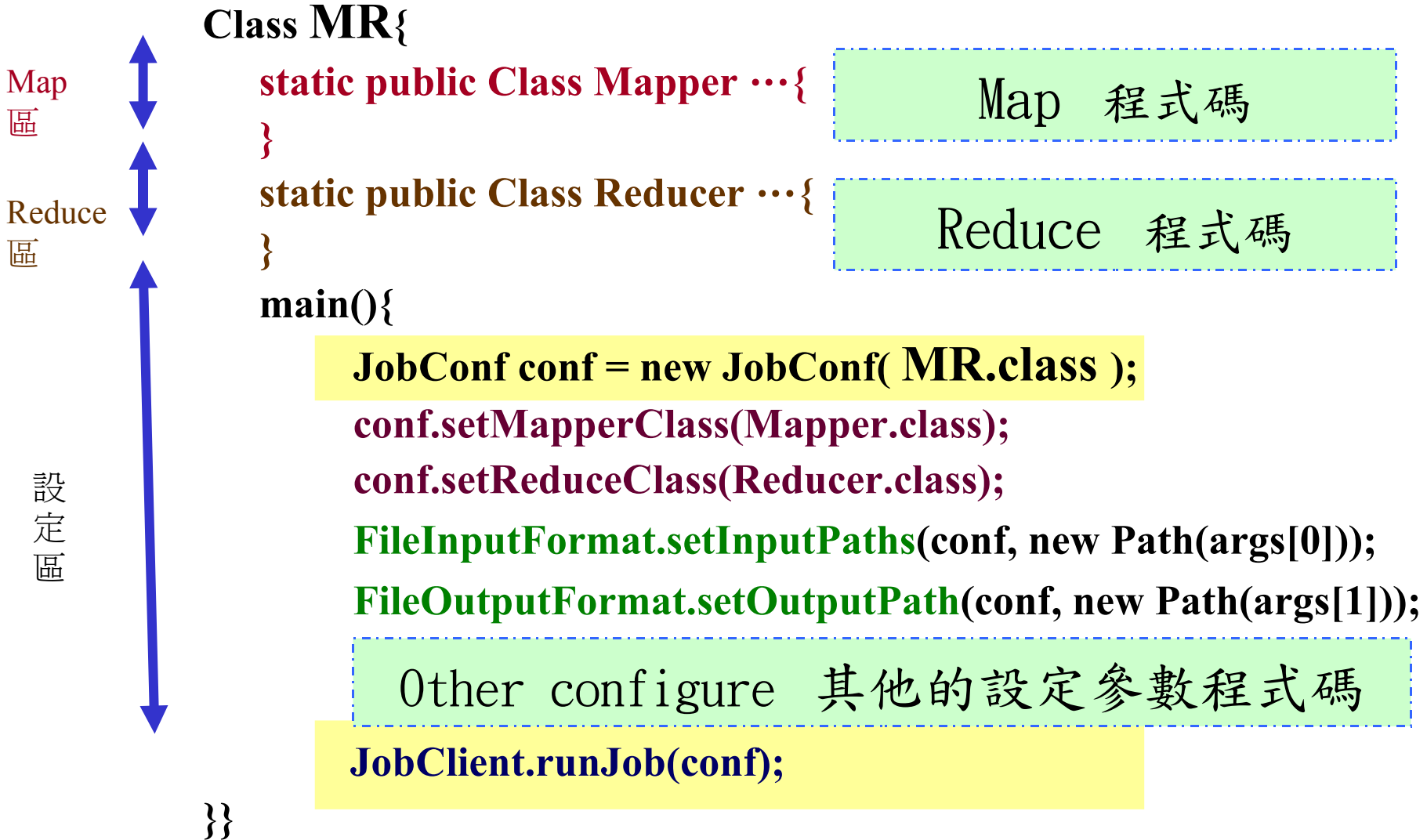
***Jazz Wang***  
***Yao-Tsung Wang***  
***jazz@nchc.org.tw***



# Program Prototype (v 0.20)

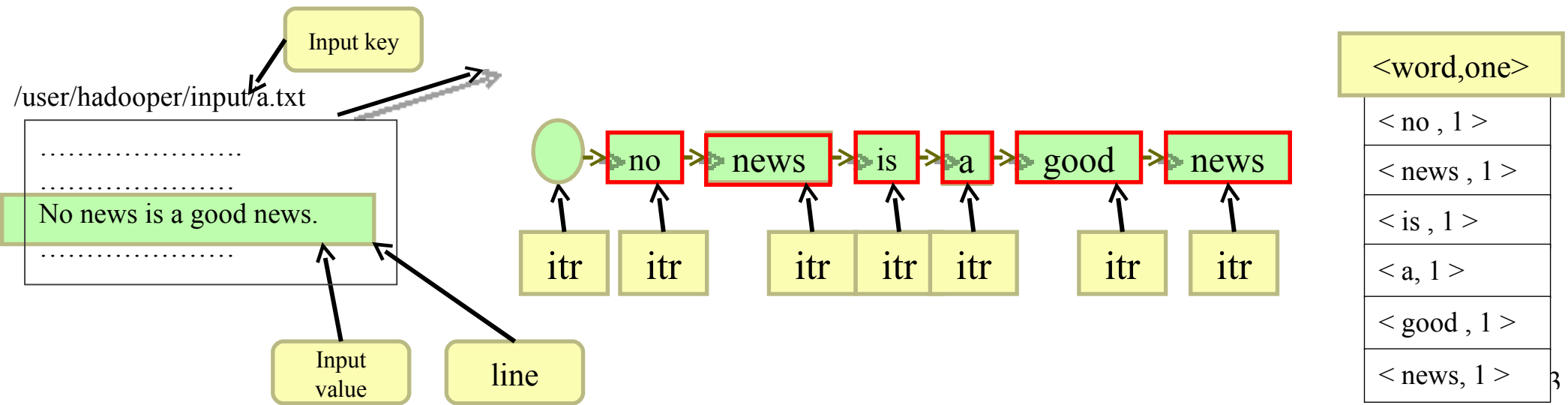


# Program Prototype (v 0.18)



# Word Count - mapper

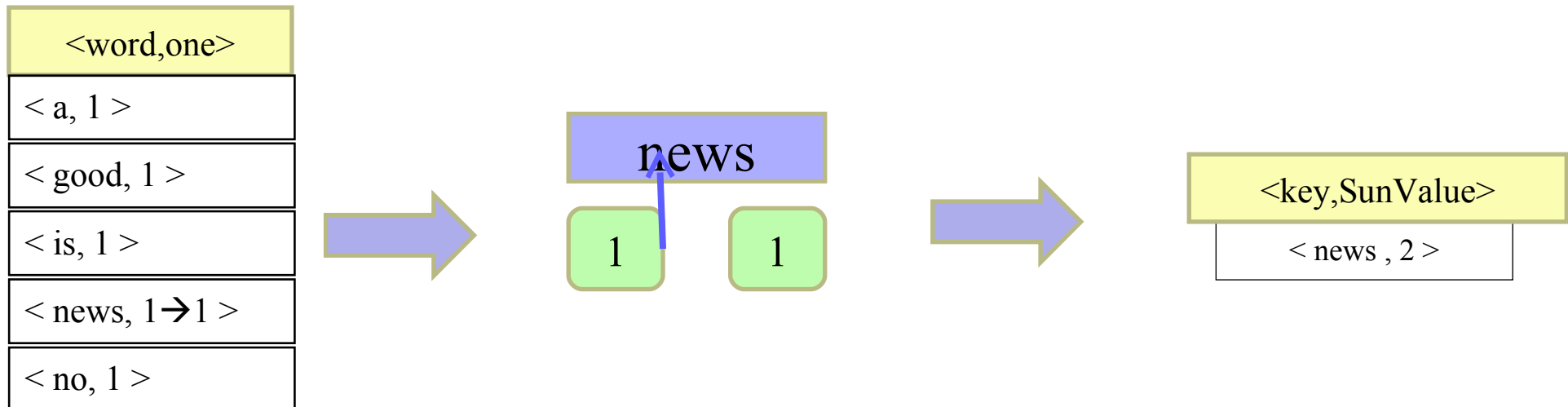
```
1 class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable> {  
2     private final static IntWritable one = new IntWritable(1);  
3     private Text word = new Text();  
4     public void map( LongWritable key, Text value, Context context)  
5         throws IOException , InterruptedException {  
6         String line = ((Text) value).toString();  
7         StringTokenizer itr = new StringTokenizer(line);  
8         while (itr.hasMoreTokens()) {  
9             word.set(itr.nextToken());  
10            context.write(word, one);  
11        }  
12    }  
13 }
```





# Word Count - reducer

```
1 class MyReducer extends Reducer< Text, IntWritable, Text, IntWritable> {  
2     IntWritable result = new IntWritable();  
3     public void reduce( Text key, Iterable<IntWritable> values, Context context)  
4         throws IOException, InterruptedException {  
5         int sum = 0;  
6         for ( IntWritable val : values )  
7             sum += val.get();  
8         result.set(sum);  
9         context.write ( key, result);  
10    }  
11 }
```



# *Word Count – main program*

```
Class WordCount{
  main()
    Configuration conf = new Configuration();
    Job job = new Job(conf, “job name” );
    job.setJarByClass(thisMainClass.class);
    job.setMapperClass(MyMapper.class);
    job.setReducerClass(MyReducer.class);
    FileInputFormat.addInputPaths(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.waitForCompletion(true);
}
```



## **Questions?**

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

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



Powered by **DRBL**



# Hadoop 進階課程

## HBase 資料庫應用

< V 0.20 >

王耀聰 陳威宇

Jazz@nchc.org.tw

waue@nchc.org.tw



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



# 一、導論

原本我們使用關聯式資料庫好好的，為何又要有新資料庫的儲存架構型態，是有其必要？或新技術可完全取而代之？還是只是一個等待泡沫化新技術的濫觴？

# 海量資料緒論

- Hadoop 能運算海量資料，然後呢？
  - ◆ 其實 Hadoop 運算出來的結果也不少
- 如何處理 Hadoop 運算出來的資料？
  - ◆ 再用 Hadoop 運算一次??
- 海量資料也需要整理
  - ◆ 排序
  - ◆ 搜尋
  - ◆ 選擇

# RDBMS / 資料庫

- Relational DataBase Management System = 關聯式資料庫管理系統
  - ◆ Oracle、IBM DB2、SQL Server、MySQL...

資料庫管理系統 (DBMS) (檢視 · 討論 · 編輯 · 歷史)

## 概念

資料庫 · 資料庫模型 · 資料庫儲存結構 · 關聯 (資料庫) · 關聯模型 · 分布式資料庫 · ACID · Null值  
關聯式資料庫 · 關聯代數 · 關聯演算 · 元組關聯演算 · 域關聯演算 · 資料庫正規化 · 參照完整性 · 關聯式資料庫管理系統  
主鍵 · 外來鍵 · 代理鍵 · 超鍵 · 候選鍵

## 資料庫元件

觸發器 · 檢視 · 資料庫表 · 指標 (資料庫) · 事務日誌 · 資料庫事務 · 資料庫索引  
儲存程式 · 資料庫分割

## SQL

分類：資料查詢語言DQL · 資料定義語言DDL · 資料操縱語言DML · 資料控制語言DCL  
指令：SELECT · INSERT · UPDATE · MERGE · DELETE · JOIN · UNION · CREATE · DROP · Begin work · COMMIT · ROLLBACK · TRUNCATE · ALTER  
安全：SQL資料隱碼攻擊 · 參數化查詢

## 資料庫管理系統的實施

### 實施型式

關聯式資料庫 · 檔案型資料庫 · Deductive · 維度化資料庫 · 階層式 · 物件資料庫 · 物件關聯式資料庫 · Temporal · XML資料庫

## 資料庫產品

物件型 (對比) · 關聯型 (對比)

## 資料庫成分

查詢語言 · 查詢最佳化器 · 查詢計畫 · 嵌入式SQL · ODBC · JDBC · OLE DB

# RDBMS 碰上大資料

- RDBMS 的好處
  - ◆ 提供了很多而且很豐富的操作方式
  - ◆ SQL語法普遍被使用
- 但當資料量愈來愈大時，會遇到單台機器的”囧”境
  - ◆ 網路頻寬有限
  - ◆ 空間有限
- 走向多台機器架構



# 跨足多台機器的 RDBMS

- 讀取的 query 比寫入的 query 多
  - ◆ **Replication**
- slave 過多時，造成每台記憶體內重複 cache 相同元素
  - ◆ **Memcached**
- 寫入的 query 超過單台可以負荷的量時，replication 技術則導致每台 Slave 一起掛
  - ◆ **Sharding**
  - ◆ 依照 id，把資料拆散到各台（如 Flickr）

# 多台機器的 RDBMS 的缺點

- 需要 application server 或是 library 配合，否則第三方程式找不到資料放在哪個 node
- 無法隨意使用 JOIN 及 transaction，即使可以硬要使用效能也很差
- 設計 schema 時必須注意，當一個 cluster 愈來愈大時要 rebalance

# 是否非RDBMS不可？

- Web 2.0 網站很多時候
  - ◆ 不需要transaction
  - ◆ 減少JOIN 次數
  - ◆ 多次 SELECT 拉資料
- 一開始寫在一台DB主機的SQL程式無法再套用於後來多台SQL主機的架構上
  - ◆ 程式有可能全部重寫

# 將RDBMS簡化吧

- RDBMS -> key-value DataBase
  - ◆ 簡化掉不需要的功能，到只剩下key-value的架構
    - GET(key)
    - SET(key, value)
    - DELETE(key)
- 類似 Excel

# Distributed key-value System

- key-value DataBase -> Distributed key-value DataBase
- 加強 key-value 架構的 scalability，使得增加機器就可以增加容量與頻寬
- 適合管理大量分散於不同主機的資料
- 通稱為 NoSQL DataBase

# 常見的 NoSQL

## OpenSource

- HBase (Yahoo!)
- Cassandra (Facebook)
- MongoDB
- CouchDB (IBM)
- SimpleDB (Amazon)

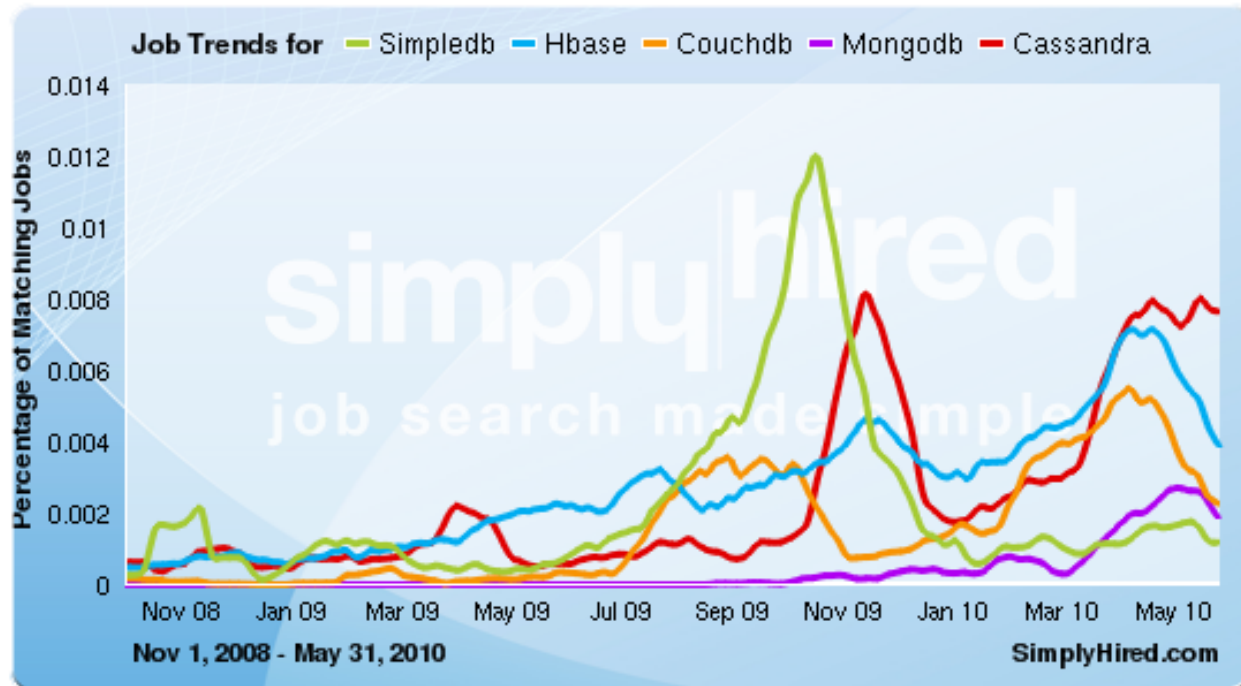
## Commercial

- BigTable (Google)

# 2010 年 NoSQL 職缺排行榜

Simpledb, Hbase, Couchdb, Mongoddb, Cassandra Trends

1. Cassandra
2. HBase
3. CouchDB
4. MongoDB
5. SimpleDB



## Simpledb, Hbase, Couchdb, Mongoddb, Cassandra Job Trends

This graph displays the percentage of jobs with your search terms anywhere in the job listing. Since November 2008, the following has occurred:

- [Simpledb jobs](#) increased 357%
- [Hbase jobs](#) increased 745%
- [Couchdb jobs](#) did not change or there is no data available
- [Mongoddb jobs](#) increased 18,480%
- [Cassandra jobs](#) did not change or there is no data available

(2010-07-25)

# 一、HBase 介紹

介紹HBase如何而來，它的 Why, What, How ....，以及它的架構

HBase, *Hadoop database*, is an open-source, distributed, versioned, column-oriented store modeled after Google' Bigtable. Use it when you need random, realtime read/write access to your Big Data.



# BigTable ?

- Bigtable: 一個結構化數據的分佈式存儲系統
- Google Style的數據庫，使用結構化的文件來存儲數據
- 不支持關聯或是類似於SQL的高級查詢。
- 大規模處理、高容錯性
- PB級的存儲能力
- 每秒數百萬的讀寫操作

# HBase

- 設計概念與結構類似Bigtable
- HBase 以 Hadoop 分散式檔案系統 (HDFS) 為基礎，提供類Bigtable 功能
- HBase 是具有以下特點的儲存系統：
  - ◆ 類似表格的資料結構 (Multi-Dimensional Map)
  - ◆ 分散式
  - ◆ 高可用性、高效能
  - ◆ 很容易擴充容量及效能
- HBase 適用於利用數以千計的一般伺服器上，來儲存Petabytes級的資料。
- HBase同時提供Hadoop MapReduce程式設計。

# 開發歷程

- Started toward by Chad Walters and Jim
- 2006.11
  - ◆ Google releases paper on BigTable
- 2007.2
  - ◆ Initial HBase prototype created as Hadoop contrib.
- 2007.10
  - ◆ First useable HBase
- 2008.1
  - ◆ Hadoop become Apache top-level project and HBase becomes subproject
- 2010.3
  - ◆ HBase graduates from Hadoop sub-project to Apache Top Level Project
- 2010.7
  - ◆ HBase 0.20.6 released

# 誰使用 HBase

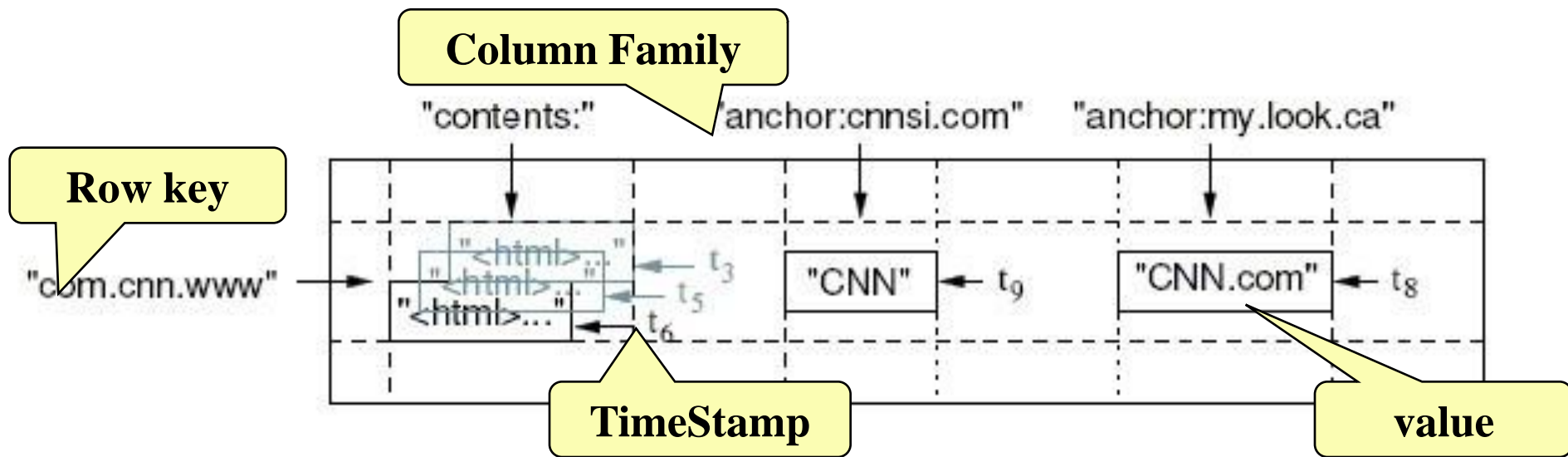
- Adobe
  - ◆ 內部使用 (Structure data)
- Kalooga
  - ◆ 圖片搜尋引擎 <http://www.kalooga.com/>
- Meetup
  - ◆ 社群聚會網站 <http://www.meetup.com/>
- Streamy
  - ◆ 成功從 MySQL 移轉到 Hbase <http://www.streamy.com/>
- Trend Micro
  - ◆ 雲端掃毒架構 <http://trendmicro.com/>
- Yahoo!
  - ◆ 儲存文件 fingerprint 避免重複 <http://www.yahoo.com/>
- More
  - ◆ <http://wiki.apache.org/hadoop/Hbase/PoweredBy>

# 為什麼使用HBase?

- 不是關聯式(Relational)資料庫系統
  - ◆ 表格(Table)只有一個主要索引 (primary index) 即 row key.
  - ◆ 不提供 join
  - ◆ 不提供 SQL 語法。
- 提供Java函式庫, 與 REST與Thrift等介面。
- 提供 getRow(), Scan() 存取資料。
  - ◆ getRow()可以取得一筆row range的資料, 同時也可以指定版本 (timestamp)。
- Scan()可以取得整個表格的資料或是一組row range (設定start key, end key)
- 有限的單元性(Atomicity)與交易 (transaction)功能.
- 只有一種資料型態 (bytes)
- 可以配合MapReduce框架, 進行複雜的分析與查詢

# Data Model

- Table依 *row key* 來自動排序
- Table schema 只要定義 *column families*.
  - ◆ 每個column family 可有無限數量的 columns
  - ◆ 每個column的值可有無限數量的時間版本(timestamp)
  - ◆ Column可以動態新增，每個row可有不同數量的columns。
  - ◆ 同一個column family的columns會群聚在一個實體儲存單元上，且依column 的名稱排序。
  - ◆ byte[] 是唯一的資料型態(Row, Family: Column, Timestamp) Value



# Data Model

- HBase實際上儲存Table時，是以column family為單位來存放

Row Key	Time Stamp	Column (Family) “content:”
com.cnn.www	t9	“<html>...”
	t6	“<html>...”

Row Key	Time Stamp	Column (Family) “anchor:”
com.cnn.www	t9	“anchor:cnnsi. com” “CNN”
	t8	“anchor:cnnsi. com” “CNN”
		“anchor:my.loc com” “MyLook”

# HTable 成員

Table, Family, Column, Qualifier, Row, TimeStamp

		Contents	Department		
			news	bid	sport
t1	com.yahoo.news.t w	“撿到學雜費，硬要分三成”	“tech”		
t2		“科研論文評比 5校進500 大”	“tech”		
t3		“罰蹲立300下！班長「住 院」 師懊悔”	“tech”		
t1	com.yahoo.bid.tw	“… iphone 4G 9/17 日上 市”		“3C”	
t1	com.yahoo.sport.t w	“Nadal 大滿貫”			“MBA”

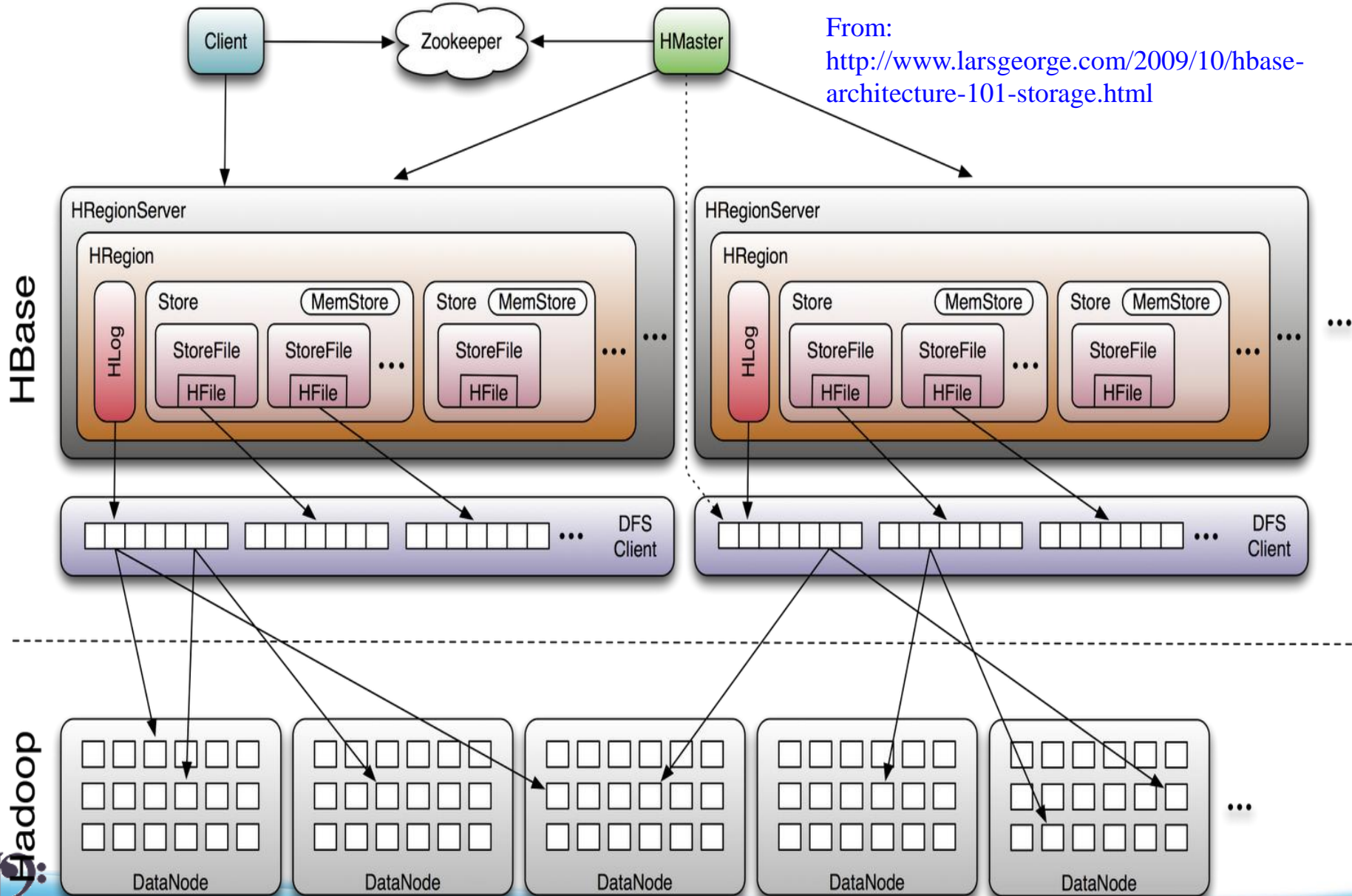


# Regions

- 表格是由一或多個 region 所構成
  - ◆ Region 是由其 startKey 與 endKey 所指定
- 每個 region 可能會存在於多個不同節點上，而且是由數個HDFS 檔案與區塊所構成，這類 region 是由 Hadoop 負責複製

Region	Row Keys	Column Family “Content”
Region 1	00000	...
	00001	...
	...	...
	09999	...
Region 2	10000	...
	...	...
	...	...
	29999	...

# HBase 與 Hadoop 搭配的架構



From:  
<http://www.larsgeorge.com/2009/10/hbase-architecture-101-storage.html>

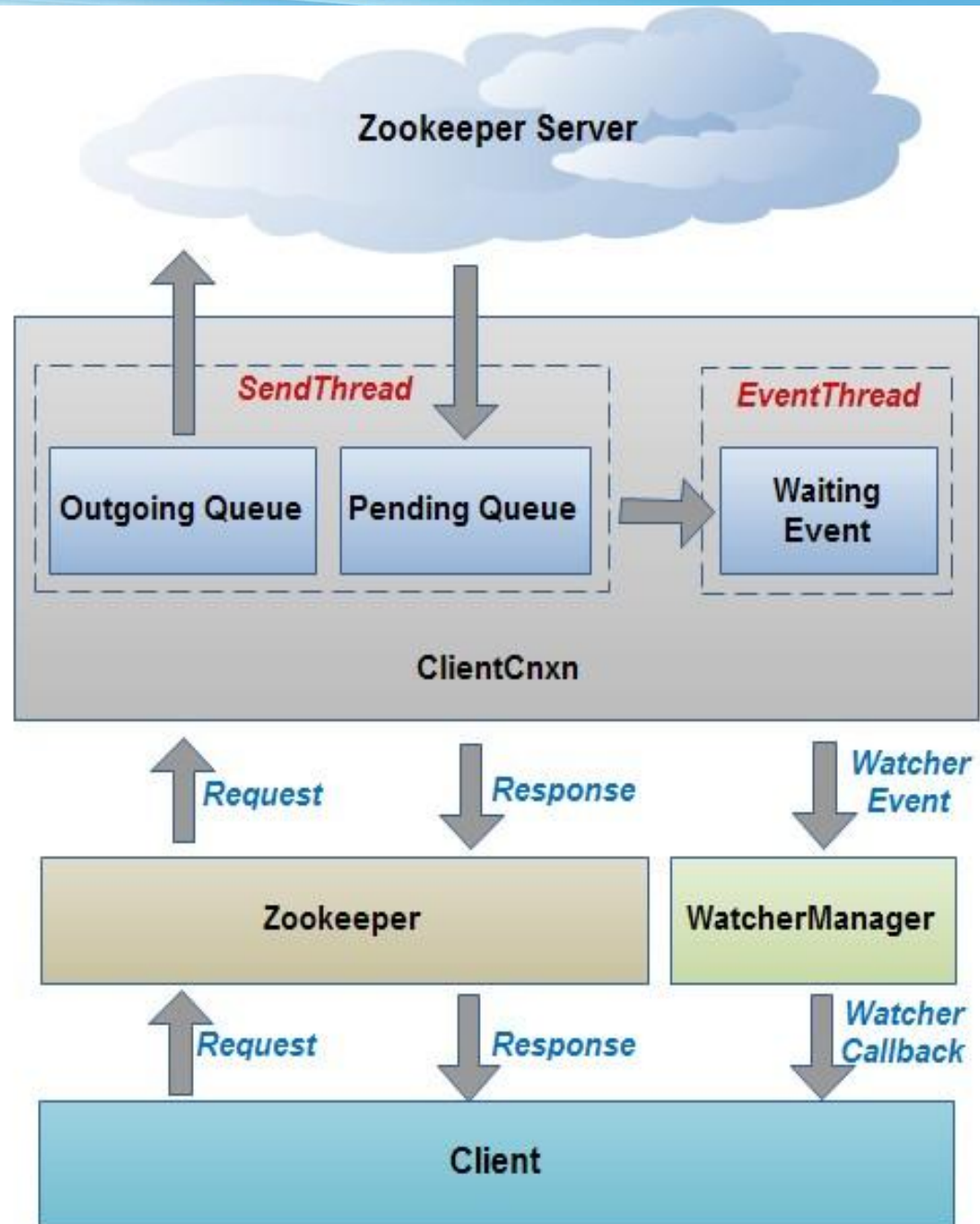


# HBase 0.20 特色

- 解決單點失效問題（single point of failure）
  - ◆ Ex: Hadoop NameNode failure
- 設定檔改變或小版本更新會重新啟動
- 隨機讀寫（Random access）效能如同 MySQL

# Zookeeper ?

- Hadoop的正式子項目
- 針對大型分散式系統的可靠協調系統
- Google的Chubby
- 存儲一些配置信息，確保文件寫入的一致性
- Master / Client 架構，Master 可由選舉而得





# 雲端入侵偵測日誌分析平台

Part-7 : Building IDS-log Cloud Analysis System (ICAS)

**Yao-Tsung Wang**

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

**Wei-Yu Chen**

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



# 專家說：雲端每個環節都有安全問題

ZDNet Taiwan - 專家談雲端：每個環節都有安全問題 - 新聞

2010/08/10 19:50:02

專家談雲端：每個環節都有安全問題

ZDNet記者曠文濤／台北報導 雲端的安全問題不是無解，只是不管是雲端服務供應商或者想要建立私有雲的企業用戶，都必須考量到每個環節。

微軟亞太區全球技術支援中心專案經理、同時也是ZDNet專欄作家林宏嘉今（10）日在ZDNet舉行的IT Priorities圓桌論壇中表示，**雲端的安全議題涉及了IaaS、PaaS乃至於SaaS的每個層面**，當然有些問題是原本就存在：例如在討論到IaaS時，就涉及到了**機房的管理**和**硬體設備的可用性**等；但是講到PaaS時，企業用戶倘若要選擇開原碼的作業系統，必須考量到後續的**安全維護**；在SaaS的層次，企業用戶必須確保每一個分區（partition）的安全更新和**資料安全**。

目前正如火如荼建立台灣第一個校園私有雲的台大計算機及資訊網路中心主任孫雅麗則呼應道，Amazon的雲端服務證實了在Hypervisor層有駭客入侵，也就是意味著過去大家在討論如何防範**虛擬機器的資料安全**，但是威脅已經深化到了更下一層。這些問題都有待解決。

「有些問題甚至是來自於內部，舉例而言，MIS可能會把存在記憶體裡的資料倒出來，或者在Hypervisor層就植入了可以蒐集資料的程式，」孫雅麗說。

安全議題是目前台灣企業對雲端持保留態度的最大主因，這也是何以台灣的大型企業對於雲端的想法，還是  
傾好建立私有雲。畢竟對用戶而言，資料放在別人家，還和其他企業，甚至具競爭對手「共處」，

# 雲端資安的範疇

用雲端  
處理資安

**Dealing Security  
issues using Cloud**

**Data Security  
In the Cloud**

雲內部  
的資安管制

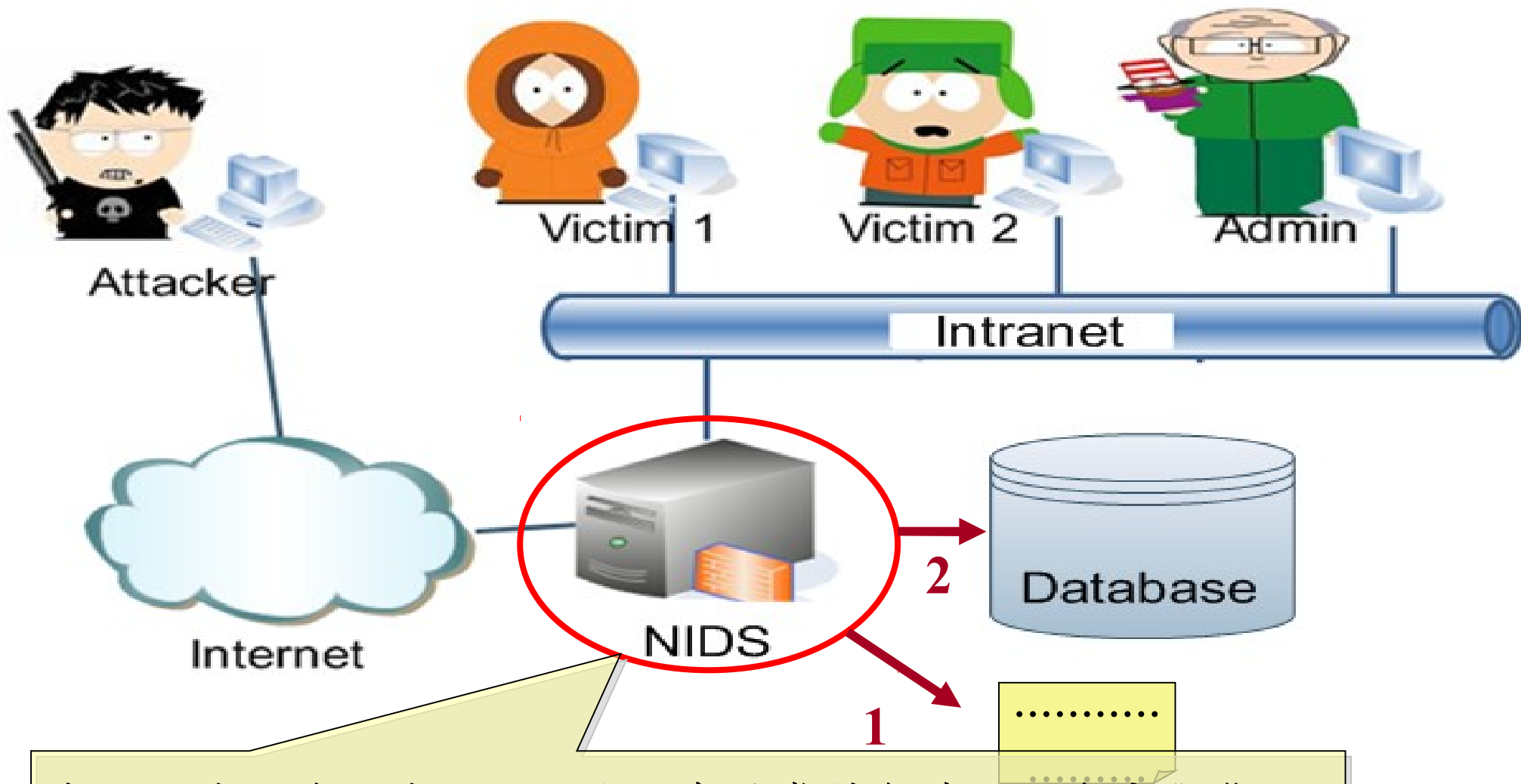
**Security Issues  
Inside the Cloud**

雲端資料  
安全性

端本身  
的資安威脅

**Security Threats  
to Internet of Things**

# 使用入侵偵測系統 (NIDS) 來找出入侵訊息



當入侵偵測系統偵測到網路上有異常封包時，就會產生警訊以告知有攻擊發生。警訊通常有兩種形式：  
1. 紀錄成 log 檔 2. 紀錄到資料庫



# 傳統 NIDS 的警訊型態 (1) 紀錄在日誌檔內

## 入侵偵測系統所產生警訊日誌檔內一小段內容

```
[**] [1:538:15] NETBIOS SMB IPC$ unicode share access [**]  
[Classification: Generic Protocol Command Decode] [Priority: 3]  
09/04-17:53:56.363811 168.150.177.165:1051 -> 168.150.177.166:139  
TCP TTL:128 TOS:0x0 ID:4000 IpLen:20 DgmLen:138 DF  
***AP*** Seq: 0x2E589B8 Ack: 0x642D47F9 Win: 0x4241 TcpLen: 20
```

```
[**] [1:1917:6] SCAN UPnP service discover attempt [**]  
[Classification: Detection of a Network Scan] [Priority: 3]  
09/04-17:53:56.385573 168.150.177.164:1032 -> 239.255.255.250:1900  
UDP TTL:1 TOS:0x0 ID:80 IpLen:20 DgmLen:161  
Len: 133
```

```
[**] [1:1917:6] SCAN UPnP service discover attempt [**]  
[Classification: Detection of a Network Scan] [Priority: 3]  
09/04-17:53:56.386910 168.150.177.164:1032 -> 239.255.255.250:1900  
UDP TTL:1 TOS:0x0 ID:82 IpLen:20 DgmLen:161  
Len: 133
```

```
[**] [1:1917:6] SCAN UPnP service discover attempt [**]  
[Classification: Detection of a Network Scan] [Priority: 3]  
09/04-17:53:56.388244 168.150.177.164:1032 -> 239.255.255.250:1900  
UDP TTL:1 TOS:0x0 ID:84 IpLen:20 DgmLen:161  
Len: 133
```

```
[**] [1:538:15] NETBIOS SMB IPC$ unicode share access [**]  
[Classification: Generic Protocol Command Decode] [Priority: 3]  
09/04-17:53:56.405923 168.150.177.164:1035 -> 168.150.177.166:139  
TCP TTL:128 TOS:0x0 ID:94 IpLen:20 DgmLen:138 DF  
***AP*** Seq: 0x82073DFF Ack: 0x2468EB82 Win: 0x4241 TcpLen: 20
```

```
[**] [1:1917:6] SCAN UPnP service discover attempt [**]  
[Classification: Detection of a Network Scan] [Priority: 3]  
09/04-17:53:56.417045 168.150.177.164:45461 -> 168.150.177.1:1900  
UDP TTL:1 TOS:0x0 ID:105 IpLen:20 DgmLen:161  
Len: 133
```

```
[**] [1:1917:6] SCAN UPnP service discover attempt [**]  
[Classification: Detection of a Network Scan] [Priority: 3]  
09/04-17:53:56.420759 168.150.177.164:45461 -> 168.150.177.1:1900  
UDP TTL:1 TOS:0x0 ID:117 IpLen:20 DgmLen:160  
Len: 132
```

```
[**] [1:1917:6] SCAN UPnP service discover attempt [**]  
[Classification: Detection of a Network Scan] [Priority: 3]  
09/04-17:53:56.422095 168.150.177.164:45461 -> 168.150.177.1:1900  
UDP TTL:1 TOS:0x0 ID:118 IpLen:20 DgmLen:161  
Len: 133
```

```
[**] [1:2351:10] NETBIOS DCERPC ISystemActivator path overflow attempt little endian  
unicode [**]  
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]  
09/04-17:53:56.442445 198.8.16.1:10179 -> 168.150.177.164:135  
TCP TTL:105 TOS:0x0 ID:49809 IpLen:20 DgmLen:1420 DF  
***A**** Seq: 0xF9589BBF Ack: 0x82CCF5B7 Win: 0xFFFF TcpLen: 20  
[Xref => http://www.microsoft.com/technet/security/bulletin/MS03-026.msp][Xref =>  
http://cgi.nessus.org/plugins/dump.php3?id=11808][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0352][Xref => http://www.securityfocus.com/bid/8205]
```

```
[**] [122:3:0] (portscan) TCP Portsweep [**]  
[Priority: 3]  
09/04-17:53:56.499016 198.8.16.1 -> 168.150.177.166  
PROTO:255 TTL:0 TOS:0x0 ID:1750 IpLen:20 DgmLen:168
```

# 傳統 NIDS 的警訊型態 (2) 紀錄在資料庫內

以下為利用瀏覽器透過網頁方式呈現警訊資料庫的內容

The screenshot shows a Mozilla browser window titled "Basic Analysis and Security Engine (BASE): Query Results - Mozilla". The browser's address bar is empty, and the main content area displays the BASE web interface. At the top, there is a blue header with the text "Basic Analysis and Security Engine (BASE)" and navigation links for "Home", "Search", and "AG Maintenance". A "[ Back ]" link is located on the right side. Below the header, a red message states "Added 0 alert(s) to the Alert cache". The "Queried DB on" timestamp is "Thu October 14, 2004 22:04:44". On the left, a table lists search criteria: Meta Criteria (any), IP Criteria (any), TCP Criteria (any), and Payload Criteria (any). On the right, a "Summary Statistics" box contains a bulleted list: Sensors, Unique Alerts (classifications), Unique addresses: source | destination, Unique IP links, Source Port: TCP | UDP, Destination Port: TCP | UDP, and Time profile of alerts. Below this, it says "Displaying alerts 1-50 of 81 total". The main part of the page is a table with the following columns: ID, Signature, Timestamp, Source Address, Dest. Address, and Layer 4 Proto. The table contains five rows of alert data, each with a checkbox in the ID column.

<input type="checkbox"/>	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
<input type="checkbox"/>	#0-(1-84)	[snort] NETBIOS SMB IPC\$ share unicode access	2004-10-08 11:25:41	192.168.1.100:1613	192.168.1.4:139	TCP
<input type="checkbox"/>	#1-(1-83)	[snort] NETBIOS SMB IPC\$ share unicode access	2004-10-08 11:25:31	192.168.1.100:1608	192.168.1.4:139	TCP
<input type="checkbox"/>	#2-(1-82)	[snort] NETBIOS SMB IPC\$ share unicode access	2004-10-08 11:25:05	192.168.1.100:1601	192.168.1.4:139	TCP
<input type="checkbox"/>	#3-(1-80)	[snort] (http_inspect) OVERSIZE CHUNK ENCODING	2004-10-04 22:25:41	192.168.1.4:42164	67.19.245.228:80	TCP
<input type="checkbox"/>	#4-(1-81)	[snort] (http_inspect) OVERSIZE CHUNK ENCODING	2004-10-04 22:25:41	192.168.1.4:42163	67.19.245.228:80	TCP

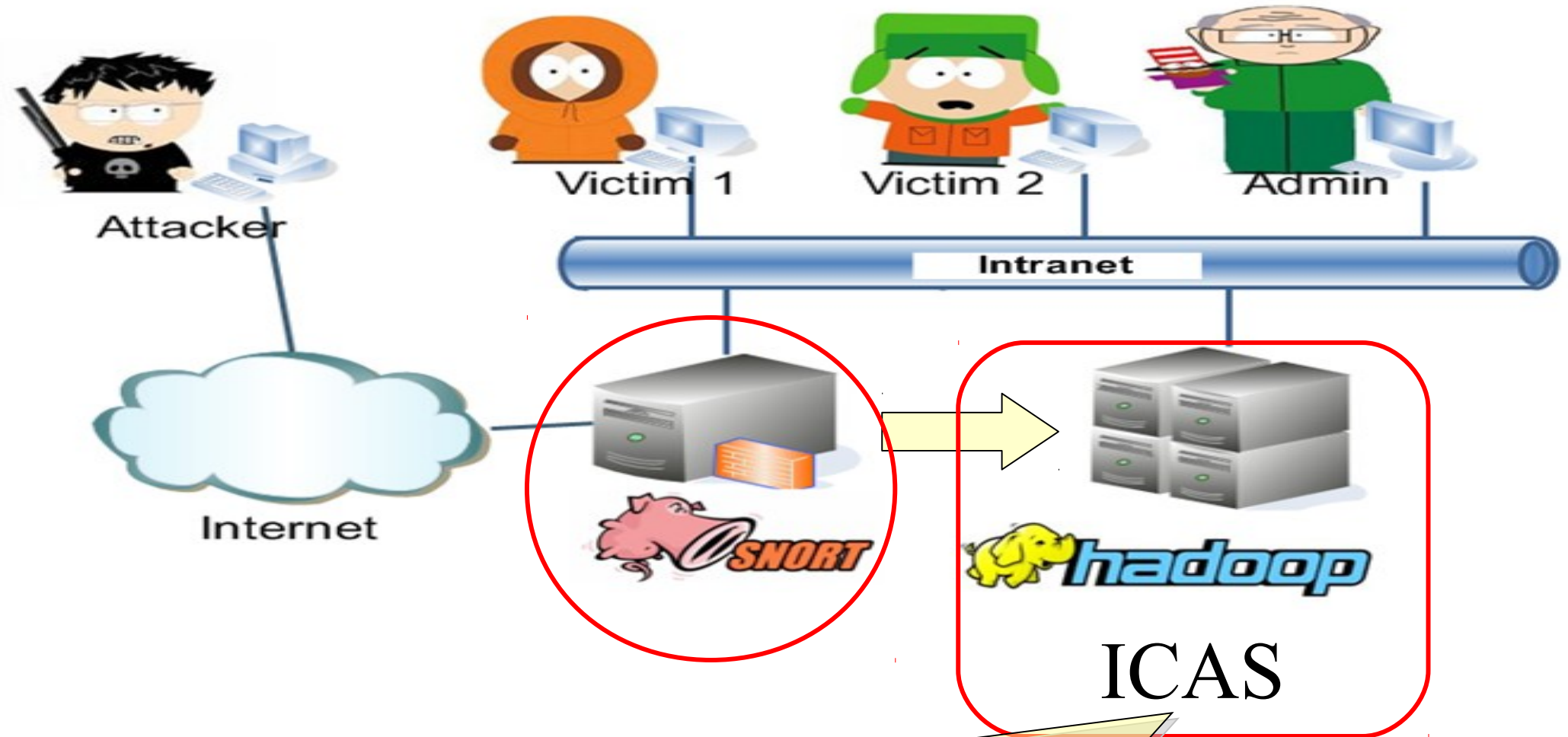
## 以上作法的缺點

- 警訊僅被『忠實』地被記錄下來，無法顯示彼此間的關聯性，因此系統管理者難以瞭解全部攻擊情形
- 過多的警訊，使得容易忽略重要內容
- 完全依賴單一台資料庫，當資料量一大，該台主機的讀寫效率將成爲瓶頸

# 使用雲端運算的解決方案：ICAS

- ICAS, *IDS Cloud Analysis System*
- 利用雲端運算的特性提供以下好處
  - 對大量資料有高效率
  - 一般主機的叢集
  - 有錯誤容忍
- 分析演算法
  - 整合
  - 關聯

# 透過 ICAS 協助分析 IDS 的警訊

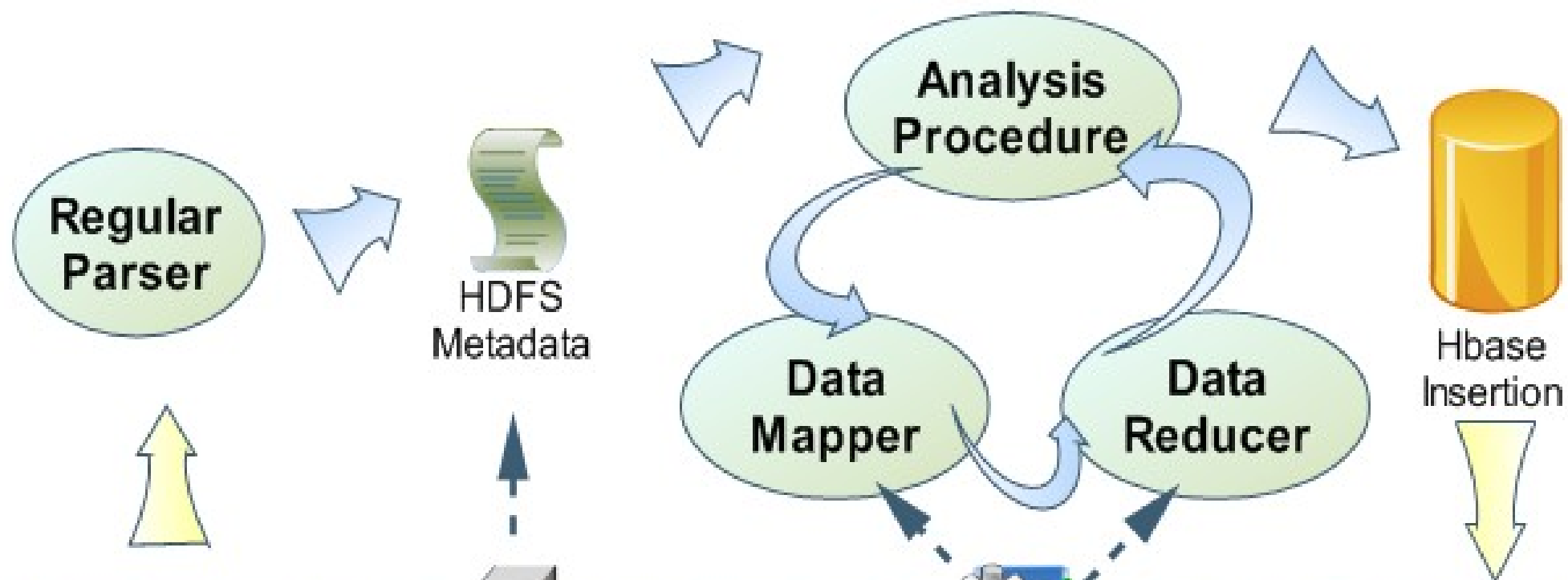


可多個 NIDS 共同產生警訊後，傳送至 ICAS，分析演算法  
目前有 ICAS-I 及 ICAS-II

# ICAS-I

- 將任意個原始警訊檔上傳到運行 ICAS-I 演算法的 Hadoop 檔案系統空間 ( HDFS )
- 利用 Hadoop 的 MapReduce 平台架構所設計的演算法來分析資料
- 分析完後的資料塞入分散式資料庫 HBase 內

# ICAS-I 流程圖



**Intrusion  
Detectoin  
System**



**HDFS**



**JobTracker**



**hadoop**

**Cloud Platform**



**HBASE**

**Database**

# ICAS-I 整合後的警訊結果

Destination IP	Attack Signature	Source IP	Destination Port	Source Port	Packet Protocol	Timestamp
Host_1	Trojan	Sip1	80	4077	tcp	T1
Host_1	Trojan	Sip2	80	4077	tcp	T2
Host_1	Trojan	Sip1	443	5002	tcp	T3
Host_2	Trojan	Sip1	443	5002	tcp	T4
Host_3	D.D.O.S	Sip3	53	6007	udp	T5
Host_3	D.D.O.S	Sip4	53	6008	tcp	T5
Host_3	D.D.O.S	Sip5	53	6007	udp	T5
Destination IP	Attack Signature	Source IP	Destination Port	Source Port	Packet Protocol	Timestamp



Key		Values				
Host_1	Trojan	Sip1,Sip2	80,443	4077,5002	tcp	T1,T2,T3
Host_2	Trojan	Sip1	443	5002	tcp	T4
Key		Values				

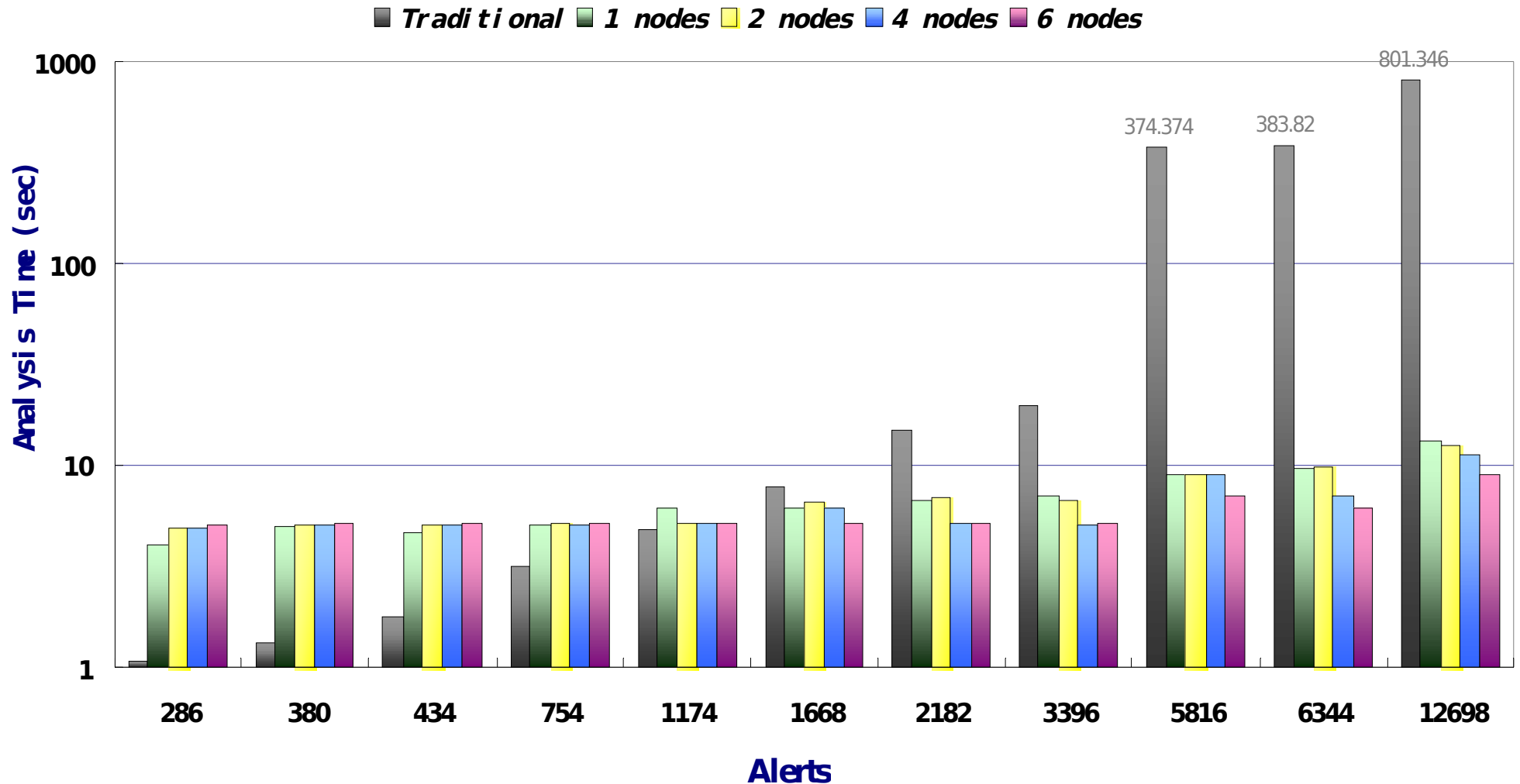


# ICAS-I 效能數據的環境

- Machine:
  - CPU : Intel quad-core, Memory : 2 GB,
- OS : Linux : Ubuntu 8.04 server
- Software : version
  - Hadoop : 0.16.4
  - Hbase : 0.1.3
  - Java : 6
- Alerts Data Sets
  - MIT Lincoln Laboratory, Lincoln Lab Data Sets
  - Computer Security group at UC Davis, tcpdump file

# ICAS-I 效能分析時間圖

## The Consuming Time of Each Number of Data Sets



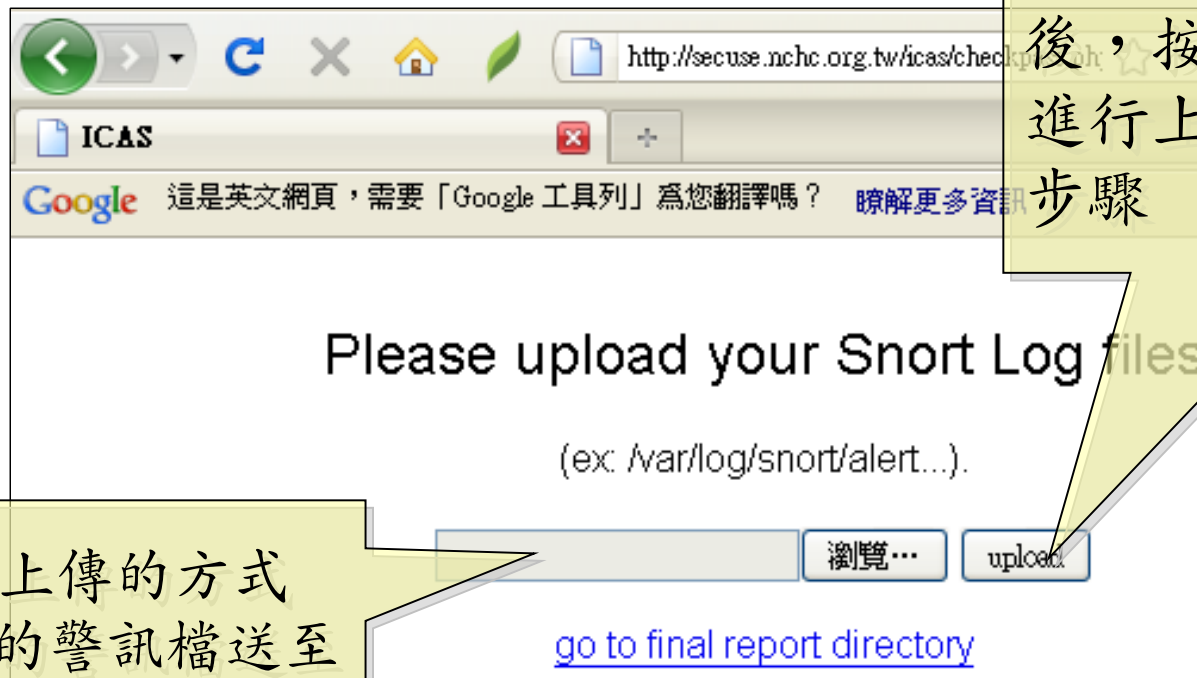
# ICAS-I 效能數據表

## Throughput Data Overall

Original Alerts	Analysis Time (sec)					Results	Reduction Rate
	Traditional	1 nodes	2 nodes	4 nodes	6 nodes		
286	1.068	4.087	4.869	4.864	5.077	30	89.51%
380	1.333	4.94	5.069	5.067	5.097	11	97.11%
434	1.76	4.61	5.066	5.068	5.09	9	97.93%
754	3.145	5.066	5.079	5.038	5.096	16	97.88%
1174	4.73	6.066	5.093	5.089	5.097	33	97.19%
1668	7.909	6.07	6.56	6.071	5.082	16	99.04%
2182	14.949	6.671	6.95	5.166	5.088	16	99.27%
3396	19.901	7.053	6.654	5.076	5.091	68	98.00%
5816	374.374	9.081	9.076	9.07	7.076	66	98.87%
6344	383.82	9.68	9.872	7.069	6.069	72	98.87%
12698	801.346	13.096	12.367	11.367	9.083	36	99.72%

# ICAS-II

- ICAS-I 僅將資料塞入資料庫，然而還是文字的敘述
- ICAS-II 將輸入的任意多個警訊整合成一張警訊關聯圖
- 資料的來源可以透過以下兩種方式上傳到分析平台
  - 系統自動設定以 SCP 傳送到 ICAS 工作目錄
  - 管理者透過 ICAS 網頁上傳

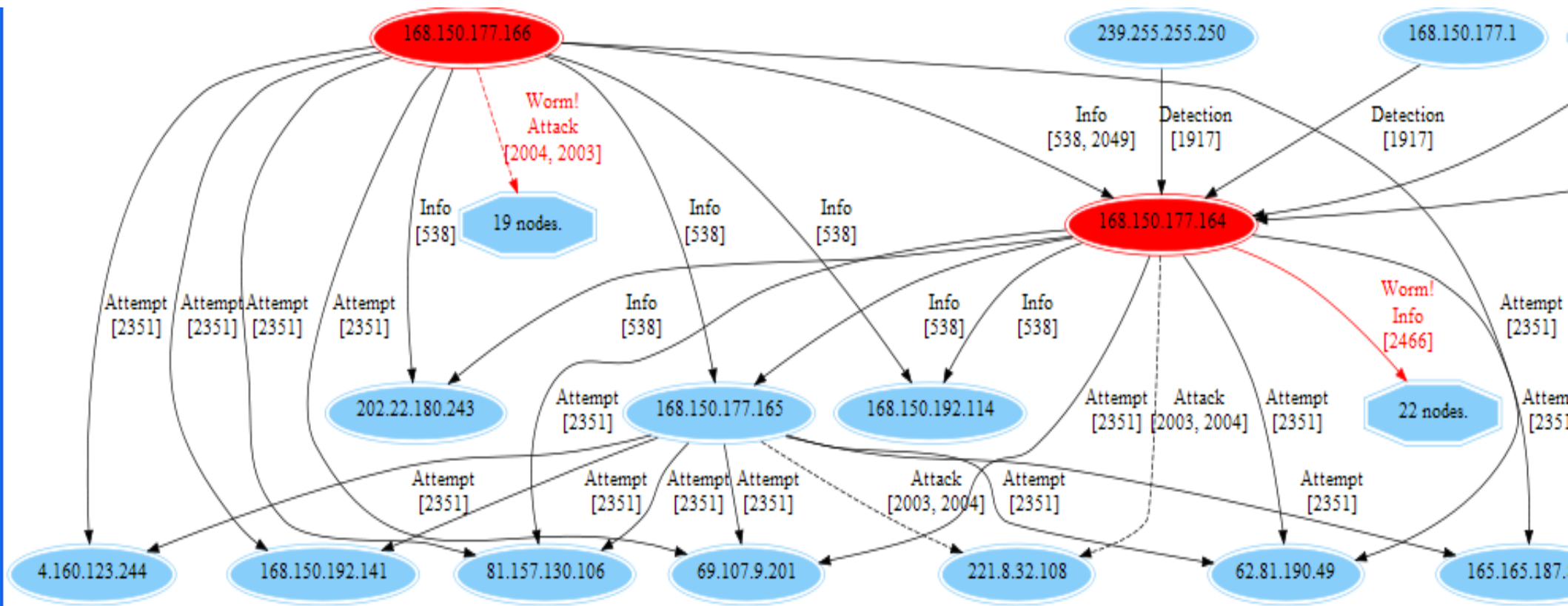


一旦選定需分析的日誌檔後，按下『上傳』，系統進行上傳→分析→繪圖等步驟

透過網頁上傳的方式將 Snort 的警訊檔送至 ICAS 分析

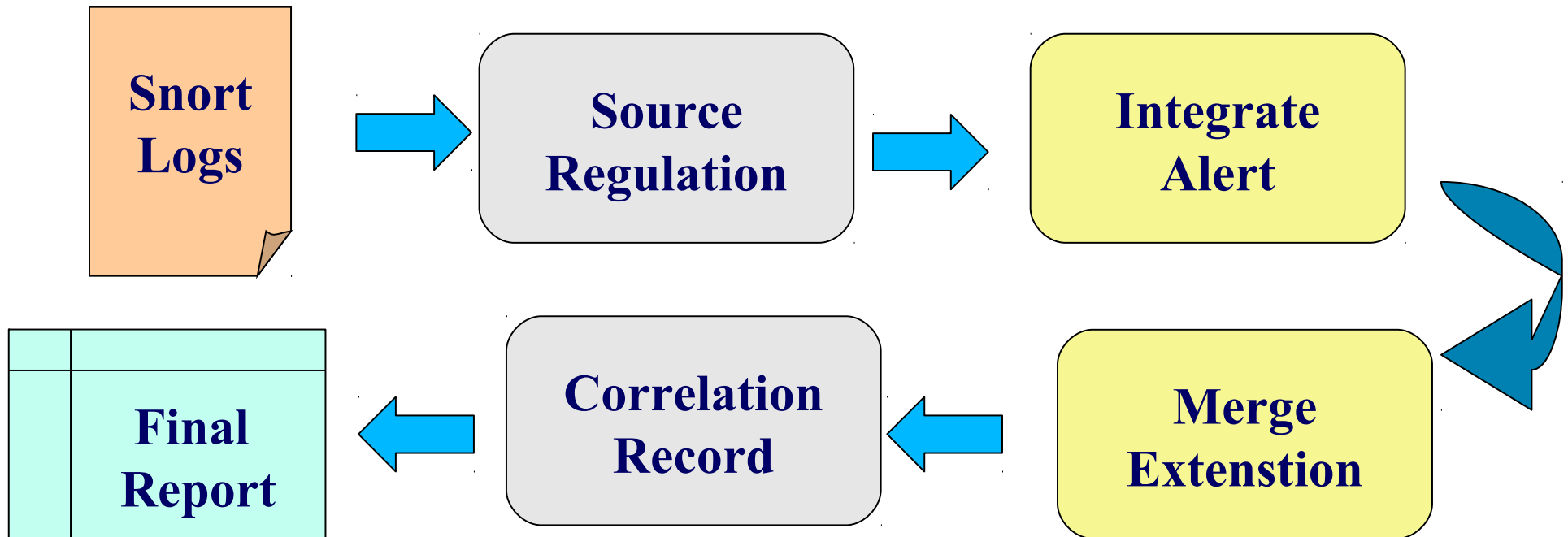
# ICAS-II 所產生的報表：警訊關聯圖

- 經過 ICAS-II 分析後，可以得到此警訊關聯圖。
- 圖中橢圓形代表節點，箭頭及線上文字代表攻擊方向與攻擊方法。
- 標為紅色則是經過系統分析之後，被判定有攻擊行為的節點與方法。
- 此圖說明 IP 168.150.177.166 與 168.150.177.164 有進行蠕蟲的攻擊行為



# ICAS-II 的分析流程

- Hadoop v 0.20



# ICAS-II 結論

- ICAS-II 可經過警訊的來源、目的、攻擊事件綜合分析
  - 提供巨觀攻擊關聯圖來瞭解攻擊事件的始末
  - 自動透過標記顏色的方法將較高危險的事件呈現出來。
- ICAS-II 尚在整合關聯式資料庫，因此未進行數據量測

# ICAS 總結

- 雲端運算處理資料格式相似且資料量大的情況下，能展現其效益
- 提供高容錯率、低獨占系統資源、多工作同時執行等能力
- 可搭配其他軟體作即時的警訊資料呈現，ICAS 可補充分析後資料的部份
- 未來工作
  - 整合多種資料來源平台
  - 產生更詳細與人性化的分析資料





National Applied Research Laboratories

National Center for High-Performance Computing

# 快速佈署叢集式搜尋引擎



陳威宇 郭文傑 楊順發

{ waue, rock , shunfa }@nchc.org.tw

國家高速網路與計算中心

自由軟體實驗室

*August 14~15, 2010*

*COSCU 2010*

# 建立專屬的搜尋引擎?

- 我的需求是...?

- 公司內部資訊網站為避免機密資料外流，需要建立內部搜尋引擎
- 個人文件可以透過自己的搜尋引擎建立索引
- 厭倦了商業搜尋引擎的搜尋結果(ex：廣告資訊)
- 某些情況下需花費許多時間過濾無意義的資訊
- 自己建一個搜尋引擎好像蠻酷的

# 建立專屬的搜尋引擎?

- 工具選擇

- 商業軟體 V.S. 自由軟體

	Goole企業版( <u>客戶</u> )	自由軟體
收費標準	18,000美金	免費!!! 元/公司
程式碼	不提供	稍高
系統操作技術門檻	較低	需自己修改程式碼
客製化	需洽相關技術人員	成本較低
系統維護成本	年費另計	



# 使用 Open Source 建搜尋引擎

- 優點

- 透明

- 擴充

- 隱私

- 客製化

免費!!!





# 我要選擇開源軟體建搜尋引擎!!!

NutchTutorial - Nutch Wiki - Mozilla Firefox

檔案 (E) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (I) 說明 (H)

http://wiki.apache.org/nutch/NutchTutorial

最常瀏覽 新手上路 即時新聞

waue/2009/0409 - Cloud Computing NutchTutorial - Nutch

Nutch Wiki 登入

NutchTutorial

首頁 最新更新 尋找頁面 說明 NutchTutorial

唯讀頁面 關於 附件 更多功能

## Requirements

1. Java 1.4.x, either from Sun or IBM on Linux is preferred.
2. Apache's Tomcat 5.x or higher.
3. On Win32, cygwin, for shell support. (If you plan to use cygwin, you must have a shell that supports the 'set' command.)
4. Up to a gigabyte of free disk space, a high-speed internet connection, and a good web browser.

## Getting Started

First, you need to get a copy of the Nutch code. You can get it from the Nutch source code directory. Or, check out the latest source code from subversion.

Try the following command:

```
bin/nutch
```

This will display the documentation for the Nutch command.

Good! You are almost ready to crawl. You need to give the crawler some instructions. You can do this by editing the configuration file.

1. Open up \$NUTCH\_HOME/conf/nutch-default.xml
2. Search for http.agent.name, and give it a value.
3. Optionally you may also set http.agent.url.

Now we're ready to crawl. There are two approaches to crawling:

1. Using the command-line to perform all the crawling.

完成

## 前言 環境

step 1 安裝好Hadoop叢集

step 2 下載與安裝

2.1 下載 nutch 並解壓縮

2.2 部署hadoop,nutch目錄結構

step 3 編輯設定檔

3.1 hadoop-env.sh

3.2 hadoop-site.xml

3.3 nutch-site.xml

3.4 slaves

3.5 crawl-urlfilter.txt

3.6 regex-urlfilter.txt

3.7 整個移植到另一台node

step 4 執行nutch

4.1 編輯url清單

4.2 上傳清單到HDFS

4.3 執行nutch crawl

step 5 瀏覽搜尋結果

5.1 安裝tomcat

5.1 tomcat server設定

5.3 下載crawl結果

5.4 設定nutch的搜尋引擎頁面到tomcat

5.5 設定搜尋引擎內容的來源路徑

5.6 啟動tomcat

step 6 享受結果

nutch

搜尋 標題 內文

0.9 requires Sun JDK 1.5 or higher.

you install, in the "Devel" category.)

Inpack the release and connect to its top-level

# 開始安裝

- 費了一番功夫後…
  - 技術文件有看沒有懂
  - 照著安裝步驟後安裝完了
  - 接下來呢…?
  - 如何驗證是否安裝成功?
  - 如何使用?
  - 怎麼解決錯誤訊息?
  - .....
  - .....
  - .....

# 糗大了 因為...

- 我想自己建搜尋引擎 又不想花錢 可是...
  - 技術文件看不懂
  - 不會寫程式
  - 系統維護是個問題

The logo for Crawlzilla is centered on the slide. It features a stylized spider with large, white, circular eyes and a black body, positioned on a yellow spiderweb. Below the spider, the word "CRAWLZILLA" is written in a large, bold, yellow font with a thick orange outline. The spiderweb is drawn with thin yellow lines.

三個願望 一次滿足

# Crawlzilla !?

## • Crawlzilla 簡介

- 於2009推出實驗版，僅提供單機版本及基本功能
- 感謝約2,800次下載量的支持並提供寶貴意見
- 於2010加入叢集版本並命名為 Crawlzilla
- 提供簡單安裝及操作管理介面，輕鬆建立搜尋引擎的  
套件工具
- 提供索引庫瀏覽功能，即時了解爬取結果



# Crawlzilla

- **Crawlzilla 提供以下特色功能：**

- 簡易安裝
- 支援叢集運算及顧全安全性
- 支援多工網頁爬取
- 網頁管理
- 支援同時存在多個搜尋引擎
- 即時瀏覽資料庫資訊
- 支援中文分詞功能
- 解決中文亂碼及中文支援
- 多種語言
- 適用於泛Linux平台

# Crawlzilla

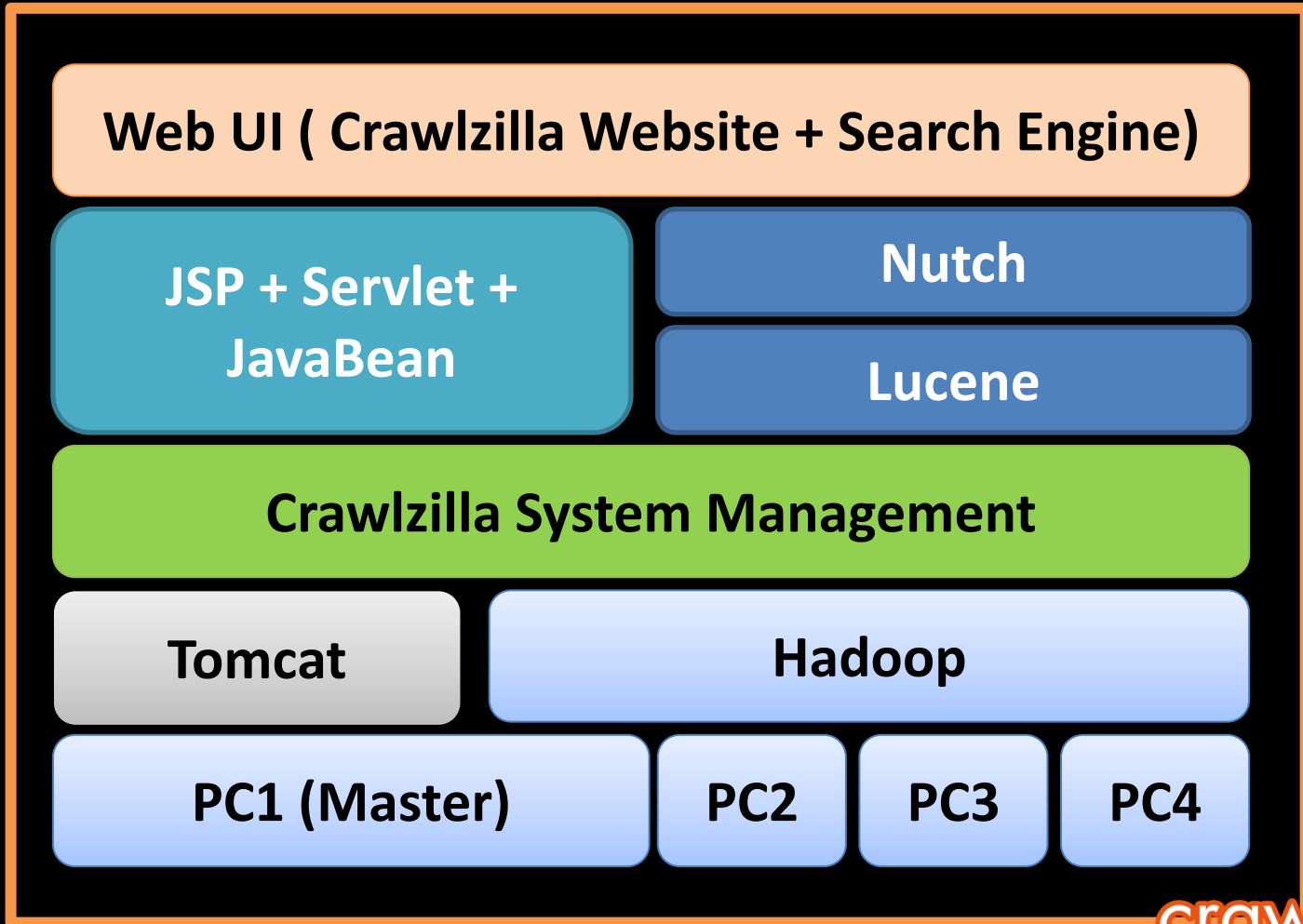
- Crawlzilla 組成

- 系統
- 管理
- 執行

- 開發者註解

- 目前Crawlzilla仍處開發階段，若系統出現不正常反應屬自然現象，不需要太恐慌並將問題回報給開發者改善後即可正常使用。

# Crawlzilla 系統架構



# Crawlzilla – 叢集系統需求!?

- 嫌一台電腦跑不夠快!?
- 閒置電腦太多!?
  - 讓多台電腦分工合作運算
- 架設叢集環境很麻煩?
  - 啟動Crawlzilla懶人叢集模式
    - 輕鬆安裝及移除
    - 可動態增加及刪除運算主機
    - 即時檢視叢集運算主機狀態



# Crawlzilla 中文分詞

- 中文分詞是...?

- 以中英文字詞結構解釋：

- 英文以”詞”為文字的基本單位
- 中文以”字”為文字的基本單位

- 例：

- 英文 → I am a good student.
- 中文(無分詞時) → 我 | 是 | 一 | 位 | 好 | 學 | 生
- 中文(加入分詞) → 我 | 是 | 一位 | 好學 | 學生



# Crawlzilla 加入中文分詞效果



簡介 常見問題

電影

Search help

Hits **1-11** (out of about **43** total matching pages):

[從《喜羊羊》到《唐山》中國電影 打造自己的阿凡達](#)

... 羊》到《唐山》中國電影 打造自己的阿凡 ... 童的心。中國電 ...

<http://www.cw.com.tw/article/index.jsp?id=41547> (cached) (explain) (anchors) (more from www.cw.com.tw)

[開心的城市就該有人跳舞 電影](#)

... 相關關鍵字: 電影 看了幾部街 ... 的街舞 (與街舞電 ...

<http://www.cw.com.tw/article/index.jsp?id=37344> (cached) (explain) (anchors) (more from www.cw.com.tw)

<http://blog.xuite.net/cwbook/blog>

... 台灣、娜娜、319鄉、電影、旅遊、美食、設計 ...

<http://blog.xuite.net/cwbook/blog> (cached) (explain) (anchors)

[天下雜誌: 行動綠生活, 台灣不碳氣!](#)

(25615 bytes) - [View as Plain Text](#)

... 他愛看藝術電影, 高職時迷上執 ... movie (專播藝術電影 ...

[http://green.cw.com.tw/Words/wordcw15\\_1.aspx](http://green.cw.com.tw/Words/wordcw15_1.aspx) (cached) (explain) (anchors) (more from green.cw.com.tw)

[天下網路書店 - 紅皇后精神](#)

... 瑪雅預言都以電影呈現虛擬實 ...

<http://www.cwbook.com.tw/common/book.jsp?productID=4326> (cached) (explain) (anchors) (more from www.cwbook.com.tw)

<http://www.cwbook.com.tw/epaper/cwbook.jsp> (cached) (explain) (anchors) (more from www.cwbook.com.tw)



# Why ik-analyzer?

	Paoding	indict	mmseg4j	ik-analyzer
使用者自定詞庫	支援	不支援	支援	支援
每秒處理文字 (基於官方介紹)	100萬字	約26萬字	15萬字~24萬字	50萬字
相關支援文件	很少,幾乎沒有	很少, 幾乎沒有	MMSeg演算法 說明文件	提供一個詳細的使用範 例及設定說明文件
其他	提供詞庫自動 更新檢測	基於ictclas 模組開發, 分詞效率佳	實作最多分詞, 但技術仍不成 熟,待改善	針對Lucene全文檢索提 供最佳化的查詢分析器- IKQueryParser,能提 高Lucene檢索的命中率

# Crawlzilla網頁管理介面



The screenshot shows the Crawlzilla web management interface. At the top, there is a blue header with the title "CrawlZilla 網頁管理介面" and a navigation menu with buttons for "HOME", "Crawl", "資料庫管理", "系統狀態", "使用者設定", and "登出系統". Below the header, the main content area is divided into two columns. The left column, titled "管理介面功能介紹", lists several features: "抓取網頁設定", "資料庫管理", "系統狀態", "管理者設定", and "修改密碼". The right column contains a sidebar with sections for "搜尋引擎快速連結" (listing 3@books, 3@businessweekly, and 3@yahoo), "系統功能" (with a link for "修改管理者密碼"), and "相關資源" (with a link for "CrawlZilla@GoogleCode").

## CrawlZilla 網頁管理介面

**管理介面功能介紹**

- \* **抓取網頁設定** : 透過網頁介面建立搜尋引擎索引
- \* **資料庫管理** : 查詢、刪除已爬取的索引資料庫
- \* **系統狀態** : 瀏覽目前系統狀態
- \* **管理者設定** : 包含個人化與語系設定
- \* **修改密碼**

**搜尋引擎快速連結**

- CrawlZilla 搜尋引擎範例
- 3@books
- 3@businessweekly
- 3@yahoo

**系統功能**

- 修改管理者密碼

**相關資源**

- CrawlZilla@GoogleCode





# Crawlzilla網頁管理介面 – 資料庫管理

## 資料總覽

起始URL	http://iso.nchc.org.tw/document/		
本機索引路徑	/home/crawler/crawlzilla/archieve/4@nchc-document/index		
總共文字數	13593	文件檔數量	303
資料庫更新日期	Wed Aug 11 23:38:41 CST 2010	使用者名稱	crawler

## 被搜尋分析到的網址:

排序	內容	引用次數	排序	內容	引用次數
0	site:www2.nchc.org.tw	130	1	site:iso.nchc.org.tw	96
2	site:www.nchc.org.tw	39	3	site:service.nchc.org.tw	20
4	site:itf.nchc.org.tw	4	5	site:edu.nchc.org.tw	2
6	site:intra.nchc.org.tw	2	7	site:www.medicalgrid.org	1
8	site:wlanrc.nchc.org.tw	1	9	site:ecogrid.nchc.org.tw	1
10	site:volunteer.nchc.org.tw	1	11	site:www.floodgrid.nchc.org.tw	1
12	site:elib.nchc.org.tw	1	13	site:www.narl.org.tw	1
14	site:pccluster.nchc.org.tw	1	15	site:colife.nchc.org.tw	1
16	site:bioinfo.nchc.org.tw	1			

## 分析的文件型態:

排序	內容	引用次數	排序	內容	引用次數
0	type:application	202	1	type:application/pdf	200
2	type:pdf	200	3	type:html	101
4	type:text	101	5	type:text/html	101
6	type:msword	1	7	type:vnd.ms-powerpoint	1
8	type:application/vnd.ms-powerpoint	1	9	type:application/msword	1



# Crawlzilla 叢集與伺服器管理介面

檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)

[管理功能選項]

請選擇：

<b>cluster_status</b>	檢查 Cluster 狀態
cluster_setup	設定 datanode & tasktracker
server_setup	設定 namenode & jobtracker
tomcat_switch	啟動/停止/重新啟動 Tomcat
tomcat_port	更改 Tomcat port
lang_switch	更換語言
client_install	Client 安裝步驟
exit	結束

< 確定 >                      < 取消 >



# Crawlzilla 系統 Demo

- 系統安裝 (Demo Video also on [Youtube](#))





# Crawlzilla系統Demo

- 網頁操作
  - 執行網頁爬取任務
  - 資料庫管理
  - 搜索引擎展示
- 系統管理





# Q & A





# mail list

- 陳威宇      waue@nchc.org.tw
- 郭文傑      rock@nchc.org.tw
- 楊順發      shunfa@nchc.org.tw





## 相關資源

- **Crawlzilla @ Google Code Project Hosting**
  - <http://code.google.com/p/crawlzilla/>
- **NCHC Cloud Computing Research Group**
  - <http://trac.nchc.org.tw/cloud>

