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
- 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

淺談雲端運算的新趨勢

Overview the trend of Cloud Computing

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw

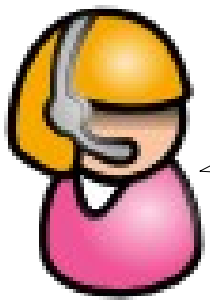
Powered by **DRBL**

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

On-demand self-service.
隨需自助服務

Broad network access
隨時隨地用任何網路裝置存取

Resource pooling
多人共享資源池

Rapid elasticity
快速重新佈署靈活度

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

Community Cloud
社群雲端

Academia 學術為主

以大型企業
為主要客戶
Enterprise is
key market

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

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，服務至上

能把 AAA 做好就很強了！

Authentication
Authorization
Accounting
as
a
Service

引用自： https://www.ibm.com/developerworks/mydeveloperworks/blogs/sbose/entry/gathering_clouds_of_xaas

2 R&D directions : Cloud or Device

兩大研究方向：你該選「雲」還是「端」？

端

symbian OS

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

Key spirit of Cloud ~

形成服務才是重點！！

Everything as a Service !!

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

Evolution of Cloud Services

雲端服務只是軟體演化史的必然趨勢

數位化 → e-Taiwan 成果系列報導 → 行動台灣 應用推動計畫 → 發展優質網路社會 (U-Taiwan) 計畫

實體 Physical	單機版 個人使用 Personal Software	網路版 多人共享 Share Service Software	行動版 隨時存取 Mobile Cloud Service
信箱 Mailbox	E-Mail	Web Mail	Mobile Mail
電視 TV	電視盒 Setop Box	Web TV Ex. Youtube	Mobile TV
打字機 Typer Writer	Office	Google Docs	M-Office
電話 Telephone	數位電話 PBX	Skype	Flash Wengo
佈告欄 Bullet Borad	電子佈告欄 BBS	部落格 Blog	微網誌 Twitter

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 1st Unix

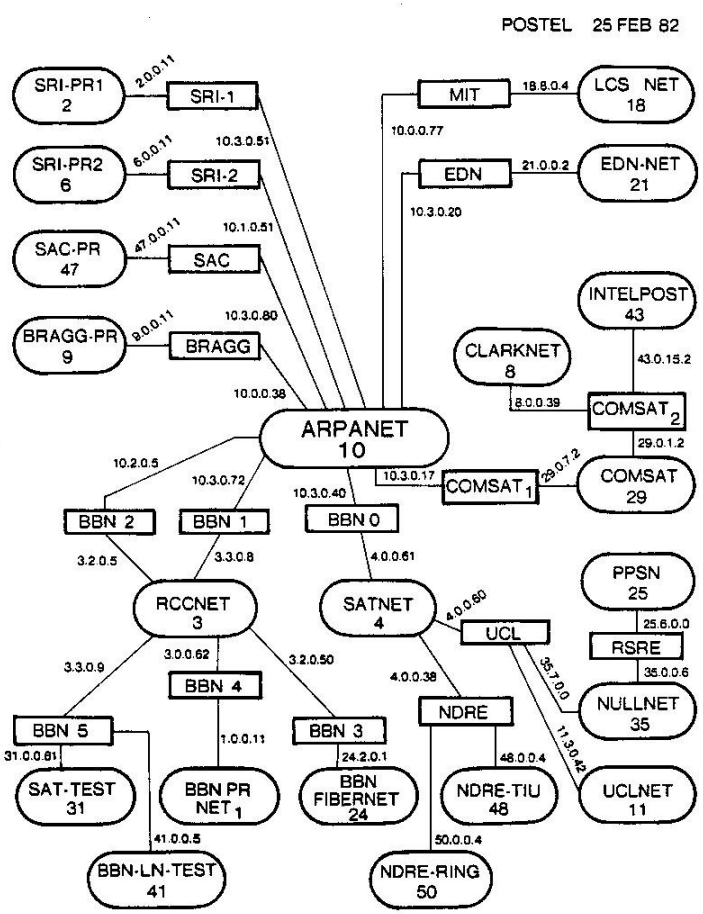
Mainframe
Super Computer

1977 Apple II

1981 IBM 1st PC 5150

Back to Year 1970s ...

1982 TCP/IP



1983 GNU

1991 Linux

Back to Year 1980s ...

Brief History of Computing (2/5)

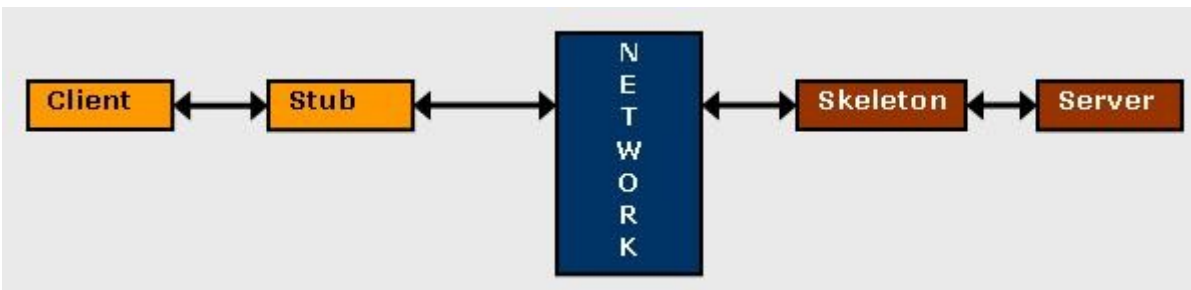
Source: http://www.nchc.org.tw

Mainframe
Super Computer

PC / Linux
Cluster Parallel

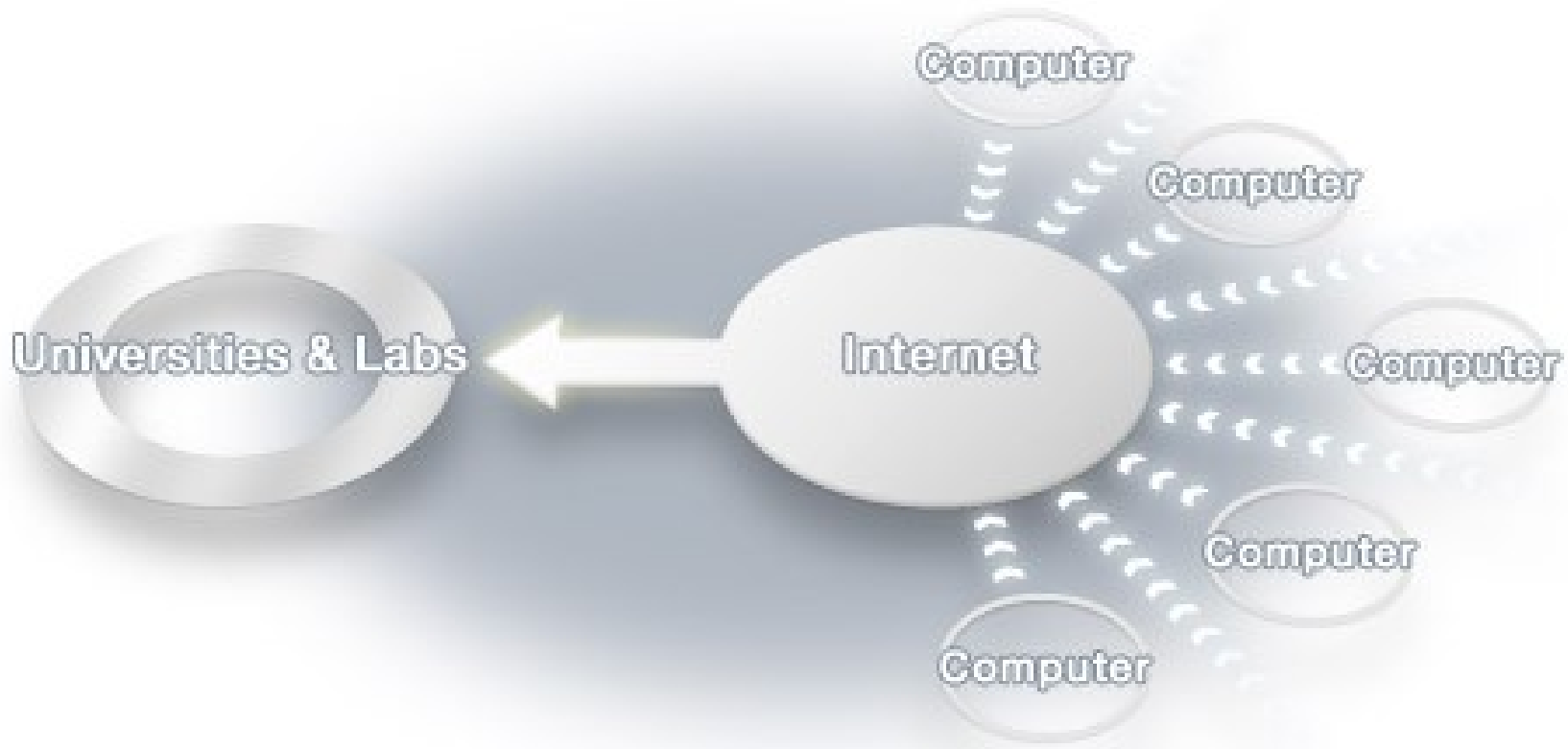
1990 World Wide Web
by CERN
...
...
1993 Web Browser
Mosaic by NCSA

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

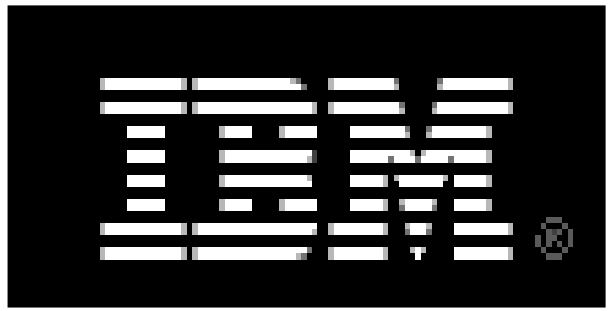
Mainframe
Super Computer

PC / Linux
Cluster Parallel

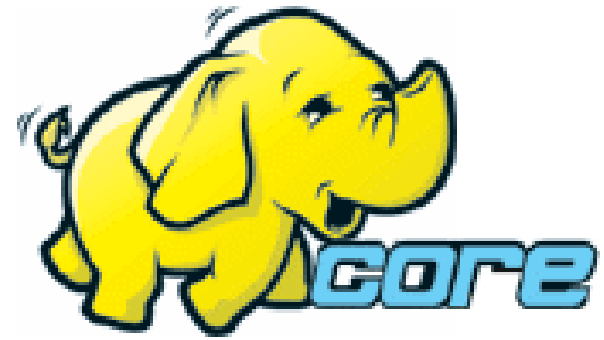
Internet
Distributed Computing

Virtual Org.
Grid Computing

2001 Autonomic Computing
IBM



2006 Apache Hadoop



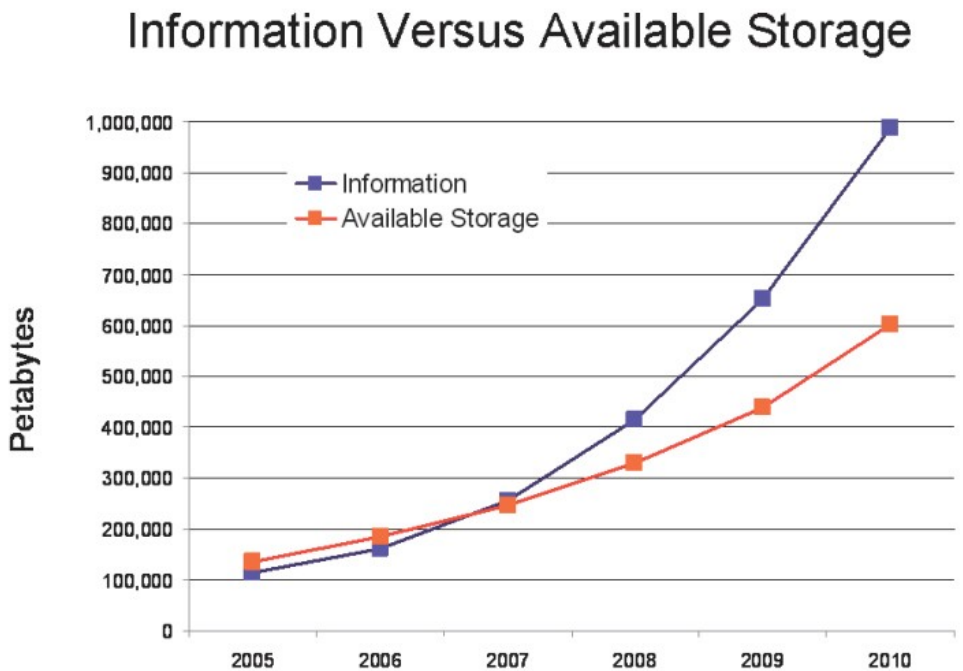
2005 Utility Computing
Amazon EC2 / S3

2007 Cloud Computing
Google + IBM

Back to Year 2007 ...

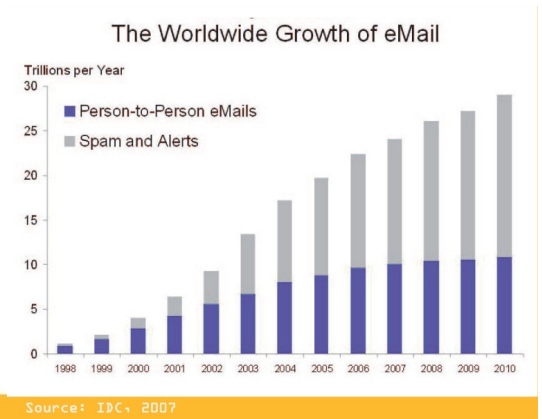
2007 Data Explore

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

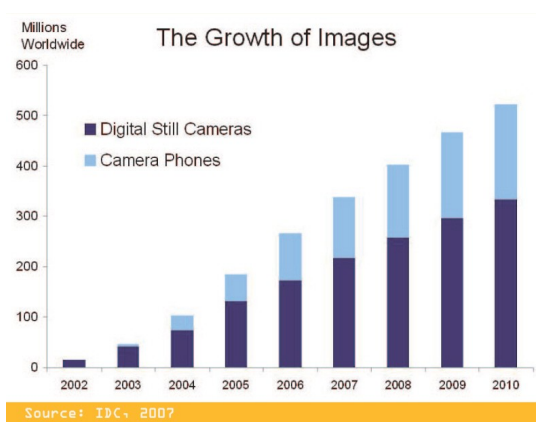


Source: http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf

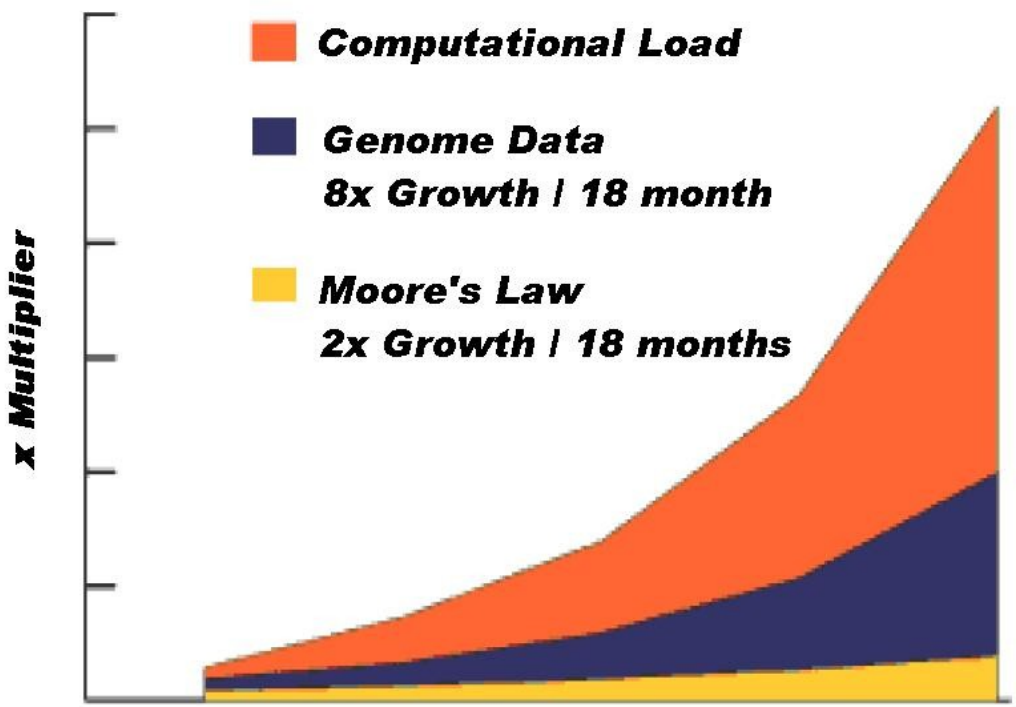
Source: IDC, 2007



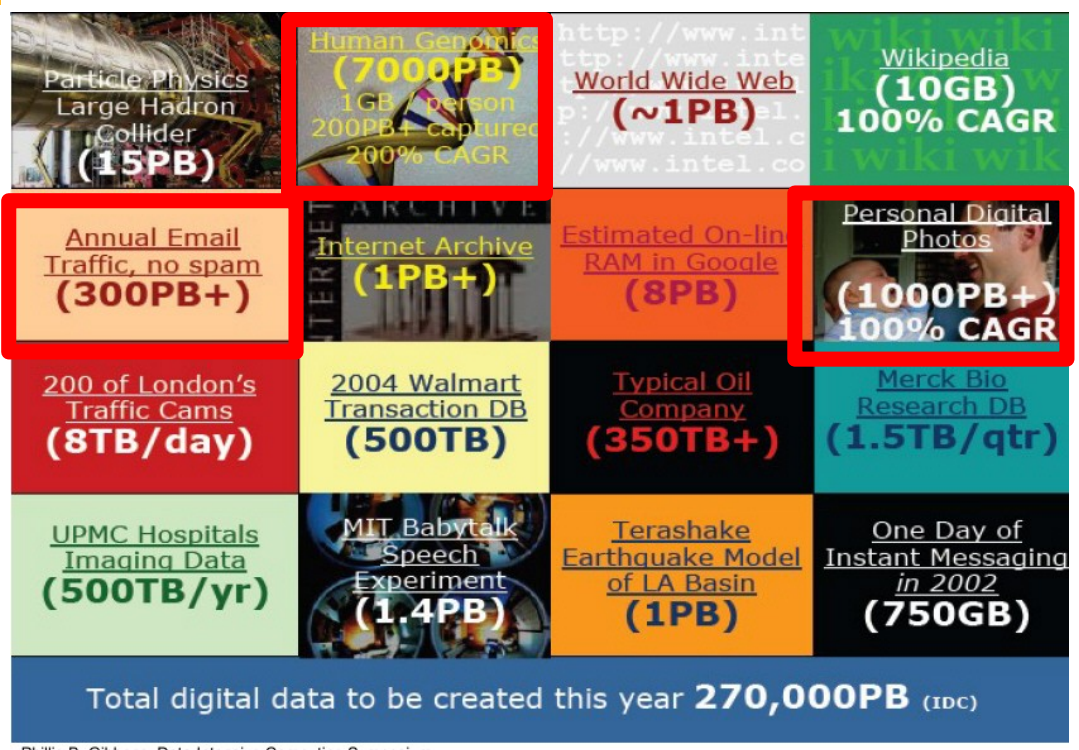
Source: IDC, 2007



Source: IDC, 2007

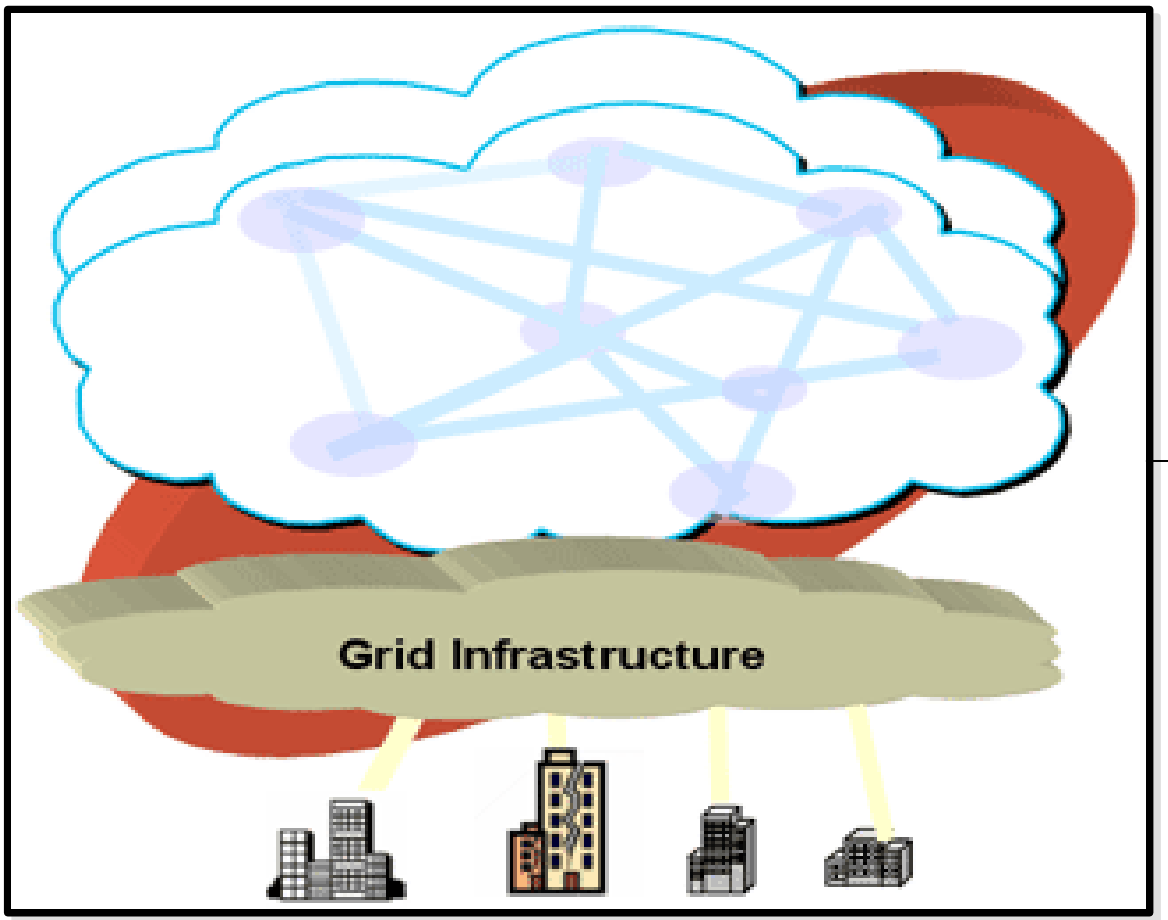


Source: http://lib.stanford.edu/files/see_pasig_dic.pdf



Phillip B. Gibbons, Data-Intensive Computing Symposium

Brief History of Computing (5|5)



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

nframe
per
nputer

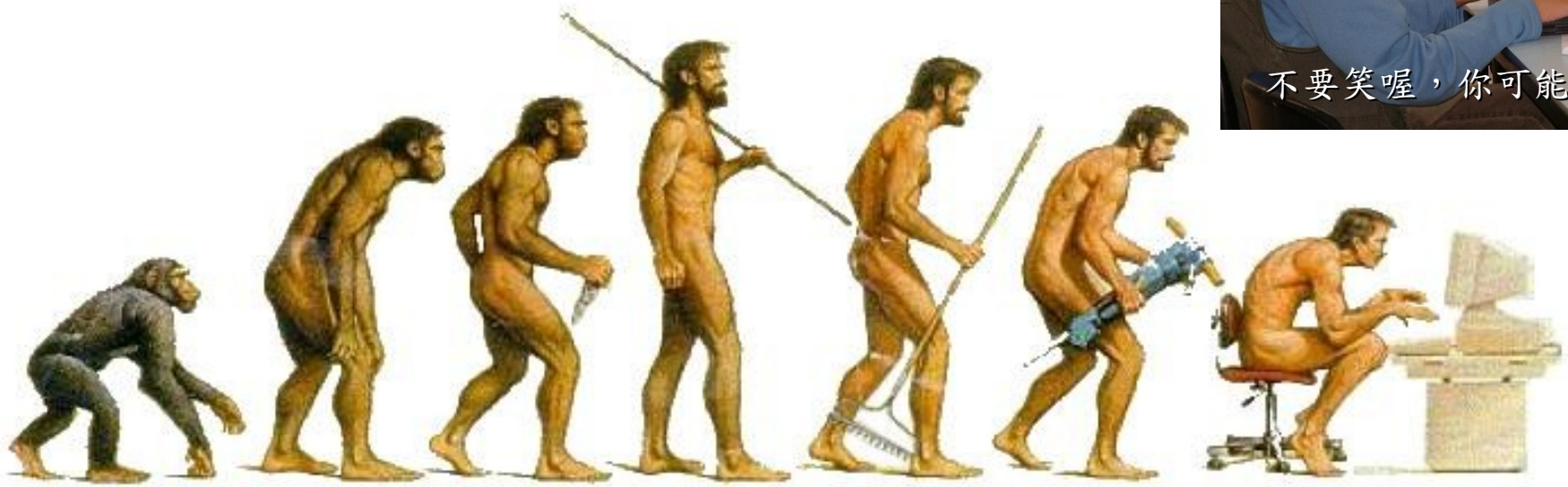
PC / Linux
Cluster
Parallel

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 ?!

Source: http://cyberpingui.free.fr/humour/evolution-white.jpg

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！

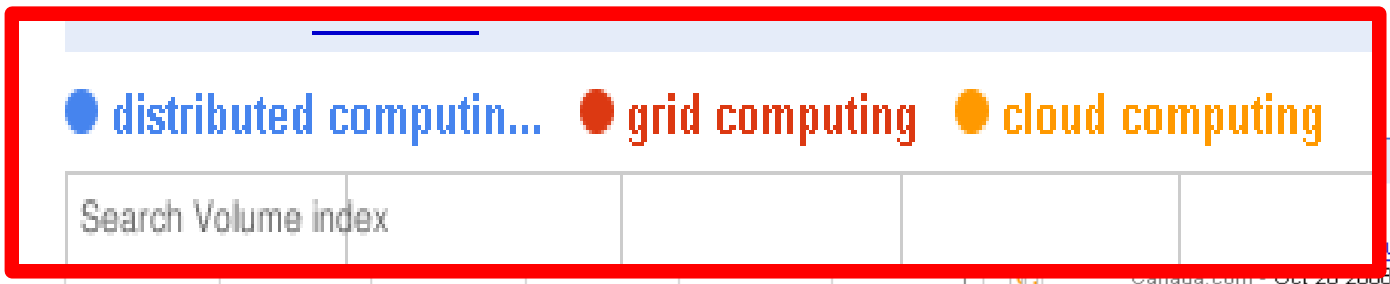
Answer #3: Total Cost of Ownership 總擁有成本

This is why Cloud Computing matters ?!

這就是為什麼雲端運算變得熱門?!

What are the trend of next 10 years ?

什麼是下個十年的熱門技能？



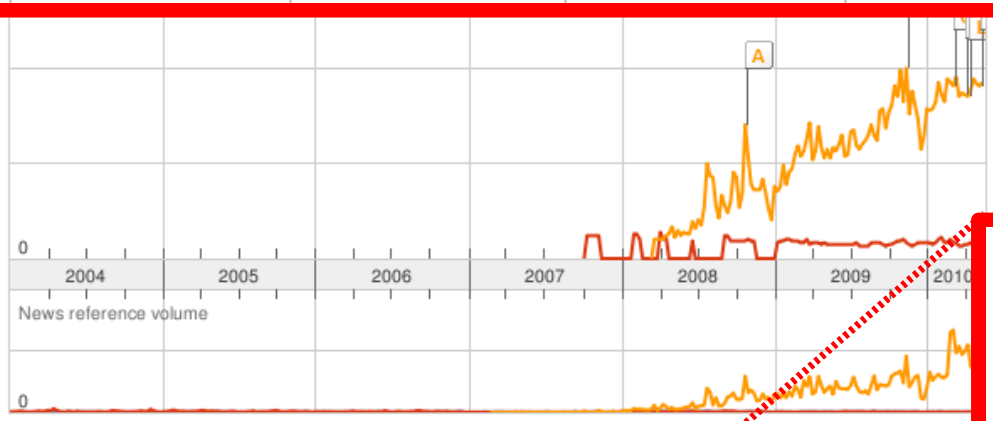
Sign in to see and export additional Tren

All regions | All years

ud" computing space

B Microsoft's cloud computing system is growing up
Philadelphia Inquirer - Nov 17 2009

C Google looks to be 'cloud-computing' rainmaker for other online business services
Winnipeg Free Press - Mar 10 2010



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 ?!**

CIO 2010 : Virtualization, Cloud and Web 2.0

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 "*

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！！

Source: 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？

Let's Talk about Public Cloud
讓我們先來談談公用雲端服務

Public Cloud
公用雲端

Microsoft

Google

Target Market is S.M.B.
主要客戶為中小企業

Hybrid Cloud

以大型企業為主要客戶
Enterprise is key market

私有雲端
Private Cloud

Community Cloud
社群雲端

Academia 學術為主

Public Cloud #1: Amazon 亞馬遜網路書店

- Amazon Web Service (AWS)
- 虛擬伺服器：Amazon EC2
  - Small (Default) $0.085 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.cisanet.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

Public Cloud #2: Google 谷歌

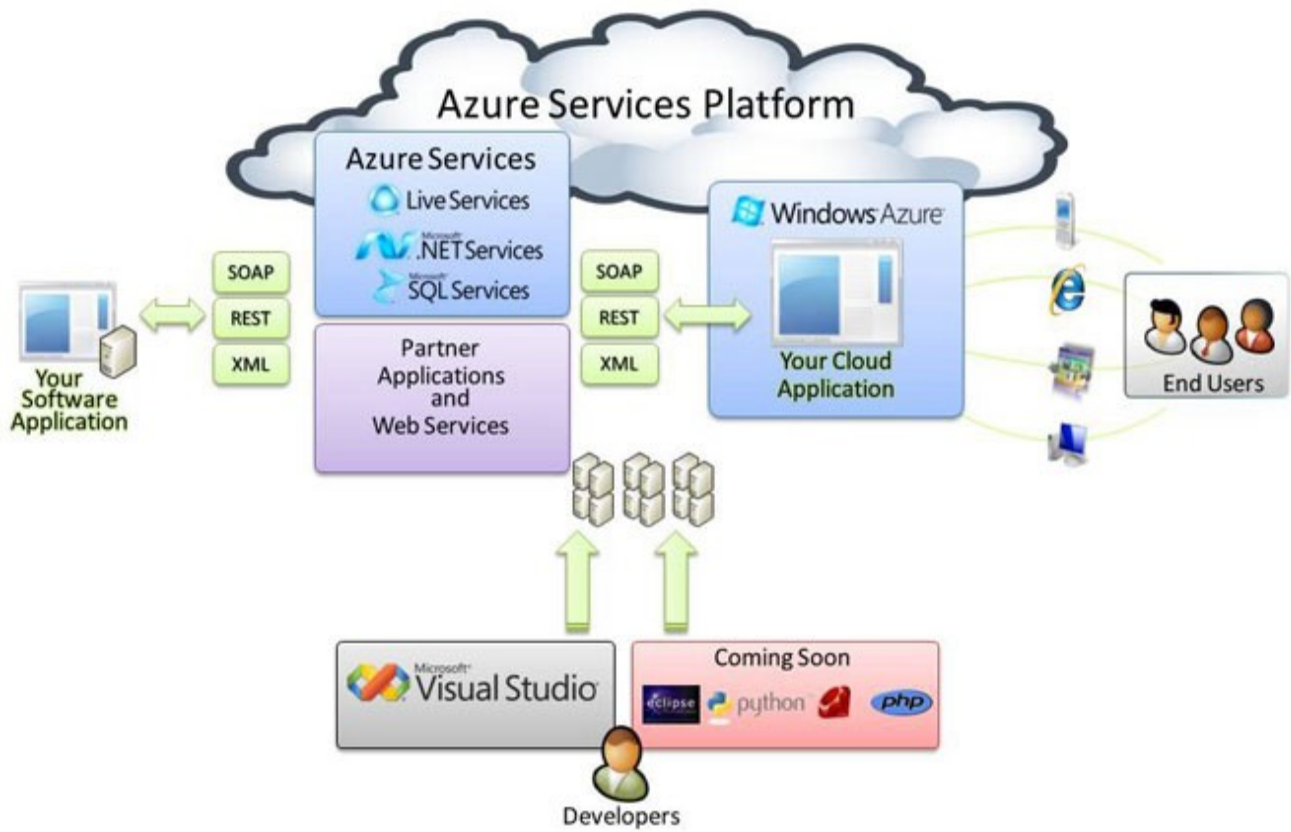
- Google App Engine (GAE)
- 讓開發者可自行建立網路應用程式於 Google 平台之上。
- 提供：
  - 500MB of storage
  - up to 5 million page views a month
  - 10 applications per developer account
- 限制：
  - 程式設計語言只能用 Python 或 Java

- 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

參考來源： http://code.google.com/intl/zh-TW/appengine/
http://code.google.com/intl/zh-TW/appengine/docs/billing.html

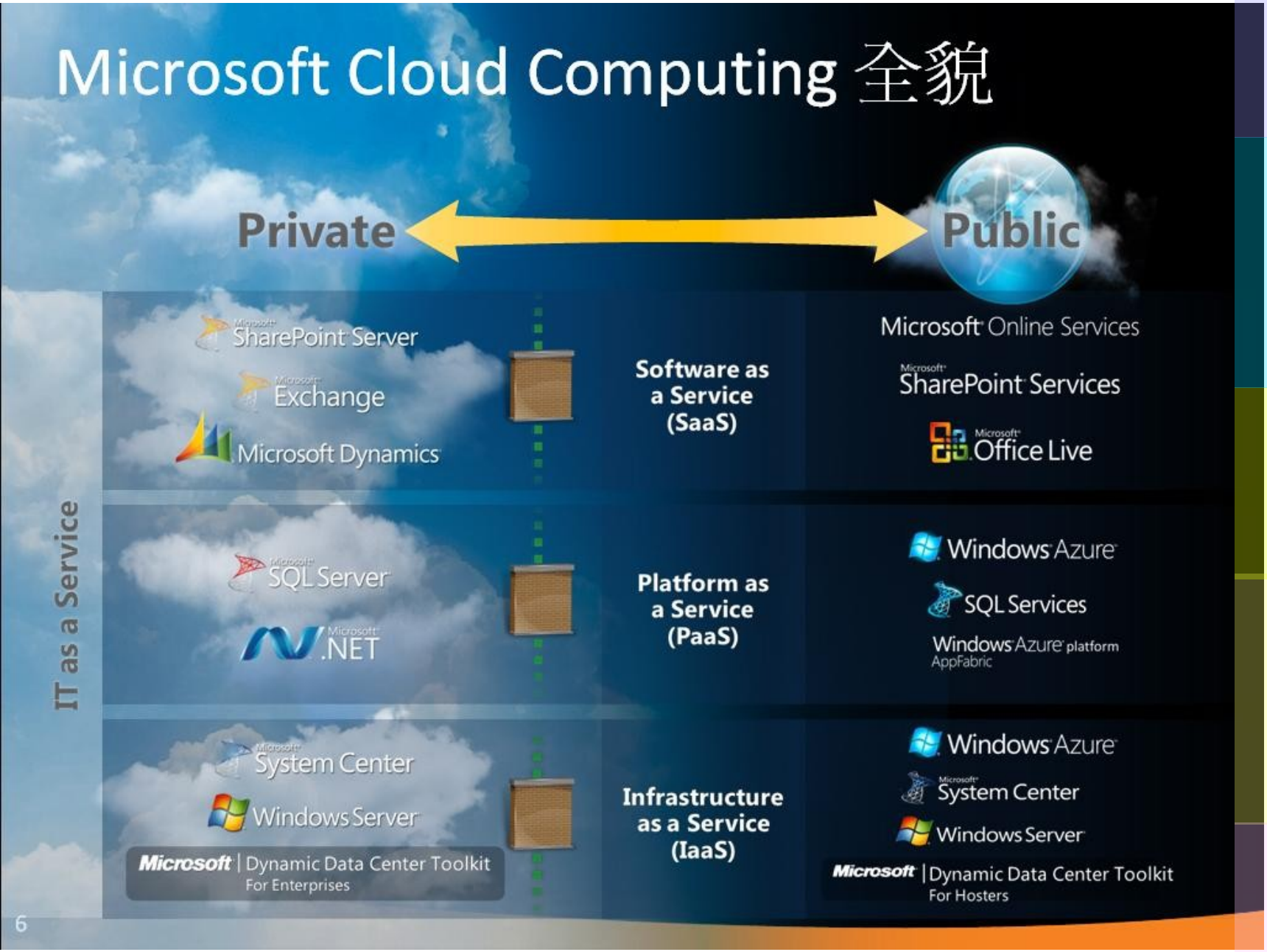
Public Cloud #3: Microsoft 微軟

- 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



參考來源： http://tech.cipper.com/index.php/archives/332

Public Cloud #3: Microsoft 微軟



Dallas
→ DaaS

Azure AppFabric
→ PaaS
(類似 GAE)

SQL Azure
→ PaaS
(雲端 SQL)

Window Azure
→ PaaS
(類似 EC2)

Hyper-V
→ IaaS
(虛擬化)

參考來源：微軟雲端運算的策略與產品藍圖，台灣微軟李匡正， 2010-02-24

Public Cloud Comparison: 公用雲端的比較

	On-Premises Apps	Small-to-Medium Web Apps	Large Web Apps	Parallel Processing Apps	Web Apps with Back-end Processing	Store Blob Data
GoGrid, Flexiscale, Others	X	X				
Amazon Web Services	X	X	X	X	X	X
Windows Azure 2009 July CTP		X	X	X	X	X
Google AppEngine			X			
Salesforce.com Force Platform			X			

25

參考來源：微軟雲端運算的策略與產品藍圖，台灣微軟李匡正， 2010-02-24

How can we build our Private Cloud ??
那我們如何打造私有雲端呢??

Public Cloud
公用雲端

Target Market is **S.M.B.**

主要客戶為**中小企業**

Hybrid Cloud

以**大型企業**為主要客戶

Enterprise is key market

私有雲端
Private Cloud

Community Cloud
社群雲端

Academia **學術**為主

Reference Cloud Architecture
雲端運算的參考架構

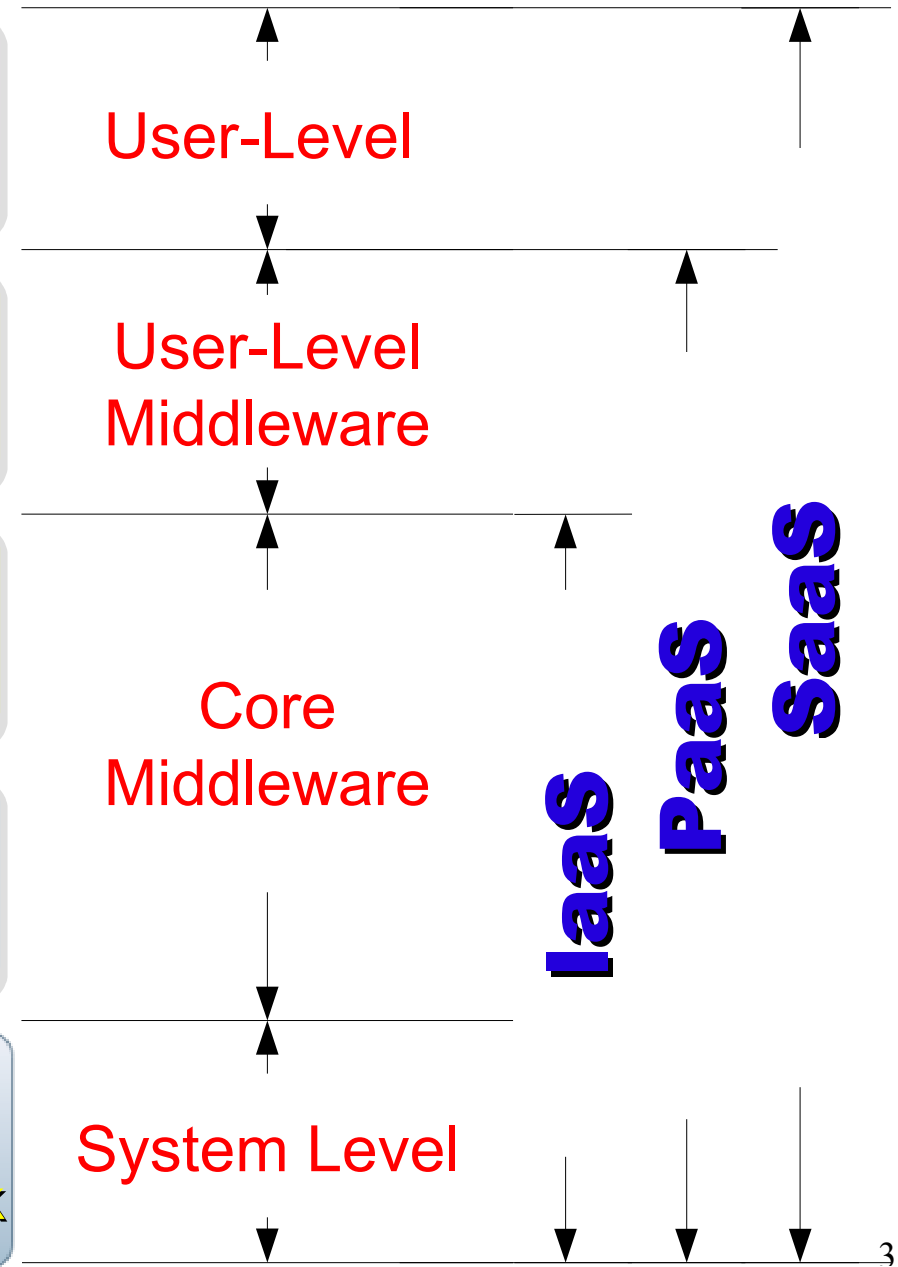
應用
Social Computing, Enterprise, ISV,…

程式語言
Web 2.0 介面, Mashups, Workflows, …

控制
Qos Neqotiation, Ddmission Control, Pricing, SLA Management, Metering…

虛擬化
VM, VM management and Deployment

硬體設施
Infrastructure: Computer, Storage, Network



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, Ddmission Control, Pricing, SLA Management, Metering…	OpenNebula, Enomaly, Eucalyptus , OpenQRM, ...
虛擬化 VM, VM management and Deployment	Xen, KVM, VirtualBox, QEMU, OpenVZ, ...

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

關於 Eucalyptus 的更多資訊，請參考
http://trac.nchc.org.tw/grid/wiki/Eucalyptus

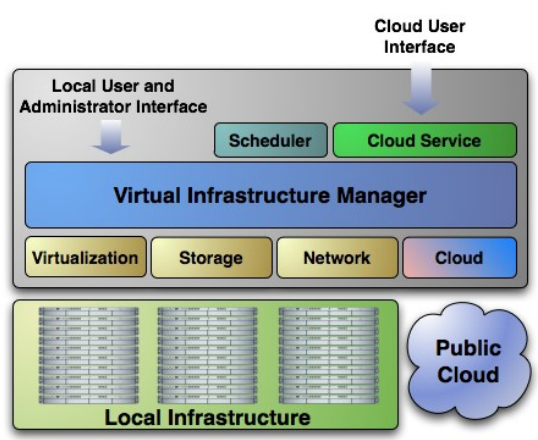
Open Cloud #2: OpenNebula

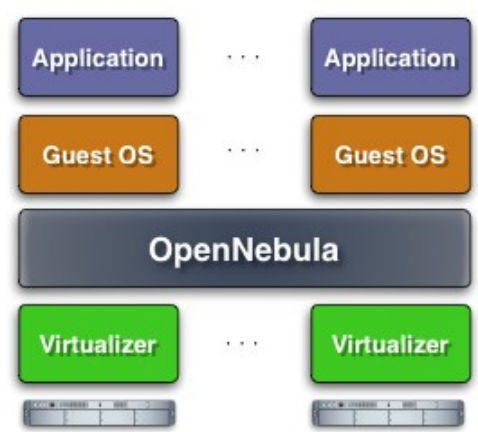
- http://www.opennebula.org
- 由歐洲研究學會 (European Union FP7) 贊助
- 將實體叢集轉換成具管理彈性的虛擬基礎設備
- 可管理虛擬叢集的狀態、排程、遷徙 (migration)
- 優勢是Ubuntu 9.04 已經收錄 OpenNebula 的套件
- 缺點：需下指令來進行虛擬機器的遷徙 (migration) 。

SEVENTH FRAMEWORK PROGRAMME

RESERVOIR

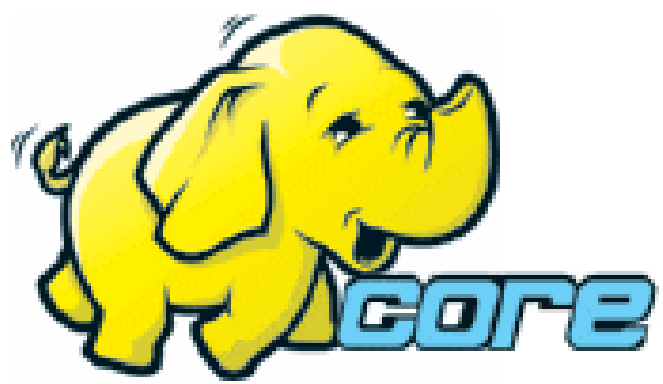
關於 OpenNebula 的更多資訊，請參考 http://trac.nchc.org.tw/grid/wiki/OpenNEbula





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。



Open Cloud #4: Sector / Sphere

- 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。

National Center for Data Mining
University of Illinois at Chicago

UIC

open data

Open Data Group
http://www.opendatagroup.com/

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, Sector/Sphere, Eucalyptus, and more ....

Questions?

Slides - http://trac.nchc.org.tw/cloud

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

Attribution-Noncommercial-Share Alike 3.0 Taiwan

creative commons

姓名標示-非商業性-相同方式分享 3.0 台灣

您可自由：

分享 — 重製、散布及傳輸本著作

重混 — 修改本著作

惟需遵照下列條件：

姓名標示 — 您必須按照著作人或授權人所指定的方式，表彰其姓名（但不得以任何方式暗示其爲您或您使用本著作的方式背書）。

非商業性 — 您不得爲商業目的而使用本著作。

相同方式分享 — 若您變更、變形或修改本著作，您僅得依本授權條款或與本授權條款類似者來散布該衍生作品。

http://creativecommons.org/licenses/by-nc-sa/3.0/tw/

These slides could be distributed by Creative Commons License.

Hadoop Overview

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)

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
  - 小工作
- JobTracker
  - 任務分派者
- TaskTracker
  - 小工作的執行者
- Client
  - 發起任務的客戶端
- Map
  - 應對
- Reduce
  - 總和

- Namenode
  - 名稱節點
- Datanode
  - 資料節點
- 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

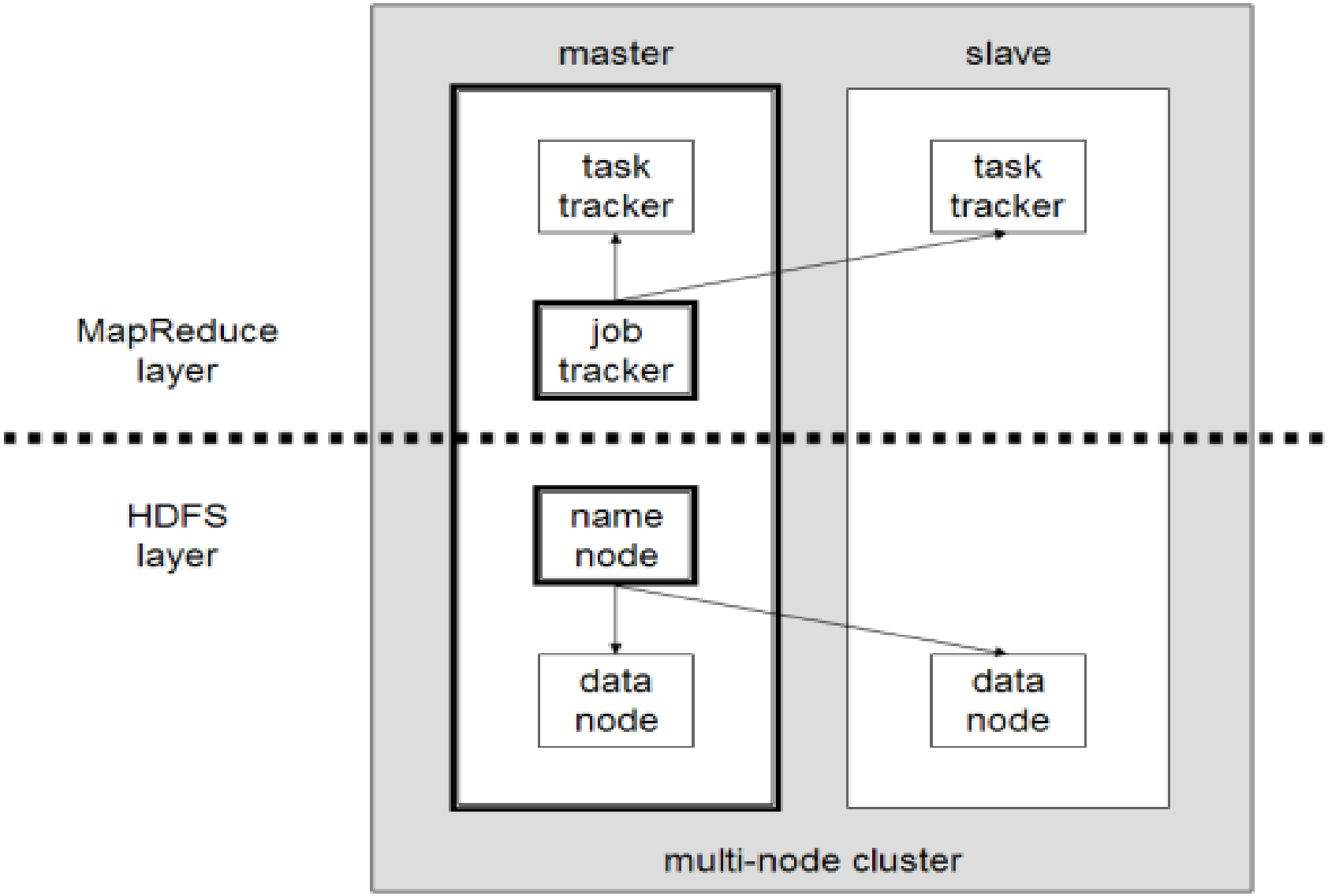
Jobtracker

- 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

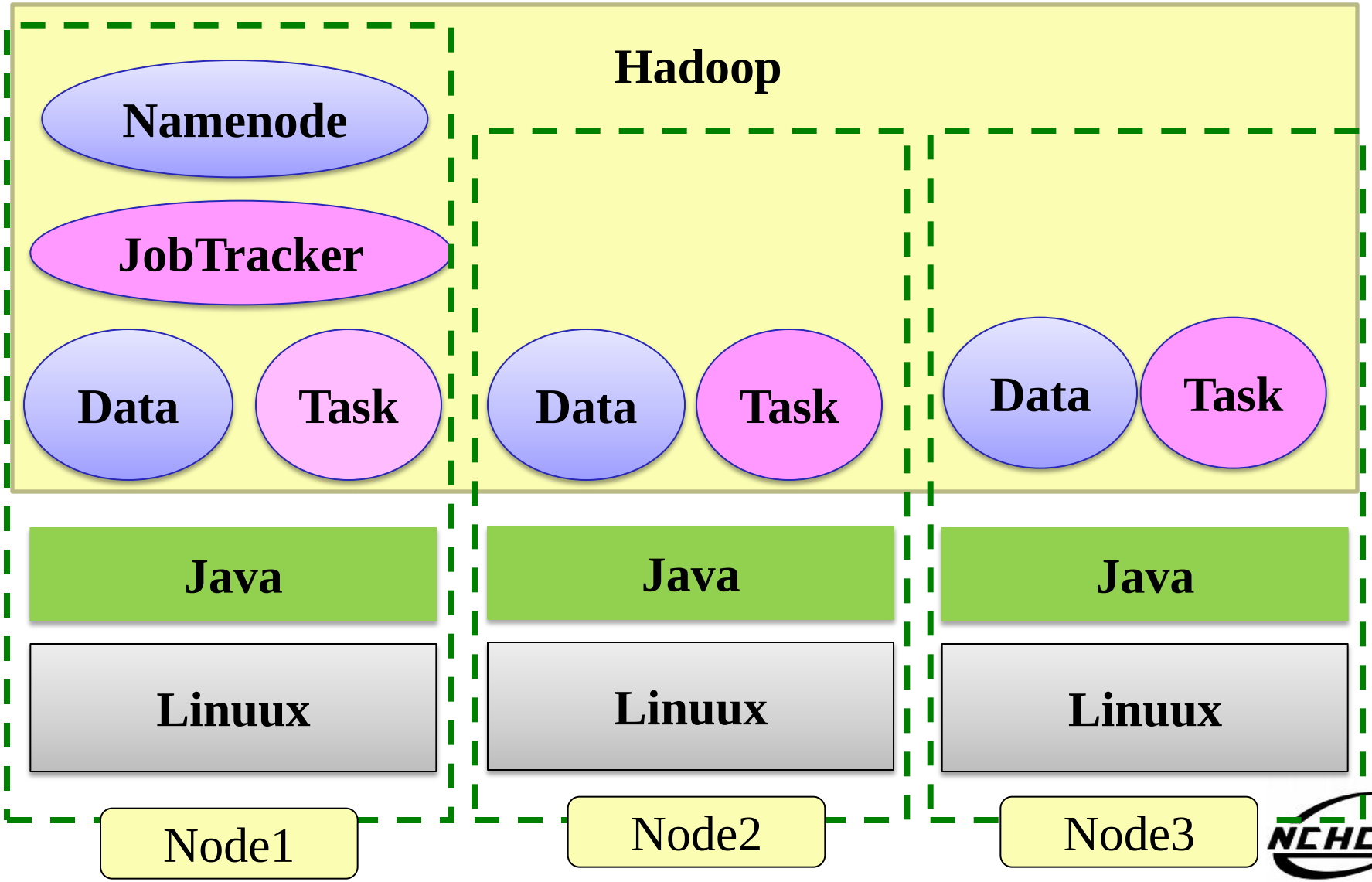
Tasktrackers

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

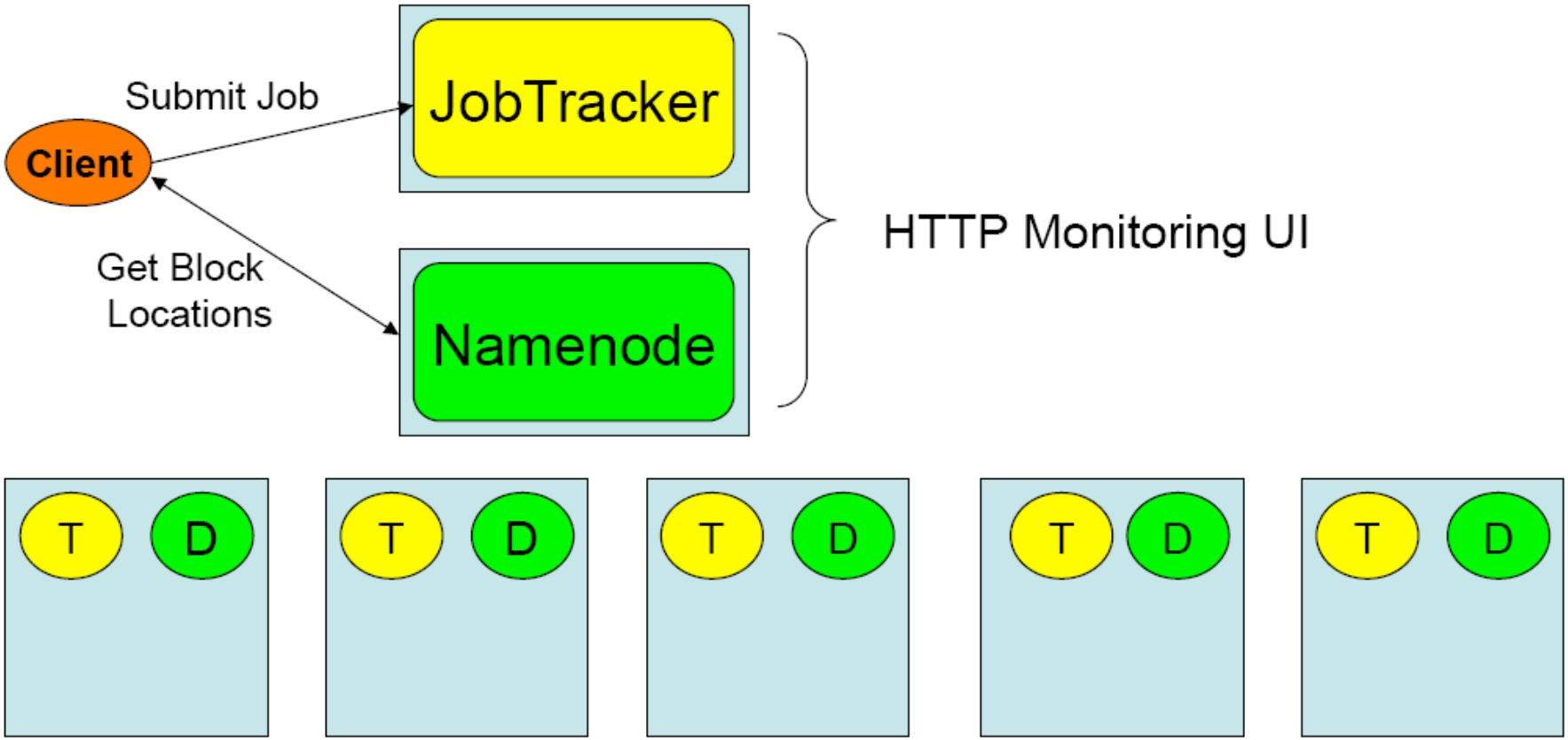
Hadoop的各種身份



Building Hadoop



不在雲裡的 Client



結論

- 所有工作都由JobTracker統一分派，由中眾多TaskTracker 執行，每個TaskTracker又可以執行多個Task threads
- 所有名稱空間與檔案的metadata都由一個Namenode統籌，檔案空間為所有Datanode的集合，hdfs的基本單位為block
- Client 只需要丟工作或存取在“雲”的資料
- 問題：
  - Hadoop為Java設計的，Java跨平台，為何Hadoop只有Linux 版本？

安裝設定補充說明

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)

Hadoop Package Topology

資料夾	說明
bin /	各執行檔：如 start-all.sh 、 stop-all.sh 、 hadoop
conf /	預設的設定檔目錄：設定環境變數 hadoop-env.sh 、各項參數 hadoop-site.conf 、工作節點 slaves 。（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： 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： jetty 、 kfs 。但主要的 hadoop 函式庫於 hadoop_home
src /	Hadoop 的原始碼。
build /	開發 Hadoop 編譯後的資料夾。需搭配 ant 程式與 build.xml
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
……
```

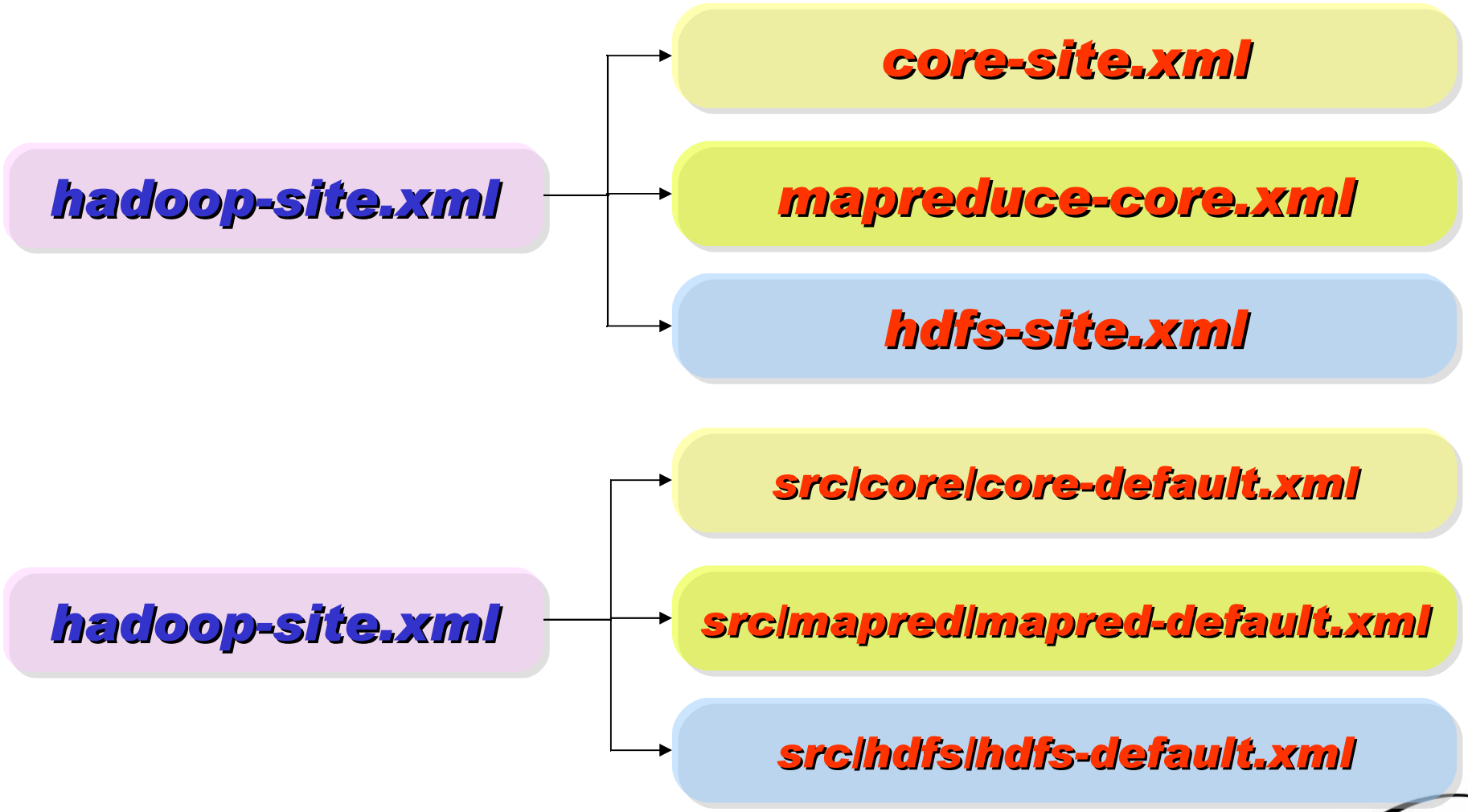
設定檔： hadoop-site.xml (0.18)

```
<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-

Hadoop 0.18 到 0.20 的轉變



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 Δ **指令** Δ 選項 Δ 參數 Δ ….

指令	用途	舉例
fs	對檔案系統進行操作	hadoopΔ**fs**Δ–putΔinΔinput
jar	啟動運算功能	hadoopΔ**jar**Δexample.jarΔwcΔinΔout
archive	封裝 hdfs 上的資料	hadoopΔ**archive**Δfoo.harΔ/dirΔ/user/hadoop
distcp	用於叢集間資料傳輸	hadoopΔ**distcp**Δhdfs://nn1:9000/aaΔhdfs://nn2:9000/aa
fsck	hdfs 系統檢查工具	hadoopΔ**fsck**Δ/aaΔ-filesΔ-blocksΔ-locations
job	操作正運算中的程序	hadoopΔ **job** Δ–kill ΔjobID
version	顯示版本	hadoopΔ**version**

Hadoop 管理者指令

$ bin/hadoop Δ **指令** Δ 選項 Δ 參數 Δ ….

指令	用途	舉例
balancer	平衡 hdfs 覆載量	hadoopΔ**balancer**
dfsadmin	配額、安全模式 等管理員操作	hadoopΔ**dfsadmin**$_{\Delta}$–setQuota$_{\Delta}$3 $_{\Delta}$/user1/
namenode	名稱節點操作	hadoopΔ**namenode**Δ-format

$ bin/hadoop Δ **指令**

datanode	成為資料節點	hadoop$_{\Delta}$datanode
jobtracker	成為工作分派者	hadoopΔ jobtracker
tasktracker	成為工作執行者	hadoopΔtasktracker

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
的定義？

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

HDFS
的特色？

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

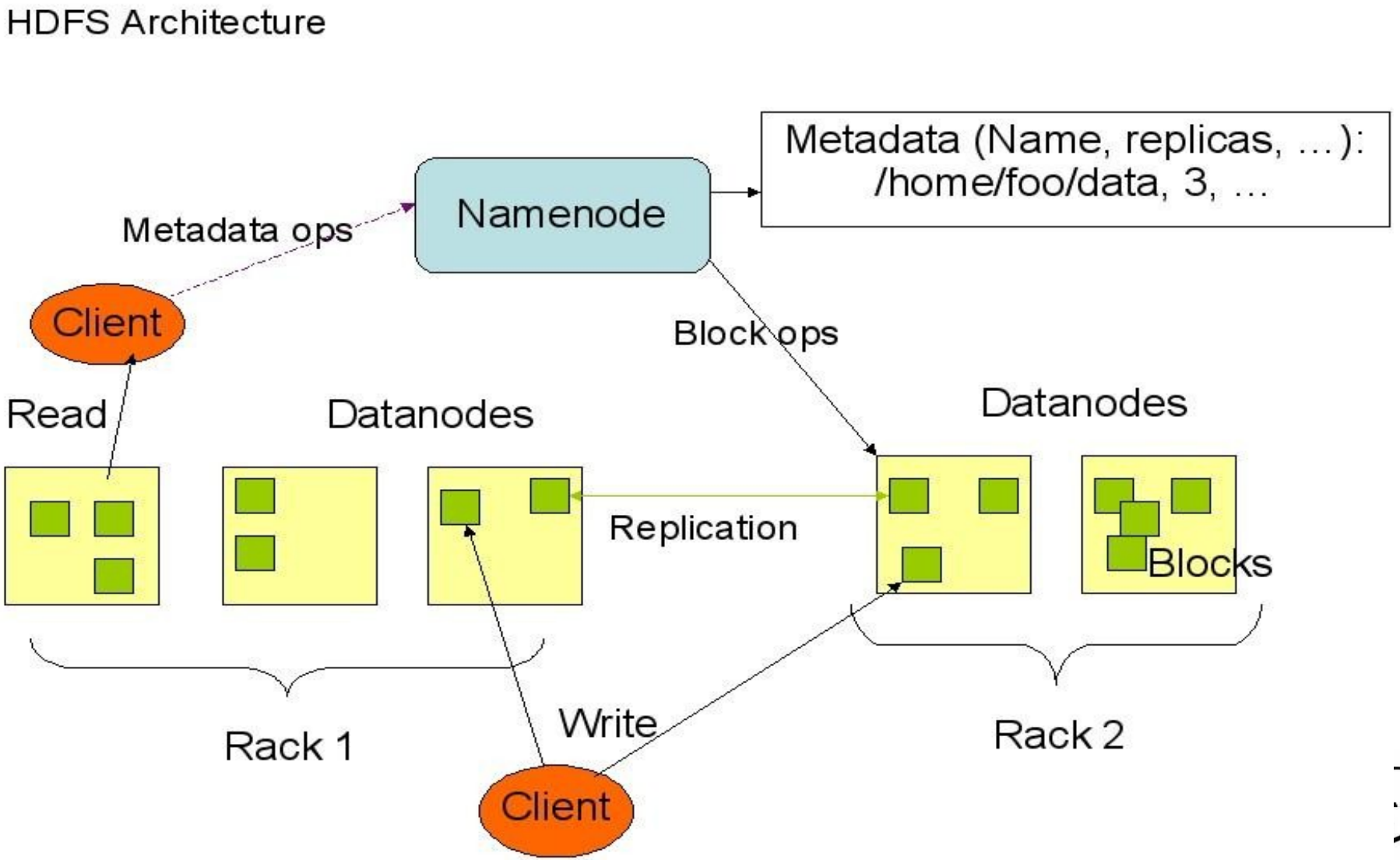
HDFS
的特色？

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

HDFS 的架構？

管理資料



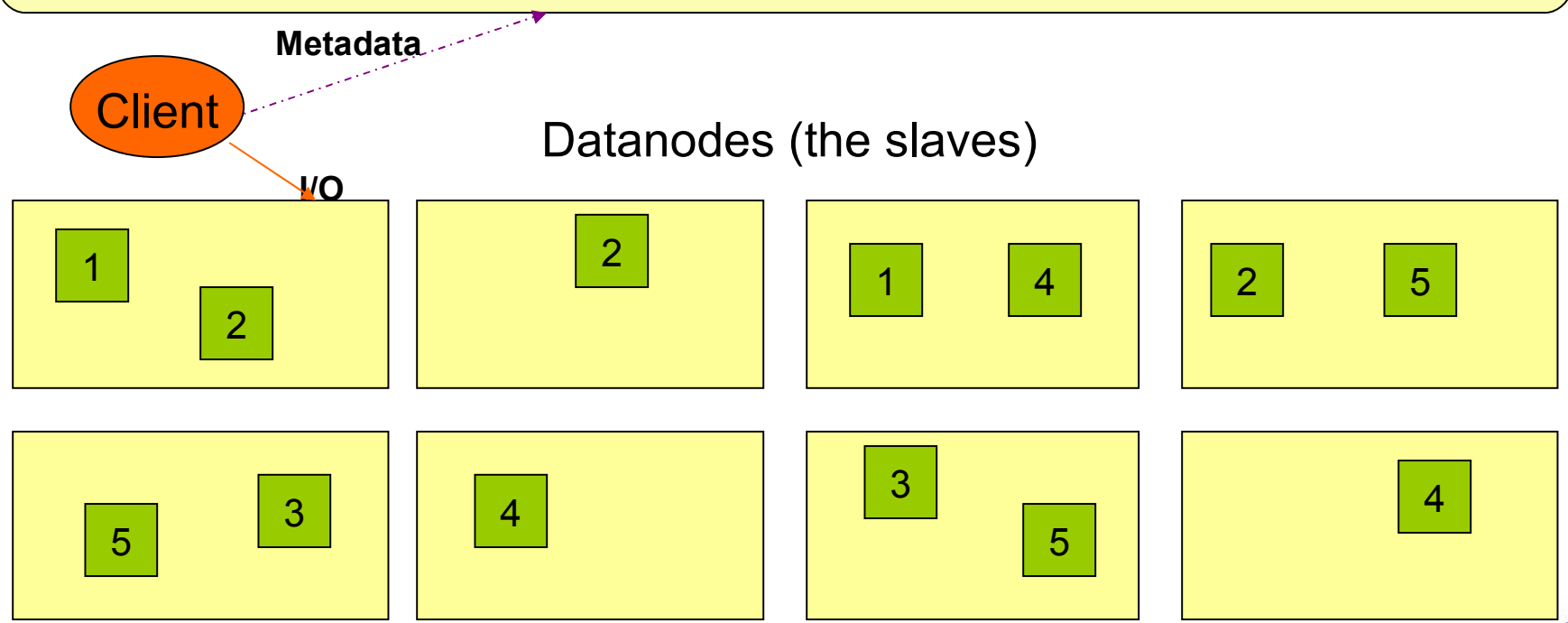
HDFS
運作方式？

HDFS 運作

Namenode (the master)

檔案路徑 – 副本數 ，由哪幾個 block 組成

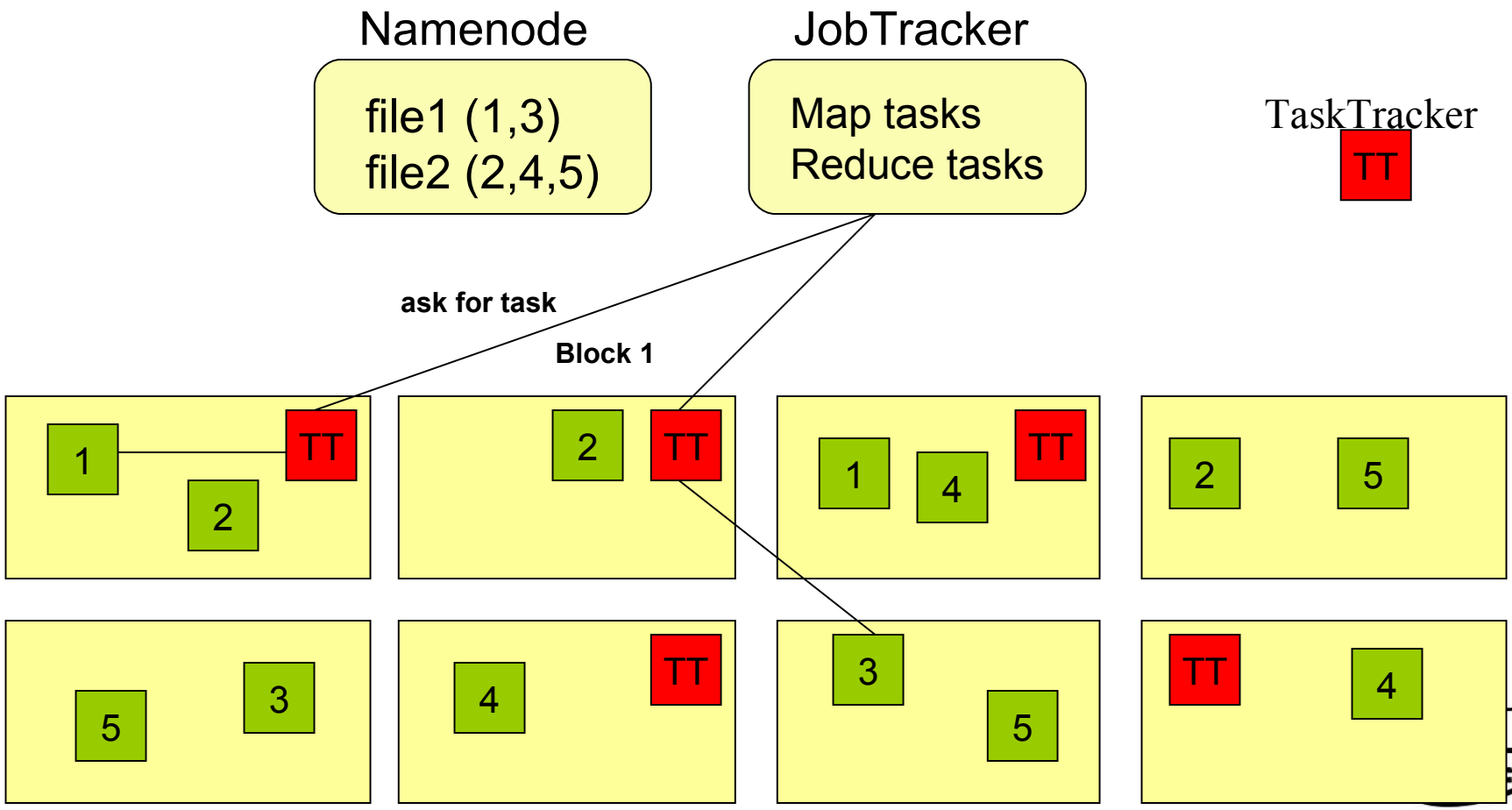
name:/users/joeYahoo/myFile - copies:2, blocks:{1,3}
name:/users/bobYahoo/someData.gzip, copies:3, blocks:{2,4,5}



HDFS
運作方式？

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）

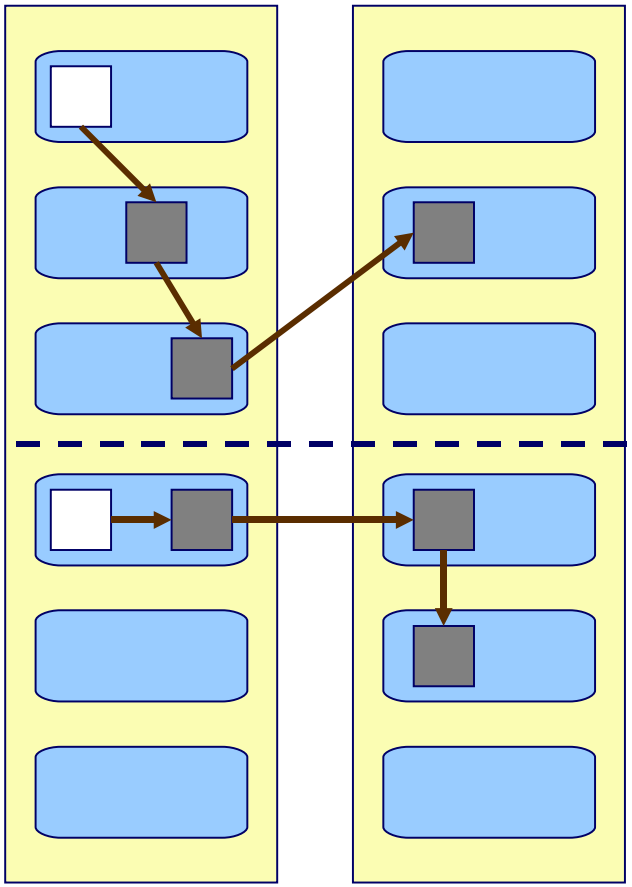


HDFS
運作方式？

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 壞掉可以手動復原

如何達成其好處？

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

功能為何？

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

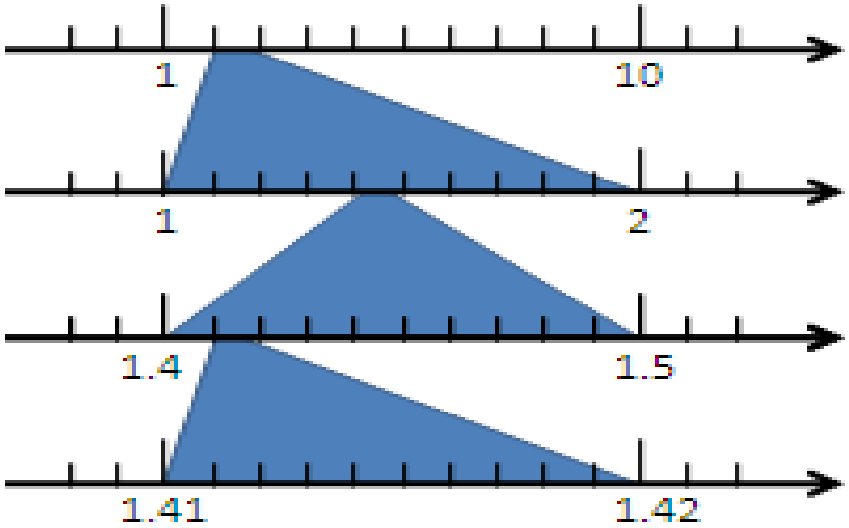
```
hadoop fs [-fs <local | file system URI>] [-conf <configuration file>]
        [-D <property=value>] [-ls <path>] [-lsr <path>] [-du <path>]
        [-dus <path>] [-mv <src> <dst>] [-cp <src> <dst>] [-rm <src>]
        [-rmr <src>] [-put <localsrc> <dst>] [-copyFromLocal <localsrc> <dst>]
        [-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]]
```

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)

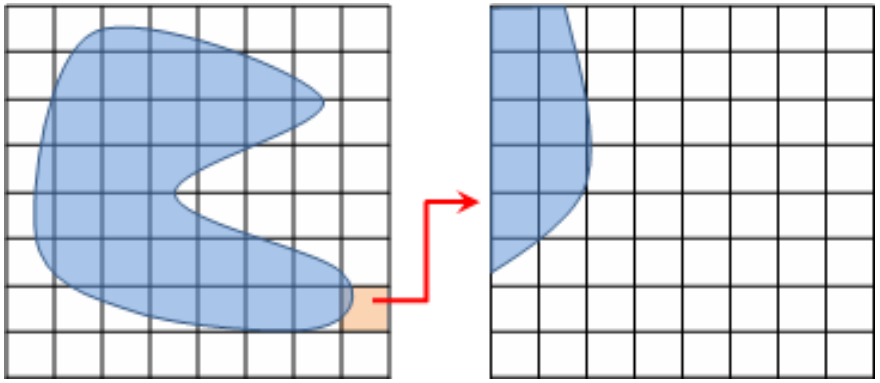
Divide and Conquer

範例一：十分逼近法

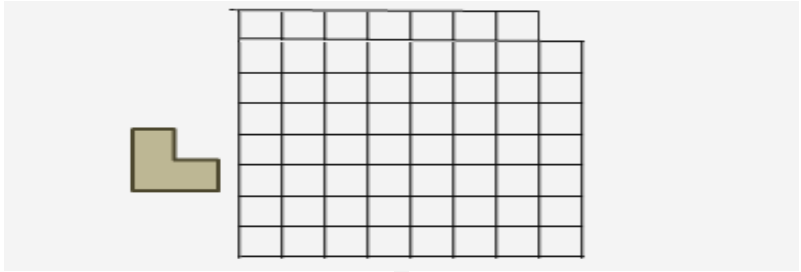


sqrt(2)

範例二：方格法求面積



範例三：鋪滿 L 形磁磚



範例四：

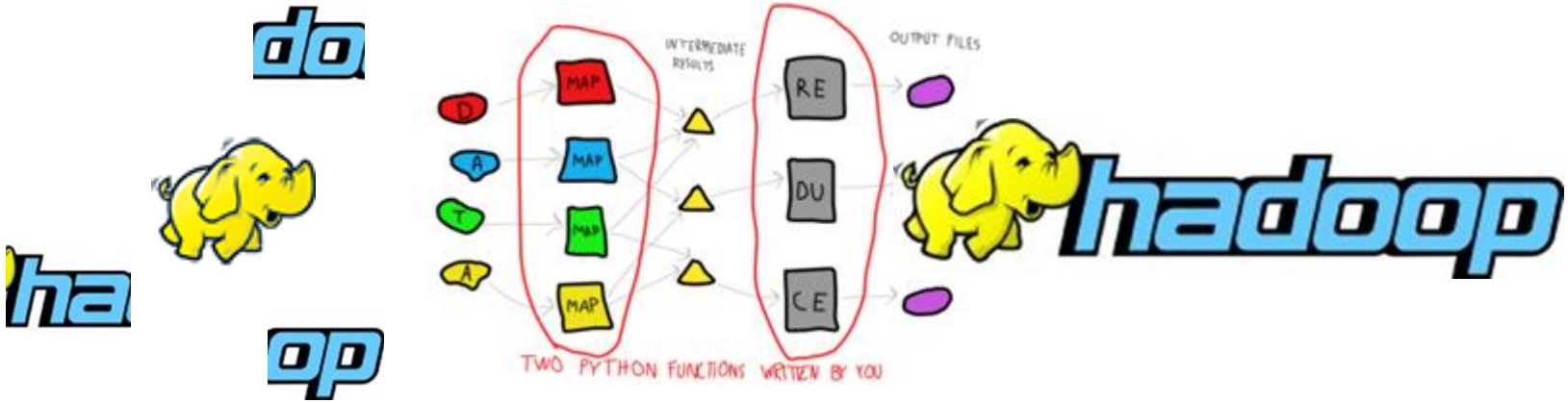
眼前有五階樓梯，每次可踏上一階或踏上兩階，那麼爬完五階共有幾種踏法？

Ex：(1,1,1,1,1) or (1,2,1,1)

Map Reduce 起源

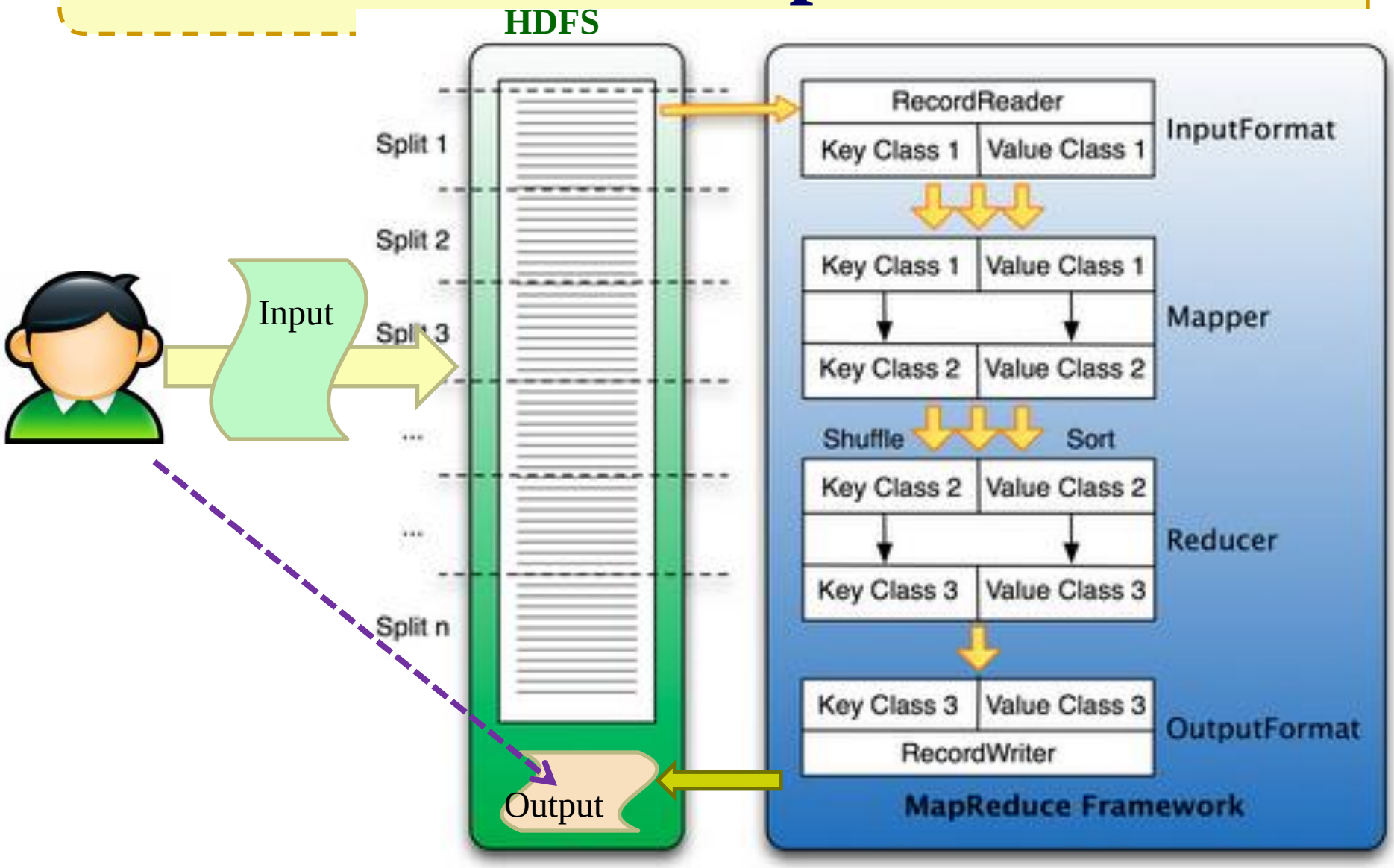
- Functional Programming : Map Reduce
  - map(...) :
    - [1,2,3,4] – (*2) -> [2,4,6,8]
  - reduce(...):
    - [1,2,3,4] - (sum) -> 10
- 演算法（Algorithms）：
  - Divide and Conquer
  - 分而治之
- 在程式設計的軟體架構內，適合使用在大規模數據的運算中

Hadoop MapReduce定義



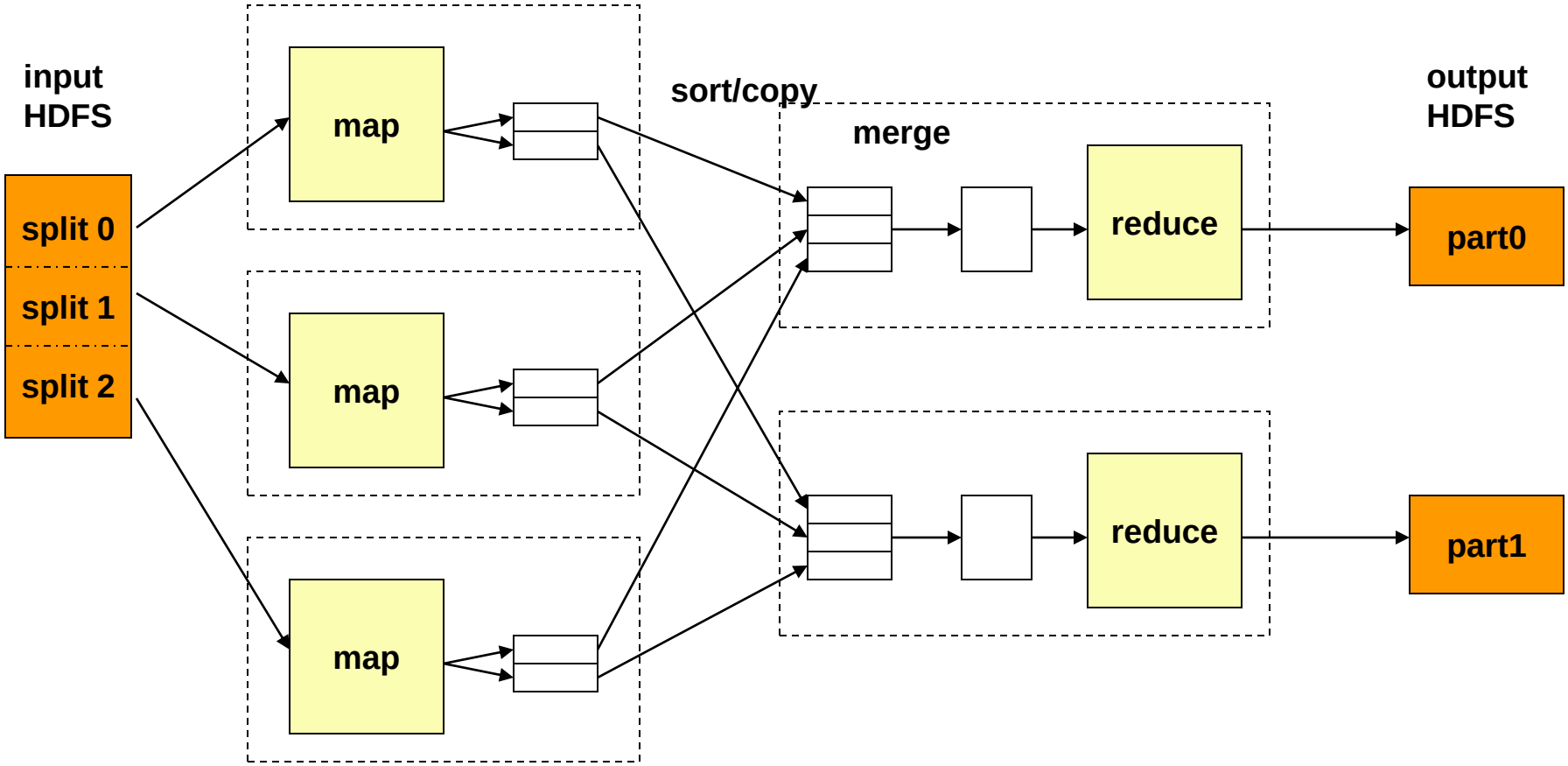
Hadoop Map/Reduce是一個易於使用的軟體平台，以MapReduce為基礎的應用程序，能夠運作在由上千台PC所組成的大型叢集上，並以一種可靠容錯的方式平行處理上P級別的資料集。

HDFS & MapReduce



部份圖片來源：http://www.larsgeorge.com/2009/05/hbase-mapreduce-101-part-i.html

Hadoop-MapReduce 運作流程



JobTracker跟NameNode取得需要運算的blocks

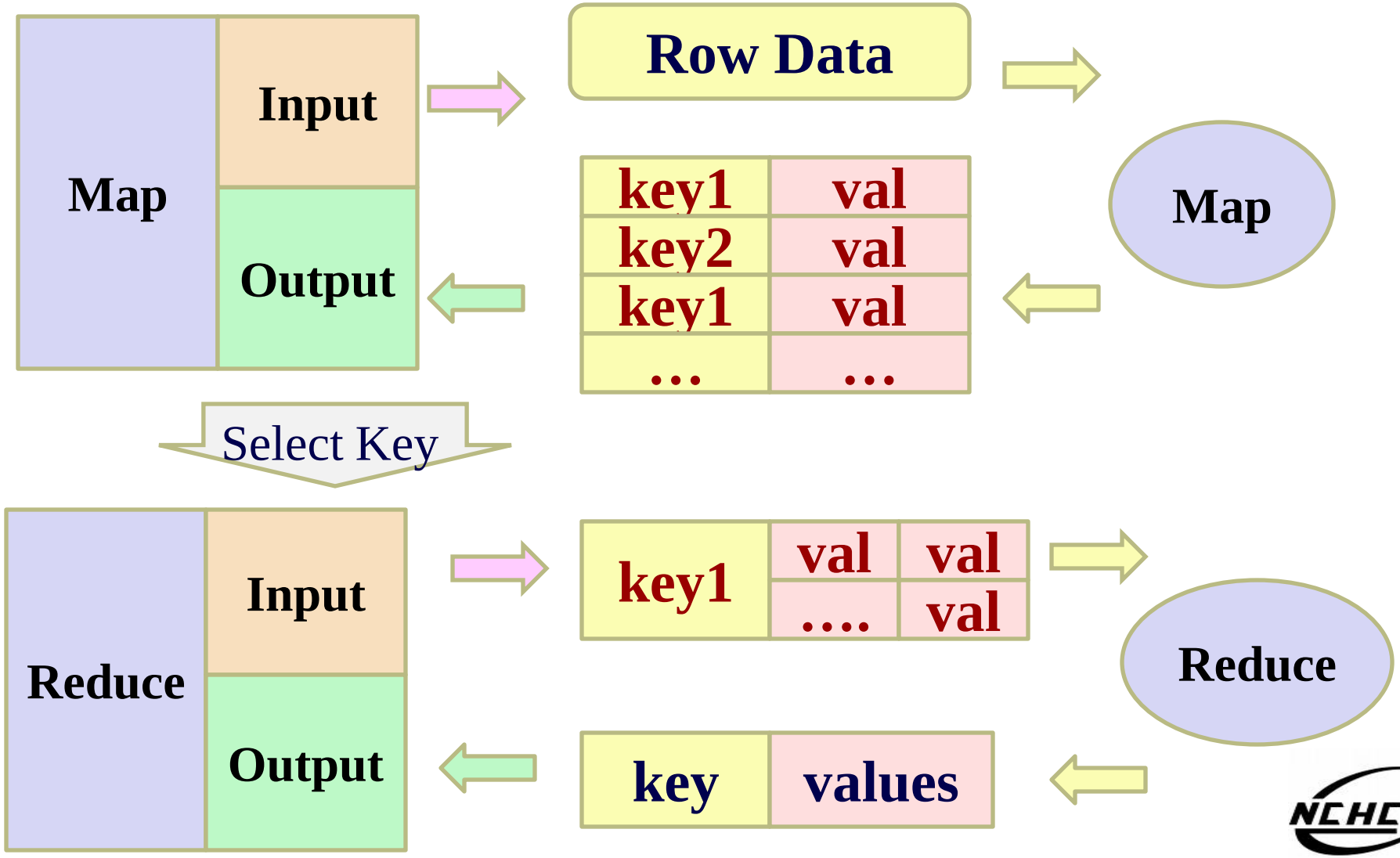
JobTracker選數個TaskTracker來作Map運算，產生些中間檔案

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

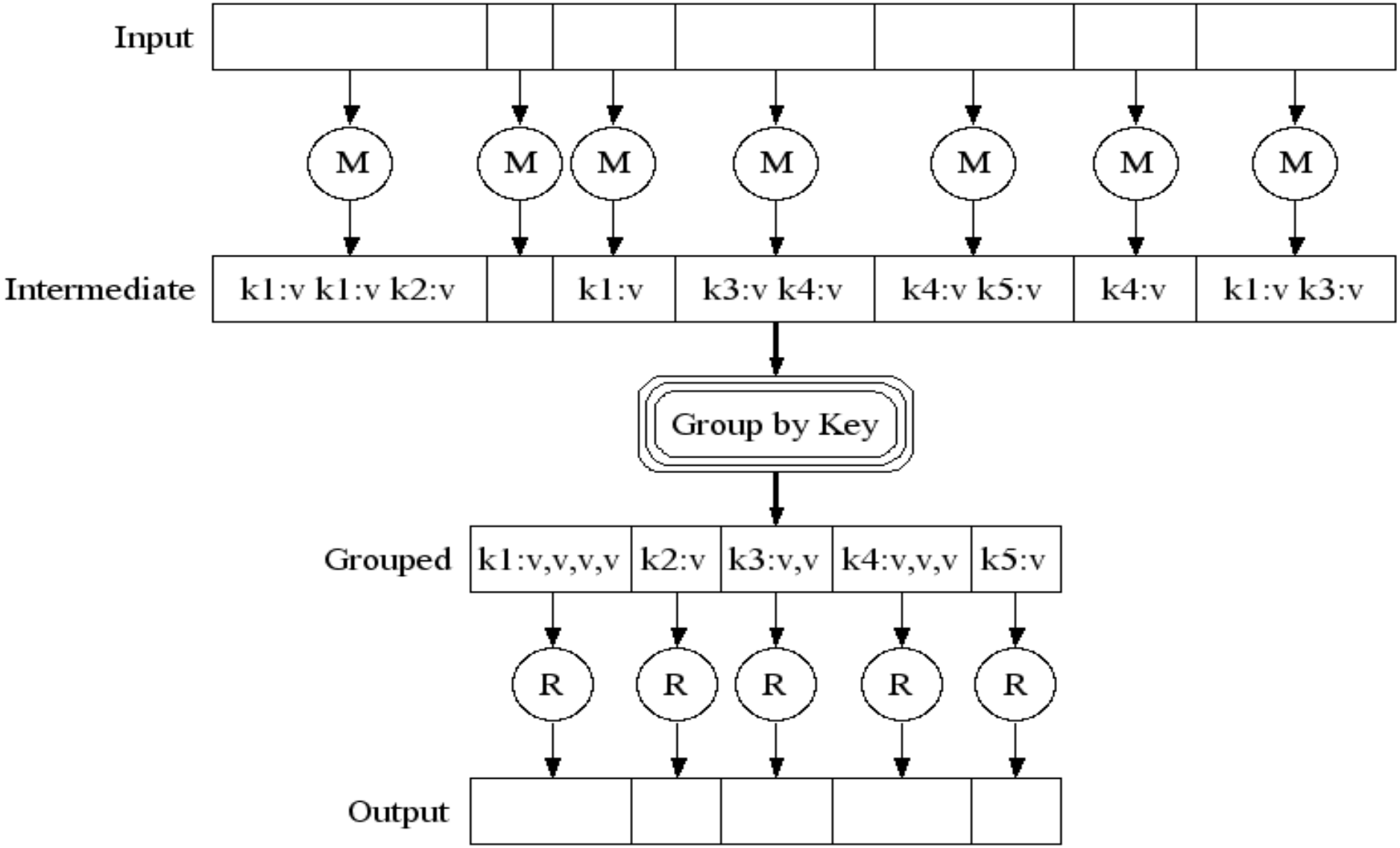
JobTracker派遣TaskTracker作reduce

reduce完後通知JobTracker與Namenode以產生output

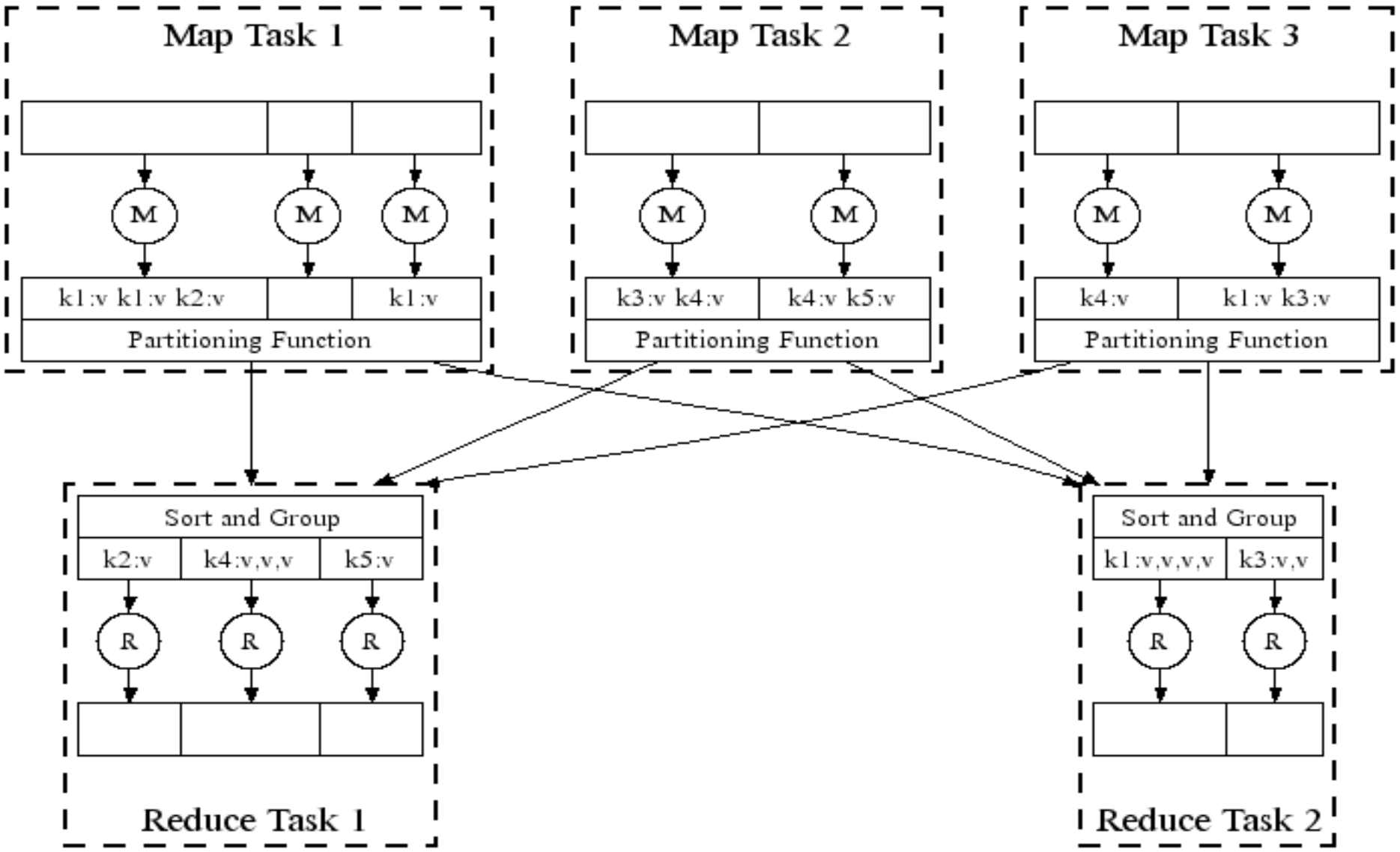
MapReduce 與 <Key, Value>



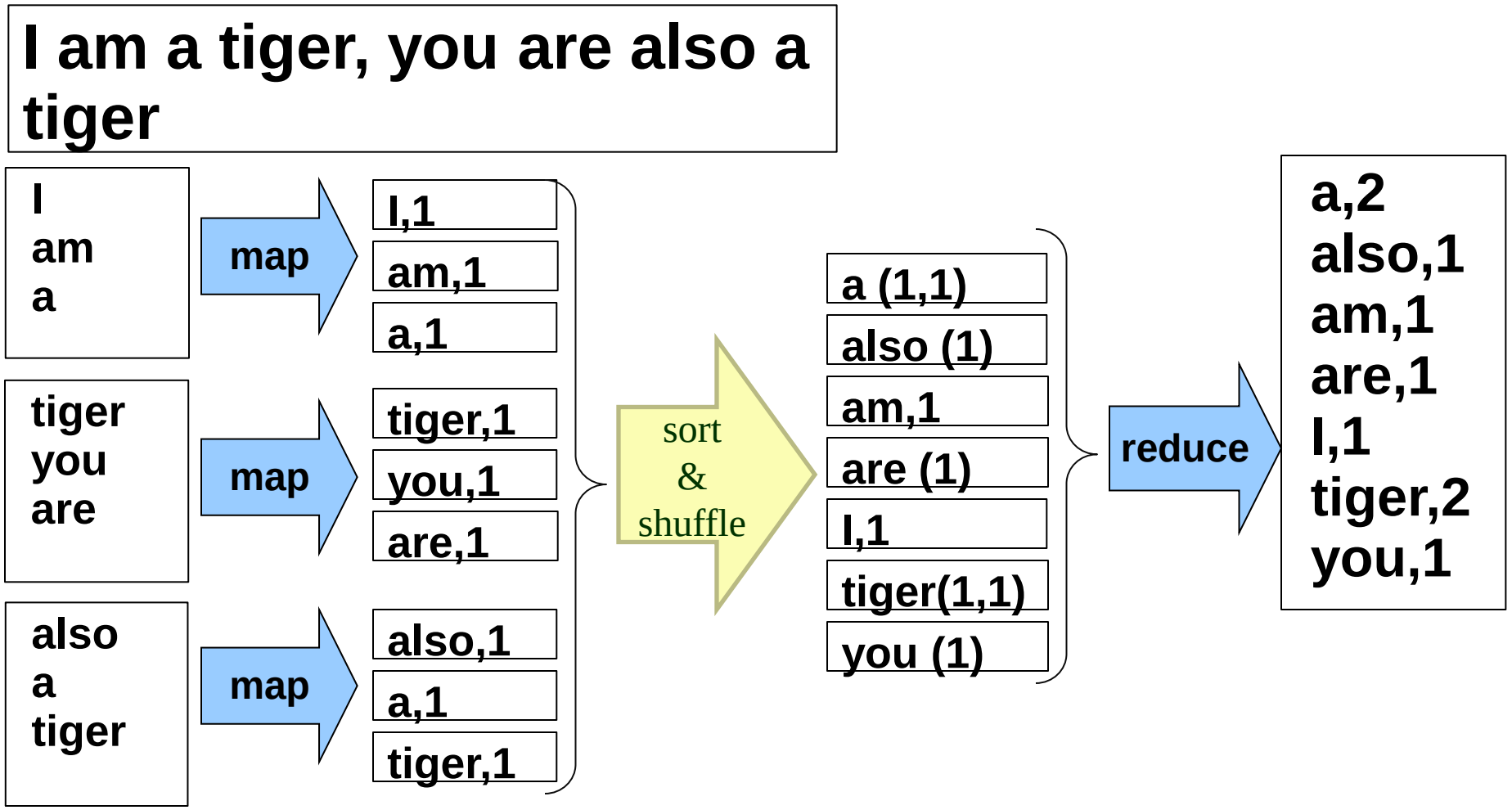
MapReduce 圖解



MapReduce in Parallel



範例



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

Hadoop適用於..

- 大規模資料集
- 可拆解的運算
- 批次處理
- 預先運算

- 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 Excange
  - 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
- IIIT, 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**，**C**ompact **M**uon **S**olenoid）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

結論

- 目前已經有許多大公司利用Hadoop，呈現其高效與廣泛性
- 適合於：複雜但可拆解的計算，大量且獨立的資料
- 問題
  - HDFS 可否不搭配MapReduce而獨立運作？
  - 承上，MapReduce 呢？

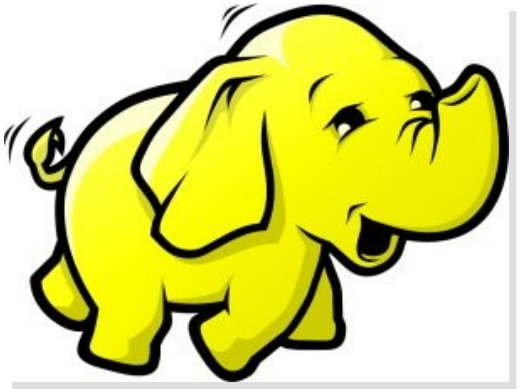
當企鵝龍遇上小飛象

DRBL-Hadoop

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



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

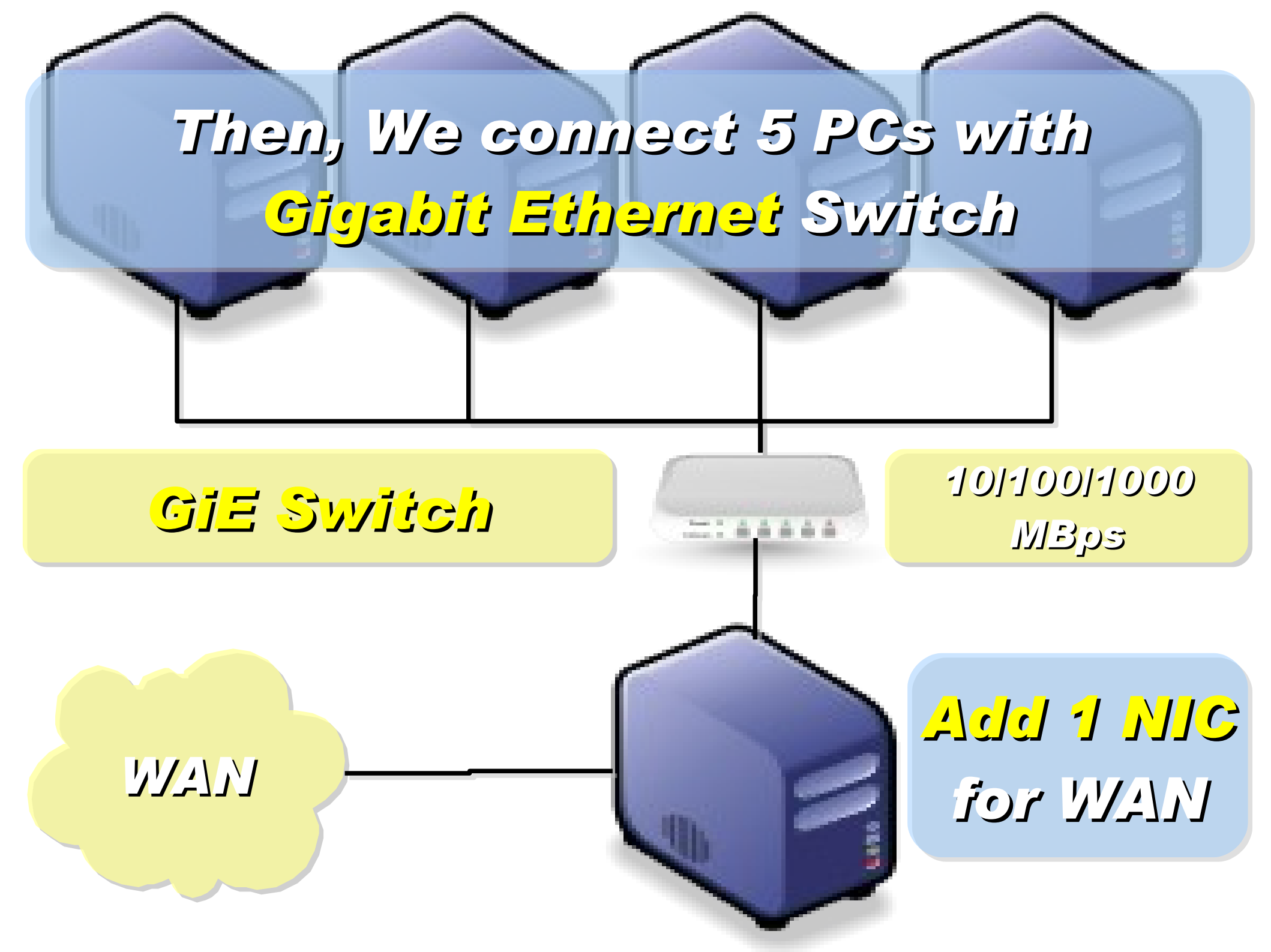
At First, We have “ 4 + 1 ” PC Cluster

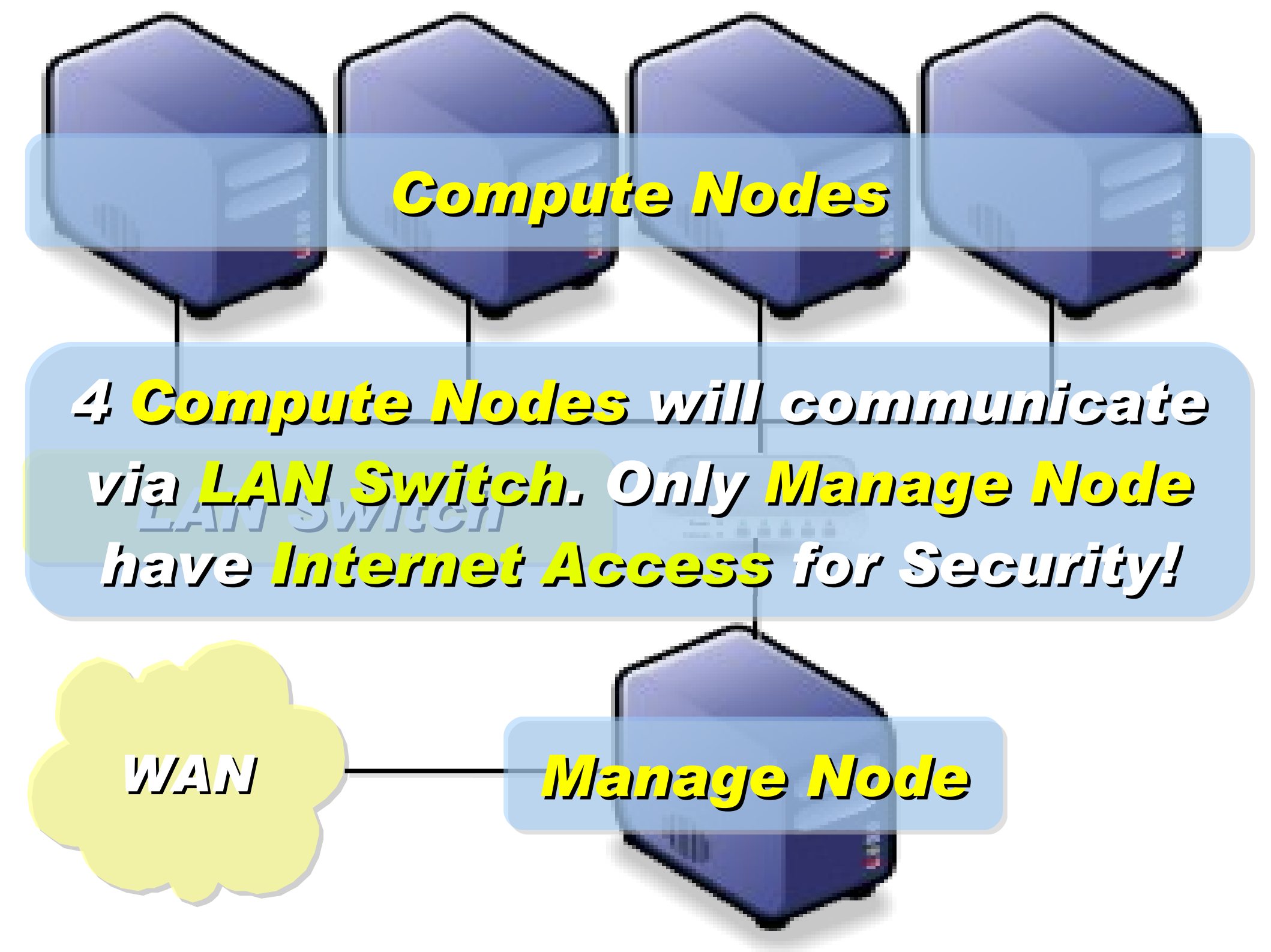
It'd better be

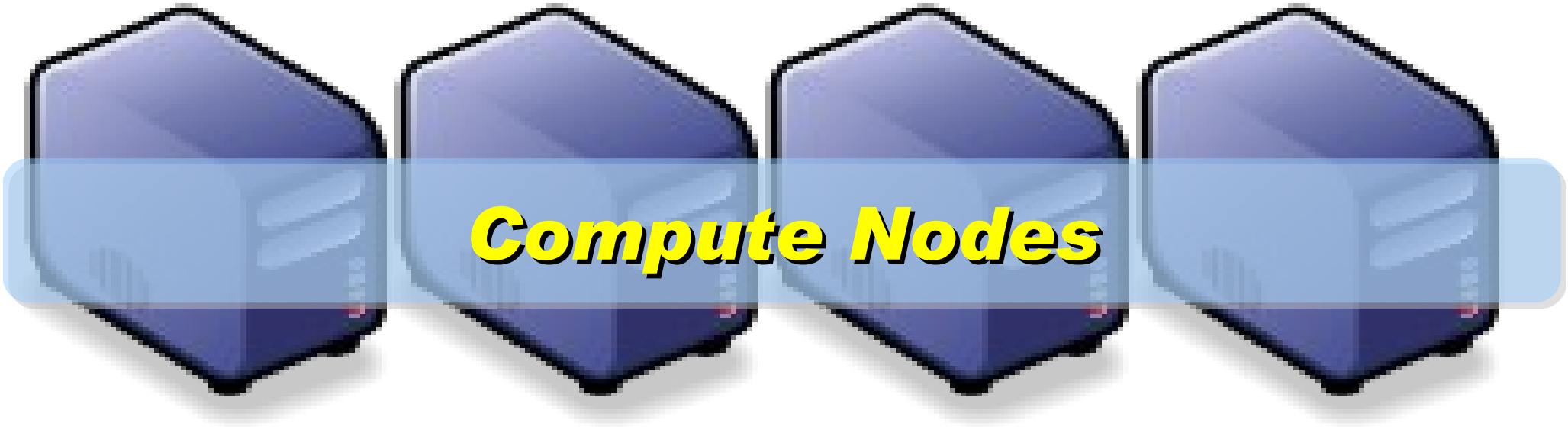
2^n

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!
LAN Switch
WAN
Manage Node



Basic System Setup for Cluster

Messaging

Account Mgnt.

MPICH

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 Mgnt.

OpenPBS

File Sharing

NFS

Extra

Messaging

Account Mgnt.

MPICH

SSHD

NIS

YP

GCC

GNU Libc

Bash

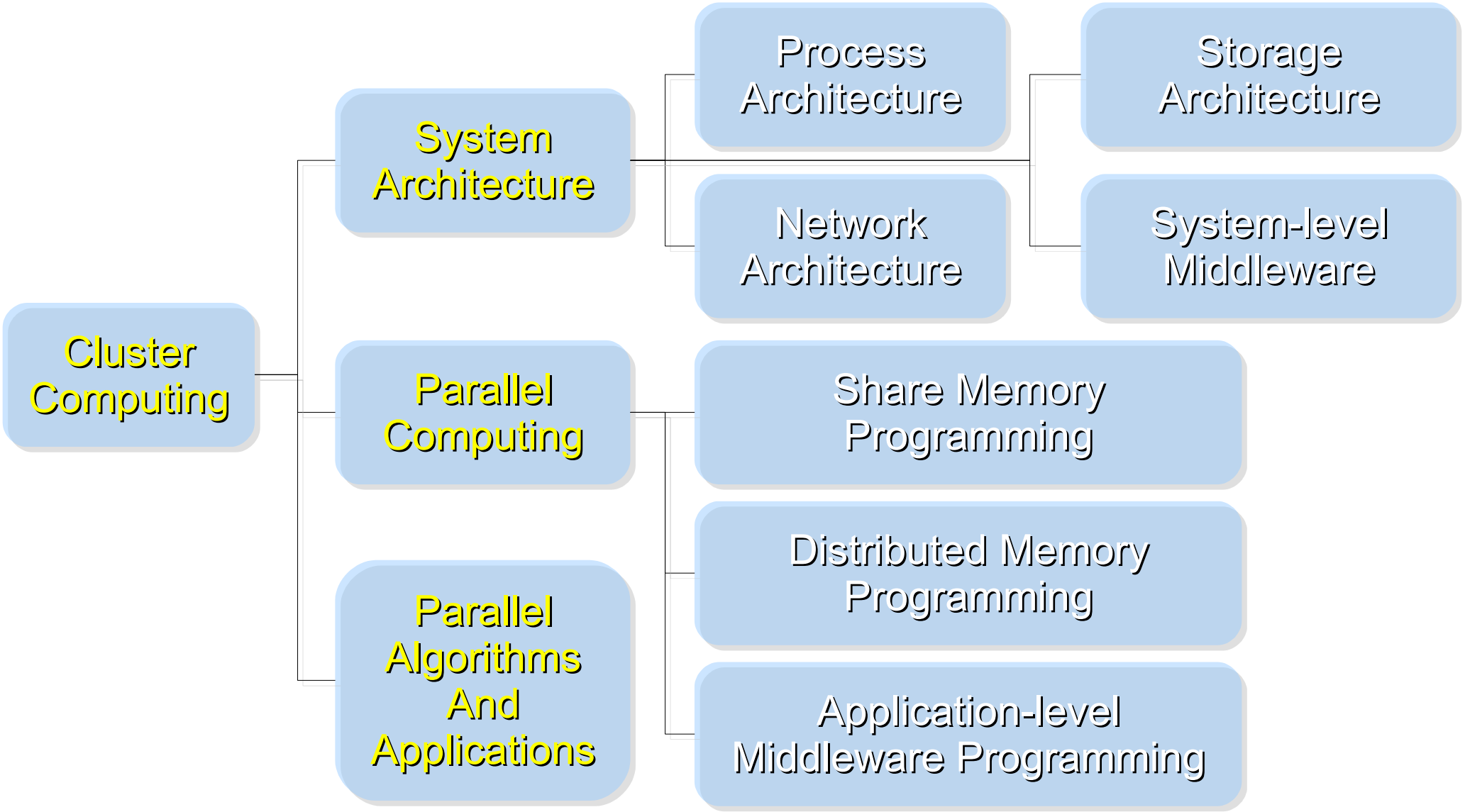
Perl

Kernel Module

Linux Kernel

Boot Loader

Research topics about PC Cluster



Ref: Cluster Computing in the Classroom: Topics, Guidelines, and Experiences
http://www.gridbus.org/papers/CC-Edu.pdf

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 Syncronization

How to deploy 4000+ Nodes ????

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**

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

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

PART 2-1 :

Hadoop Deployment Tool

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw

SmartFrog

- 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

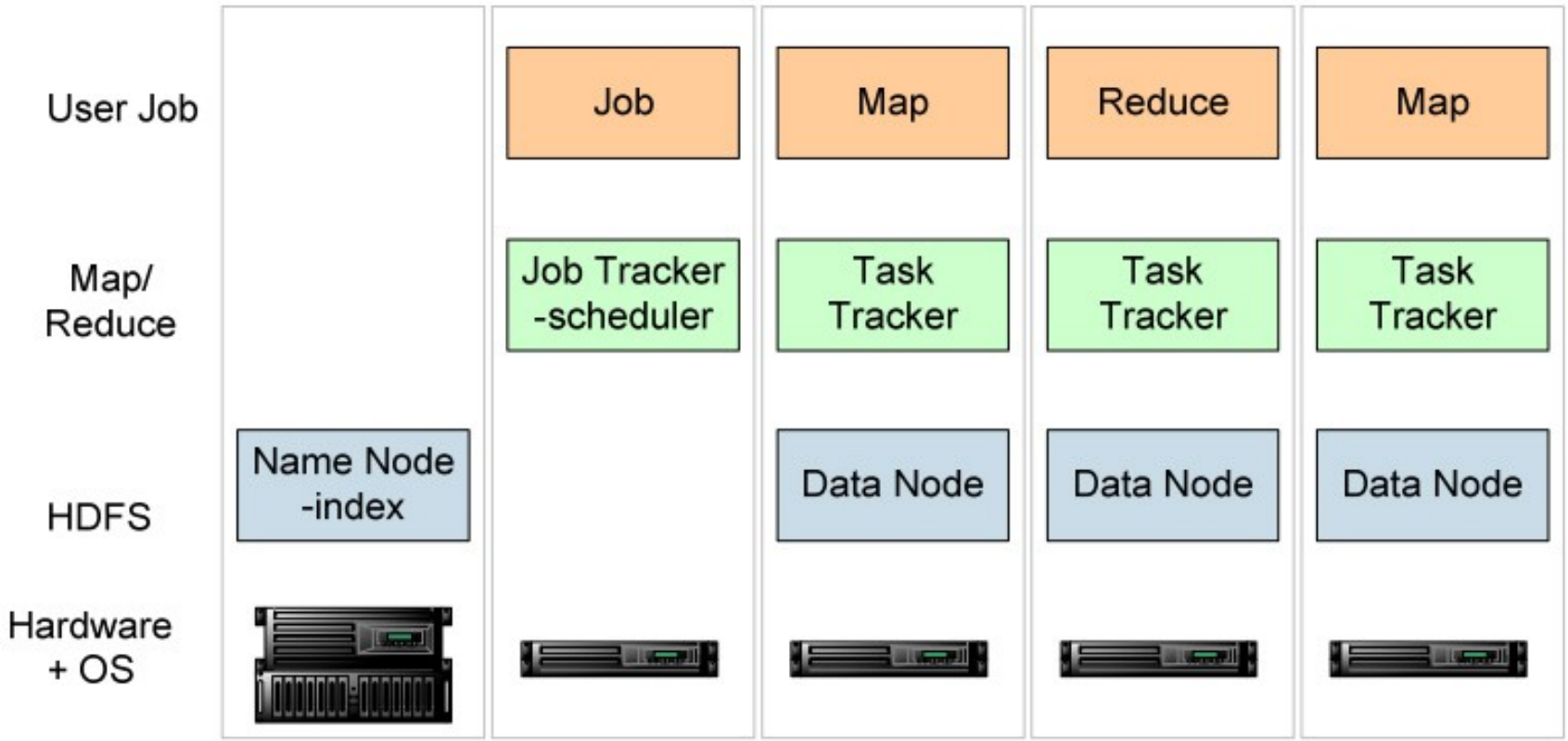
SmartFrog - HPLabs' CM tool

- 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

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

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/~stevel/slides/deploying_hadoop_with_smartfrog.pdf

The locked-down cluster

- PXE Preboot of OS images
- RedHat Kickstart to serve up (see 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

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

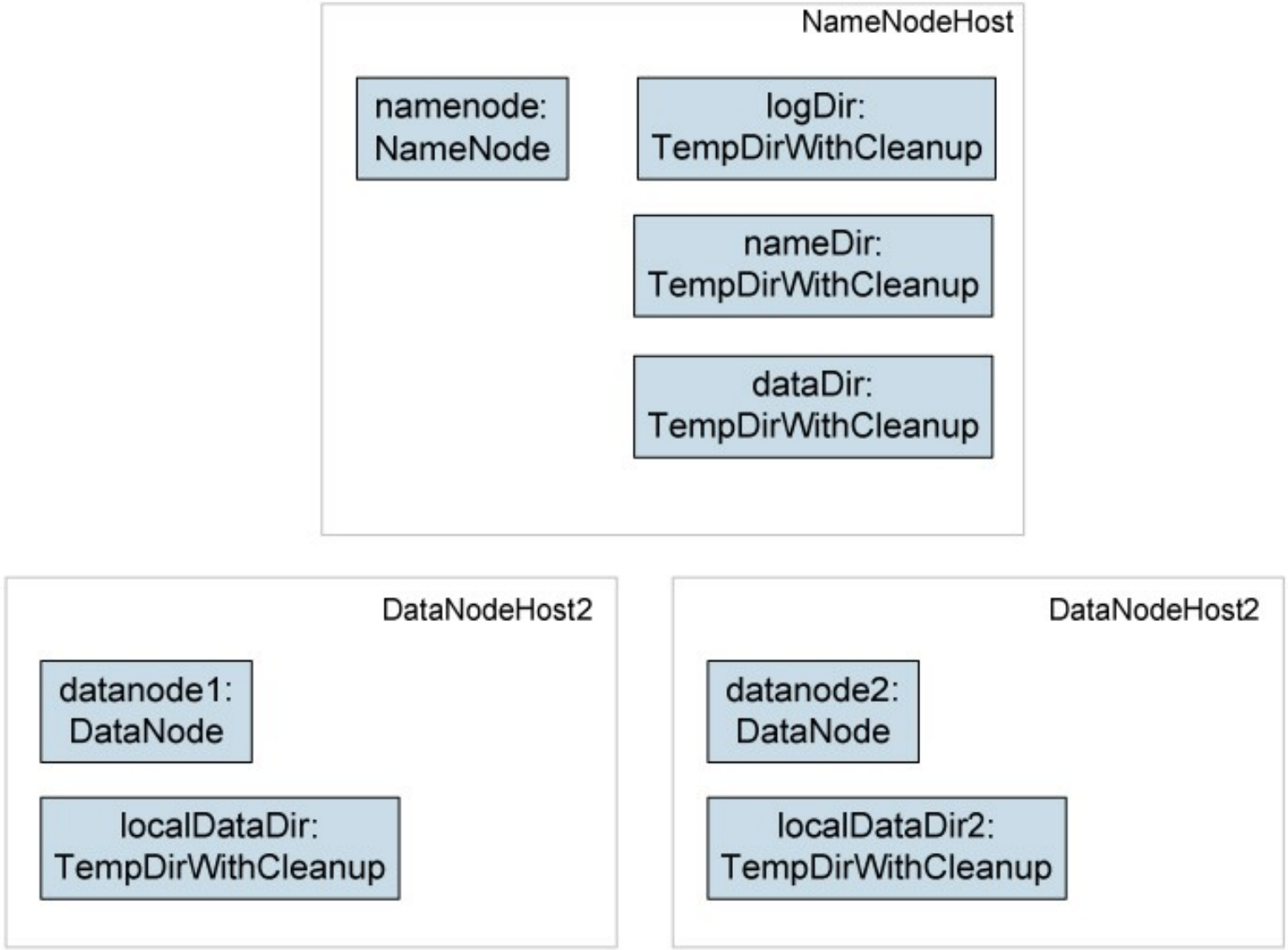
Model the system in the SmartFrog language

```
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/~stevel/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

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/~stevel/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連接到一台伺服器主機

Powered by DRBL

Diskfull PC

=

+

+

Diskless PC

Server

source: http://www.mren.com.tw

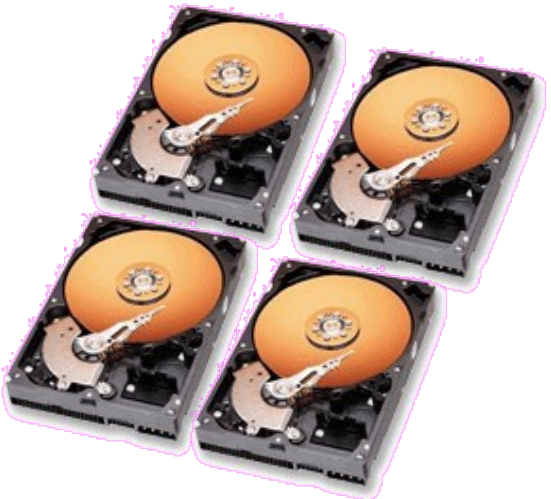
何謂再生龍 Clonezilla ??

- **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

Disk to Disk

Disk to Image

Image to N Disks



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 1台 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=120,000 (MS Windows授權1台電腦之授權費NT$3,000)	軟體授權 NT$0
合法複製、分享	需負擔授權費	複製合法 NT$0
多元化電腦教學	不同系統無法並存	Linux 與M$ Windows可並存

教育單位採用 DRBL

高速計算研究
資料儲存備援

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

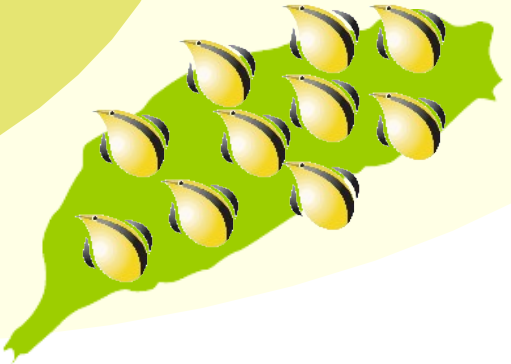
以某商業獨家軟體每機 3000 元授權費計，
每班 35 台電腦 (3000*35*939)

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

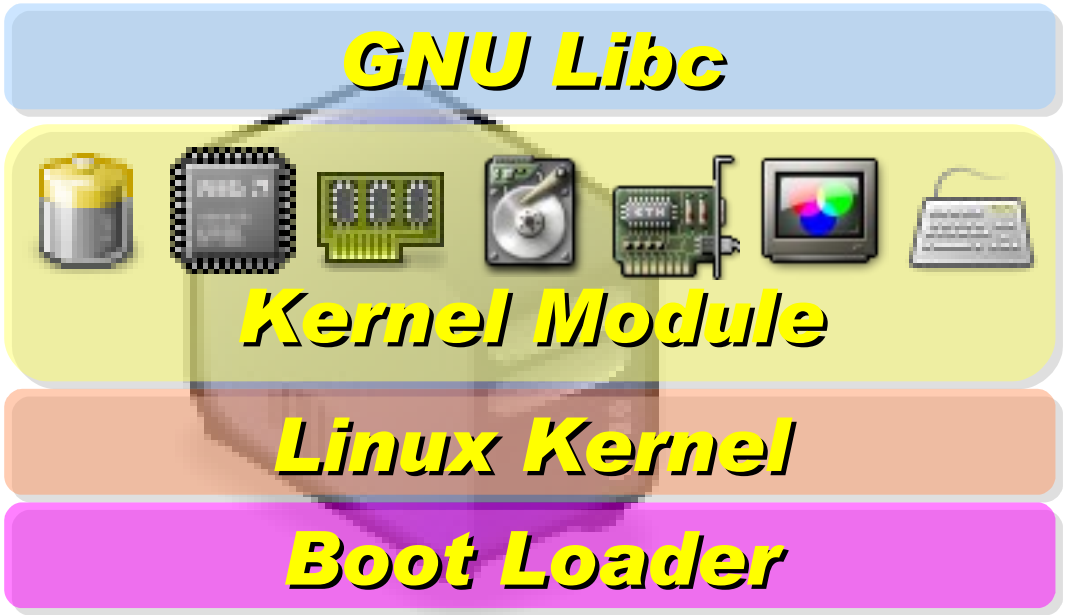
Yao-Tsung Wang

jazz@nchc.org.tw

Powered by DRBL

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

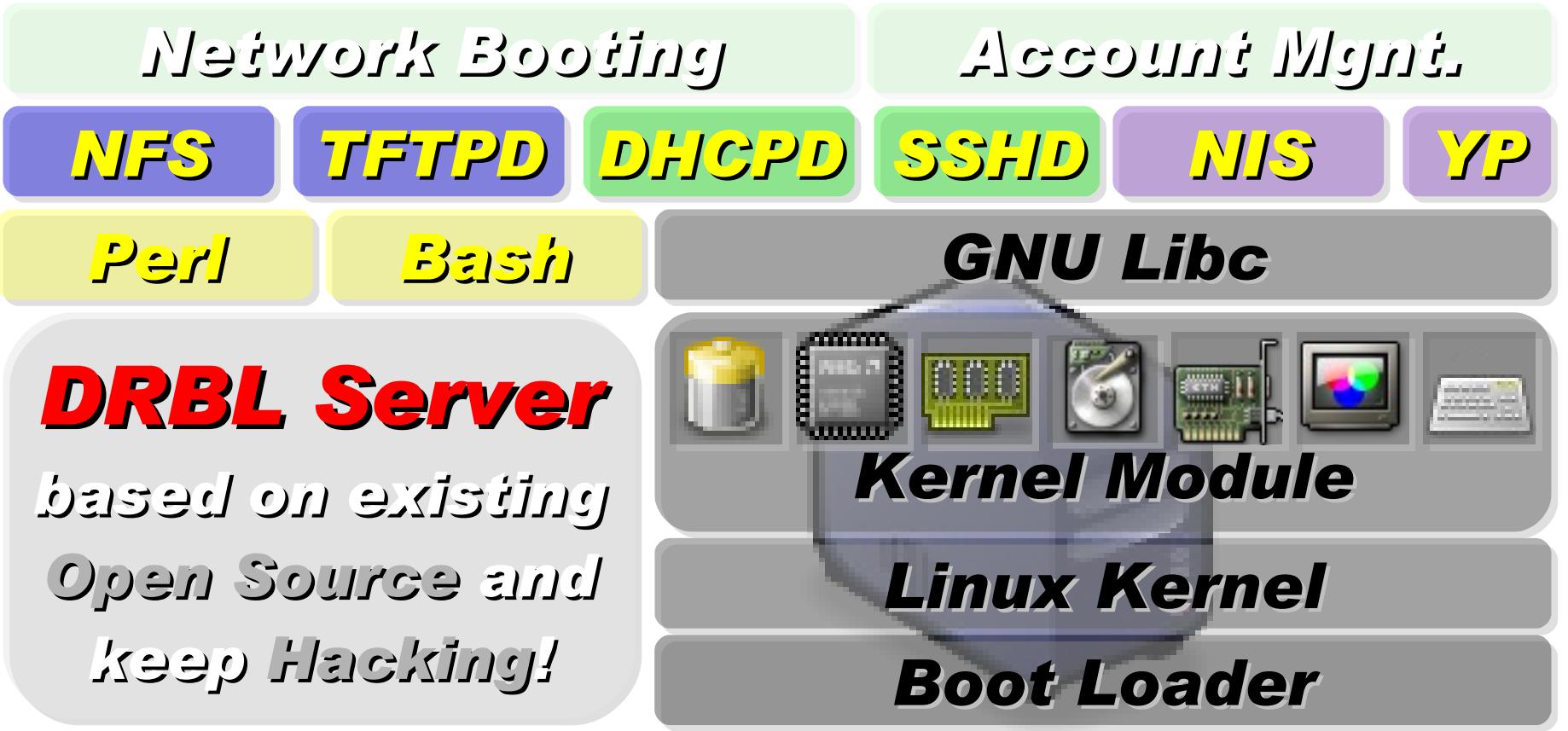
Redhat, Fedora, CentOS, Mandriva, Ubuntu, Debian, ...



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

There are lots of service needed:

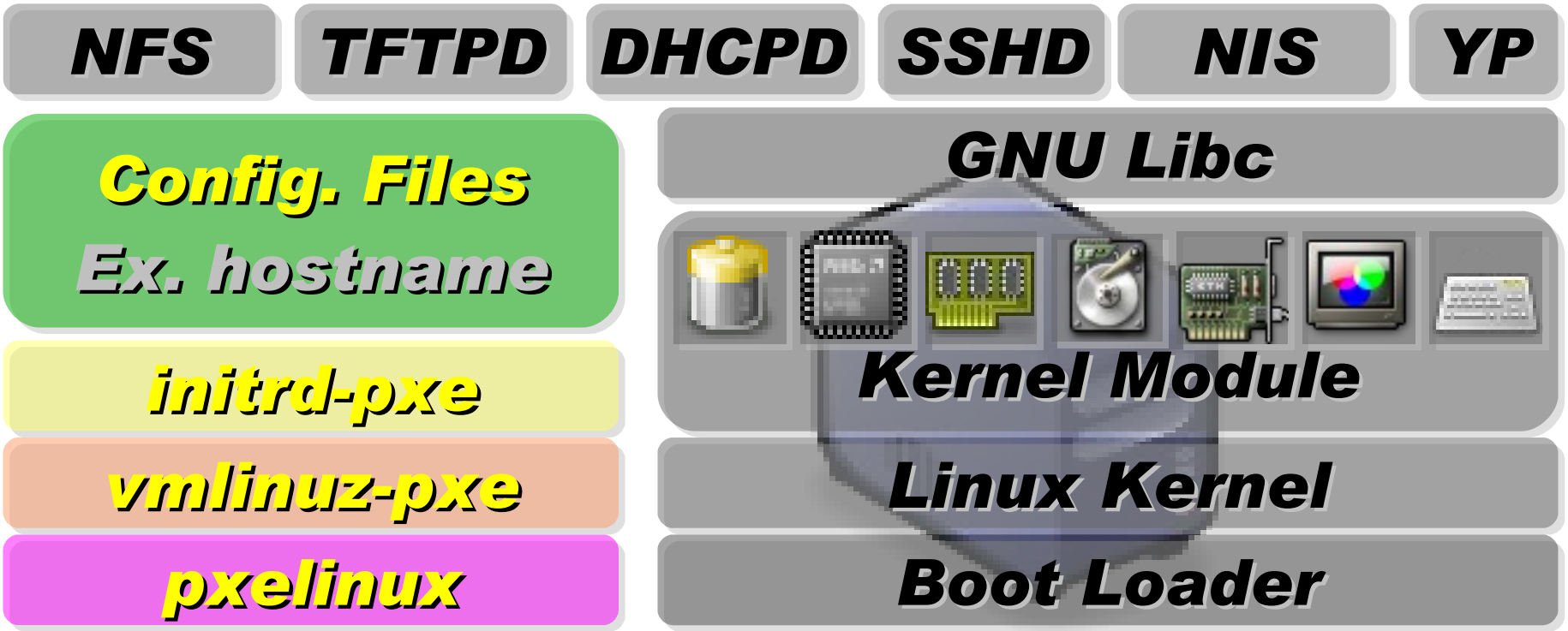
SSHD, DHCPD, TFTPD, NFS Server,
NIS Server, YP Server ...



DRBL Server

based on existing Open Source and keep Hacking!

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



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

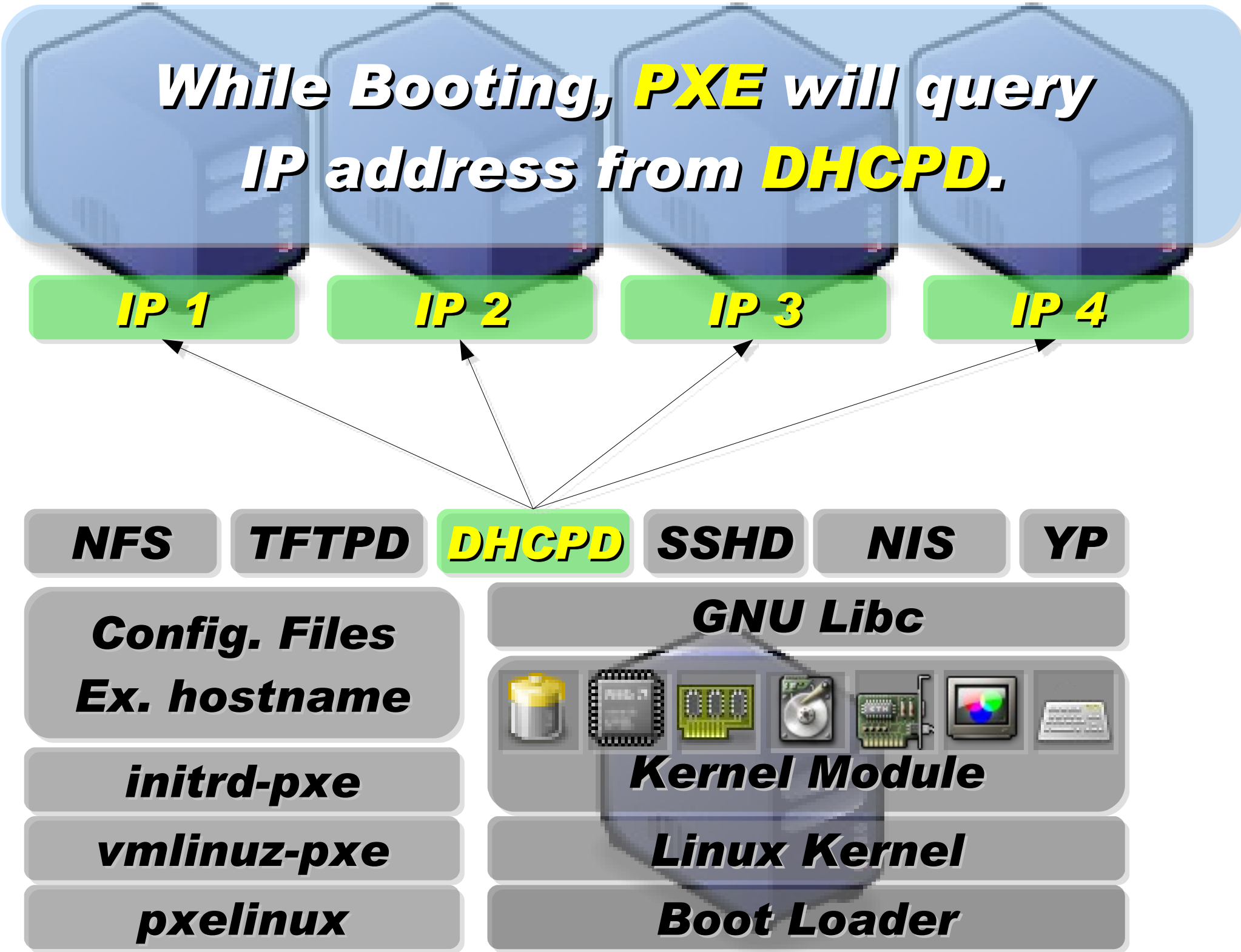
GNU Libc

Kernel Module

Linux Kernel

Boot Loader

While Booting, PXE will query IP address from DHCPD.



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

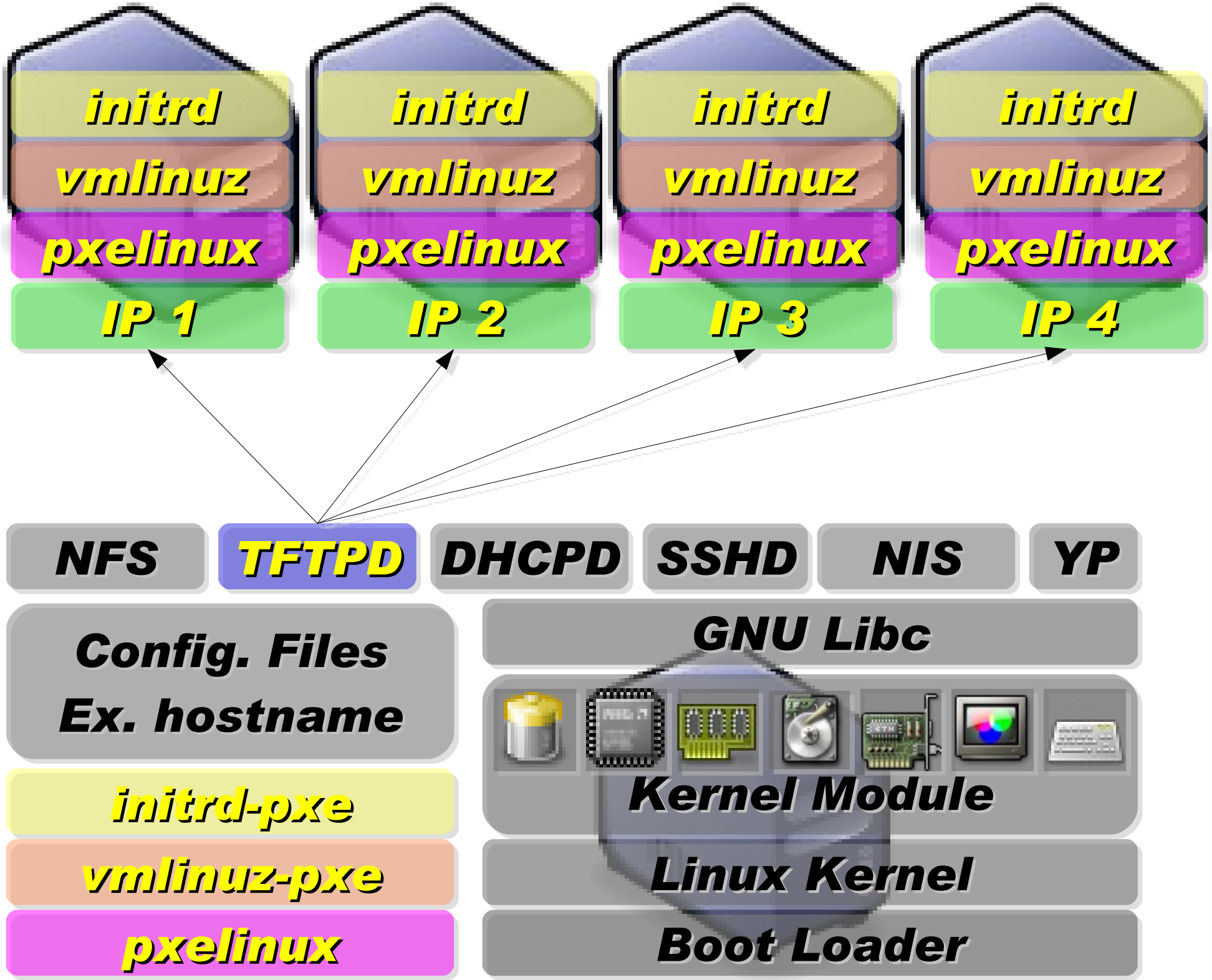
pxelinux

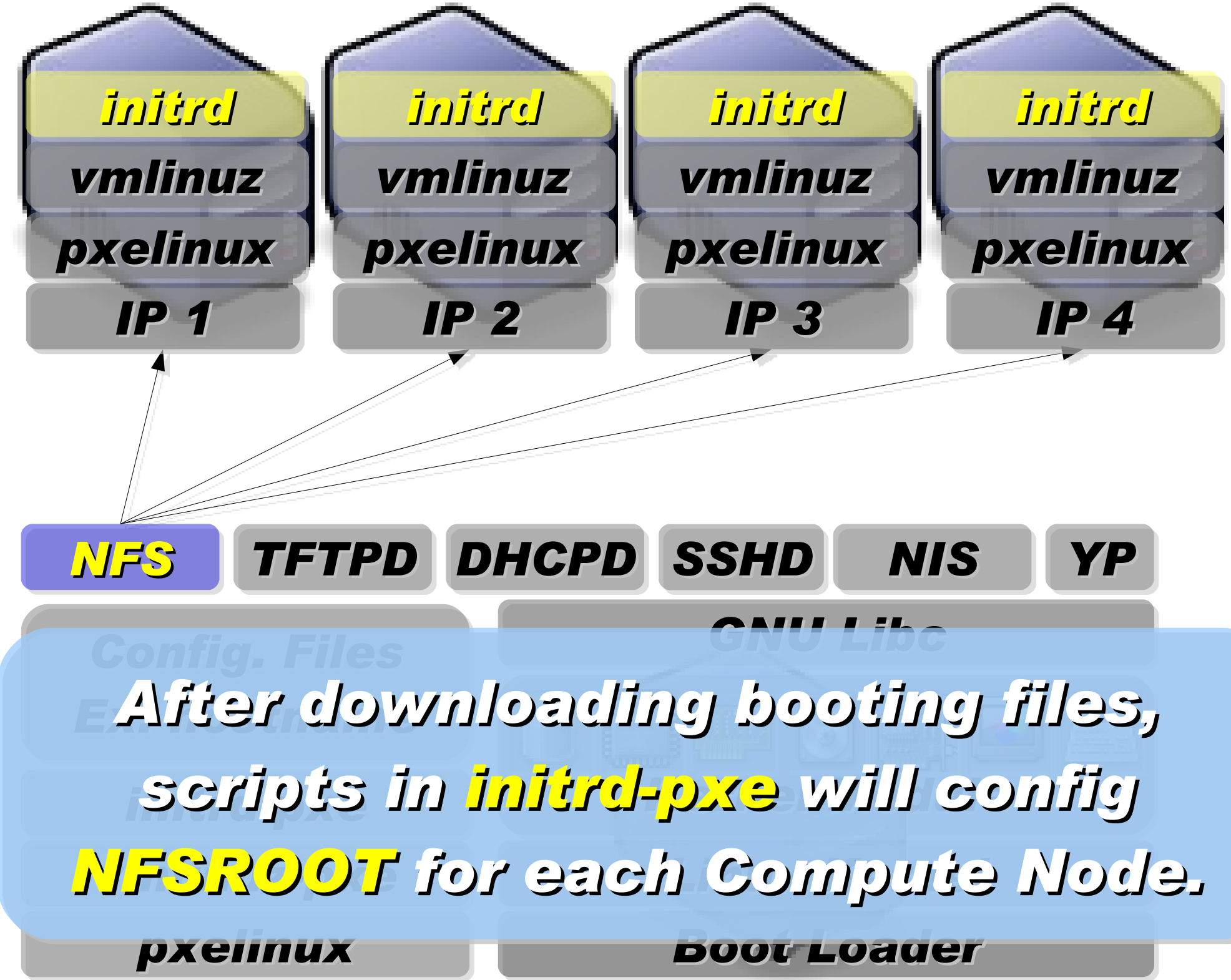
GNU Libc

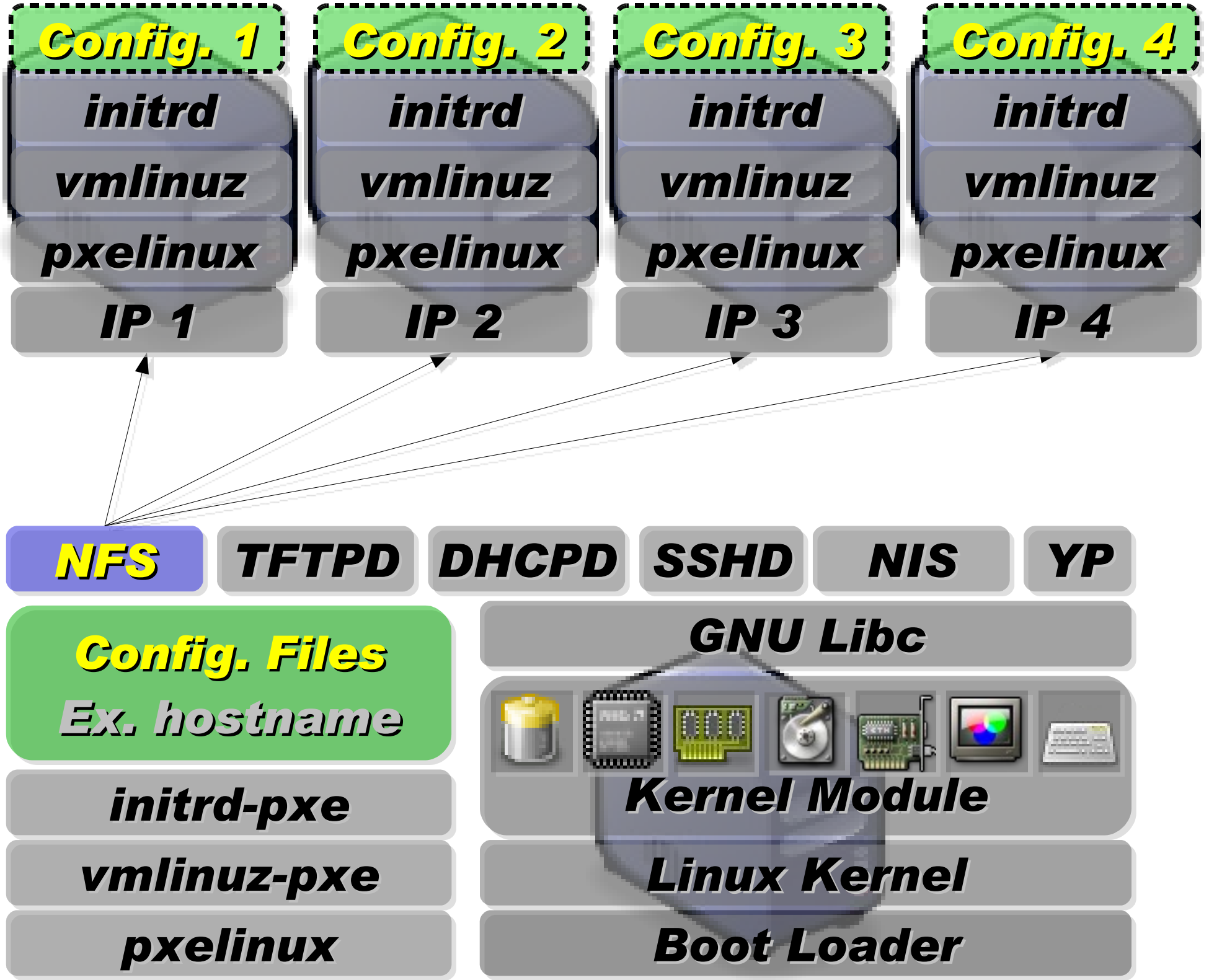
Kernel Module

Linux Kernel

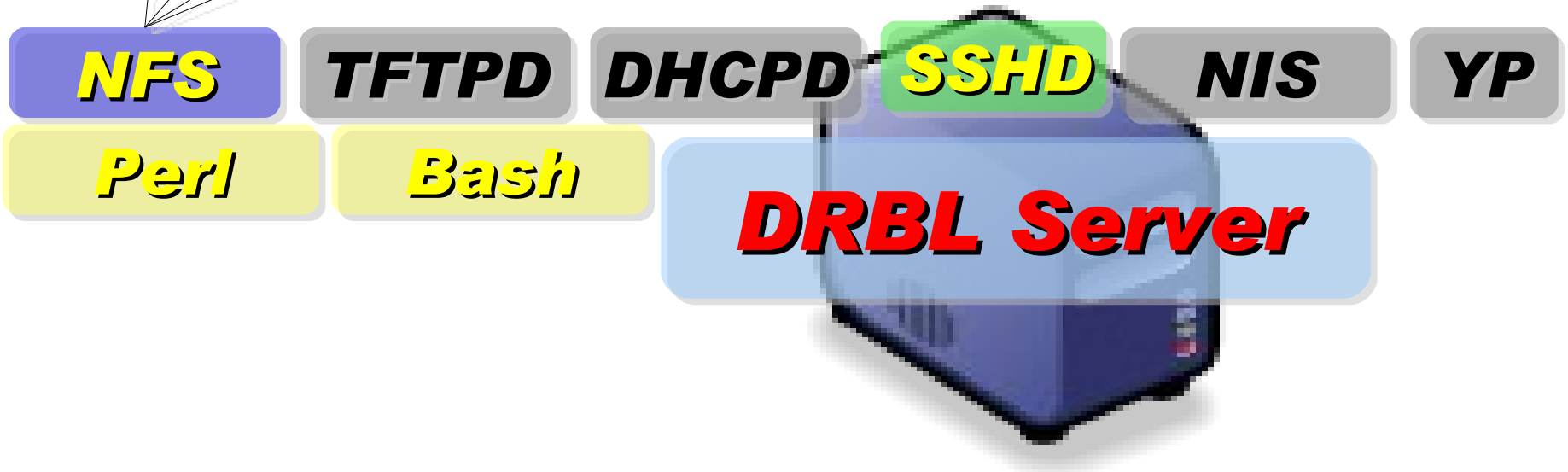
Boot Loader

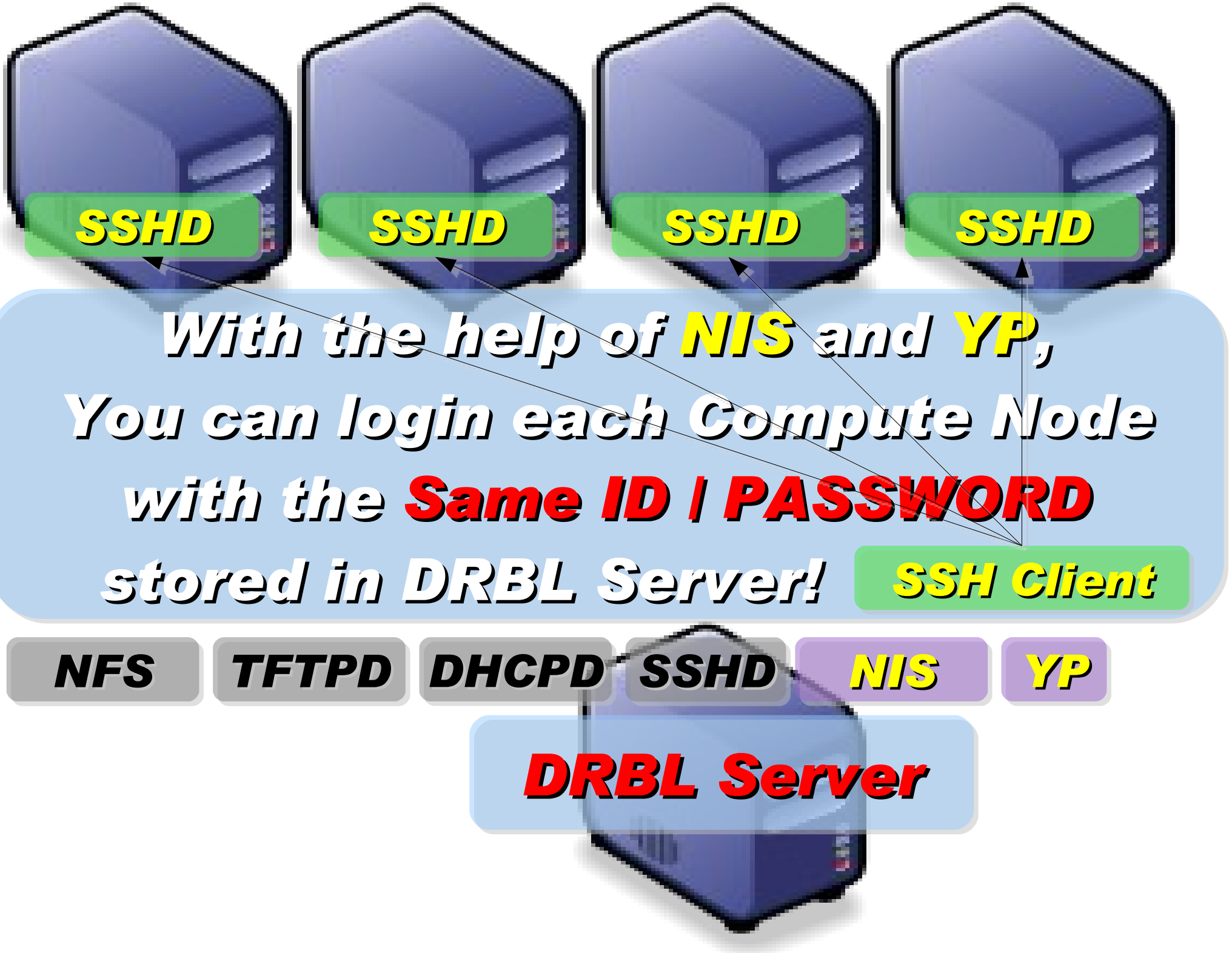

initrd
vmlinuz
pxelinux
IP 1
initrd
vmlinuz
pxelinux
IP 2
initrd
vmlinuz
pxelinux
IP 3
initrd
vmlinuz
pxelinux
IP 4
NFS
TFTPD
DHCPD
SSHD
NIS
YP
Config. Files
Ex. hostname
initrd-pxe
vmlinuz-pxe
pxelinux
GNU Libc
Kernel Module
Linux Kernel
Boot Loader


initrd
vmlinuz
pxelinux
IP 1
initrd
vmlinuz
pxelinux
IP 2
initrd
vmlinuz
pxelinux
IP 3
initrd
vmlinuz
pxelinux
IP 4
NFS
TFTPD
DHCPD
SSHD
NIS
YP
After downloading booting files, scripts in initrd-pxe will config NFSROOT for each Compute Node.


Config. 1
Config. 2
Config. 3
Config. 4
initrd
vmlinuz
pxelinux
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

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




SSHD
SSHD
SSHD
SSHD
With the help of NIS and YP,
You can login each Compute Node
with the Same ID / PASSWORD
stored in DRBL Server!
SSH Client
NFS
TFTPD
DHCPD
SSHD
NIS
YP
DRBL Server

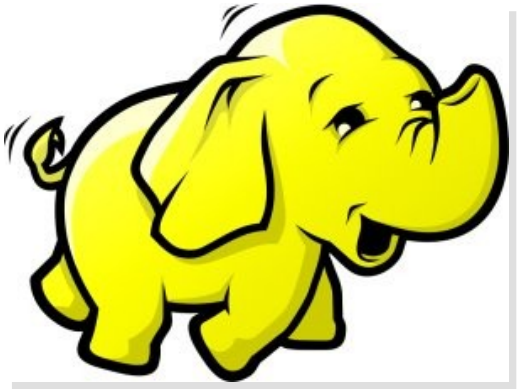
PART 2 -1:

當企鵝龍遇上小飛象

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



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
```

trac
Integrated SCM & Project Management

root / drbl-hadoop-0.1

Name
../
drbl-hadoop
drbl-hadoop-mount-disk

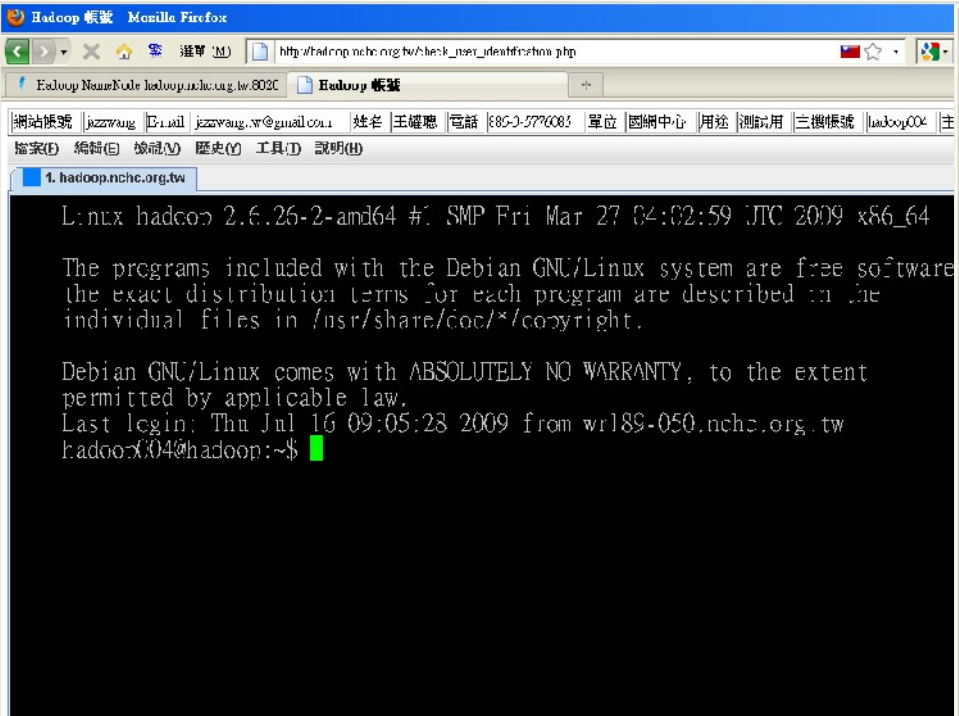
trac
Integrated SCM & Project Management

root / hadoop-register

Name	Size	Rev	Age	Last
../				
etc		103	4 weeks	wa
adduser.php	1.3 kB	85	6 weeks	wa
check_activate_code.php	2.2 kB	85	6 weeks	wa

關於 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



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**

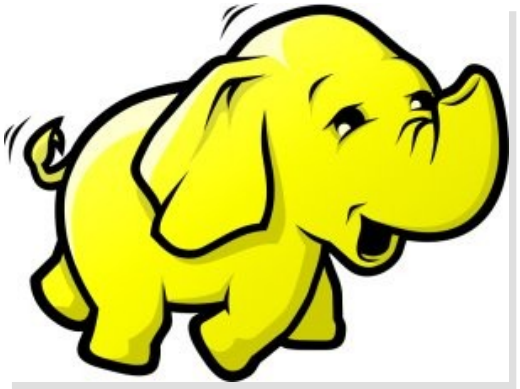
PART 2 -2:

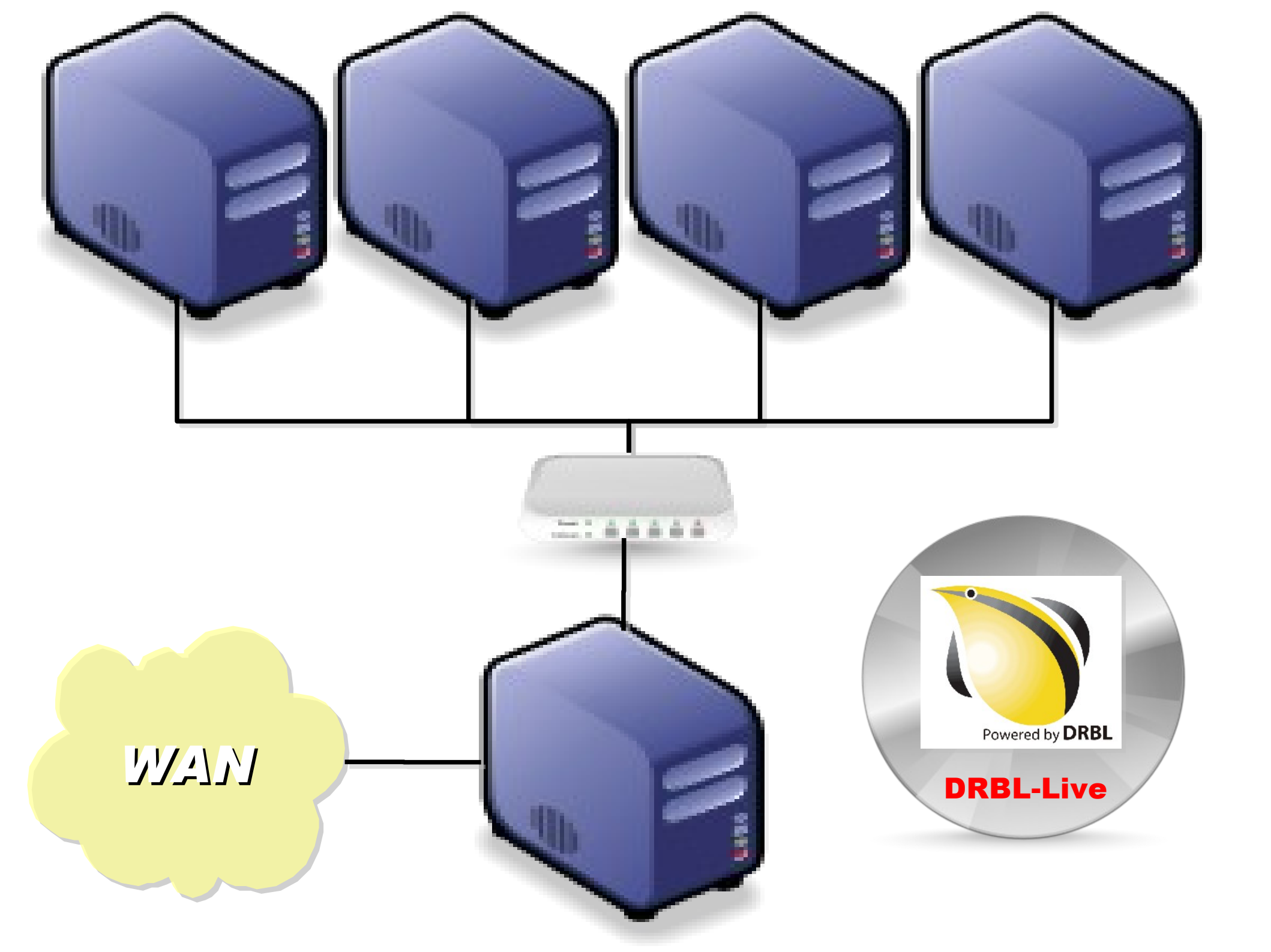
Live Demo

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw




WAN
Powered by DRBL
DRBL-Live

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