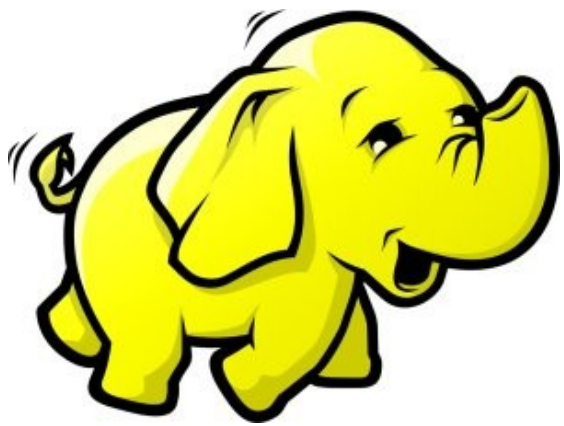
淺談海量資料的趨勢、挑戰與因應對策

Big Data : the Trends, Challenges and Solutions

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



Agenda 演講大綱

What is Big Data ? 何謂海量資料

Why should we care? 爲何需要關切

When to deploy it ? 何時導入技術

How to handle it ? 三大因應策略

Who is key player ? 誰是成功關鍵

WHAT

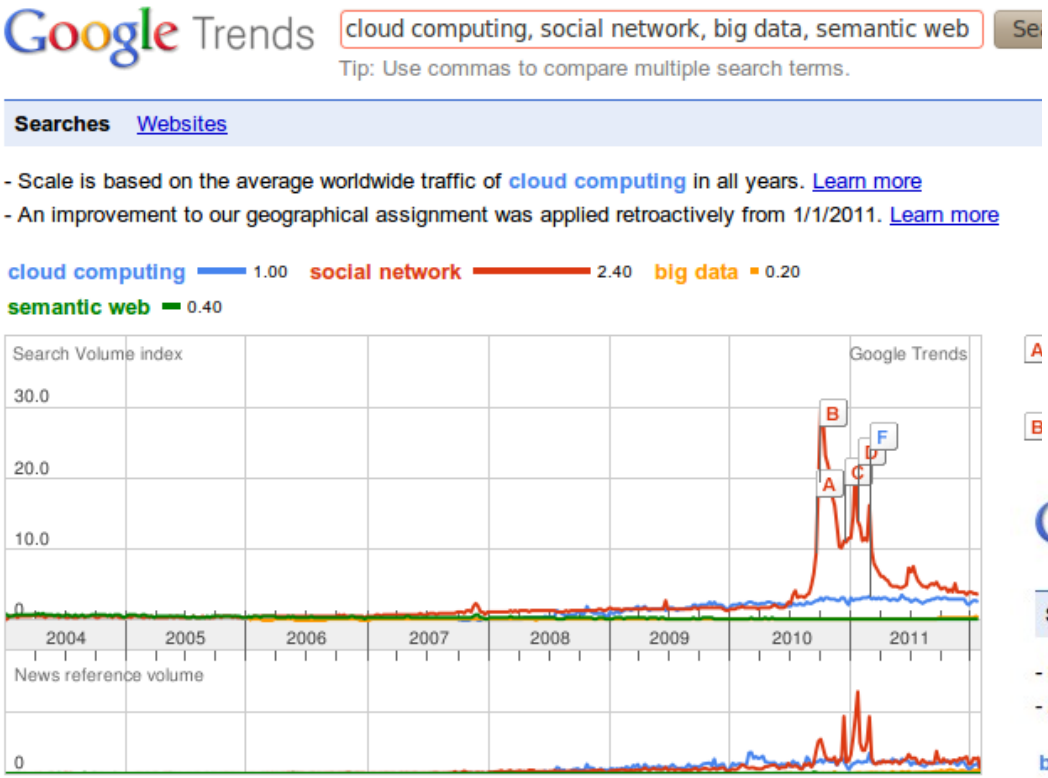
What is Big Data ?　何謂海量資料

趨勢 Trends → 定義 Definitions → 挑戰：管理維度 The Six Dimensions

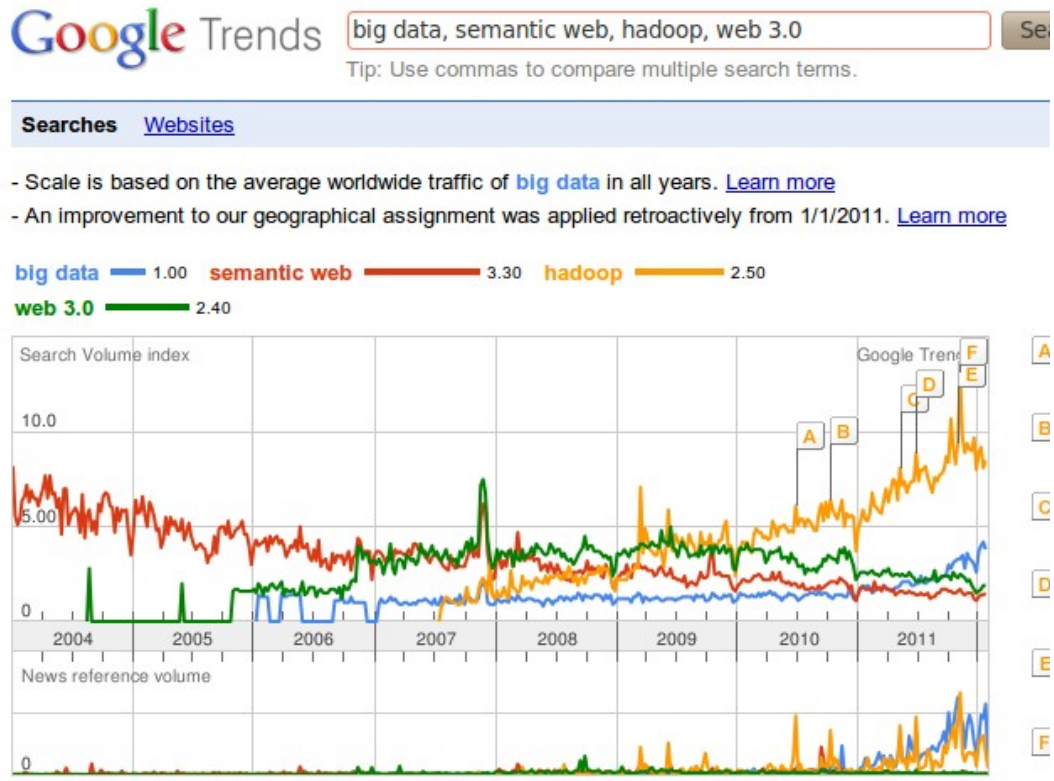
Source: http://www.2010taipeiexpo.tw/ct.asp?xItem=17186&CtNode=5952&mp=3

Trends …. It's all about Buzzwords ….. 「趨勢」亦或「流行語」？

Web 3.0, Cloud Computing, Social Network, Big Data, ….

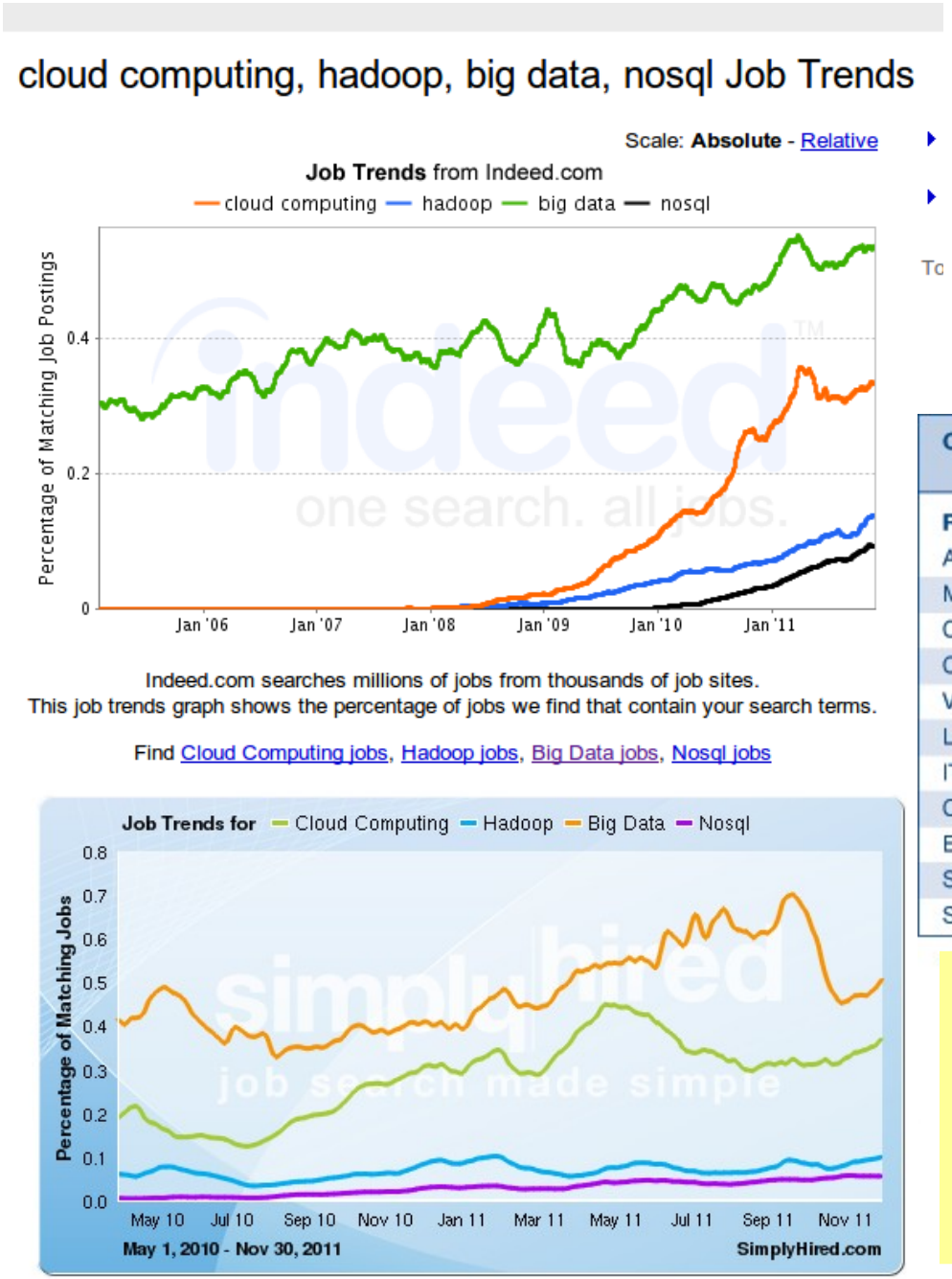


語意網（Semantic Web）從2001年開始制定標準後，逐漸下滑。而同義詞Web 3.0也呈現相似趨勢。海量資料（Big Data）與其關鍵技術Hadoop，則仍在上揚中。



整體而言，雲端運算（Cloud Computing）與社交網路（Social Network）呈現上揚。且社交網路比雲端運算還引人注目。

Trends of Market Needs 市場需求趨勢



美國軟體就業市場分析，根據 indeed 與 simply hired 兩間公司的趨勢觀察，都得到一樣的結果：

Big Data > Cloud Computing > Hadoop > NoSQL

CIO technologies	Ranking of technologies CIOs selected as one of their top 3 priorities in 2012			
Ranking	**2012**	**2011**	**2010**	**2009**
Analytics and business intelligence	1	5	5	1
Mobile technologies	2	3	6	12
Cloud computing (SaaS, IaaS, PaaS)	3	1	2	16
Collaboration technologies (workflow)	4	8	11	5
Virtualization	5	2	1	3
Legacy modernization	6	7	15	4
IT management	7	4	10	*
Customer relationship management	8	18	*	*
ERP applications	9	13	14	2
Security	10	12	9	8
Social media/Web 2.0	11	10	3	15

Gartner CIO Agenda 2012 前三名：

[1] Business Intelligence (Big Data)

[2] Mobile technology

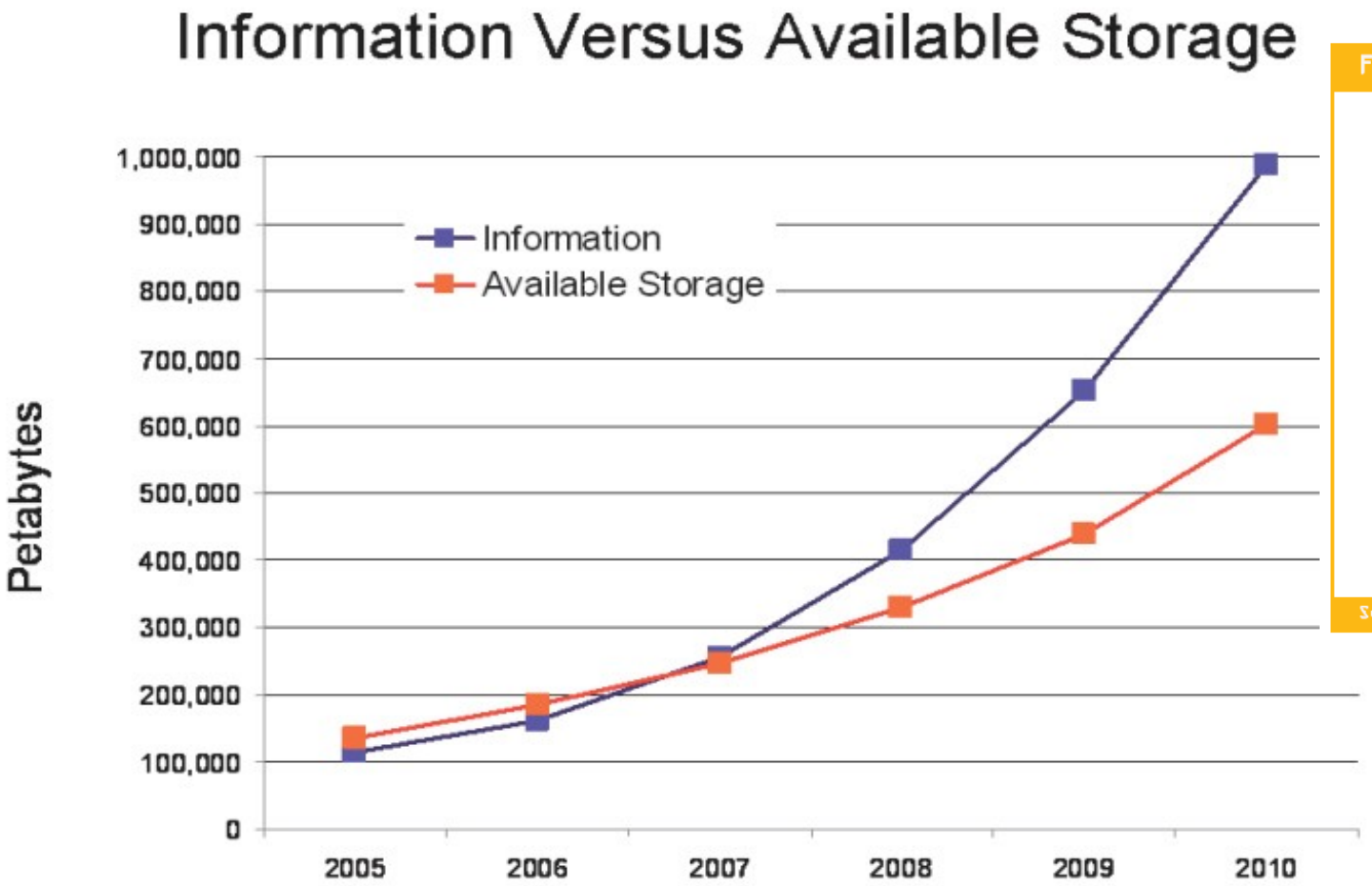
[3] Cloud Computing

參考來源：http://www.gartner.com/DisplayDocument?id=1524714

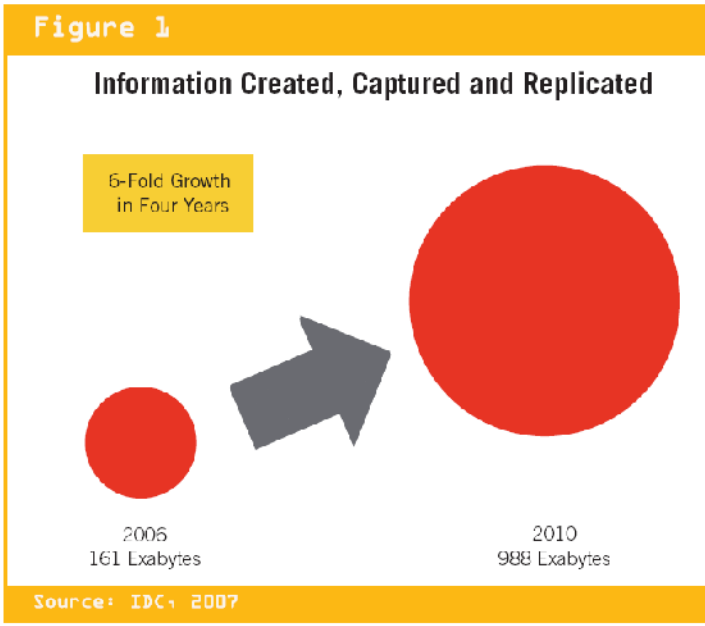
How BIG? 讓我們先來認識一下容量單位

Bit (b)	1 or 0
Byte (B)	8 bits
Kilobyte (KB)	1,000 bytes
Megabyte (MB)	1,000 KB
Gigabyte (GB)	1,000 MB
Terabyte (TB)	1,000, GB
Petabyte (PB)	1,000 TB
Exabyte (EB)	1,000 PB
Zettabyte (ZB)	1,000 EB

Data Explosion!! 始於 2007 的「資料大爆炸」時代



Source: IDC, 2007

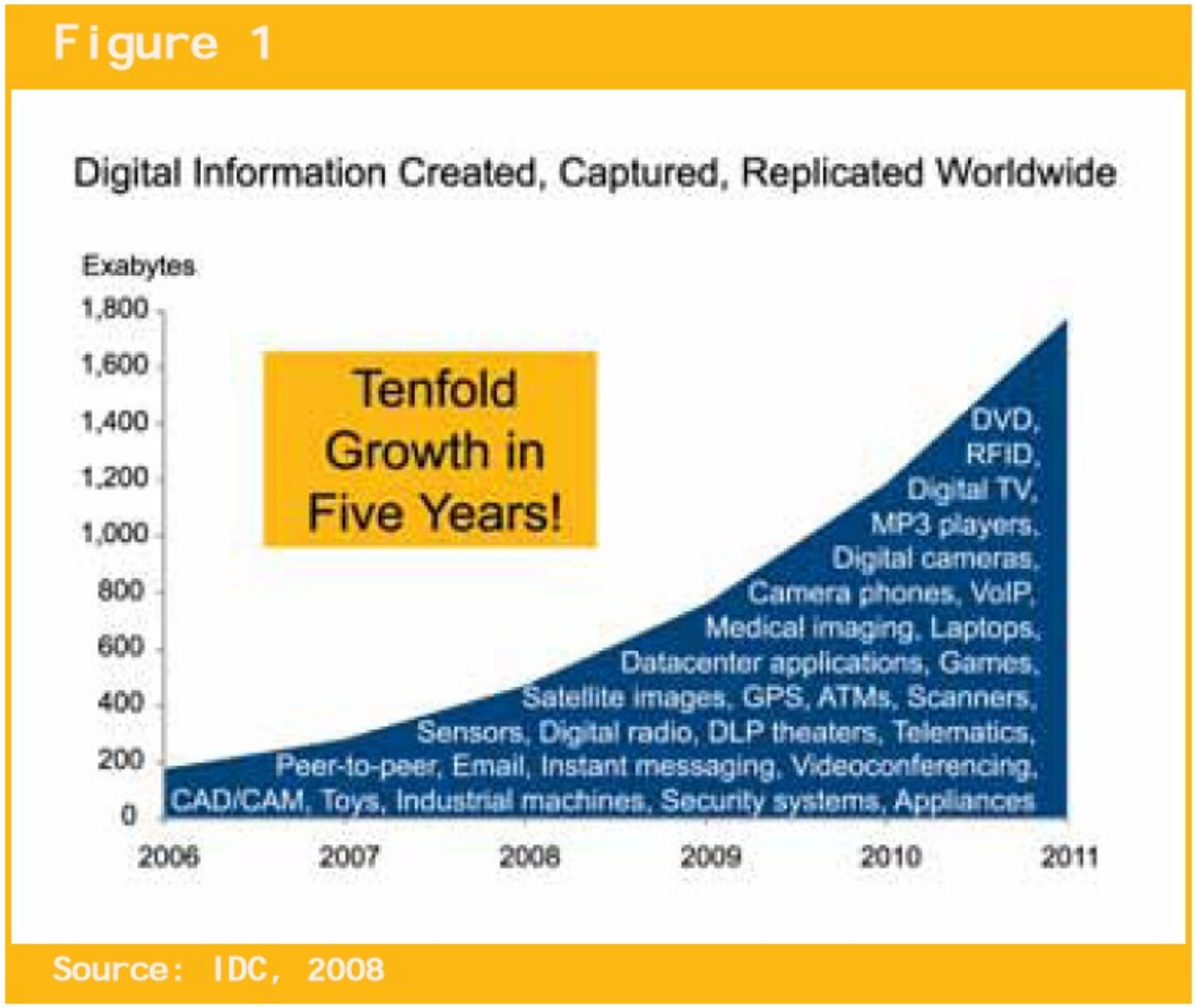


2007 年，IDC 預估
2010 年會成長六倍！
（相較 2006 年）

2006 161 EB
2010 988 EB（預測）

出處：The Expanding Digital Universe,
A Forecast of Worldwide Information Growth Through 2010,
March 2007, An IDC White Paper - sponsored by EMC
http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf

Data Explosion!! 始於 2007 的「資料大爆炸」時代

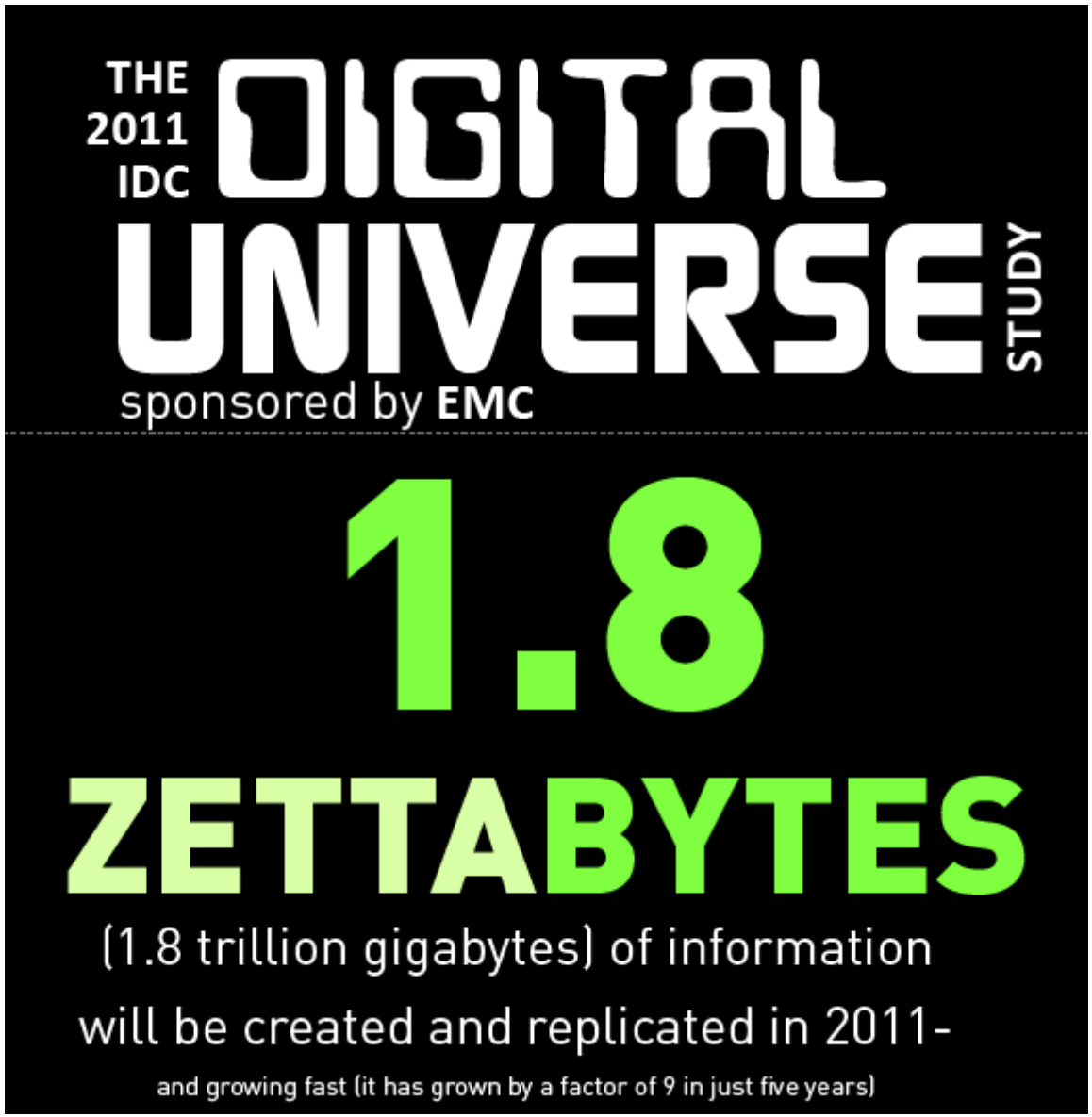


2009 年， IDC 預估
2011 年會成長十倍！
（相較 2006 年）

2006　161 EB
2007　281 EB
2010　988 EB（預測）
2011　1773 EB（預測）

出處：The Diverse and Exploding Digital Universe,
An Updated Forecast of Worldwide Information Growth Through 2011
March 2008, An IDC White Paper - sponsored by EMC
http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf

Data expanded 2x each year !! 每年約略兩倍



追蹤歷年的 IDC 數據：

2006	161	EB
2007	281	EB
2008	487	EB
2009	800	EB (0.8 ZB)
2010	988	EB (預測)
2010	1200	EB (1.2 ZB)
2011	1773	EB (預測)
2011	1800	EB (1.8 ZB)

景氣差而成長趨緩？
或受新技術抑制？

出處：Extracting Value from Chaos,
June 2011, An IDC White Paper - sponsored by EMC
http://www.emc.com/collateral/about/news/idc-emc-digital-universe-2011-infographic.pdf

What is Big Data?! 何謂『海量資料』？

海量資料泛指資料大小已無法用一般軟體擷取、管理與處理；
單一資料集大小介於數十 TB 至數 PB 的資料。

'Big Data' = few dozen TeraBytes to PetaBytes in single data set.

Definition [edit]

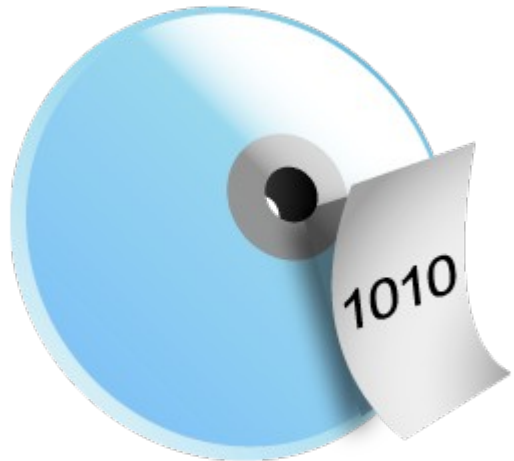
Big data is a term applied to data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target currently ranging from a few dozen terabytes to many petabytes of data in a single data set.

In a 2001 research report[14] and related conference presentations, then META Group (now Gartner) analyst, Doug Laney, defined data growth challenges (and opportunities) as being three-dimensional, i.e. increasing volume (amount of data), velocity (speed of data in/out), and variety (range of data types, sources). Gartner continues to use this model for describing big data.[15]

出處： http://en.wikipedia.org/wiki/Big_data

多個檔案，容量 100TB

一個資料庫，容量 100TB

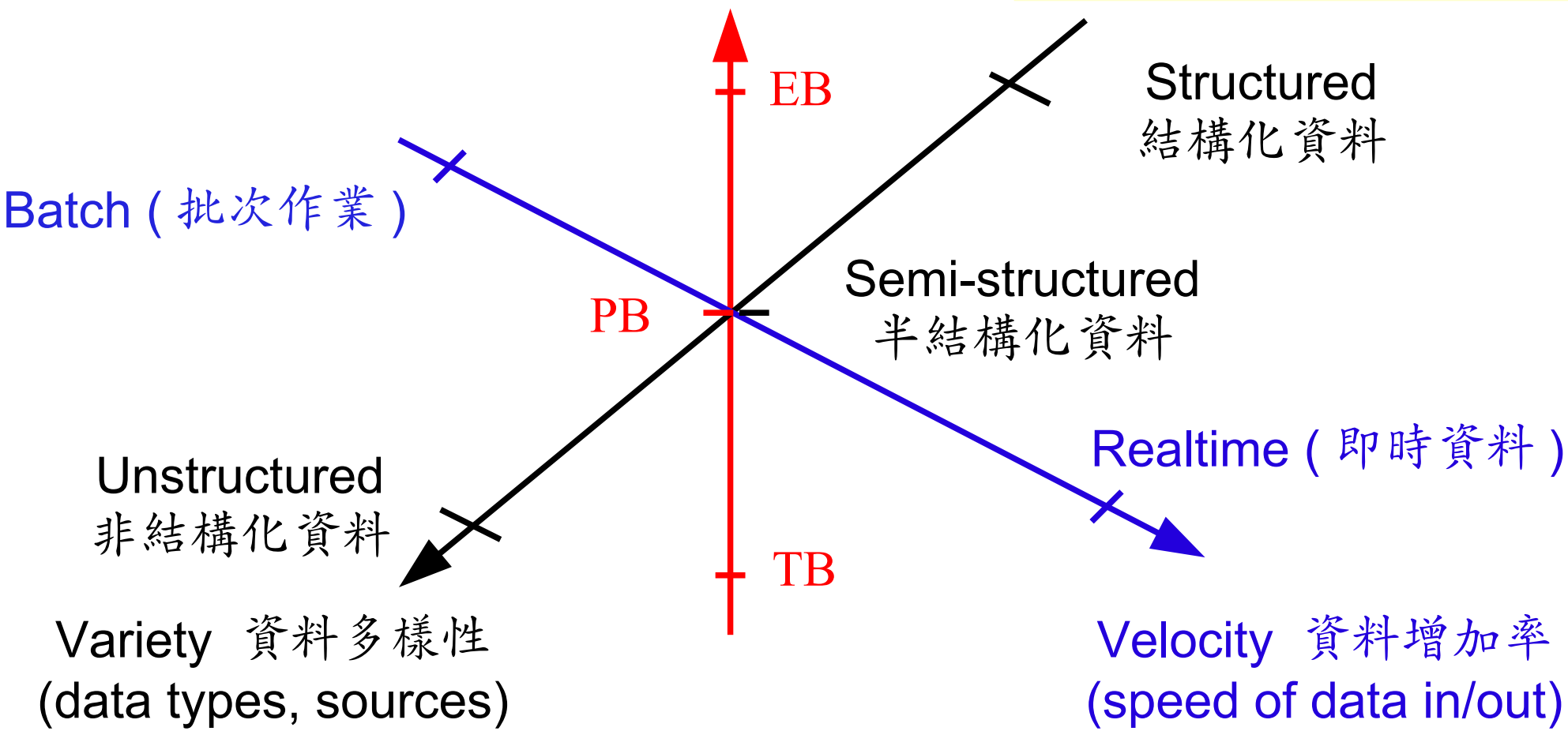


一個檔案，容量 100TB

Gartner Big Data Model？ 海量資料的模型？

海量資料的挑戰在於如何管理「數量」、「增加率」與「多樣性」

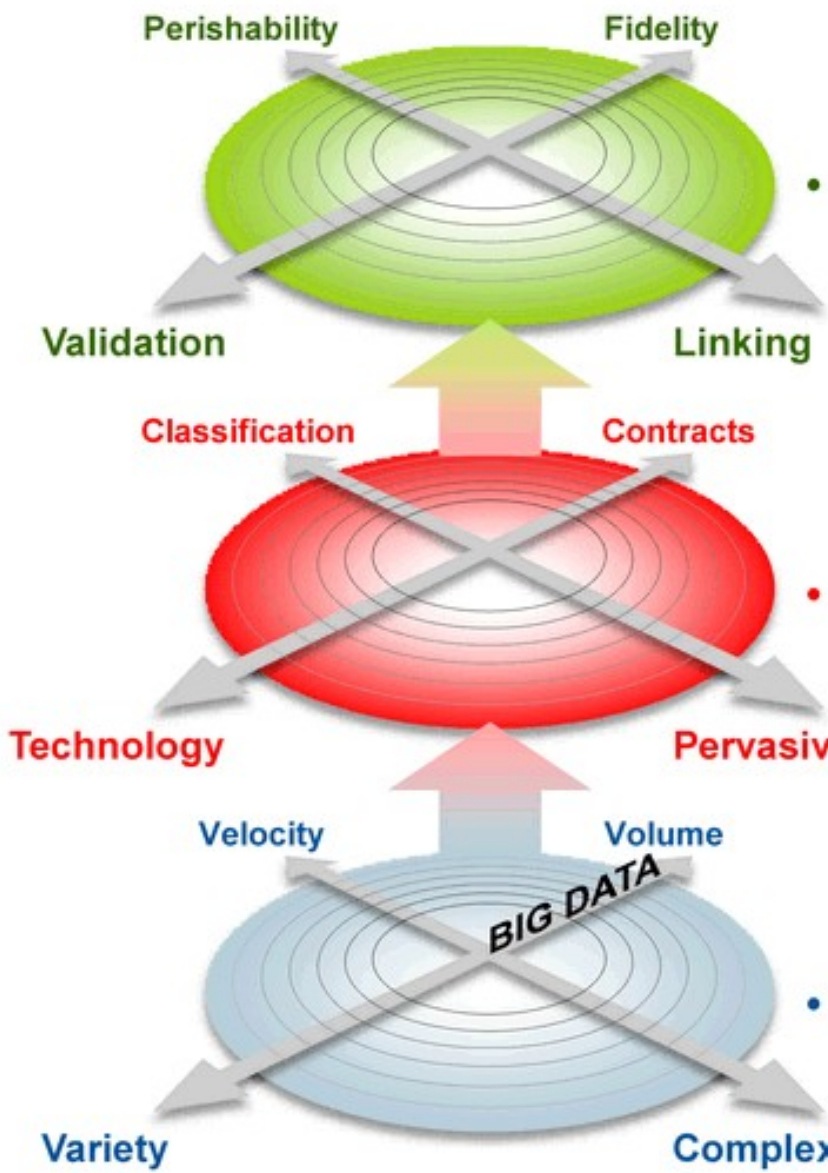
參考來源：
[1] Laney, Douglas. "3D Data Management: Controlling Data Volume, Velocity and Variety" (6 February 2001)
[2] Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data, June 2011



Six Dimensions of Big Data? 六個維度？

Source: Big Data, not Big Problems, http://www.talend.com/products-big-data/

12D of Information Management? 12 個維度？



品質管控

- Qualification and Assurance

權限管控

- Access Enablement and Control

數量管控

- Quantification

Big Data 只是終極資訊管理的開端！

Source: Gartner (March 2011), 'Big Data' Is Only the Beginning of Extreme Information Management, 7 April 2011, http://www.gartner.com/id=1622715

Agenda 演講大綱

What is Big Data ? 何謂海量資料

Why should we care? 爲何需要關切

資料 Data → 知識 Knowledge → 智慧 Wisdom

Why we call it " SMART " !!
智慧打哪兒來？！

Smart Phone
智慧手機

Smart Car
智慧車輛

Smart Grid
智慧電網

SMART
哪裡長
智慧了？

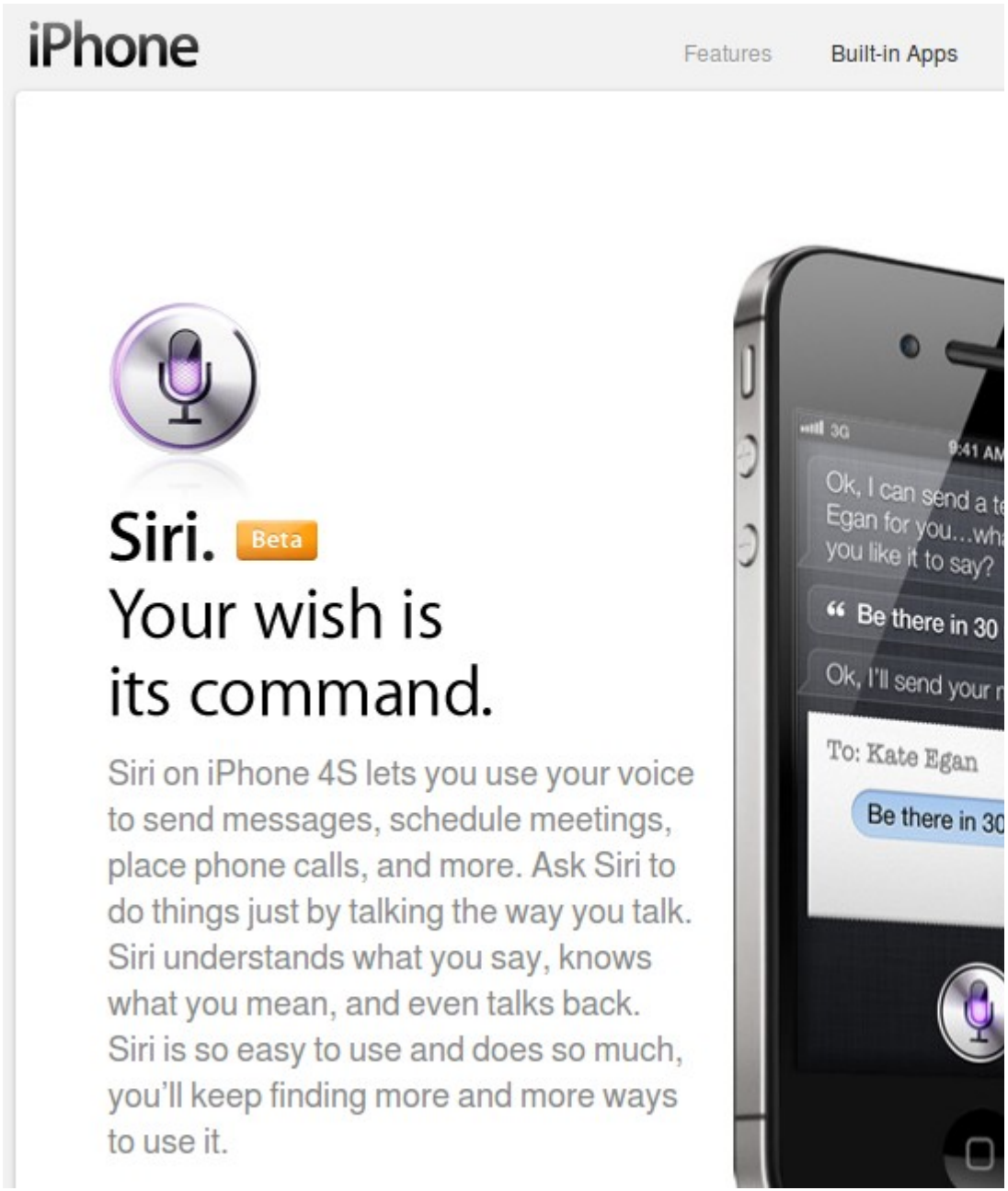
Smart City
智慧城市

Smart Home
智慧家庭

Smart Meter
智慧電錶

資料 Data → 知識 Knowledge → 智慧 Wisdom

Can Machine understand You? 讓機器更懂你？



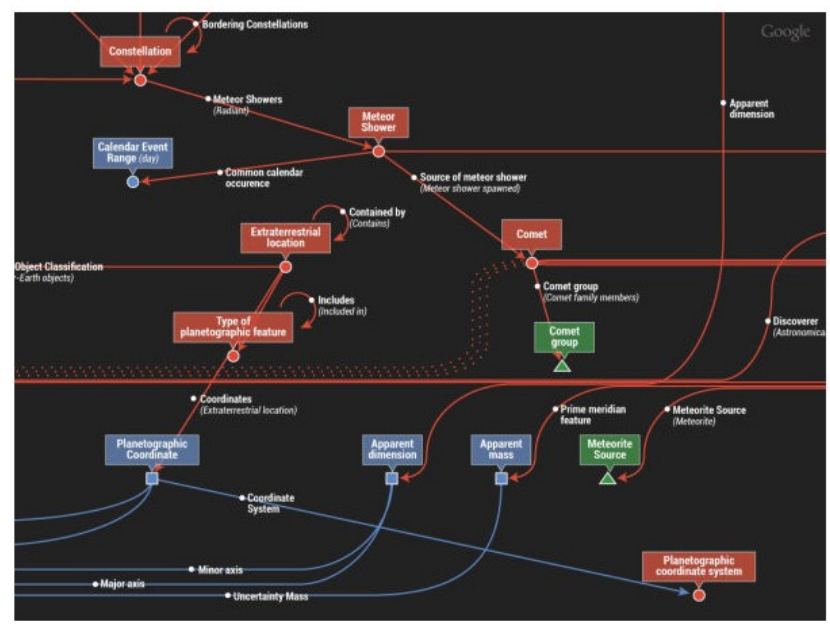
Google將發展「人工智慧」　永久改變搜尋引擎

2012年02月15日 00:11

點評：*超級阿斯拉，衝啊！（阿斯拉：好的，隼人！）*

記者黃郁棋／綜合報導

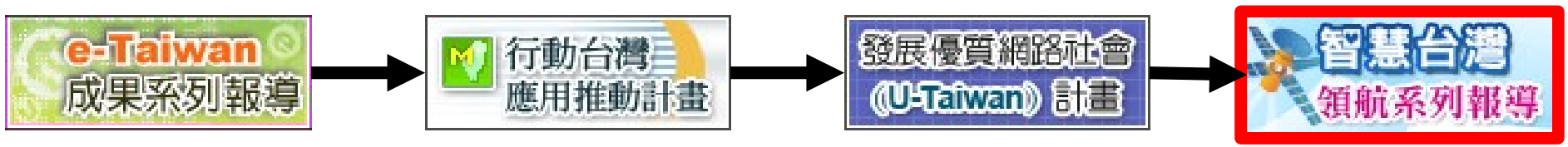
每個人都在猜，下一波網路革命是什麼？每個人都在猜，未來的世界會如何運作？Google的資深副總Amit Singhai透露了一點訊息。「Google正努力從『單字』層面進展到『意義』層面，未來搜尋引擎提供的不只是關鍵字搜尋，搜尋引擎甚至會『明白』你到底要什麼。」



▲Google未來將會朝「人工智慧」前進。(圖／取自mashable.com)

http://www.ettoday.net/news/20120215/25085.htm

Evolution of Software / Service
軟體演化勢必走向『智能化』



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

The wisdom of Clouds (Crowds)
雲端序曲：雲端的智慧始終來自於群眾的智慧

2006 年 8 月 9 日

Google 執行長施密特（Eric Schmidt）於SES '06會議中首次使用「雲端運算（Cloud Computing）」來形容無所不在的網路服務

2006 年 8 月 24 日

Amazon 以 Elastic Compute Cloud 命名其虛擬運算資源服務

Source: http://www.cnet.co.uk/i/c/blg/cat/software/cloudcomputing/clouds1.jpg

Data is the source of Wisdom !!
用雲掌握資料，加以分析，形成智能給端用

雲

資料中心
提供服務

雲端設計新思維：端的智能來自於雲的服務

Devices share the wisdom of Cloud

端

各類裝置
存取服務

Agenda 演講大綱

- What is Big Data ? 何謂海量資料
- Why should we care? 爲何需要關切
- **When** to deploy it ? 何時導入技術

基礎建設 IaaS → 分析平台 PaaS → 智慧服務 SaaS

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.
主要客户爲中小企業

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

以大型企業爲主要客户
Enterprise is key market

私有雲端
Private Cloud

Community Cloud
社群雲端

Academia 學術爲主

3 Service Models of Cloud Computing
雲端運算的三種服務模式（市場區隔）

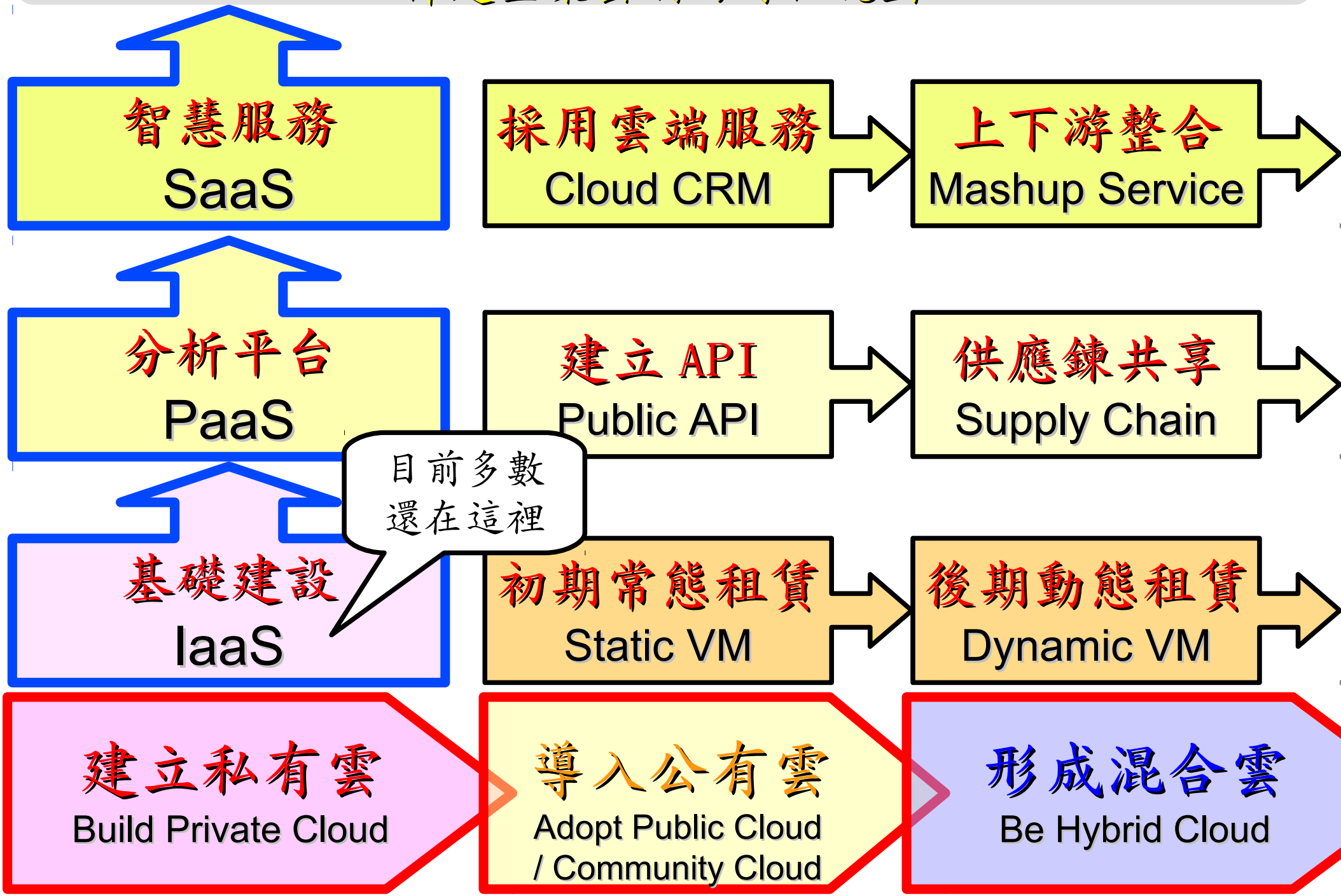
IaaS
Infrastructure as a Service
架構即服務

PaaS
Platform as a Service
平台即服務

SaaS
Software as a Service
軟體即服務

Roadmap to build Your Enterprise Cloud !!

佈建企業雲端的時程規劃



Agenda 演講大綱

What is Big Data ? 何謂海量資料

Why should we care? 爲何需要關切

When to deploy it ? 何時導入技術

How to handle it ? 三大因應策略

儲存虛擬化 Dedup. → 資料安全 Security → 智慧服務 SaaS

Three Solutions !! 三種服務模式 vs. 三類因應對策

SaaS Software as a Service 軟體即服務	**Web 2.0** 網頁服務	(A) 提供 API 介面 (B) 分散式資料庫
PaaS Platform as a Service 平台即服務	**Data Analysis** 資料分析	(A) 資料整合 (B) 資料探勘
IaaS Infrastructure as a Service 架構即服務	**Virtualization** 虛擬化技術	(A) 儲存虛擬化 (B) 備援與加密

Agenda 演講大綱

- What is Big Data ? 何謂海量資料
- Why should we care? 爲何需要關切
- When to deploy it ? 何時導入技術
- How to handle it ? 三大因應策略
- **Who** is key player ? 誰是成功關鍵

Data Scientist !! 資料科學家！！

Data scientist: The hot new gig in tech

By Michal Lev-Ram, writer September 6, 2011: 5:00 AM ET

Companies that want to make sense of all their bits and bytes are hiring so-called data scientists - if they can find any.

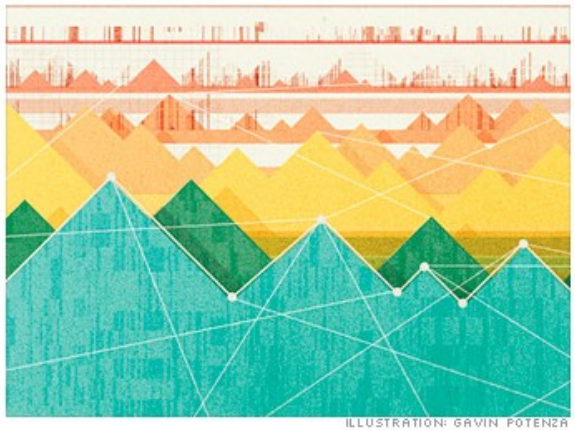


ILLUSTRATION: GAVIN POTENZA

FORTUNE -- The unemployment rate in the U.S. continues to be abysmal (9.1% in July), but the tech world has spawned a new kind of highly skilled, nerdy-cool job that companies are scrambling to fill: data scientist.

會「統計」的人照過來！

財星雜誌(FORTUNE)等均報導今年最熱門的職缺是「資料科學家」！

Source :
http://tech.fortune.cnn.com/2011/09/06/data-scientist-the-hot-new-gig-in-tech/

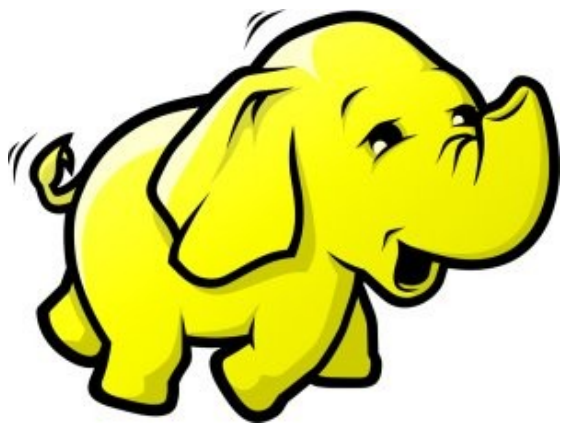
處理海量資料的資訊架構與關鍵技術

Technologies to build IT Stack for Big Data

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



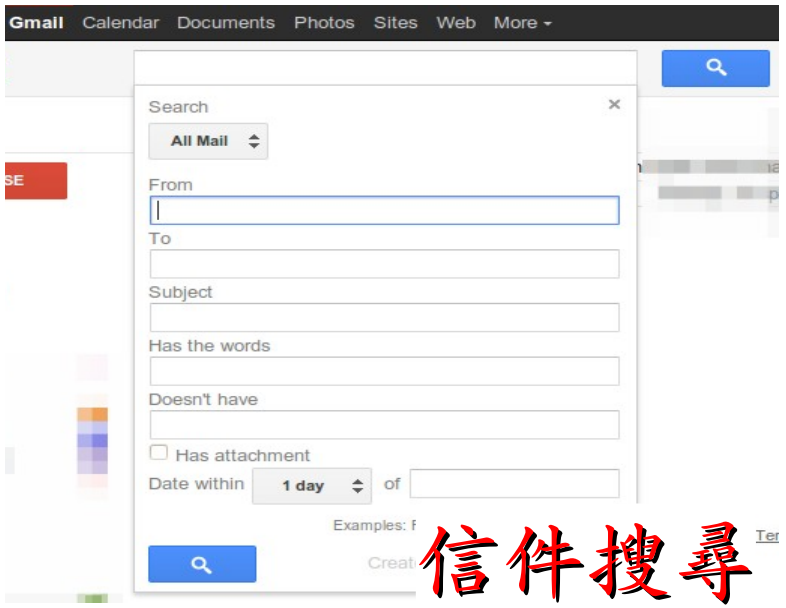
Hot Jobs in Big Data
從海量資料的熱門工作談起

Data Mining	資料探勘
Data Visualization	資料視覺化
Data Analysis	資料分析
Data Manipulation	資料操控
Data Discovery	資料鑑識

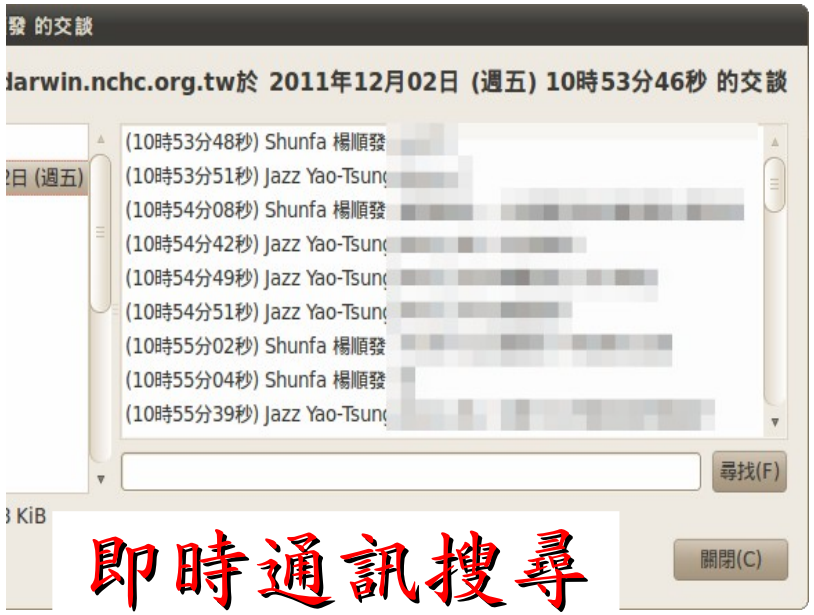
How to Get a Hot Job in Big Data, Dan Tynan, InfoWorld, March 19, 2012
出處：http://www.cio.com/article/print/702388

Applications of Data Mining
資料探勘的應用 - 搜尋引擎

檔案搜尋



信件搜尋

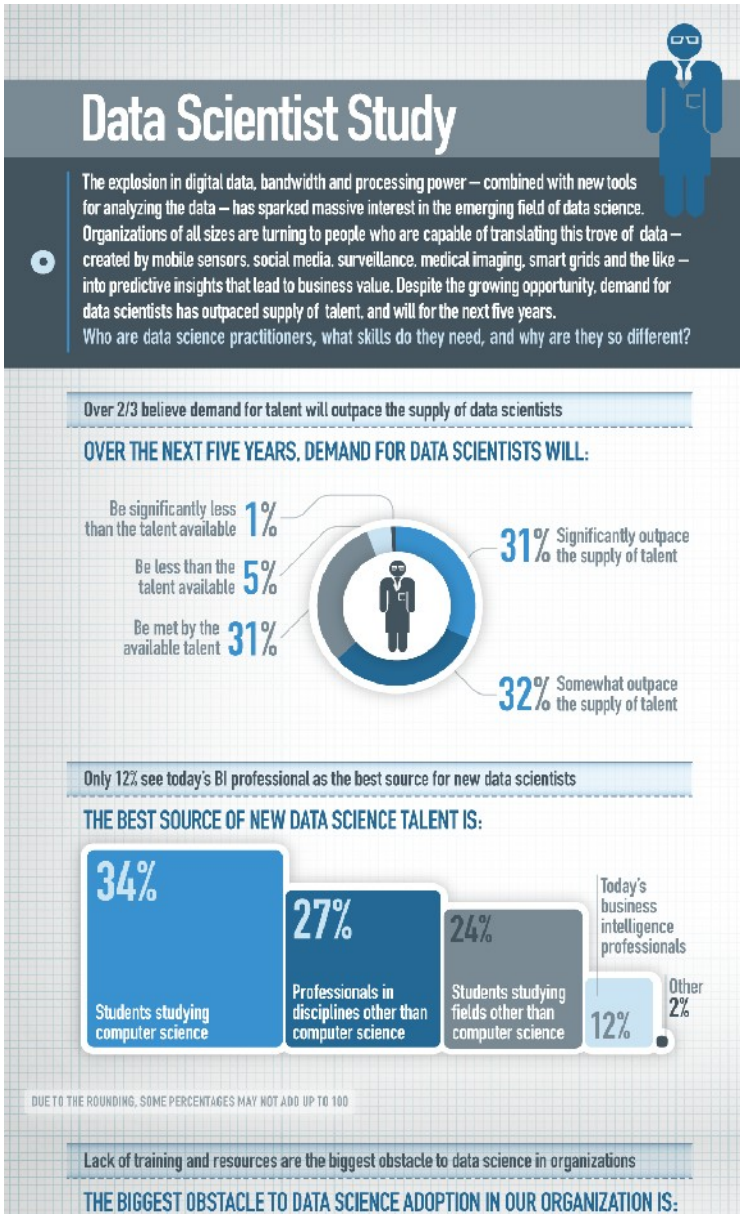


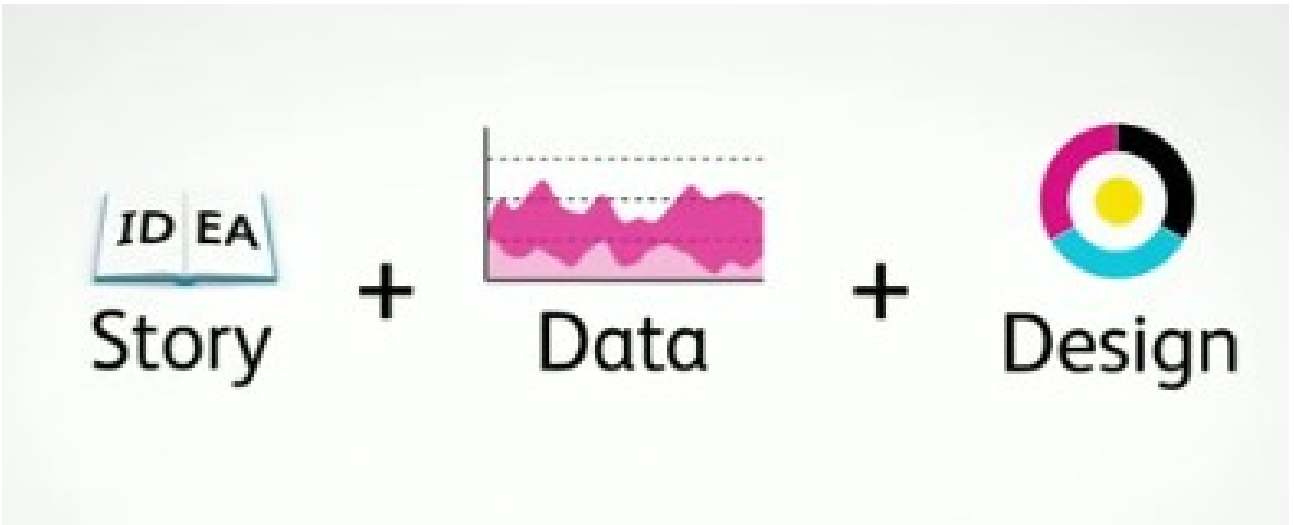
即時通訊搜尋

資料庫搜尋

「網頁搜尋」

Applications of Data Visualization
資料視覺化的應用 -Infographics





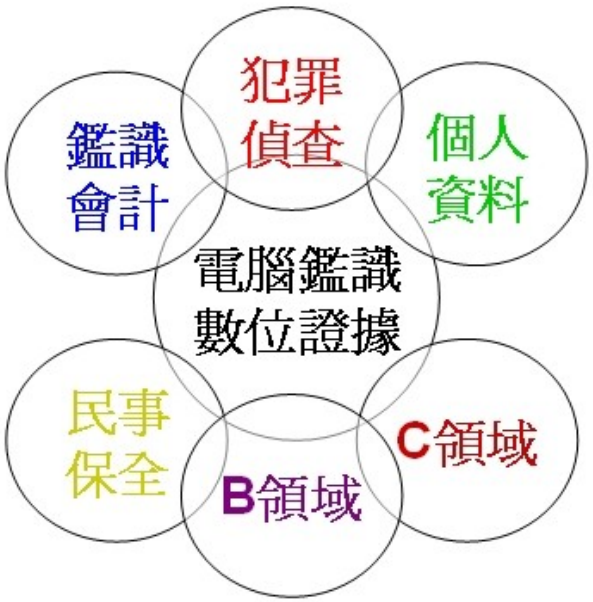
參考來源：未來「夯」職業：資料科學家 http://www.bnext.com.tw/print/article/id/21740
淺談超吸睛的資訊圖表 http://www.inside.com.tw/2011/04/13/infographics

Applications of Data Analysis
資料分析的應用 - 商業智慧 (BI)

圖片來源： http://bizcubic.com/wp-content/uploads/2010/01/business-intelligence-dashboard.jpg

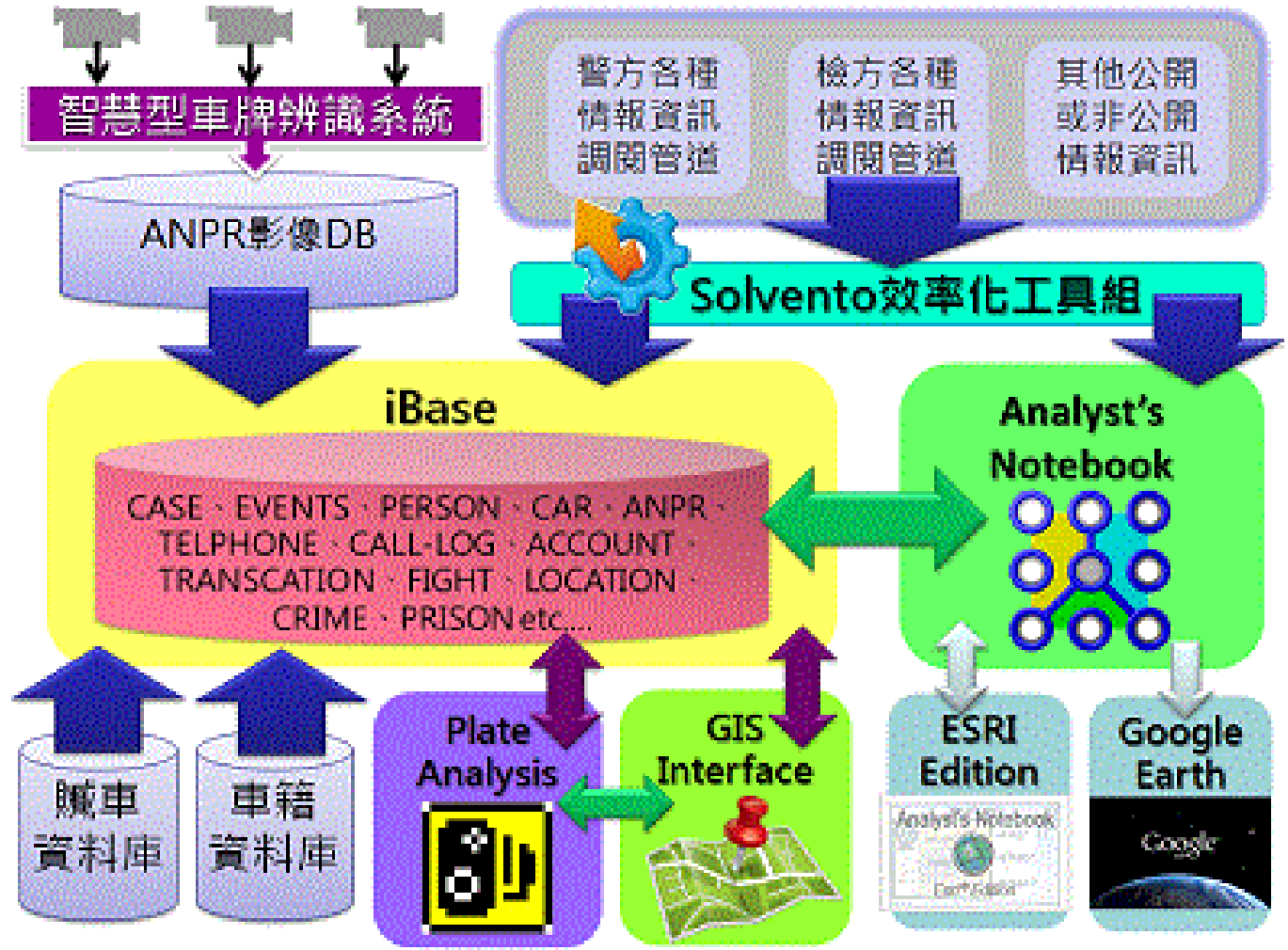
Applications of Data Discovery
數位鑑識 - 資訊與法律的結合

電腦鑑識&會計鑑識

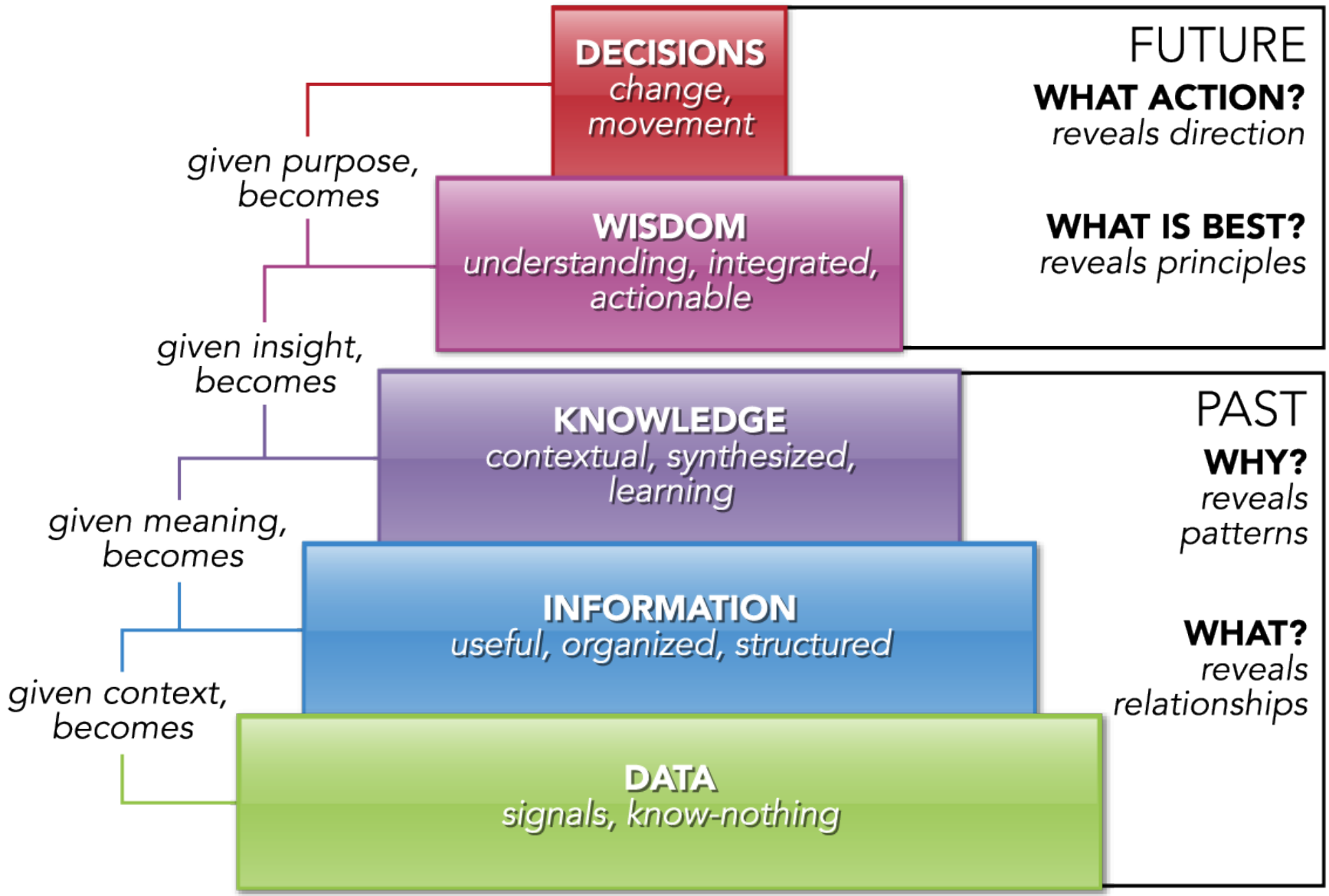


http://blog.udn.com/kf0630/6018593

http://www.solventosoft.com/upload/ANPR_02s.gif



Data, Information, Knowledge, Wisdom
知識管理模型：資料、資訊、知識與智慧

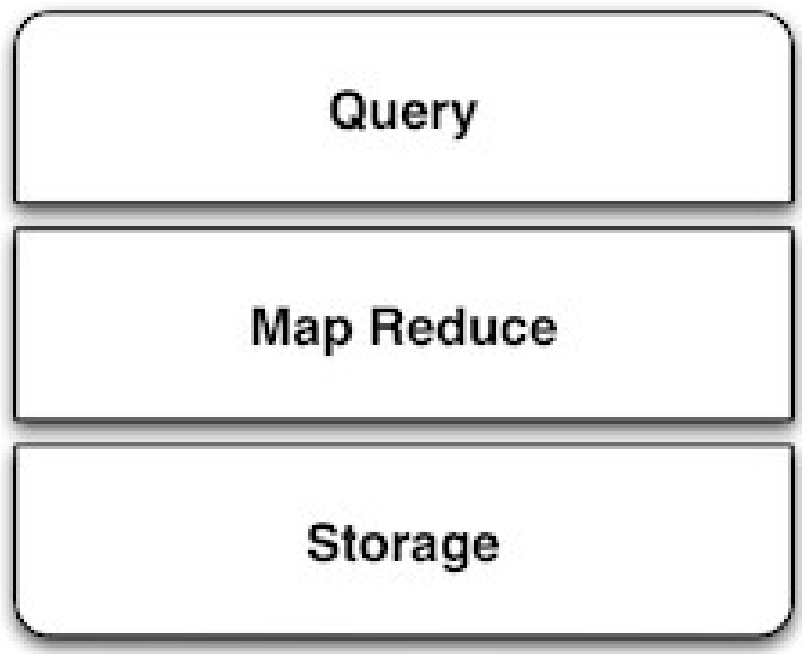


http://www.pursuantgroup.com/blog/tag/dikw-model/

The SMAQ stack for big data
海量資料處理的資訊架構

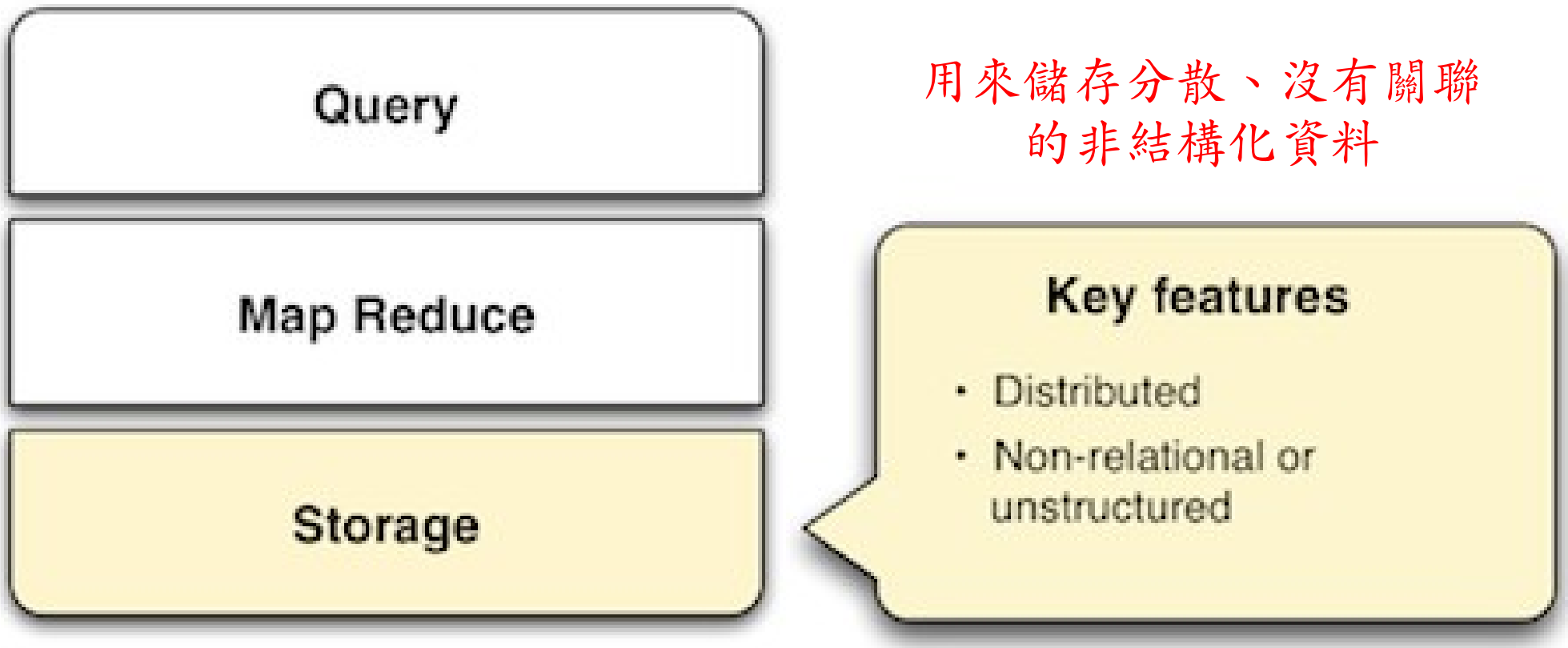
做網頁相關的人可能聽過 LAMP

未來處理海量資料的人必需知道 SMAQ（Storage, MapReduce and Query）



參考來源：The SMAQ stack for big data，Edd Dumbill，22 September 2010，
http://radar.oreilly.com/2010/09/the-smaq-stack-for-big-data.html
圖片來源：http://smashingweb.ge6.org/wp-content/uploads/2011/10/apache-php-mysql-ubuntu.png

The SMAQ stack for big data
海量資料處理的資訊架構

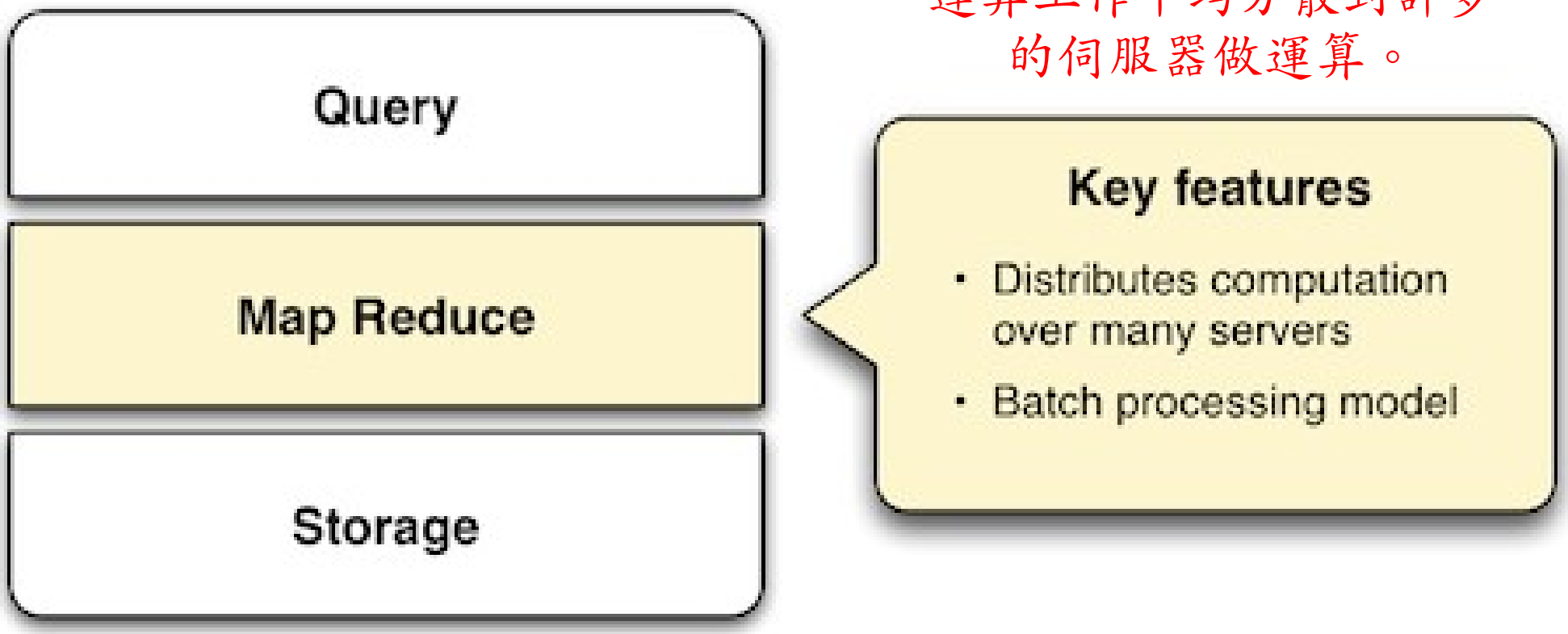


用來儲存分散、沒有關聯的非結構化資料

參考來源：The SMAQ stack for big data，Edd Dumbill，22 September 2010，http://radar.oreilly.com/2010/09/the-smaq-stack-for-big-data.html

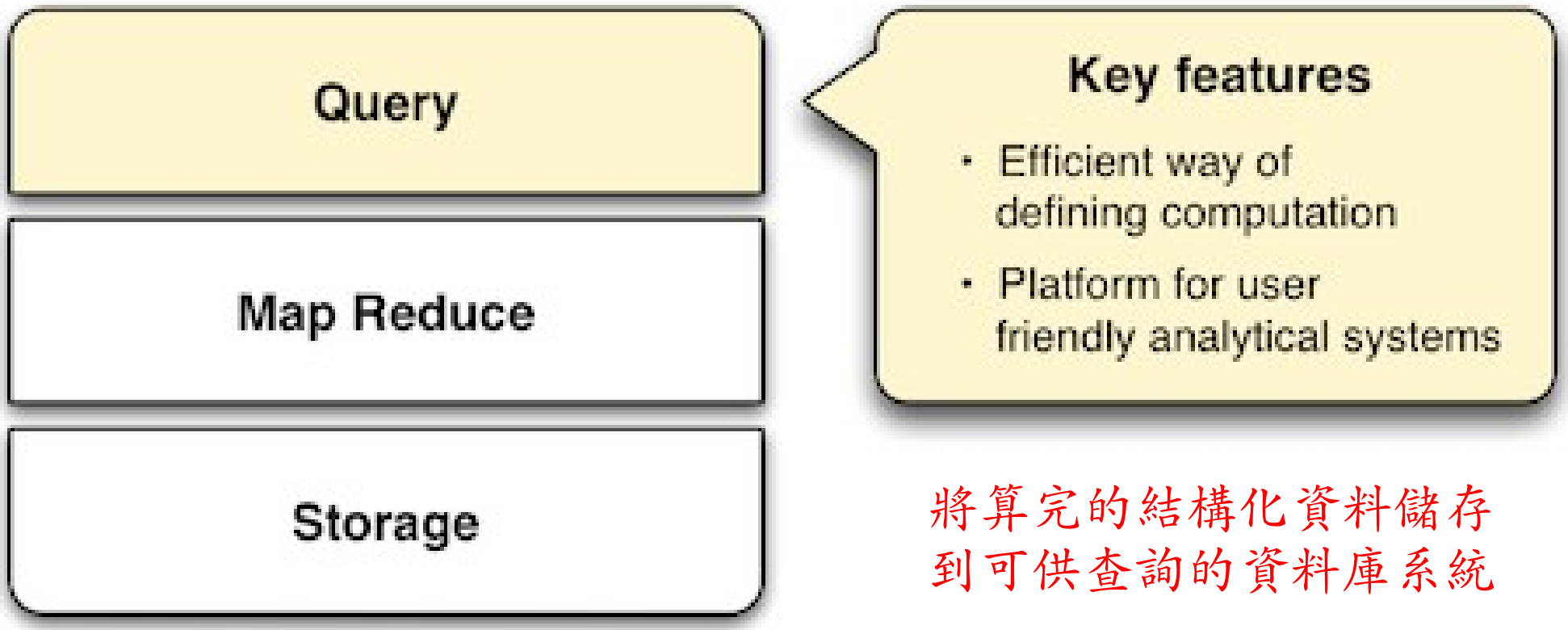
The SMAQ stack for big data
海量資料處理的資訊架構

運用批次處理的方式，將運算工作平均分散到許多的伺服器做運算。



參考來源：The SMAQ stack for big data，Edd Dumbill，22 September 2010，http://radar.oreilly.com/2010/09/the-smaq-stack-for-big-data.html

The SMAQ stack for big data
海量資料處理的資訊架構



將算完的結構化資料儲存到可供查詢的資料庫系統

參考來源：The SMAQ stack for big data，Edd Dumbill，22 September 2010，http://radar.oreilly.com/2010/09/the-smaq-stack-for-big-data.html

Three Core Technologies 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 : Simplifed 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

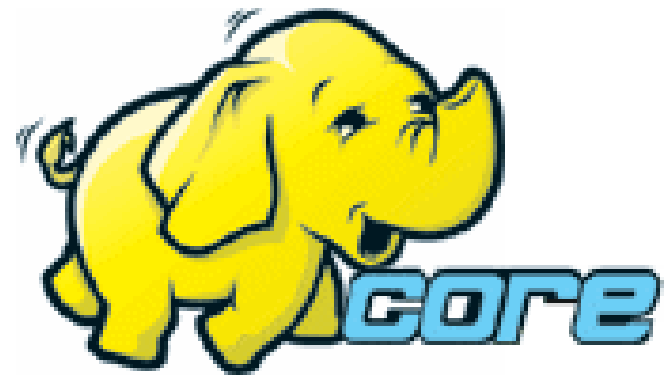
更多不同語言的 MapReduce API 實作：
http://trac.nchc.org.tw/grid/intertrac/wiki%3Ajazz/09-04-14%23MapReduce

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/

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MalStone效能評比軟體
- **Open Cloud Consortium provide Open Cloud Testbed and develop MalStone toolkit for benchmark**

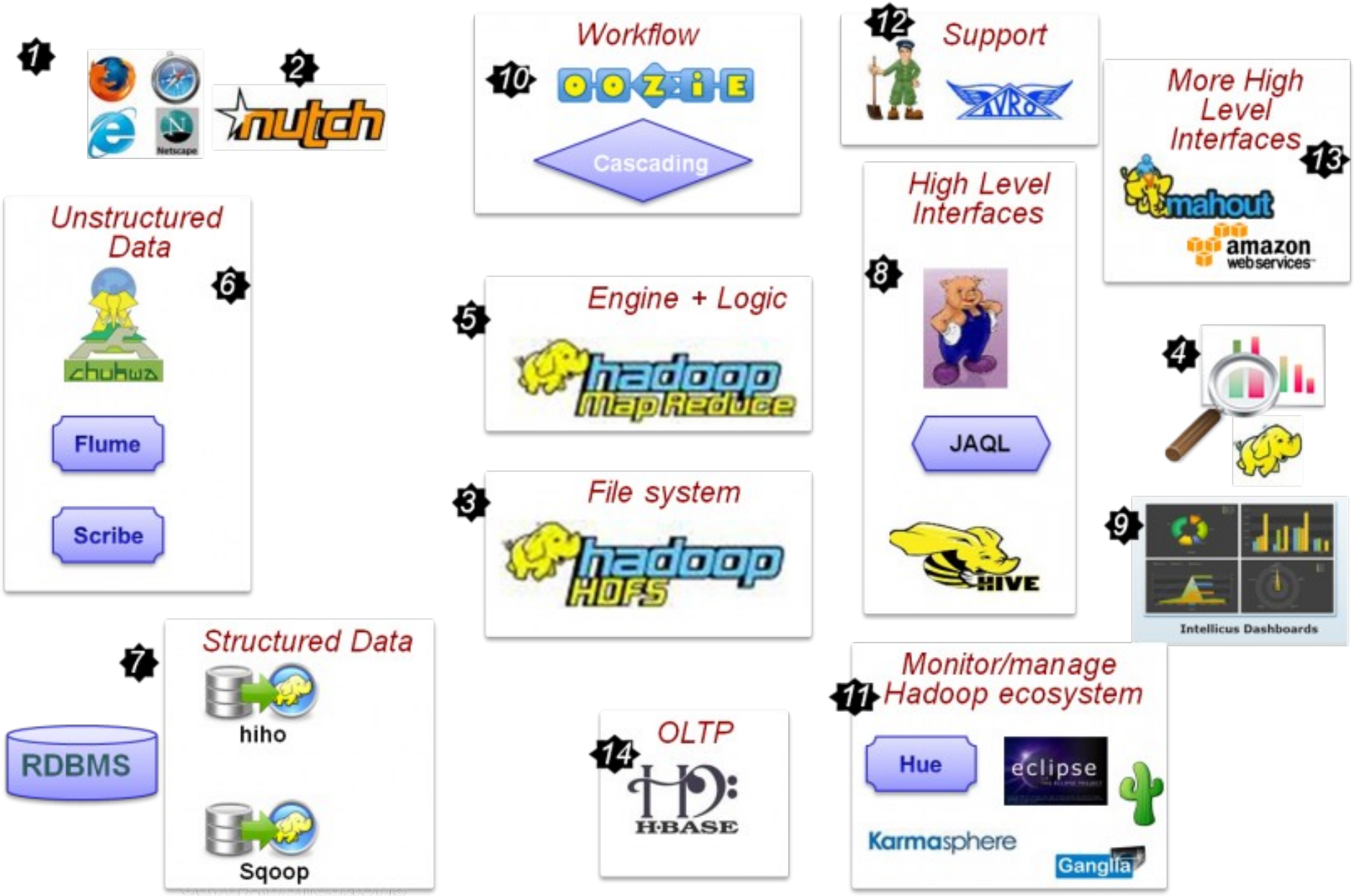
Sector-Sphere

National Center for Data Mining
University of Illinois at Chicago
UIC

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

Why we choice Hadoop? Good Ecosystem!

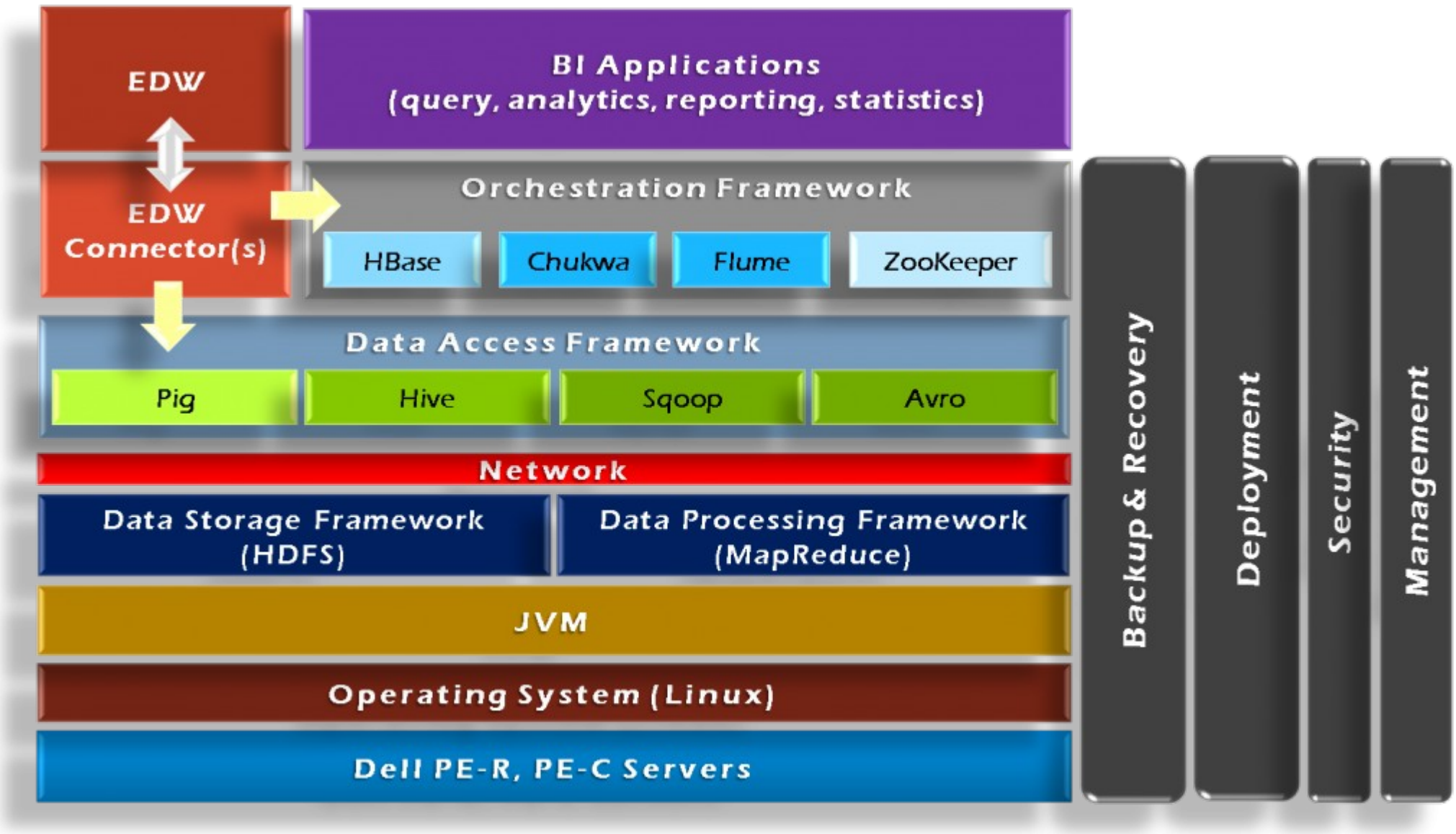
豐富的生態系建構出處理海量資料的工具庫



http://rationalintelligence.com/wp_log/?p=104

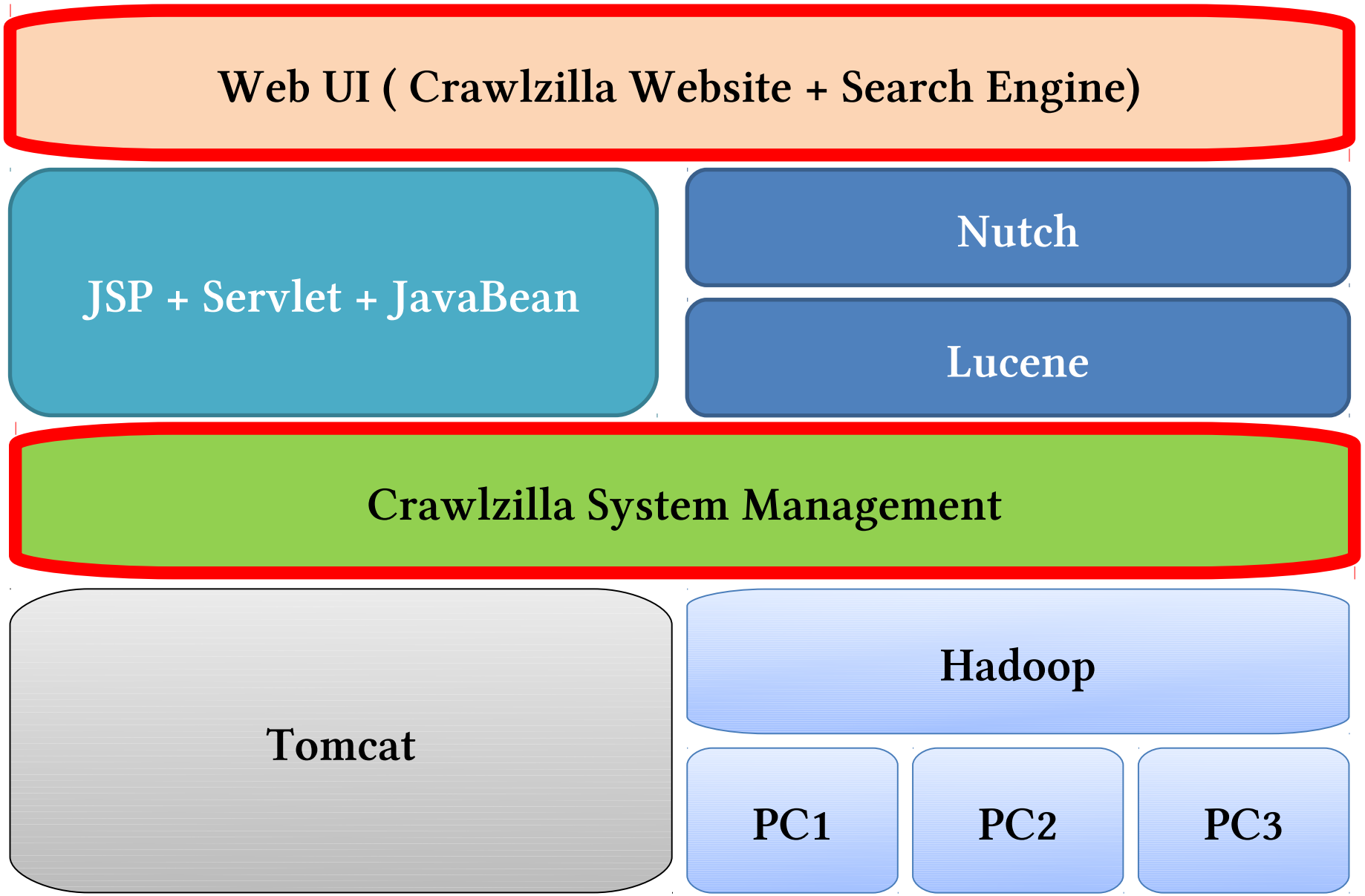
BI and EDW build on Hadoop Ecosystem

運用 Hadoop 生態系搭建資料倉儲與商業智慧分析

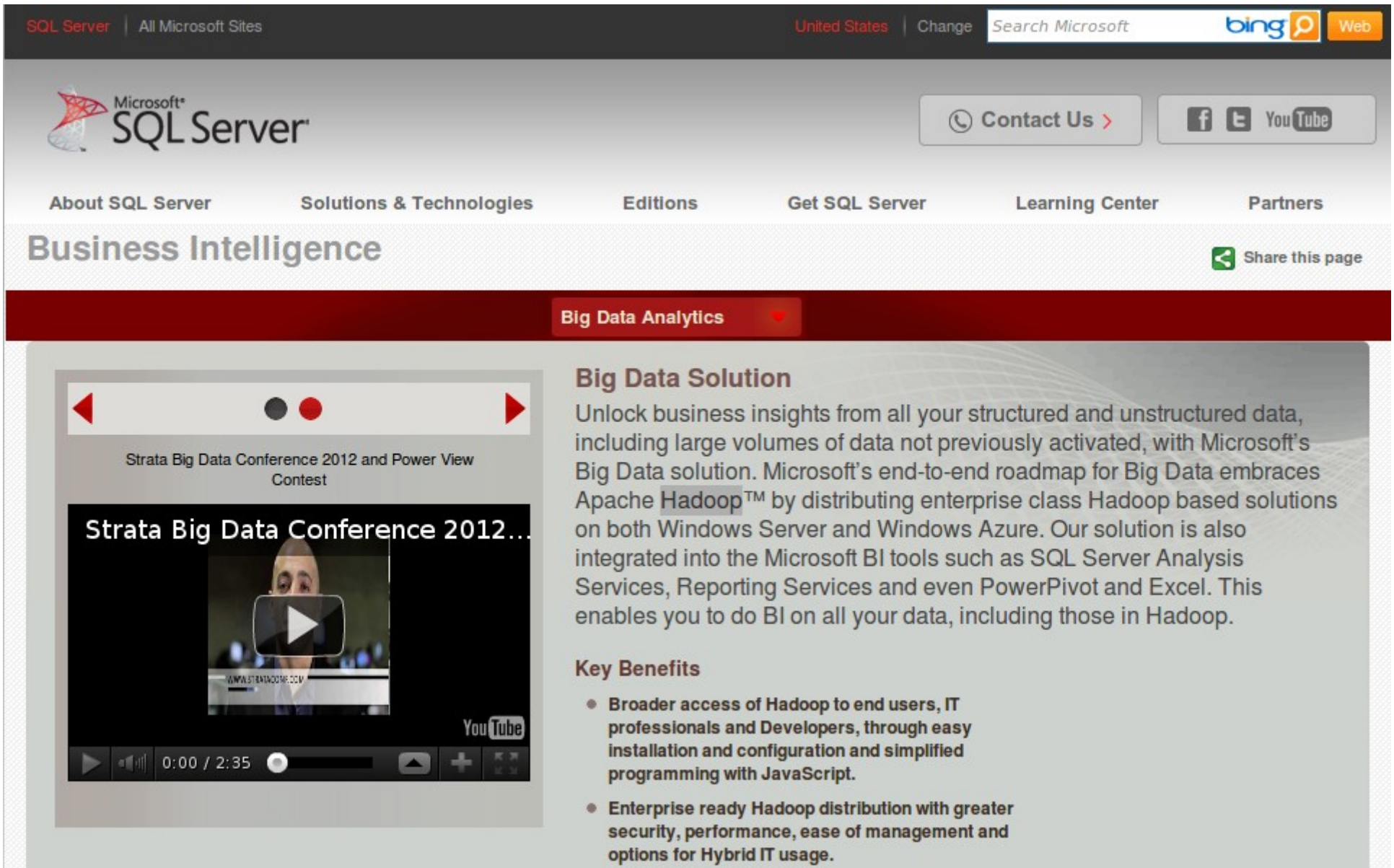


http://rationalintelligence.com/wp_log/?p=104

Build your own search engine, too
您也能用 Hadoop 搭建自己的搜尋引擎



Microsoft love Hadoop, too
微軟幫 Azure 還有 SQL Server 都接上 Hadoop



參考來源：Big Data Solution | Microsoft SQL Server 2008 R2
http://www.microsoft.com/sqlserver/en/us/solutions-technologies/business-intelligence/big-data-solution.aspx

Oracle love Hadoop, too
Oracle 也接上 Hadoop

CNET › News › Software, Interrupted

Cloudera teams up to connect Oracle and Hadoop

Cloudera and Quest software are partnering to provide connectivity between Oracle and Hadoop.

by Dave Rosenberg | June 21, 2010 5:30 AM PDT

Follow

This week Cloudera, a provider of software and services for the Apache Hadoop project, is set to announce a partnership with Quest Software to develop, support, and distribute an Oracle connector for Hadoop.

參考來源：Cloudera teams up to connect Oracle and Hadoop
http://news.cnet.com/8301-13846_3-20008242-62.html

Hinet Application of Big Data
中華電信已經在做的海量資料應用

Business Next 數位時代

中華電信：分析駭客行為，拓展對外新服務

撰文者：趙郁竹

發表日期：2012-03-06

[214期雜誌精選]

全台最大的中華電信提供行動電話、市話、寬頻固網、MOD……，各種業務服務，加起來的用戶數就有3000萬，比全台灣人口還多，光是單月帳務數量就高達100億筆資料。除了電信、寬頻服務，還有日益增加的數位服務、行動加值服務，從服務內容到客戶端，累積出的資料相當驚人。

「資料量越來越大，日常分析工作需要很多時間，但新的運算技術有效解決了這個問題，」中華電信資訊處處長陳明仕說。2010年開始，因為中華電信本身的資料運算需求，採用分散式運算架構Hadoop技術，打造出大資料運算平台，不但解決了自身的資料問題，還能對外提供資料運算應用。

以MOD為例，一天有幾千萬筆資料，如何找出使用者在什麼時段做了什麼事？廣告效益又如何？「用傳統的方法，需要400分鐘才能分析完；用Hadoop大資料平台，13分鐘就能解決，節省非常多時間，」他說。

追蹤再拆解

大資料運算技術除了節省時間，還能防止駭客入侵。「駭客的攻擊行為都有模式可循，」陳明仕解釋，就像球賽一樣，了解進攻模式就能防守。用戶的資料保護是第一要務，因此透過行為模式分析，能有效保護企業資訊安全，也保障客戶的個資安全。

參考來源：中華電信：分析駭客行爲，拓展對外新服務，發表日期： 2012-03-06
http://www.bnext.com.tw/print/article/id/22333

Hinet Application of Big Data
中華電信已經在做的海量資料應用

iT ithome.com.tw

中華電信用Hadoop技術分析通話明細

READ LATER

面對資料快速成長以及非結構性資料的增加，中華電信資訊處第四科科長楊秀一表示，中華電信近來利用Hadoop雲端運算技術自行開發了一個專門用來分析非結構化資料的巨量資料（Big Data）運算平臺，嘗試在資料進到資料倉儲系統之前，先進行資料的分析與處理以減少資料倉儲的資料量。

近年來行動語音市場趨於飽和，為了掌握用戶特性進行客製化行銷，一份資料要進行分析，就會被多次複製，因此即使用戶增加趨緩，但中華電信擁有的資料量仍快速暴增。

中華電信用來分析的資料模型最早於10多年前已有雛形，但當初主要用於行動語音分析。一直到2009年，他們完整導入Teradata的電信業邏輯資料模型cLDM 9.0版，整合更多電信服務的用戶資料。楊秀一表示，當初導入該模型的目的主要是為了整合行動語音、固網、數據的資料，進行以人為中心的分析模式。在導入之前，中華電信的資料模型是以設備為中心，因為不同設備的記錄資料儲存在不同的資料庫，無法進行整合性的分析。

參考來源：中華電信用 Hadoop 技術分析通話明細，發表日期： 2011-06-12
http://www.ithome.com.tw/itadm/article.php?c=68023

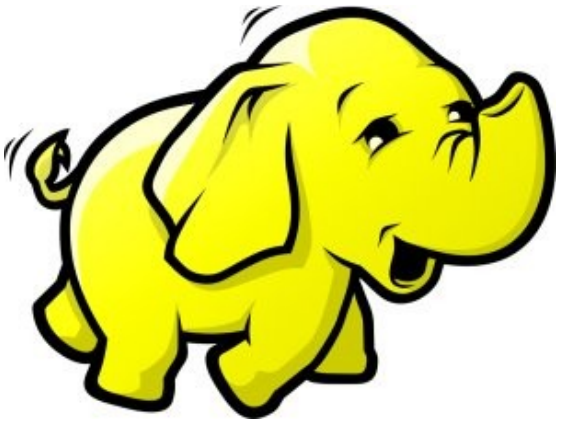
Hadoop 簡介：源起與術語

Introduction to Hadoop : History and Terminology

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



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。

Features of Hadoop …
Hadoop 這套軟體的特色是 …

- **海量 Vast Amounts of Data**
  - 擁有儲存與處理大量資料的能力
  - Capability to STORE and PROCESS vast amounts of data.
- **經濟 Cost Efficiency**
  - 可以用在由一般 PC 所架設的叢集環境內
  - Based on large clusters built of commodity hardware.
- **效率 Parallel Performance**
  - 透過分散式檔案系統的幫助，以致得到快速的回應
  - With the help of HDFS, Hadoop have better performance.
- **可靠 Robustness**
  - 當某節點發生錯誤，能即時自動取得備份資料及佈署運算資源
  - Robustness to add and remove computing and storage resource without shutdown entire system.

Founder of Hadoop – Doug Cutting

Hadoop 這套軟體的創辦人 Doug Cutting

Doug Cutting Talks About The Founding Of Hadoop

http://www.youtube.com/watch?v=qxC4urJOchs

History of Hadoop ... 2002~2004

Hadoop 這套軟體的歷史源起 ... 2002~2004

- Lucene
  - http://lucene.apache.org/
  - 用Java 設計的高效能文件索引引擎API
  - a high-performance, full-featured **text search engine library** written entirely in **Java**.
  - 索引文件中的每一字，讓搜尋的效率比傳統逐字比較還要高的多
  - Lucene create an inverse index of every word in different documents. It enhance performance of text searching.

History of Hadoop ... 2002~2004

Hadoop 這套軟體的歷史源起 ... 2002~2004

- Nutch
  - http://nutch.apache.org/
  - Nutch 是基於開放原始碼所開發的網站搜尋引擎
  - Nutch is open source web-search software.
  - 利用Lucene 函式庫開發
  - It builds on Lucene and Solr, adding web-specifics, such as a crawler, a link-graph database, parsers for HTML and other document formats, etc.

Three Gifts from Google ....

來自 Google 的三個禮物 ....

- Nutch 後來遇到儲存大量網站資料的瓶頸
- Nutch encounter storage issue
- 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 : Simplifed 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

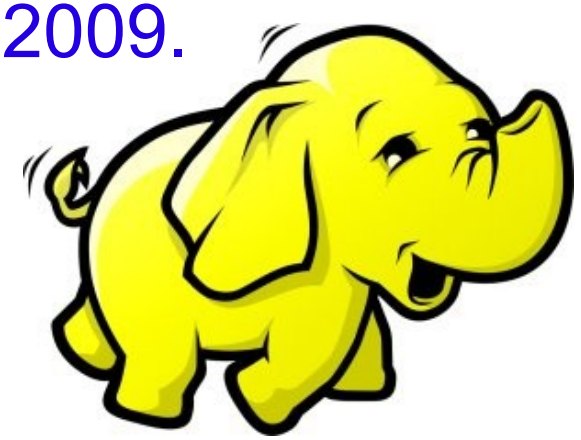
History of Hadoop … 2004 ~ Now

Hadoop 這套軟體的歷史源起 ... 2004 ~ Now

- Dong Cutting reference from Google's publication
- Added DFS & MapReduce implement to Nutch
- According to user feedback on the mail list of Nutch ....
- Hadoop became separated project since Nutch 0.8
- Nutch DFS → Hadoop Distributed File System (HDFS)
- Yahoo hire Dong Cutting to build a team of web search engine at year 2006.
  - Only 14 team members (engineers, clusters, users, etc.)
- Doung Cutting joined Cloudera at year 2009.

YAHOO!

cloudera



Who Use Hadoop ??

有哪些公司在用 Hadoop 這套軟體 ??

- Yahoo is the key contributor currently.
- IBM and Google teach Hadoop in universities …
- http://www.google.com/intl/en/press/pressrel/20071008_ibm_univ.html
- The New York Times used 100 Amazon EC2 instances and a Hadoop application to process 4TB of raw image TIFF data (stored in S3) into 11 million finished PDFs in the space of 24 hours at a computation cost of about $240 (not including bandwidth)

  – from http://en.wikipedia.org/wiki/Hadoop
- http://wiki.apache.org/hadoop/AmazonEC2
- http://wiki.apache.org/hadoop/PoweredBy

- A9.com
- ADSDAQ by Contextweb
- EHarmony
- Facebook
- Fox Interactive Media
- IBM
- ImageShack
- ISI
- Joost
- Last.fm
- Powerset
- The New York Times
- Rackspace
- Veoh
- Metaweb

Hadoop in production run ....
商業運轉中的 Hadoop 應用 ....

- February 19, 2008
- Yahoo! Launches World's Largest Hadoop Production Application
- http://developer.yahoo.net/blogs/hadoop/2008/02/yahoo-worlds-largest-production-hadoop.html

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

Hadoop in production run ....
商業運轉中的 Hadoop 應用 ....

- September 30, 2008
- Scaling Hadoop to 4000 nodes at Yahoo!
- http://developer.yahoo.net/blogs/hadoop/2008/09/scaling_hadoop_to_4000_nodes_a.html

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**

Comparison between Google and Hadoop
Google 與 Hadoop 的比較表

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

Why should we learn Hadoop ?
為何需要學習 Hadoop ??

Search Jobs | Browse Jobs | Local Jobs | Salaries | Employment Trends

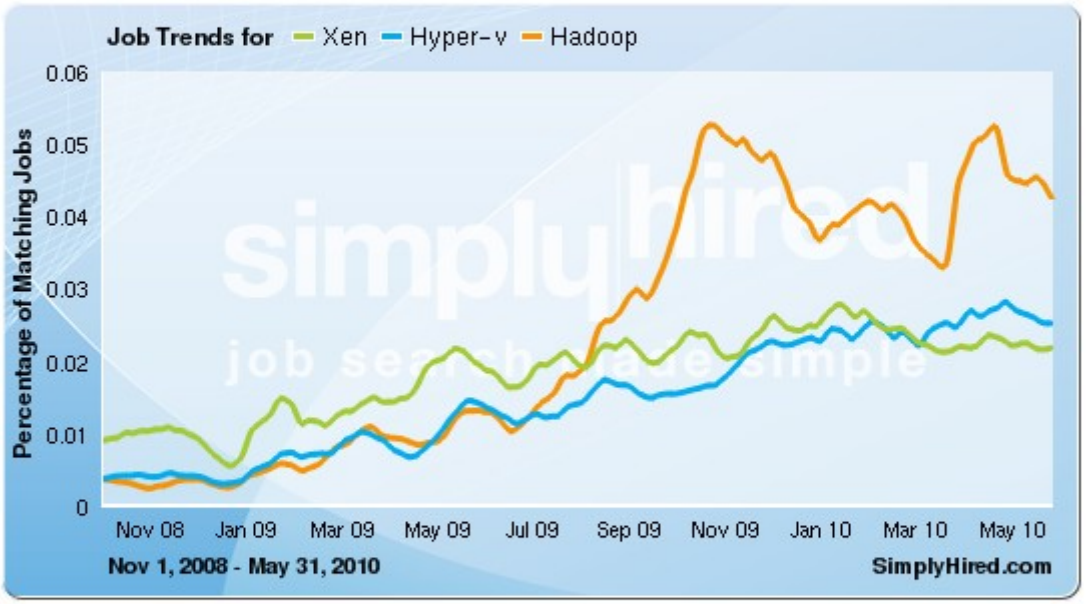
simplyhired®
job search made simple

Employment Trends

Xen, Hyper-V, Hadoop

Tip: You can compare trends by separating them with commas.

Xen, Hyper-v, Hadoop Trends



Xen, Hyper-v, Hadoop Job Trends

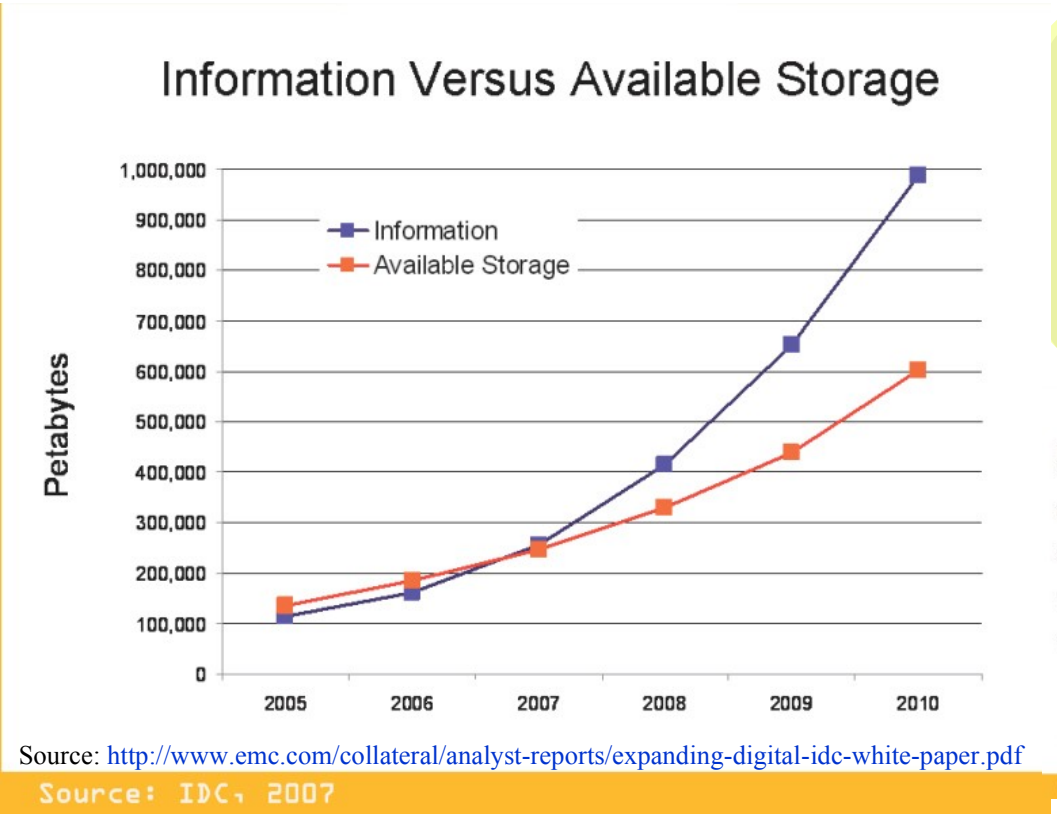
This graph displays the percentage of jobs with your search terms anywhere in the job listing. Since November 2008, the following has occurred:

- Xen jobs increased 141%
- Hyper-v jobs increased 551%
- Hadoop jobs did not change or there is no data available

1. Data Explore
資訊大爆炸

2. Data Mining Tool
方便作資料探勘的工作

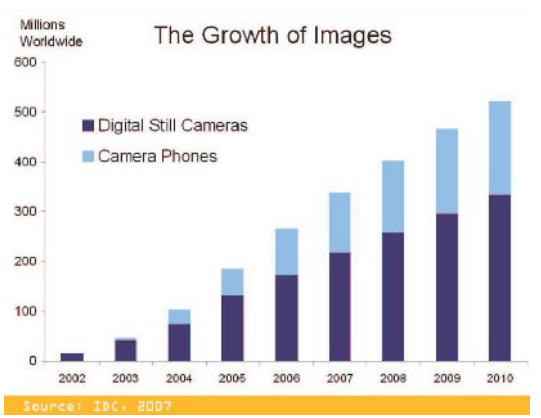
3. Looking for Jobs
好找工作 !!

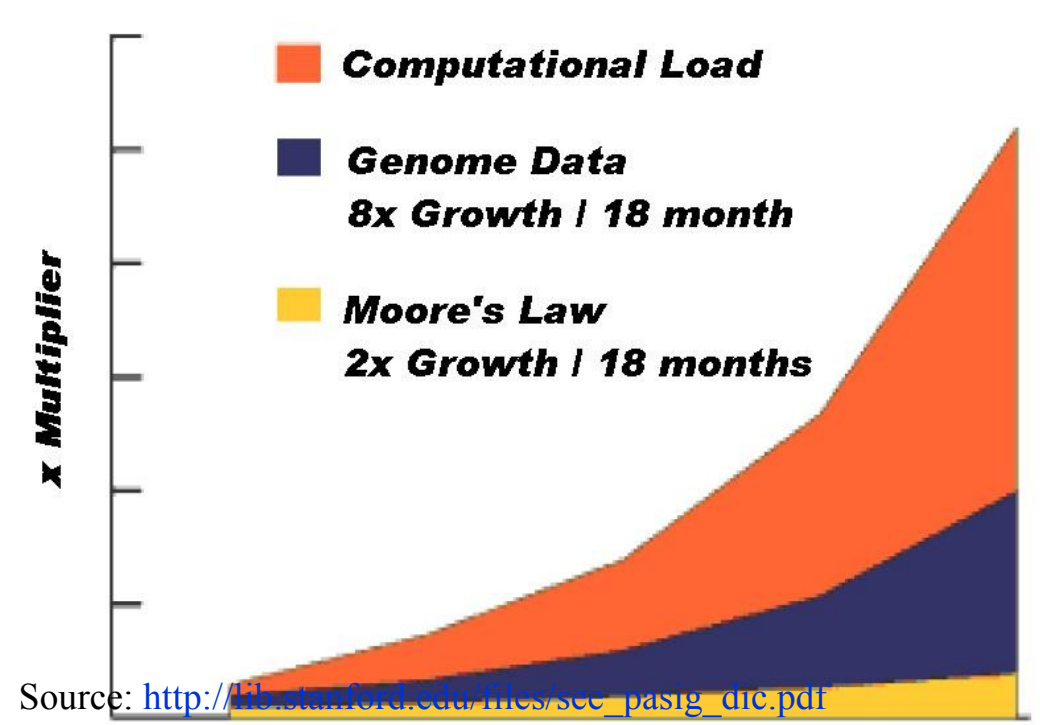


Source: http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf

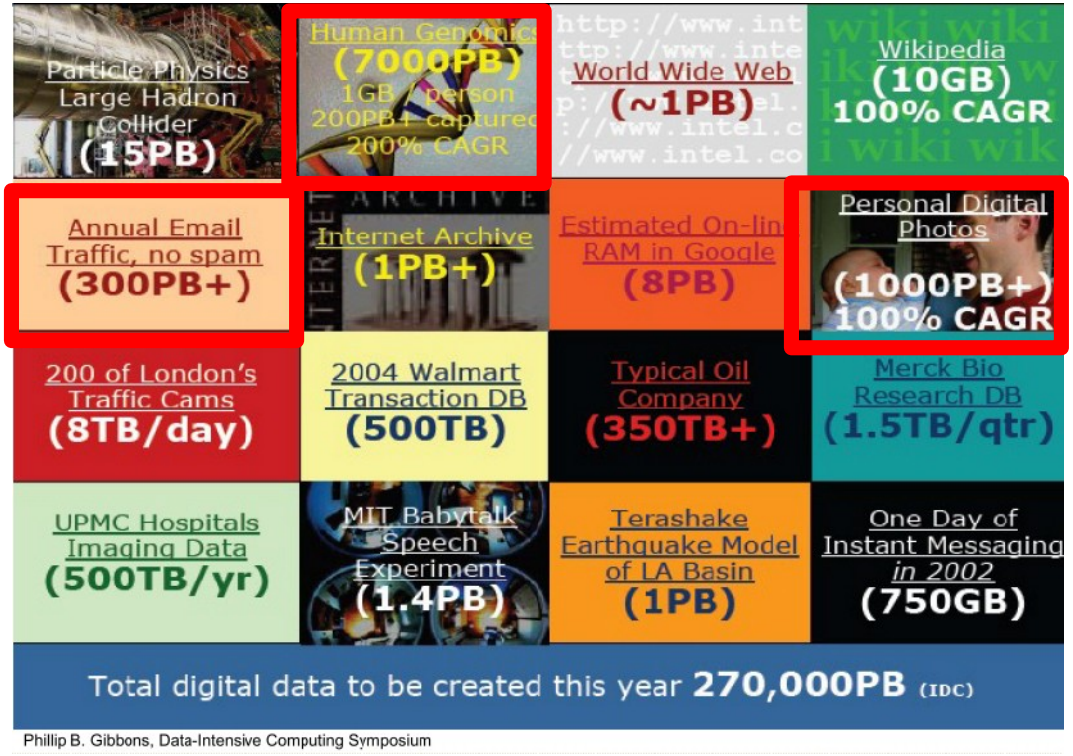
2007 Data Explore

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://dlib.stanford.edu/files/see_pasig_dic.pdf



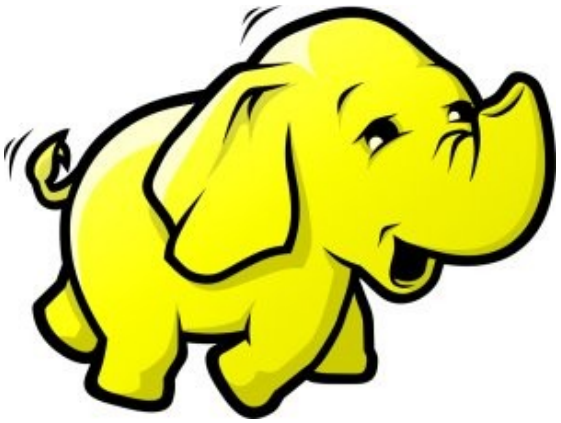
Hadoop 專業術語

Introduction to Hadoop Terminology

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



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

Terminologies of Hadoop

Hadoop 文件中的專業術語

- 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

Two Key Roles of HDFS
HDFS 軟體架構的兩種關鍵角色

名稱節點 NameNode

- Master Node
- Manage NameSpace of HDFS
- Control Permission of Read and Write
- Define the policy of Replication
- Audit and Record the NameSpace
- Single Point of Failure

資料節點 DataNode

- Worker Nodes
- Perform operation of Read and Write
- Execute the request of Replication
- Multiple Nodes

Two Key Roles of Job Scheduler
程序排程的兩種關鍵角色

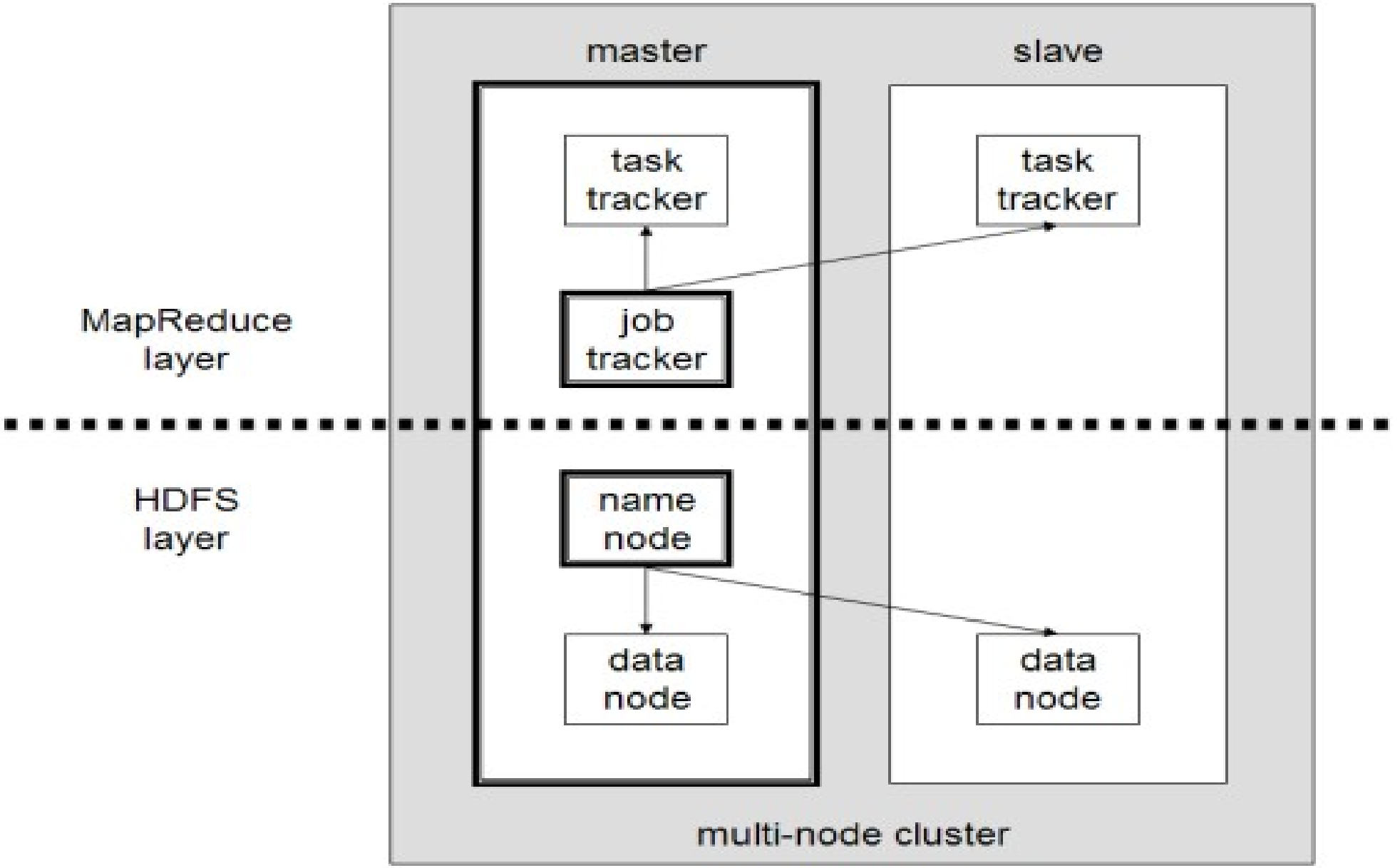
JobTracker

- **Master Node**
- **Receive Jobs from Hadoop Clients**
- **Assigned Tasks to TaskTrackers**
- **Define Job Queuing Policy, Priority and Error Handling**
- **Single Point of Failure**

TaskTracker

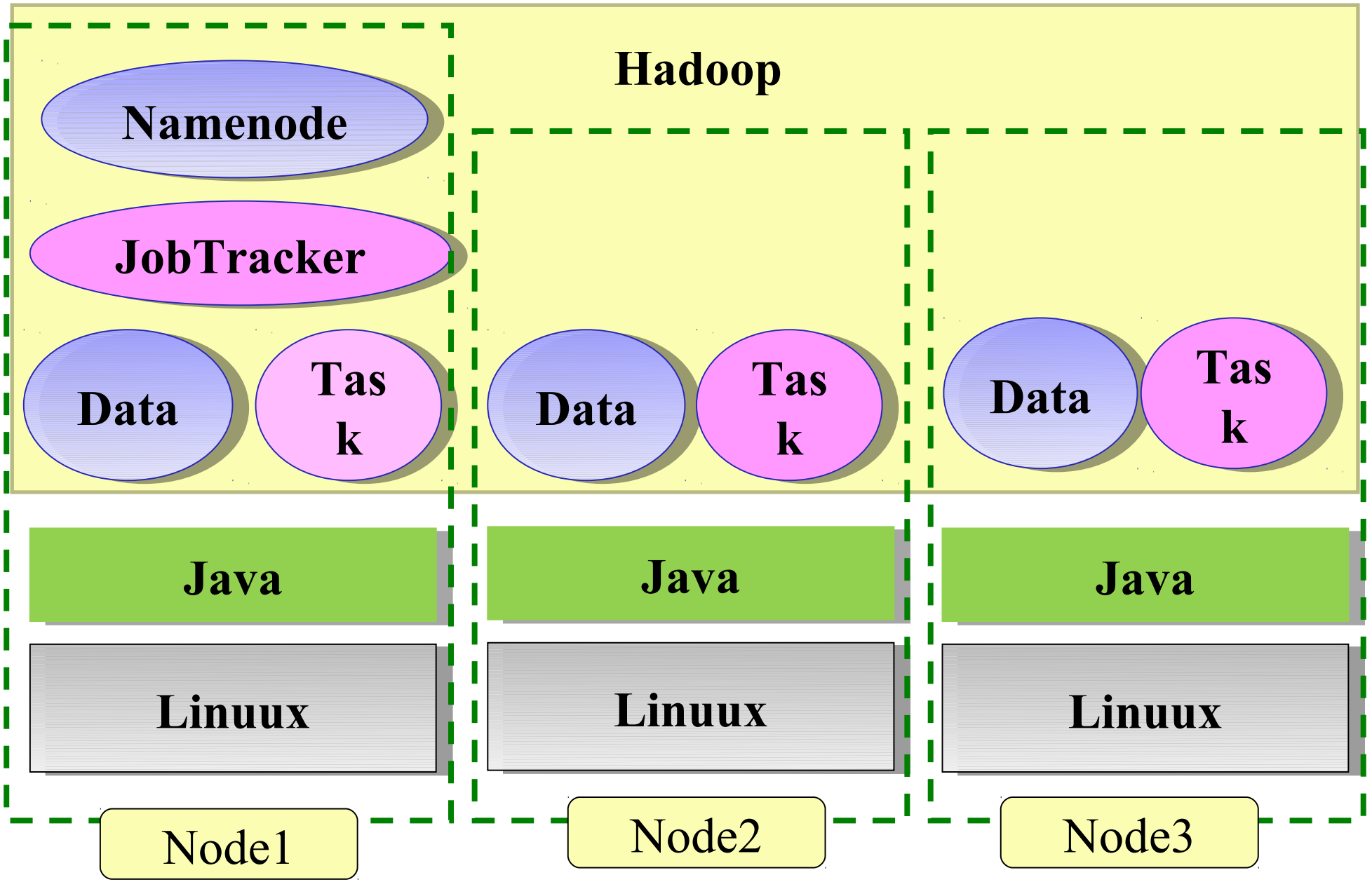
- **Worker Nodes**
- **Excute Mapper and Reducer Tasks**
- **Save Results and report task status**
- **Multiple Nodes**

Different Roles of Hadoop Architecture
Hadoop 軟體架構中的不同角色

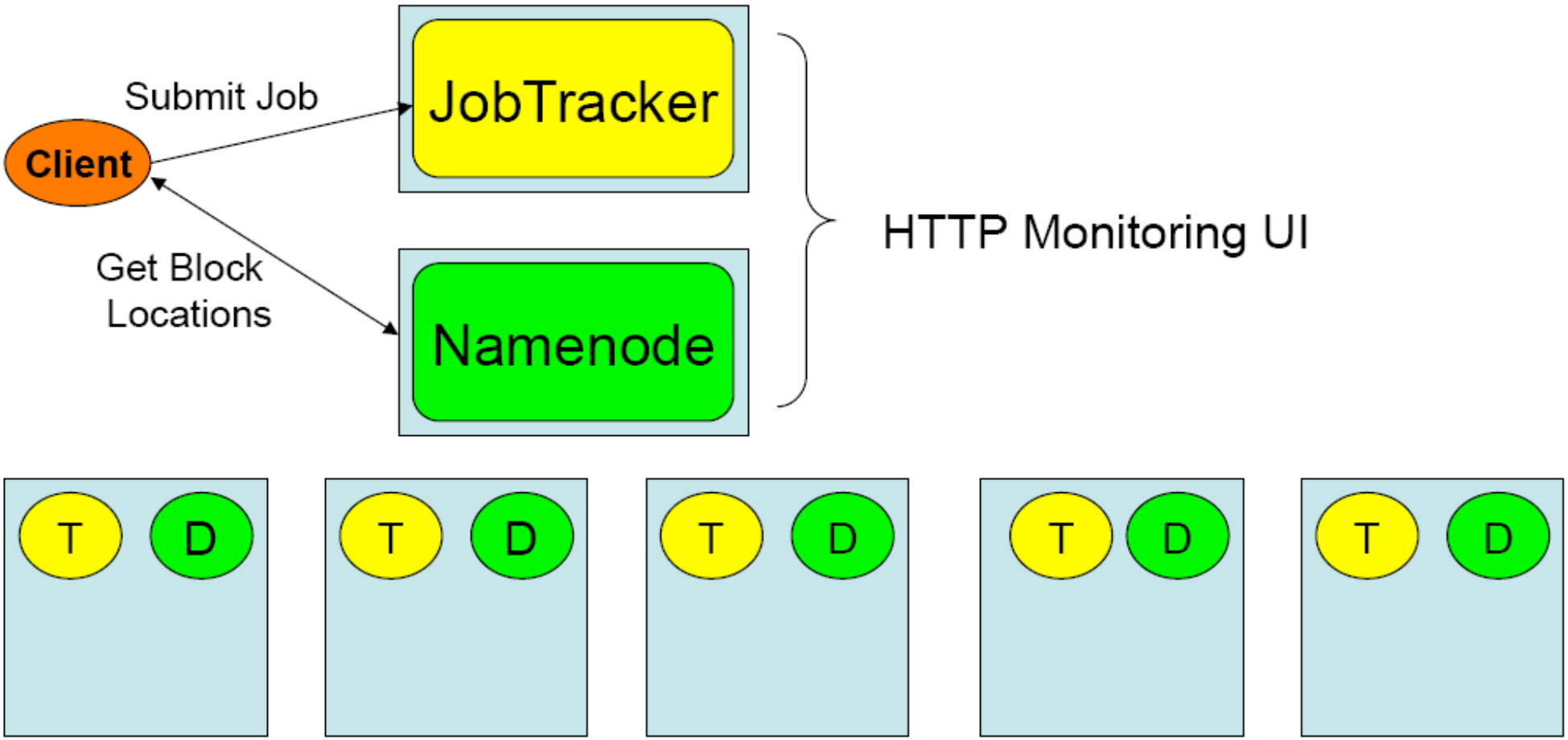


Distributed Operating System of Hadoop

Hadoop 建構成一個分散式作業系統



About Hadoop Client ...
不在雲裡的 Hadoop Client



What we learn today ?

WHAT	Hadoop 是運算海量資料的軟體平台 !! hadoop is a software platform to process vast amount of data!!
WHO	始祖是 Doug Cutting ， Apache 社群支持， Yahoo 贊助 From Doug Cutting to Apache Community, Yahoo and more !
WHEN	Hadoop 是 2004 年從 Nutch 分裂出來的專案 !! Hadoop became separate project since year 2004 !!
WHY	資料大爆炸、資料探勘、找工作 Data Explore, Data Mining, Jobs !!
HOW	建構在大型的個人電腦叢集之上 Install on large clusters built of commodity hardware !!

Questions?

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

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

Powered by DRBL

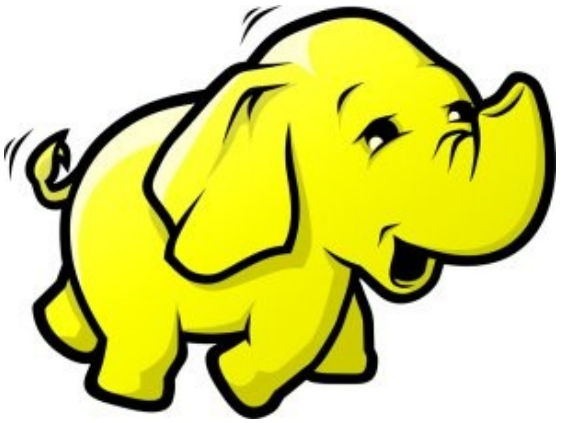
HDFS 簡介

Introduction to Hadoop Distributed File System

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



What is HDFS ??
什麼是 HDFS ??

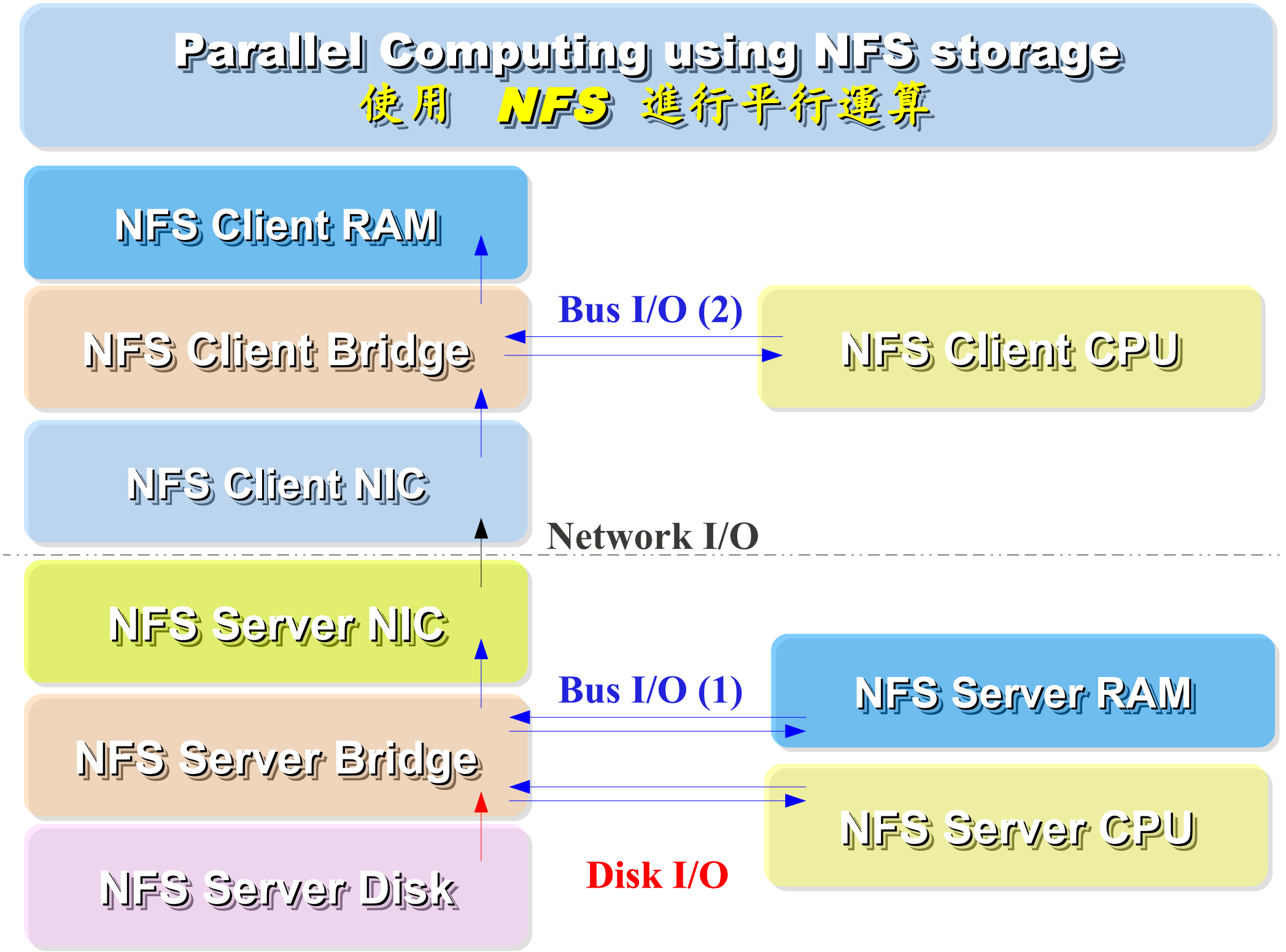
- **Hadoop Distributed File System**
  - 實現類似 Google File System 分散式檔案系統
  - Reference from Google File System.
  - 一個易於擴充的分散式檔案系統，目的爲對大量資料進行分析
  - A scalable distributed file system for large data analysis .
  - 運作於廉價的普通硬體上，又可以提供容錯功能
  - based on commodity hardware with high fault-tolerant.
  - 給大量的用户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 ...

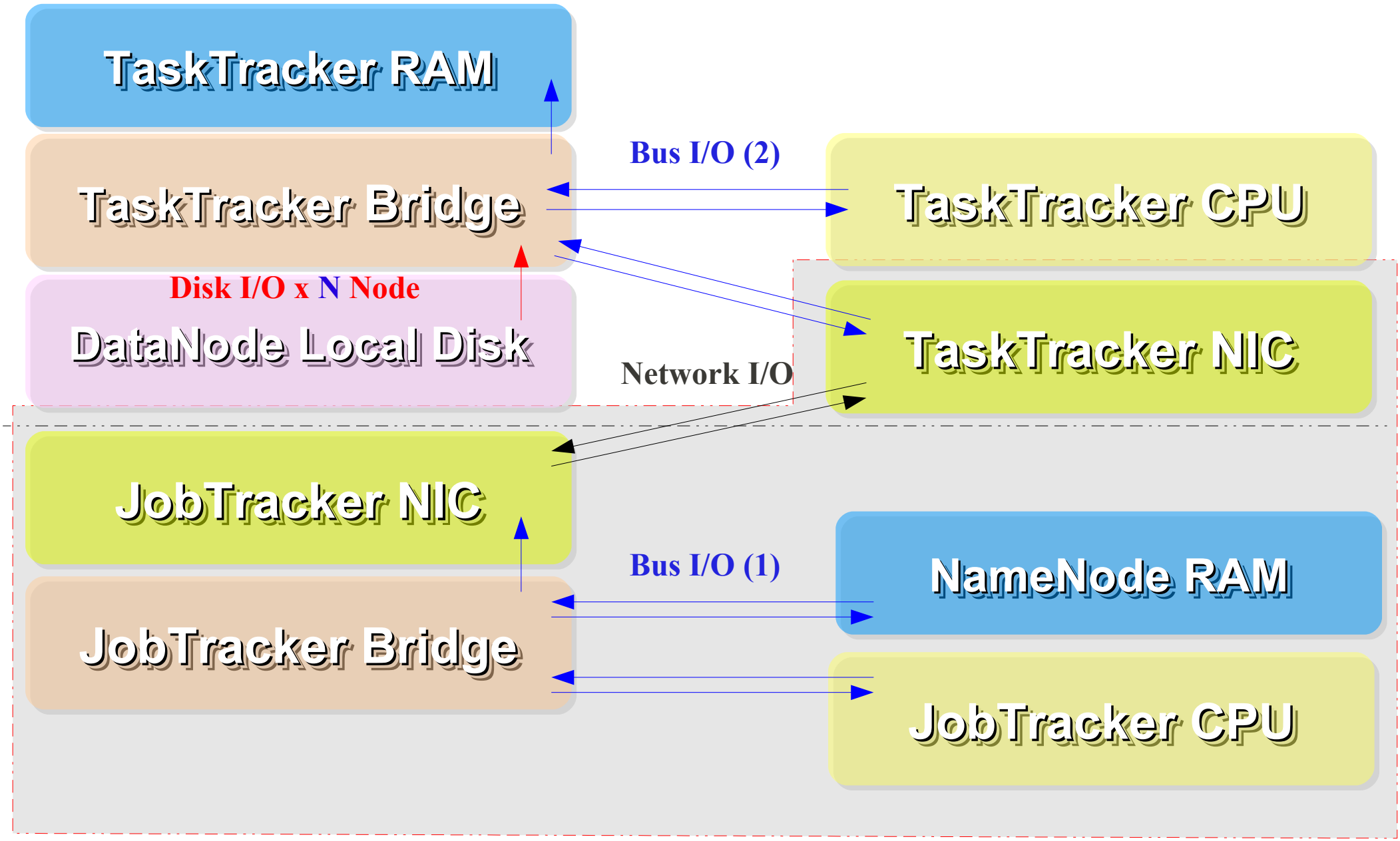
- **硬體錯誤容忍能力 Fault Tolerance**
  - 硬體錯誤是正常而非異常
  - Failure is the norm rather than exception
  - 自動恢復或故障排除
  - automatic recovery or report failure
- **串流式的資料存取 Streaming data access**
  - 批次處理多於用户交互處理
  - Batch processing rather than interactive user access.
  - 高 Throughput 而非低 Latency
  - High aggregate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

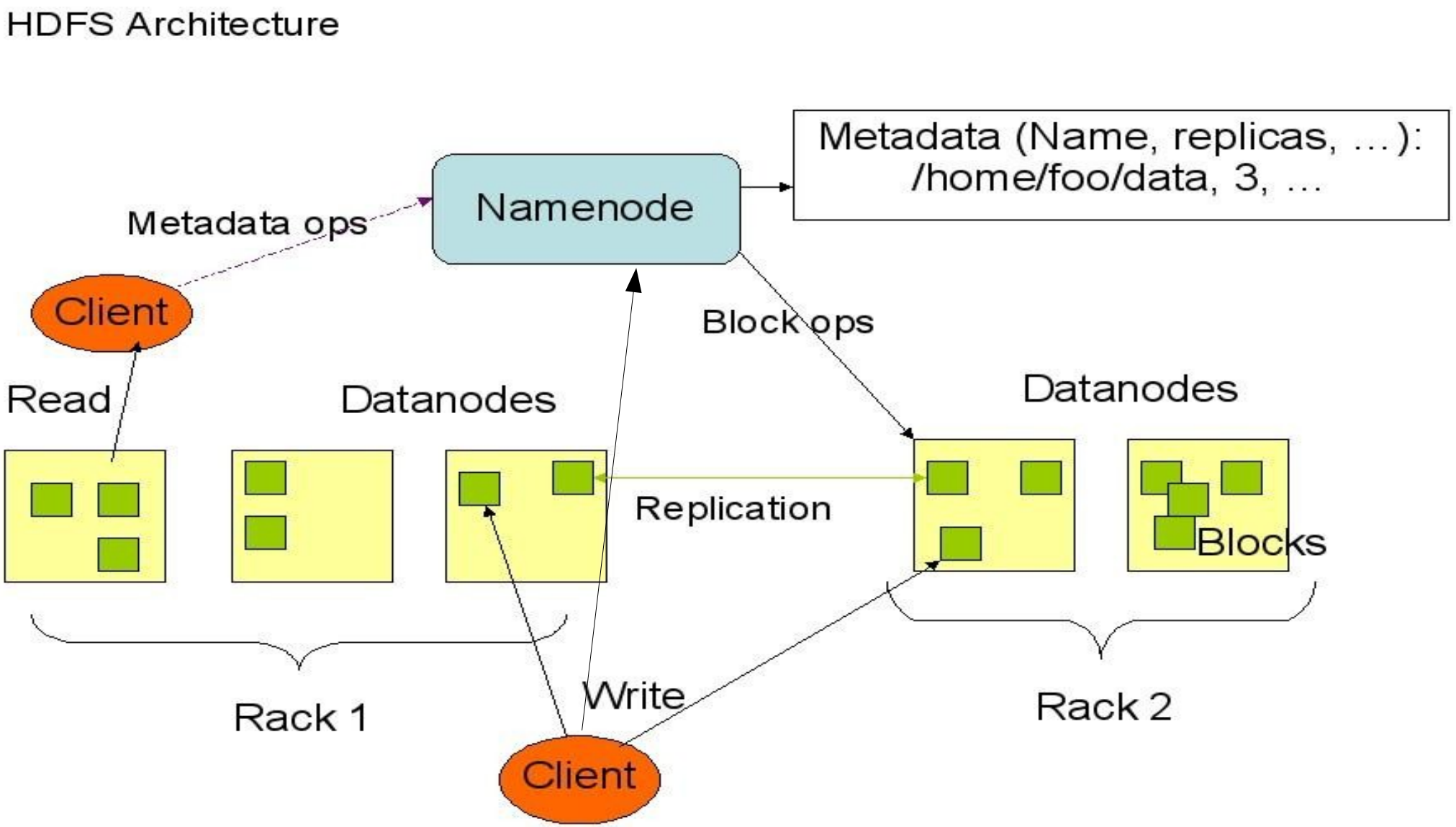

Parallel Computing using NFS storage
使用 NFS 進行平行運算
NFS Client RAM
NFS Client Bridge
NFS Client NIC
Bus I/O (2)
NFS Client CPU
Network I/O
NFS Server NIC
Bus I/O (1)
NFS Server RAM
NFS Server Bridge
NFS Server CPU
NFS Server Disk
Disk I/O

Parallel Computing using HDFS
使用 HDFS 進行平行運算



How HDFS manage data ...

HDFS 如何管理資料 ...



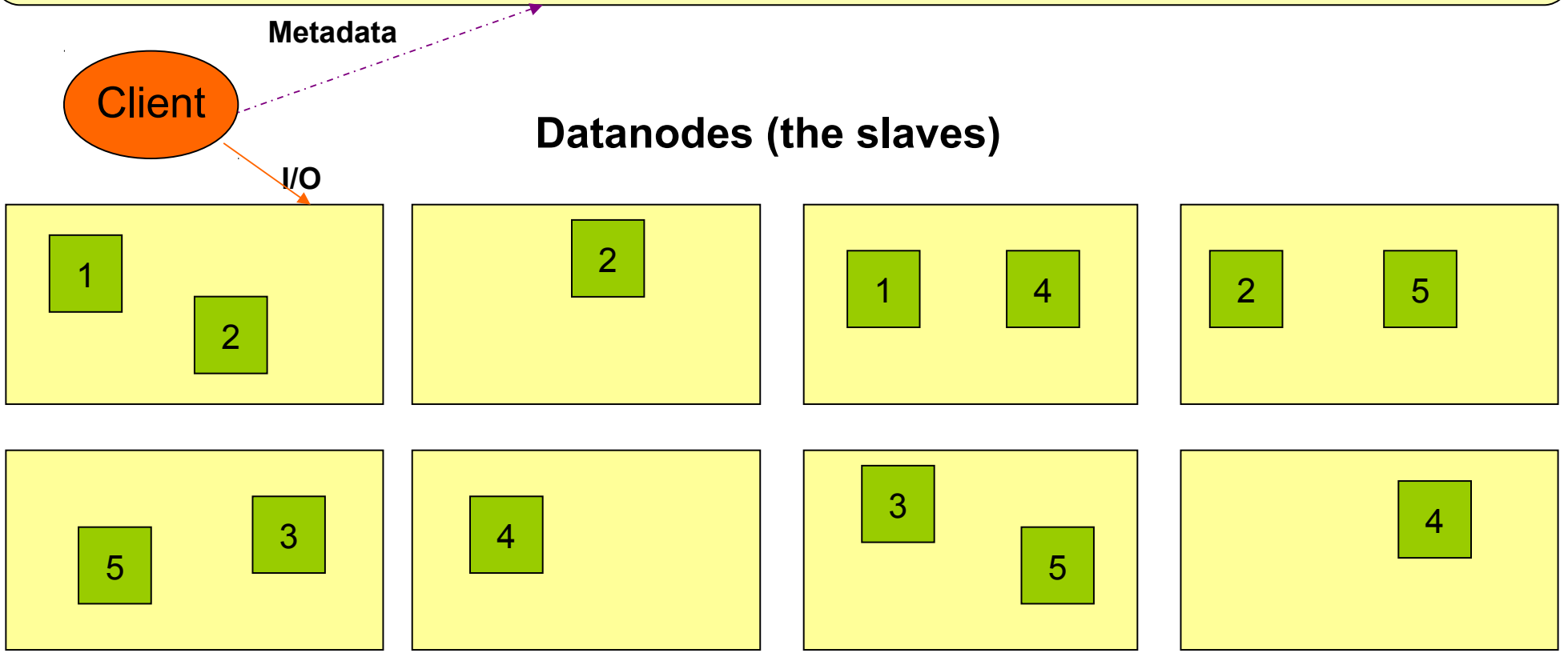
How does HDFS work ...

HDFS 如何運作 ...

Namenode (the master)

Path and Filename – Replication , blocks

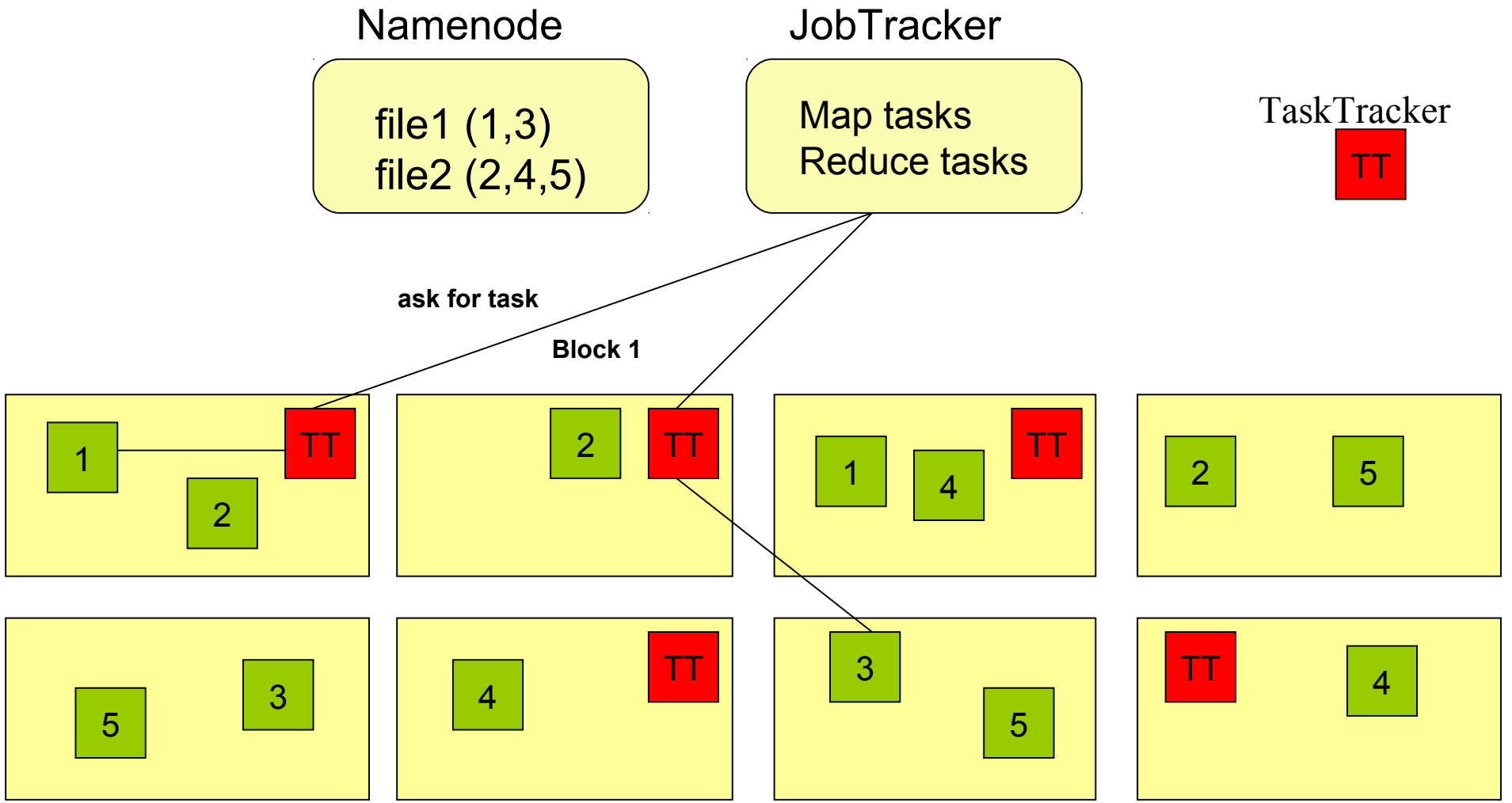
name:/users/joeYahoo/myFile - copies:2, blocks:{1,3}
name:/users/bobYahoo/someData.gzip, copies:3, blocks:{2,4,5}



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 bandwith： distribute read （ but increase write bottlenet ）



9

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
  - Replcae corrupt block with replication one
- Heartbeat
  - Datanode send heartbeat to Namenode
- Metadata
  - FSImage、Editlog 爲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

10

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 爲單位
  - 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

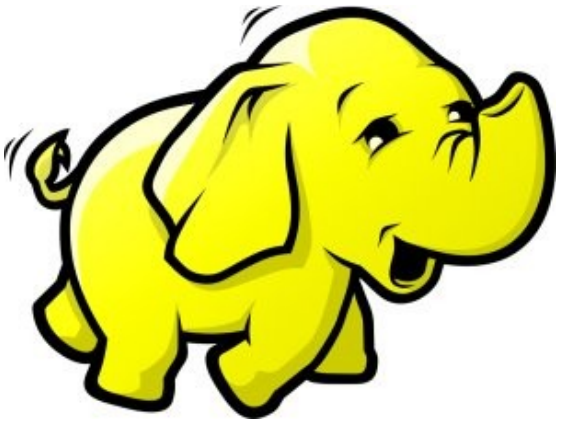
MapReduce 簡介

Introduction to MapReduce

Jazz Wang

Yao-Tsung Wang

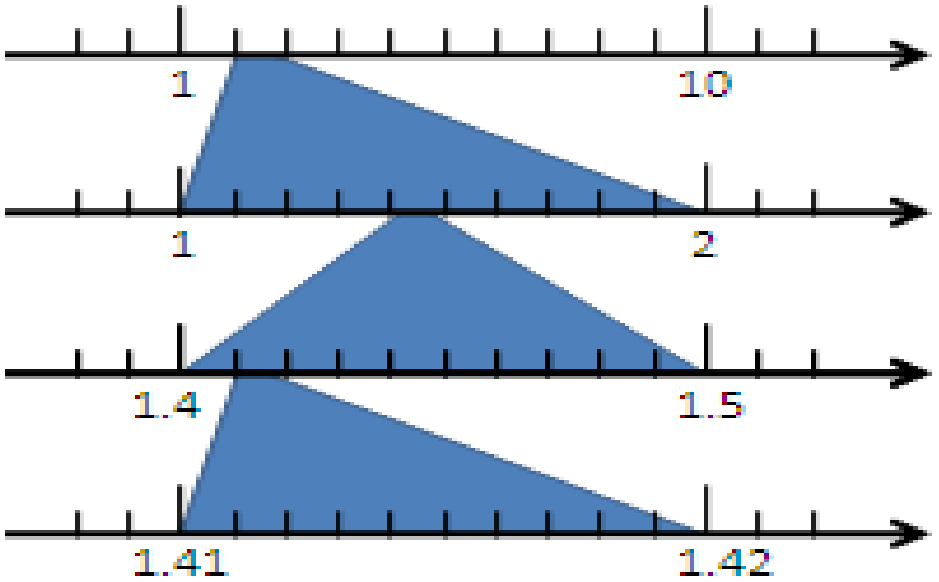
jazz@nchc.org.tw



Divide and Conquer Algorithms

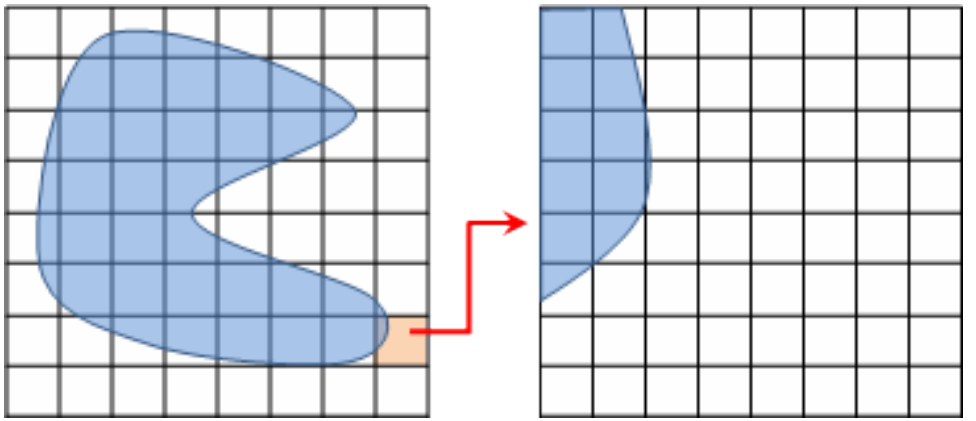
分而治之演算法

Example 1:

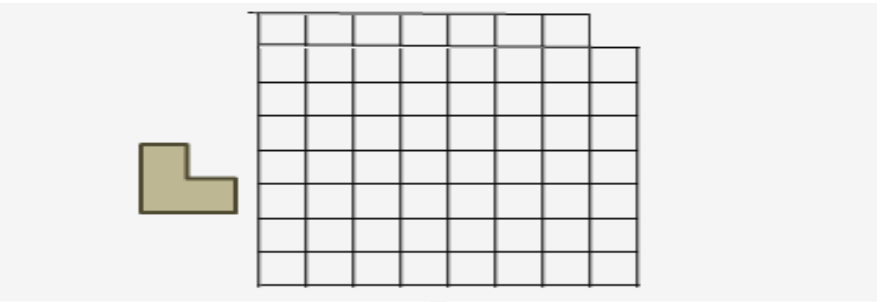


sqrt(2)

Example 2:



Example 3:



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)

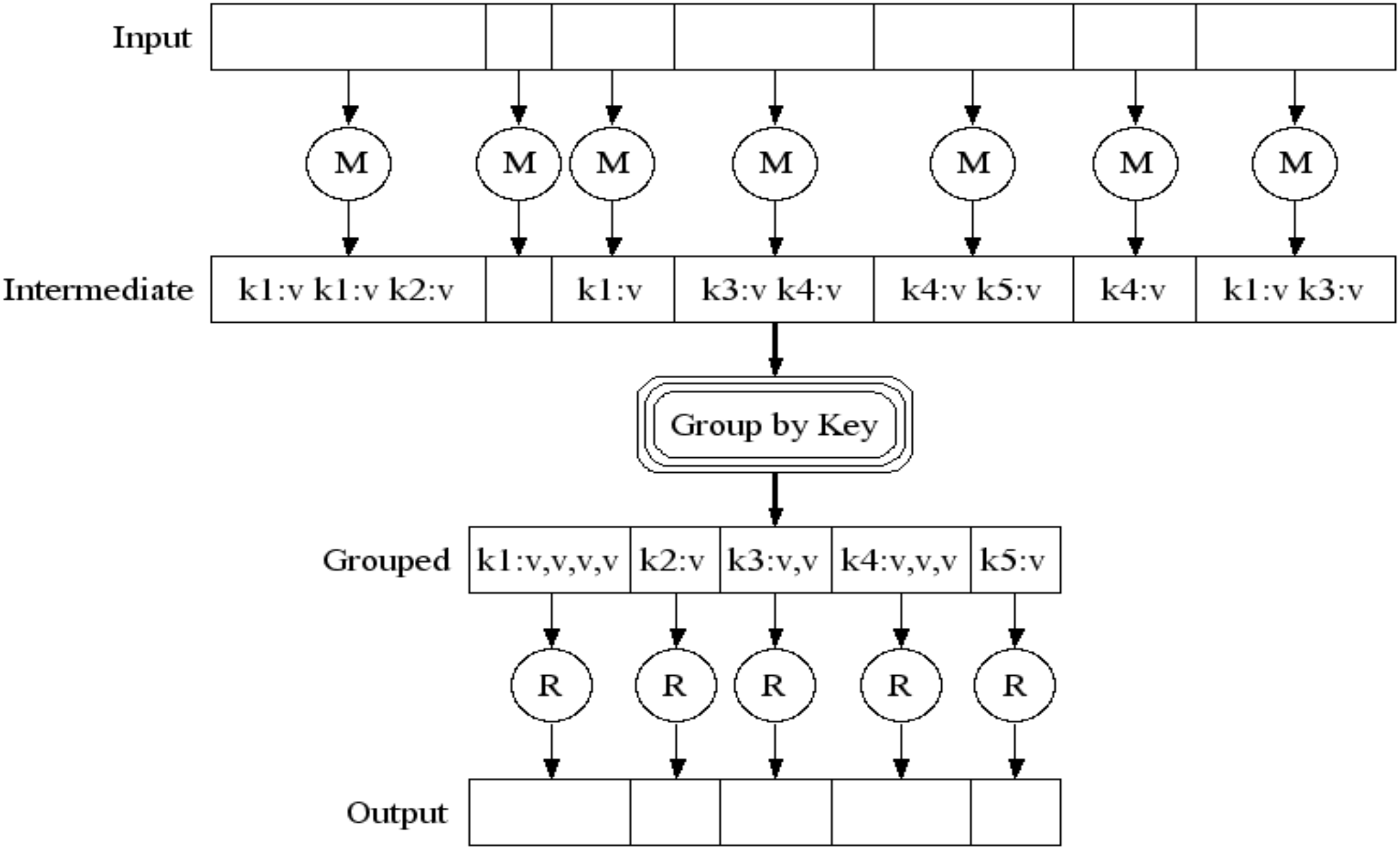
What is MapReduce ??
什麼是 MapReduce ??

- 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.**
- 啓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 → N
    - Ex. [1,2,3,4] – (*2) -> [2,4,6,8]
  - Reduce(...): N → 1
    - [1,2,3,4] - (sum) -> 10
- **Logical view of MapReduce**
    - Map(k1, v1) -> list(k2, v2)
    - Reduce(k2, list (v2)) -> list(k3, v3)

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

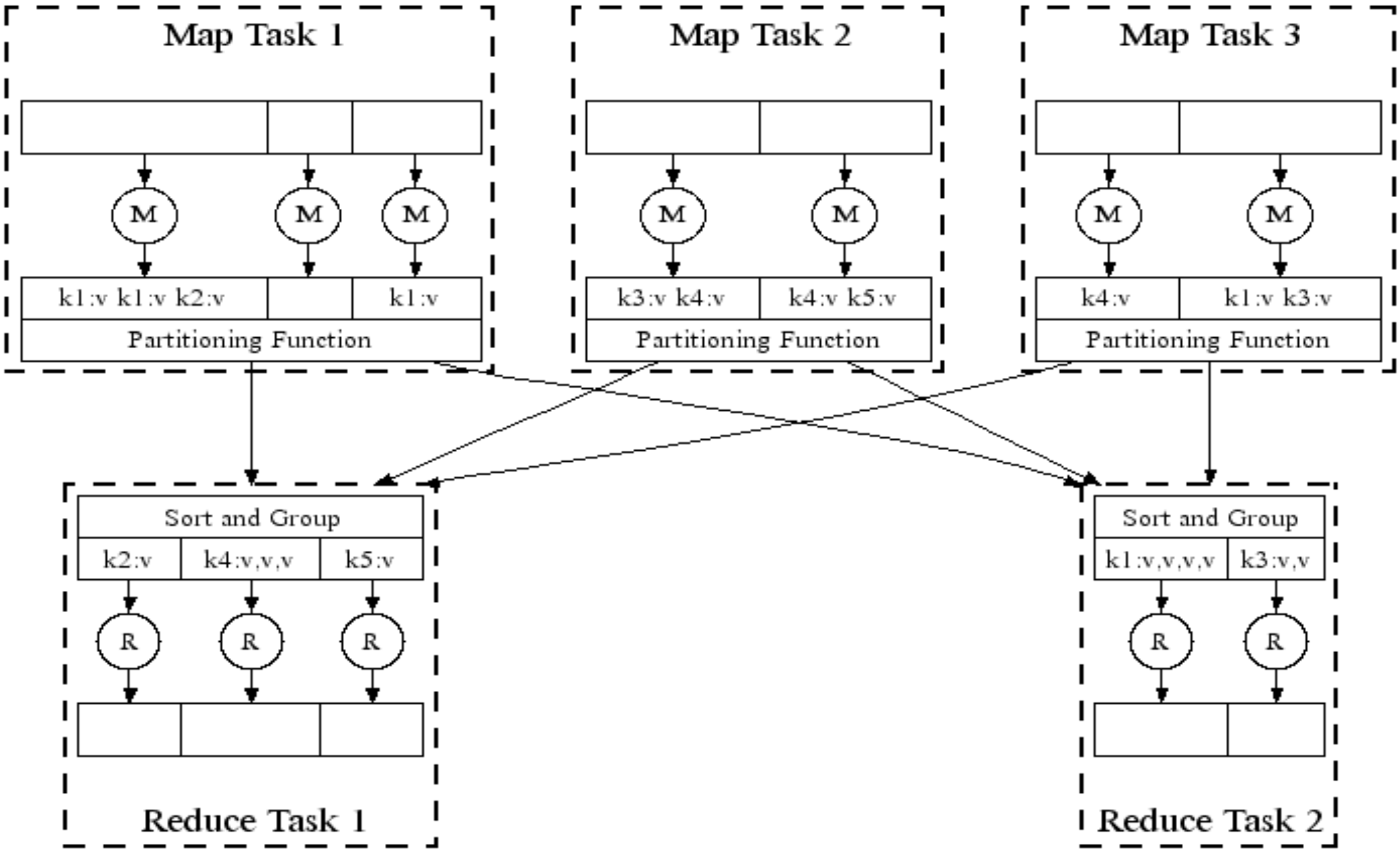
Google's MapReduce Diagram

Google 的 MapReduce 圖解



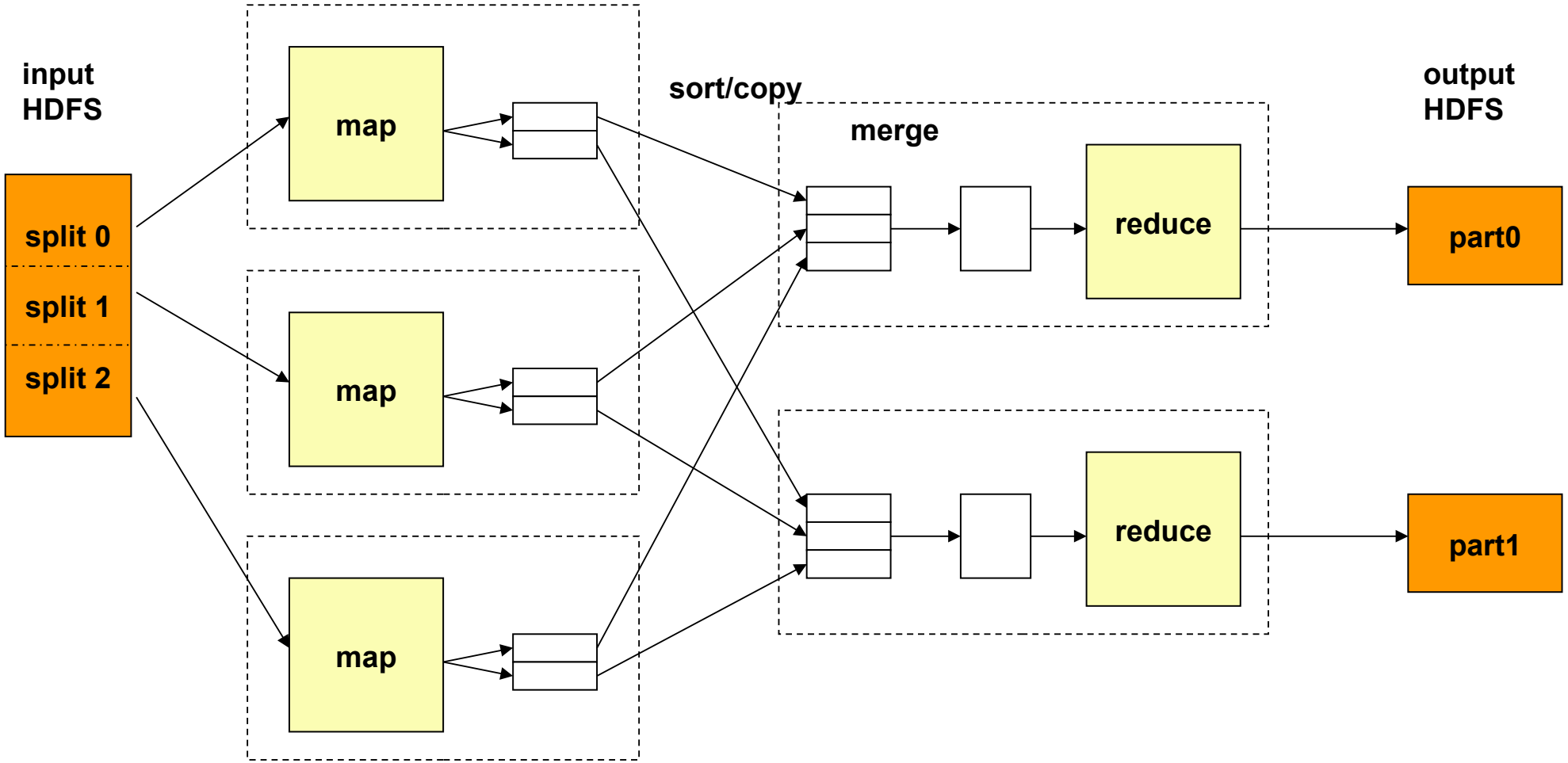
Google's MapReduce in Parallel

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

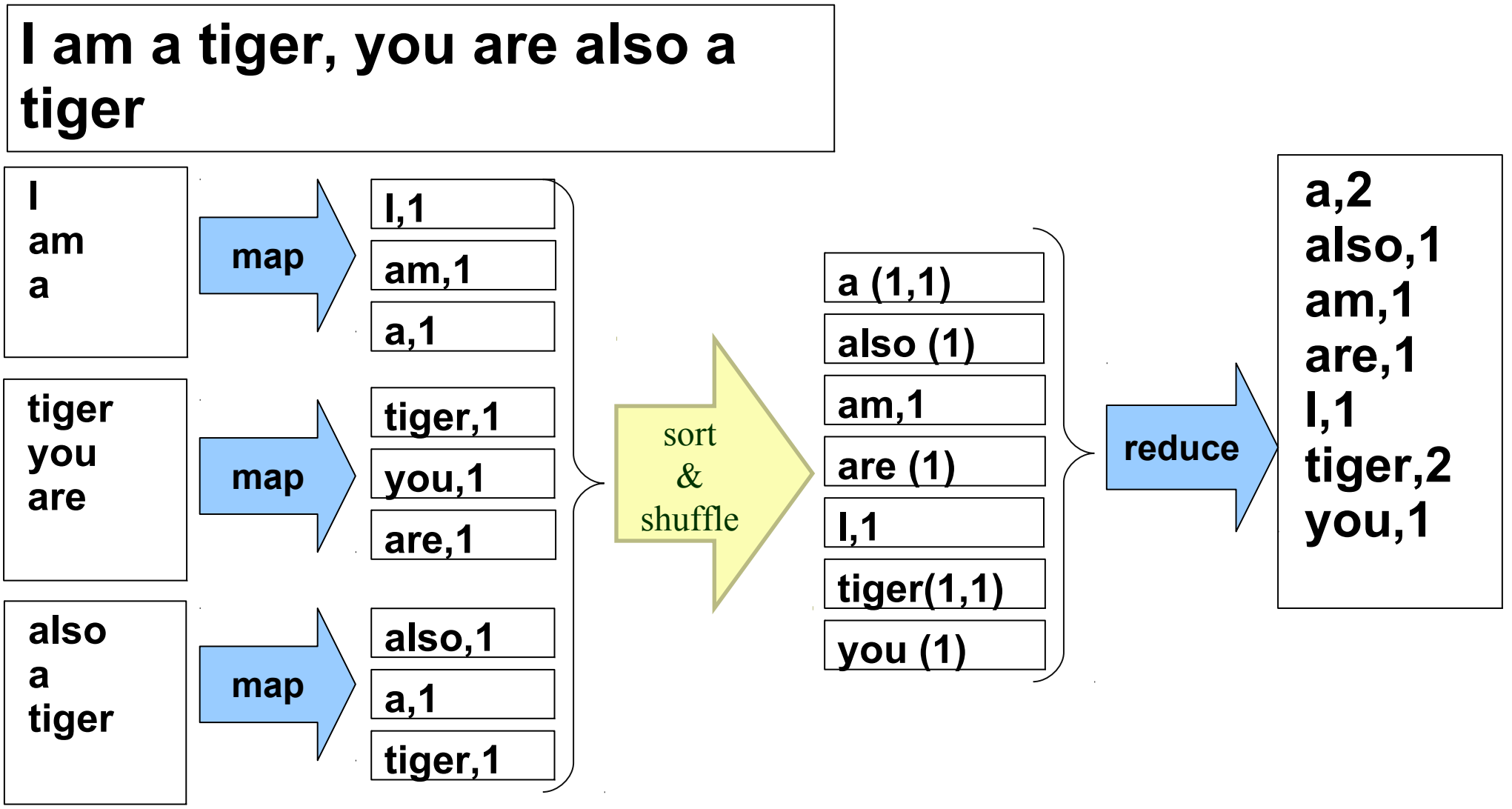
JobTracker 將中間檔案整合排序後，複製到需要的 TaskTracker 去

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)



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)

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

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

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

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
```

```
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

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

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

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

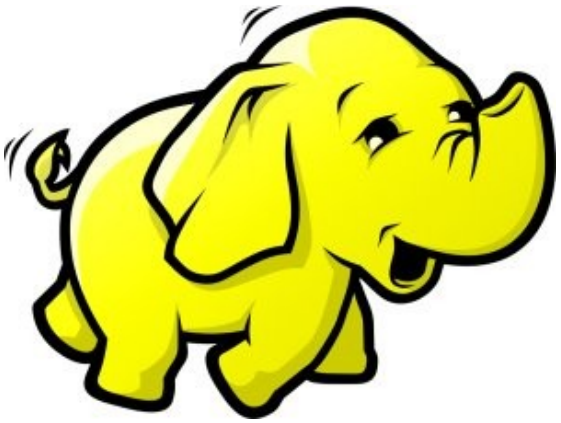
MapReduce 程式設計入門

MapReduce Programing 101

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw



Program Prototype (v 0.20)

Map 區

Reduce 區

設定區

```
Class MR{
    static public Class Mapper …{
    }
    static public Class Reducer …{
    }
    main(){
        Configuration conf = new Configuration();
        Job job = new Job(conf, “job name");
        job.setJarByClass(thisMainClass.class);
        job.setMapperClass(Mapper.class);
        job.setReduceClass(Reducer.class);
        FileInputFormat.addInputPaths(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        job.waitForCompletion(true);
}}
```

Mapper 程式碼

Reducer 程式碼

Other configure 其他的設定參數程式碼

Program Prototype (v 0.18)

Map 區

Reduce 區

設定區

```
Class MR{
    static public Class Mapper ...{
    }
    static public Class Reducer ...{
    }
    main(){
        JobConf conf = new JobConf( MR.class );
        conf.setMapperClass(Mapper.class);
        conf.setReduceClass(Reducer.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        JobClient.runJob(conf);
}}
```

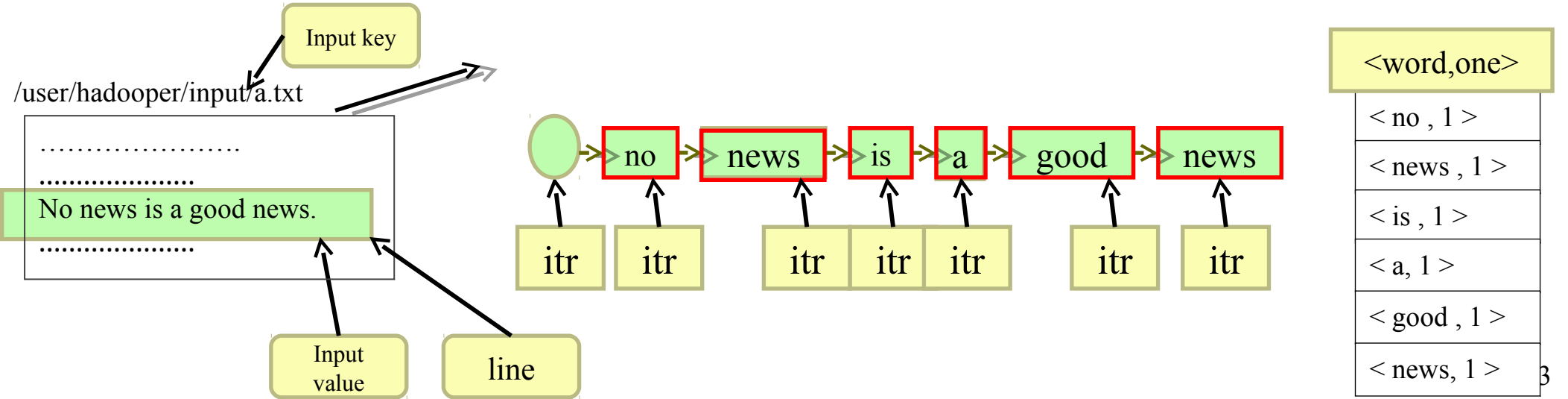
Map 程式碼

Reduce 程式碼

Other configure 其他的設定參數程式碼

Word Count - mapper

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



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)
           throws IOException, InterruptedException {
4                  int sum = 0;
5                  for ( IntWritable val : values )
6                  sum += val.get();
7                  result.set(sum);
8                  context.write ( key, result);
   }}
```

```
for ( int i ; i < values.length ; i ++ ){
  sum += values[i].get()
}
```

<word,one>

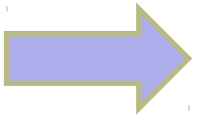
< a, 1 >

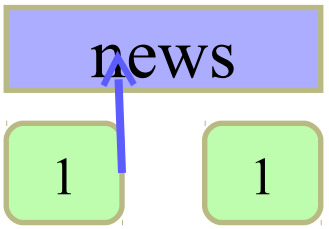
< good, 1 >

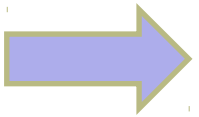
< is, 1 >

< news, 1→1 >

< no, 1 >







<key,SunValue>

< news , 2 >

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.setReduceClass(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

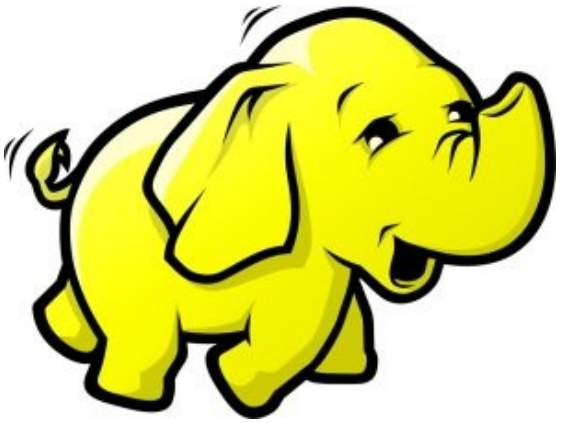
Hadoop 相關計畫

Hadoop Ecosystem

Jazz Wang

Yao-Tsung Wang

jazz@nchc.org.tw

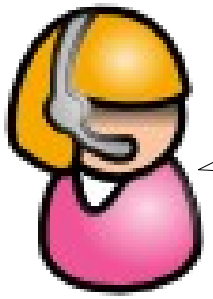


Hadoop 只支援用 **Java** 開發嘛？

Is Hadoop only support Java ?

總不能全部都重新設計吧？如何與舊系統相容？

Can Hadoop work with existing software ?



可以跟資料庫結合嘛？

Can Hadoop work with Databases ?

開發者們有聽到大家的需求 . . . . .

Yes, we hear the feedback of developers ...

Is Hadoop only support Java ?

- Although the Hadoop framework is implemented in Java™, Map/Reduce applications need not be written in Java.
- Hadoop Streaming is a utility which allows users to create and run jobs with any executables (e.g. shell utilities) as the mapper and/or the reducer.
- Hadoop Pipes is a SWIG-compatible C++ API to implement Map/Reduce applications (non JNI™ based).

Hadoop Pipes (C++, Python)

- Hadoop Pipes allows C++ code to use Hadoop DFS and map/reduce.
- The C++ interface is "swigable" so that interfaces can be generated for python and other scripting languages.
- For more detail, check the API Document of org.apache.hadoop.mapred.pipes
- You can also find example code at

  hadoop-*/src/examples/pipes
- About the pipes C++ WordCount example code:

  http://wiki.apache.org/hadoop/C++WordCount

Hadoop Streaming

- Hadoop Streaming is a utility which allows users to create and run Map-Reduce jobs with any executables (e.g. Unix shell utilities) as the mapper and/or the reducer.
- It's useful when you need to run existing program written in shell script, perl script or even PHP.
- Note: both the mapper and the reducer are executables that read the input from STDIN (line by line) and emit the output to STDOUT.
- For more detail, check the official document of Hadoop Streaming

Running Hadoop Streaming

```
jazz@hadoop:~$ hadoop jar hadoop-streaming.jar -help
10/08/11 00:20:00 ERROR streaming.StreamJob: Missing required option -input
Usage: $HADOOP_HOME/bin/hadoop [--config dir] jar \
          $HADOOP_HOME/hadoop-streaming.jar [options]
Options:
  -input    <path>     DFS input file(s) for the Map step
  -output   <path>     DFS output directory for the Reduce step
  -mapper   <cmd|JavaClassName>      The streaming command to run
  -combiner <JavaClassName> Combiner has to be a Java class
  -reducer  <cmd|JavaClassName>      The streaming command to run
  -file     <file>     File/dir to be shipped in the Job jar file
  -dfs    <h:p>|local  Optional. Override DFS configuration
  -jt     <h:p>|local  Optional. Override JobTracker configuration
  -additionalconfspec specfile  Optional.
  -inputformat TextInputFormat(default)|SequenceFileAsTextInputFormat|
JavaClassName Optional.
  -outputformat TextOutputFormat(default)|JavaClassName  Optional.
```

… More …

Hadoop Streaming with shell commands (1)

```
hadoop:~$ hadoop fs -rmr input output
hadoop:~$ hadoop fs -put /etc/hadoop/conf input
hadoop:~$ hadoop jar hadoop-streaming.jar -input
input -output output -mapper /bin/cat
-reducer /usr/bin/wc
```

Hadoop Streaming with shell commands (2)

```
hadoop:~$ echo "sed -e \"s/ /\n/g\" | grep ." > streamingMapper.sh
hadoop:~$ echo "uniq -c | awk '{print \$2 \"\t\" \$1}'" > streamingReducer.sh
hadoop:~$ chmod a+x streamingMapper.sh
hadoop:~$ chmod a+x streamingReducer.sh
hadoop:~$ hadoop fs -put /etc/hadoop/conf input
hadoop:~$ hadoop jar hadoop-streaming.jar -input input -output output -mapper streamingMapper.sh -reducer streamingReducer.sh -file streamingMapper.sh -file streamingReducer.sh
```

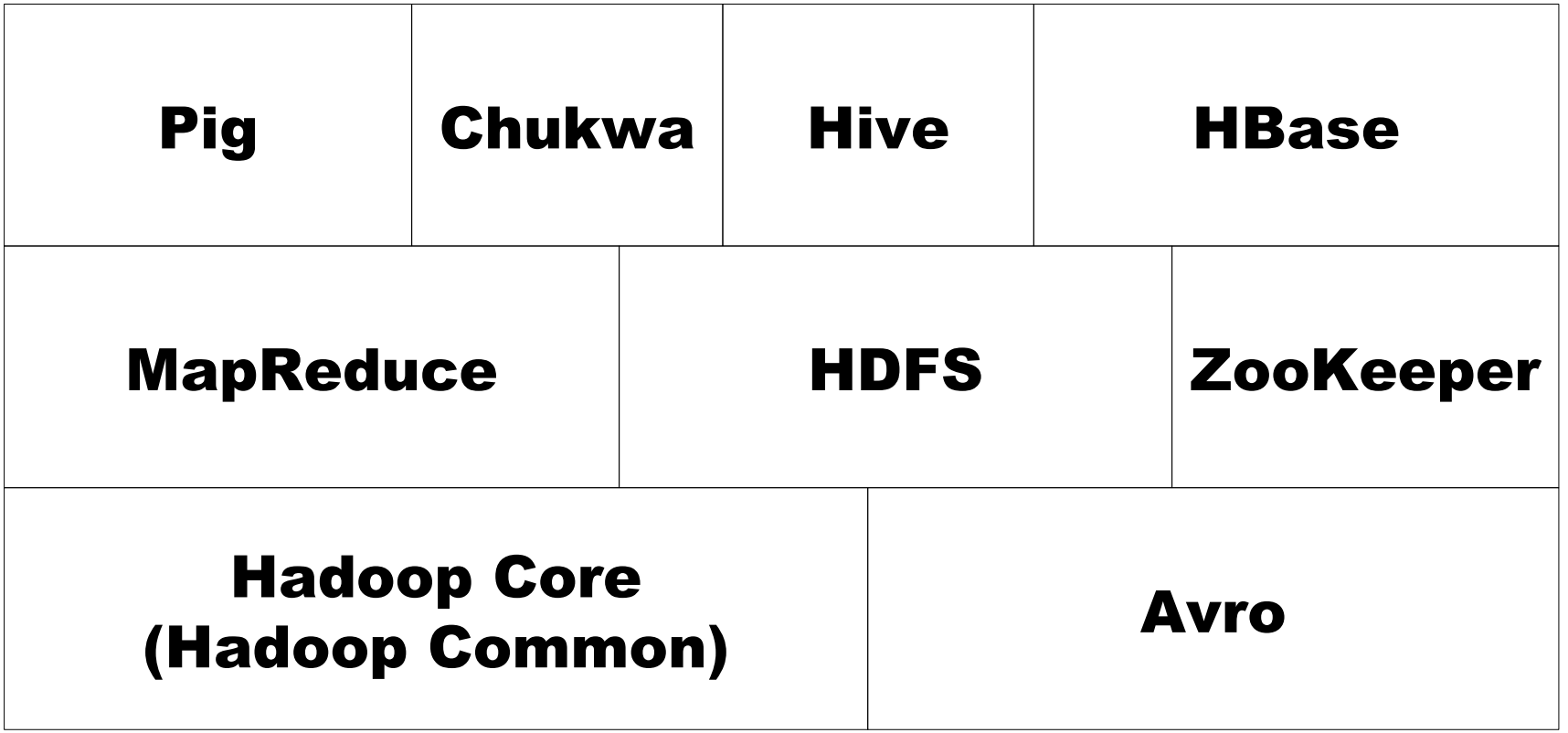
There are serveral Hadoop subprojects

- Hadoop Common: The common utilities that support the other Hadoop subprojects.
- HDFS: A distributed file system that provides high throughput access to application data.
- MapReduce: A software framework for distributed processing of large data sets on compute clusters.

Other Hadoop related projects

- Chukwa: A data collection system for managing large distributed systems.
- HBase: A scalable, distributed database that supports structured data storage for large tables.
- Hive: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- Pig: A high-level data-flow language and execution framework for parallel computation.
- ZooKeeper: A high-performance coordination service for distributed applications.

Hadoop Ecosystem



Source: *Hadoop: The Definitive Guide*

Avro

- Avro is a data serialization system.
- It provides:
  - *Rich data structures.*
  - *A compact, fast, binary data format.*
  - *A container file, to store persistent data.*
  - *Remote procedure call (RPC).*
  - *Simple integration with dynamic languages.*
- Code generation is not required to read or write data files nor to use or implement RPC protocols. Code generation as an optional optimization, only worth implementing for statically typed languages.
- For more detail, please check the official document: http://avro.apache.org/docs/current/

Zoo Keeper

- http://hadoop.apache.org/zookeeper/
- ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications.
- *Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them ,which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.*

Pig

- http://hadoop.apache.org/pig/
- Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs.
- Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs
- Pig's language layer currently consists of a textual language called Pig Latin, which has the following key properties:
  - Ease of programming
  - Optimization opportunities
  - Extensibility

Hive

- http://hadoop.apache.org/hive/
- Hive is a data warehouse infrastructure built on top of Hadoop that provides tools to enable easy data summarization, adhoc querying and analysis of large datasets data stored in Hadoop files.
- Hive QL is based on SQL and enables users familiar with SQL to query this data.

Chukwa

- http://hadoop.apache.org/chukwa/
- Chukwa is an open source data collection system for monitoring large distributed systems.
- built on top of HDFS and Map/Reduce framework
- includes a flexible and powerful toolkit for displaying, monitoring and analyzing results to make the best use of the collected data.

Mahout

- http://mahout.apache.org/
- Mahout is a scalable machine learning libraries.
- implemented on top of Apache Hadoop using the map/reduce paradigm.
- Mahout currently has
  - Collaborative Filtering
  - User and Item based recommenders
  - K-Means, Fuzzy K-Means clustering
  - Mean Shift clustering
  - More ...

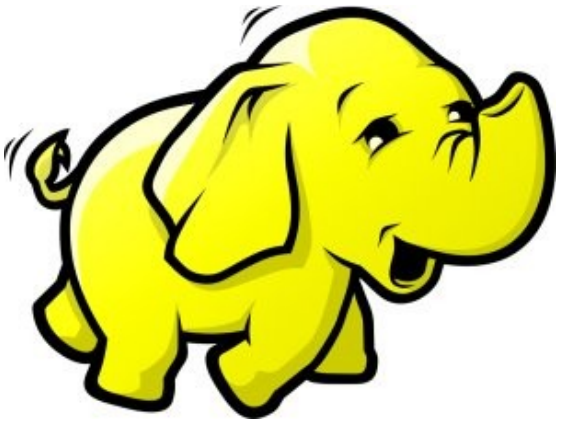
HBase 雲端資料庫

Introduction to HBase

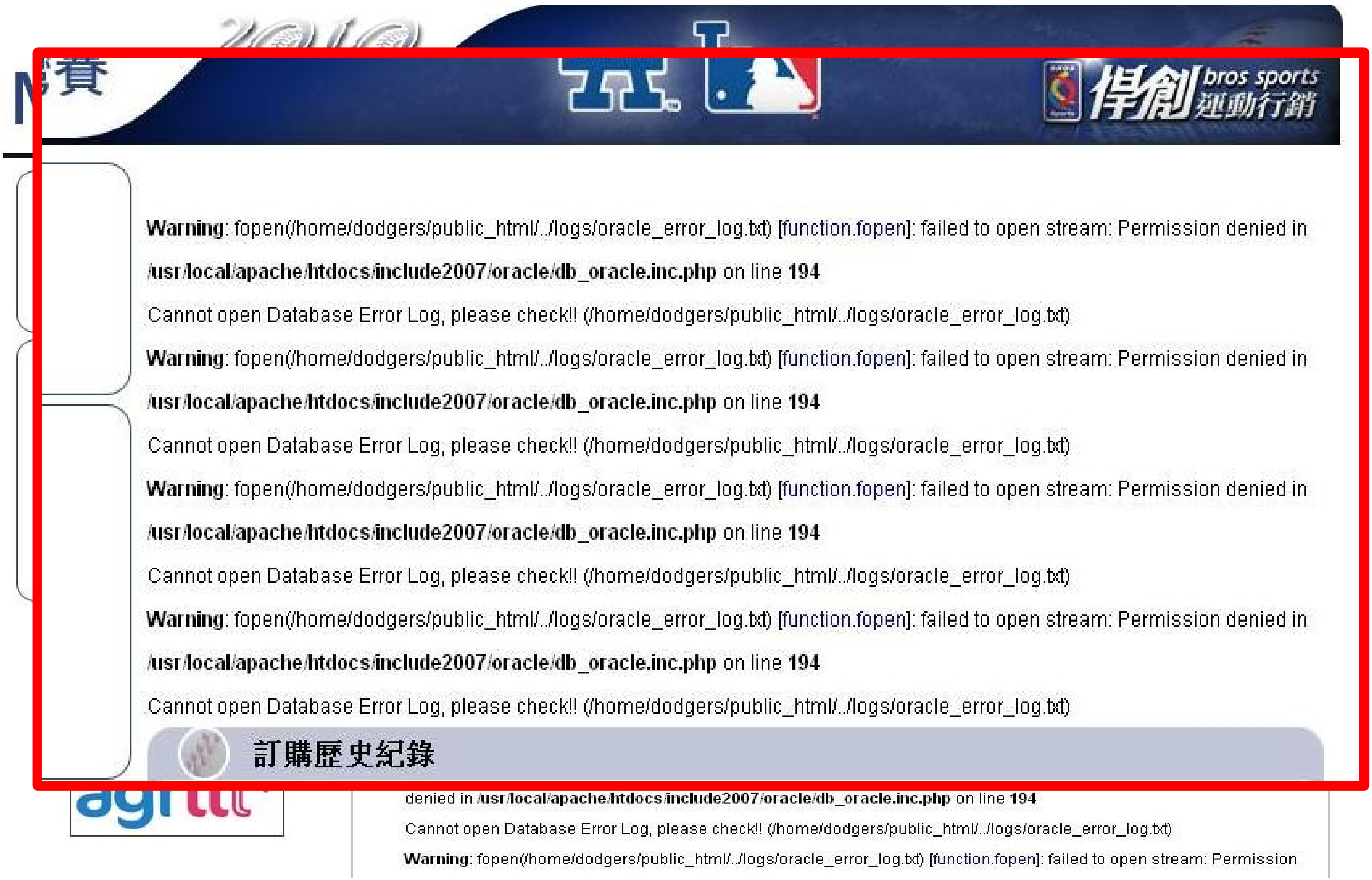
Jazz Wang

Yao-Tsung Wang

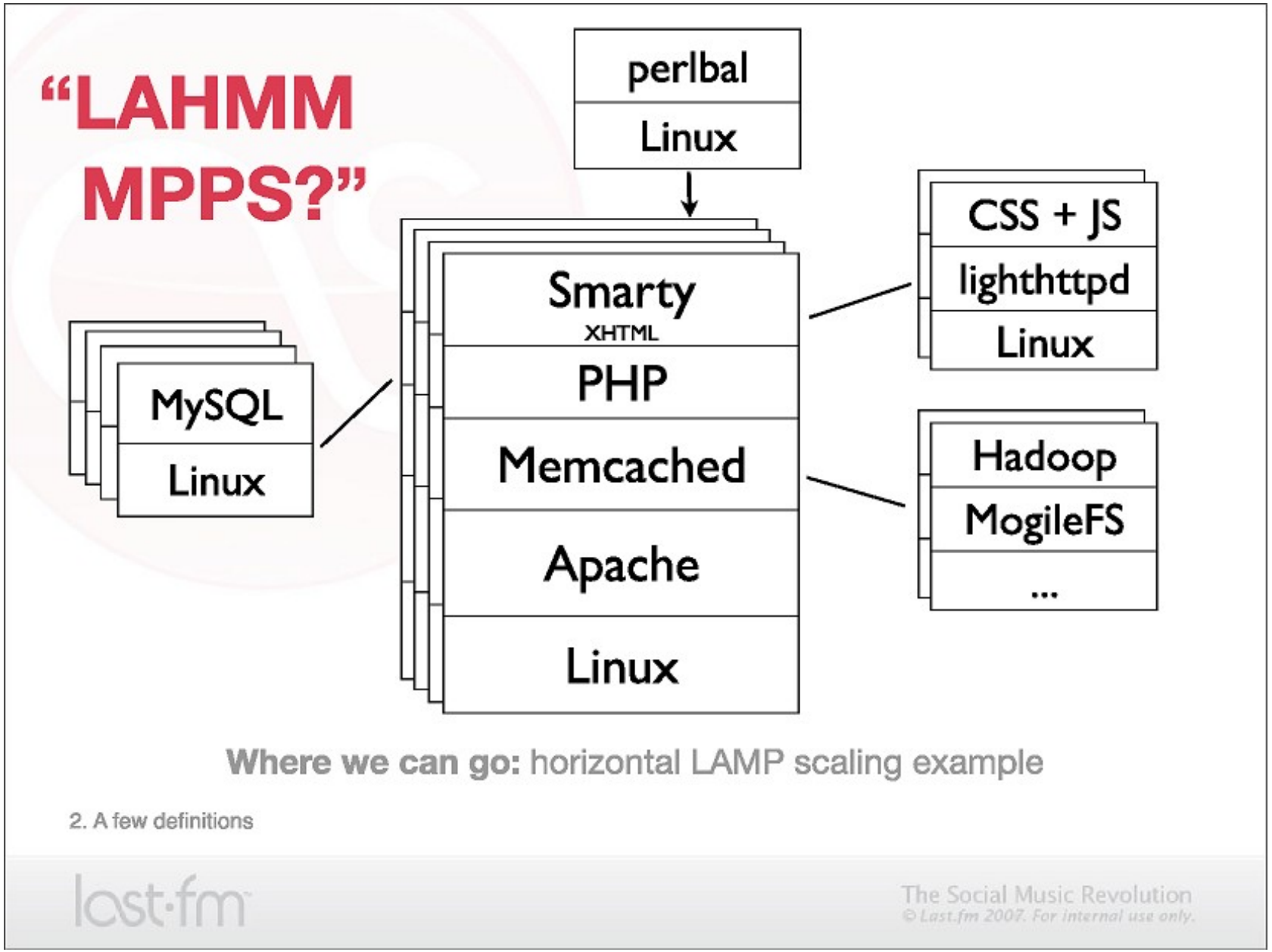
jazz@nchc.org.tw



It's all about SCALE!!



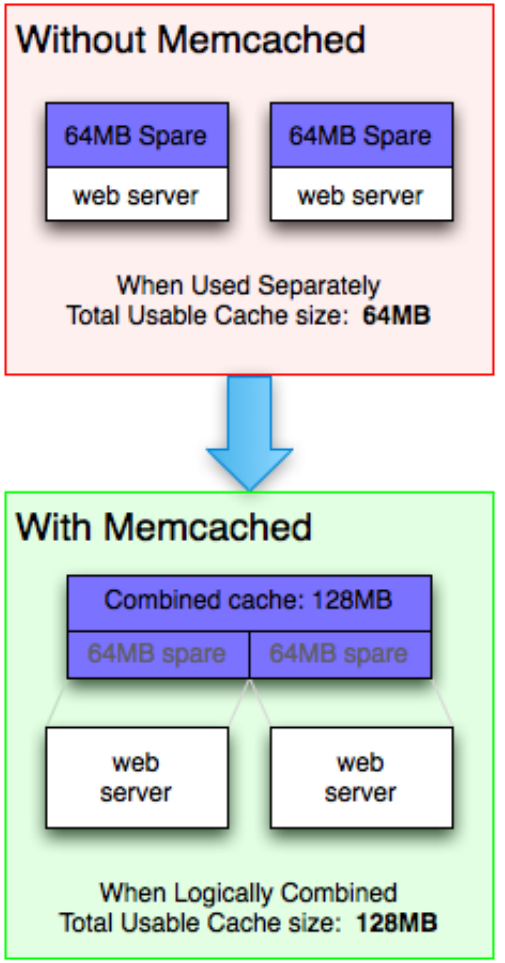
How to scale up web service in the past ?



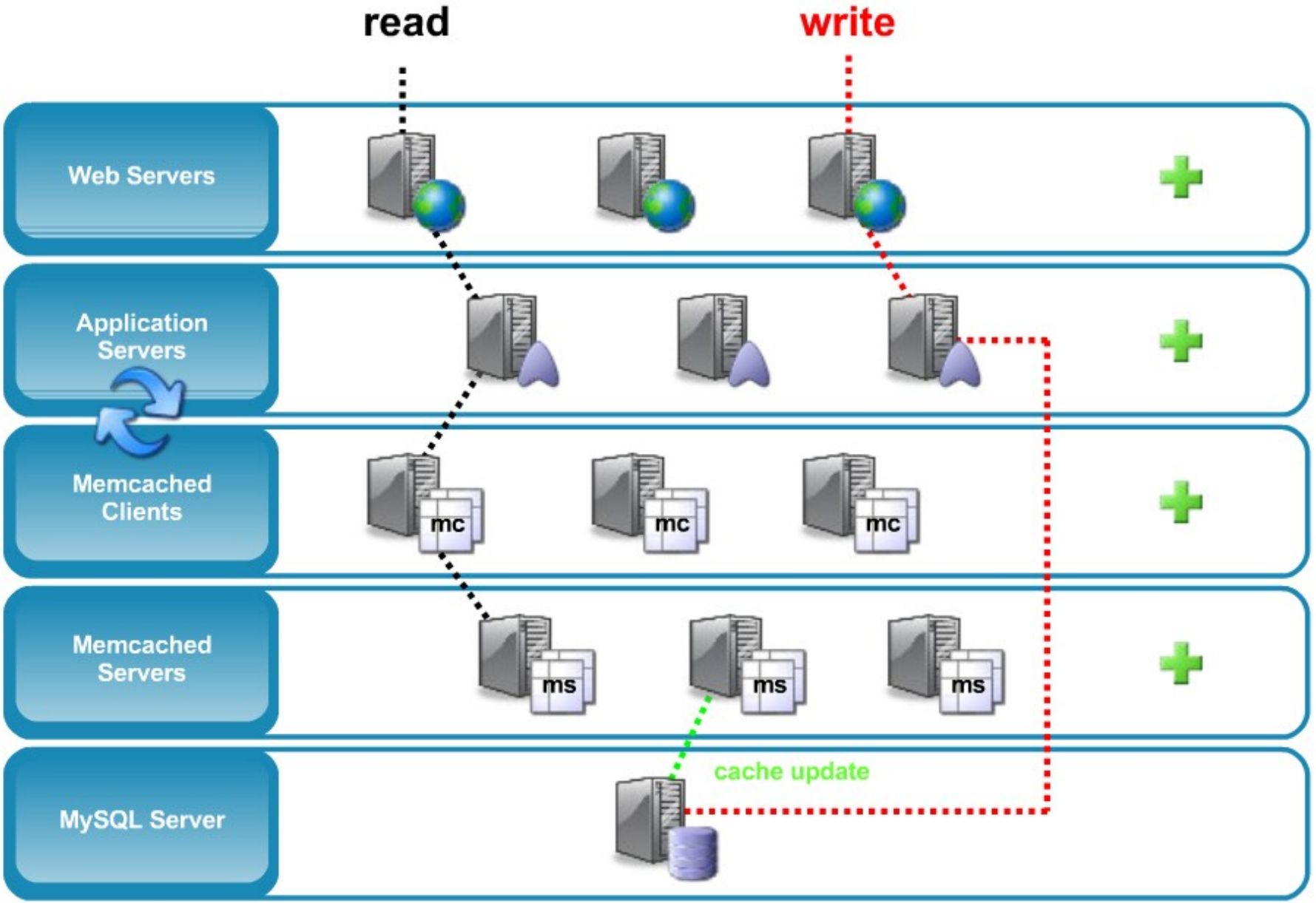
Source: http://www.slideshare.net/mallipeddi/scalable-lamp-development-for-growing-web-apps

Tools used by large scale websites

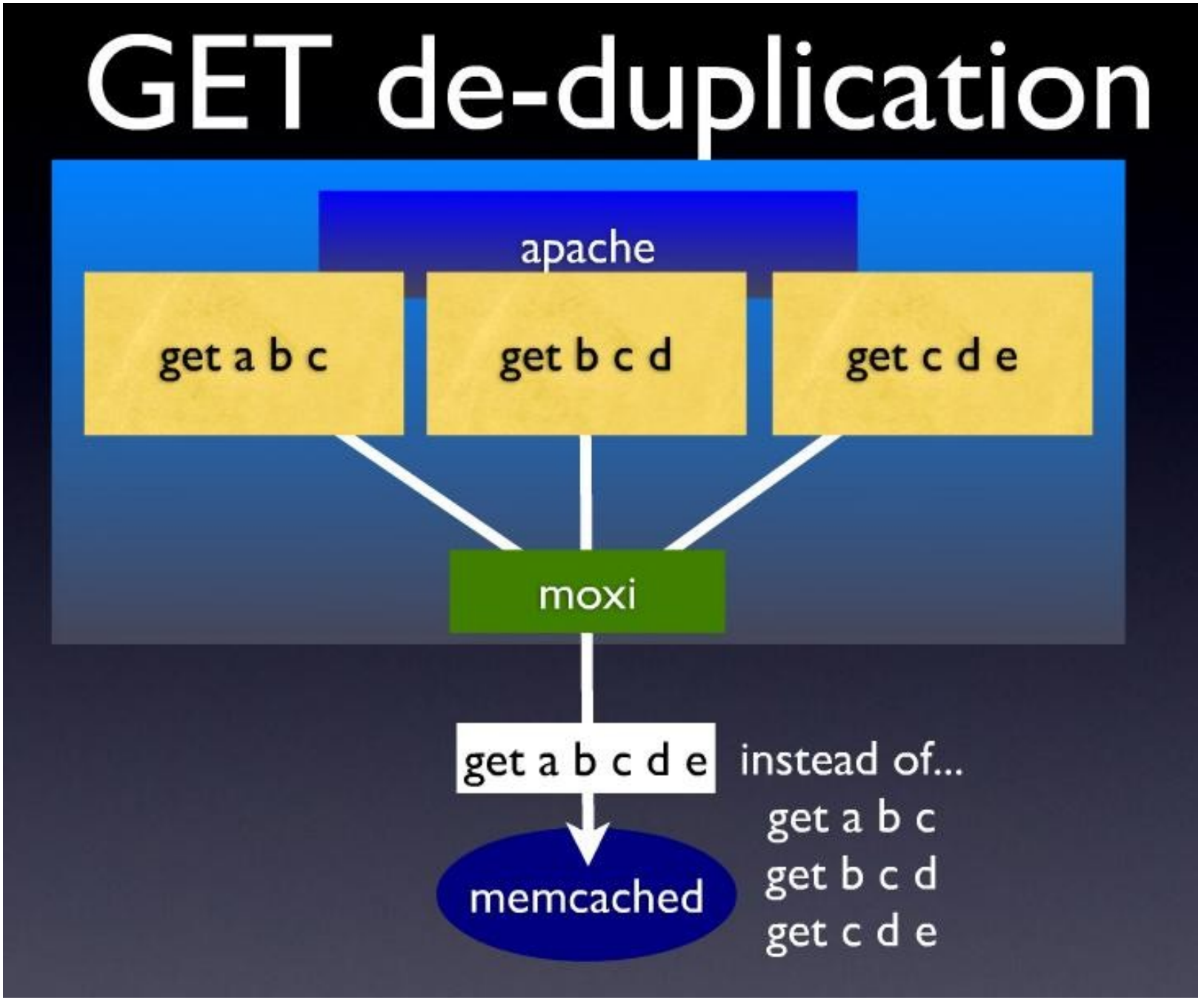
- Perlbal - http://www.danga.com/perlbal/
  - 多個網頁伺服器的負載平衡
  - Load balancer
- MogileFS - http://www.danga.com/mogilefs/
  - 分散式檔案系統
  - Distributed File System fo small files
  - 有公司認爲 MogileFS 比起 Hadoop 適合拿來處理小檔案
- memcached - http://memcached.org/
  - 共享記憶體 ??
  - Share Memory
  - 把資料庫或經常讀取的部分，
    用記憶體快取 (Cache) 方式存放
- Moxi - http://code.google.com/p/moxi/
  - Memcache 的 PROXY
- More Resource:
  - http://code.google.com/p/memcached/wiki/HowToLearnMoreScalability
  - http://www.slideshare.net/techdude/scalable-web-architectures-common-patterns-and-approaches



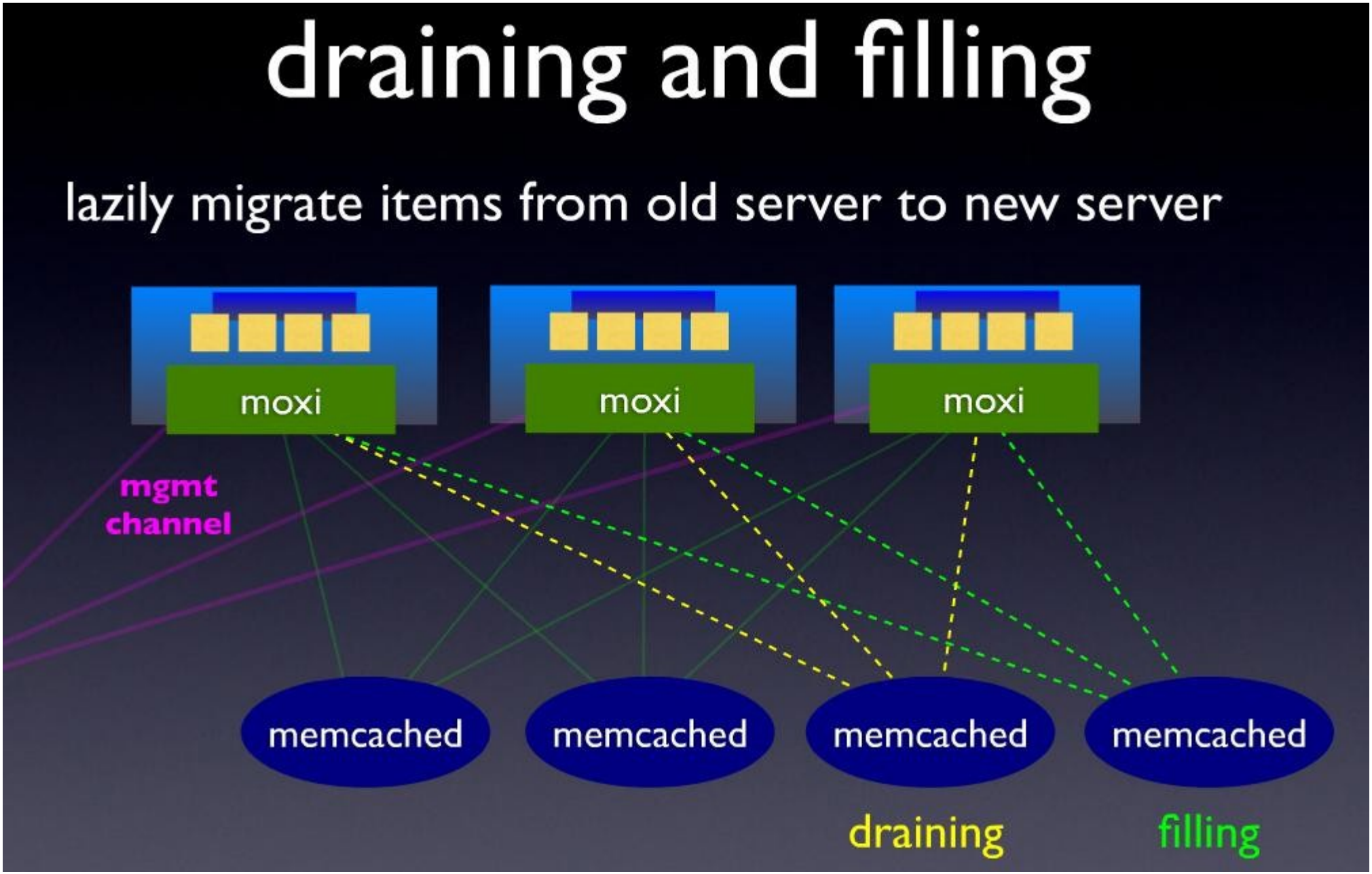
Memcached & MySQL



Source: http://mashraqi.com/2008/07/memcached-for-mysql-advanced-use-cases_09.html



Source: http://www.slideshare.net/northscale/moxi-memcached-proxy



Source: http://www.slideshare.net/northscale/moxi-memcached-proxy

HBase is ..

- HBase is a distributed column-oriented database built on top of HDFS.
- A distributed data store that can scale horizontally to 1,000s of commodity servers and petabytes of indexed storage.
- Designed to operate on top of the Hadoop distributed file system (HDFS) or Kosmos File System (KFS, aka Cloudstore) for scalability, fault tolerance, and high availability.
- Integrated into the Hadoop map-reduce platform and paradigm.

Benefits

- Distributed storage
- Table-like in data structure
  - multi-dimensional map
- High scalability
- High availability
- High performance

Who use 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
  - Migrate from MySQL to 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

Backdrop

- 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
- 2008.10~
  - HBase 0.18, 0.19 released

HBase Is Not …

- Tables have one primary index, the *row key*.
- No join operators.
- Scans and queries can select a subset of available columns, perhaps by using a wildcard.
- There are three types of lookups:
  - Fast lookup using row key and optional timestamp.
  - Full table scan
  - Range scan from region start to end.

HBase Is Not …(2)

- Limited atomicity and transaction support.
  - HBase supports multiple batched mutations of single rows only.
  - Data is unstructured and untyped.
- No accessed or manipulated via SQL.
  - Programmatic access via Java, REST, or Thrift APIs.
  - Scripting via JRuby.

Why Bigtable?

- Performance of RDBMS system is good for transaction processing but for very large scale analytic processing, the solutions are commercial, expensive, and specialized.
- Very large scale analytic processing
  - Big queries – typically range or table scans.
  - Big databases (100s of TB)

Why Bigtable? (2)

- Map reduce on Bigtable with optionally Cascading on top to support some relational algebras may be a cost effective solution.
- Sharding is not a solution to scale open source RDBMS platforms
  - Application specific
  - Labor intensive (re)partitionaing

Why HBase ?

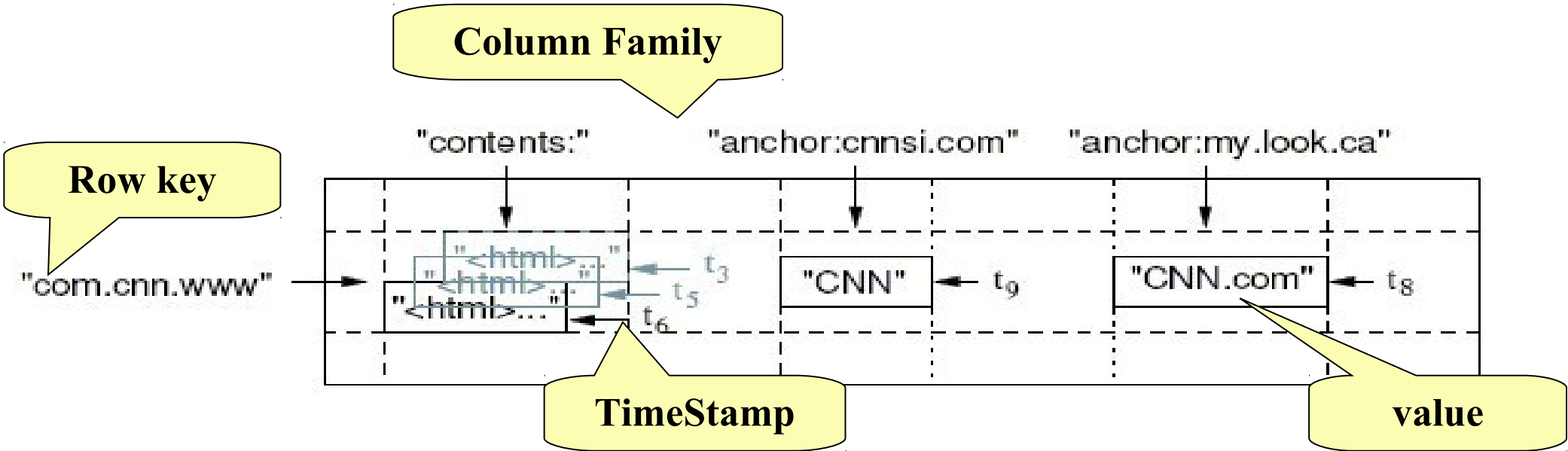
- HBase is a Bigtable clone.
- It is open source
- It has a good community and promise for the future
- It is developed on top of and has good integration for the Hadoop platform, if you are using Hadoop already.
- It has a Cascading connector.

HBase benefits than RDBMS

- *No real indexes*
- *Automatic partitioning*
- *Scale linearly and automatically with new nodes*
- *Commodity hardware*
- *Fault tolerance*
- *Batch processing*

Data Model

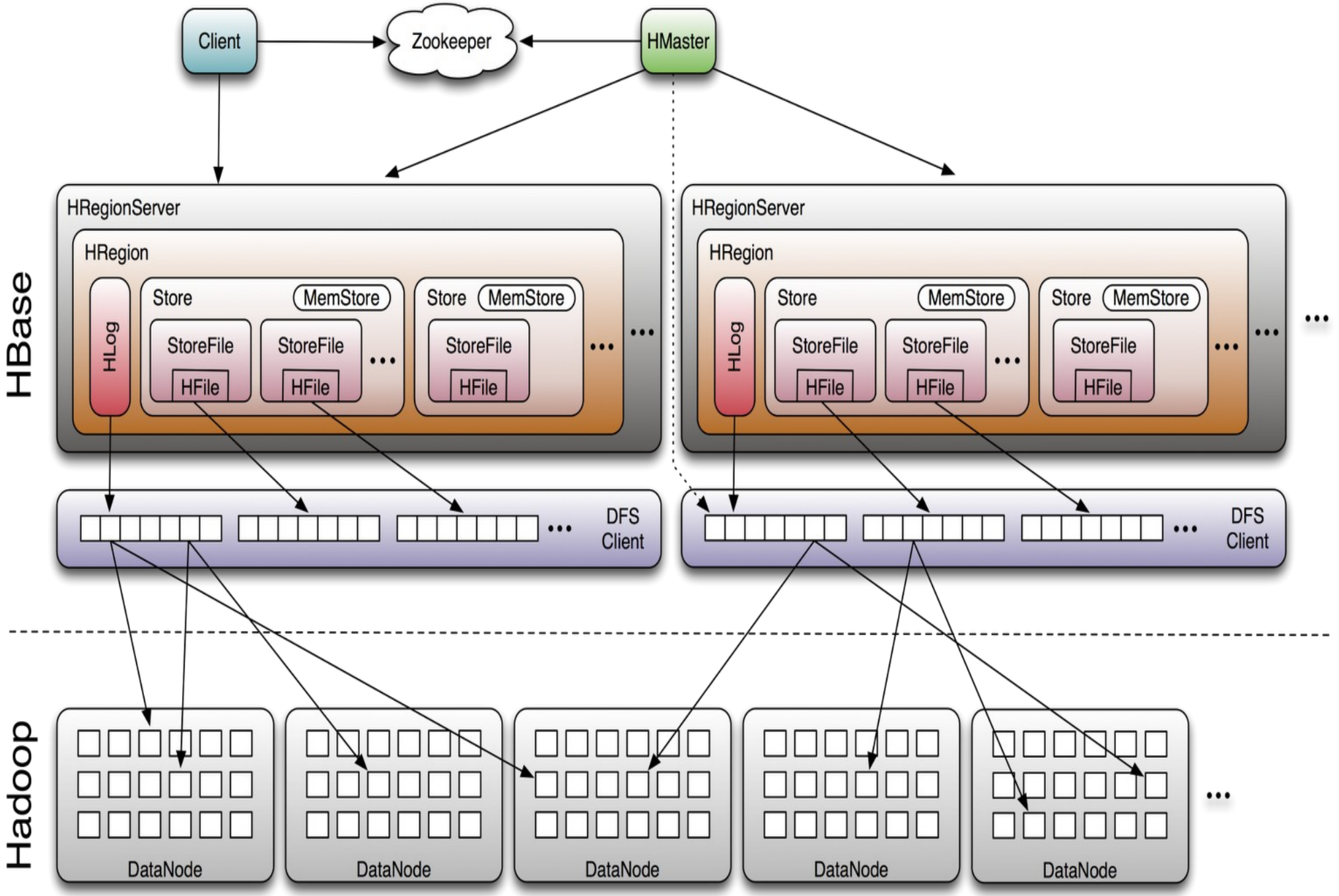
- Tables are sorted by Row
- Table schema only define it's *column families* .
  - Each family consists of any number of columns
  - Each column consists of any number of versions
  - Columns only exist when inserted, NULLs are free.
  - Columns within a family are sorted and stored together
- Everything except table names are byte[]
- (Row, Family: Column, Timestamp) → Value



Members

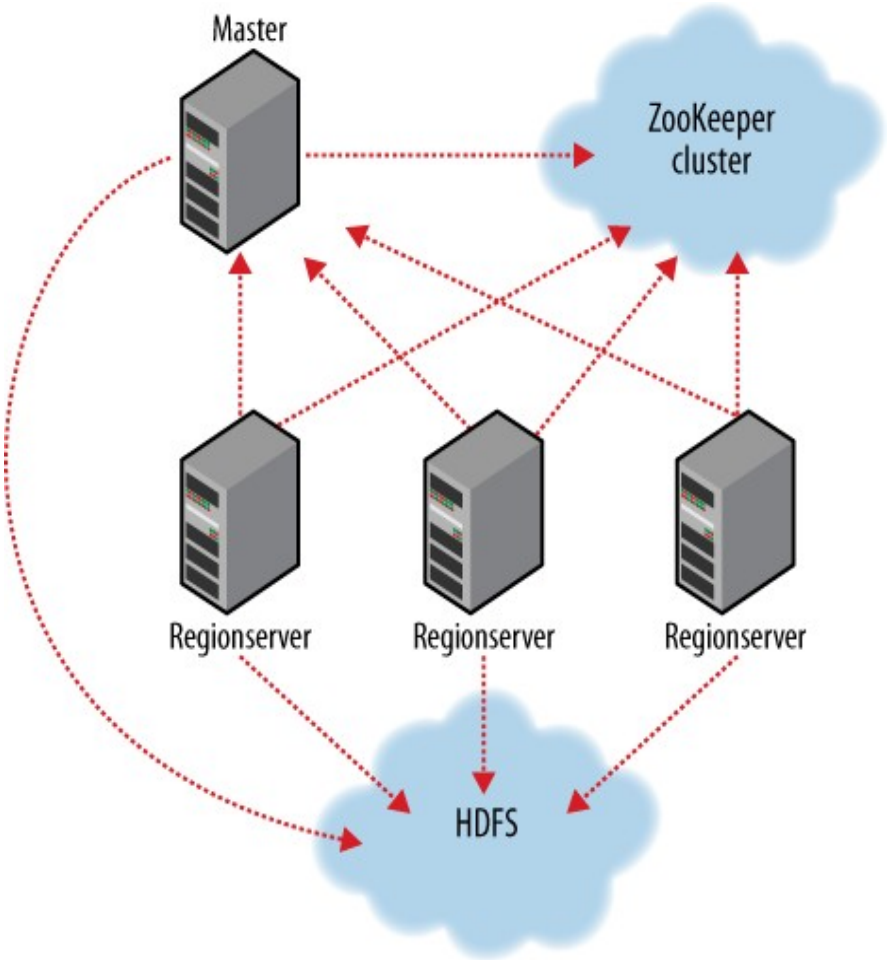
- *Master*
  - Responsible for monitoring region servers
  - Load balancing for regions
  - Redirect client to correct region servers
  - The current SPOF
- *regionserver* slaves
  - Serving requests(Write/Read/Scan) of Client
  - Send HeartBeat to Master
  - Throughput and Region numbers are scalable by region servers

Architecture

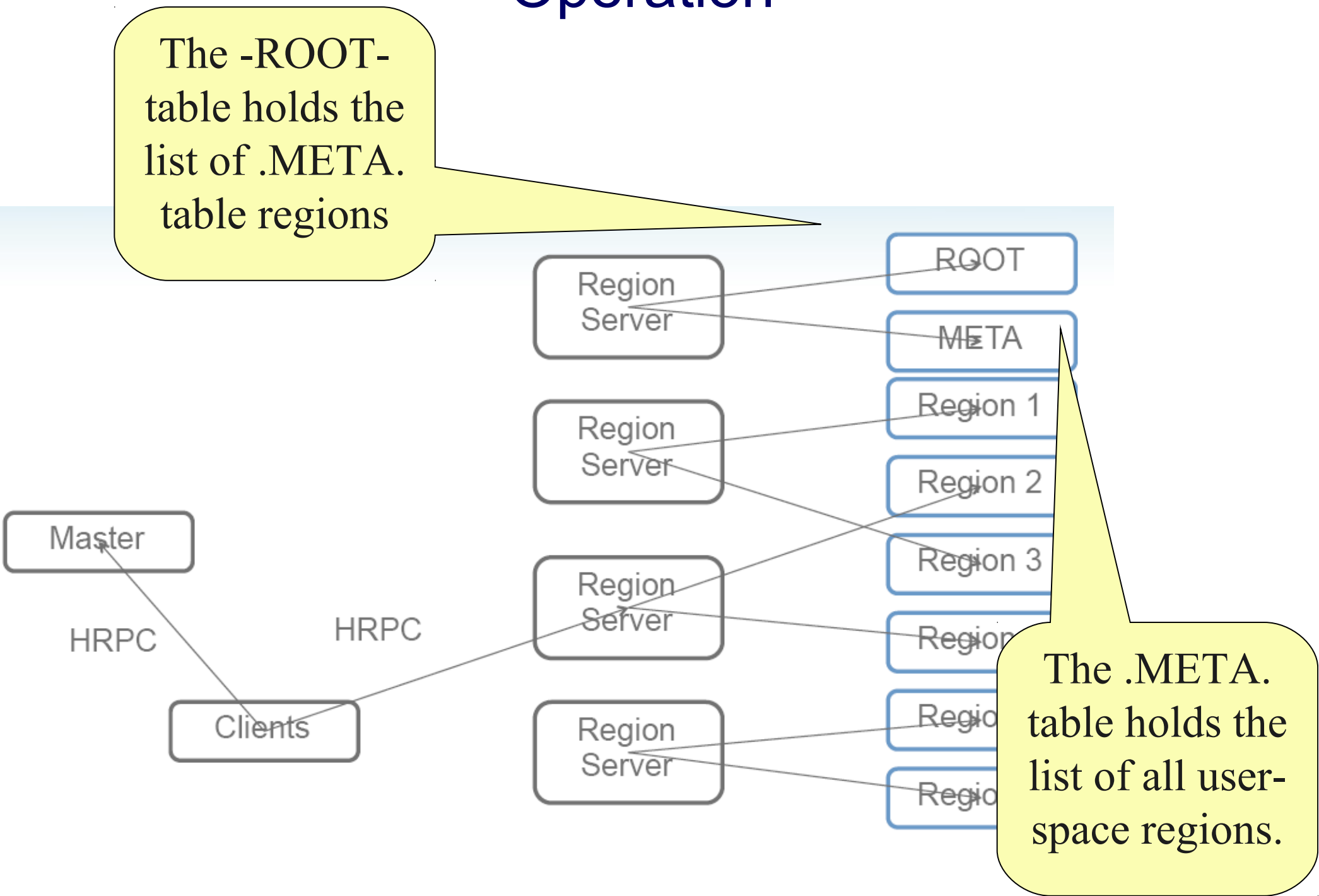


ZooKeeper

- HBase depends on ZooKeeper (Chapter 13) and by default it manages a ZooKeeper instance as the authority on cluster state



Operation


The -ROOT- table holds the list of .META. table regions
Region Server
Region Server
Region Server
Region Server
ROOT
META
Region 1
Region 2
Region 3
Region
Region
Region
Master
HRPC
HRPC
Clients
The .META. table holds the list of all user-space regions.

Questions?

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

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

Powered by DRBL

Introduction to Pig programming

Yahoo Search Engineering

陳奕瑋 (**Yiwei Chen**)

任務！

殺很大、插很大(+瑤瑤寫真性感精選54P) @ osaki's Blog :: Xuite日誌

殺不用錢~~殺online瑤瑤性感變裝照+精選性感寫真童顏巨乳的娃娃音美少女瑤瑤 本名：郭書瑤 暱稱：瑤瑤身高：155cm 體重：42kg 三圍：33E/23/33 生日：1990/7/18 ...

blog.xuite.net/osaki99/blog/21865265 - 頁庫存檔 - 類似內容

電玩美少女瑤瑤精選影音(ヤオヤオ童顏Fカップ爆乳美少女映画videos ...

2008年8月31日 ... 18歲電玩少女瑤瑤半工半讀扛家計 (內有瑤瑤男友) http://blog.xuite.net /kaiger/daily/23136438 ... 20080913 我猜嗲嗲美少女第二段2號Kiki 3號瑤瑤 ...

blog.xuite.net/kaiger/daily/19128818 - 頁庫存檔 - 類似內容

顯示更多來自 blog.xuite.net 的結果

jays1943 分享正妹:NO.24 無名瑤瑤- 樂多日誌

瑤瑤也沒有哪裡得罪你們押你們為審麼這樣罵他說害女生生氣我看你們長的很醜吧不要自以為是喔死網友還罵人く死勒你要不要臉瑤瑤可是我的偶像你們最好是向一點 ...

blog.roodo.com/jays1943/archives/6850053.html - 頁庫存檔 - 類似內容

任務！

瑤瑤航空 - Powered by Discuz!

瑤瑤航空 - Discuz! Board ... 歡迎VIP旅客-魏如昀加入瑤瑤航空(2008-4-7) 歡迎VIP旅客-賴鋐偉加入瑤瑤航空(2008-3-13) ... 瑤瑤家族 瑤瑤在飛虎的第一家族 瑤瑤天空部落格 林佩瑤在天空的部落格 林佩瑤 無名網誌 瑤瑤的新照片都在無名啦! 無不癡齋 ...

www.yaoyaofly.com - 庫存頁面 - 更多此站結果

瑤瑤喵小屋~ - 無名小站

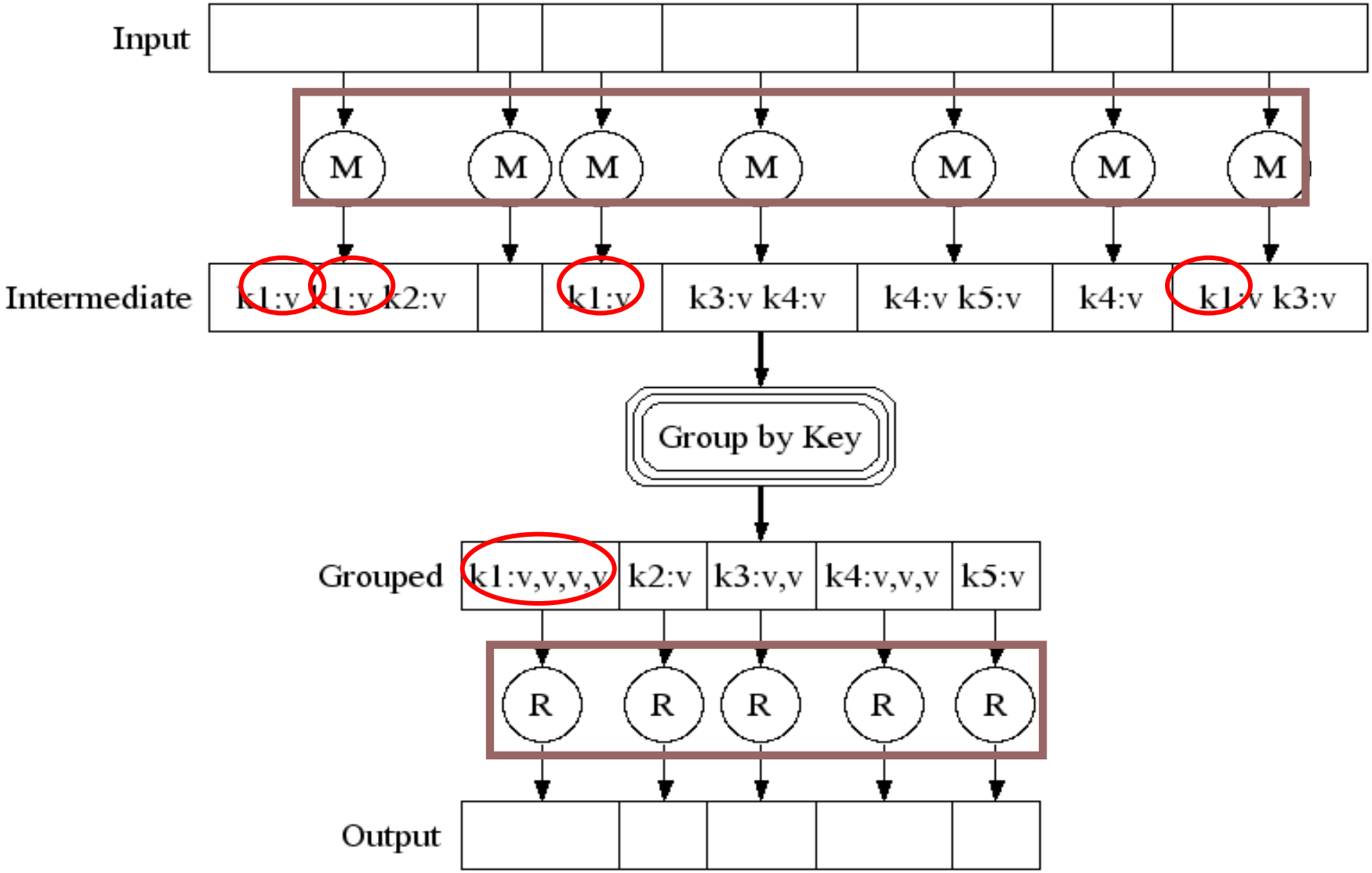
瑤瑤喵小屋~ - 無名小站 Blog Album... 最近好煩煩煩. 我覺得我的腦容量變小了... 好多事情消化不良 好多念頭讓我無法抉擇 (More.) goukigouki at 無名小站 at 02:39 PM post | Reply(27) | Trackback(0) | prosecute ...

www.wretch.cc/blog/goukigouki - 74k - 庫存頁面 - 更多此站結果

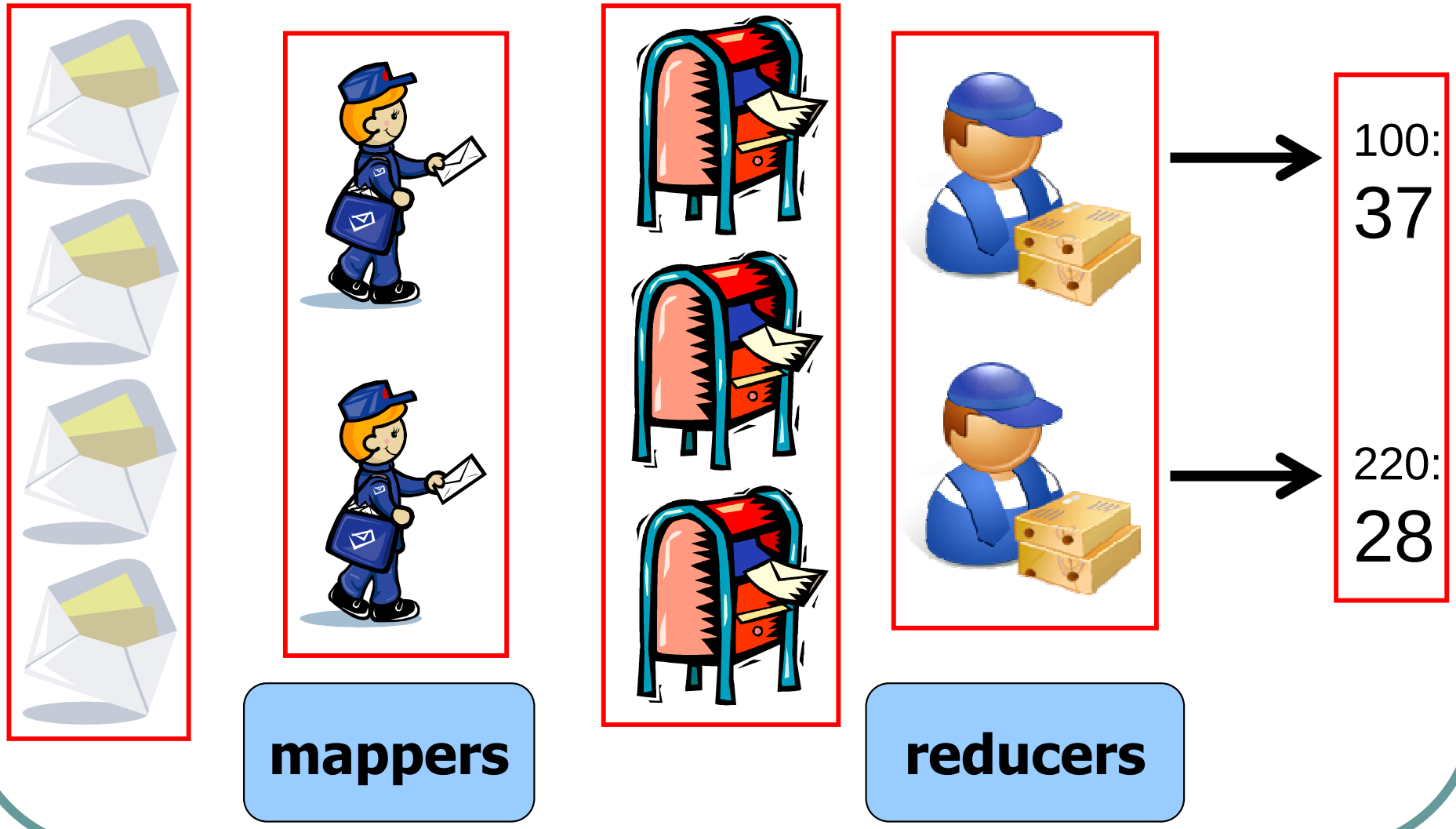
任務！

- 你怎麼知道我們放的網頁比較好？
- 你怎麼知道第一筆結果應該要多熱門？

Hadoop Programming – Map/Reduce



Map / Reduce



Map-Reduce

- 全新想法
- 須分別撰寫 mappers & reducers
- 會有超級無敵霹靂多的 mapper/reducer 要維護！

We usually do …

- 大部份時候:
  - filtering, projecting
  - grouping, aggregation, joining
- 今天有多少人搜尋「美國生」

Pig (Latin)

- Procedural dataflow language (Pig Latin) for Map-Reduce
  - 很像 SQL
    - group, join, filter, sort …
  - 人人都會 SQL

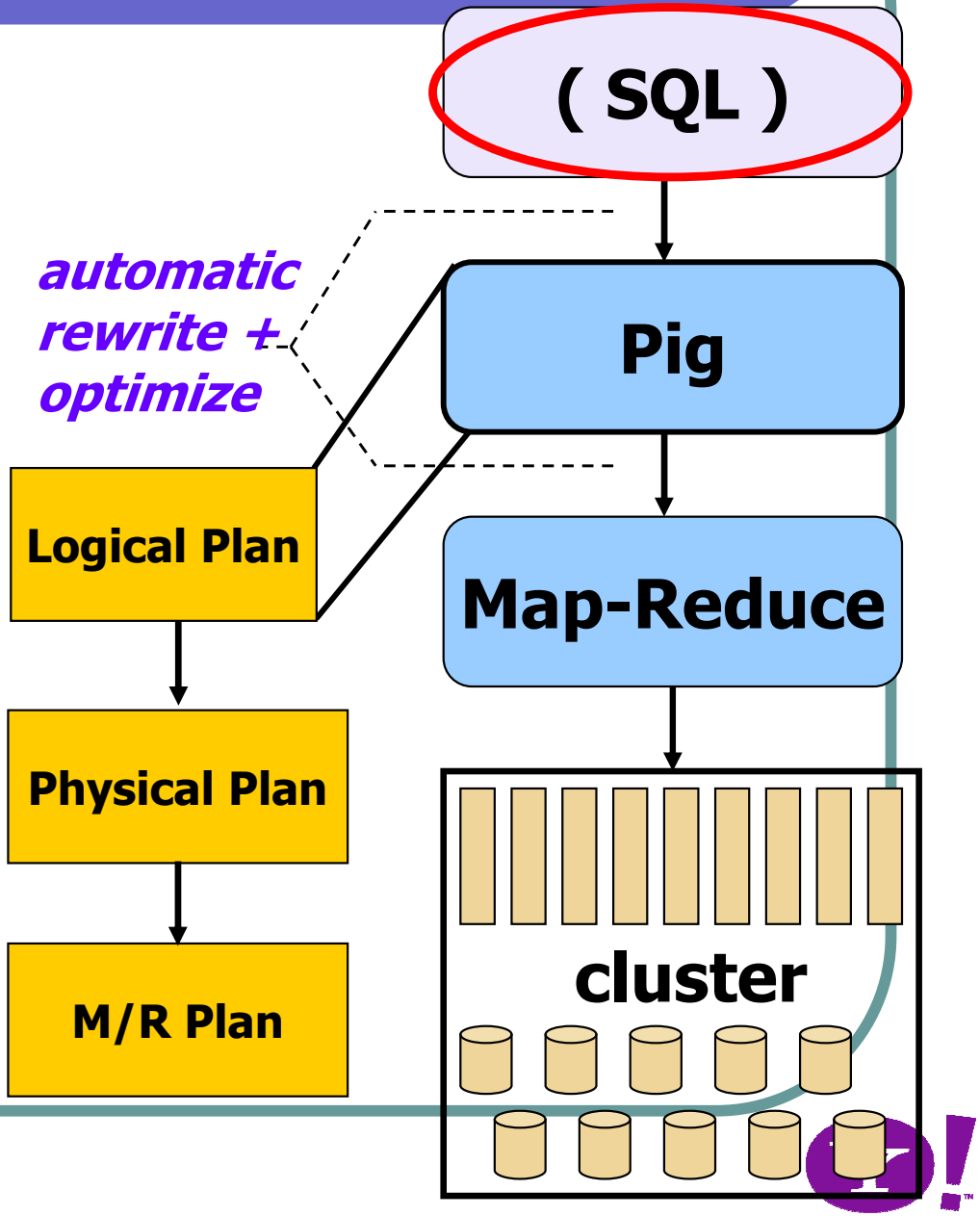
Pig Script Example

- Top sites visited by users aged 18 to 25

```
Users = LOAD ‘users.in’ AS (name, age);
Fltrd = FILTER Users by age >= 18 and age <= 25;

Pages = LOAD ‘pages.in’ AS (user, url);

Jnd   = JOIN Fltrd BY name, Pages BY user;
Grpd  = GROUP Jnd by url;
Smmd  = FOREACH Grpd GENERATE group, COUNT(Jnd) AS
        clicks;

Srtd  = ORDER Smmd BY clicks;
Top100 = LIMIT Srtd 100;

STORE Top100 INTO ‘top100sites.out’;
```

Pig script → Map/Reduce

- 不需懂底下 Map-Reduce 運作
- Pig 幫忙翻譯



Why Pig?

- 容易學
- 開發快
- 一目瞭然

Why Pig?

```
Users = LOAD ‘users’ AS (name, age);
Fltrd = FILTER Users by age >= 18 and age <= 25;
Pages = LOAD ‘pages’ AS (user, url);
Jnd   = JOIN Fltrd BY name, Pages BY user;
Grpd  = GROUP Jnd by url;
Smmd  = FOREACH Grpd GENERATE group, COUNT(Jnd) AS
        clicks;
Srtd  = ORDER Smmd BY clicks;
Top100 = LIMIT Srtd 100;

STORE Top100 INTO ‘top100sites’;
```

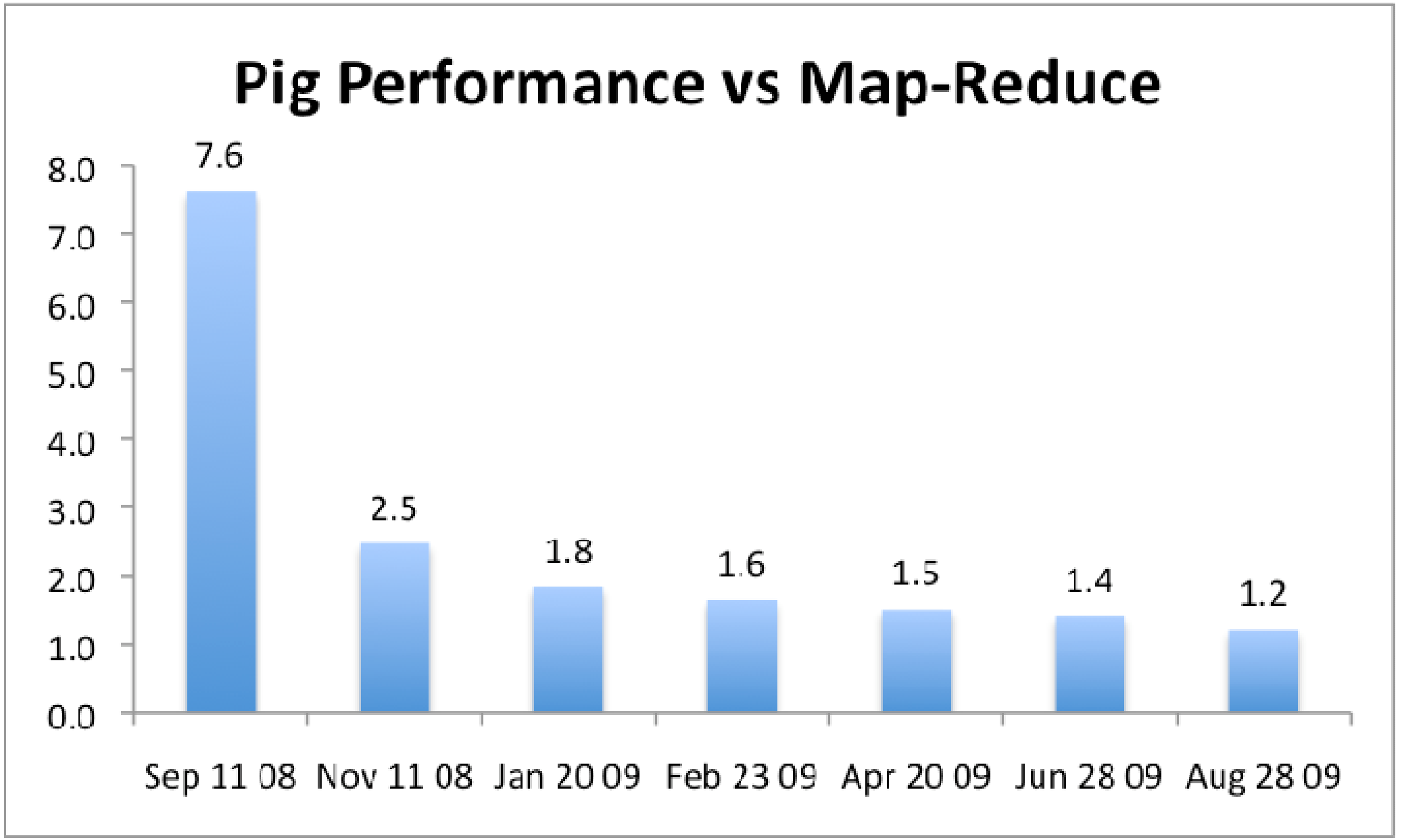
```
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.WritableComparable;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.KeyValueTextInputFormat;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.RecordReader;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.mapred.SequenceFileInputFormat;
import org.apache.hadoop.mapred.SequenceFileOutputFormat;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.jobcontrol.Job;
import org.apache.hadoop.mapred.jobcontrol.JobControl;
import org.apache.hadoop.mapred.lib.IdentityMapper;

public class MRExample {
    public static class LoadPages extends MapReduceBase
        implements Mapper<LongWritable, Text, Text, Text> {

        public void map(LongWritable k, Text val,
                OutputCollector<Text, Text> oc,
                Reporter reporter) throws IOException {
            // Pull the key out
            String line = val.toString();
            int firstComma = line.indexOf(',');
            String key = line.substring(0, firstComma);
            String value = line.substring(firstComma + 1);
            Text outKey = new Text(key);
            // Prepend an index to the value so we know which file
            // it came from.
            Text outVal = new Text("1" + value);
            oc.collect(outKey, outVal);
        }
    }
    public static class LoadAndFilterUsers extends MapReduceBase
        implements Mapper<LongWritable, Text, Text, Text> {

        public void map(LongWritable k, Text val,
                OutputCollector<Text, Text> oc,
                Reporter reporter) throws IOException {
            // Pull the key out
            String line = val.toString();
            int firstComma = line.indexOf(',');
            String value = line.substring(firstComma + 1);
            int age = Integer.parseInt(value);
            if (age < 18 || age > 25) return;
            String key = line.substring(0, firstComma);
            Text outKey = new Text(key);
            // Prepend an index to the value so we know which file
            // it came from.
            Text outVal = new Text("2" + value);
            oc.collect(outKey, outVal);
        }
    }
    public static class Join extends MapReduceBase
        implements Reducer<Text, Text, Text, Text> {

        public void reduce(Text key,
                Iterator<Text> iter,
                OutputCollector<Text, Text> oc,
                Reporter reporter) throws IOException {
            // For each value, figure out which file it's from and
store it
            // accordingly.
            List<String> first = new ArrayList<String>();
            List<String> second = new ArrayList<String>();

            while (iter.hasNext()) {
                Text t = iter.next();
                String value = t.toString();
                if (value.charAt(0) == '1')
first.add(value.substring(1));
                else second.add(value.substring(1));
                reporter.setStatus("OK");
            }

            // Do the cross product and collect the values
            for (String s1 : first) {
                for (String s2 : second) {
                    String outval = key + "," + s1 + "," + s2;
                    oc.collect(null, new Text(outval));
                    reporter.setStatus("OK");
                }
            }
        }
    }
    public static class LoadJoined extends MapReduceBase
        implements Mapper<Text, Text, Text, LongWritable> {

        public void map(
                Text key,
                Text val,
                OutputCollector<Text, LongWritable> oc,
                Reporter reporter) throws IOException {
            // Find the url
            String line = val.toString();
            int firstComma = line.indexOf(',');
            int secondComma = line.indexOf(',', firstComma);
            String key = line.substring(firstComma, secondComma);
            // drop the rest of the record, don't need it anymore,
            // just pass a 1 for the combiner/reducer to sum instead.
            Text outKey = new Text(key);
            oc.collect(outKey, new LongWritable(1L));
        }
    }
    public static class ReduceUrls extends MapReduceBase
        implements Reducer<Text, LongWritable, WritableComparable,
Writable> {

        public void reduce(
                Text key,
                Iterator<LongWritable> iter,
                OutputCollector<WritableComparable, Writable> oc,
                Reporter reporter) throws IOException {
            // Add up all the values we see

            long sum = 0;
            while (iter.hasNext()) {
                sum += iter.next().get();
                reporter.setStatus("OK");
            }

            oc.collect(key, new LongWritable(sum));
        }
    }
    public static class LoadClicks extends MapReduceBase
        implements Mapper<WritableComparable, Writable, LongWritable,
Text> {

        public void map(
                WritableComparable key,
                Writable val,
                OutputCollector<LongWritable, Text> oc,
                Reporter reporter) throws IOException {
            oc.collect((LongWritable)val, (Text)key);
        }
    }
    public static class LimitClicks extends MapReduceBase
        implements Reducer<LongWritable, Text, LongWritable, Text> {

        int count = 0;
        public void reduce(
            LongWritable key,
            Iterator<Text> iter,
            OutputCollector<LongWritable, Text> oc,
            Reporter reporter) throws IOException {

            // Only output the first 100 records
            while (count < 100 && iter.hasNext()) {
                oc.collect(key, iter.next());
                count++;
            }
        }
    }
    public static void main(String[] a
        JobConf lp = new JobConf(MRExa
        lp.setJobName("Load Pages");
        lp.setInputFormat(TextInputFor
        lp.setOutputKeyClass(Text.clas
        lp.setOutputValueClass(Text.cl
        lp.setMapperClass(LoadPages.cl
        FileInputFormat.addInputPath(l
Path("/user/gates/pages"));
        FileOutputFormat.setOutputPath
            new Path("/user/gates/tmp/
        lp.setNumReduceTasks(0);
        Job loadPages = new Job(lp);

        JobConf lfu = new JobConf(MRExa
        lfu.setJobName("Load and Filte
        lfu.setInputFormat(TextInputFo
        lfu.setOutputKeyClass(Text.cla
        lfu.setOutputValueClass(Text.c
        lfu.setMapperClass(LoadAndFilt
        FileInputFormat.addInputPath(l
Path("/user/gates/users"));
        FileOutputFormat.setOutputPath
            new Path("/user/gates/tmp/
        lfu.setNumReduceTasks(0);
        Job loadUsers = new Job(lfu);

        JobConf join = new JobConf(MRE
        join.setJobName("Join Users an
        join.setInputFormat(KeyValueTe
        join.setOutputKeyClass(Text.cl
        join.setOutputValueClass(Text.
        join.setMapperClass(IdentityMa
        join.setReducerClass(Join.clas
        FileInputFormat.addInputPath(j
Path("/user/gates/tmp/indexed_pages"))
        FileInputFormat.addInputPath(
Path("/user/gates/tmp/filtered_users"
        FileOutputFormat.setOutputPath
Path("/user/gates/tmp/joined"));
        join.setNumReduceTasks(50);
        Job joinJob = new Job(join);
        joinJob.addDependingJob(loadPa
        joinJob.addDependingJob(loadUs

        JobConf group = new JobConf(MR
        group.setJobName("Group URLs")
        group.setInputFormat(KeyValueT
        group.setOutputKeyClass(Text.c
        group.setOutputValueClass(Long
        group.setOutputFormat(Sequence
        group.setMapperClass(LoadJoine
        group.setCombinerClass(ReduceU
        group.setReducerClass(ReduceUr
        FileInputFormat.addInputPath(g
Path("/user/gates/tmp/joined"));
        FileOutputFormat.setOutputPath
Path("/user/gates/tmp/grouped"));
        group.setNumReduceTasks(50);
        Job groupJob = new Job(group);
        groupJob.addDependingJob(joinJ

        JobConf top100 = new JobConf(MR
        top100.setJobName("Top 100 sit
        top100.setInputFormat(Sequence
        top100.setOutputKeyClass(LongW
        top100.setOutputValueClass(Tex
        top100.setOutputFormat(Sequenc
        top100.setMapperClass(LoadClic
        top100.setCombinerClass(LimitC
        top100.setReducerClass(LimitCl
        FileInputFormat.addInputPath(t
Path("/user/gates/tmp/grouped"));
        FileOutputFormat.setOutputPath
Path("/user/gates/top100sitesforusers1
        top100.setNumReduceTasks(1);
        Job limit = new Job(top100);
        limit.addDependingJob(groupJob

        JobControl jc = new JobControl
18 to 25");
        jc.addJob(loadPages);
```

Why (NOT) Pig?

- 不是史上究極霹靂大無敵武器
  - Focus: aggregation, filter, join,…
- 另一種做分散運算工作的方式

Sweet spot between SQL – M/R

	SQL	Pig	Map-Reduce
Programming style	Large blocks of declarative constraints	➔	"Plug together pipes"
Built-in data manipulations	Group-by, Sort, Join, Filter, Aggregate, Top-k, etc...	➔	Group-by, Sort
Execution model	Fancy; trust the query optimizer	➔	Simple, transparent
Opportunities for automatic optimization	Many	➔	Few (logic buried in map() and reduce())
Data Schema	Must be known at table creation	➔	Not required, may be defined at runtime


Pig Performance vs Map-Reduce
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
7.6
2.5
1.8
1.6
1.5
1.4
1.2
Sep 11 08
Nov 11 08
Jan 20 09
Feb 23 09
Apr 20 09
Jun 28 09
Aug 28 09

Execution and Syntax

Pig Example

- Show users aged 18-25

```
Users = LOAD 'users.txt'
        USING PigStorage(',') AS (name, age);
Fltrd = FILTER Users
        BY age >= 18 AND age <= 25;
Names = FOREACH Fltrd GENERATE name;

STORE Names INTO 'names.out';
```

How to execute

- Local:
  - `pig -x local foo.pig`
- Hadoop (HDFS):
  - `pig foo.pig`
  - `pig -Dmapred.job.queue.name=xxx foo.pig`
    - `hadoop queue -showacls`

How to execute

- Interactive pig shell
  - `$ pig`
  - grunt> _

Load Data

```
Users = LOAD 'users.txt'
        USING PigStorage(',') AS (name, age);
```

- LOAD … AS …
- PigStorage(‘,’) to specify separator

```
John,18
Mary,20
Bob,30
```

name	age
John	18
Mary	20
Bob	30

Filter

```
Fltrd = FILTER Users
        BY age >= 18 AND age <= 25;
```

- FILTER … BY …
  - constraints can be composite

name	age
John	18
Mary	20
Bob	30

name	age
John	18
Mary	20

Generate / Project

```
Names = FOREACH Fltrd GENERATE name;
```

- FOREACH … GENERATE

name	age
John	18
Mary	20

name
John
Mary

Store Data

```
STORE Names INTO 'names.out';
```

- STORE … INTO …
  - PigStorage(‘,’) to specify separator if multiple fields

Command - JOIN

```
Users = LOAD 'users' AS (name, age);
Pages = LOAD 'pages' AS (user, url);
Jnd   = JOIN Users BY name, Pages BY user;
```

name	age
John	18
Mary	20
Bob	30

user	url
John	yaho
Mary	goog
Bob	bing

name	age	user	url
John	18	John	yaho
Mary	20	Mary	goog
Bob	30	Bob	bing

Command - GROUP

```
Grpd  = GROUP Jnd by url;
describe Grpd;
```

name	age	url
John	18	yhoo
Mary	20	goog
Dee	25	yhoo
Kim	40	bing
Bob	30	bing

yhoo	(John, 18, yhoo) (Dee, 25, yhoo)
goog	(Mary, 20, goog)
bing	(Kim, 40, bing) (Bob, 30, bing)

Other Commands

- `PARALLEL` – controls #reducer
- `ORDER` – sort by a field
- `COUNT` – eval: count #elements
- `COGROUP` – structured JOIN
- More at http://hadoop.apache.org/pig/docs/r0.5.0/piglatin_reference.html

Features

Parameter Substitution

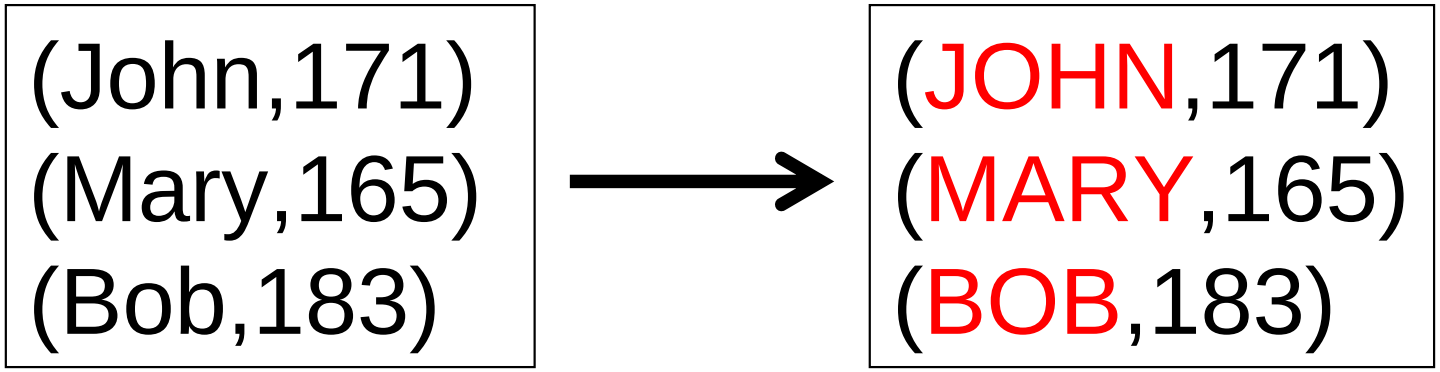
```
%default TYPE 'view'
%declare ID '18987'
A = load '/data/$DATE/$ID/$TYPE‘
```

- $ pig a.pig
- $ pig -param DATE=20091009 a.pig
- $ pig -param DATE=20091009 –param TYPE=click a.pig

RegEx Comparison

- `itsyou = FILTER urls by ($0 MATCHES 'http://.*\\.yahoo\\.com.*')`
- MATCHES matches ‘whole’ string
  - `‘aaaa’ MATCHES ‘aaa.*’` is true
  - `‘bbaaaa’ MATCHES ‘aaa.*’` is false
- pattern syntax: java.util.regex.Pattern

User-defined Function (UDF)



UDF – user function part

```
package myudf;
import java.io.IOException;
import org.apache.pig.EvalFunc;
import org.apache.pig.data.Tuple;

public class UPPER extends EvalFunc<String>
{
    public String exec(Tuple in) throws IOException {
        if (in == null || in.size() == 0) return null;
        String str = (String)in.get(0);
        return str.toUpperCase();
    }
}
```

UDF

- http://hadoop.apache.org/pig/docs/r0.3.0/udf.html
- http://hadoop.apache.org/pig/javadoc/docs/api/
- PiggyBank
  - Pig users UDF repo
  - http://wiki.apache.org/pig/PiggyBank

Embedded in Java

```
/* create a pig server in the main class*/
{
  PigServer pigserver = new PigServer(args[0]);
   runMyQuery(pigServer, "/user/viraj/mydata.txt")
}

/* submit in function runMyQuery */

runMyQuery(PigServer pigServer, String inputFile) throws
IOException {
        pigServer.registerQuery("A = load '" + inputFile +
"' as (f1,f2,f3);");
        pigServer.registerQuery("B = group A by f1;");
        pigServer.registerQuery("C = foreach B generate
flatten(group);");
        pigServer.store("C", "/user/viraj/myoutput");
}
```

References

- FAQ
  - http://wiki.apache.org/pig/FAQ
- Documentation
  - http://hadoop.apache.org/pig/docs/r0.5.0/
- Talks & papers
  - http://wiki.apache.org/pig/PigTalksPapers
  - http://www.cloudera.com/hadoop-training-pig-introduction

Questions?

Backup slides

Parameter Substitution

```
$ pig -param myparam=val foo.pig
```

```
B = filter A by ($0 eq '$myparam');
```

- `pig -dryrun` produces processed script

```
B = filter A by ($0 eq 'val');
```

Parameter Substitution

- Params in file instead of command line
- `$ pig -param_file myparams.txt a.pig`

```
#myparams.txt
DATE=20081009
TYPE=clicks
```

UDF – build user function

- javac
  -cp $PIG_HOME/lib/pig.jar
  -sourcepath src
  -d classes
  src/myudf/UPPER.java
- jar cf myudf.jar -C classes myudf/UPPER.class

UDF – pig latin part

```
register myudf.jar;
B =
foreach A generate
  myudf.UPPER(name),height;
```

SQL vs. Pig Latin

SQL	Pig	Example
From table	Load file(s)	**SQL**: from X; **Pig**: A = load 'mydata' using PigStorage('\t') as (col1, col2, col3);
Select	Foreach … generate	**SQL**: select col1 + col2, col3 … **Pig**: B = foreach A generate col1 + col2, col3;
Where	Filter	**SQL**: select col1 + col2, col3 from X where col2>2; **Pig**: C = filter B by col2 > '2';

(adapted from Viraj's slide)

SQL vs. Pig Latin

SQL	Pig	Example
Group by	Group + foreach … generate	**SQL:** select col1, col2, sum(col3) from X group by col1, col2; **Pig**: D = group A by (col1, col2); E = foreach D generate flatten(group), SUM(A.col3);
Having	Filter	**SQL**: select col1, sum(col2) from X group by col1 having sum(col2) > 5; **Pig**: F = filter E by $1 > '5';
Order By	Order … By	**SQL**: select col1, sum(col2) from X group by col1 order by col1; **Pig**: H = ORDER E by $0;

(adapted from Viraj's slide)

SQL vs. Pig Latin

SQL	Pig	Example
Distinct	Distinct	**SQL:** select distinct col1 from X; **Pig:** I = foreach A generate col1; J = distinct I;
Distinct Agg	Distinct in foreach	**SQL:** select col1, count (distinct col2) from X group by col1; **Pig:** K = foreach D { L = distinct A.col2; generate flatten(group), SUM(L); }

(adapted from Viraj's slide)

SQL vs. Pig Latin

SQL	Pig	Example
Join	**Cogroup + flatten** **(also shortcut: JOIN)**	**SQL:** select A.col1, B.col3 from A join B using (col1); **Pig:** A = load 'data1' using PigStorage('\t') as (col1, col2); B = load 'data2' using PigStorage('\t') as (col1, col3); C = cogroup A by col1 **inner**, B by col1 **inner**; D = foreach C generate flatten(A), flatten(B); E = foreach D generate A.col1, B.col3;

(adapted from Viraj's slide)

Debug Tips

- Use small data and `pig -x local`
- `LIMIT`
  - `A = LOAD 'data' AS (a1,a2,a3)`
  - `B = LIMIT A 3;`
- `DUMP , DESCRIBE`

FAQ

- http://wiki.apache.org/pig/FAQ
  - can assign #reducer
  - support regex
  - can use allocated HOD cluster

pig.vim

- http://www.vim.org/scripts/script.php?script_id=2186

```
A = load 'data.txt' as (f1,f2,f3);
dump A;
B = foreach A generate f1,f3;
dump B;
store B into 'output.txt' using PigStorage('-');
```