



Hadoop数据分析平台 第1周

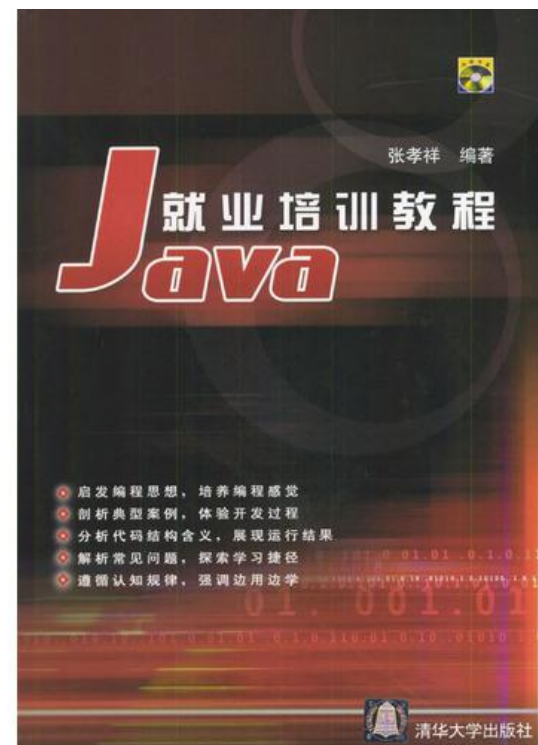
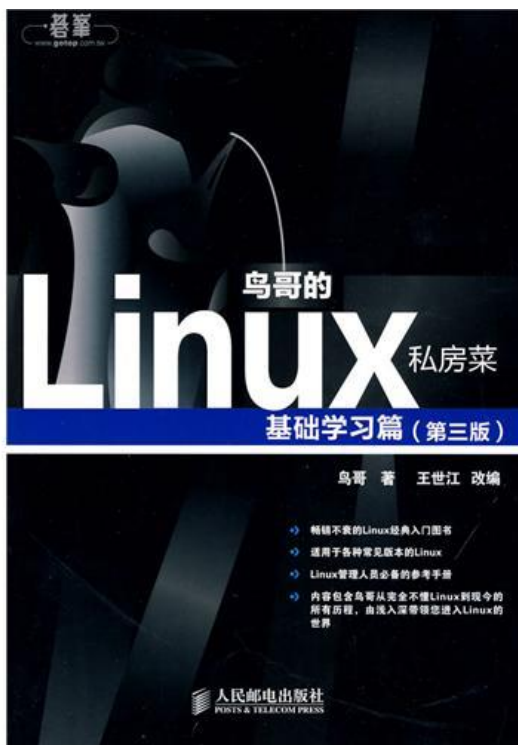
【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

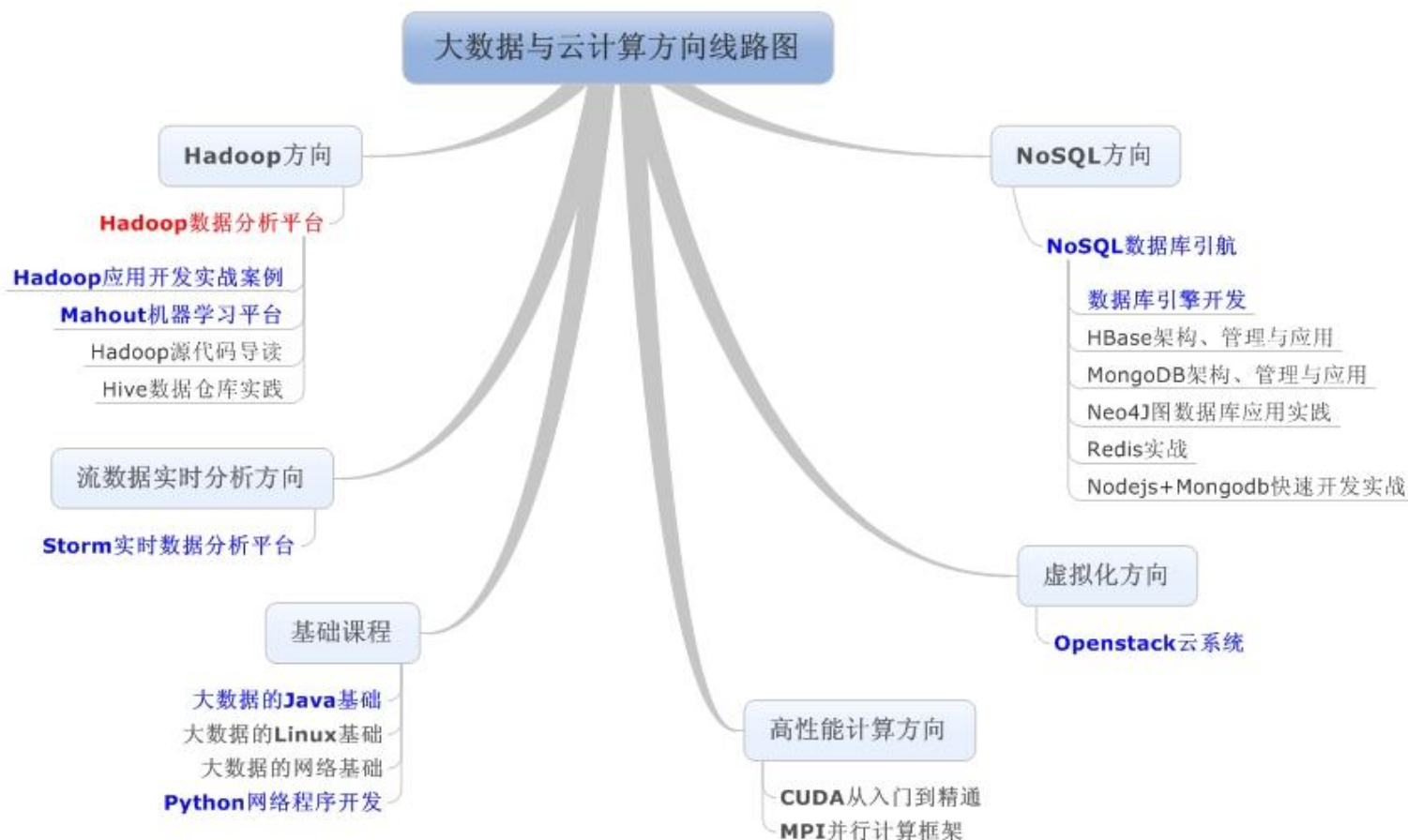
关于本课程的预备知识

- Linux：懂基本操作
- Java：能看懂Java程序



Hadoop是什么？

- 面向**大数据**处理
- 擅长**离线**数据分析
- **分布式**文件系统+计算框架
- Hadoop**不是数据库**，Hbase才是数据库
- Hadoop是一个**快速进化**的生态系统
- Hadoop会使用在哪些行业？



- 运维
- Hadoop程序员（M-R，源代码）
- 架构师
- 数据仓库工程师



- 部署：Hadoop（v1和v2），Hbase，Hive，Pig，Mahout
- 数据集成：Sqoop，Chukwa，Flume，与Oracle、Mysql等关系型数据库集成，与应用集成，与R等常用数据分析工具集成
- 掌握HDFS原理和基本操作
- 掌握Map-Reduce工作原理，知道怎样把M-R算法用于解决实际场景
- 掌握YARN框架的原理及使用方法，知道怎样写YARN应用程序
- 能书写基本的Map-Reduce程序，懂得提交作业和对运行状况进行监视
- 了解整个Hadoop生态系统的各个子产品，知道在大数据平台架构时应该使用哪些产品
- 具备初步阅读源代码的能力

典型实验环境（拥有服务器）

- 服务器：ESXi，可以在上面部署多台虚拟机，能同时启动3台
- PC：要求linux环境或windows+Cygwin，linux可以是standalone或者使用虚拟机
- SSH：windows下可以使用SecureCRT或putty等ssh client程序，作用是用来远程连接linux服务器，linux下可以直接使用ssh命令
- Vmware client：用于管理ESXi
- Hadoop：使用1.x或2.x

典型实验环境（只有PC或笔记本，基于win）

- 至少4G内存，最好运行64位windows系统，因为32位xp只能支持3G多的内存
- 安装vmware workstation或virtual box
- 部署3台虚拟机，能同时运行，如果只能运行2台虚拟机，那么可以把host也作为一个节点（使用cygwin），虚拟网络配置为网桥方式
- 安装linux和java
- 如果配置实在太低只好使用伪分布式

Hadoop的思想之源：Google

- Google搜索引擎，Gmail，安卓，AppspotGoogle Maps，Google earth，Google 学术，Google翻译，Google+，下一步Google what？？



Google的低成本之道

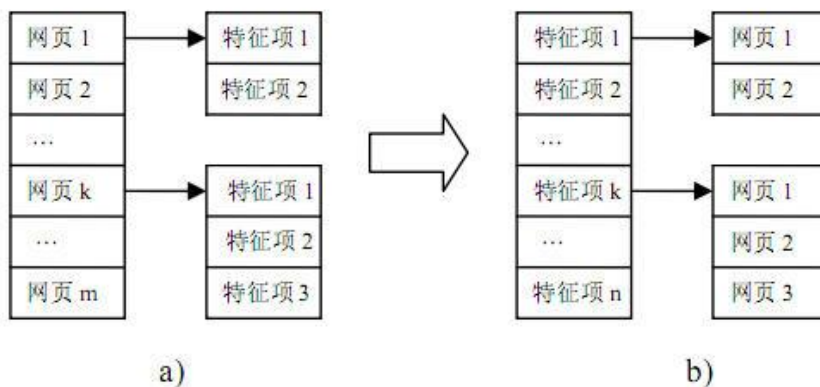
- 不使用超级计算机，不使用存储（淘宝的去i，去e，去o之路）
- 大量使用普通的pc服务器（去掉机箱，外设，硬盘），提供有冗余的集群服务
- 全世界多个数据中心，有些附带发电厂
- 运营商向Google倒付费



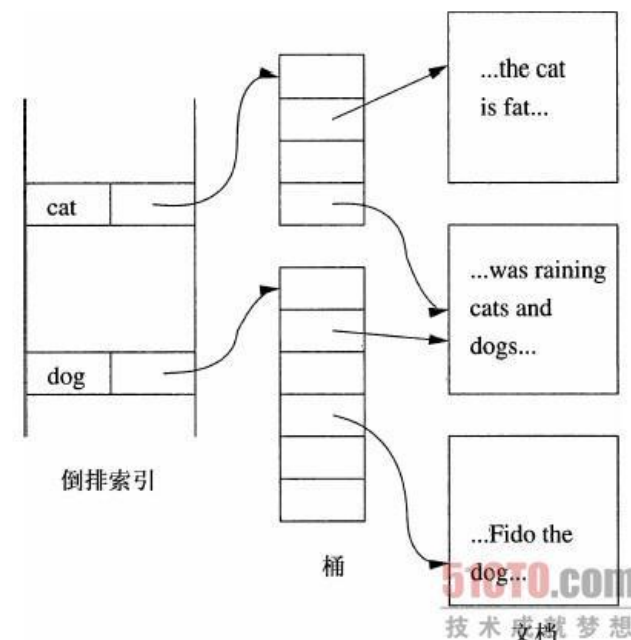
- 位于 Mountain View , Calif 总部的数据中心
- 总功率为10000千瓦，拥有45个集装箱，每个集装箱中有1160台服务器，该数据中心的能效比为1.25（ PUE 为 1 表示数据中心没有能源损耗，而根据2006年的统计，一般公司数据中心的能效比为 2.0 或更高。Google 的 1.16 已经低于美国能源部2011年的1.2 的目标）



- 大量的网页怎么存储？
- 搜索算法
- Page-Rank计算问题



单词ID	单词	倒排列表 (DocID;TF)
1	谷歌	(1;1),(2;1),(3;2),(4;1),(5;1)
2	地图	(1;1),(2;1),(3;1),(4;1),(5;1)
3	之父	(1;1),(2;1),(4;1),(5;1)
4	跳槽	(1;1),(4;1)
5	Facebook	(1;1),(2;1),(3;1),(4;1),(5;1)
6	加盟	(2;1),(3;1),(5;1)
7	创始人	(3;1)
8	拉斯	(3;1),(5;1)
9	离开	(3;1)
10	与	(4;1)
11	Wave	(4;1)
12	项目	(4;1)
13	取消	(4;1)
14	有关	(4;1)
15	社交	(5;1)
16	网站	(5;1)



- 这是Google最核心的算法，用于给每个网页价值评分，是Google “在垃圾中找黄金” 的关键算法，这个算法成就了今天的Google

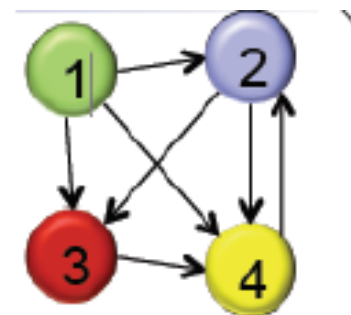
PageRank vector \mathbf{q} is defined as $\mathbf{q} = G\mathbf{q}$

where $G = \alpha S + (1 - \alpha) \frac{1}{n} U$

- S is the destination-by-source stochastic matrix,
- U is all one matrix.
- n is the number of nodes
- α is the weight between 0 and 1 (e.g., 0.85)

Algorithm: Iterative powering for finding the first eigen-vector

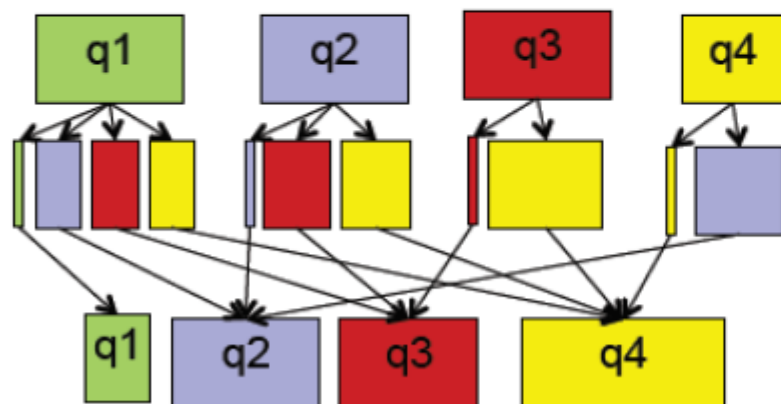
$$\mathbf{q}^{next} = G\mathbf{q}^{cur}$$



$$G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 0 & 0 \\ 1/3 & 1/2 & 1 & 0 \end{bmatrix}$$



Map: distribute PageRank q_i



Reduce: update new PageRank

PageRank Map()

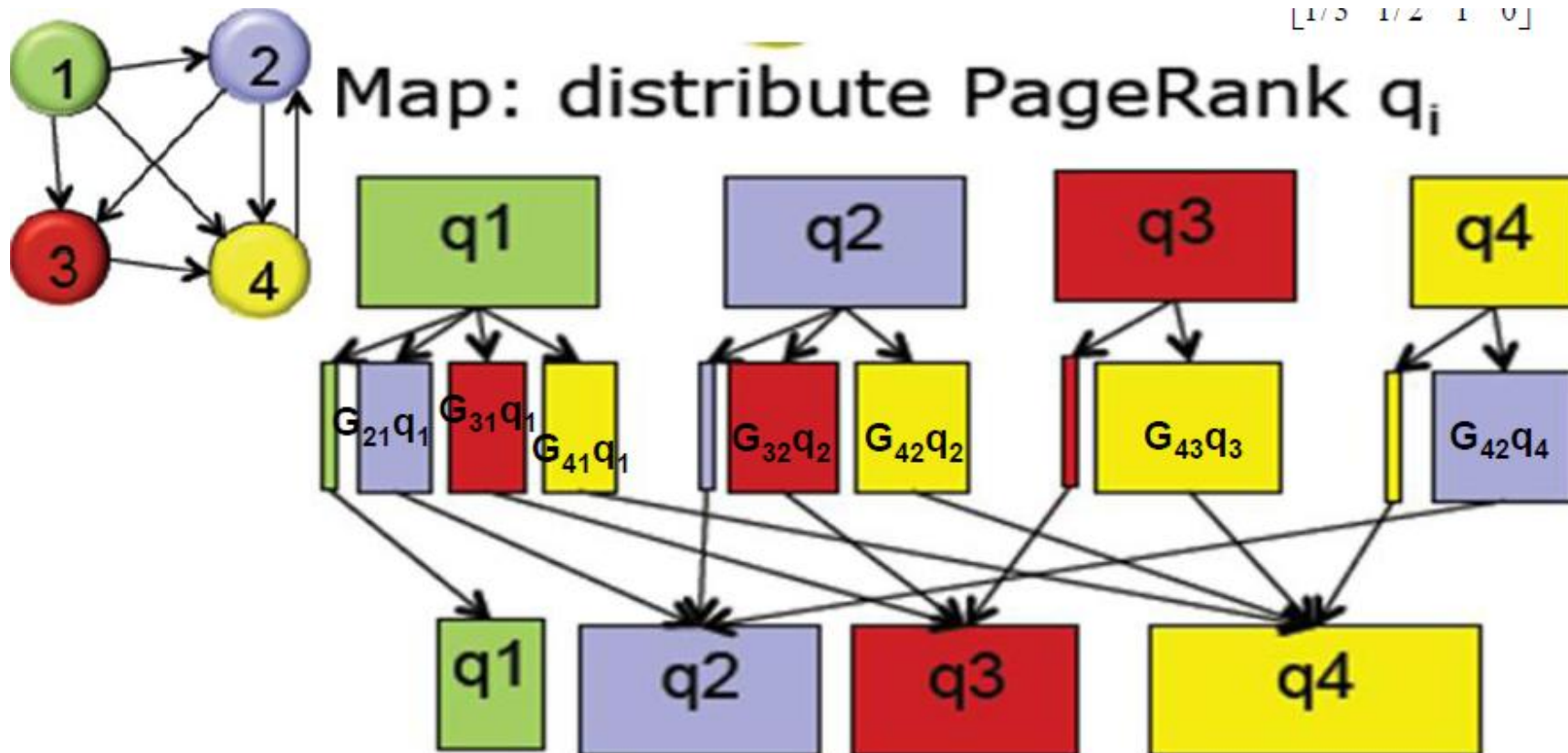
- Input:

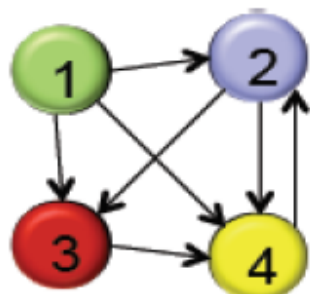
- key = page x ,
- value = $(q_x, \text{links}[y_1 \dots y_m])$

- Output:

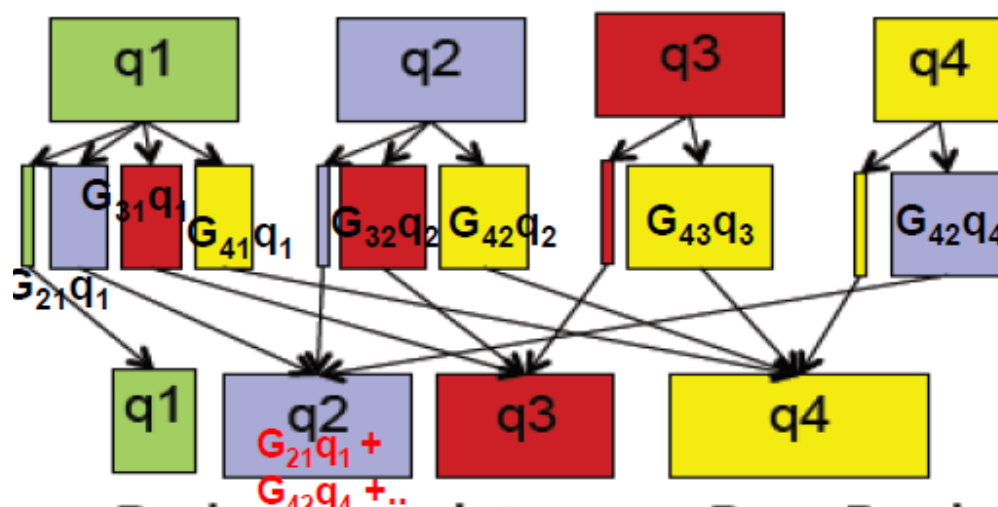
- key = page x ,
- value = partial_x
 - $\text{Emit}(x, 0)$ //guarantee all pages will be emitted
 - For each outgoing link y_i :
 $\text{Emit}(y_i, G_{ix}q_x)$

[1/3 1/2 1 0]





Map: distribute PageRank q_i



Reduce: update new PageRank

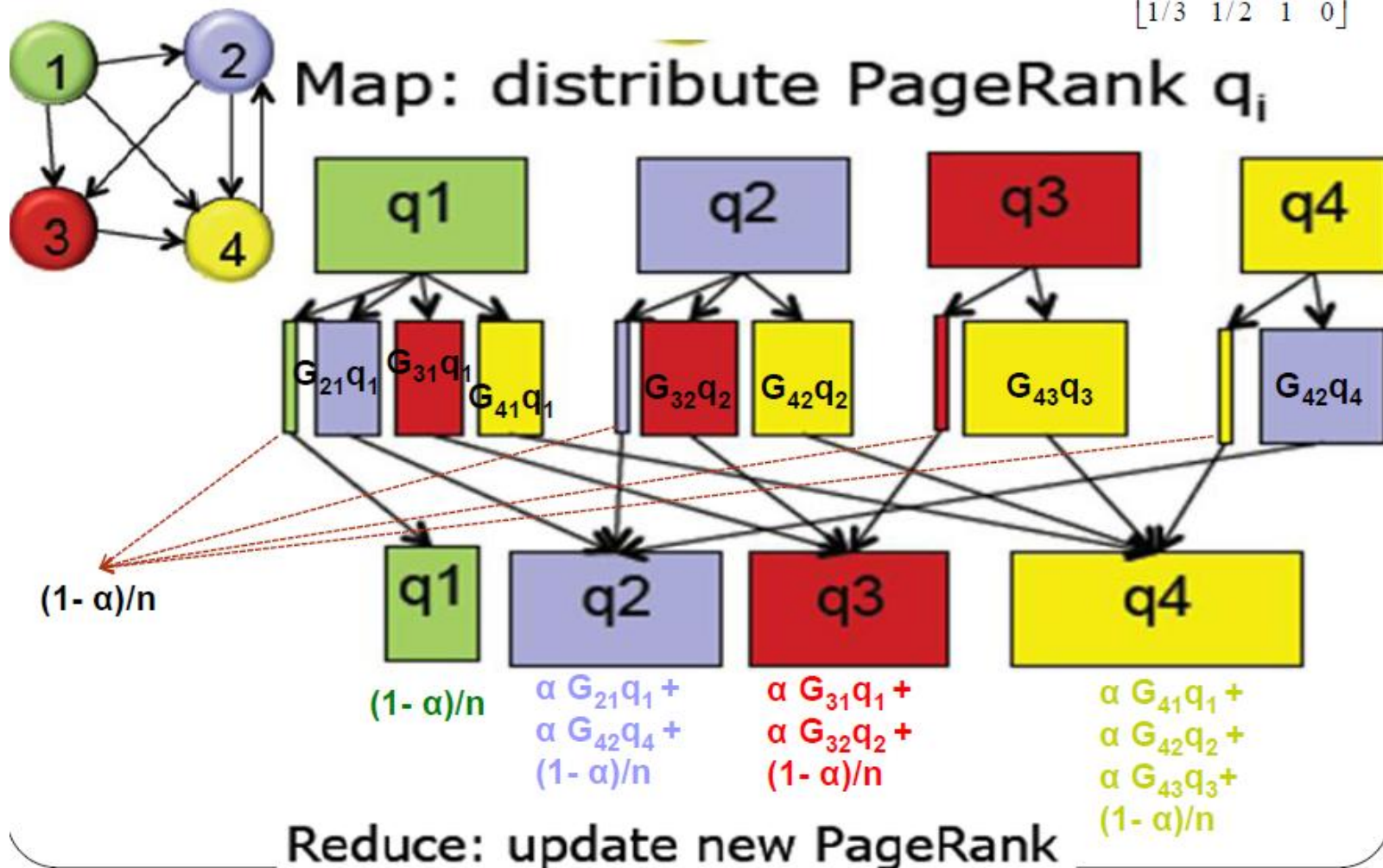
PageRank Reduce()

- Input:
 - key = page x ,
 - value = the list of $[\text{partial}_x]$
- Output:
 - key = page x ,
 - value = PageRank q_x
 - $q_x = 0$
 - For each partial value d in the list:

$$q_x += d$$
 - $q_x = \alpha q_x + (1 - \alpha)/n$
 - Emit(x , q_x)

$$q^{next} = Gq = \alpha Sq + (1 - \alpha) \frac{1}{n} Uq$$

[1/3 1/2 1 0]



Google带给我们的关键技术和思想

- GFS
- Map-Reduce
- Bigtable (后面讲)

- Doug Cutting开创的开源软件，用java书写代码，实现与Google类似的全文搜索功能，它提供了全文检索引擎的架构，包括完整的查询引擎和索引引擎
- 早期发布在个人网站和SourceForge，2001年年底成为apache软件基金会jakarta的一个子项目
- Lucene的目的是为软件开发人员提供一个简单易用的工具包，以方便的在目标系统中实现全文检索的功能，或者是以此为基础建立起完整的全文检索引擎
- 对于大数量的场景，Lucene面对与Google同样的困难。迫使Doug Cutting学习和模仿Google解决这些问题的办法
- 一个微缩版：Nutch



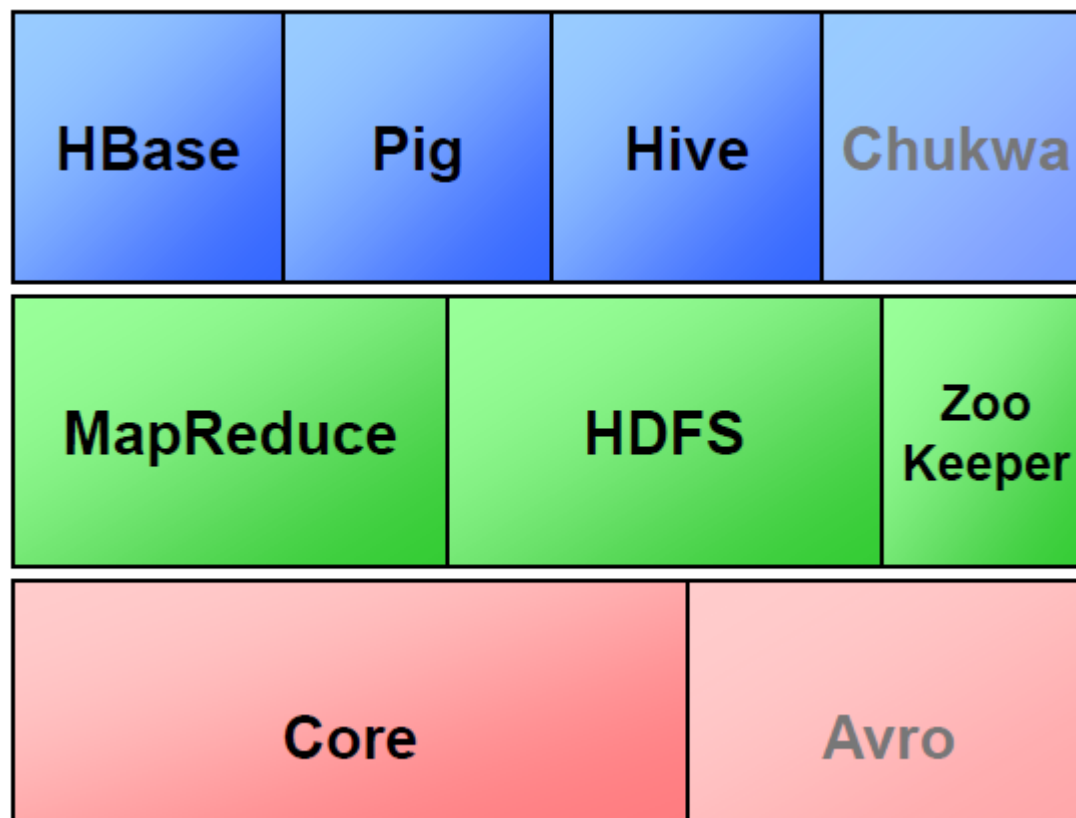
从lucene到nutch，从nutch到hadoop

- 2003-2004年，Google公开了部分GFS和Mapreduce思想的细节，以此为基础Doug Cutting等人用了2年业余时间实现了DFS和Mapreduce机制，使Nutch性能飙升
- Yahoo招安Doug Cutting及其项目
- Hadoop 于 2005 年秋天作为 [Lucene](#)的子项目 [Nutch](#)的一部分正式引入Apache基金会。2006 年 3 月份，Map-Reduce 和 Nutch Distributed File System (NDFS) 分别被纳入称为 Hadoop 的项目中
- 名字来源于Doug Cutting儿子的玩具大象

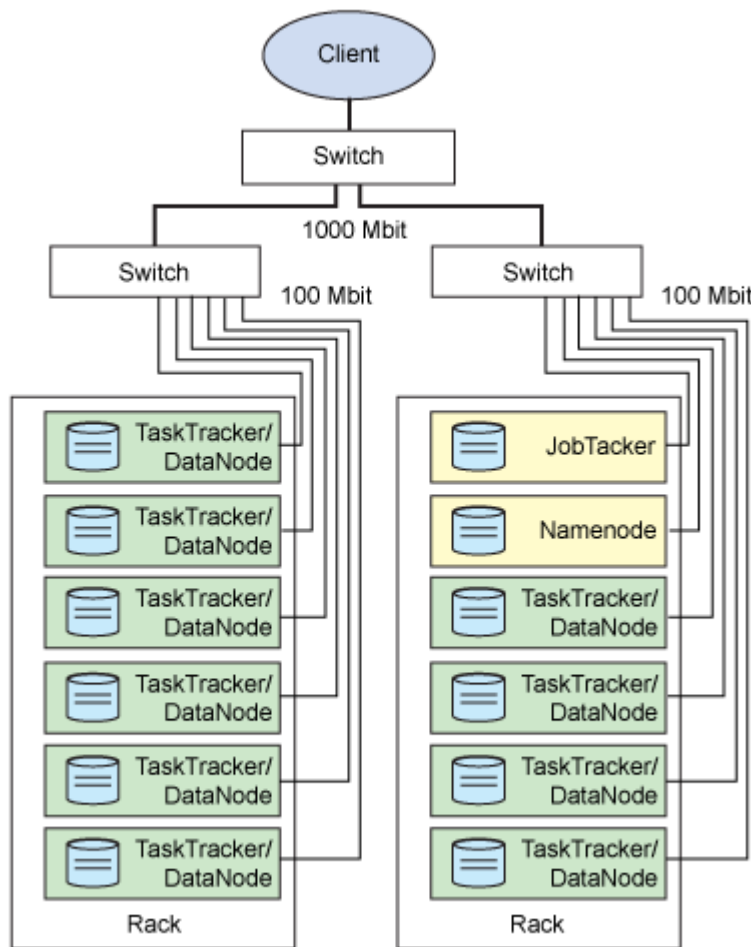


- 实现云计算的事实标准开源软件
- 包含数十个具有强大生命力的子项目
- 已经能在数千节点上运行，处理数据量和排序时间不断打破世界纪录

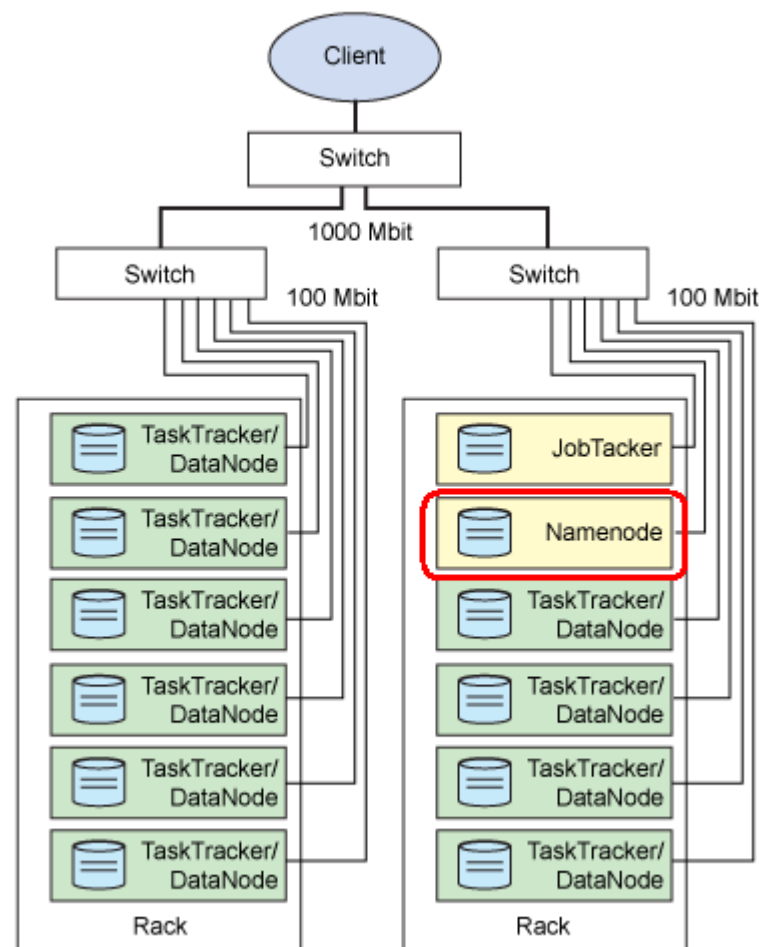




Hadoop的架构

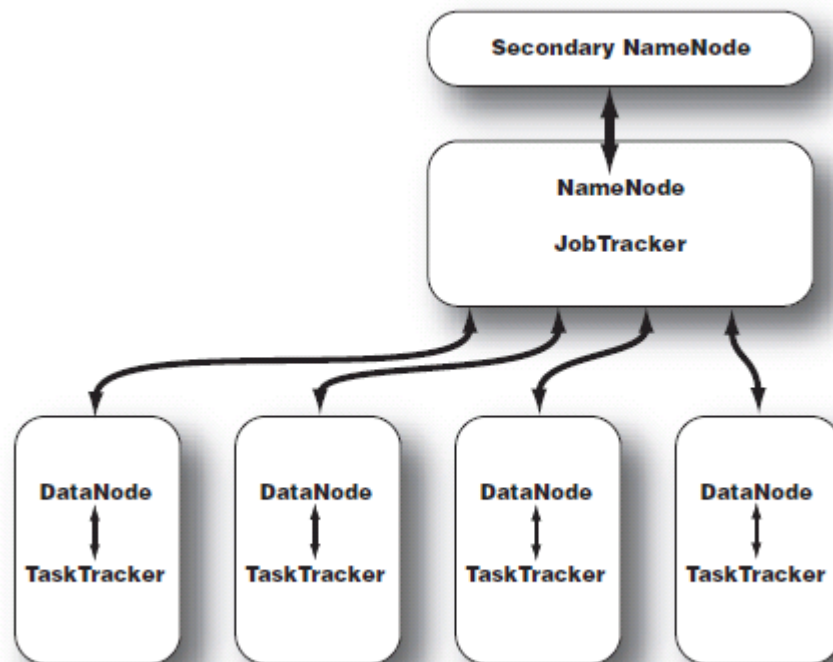


- HDFS的守护程序
- 纪录文件是如何分割成数据块的，以及这些数据块被存储到哪些节点上
- 对内存和I/O进行集中管理
- 是个单点，发生故障将使集群崩溃

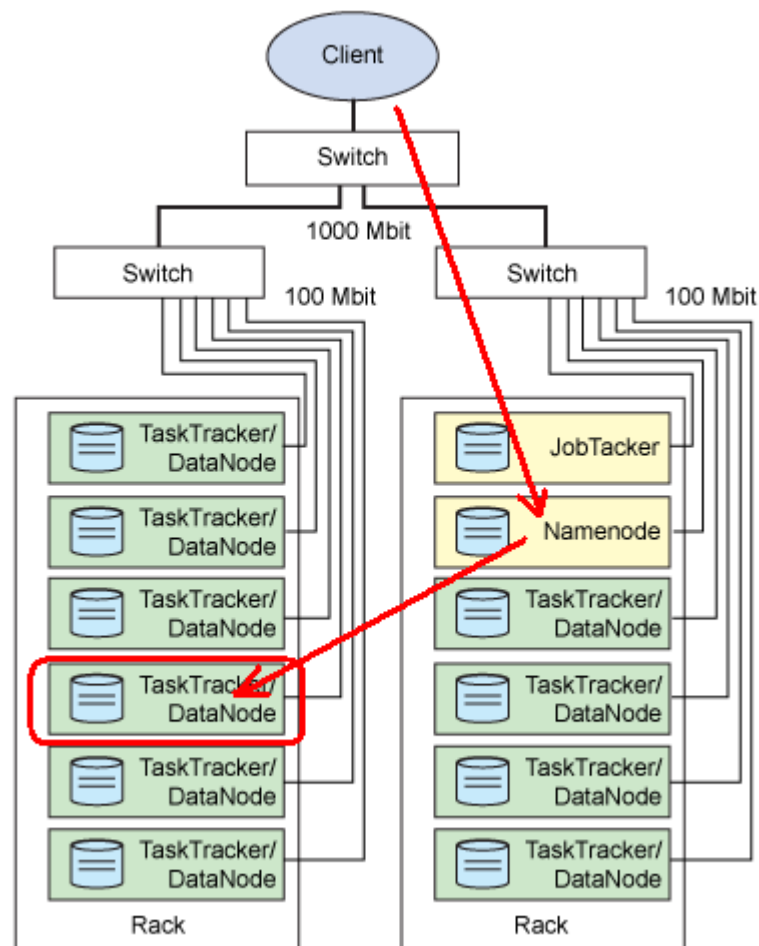


Secondary Namenode

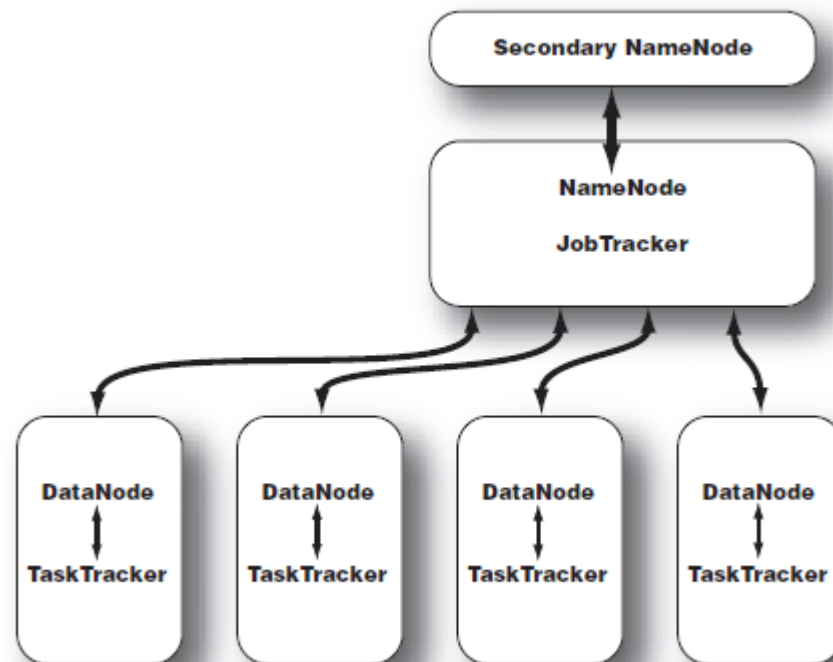
- 监控HDFS状态的辅助后台程序
- 每个集群都有一个
- 与NameNode进行通讯，定期保存HDFS元数据快照
- 当NameNode故障可以作为备用NameNode使用



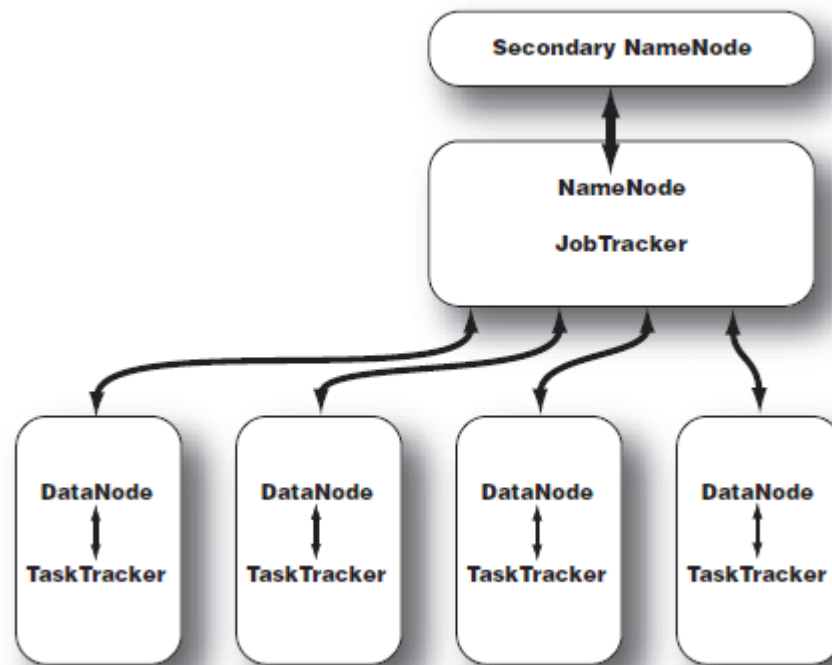
- 每台从服务器都运行一个
- 负责把HDFS数据块读写到本地文件系统

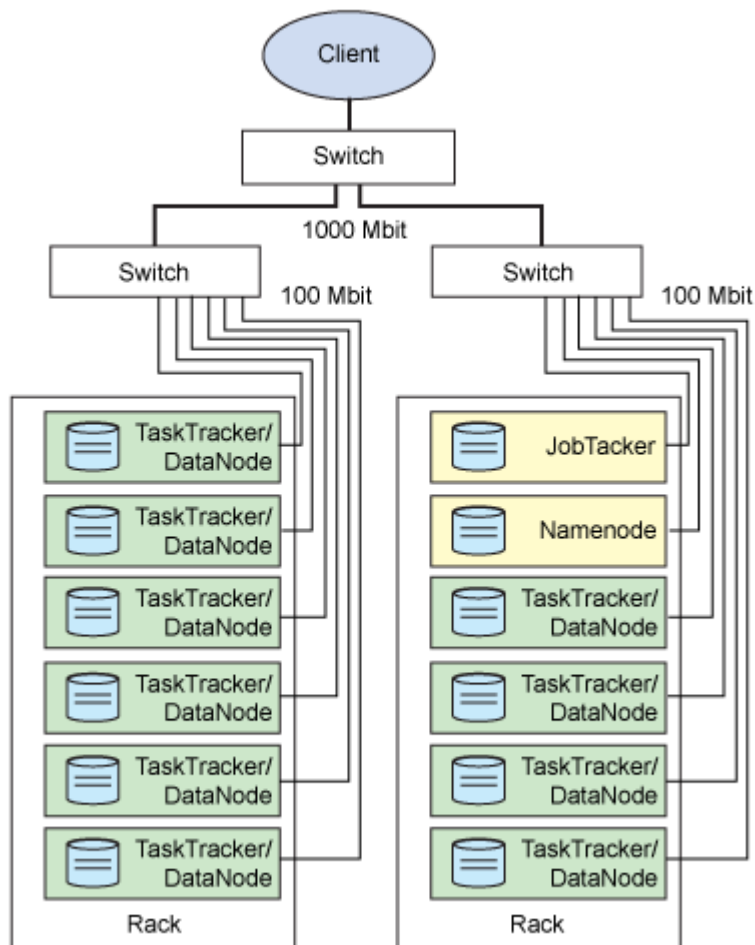


- 用于处理作业（用户提交代码）的后台程序
- 决定有哪些文件参与处理，然后切割task并分配节点
- 监控task，重启失败的task（于不同的节点）
- 每个集群只有唯一一个JobTracker，位于Master节点



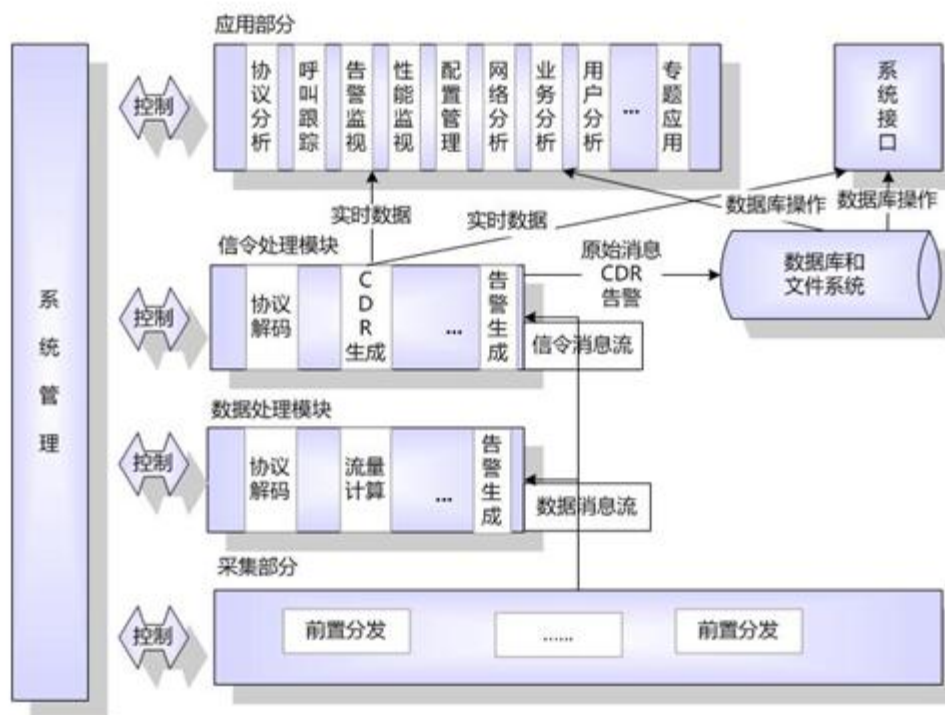
- 位于slave节点上，与datanode结合（代码与数据一起的原则）
- 管理各自节点上的task（由jobtracker分配）
- 每个节点只有一个tasktracker，但一个tasktracker可以启动多个JVM，用于并行执行map或reduce任务
- 与jobtracker交互





- Master : Namenode、Secondary Namenode、Jobtracker。浏览器（用于观看 管理界面），其它Hadoop工具
- Slave : Tasktracker、Datanode
- Master不是唯一的

Why hadoop ?

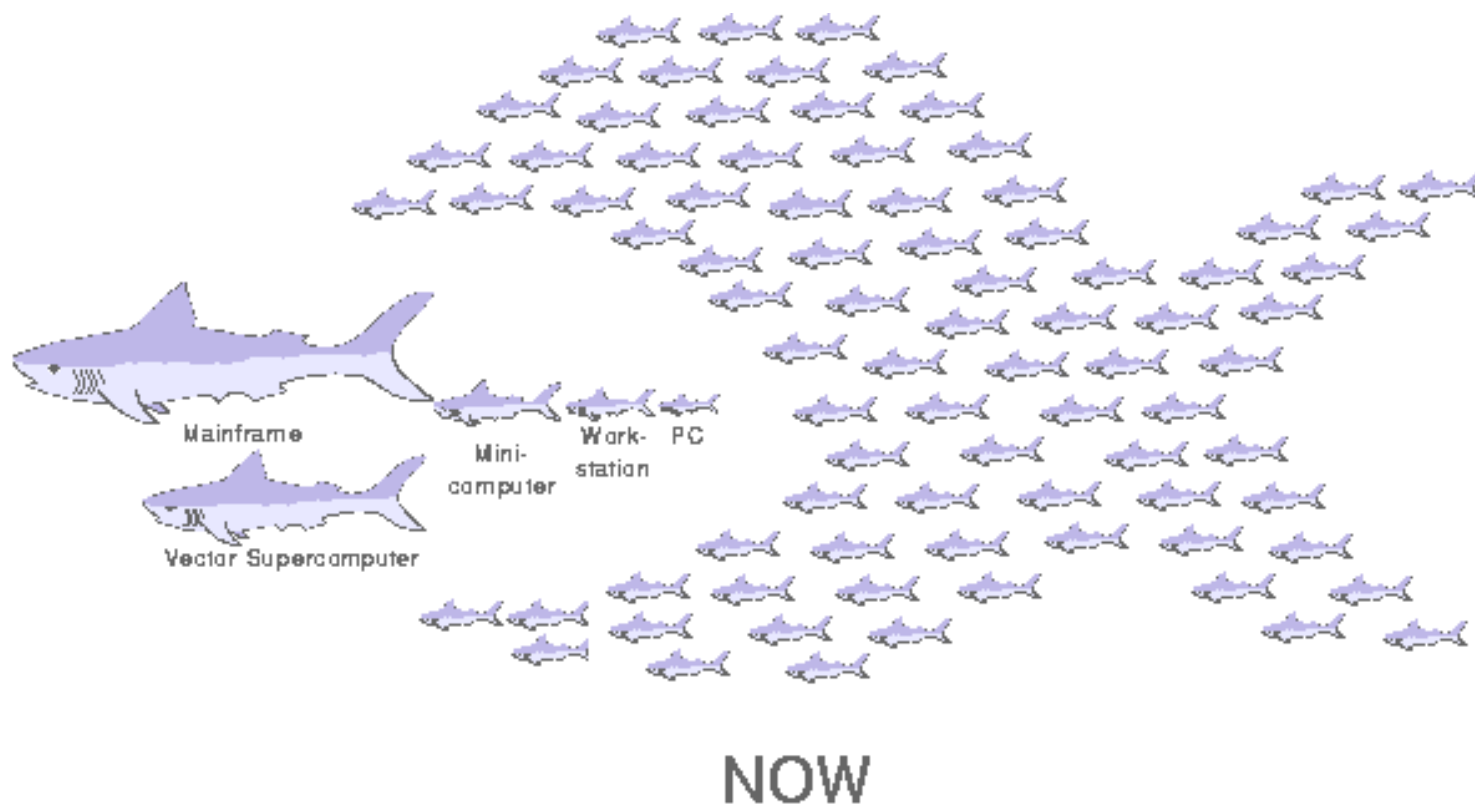


- 原数据库服务器配置：HP小型机，128G内存，48颗CPU，2节点RAC，其中一个节点用于入库，另外一个节点用于查询
- 存储：HP虚拟化存储，>1000个盘
- 数据库架构采用Oracle双节点RAC
- 问题：1 **入库瓶颈** 2 **查询瓶颈**

数据分析者面临的问题

- 数据日趋庞大，无论是入库和查询，都出现性能瓶颈
- 用户的应用和分析结果呈整合趋势，对实时性和响应时间要求越来越高
- 使用的模型越来越复杂，计算量指数级上升

- 完美解决性能瓶颈，在可见未来不容易出现新瓶颈
- 过去所拥有的技能可以平稳过渡。比如SQL、R
- 转移平台的成本有多高？平台软硬件成本，再开发成本，技能再培养成本，维护成本



Why not Hadoop ?

- Java ?
- 难以驾驭 ?
- 数据集成困难 ?
- Hadoop vs Oracle

- 主流：Java程序
- 轻量级的脚本语言：Pig
- SQL技巧平稳过渡：Hive
- NoSQL：HBase

- **Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。**
- **关于逆向收费式网络的详情，请看我们的培训网站 <http://edu.dataguru.cn>**



Thanks

FAQ时间