

# 计算机体系结构

---

周学海

[xhzhou@ustc.edu.cn](mailto:xhzhou@ustc.edu.cn)

0551-63601556, 63492271

中国科学技术大学



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

## 6.1 引言

---

- ❖ 仓库级计算机（WSC）是许多人每日所用因特网服务的基础，这些服务包括：搜索、社交网络、在线地图、视频共享、网上购物、电子邮件服务，等等。
- ❖ 此类因特网服务深受大众喜爱，从而有了创建WSC的必要，以满足公众迅速增长的需求。
- ❖ 尽管WSC看起来可能就像一些大型数据中心，但是他们的体系结构和操作有很大的不同。



# WSC架构师的目标

---

- ❖ 成本-性能
- ❖ 能耗效率
- ❖ 通过冗余提高可靠性
- ❖ 网络 I/O
- ❖ 交互式与批处理工作负载
- ❖ 充足的并行
- ❖ 运行成本计算
- ❖ 规模、与规模相关的机会/问题



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

Approx. number events in 1st year	Cause	Consequence
1 or 2	Power utility failures	Lose power to whole WSC; doesn't bring down WSC if UPS and generators work (generators work about 99% of time).
4	Cluster upgrades	Planned outage to upgrade infrastructure, many times for evolving networking needs such as recabling, to switch firmware upgrades, and so on. There are about 9 planned cluster outages for every unplanned outage.
1000s	Hard-drive failures	2% to 10% annual disk failure rate [Pinheiro 2007]
	Slow disks	Still operate, but run 10x to 20x more slowly
	Bad memories	One uncorrectable DRAM error per year [Schroeder et al. 2009]
	Misconfigured machines	Configuration led to ~30% of service disruptions [Barroso and Hölzle 2009]
	Flaky machines	1% of servers reboot more than once a week [Barroso and Hölzle 2009]
5000	Individual server crashes	Machine reboot, usually takes about 5 minutes

**Figure 6.1** List of outages and anomalies with the approximate frequencies of occurrences in the first year of a new cluster of 2400 servers. We label what Google calls a cluster an *array*; see Figure 6.5. (Based on Barroso [2010].)

一个由**2400**台服务器组成的新集群在第一年发生停用与异常，并给出了近似的  
发生频率



**嵌入式系统实验室**

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# 例题

- ❖ 一个服务运行在表格6-1中的2400台服务器上，试计算该服务的可用性。
- ❖ 本例题中的服务于实际WSC中的服务不同，它不能容忍硬件或软件故障。假定重启软件的时间为5分钟，修改硬件的时间为1小时。
- ❖ 解答：

$$\begin{aligned}\text{Hours Outage}_{\text{service}} &= (4 + 250 + 250 + 250) \times 1 \text{ hour} + (250 + 5000) \times 5 \text{ minutes} \\ &= 754 + 438 = 1192 \text{ hours}\end{aligned}$$

Since there are  $365 \times 24$  or 8760 hours in a year, availability is:

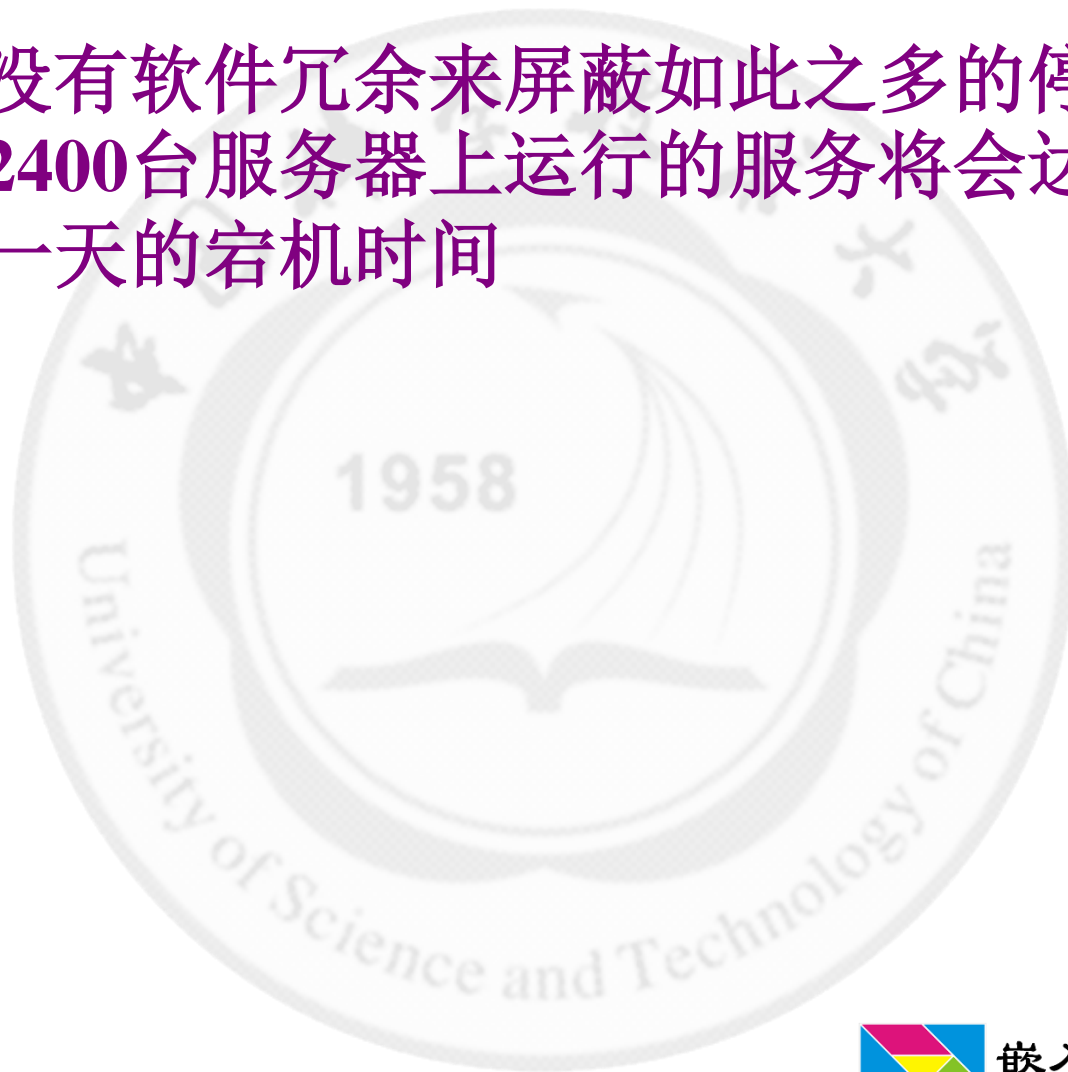
$$\text{Availability}_{\text{system}} = \frac{(8760 - 1192)}{8760} = \frac{7568}{8760} = 86\%$$



# 例题

---

- ❖ 如果没有软件冗余来屏蔽如此之多的停用次数，在这2400台服务器上运行的服务将会达到平均每周一天的宕机时间



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# WSC与HPC

- ❖ HPC处理器和节点之间的网络通常要比WSC快得多，HPC应用程序独立性更强，通信更频繁。
- ❖ HPC更倾向于使用定制硬件
- ❖ WSC可以通过使用大众化商用芯片来降低成本
- ❖ HPC强调线程级并行或数据级并行，通常强调完成单项任务的延迟
- ❖ WSC通过请求级并行完成许多独立任务
- ❖ HPC集群往往拥有长时间运行作业，会使服务器满负荷运行，甚至能持续数周以上
- ❖ WSC服务器的利用率通常在10%~50%之间，而且每天都会发生变化





# WSC与传统数据中心

- ❖ 传统数据中心往往拥有各种不同的硬件和软件
- ❖ WSC的硬件一致性要强得多
- ❖ 传统数据中心关注点通常是将许多服务整合到较少的机器中，这些机器互相隔离，以保护敏感信息
- ❖ WSC为一家组织中的不同用户提供服务，WSC的目标是让仓库中的硬件/软件看起来像是只有一台计算机



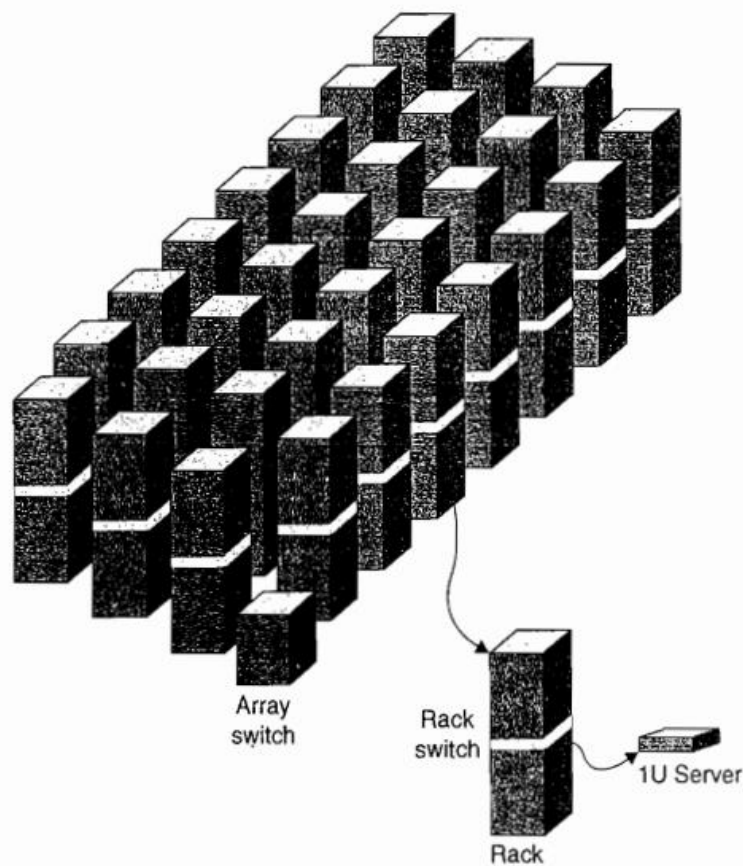


## 6.2 仓库级计算机的计算机体系结构

- ❖ 网络是将50000台服务器连接在一起的结缔组织。
- ❖ 机架内的速度对于每个服务器都是一样的。
- ❖ 离开机架的速度是机架内速度的1/6到1/24(8/48到2/48)。这一比值称为超额认购率(oversubscription)



# 仓库级计算机的计算机体系结构



**Figure 6.5** Hierarchy of switches in a WSC. (Based on Figure 1.2 of Barroso and Hölzle [2009].)



# 存储

---

- ❖ 将磁盘包含在服务器中，通过以太网连接访问远程服务器磁盘上的信息
- ❖ 使用网络连接存储（NAS），可能是通过类似于Infiniband的存储网络
- ❖ WSC通常会依靠本地磁盘，并提供用于处理连接性和可靠性的存储软件



# WSC存储层次结构

## WSC内部存储器层次结构的延迟、带宽和容量

	Local	Rack	Array
DRAM latency (microseconds)	0.1	100	300
Disk latency (microseconds)	10,000	11,000	12,000
DRAM bandwidth (MB/sec)	20,000	100	10
Disk bandwidth (MB/sec)	200	100	10
DRAM capacity (GB)	16	1040	31,200
Disk capacity (GB)	2000	160,000	4,800,000

**Figure 6.6** Latency, bandwidth, and capacity of the memory hierarchy of a WSC [Barroso and Hölzle 2009]. Figure 6.7 plots this same information.



# WSC存储层次结构

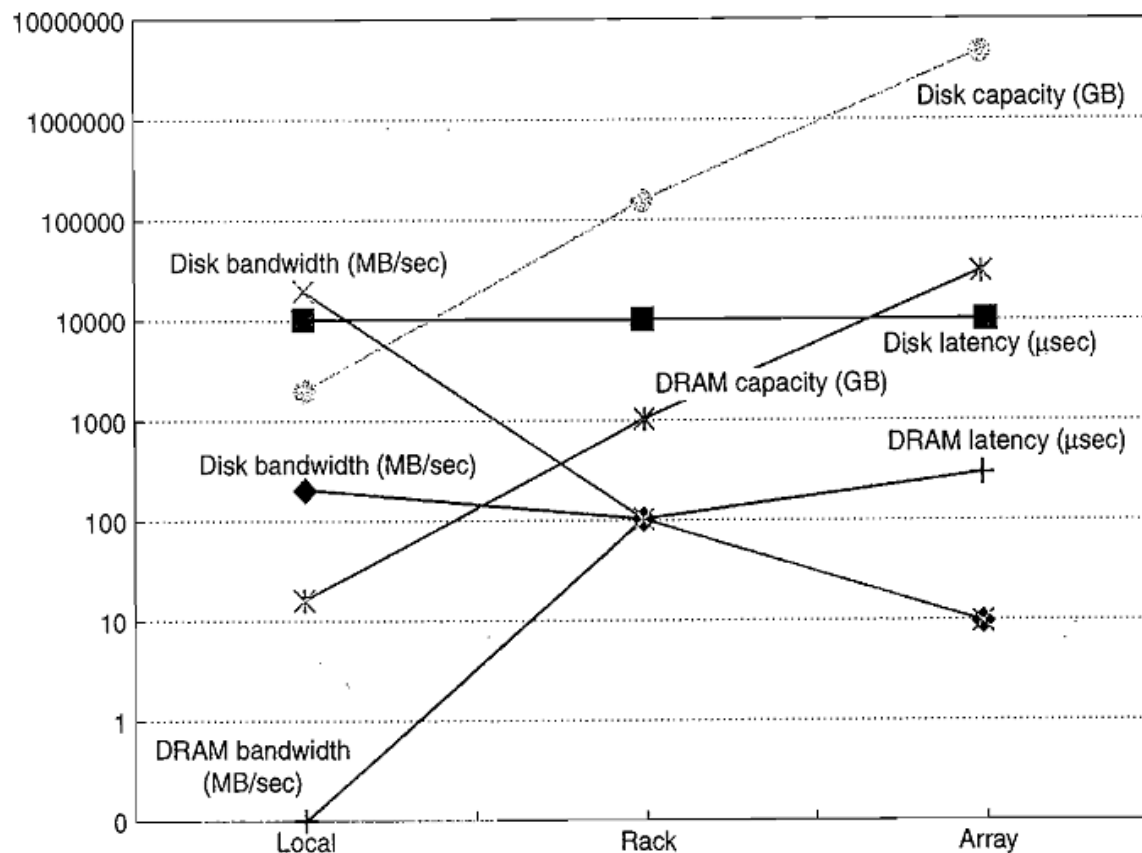


Figure 6.7 Graph of latency, bandwidth, and capacity of the memory hierarchy of a WSC for data in Figure 6.6 [Barroso and Hölzle 2009].



# 例题

	Local	Rack	Array
DRAM latency (microseconds)	0.1	100	300
Disk latency (microseconds)	10,000	11,000	12,000
DRAM bandwidth (MB/sec)	20,000	100	10
Disk bandwidth (MB/sec)	200	100	10
DRAM capacity (GB)	16	1040	31,200
Disk capacity (GB)	2000	160,000	4,800,000

**Figure 6.6** Latency, bandwidth, and capacity of the memory hierarchy of a WSC [Barroso and Hölzle 2009]. Figure 6.7 plots this same information.

假定90%的访问服务器是本地访问，9%的访问超出服务器但在机架范围内，1%的访问超出机架但在阵列范围内，则，平均存储器延迟为多少？

**Answer** The average memory access time is

$$(90\% \times 0.1) + (9\% \times 100) + (1\% \times 300) = 0.09 + 9 + 3 = 12.09 \text{ microseconds}$$



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# 例题

	Local	Rack	Array
DRAM latency (microseconds)	0.1	100	300
Disk latency (microseconds)	10,000	11,000	12,000
DRAM bandwidth (MB/sec)	20,000	100	10
Disk bandwidth (MB/sec)	200	100	10
DRAM capacity (GB)	16	1040	31,200
Disk capacity (GB)	2000	160,000	4,800,000

**Figure 6.6** Latency, bandwidth, and capacity of the memory hierarchy of a WSC [Barroso and Hölzle 2009]. Figure 6.7 plots this same information.

在服务器内部的磁盘之间、在机架的服务器之间、在阵列中不同的机架内的服务器之间，传递**1000MB**需要多少时间？在这三种情况下，在**DRAM**之间传递**1000MB**可以加快多少时间？

**Answer** A 1000 MB transfer between disks takes:

Within server =  $1000/200 = 5$  seconds

Within rack =  $1000/100 = 10$  seconds

Within array =  $1000/10 = 100$  seconds

A memory-to-memory block transfer takes

Within server =  $1000/20000 = 0.05$  seconds

Within rack =  $1000/100 = 10$  seconds

Within array =  $1000/10 = 100$  seconds

式系统实验室



EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC



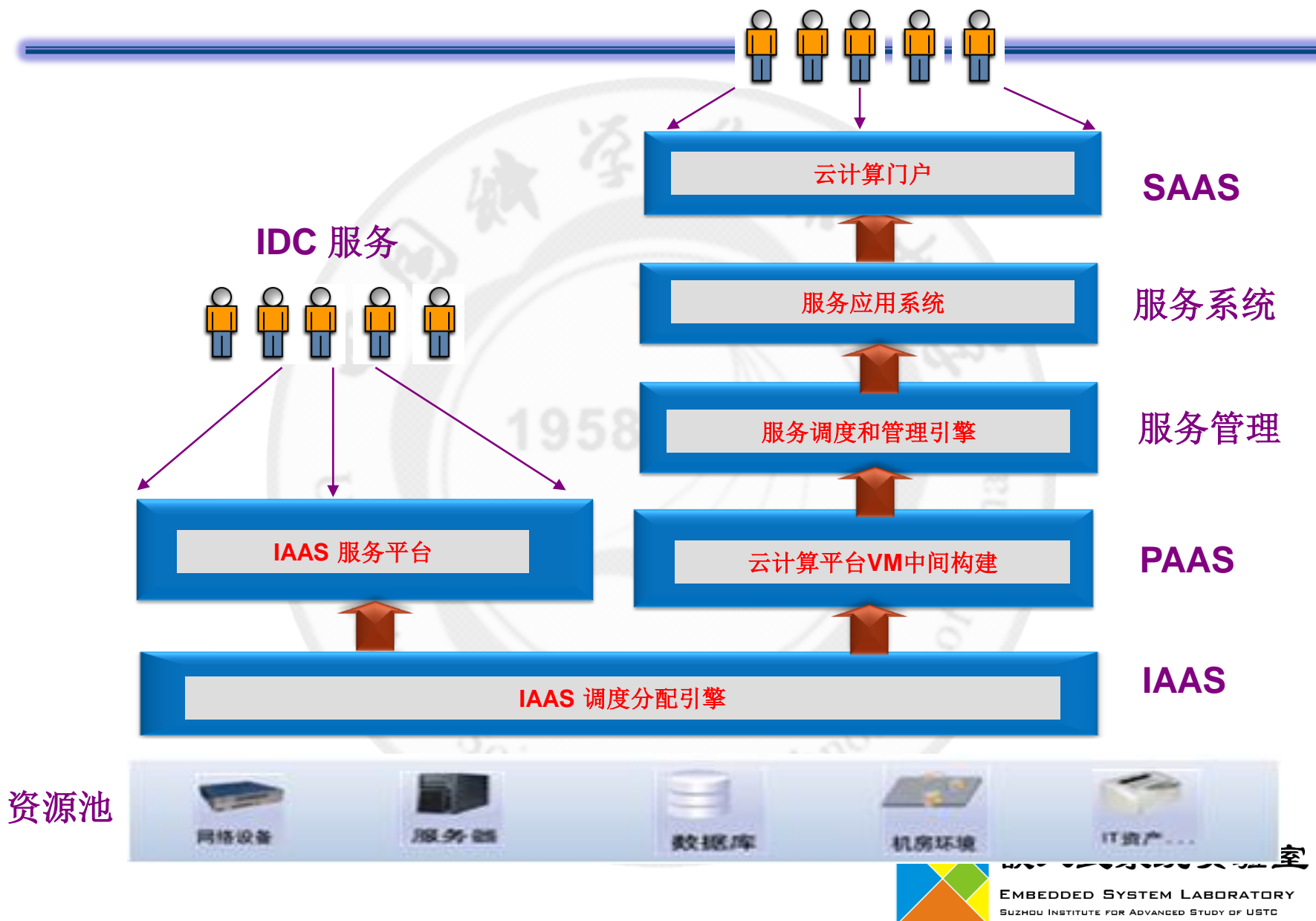
## 6.3 云计算概念

---

- ❖ Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet.
- ❖ 云计算是一种服务模式



# 云计算体系架构



# 云计算

## SaaS

(Software as a Service, 软件即服务)

应用软件对最终用户按需分配

e.g., *Salesforce.com*

## PaaS

(Platform as a Service, 平台即服务)

APIs for developing new Cloud apps

e.g., *Google App Web*

## IaaS

(Infrastructure as a Service, 基础架构即服务)

服务器, 网络, 存储按需分配

e.g., *Amazon EC2, S3*



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

## ❖ IaaS(Infrastructure as a Service)基础设施即服务

- 以服务的形式提供虚拟硬件资源，如虚拟主机、存储、网络、数据库管理等资源；
- 用户无需购买服务器、网络设备、存储设备，只需通过互联网租赁即可搭建自己的应用系统；
- 典型案例：Amazon Web Service (AWS)



# PaaS

---

## ❖ PaaS (Platform-as-a-Service: 平台即服务)

- 提供应用服务引擎，如互联网应用编程接口、运行平台等，用户基于该应用服务引擎，可以构建该类应用；
- 典型案例：Google App Engine、Force.com和微软的Azure服务平台；



# SaaS

---

## ❖ SaaS (Software-as-a-service: 软件即服务)

- 用户通过Internet来使用软件，不必购买软件，只需按需租用软件服务即可；
- 典型案例：Google Docs、Salesforce CRM、Oracle CRM On Demand和Office Live Workspace；



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# 公共云

## User

- 上网机
- Internet 接入上网



## User

- 手机
- 移动上网 – 3G

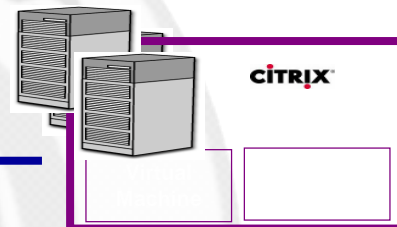


## User

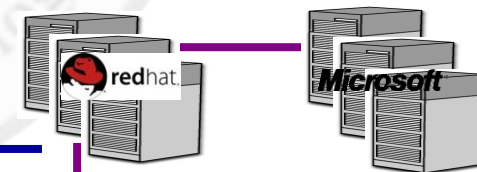
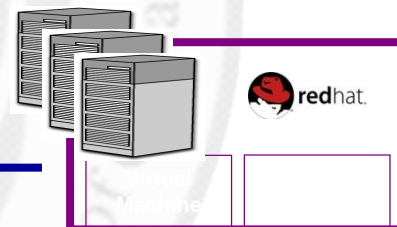
- Workstation
- VIP 接入



PC cloud



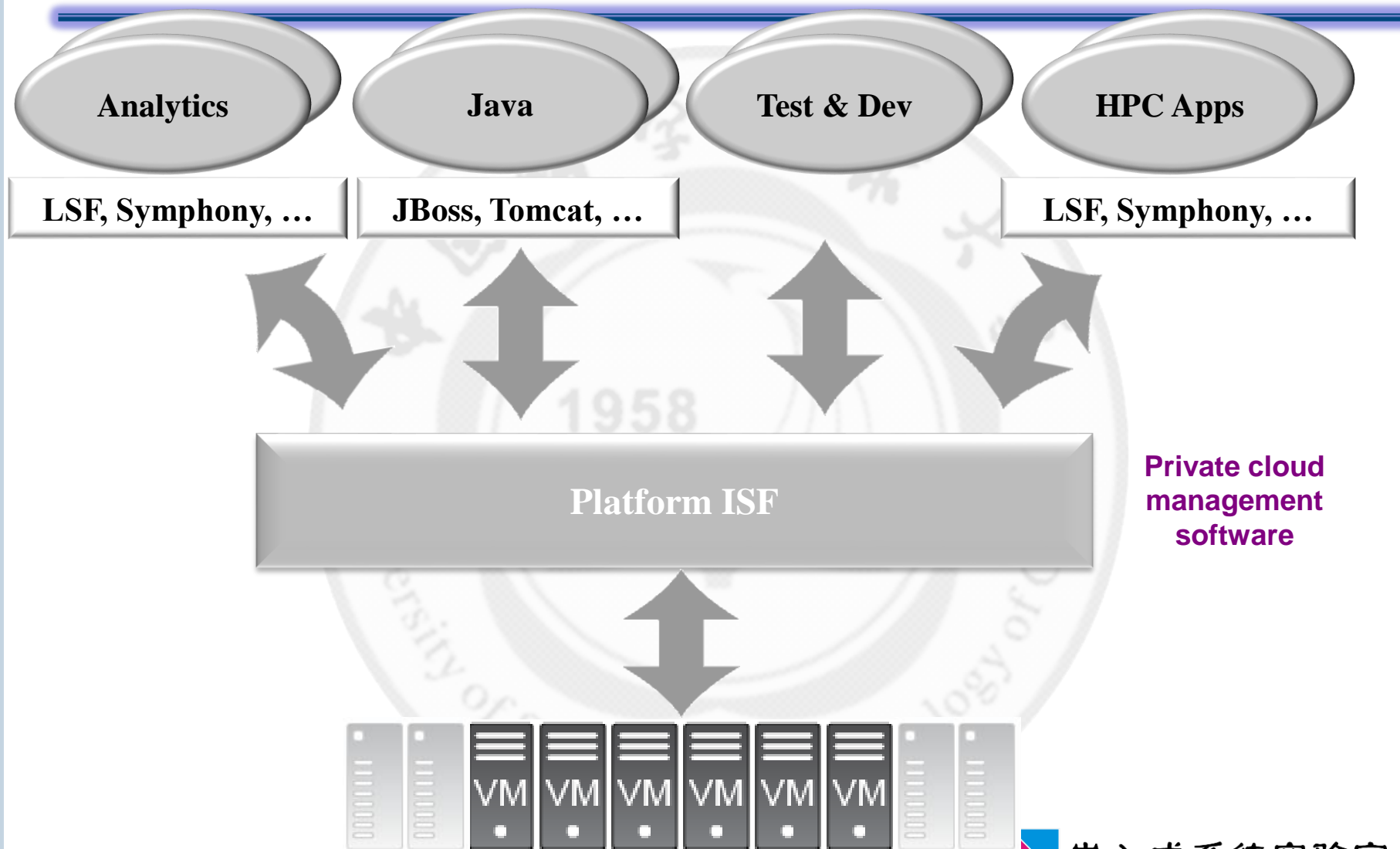
虚拟机



物理机



# 私有云



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# 云计算关键技术

---

- ❖ 虚拟化技术
- ❖ 分布式存储技术
- ❖ 并行编程技术
- ❖ 数据管理技术
- ❖ 分布式资源管理技术



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# 虚拟化技术

- ❖ 虚拟化技术是一种调配计算资源的方法，它将应用系统的不同层面——硬件、软件、数据、网络、存储等一一隔离开来，从而打破数据中心、服务器、存储、网络、数据和应用中的物理设备之间的划分，实现架构动态化，并达到集中管理和动态使用物力资源的目的；
- ❖ 随着IT环境逐步向云计算时代跨越，虚拟化技术从最初的侧重于整合数据中心内的资源，发展到跨越IT架构实现包括资源、网络、应用和桌面在内的全系统虚拟化。



# 虚拟化技术分类

## ❖ 按照与物理硬件及操作系统的耦合程度分类

- 全虚拟化
- 半虚拟化
- 操作系统级的虚拟化
- CPU虚拟化

## ❖ 按系统层级划分的虚拟化技术

- 服务器虚拟化
- 存储虚拟化
- 网络虚拟化
- 应用虚拟化
- 客户端虚拟化



- ❖ 在云计算环境下，所有虚拟化解决方案都是集服务器、存储系统、网络设备、软件及服务于一体的系统整合方案；
- ❖ 它们包括了硬件资源虚拟化、网络设施虚拟化、应用虚拟化、桌面虚拟化等技术，并根据不同的应用环境灵活地将若干层面组合以实现不同模式虚拟化方案。



# 海量分布式存储技术

- ❖ 云计算的数据存储系统主要有Google GFS和Hadoop团队开发的HDFS;
- ❖ GFS的设计受到Google应用负载和技术环境的影响，主要体现在四个方面：
  - 集群中的节点失效是一种常态
  - Google系统中文件大小与通常文件系统中文件大小概念不同，文件大小通常以G字节计
  - Google文件系统中的文件读写模式和传统的文件系统不同
  - 文件系统的某些具体操作不再透明，而且需要应用程序的协助完成



# 并行编程模式

- ❖ 为高效利用资源，云计算采用**MapReduce**编程模式，将任务自动分成多个子任务，通过**Map**和**Reduce**两步实现任务在大规模计算节点中的调度和分配，保证了后台复杂的并行执行和任务调度的透明性；
- ❖ **MapReduce**系统主要由三个模块组成
  - 客户端用于将用户撰写的并行处理作业提交至**Master**节点
  - 主节点（**Master**）自动将用户作业分解为**Map**任务和**Reduce**任务，并将任务调度到工作节点（**Worker**）
  - 工作节点向**master**请求执行任务，同时多个工作节点组成的分布式文件系统用于存储输入输出数据





# 数据管理技术

---

- ❖ 云计算系统对大数据集进行处理、分析，向用户提供高效的服务，因此，数据管理技术必须能够高效的管理大数据集；
- ❖ 云系统的数据管理往往采用列存储的数据管理模式，保证海量数据存储和分析性能；
- ❖ 云计算数据管理技术代表为：Google的BigTable数据管理技术，以及Hadoop团队开发的开源数据管理模块HBase



# 分布式资源管理技术

- ❖ 在多节点并发执行环境中，分布式资源管理系统是保证系统状态正确性的关键技术；
- ❖ Google的Chubby是最著名的分布式资源管理系统；
- ❖ Chubby是一个lock service，通过它可以解决分布式中的一致性问题，而lock service的实现则是基于一个分布式的文件系统



## 6.4 Google的云计算平台

- ❖ Google的云计算技术实际上是针对Google特定的网络应用程序而定制的。针对内部网络数据规模超大的特点，Google提出了一整套基于分布式并行集群方式的基础架构，利用软件的能力来处理集群中经常发生的节点失效问题。
- ❖ Google使用的云计算基础架构模式包括三个相互独立又紧密结合在一起的系统。包括Google建立在集群之上的文件系统Google File System，针对Google应用程序的特点提出的Map/Reduce编程模式，以及Google开发的模型简化的大规模分布式数据库BigTable。



## 6.4.1 Google File System 文件系统

---

- ❖ 1. 集群中的节点失效是一种常态，而不是一种异常
- ❖ 2. 文件大小通常以G字节计。
- ❖ 3. 不是覆盖原有数据，而是在文件尾追加新数据。
- ❖ 4. 文件系统的某些具体操作不再透明，而且需要应用程序的协助完成，应用程序和文件系统API的协同设计提高了整个系统的灵活性。



# Google File System 文件系统

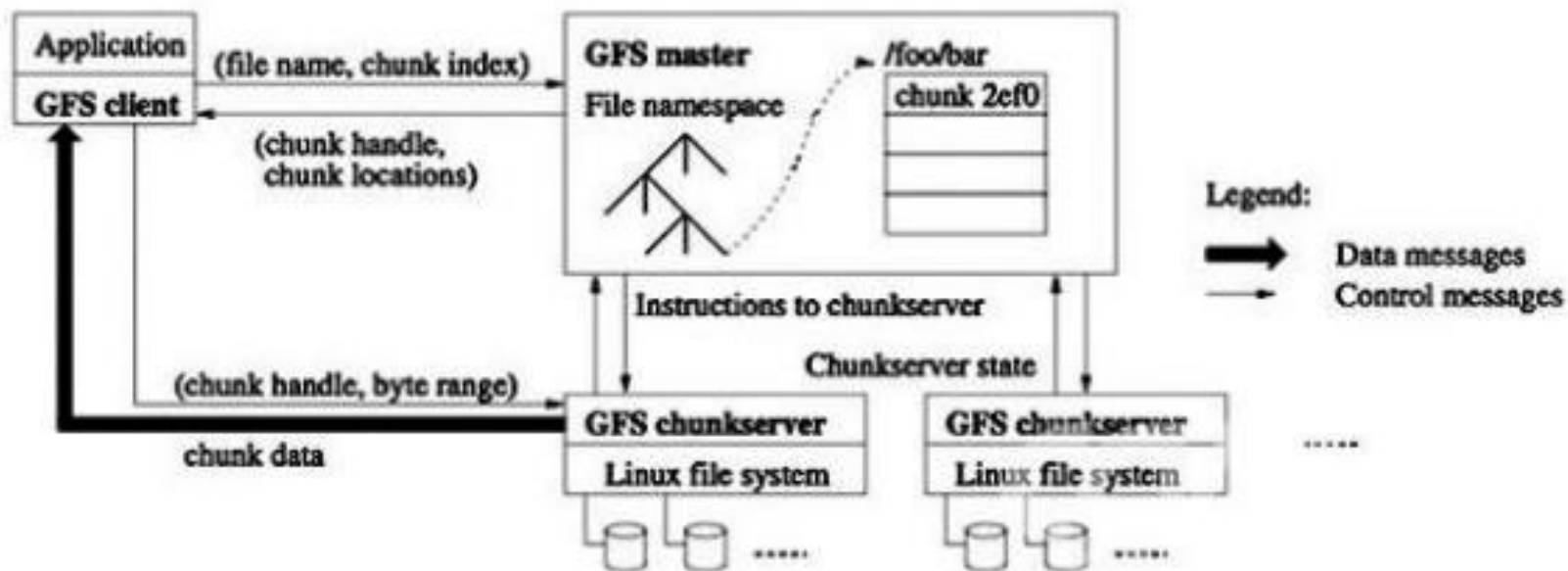


图1 Goolge File System的系统架构



# Google File System 文件系统

- ❖ 一个GFS集群包含一个主服务器和多个块服务器，被多个客户端访问。
- ❖ 文件被分割成固定尺寸的块。在每个块创建的时候，服务器分配给它一个不变的、全球惟一的64位块句柄对它进行标识。
- ❖ 块服务器把块作为linux文件保存在本地硬盘上，并根据指定的块句柄和字节范围来读写块数据。
- ❖ 为了保证可靠性，每个块都会复制到多个块服务器上，缺省保存三个备份。
- ❖ 主服务器管理文件系统所有的元数据，包括名字空间、访问控制信息和文件到块的映射信息，以及块当前所在的位置。
- ❖ GFS客户端代码被嵌入到每个程序里，它实现了Google文件系统API，帮助应用程序与主服务器和块服务器通信，对数据进行读写。
- ❖ 客户端跟主服务器交互进行元数据操作，但是所有的数据操作的通信都是直接和块服务器进行的。





## 6.4.2 MapReduce分布式编程环境

- ❖ 为了让内部非分布式系统方向背景的员工能够有机会将应用程序建立在大规模的集群基础之上，Google还设计并实现了一套大规模数据处理的编程规范Map/Reduce系统。这样，非分布式专业的程序编写人员也能够为大规模的集群编写应用程序而不用去顾虑集群的可靠性、可扩展性等问题。应用程序编写人员只需要将精力放在应用程序本身，而关于集群的处理问题则交由平台来处理。





# MapReduce分布式编程环境

- ❖ Map/Reduce通过“Map(映射)”和“Reduce(化简)”这样两个简单的概念来参加运算，用户只需要提供自己的Map函数以及Reduce函数就可以在集群上进行大规模的分布式数据处理。
- ❖ MapReduce是Google提出的一个软件架构，用于大规模数据集（大于1TB）的并行运算。当前的软件实现是指定一个Map（映射）函数，用来把一组键值对映射成一组新的键值对，指定并发的Reduce（化简）函数，用来保证所有映射的键值对中的每一个共享相同的键组。



# MapReduce分布式编程环境

- ❖ 简单说来，一个映射函数就是对一些独立元素组成的概念上的列表的每一个元素进行指定的操作。
- ❖ 事实上，每个元素都是被独立操作的，而原始列表没有被更改，因为这里创建了一个新的列表来保存新的答案。这就是说，Map操作是可以高度并行的，这对高性能要求的应用以及并行计算领域的需求非常有用。
- ❖ 而化简操作指的是对一个列表的元素进行适当的合并。虽然他不如映射函数那么并行，但是因为化简总是有一个简单的答案，大规模的运算相对独立，所以化简函数在高度并行环境下也很有用。



# MapReduce分布式编程环境

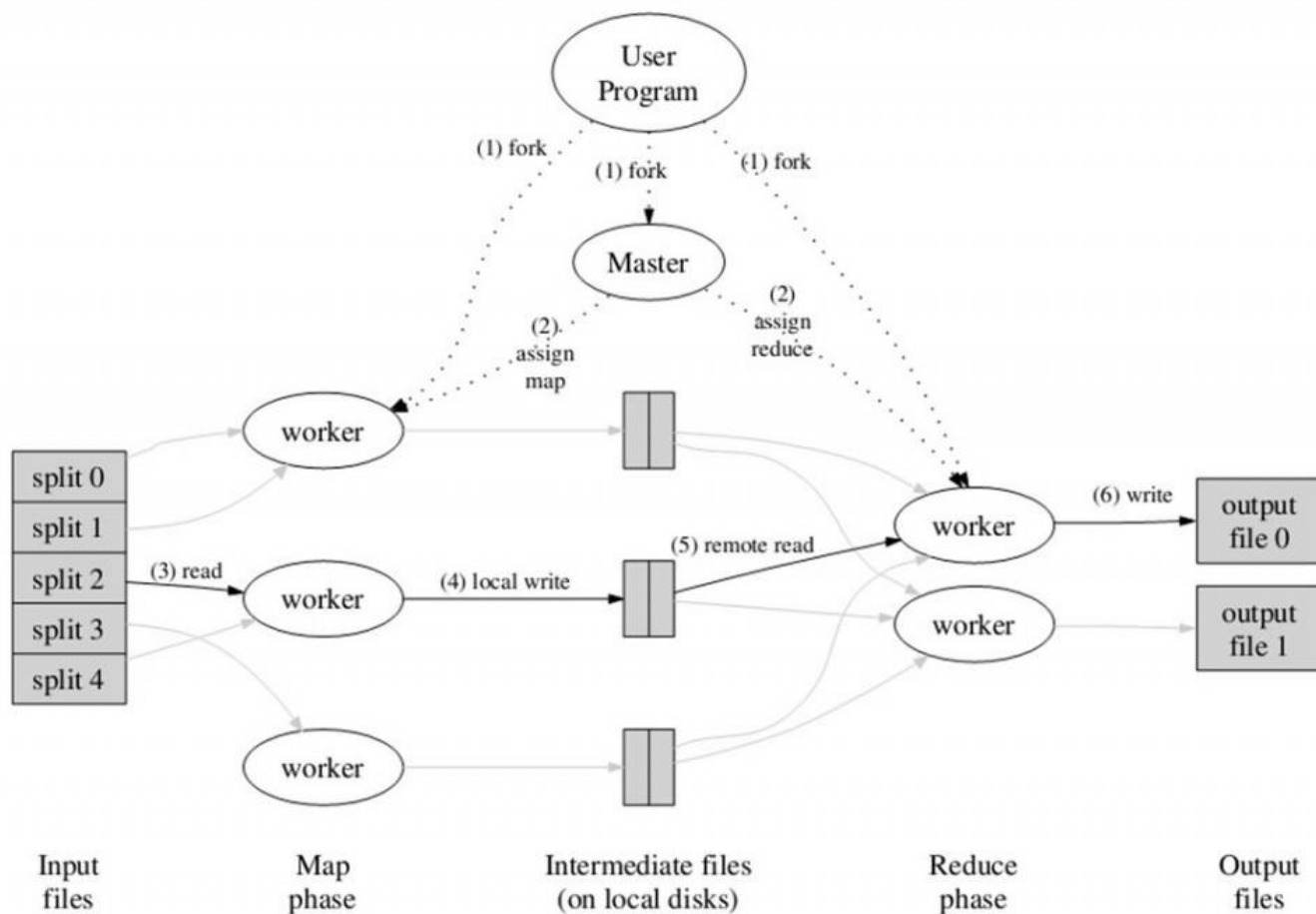


Figure 1: Execution overview

# MapReduce分布式编程环境

- ❖ MapReduce通过把对数据集的大规模操作分发给网络上的每个节点实现可靠性；
- ❖ 每个节点会周期性的把完成的工作和状态的更新报告回来。如果一个节点保持沉默超过一个预设的时间间隔，主节点（记录下这个节点状态为死亡，并把分配给这个节点的数据发到别的节点。
- ❖ 每个操作使用命名文件的原子操作以确保不会发生并行线程间的冲突；
- ❖ 当文件被改名的时候，系统可能会把他们复制到任务名以外的另一个名字上去。



# MapReduce分布式编程环境

---

- ❖ 化简操作工作方式很类似，但是由于化简操作在并行能力较差，主节点会尽量把化简操作调度在一个节点上，或者离需要操作的数据尽可能近的节点上。



## 6.4.3 分布式大规模数据库管理系统BigTable

- ❖ 很多应用程序对于数据的组织还是非常有规则的。一般来说，数据库对于处理格式化的数据还是非常方便的，但是由于关系数据库很强的一致性要求，很难将其扩展到很大的规模。
- ❖ 为了处理Google内部大量的格式化以及半格式化数据，Google构建了弱一致性要求的大规模数据库系统BigTable。





# 分布式大规模数据库管理系统BigTable

- ❖ 图2给出了在BigTable模型中的数据模型。
- ❖ 数据模型包括行列以及相应的时间戳，所有的数据都存放在表格中的单元里。
- ❖ BigTable的内容按照行来划分，将多个行组成一个小表，保存到某一个服务器节点中。这一个小表就被称为Tablet。

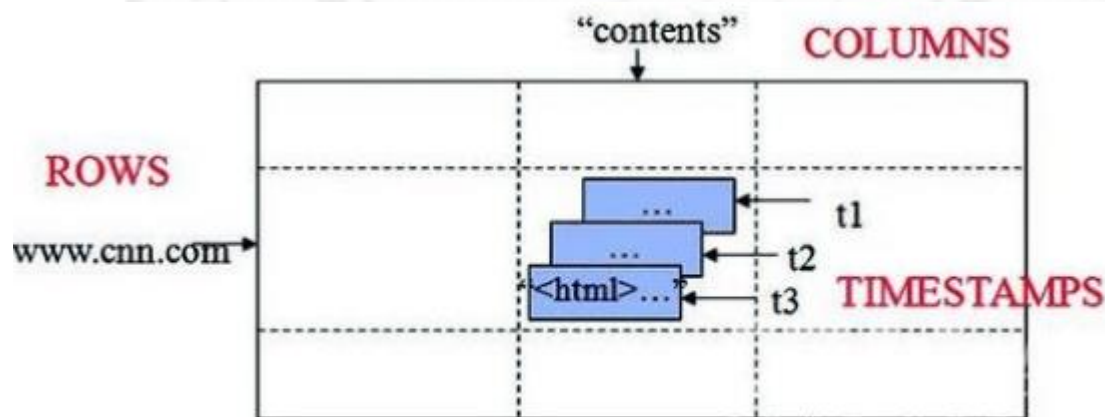
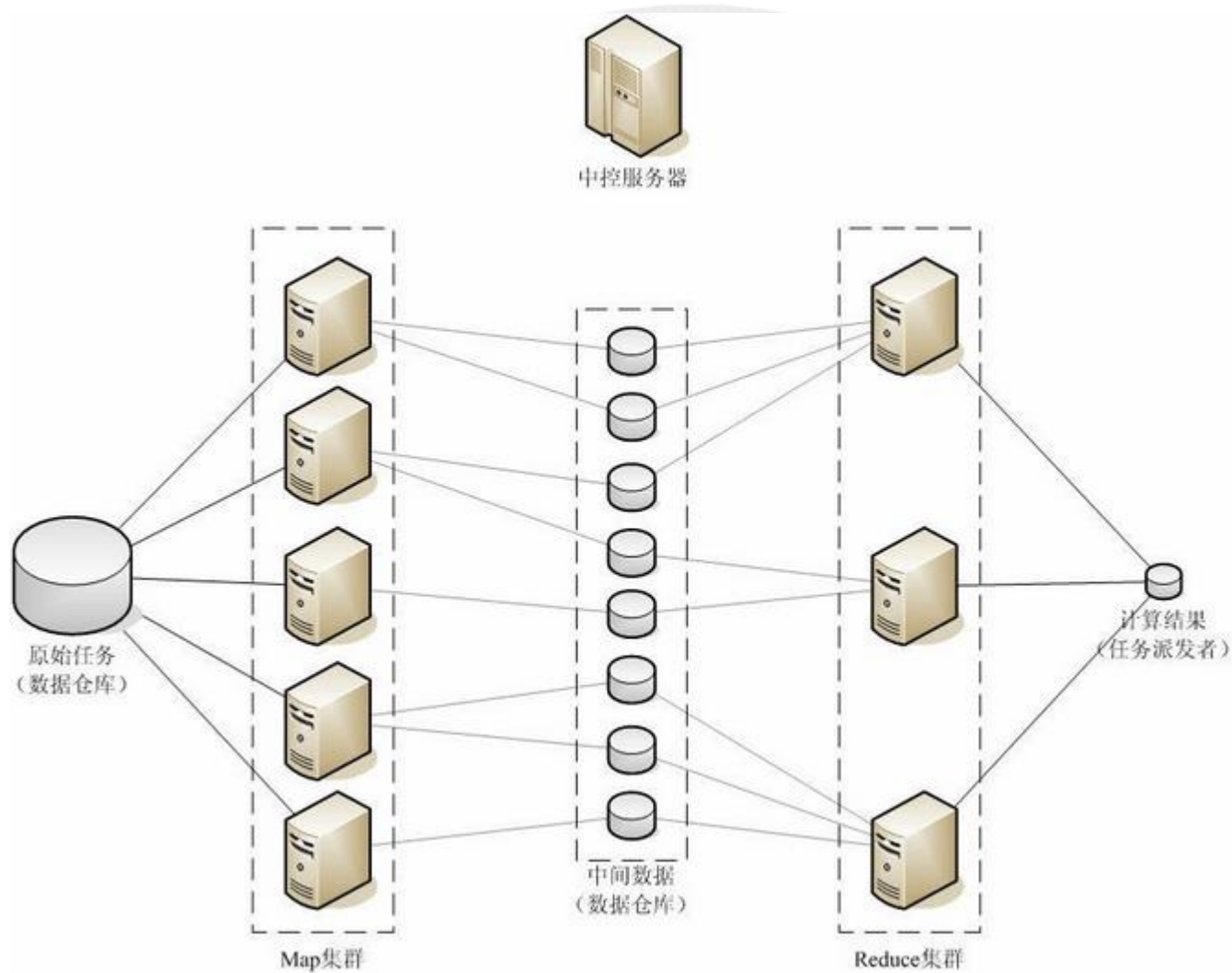


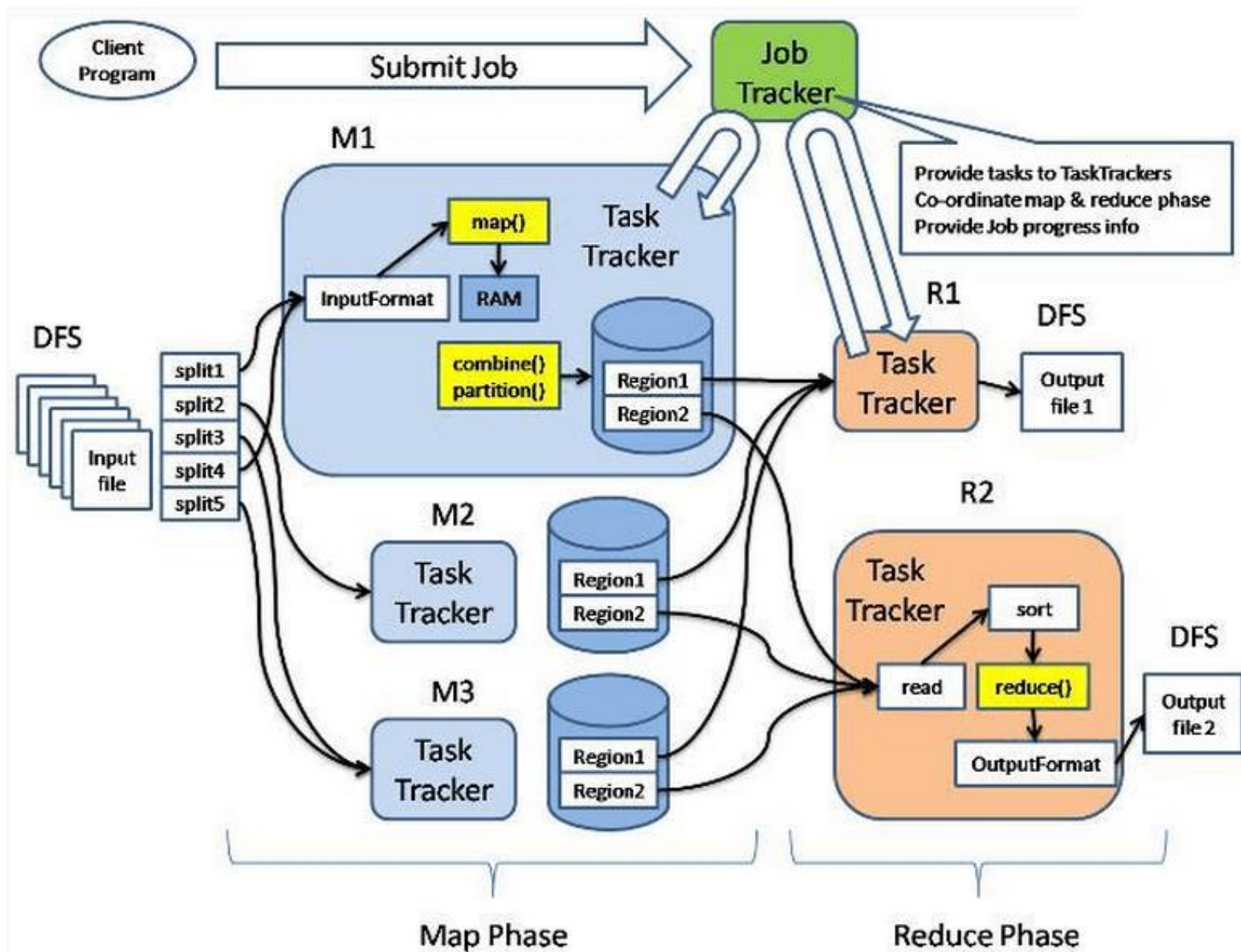
图2 Google BigTable的数据模型



## 6.4.4 MapReduce架构



# MapReduce执行流程



---



# MapReduce执行流程

- ❖ 1. MapReduce库先把user program的输入文件划分为M份（由输入文件的大小和用户定义的块大小决定）
- ❖ 2. master是负责调度的，为空闲worker分配作业（Map作业或者Reduce作业）
- ❖ 3. 被分配了Map作业的worker，开始读取对应分片的输入数据，Map作业数量和split一一对应；Map作业从输入数据中抽取出键值对，每一个键值对都作为参数传递给map函数，map函数产生的中间键值对被缓存在内存中。



# MapReduce执行流程

- ❖ 4. 缓存的中间键值对会被定期写入本地磁盘，而且被分为R个区，R的大小是由用户定义的，将来每个区会对应一个Reduce作业；这些中间键值对的位置会被通报给master，master负责将信息转发给Reduce worker。
- ❖ 5. master通知分配了Reduce作业的worker它负责的分区在什么位置，当Reduce worker把所有它负责的中间键值对都读过来后，先对它们进行排序，使得相同键的键值对聚集在一起。





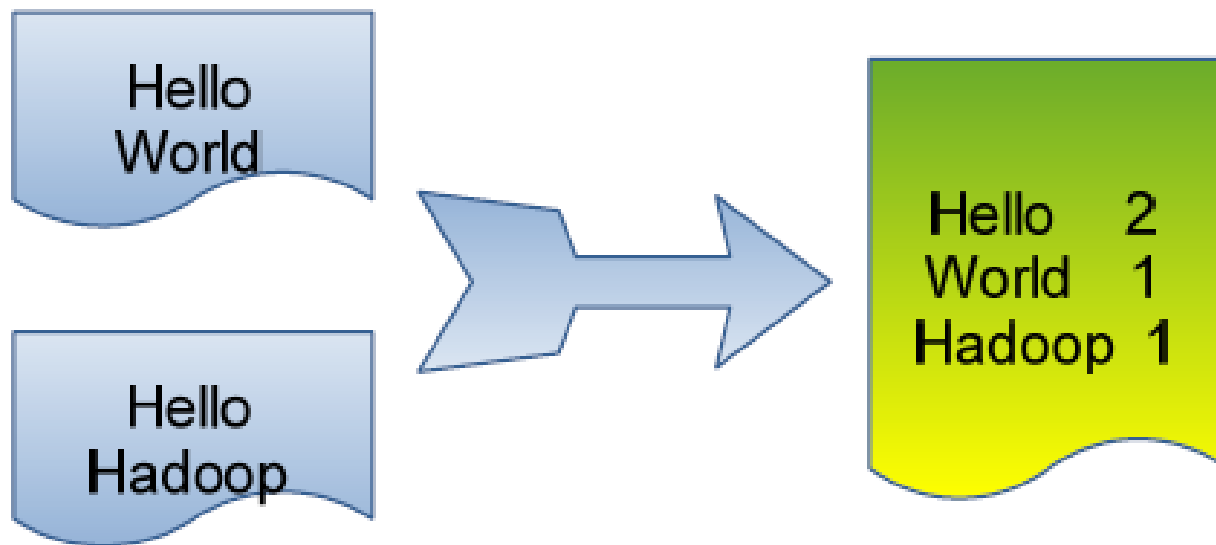
# MapReduce执行流程

- ❖ 6. reduce worker遍历排序后的中间键值对，对于每个唯一的键，都将键与关联的值传递给reduce函数，reduce函数产生的输出会添加到这个分区的输出文件中。
- ❖ 7. 当所有的Map和Reduce作业都完成了，master唤醒正版的user program，MapReduce函数调用返回user program的代码。



## 6.4.5 WordCount处理过程

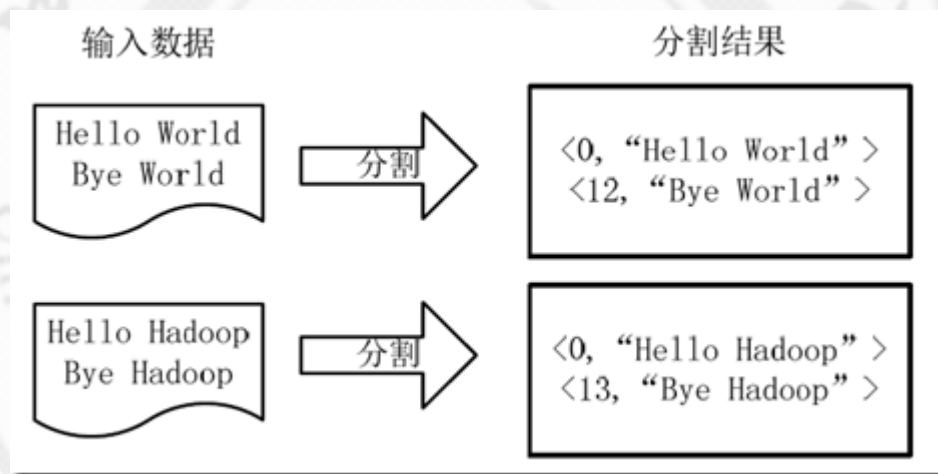
- ❖ 单词计数主要完成功能是：统计一系列文本文件中每个单词出现的次数，如下图所示。





# WordCount处理过程

- ❖ 1) 将文件拆分成splits, 由于测试用的文件较小, 所以每个文件为一个split, 并将文件按行分割形成<key,value>对



分割过程

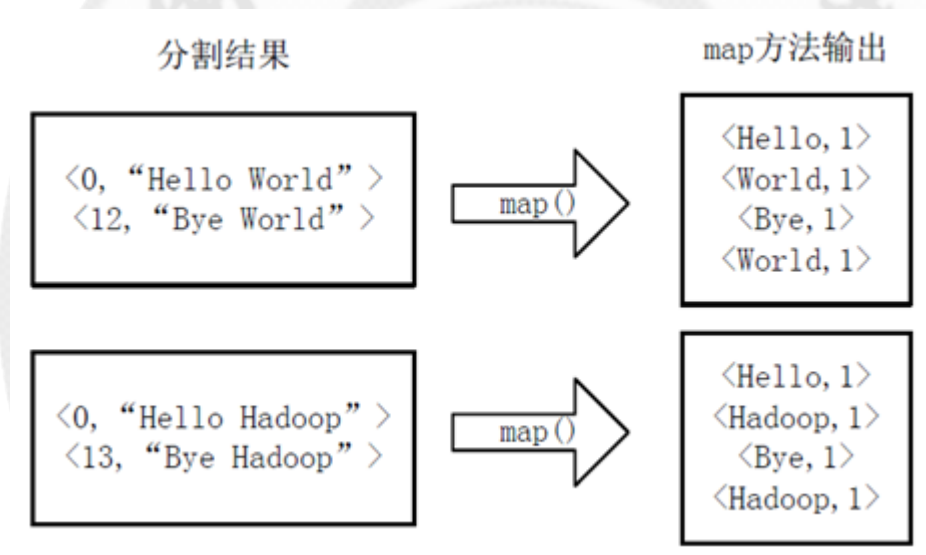


嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# WordCount处理过程

- ❖ 2) 将分割好的<key,value>对交给用户定义的map方法进行处理，生成新的<key,value>对



执行Map方法

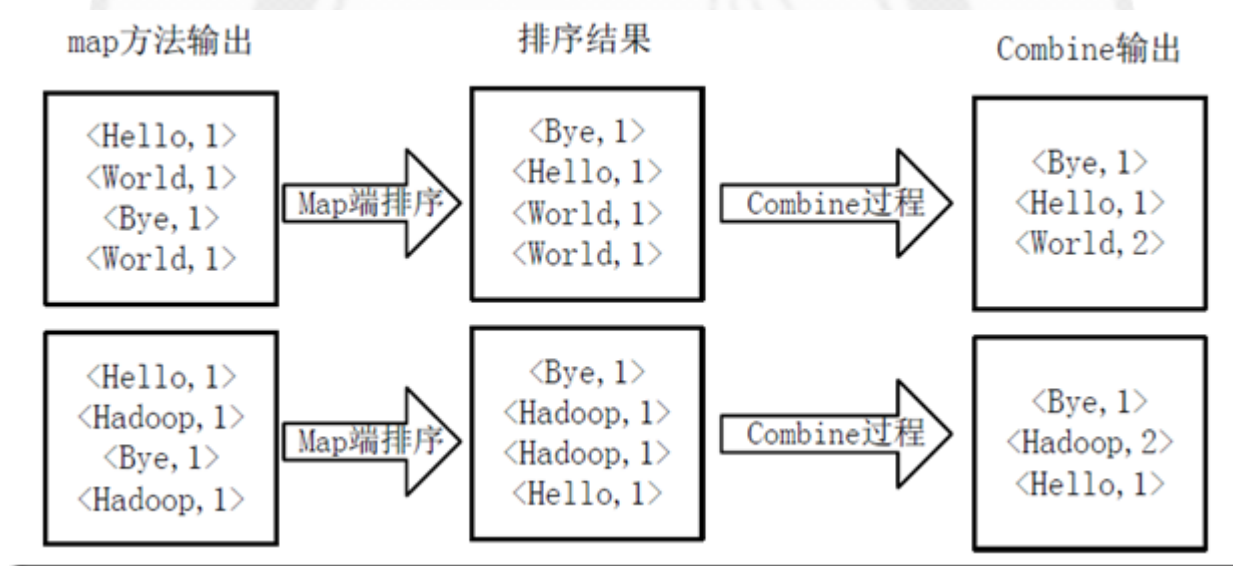


嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

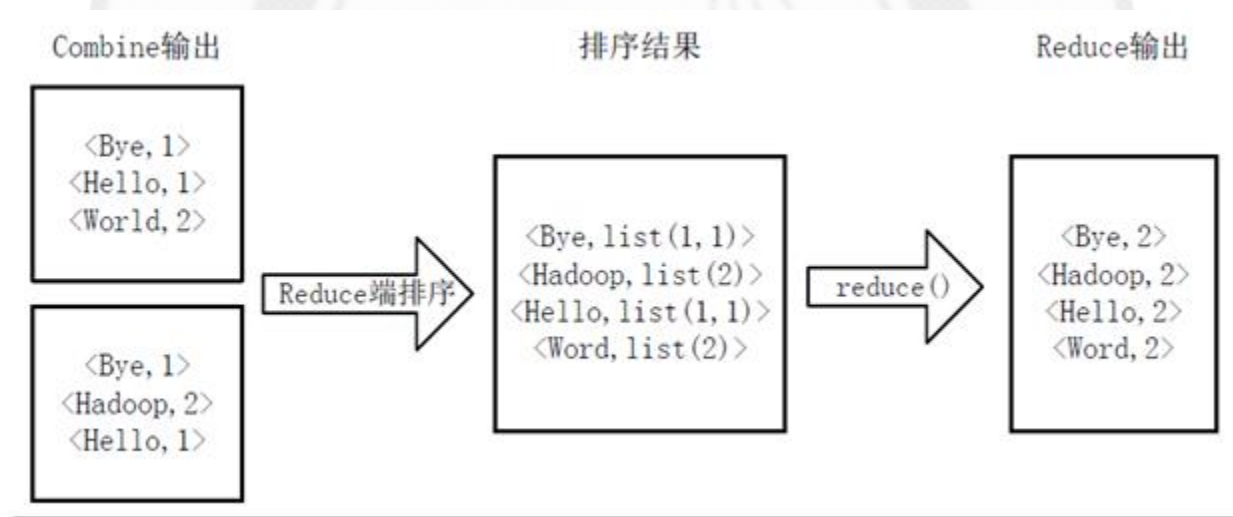
# WordCount处理过程

- ❖ 3) 得到map方法输出的<key,value>对后, Mapper会将它们按照key值进行排序, 并执行Combine过程, 将key至相同value值累加, 得到Mapper的最终输出结果。



# WordCount处理过程

- ❖ 4) Reducer先对从Mapper接收的数据进行排序，再交由用户自定义的reduce方法进行处理，得到新的<key,value>对，并作为WordCount的输出结果



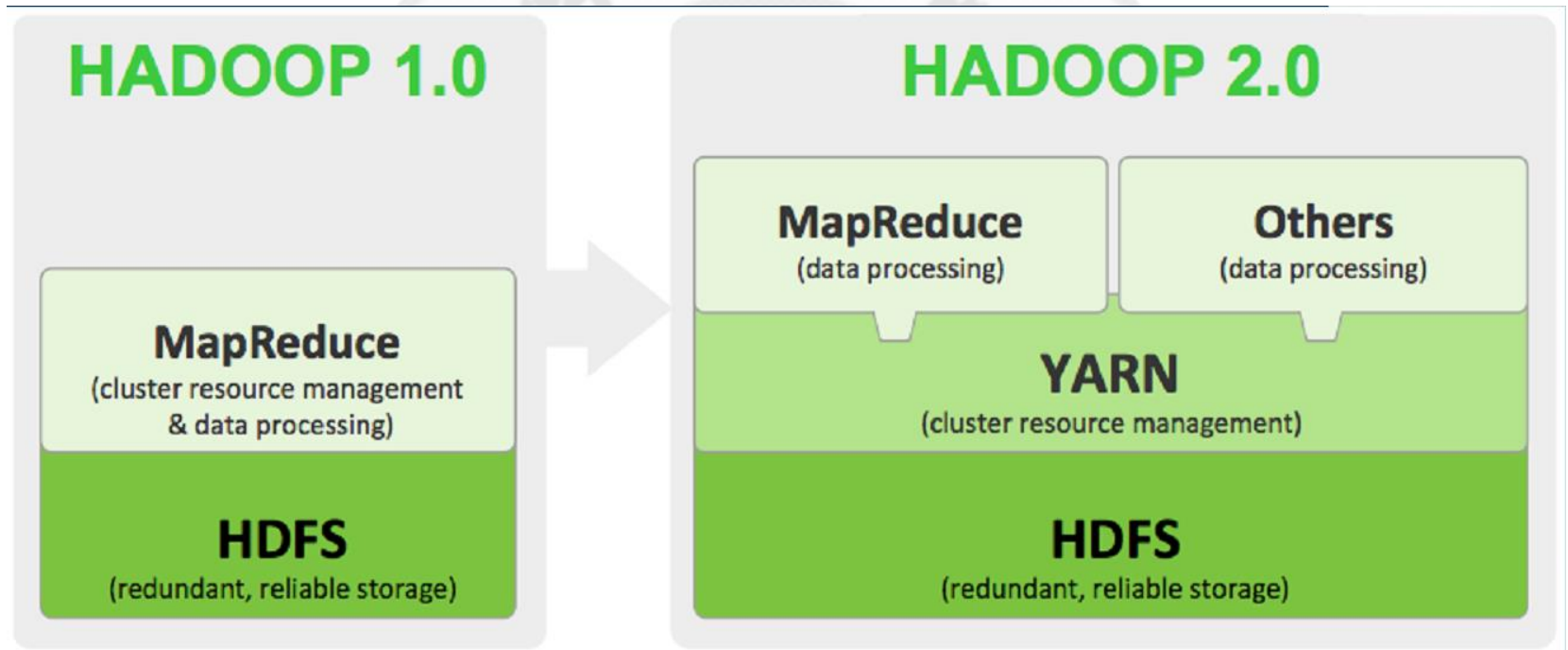
Reduce端排序及输出结果



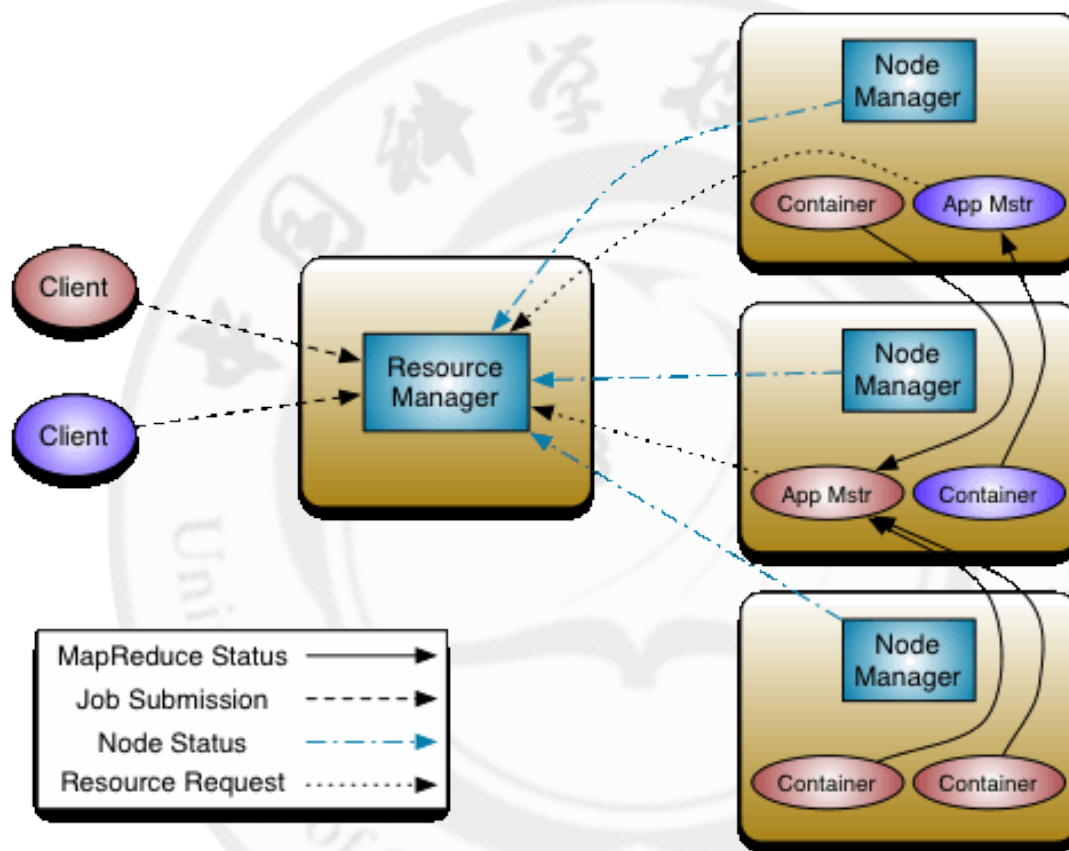
嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

## 6.5 Hadoop 2.0



# Hadoop新MapReduce框架YARN



# Hadoop YARN 产生背景

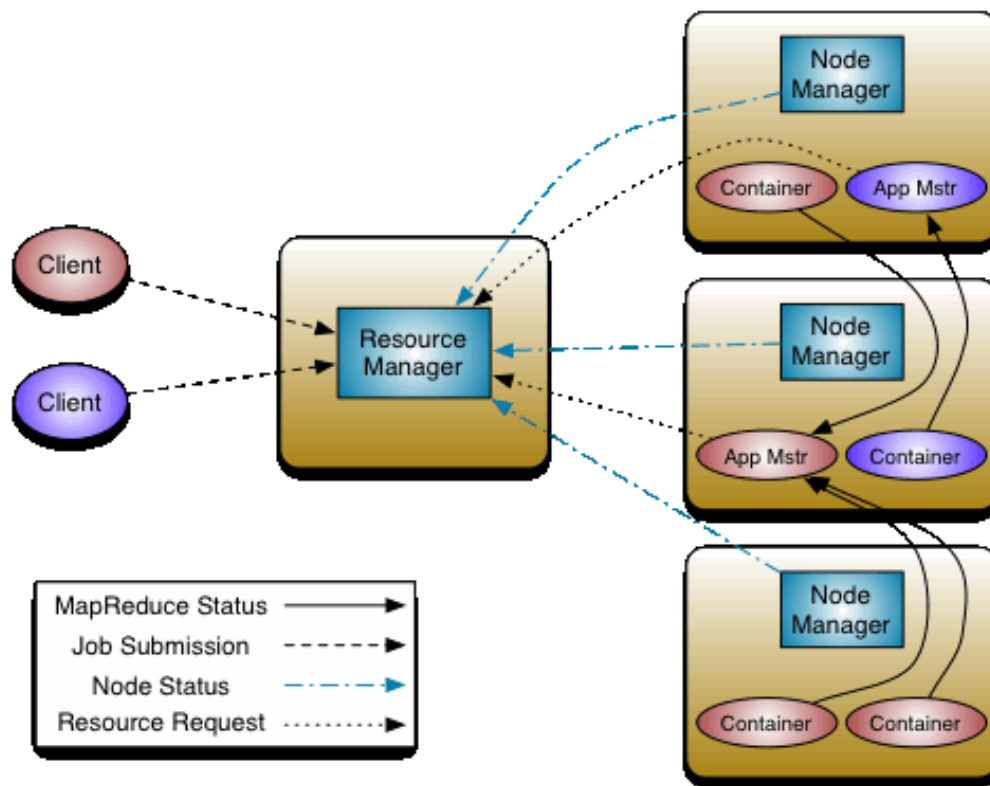
- ❖ 直接源于MRv1在几个方面的缺陷
  - 扩展性受限
  - 单点故障
  - 难以支持MR之外的计算
- ❖ 多计算框架各自为战，数据共享困难
  - MR：离线计算框架
  - Storm：实时计算框架
  - Spark：内存计算框架





# Hadoop YARN 基本架构

MRv2的基  
源管理和  
个全局的  
有一个Ap



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# Hadoop YARN 各模块组成

## ❖ ResourceManager

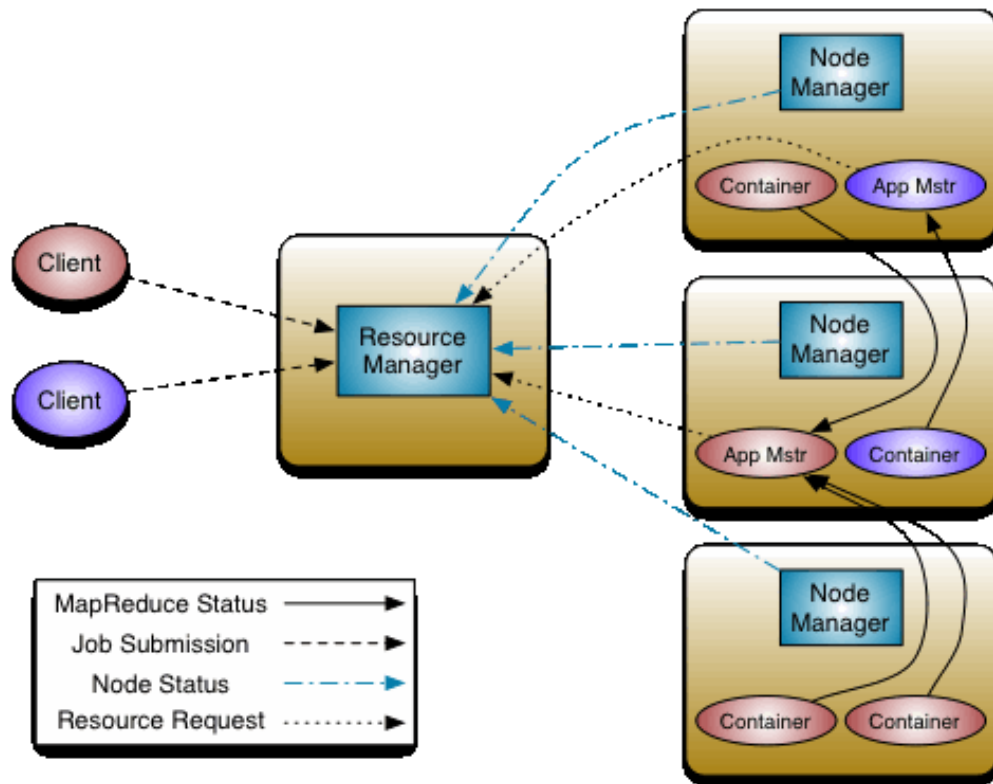
- 处理客户端请
- 启动/监控App
- 监控NodeMa
- 资源分配与调

## ❖ NodeManager

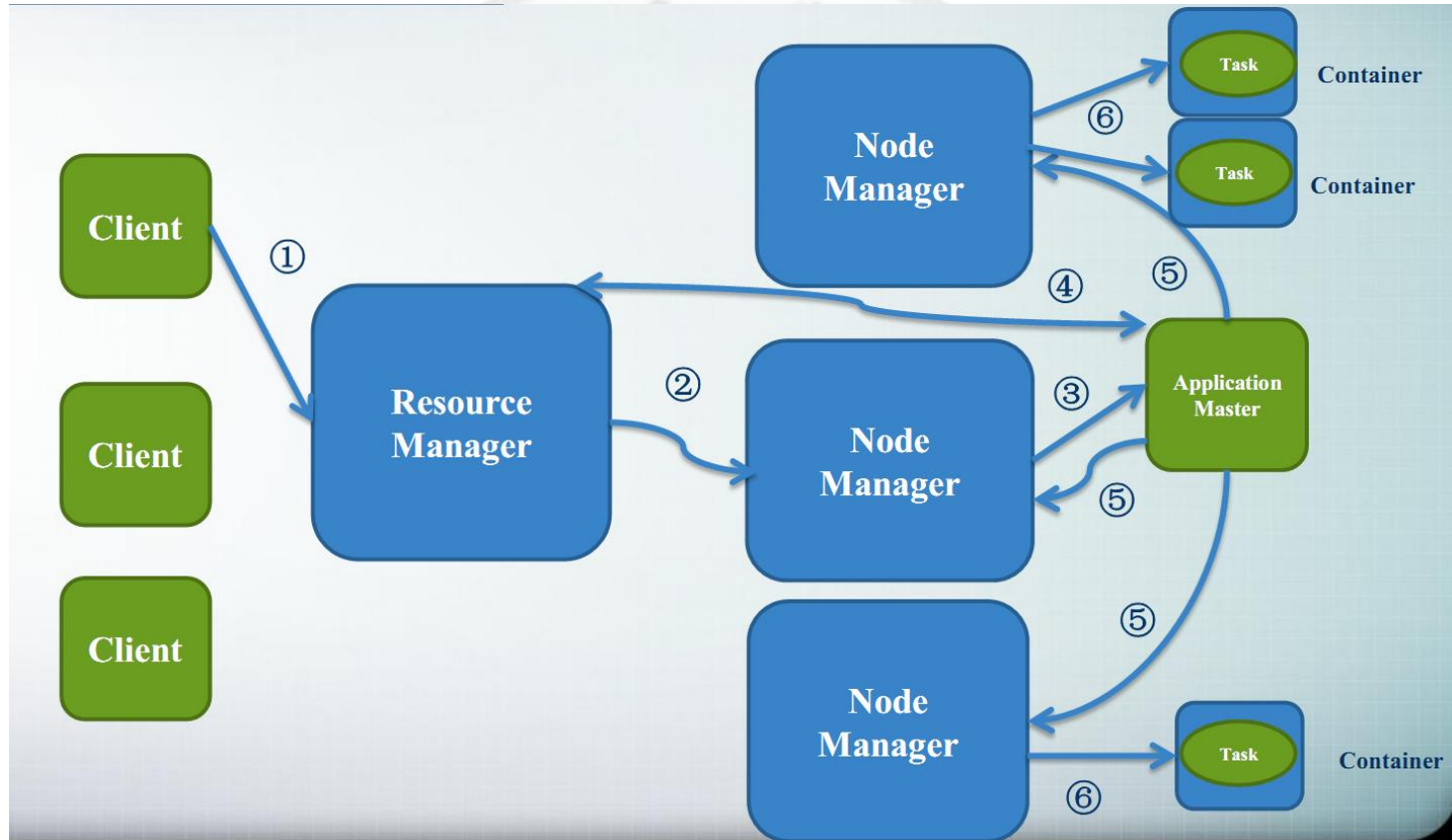
- 单个节点上的
- 处理来自Resc
- 处理来自App

## ❖ ApplicationMaster

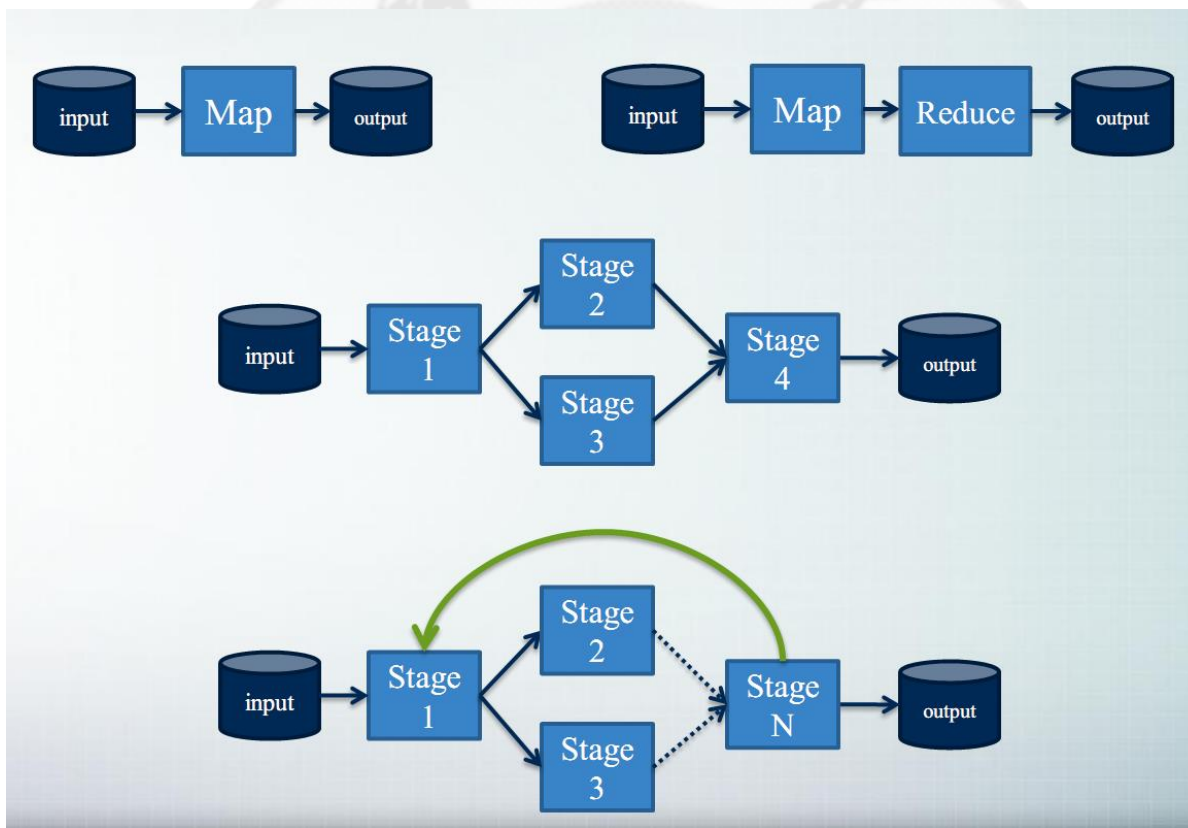
- 数据切分
- 为应用程序申
- 任务监控与容错



# Hadoop YARN 运行流程分析



# 应用程序的运行模型



# 运行在YARN 上的计算框架

---

- ❖ 离线计算框架: **MapReduce**
- ❖ DAG计算框架: **Tez**
- ❖ 流式计算框架: **Storm**
- ❖ 内存计算框架: **Spark**
- ❖ 图计算框架: **Giraph、GraphLib**



# 大数据时代

---

- ❖ 2011年，大数据开始火热
- ❖ 2012年，大数据开始被运营
- ❖ 2013年，大数据有了一定的积淀和运营基础，开始被运用在各种商业场景，大数据不再单单服务于某个公司，而是开放出来服务于行业乃至社会

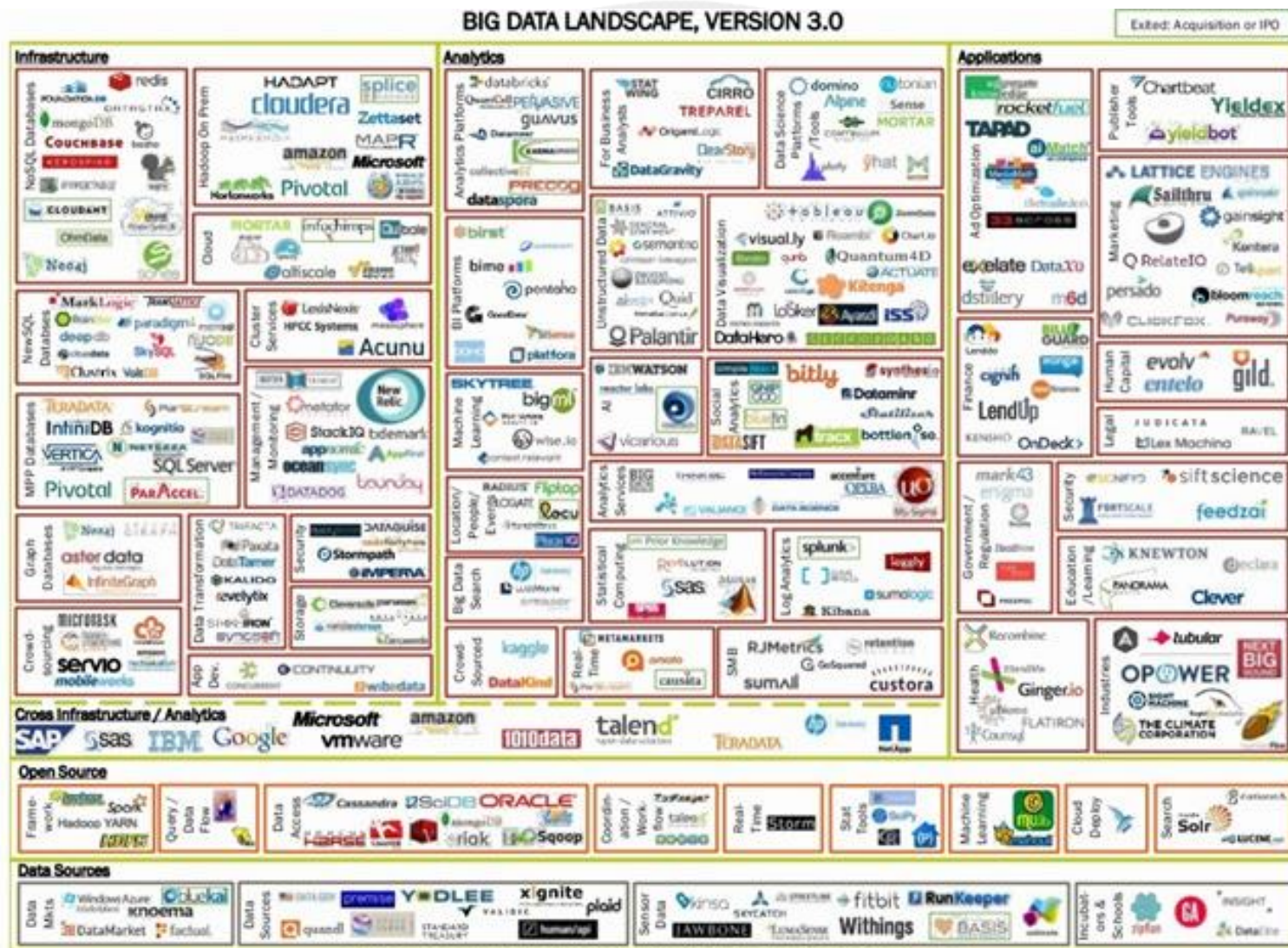


嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC



# 大数据生态地图



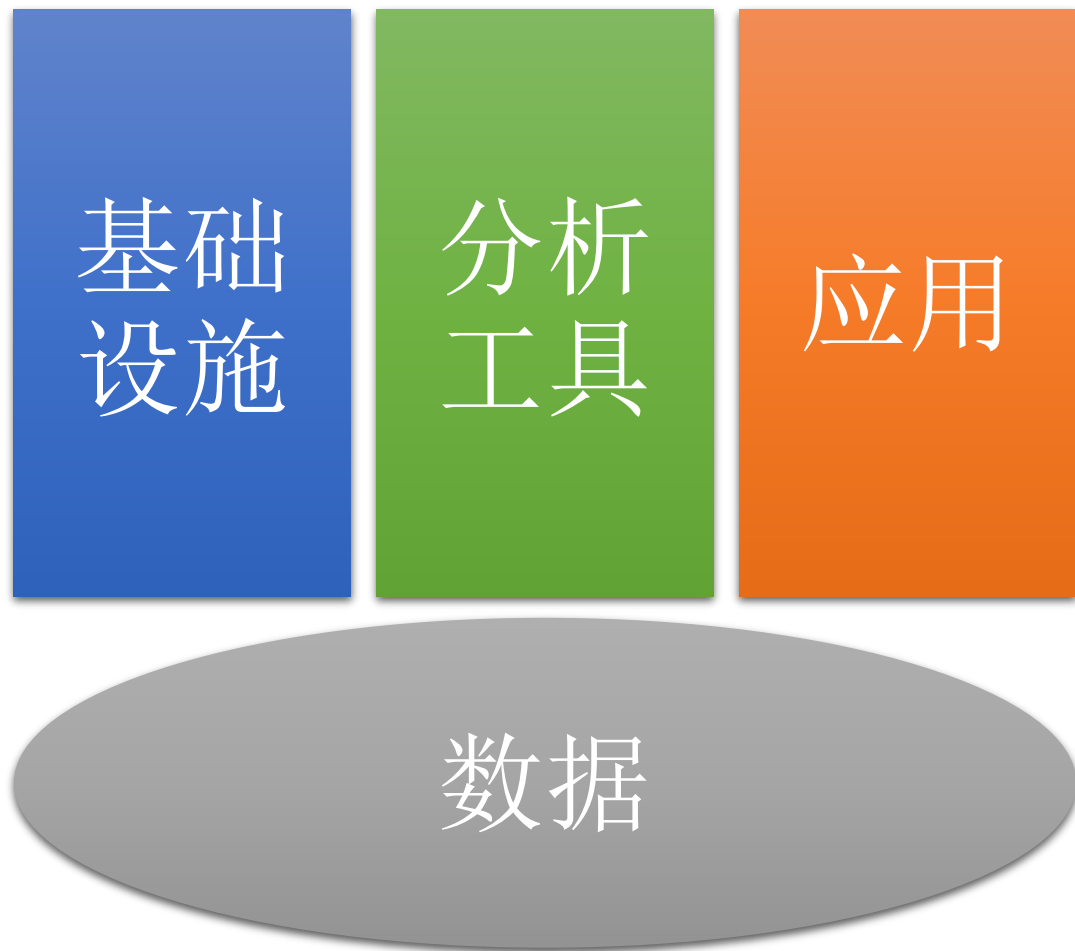
© Matt Turck (@mattturck), Sutan Dong (@sutandong) & FirstMark Capital (@firstmarkcap)

实验室

LABORATORY

SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC





大数据基础设施：  
Hadoop/Spark

大数据分析工具：从电子表格  
到时间线动画再到3D可视化

大数据应用：大数据应用的发展进程相对缓慢，但目前阶段大数据确实已经进入了应用层。从大数据生态地图3.0中我们可以看到，一些创业公司开发出了大数据通用应用，例如**大数据营销工具、CRM工具或防欺诈解决方案**等。还有一些大数据创业公司开发出了面向行业用户的垂直应用。**金融和广告行业**是大数据应用起步最早的行业，甚至在大数据概念出现之前就已经开始了。未来大数据还将在更多行业得到广泛应用，例如**医疗、生物科技（尤其是基因组学）和教育**等

# 大数据运用

- ❖ 商业运营
- ❖ 市场研究
- ❖ 客户跟踪分析
- ❖ 仓储问题
- ❖ 广告投放分析
- ❖ 零售业
- ❖ 快餐行业
- ❖ 铁路系统
- ❖ 城市管理
- ❖ 城市交通
- ❖ 公共安全
- ❖ 网络安全
- ❖ 能源行业
- ❖ 社交网络
- ❖ 政府作为
- ❖ 总统选举
- ❖ 汽车行业
- ❖ 智慧医疗
- ❖ 保险行业
- ❖ 通讯行业
- ❖ 个人服务
- ❖ 探索人生
- ❖ 体育运动
- ❖ 人才分析



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC

# 商业运营

- ❖ IBM智慧分析与洞察-美味的数据
- ❖ IBM智慧的分析洞察案例 美国辛辛那提动物园
- ❖ [1] 辛辛那提动物园使用了Cognos，为iPad提供了单一视图查看管理即时访问的游客和商务信息的服务。借此，动物园可以获得新的收入来源和提高营收，并根据这些信息及时调整营销政策。
- ❖ [2] 数据收集和分析工具能够帮助银行设立最佳网点，确定最好的网点位置，帮助这个银行更好地运作业务，推动业务的成长。



# 警务大数据

## ❖ 警务2.0：用大数据预防犯罪

- 《执法部门的Hadoop大数据解决方案》
- 具备预测能力的警察（IBM）
- 数据分析 IBM孟菲斯警察局案例

## ❖ 警务大数据案例：大数据预测分析与犯罪预防

- 洛杉矶警察局已经能够利用大数据分析软件成功的把辖区里的盗窃犯罪降低了33%，暴力犯罪降低了21%，财产类犯罪降低了12%。

## ❖ 警务大数据：纽约警察使用Twitter预测犯罪

- 根据研究结果，基于推特的KDE算法与传统KDE方法相比，在预测跟踪，刑事损害，和赌博问题的准确度上有所提升。但在纵火，绑架，勒索等问题上，准确度却相对较低。



# 智慧交通

- ❖ [1] 通过在城市多处设置的传感器，我们能随时掌握在某个地方有多少辆汽车，车速是多少。有了这些数据就可以建立起模型进行分析，从而指导人们确定出行计划，避免拥堵。
- ❖ [2] IBM的六位数据分析工程师准备通过整合、分析现有交通数据，以及来自社交媒体（Twitter）的新数据源，来医治波士顿的交通恶瘤。这些数据包括市政网联网能够实时采集的交通信号灯、二氧化碳传感器甚至汽车的数据，这些数据能够帮助乘客重新调整路线，节省时间，节省汽油。



---

# 欢迎加入 嵌入式系统实验室



嵌入式系统实验室

EMBEDDED SYSTEM LABORATORY  
SUZHOU INSTITUTE FOR ADVANCED STUDY OF USTC