

云计算

第2讲

商业云平台

任桐炜，李传艺

南京大学软件学院

2017-09-18



背景

- 云计算起源于商业需求，并在商业中得到广泛应用
- 商业云平台的技术大部分都不公开
- 代表性商业云平台
 - **Google (Google App Engine, Google services)**
 - **Amazon (Amazon AWS)**
 - **Microsoft Azure**
 - **Alibaba (Aliyun ODPS)**
 -



Google云平台

SaaS

Google 文件

Google 地图
谷歌 中国

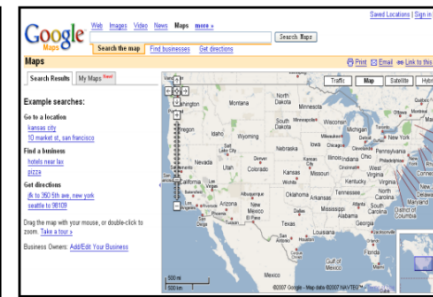
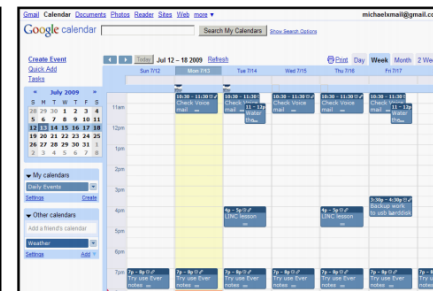
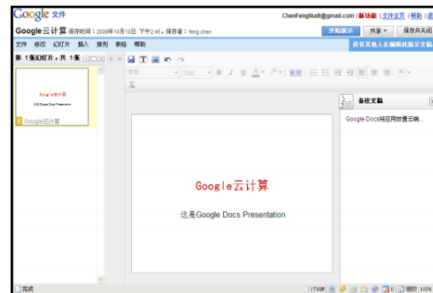
Google 日历

Gmail
by Google BETA



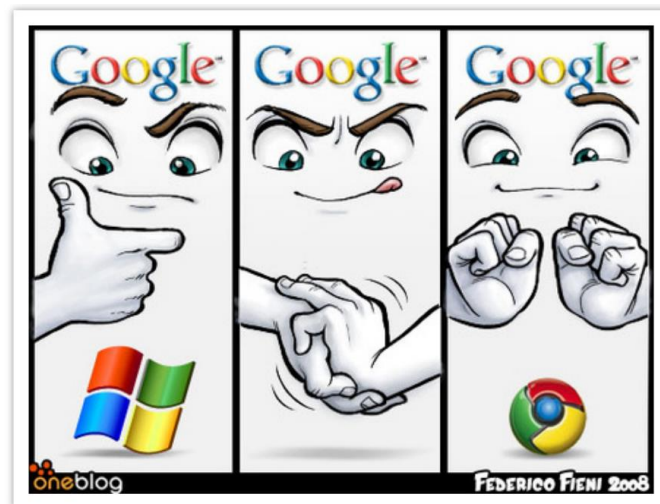
PaaS

Google
app engine



需求和设想

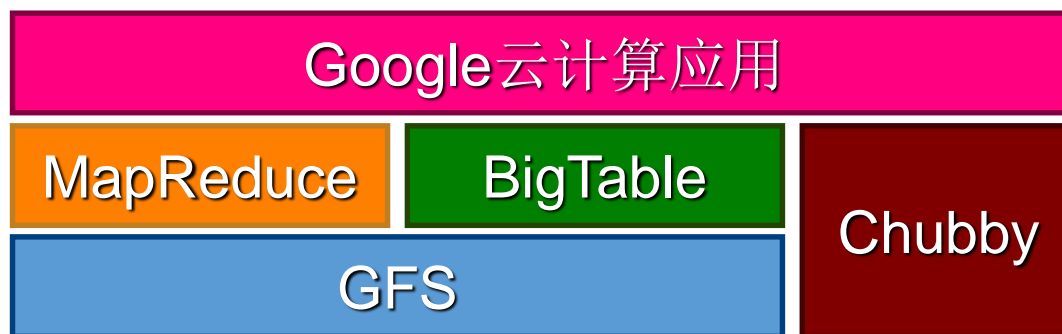
- **Google**应用的特性
 - 海量用户 + 海量数据
 - 需要具备较强的可伸缩性
- **Google**系统架构的设想
 - 应用向互联网迁移
 - 数据向互联网迁移
 - 计算能力向互联网迁移
 - 存储空间向互联网迁移



浏览器＝操作系统

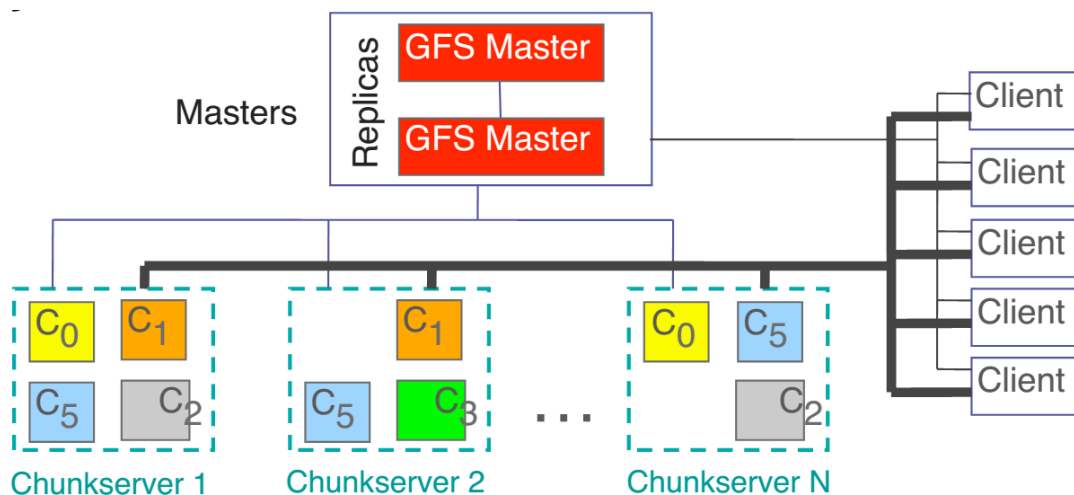
Google的四大法宝

- **Google**云计算平台的技术架构
 - 文件存储: **GFS**
 - The Google File System
 - 并行数据处理: **MapReduce**
 - MapReduce: Simplified Data Processing on Large Clusters
 - 结构化数据表: **BigTable**
 - Bigtable: A Distributed Storage System for Structured Data
 - 分布式锁: **Chubby**
 - The Chubby lock service for loosely-coupled distributed systems



GFS

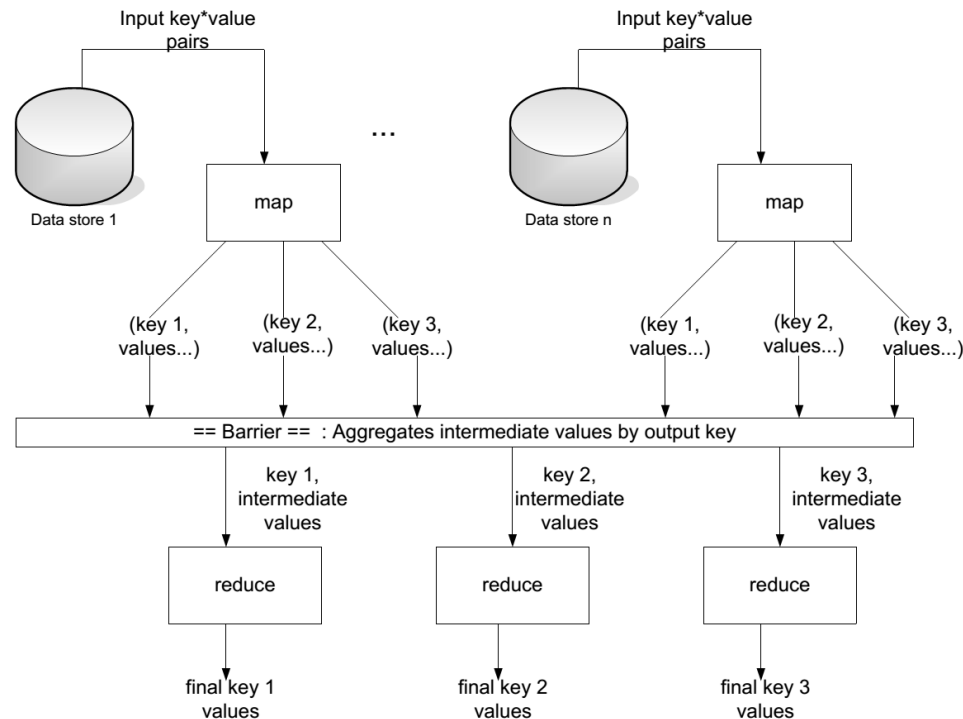
- 设计思路
 - 将文件划分为固定大小的块（**Chunk**）存储
 - 通过冗余来提高可靠性
 - 通过“单个” **master**来协调数据访问、元数据存储
 - 无缓存



MapReduce

Two phases of data processing

- Map: $(in_key, in_value) \rightarrow \{(key_j, value_j) \mid j = 1 \dots k\}$
- Reduce: $(key, [value_1, \dots, value_m]) \rightarrow (key, f_value)$



BigTable

`(row:string, column:string, time:int64)->string`

- 行：表中的数据根据行关键字按词典序排序
- 列：按照列族存储，每个族中的数据属于同一个类型
- 时间戳：保存不同时期的数据
- 物理划分：表 => 子表 => **SSTable**文件

Row Key	Time Stamp	Column Contents	Column Anchor		Column "mime"
			cnnsi.com	my.look.ca	
"com.cnn. www"	T9		CNN		
	T8			CNN.COM	
	T6	"<html>.." "			Text/html
	T5	"<html>.." "			
	t3	"<html>.." "			



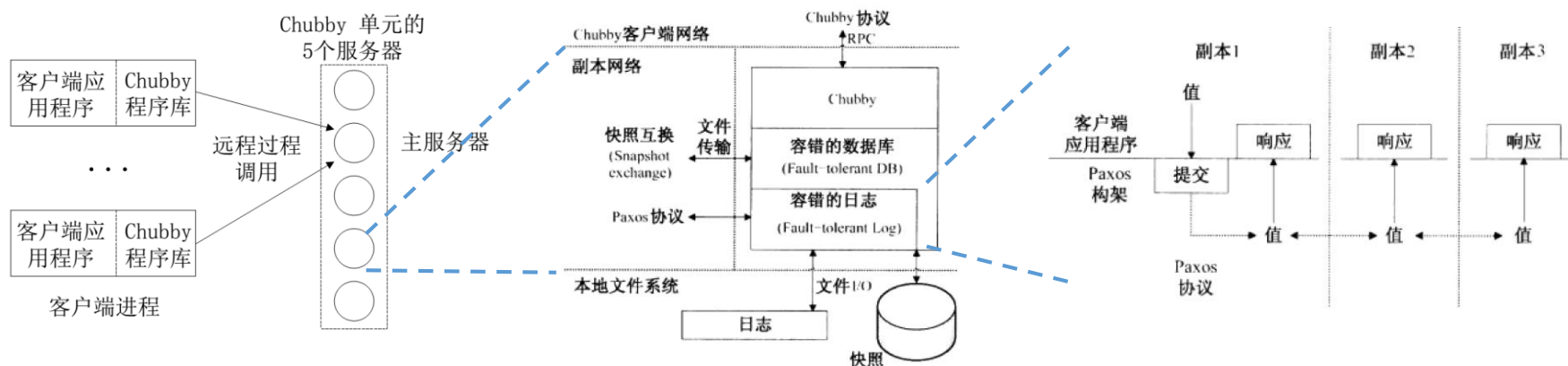
Row Key	Time Stamp	Column: Contents	
Com.cnn.www	T6	"<html>.." "	
	T5	"<html>.." "	
	T3	"<html>.." "	

Row Key	Time Stamp	Column: Anchor	
Com.cnn.www	T9	Anchor:cnnsi.com	CNN
	T5	Anchor:my.look.ca	CNN.COM

Row Key	Time Stamp	Column: mime
Com.cnn.www	T6	text/html

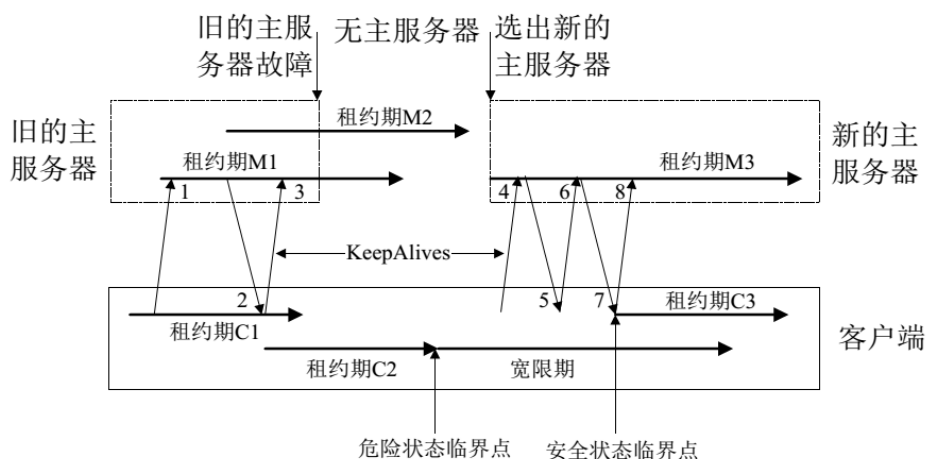
Chubby

- 基于松耦合分布式系统的锁服务
 - 采用**Paxos**算法：解决一致性问题
 - 粗粒度的锁：更长的持续时间（减少换锁的系统开销）
 - 建议性的锁（而非强制性的锁）：更大的灵活性
- 功能
 - 服务器端：选举主服务器
 - 客户端：与服务器端通过远程过程调用（RPC）连接，每个应用程序有一个**Chubby**程序库

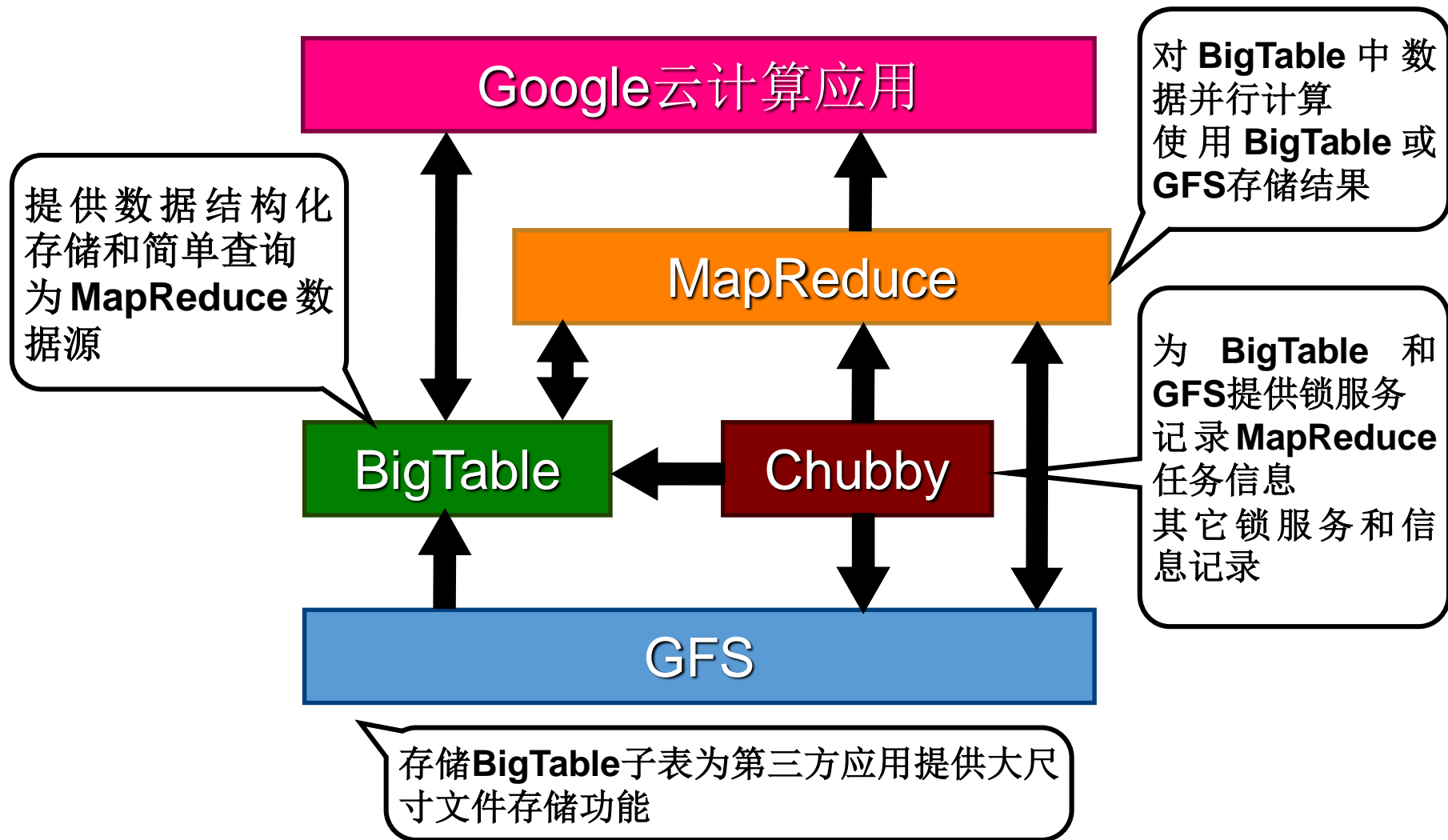


Chubby (续)

- 分布式、存储大量小文件的文件系统
 - 服务信息的直接存储
 - 客户端和主服务器的通信：**KeepAlive**握手协议
 - 客户端租约过期：宽限期（默认45秒），不断探询
=> 与新主服务器续约或终止会话
 - 主服务器出错：对用户透明，选举新的主服务器



Google云平台小结



Amazon云平台

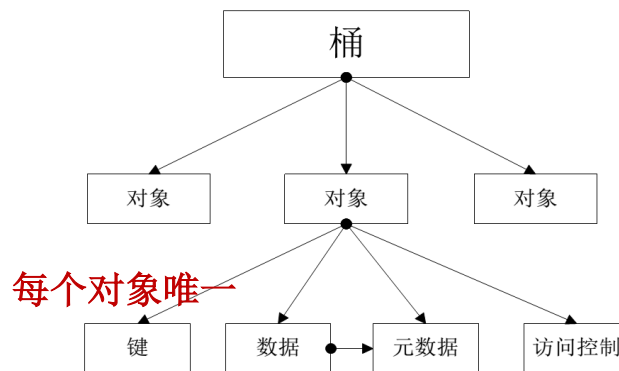
- 目的
 - 将硬件设备等基础资源封装成服务供用户使用（**IaaS**）
 - 在此基础上，用户构建应用层，并进一步开发应用程序
- 提供的云服务
 - 简单存储服务**S3**（**Simple Storage Service**）
 - 弹性计算云**EC2**（**Elastic Compute Cloud**）
 - 简单数据库服务**SimpleDB**（**Simple Database**）
 - 简单队列服务**SQS**（**Simple Queue Service**）
 -



简单存储服务S3

- 作用
 - 通过接口将任意类型的数据临时或永久地存储在服务器上（架构在**Dynamo**上）
 - 特点：可靠，易用，低成本
- 主要操作（操作目标可以是桶或对象）
 - **Get**：获取桶中的对象；获取对象的数据和元数据
 - **Put**：创建或更新桶；创建或更新对象
 - **List**：列出桶中所有的键
 - **Delete**：删除桶；删除对象
 - **Head**：获取对象的元数据

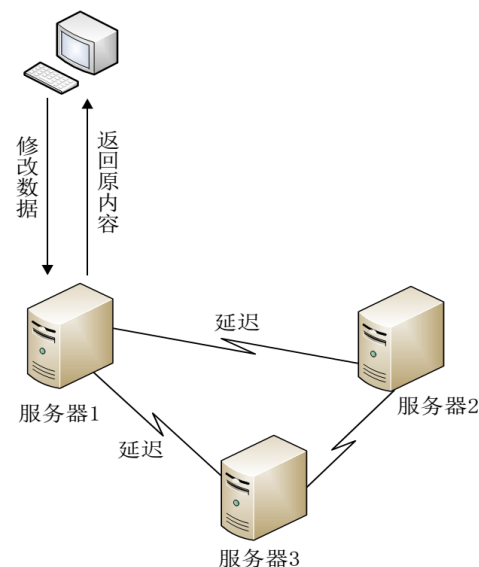
桶不可以嵌套
桶的名称在**S3**服务器中全局唯一



不可以重命名对象
不可以部分修改对象的数据

简单存储服务S3（续）

- 数据安全性
 - 采用冗余存储的方式，每个数据都产生多个副本，并将这些副本保存在不同的服务器上
- 最终一致性模型
 - 当数据被传播到所有节点前，返回原数据
 - 几种情形
 - 读取新写入的对象
 - 新写入对象后列出桶中所有对象
 - 更新数据后立即读取
 - 删除后尝试读取
 - 删除后列出桶中所有对象



弹性计算云EC2

- 目标
 - 向用户提供弹性的计算资源
- 特性
 - 灵活性：允许用户对运行的实例类型、数量自行配置，选择实例运行的地理位置，随时改变实例的使用数量
 - 低成本：按小时收费，不需要购买硬件设备
 - 安全性：提供了基于密钥对的**SSH**方式访问、可配置的防火墙机制等安全措施，允许用户对应用程序进行监控
 - 易用性：用户可以利用模块自由构建应用程序，**EC2**会自动对服务请求进行负载均衡
 - 容错性：提供弹性**IP**的机制，在故障发生时尽可能保证用户服务的稳定



弹性计算云EC2（续）

- Amazon Machine Image (AMI)

- 相当于PC中的操作系统（可将用户的应用程序、配置等一起打包）
- 类型：公共AMI，私有AMI，付费AMI，共享AMI

- 实例（Instance）

- 相当于主机，提供计算能力
- 自身携带一个存储模块，临时存放用户数据

实例重启时，用户数据会保留
出现故障或实例终止，用户数据将消失

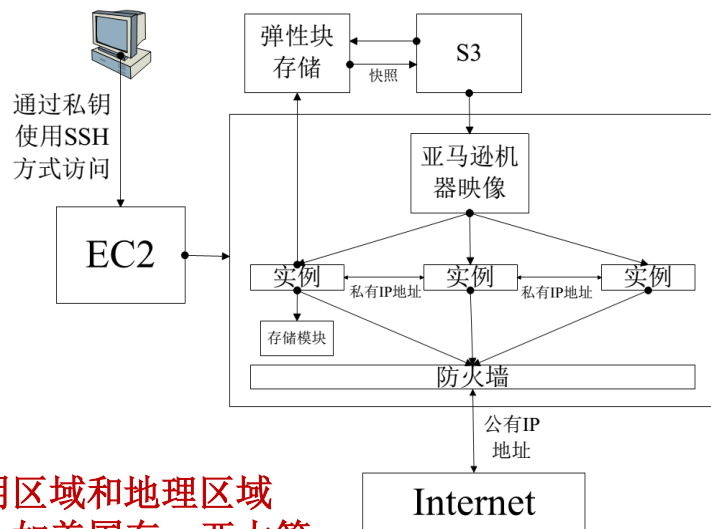
- Elastic Block Store (EBS)

- 长期保存或者存储比较重要的数据，直至用户删除

- 通信机制

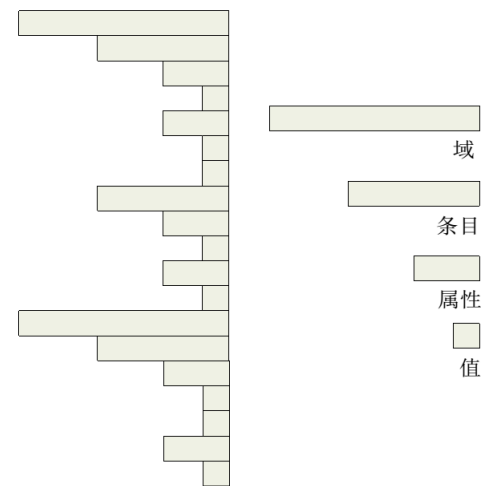
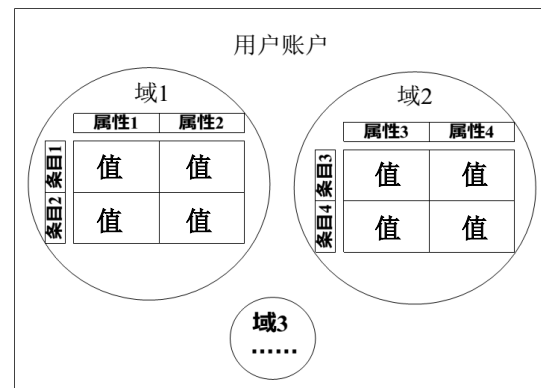
- 公共IP地址：负责和外界进行通信
- 私有IP地址：用于实例间通信
- 弹性IP地址：与用户帐号绑定，可在实例出现故障时将弹性IP地址重新映射到一个新的实例

用户最好将多个实例分布在不同的可用区域和地理区域
地理区域：按照实际地理位置划分的，如美国东、亚太等
可用区域：有独立的供电系统和冷却系统，一般指一个数据中心



简单数据库服务SimpleDB

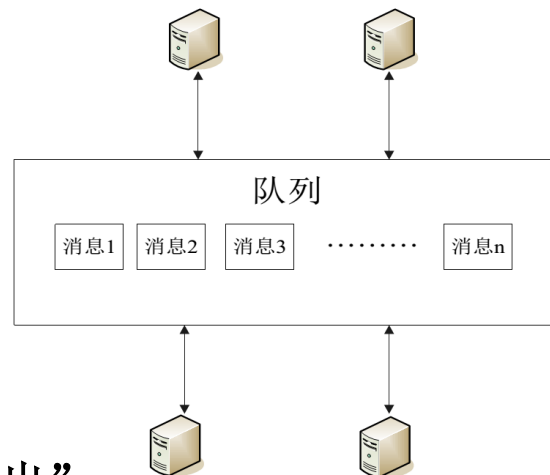
- 目的
 - 存储结构化数据，提供查找、删除等基本的数据库功能
 - **vs S3**: S3存储大型的非结构化的数据
 - **vs 关系数据库**: 非表结构，功能差异
- 数据都以**UTF-8**编码的字符串存储
 - 查询时按照词典顺序
- 基本结构：树状结构
 - 域：数据库操作的基本单位
 - 查询只在一个域内进行，域间操作不允许
 - 条目：域内命名唯一
 - 不需要事先定义模式
 - 属性：条目的特征
 - 值：允许多值属性
 - 每个属性值的大小不能超过**1KB**
 - **SimpleDB**存放指针，指向存放在**S3**中的较大的数据
- 采用最终一致性数据模型



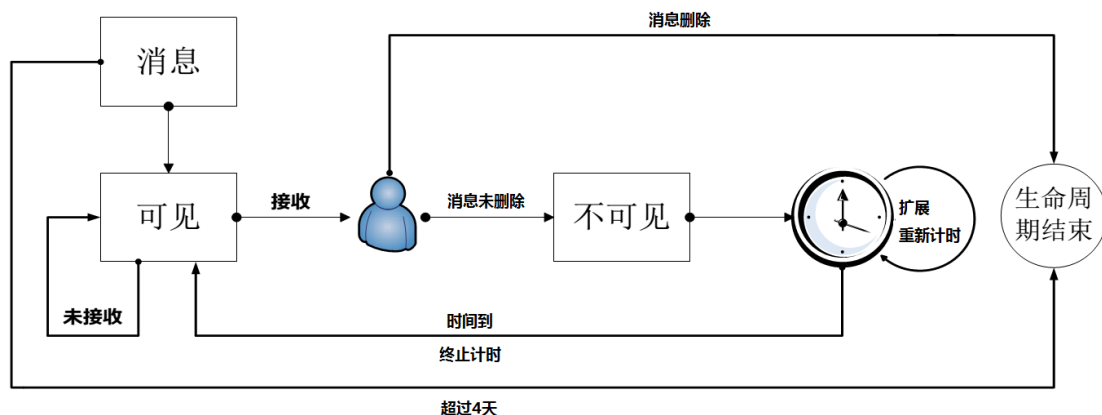
简单队列服务SQS

- 目标
 - 解决低耦合系统间的通信问题
 - 支持分布式计算机系统之间的工作流
- 队列
 - 存放消息的容器
- 消息
 - 一定格式的文本，不超过**8KB**，尽可能“先进先出”
 - 被冗余存储，采用基于加权随机分布的消息取样

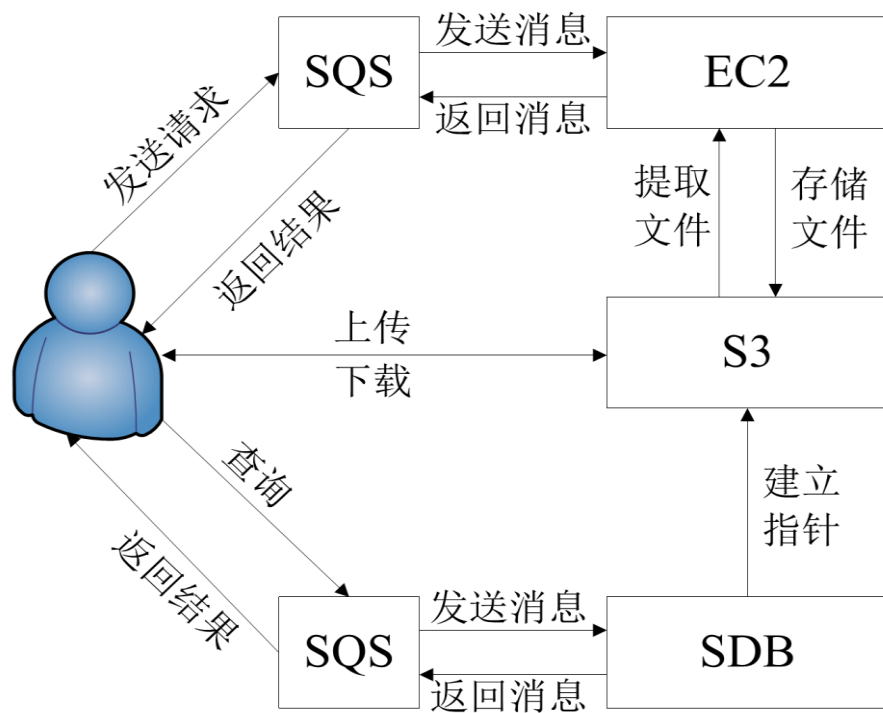
队列的数量是任意的，但名称必须唯一



用户查询消息时，会随机选择部分服务器，并返回这些服务器上所保存的所查询队列中消息的副本

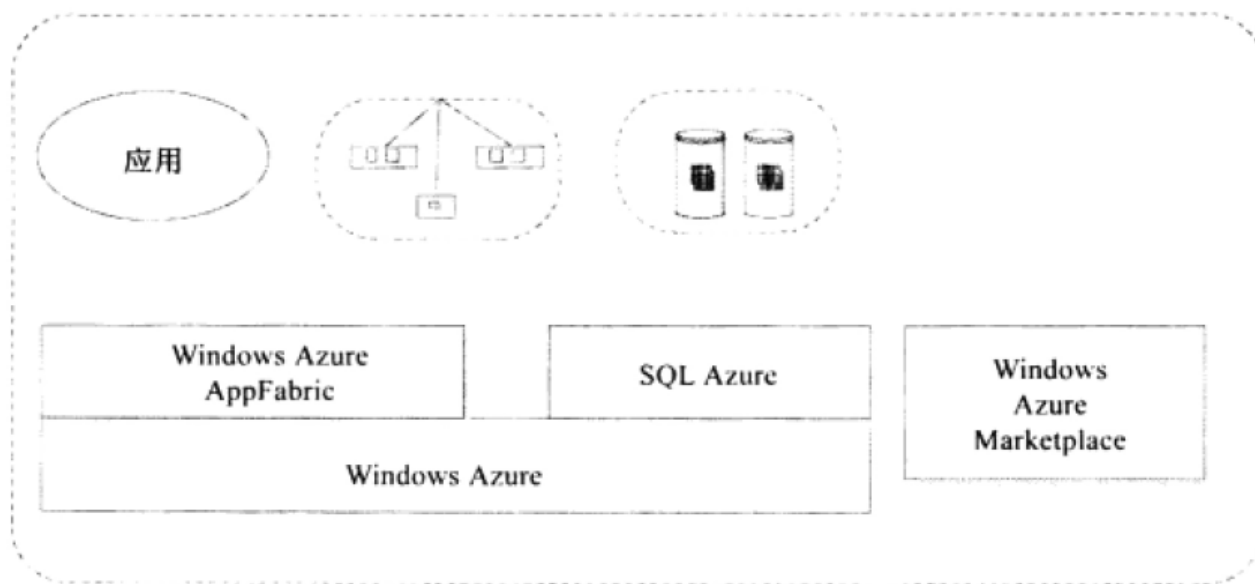


Amazon AWS小结



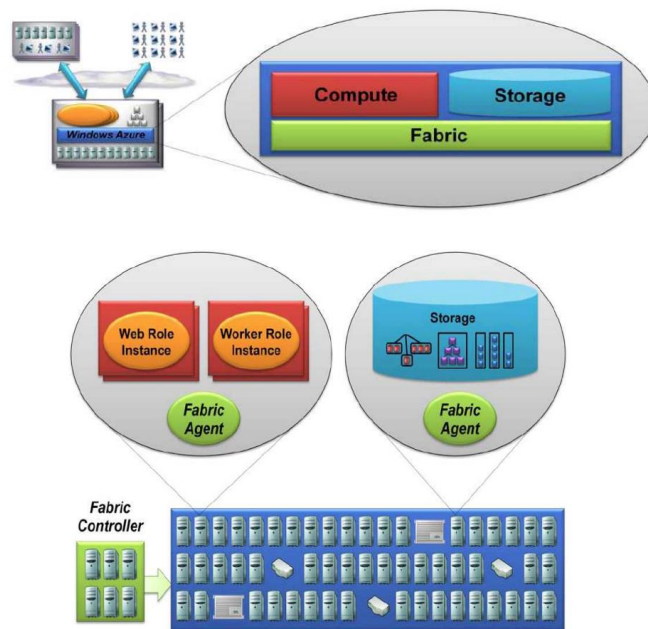
微软Azure云平台

- 基本思路
 - “云+端” 模式
 - 软件+服务（S+S）战略
- 体系结构




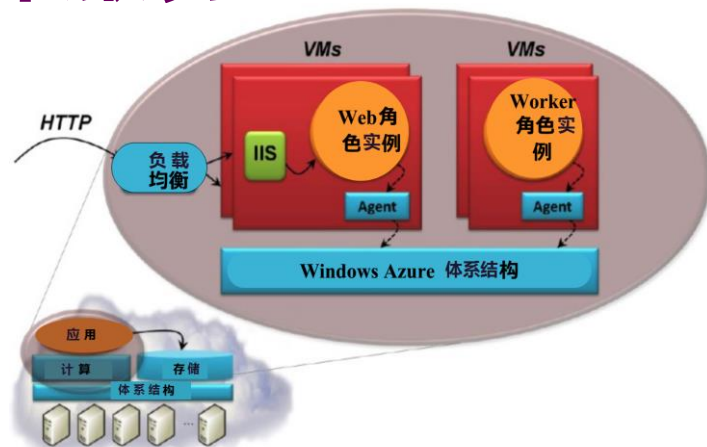
云端操作系统Windows Azure

- 作用
 - 可以在微软数据中心上运行应用程序和存储应用程序数据
- 组成部分
 - 计算服务，存储服务，**Fabric**控制器，内容分发网络**CDN**，**Windows Azure Connect**
- 机制
 - 通过**Fabric**将机器的处理能力整合为一体
 - **Fabric**
 - 由位于数据中心的大量机器组成（5-7台一组）
 - 由“**Fabric**控制器”软件来管理
 - 依赖于应用所带的**XML**格式配置文件



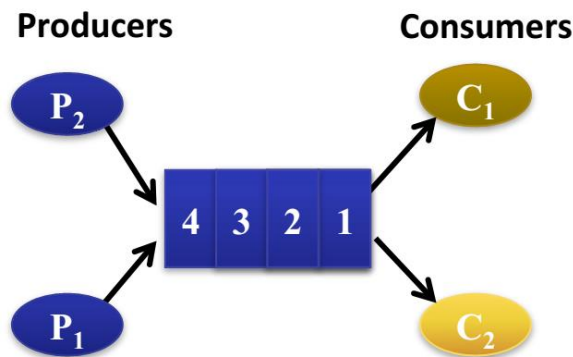
Windows Azure: 计算服务

- 目标
 - 支持有大量并行用户的应用程序
 - 运行机制
 - 每个应用程序运行多个实例
 - 每个实例运行自己的虚拟机
 - 每个虚拟机运行一个**64位的Windows Server 2008**
 - 角色（role）
 - **Web Role**: 提供**Web**服务的角色（支持 **HTTP/HTTPS**协议，提供**WCF**服务）
 - **Worker Role**: 在后台运行的应用程序（可以在后台访问任何网络资源、数据源并进行操作）
- 
- The diagram illustrates the Windows Azure architecture. At the top, an 'HTTP' request enters a '负载均衡' (Load Balancing) box. This box directs traffic to two main role instances: 'Web角色实例' (Web Role Instance) and 'Worker角色实例' (Worker Role Instance). Each role instance contains an 'IIS' (Internet Information Services) component and an 'Agent' component. These instances are part of the 'Windows Azure 体系结构' (Windows Azure Architecture). Below the role instances, a cloud represents the '应用' (Application) layer, which includes '计算' (Compute) and '存储' (Storage) components, all within a '体系结构' (Architecture) layer. At the bottom, several server icons represent the physical infrastructure.



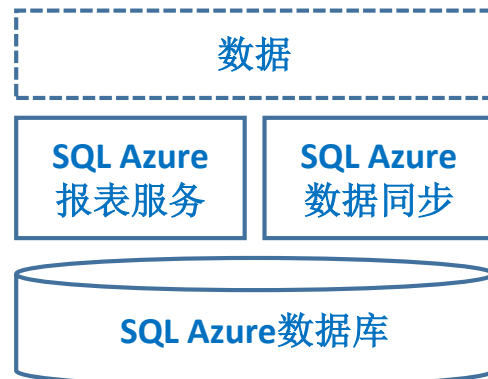
Windows Azure: 存储服务

- 三种数据模型
 - **Blob**: 提供二进制的大块数据存储服务
 - **Table**: 提供结构化的存储
 - **Queue**: 提供可靠的消息存储和消息服务
 - 可以反映后台处理节点的负载大小
 - 可以使用不同技术和编程语言来实现程序的不同部分
 - 采用缓存机制来处理突发流量和应用组件失效



SQL Azure

- 目标
 - 基于**SQL Server**技术构建
 - 提供关系型数据库存储服务
- 数据模型
 - 数据中心 **Authority -> Container -> Entity**
- 功能
 - **SQL Azure数据库**：使本地应用和云应用可以在微软数据中心上存储数据
 - **SQL Azure报表服务**：在**SQL Azure**存储的数据中创建标准的**SSRS**（**SQL Server Reporting Service**）报表
 - **SQL Azure数据同步**：同步**SQL Azure**数据库与本地**SQL Server**数据库中的数据，或在微软数据中心之间同步不同**SQL Azure**数据库
 - **Hub-and-Spoke模型**：所有的变化会先被复制到**SQL Azure**数据库“hub”上，然后再传送到其它“spoke”上。



Windows Azure AppFabric

- 作用
 - 为本地应用和云应用提供分布式的基础架构服务，使本地应用于云应用进行安全联接和信息传递
- 功能
 - 互联网服务总线：简化云应用的公开终端的访问
 - 访问控制：简化数字身份认证
 - 高速缓存：提升应用对同一数据重复访问的效率



阿里巴巴开放数据处理服务

- 开放数据处理服务**ODPS**（**Open Data Processing Service**）

- 基于飞天平台实现，用于海量数据存储和计算的服务
- 应用



- 淘宝、天猫双十一：每秒创建订单**8万**，每小时处理数据**17PB**，当天交易总额**571亿**（**2014年**）



- 余额宝：客户数量接近**1.5亿**，规模突破**5000亿元**，每秒处理**11000笔**请求



- 蚂蚁金服：1秒审批，小微贷款坏账率**<1%**（截至**2015年1月**），累计贷款**30万家**

需求和设想

- 2009年的状态

存

- IOE + Greenplum + Hadoop +

- 存储昂贵，可扩展性差

- 数据孤岛

通

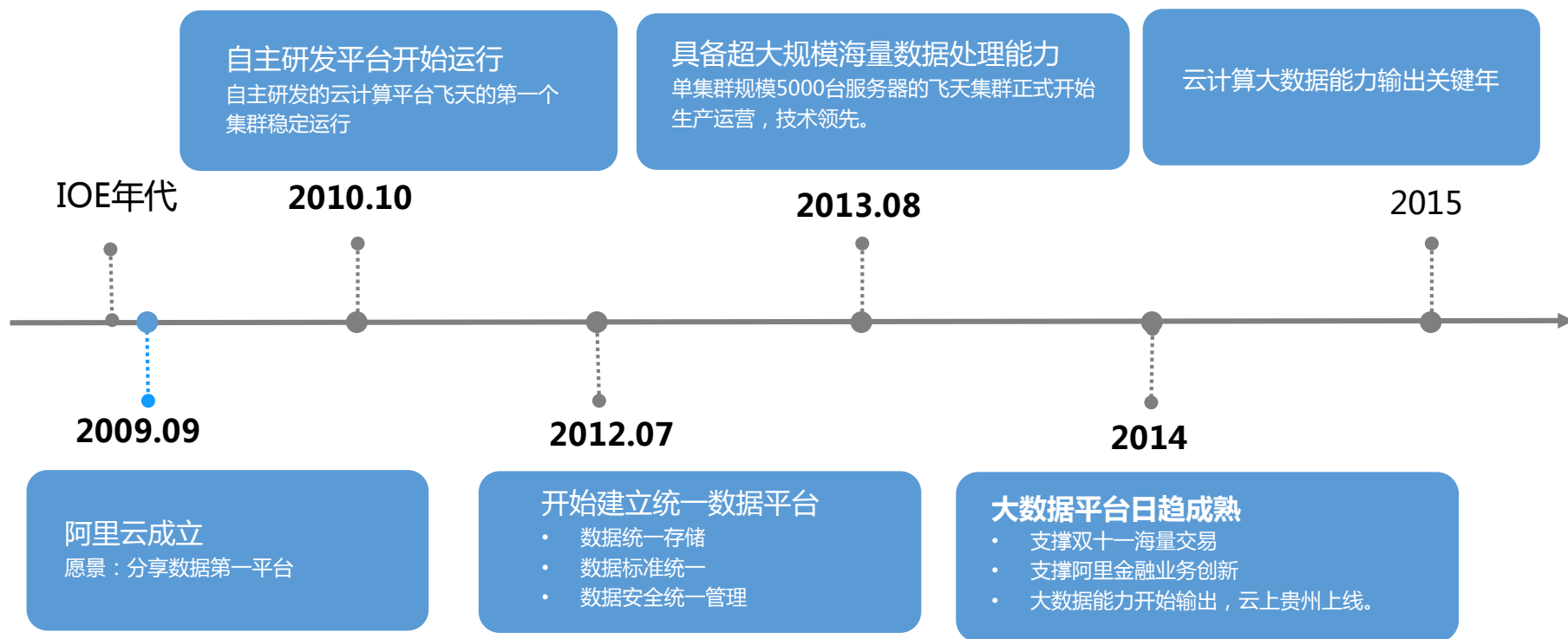
- 各业务部门的数据散落在多个集群，彼此之间数据不通，数据共享太难

- 数据重复建设

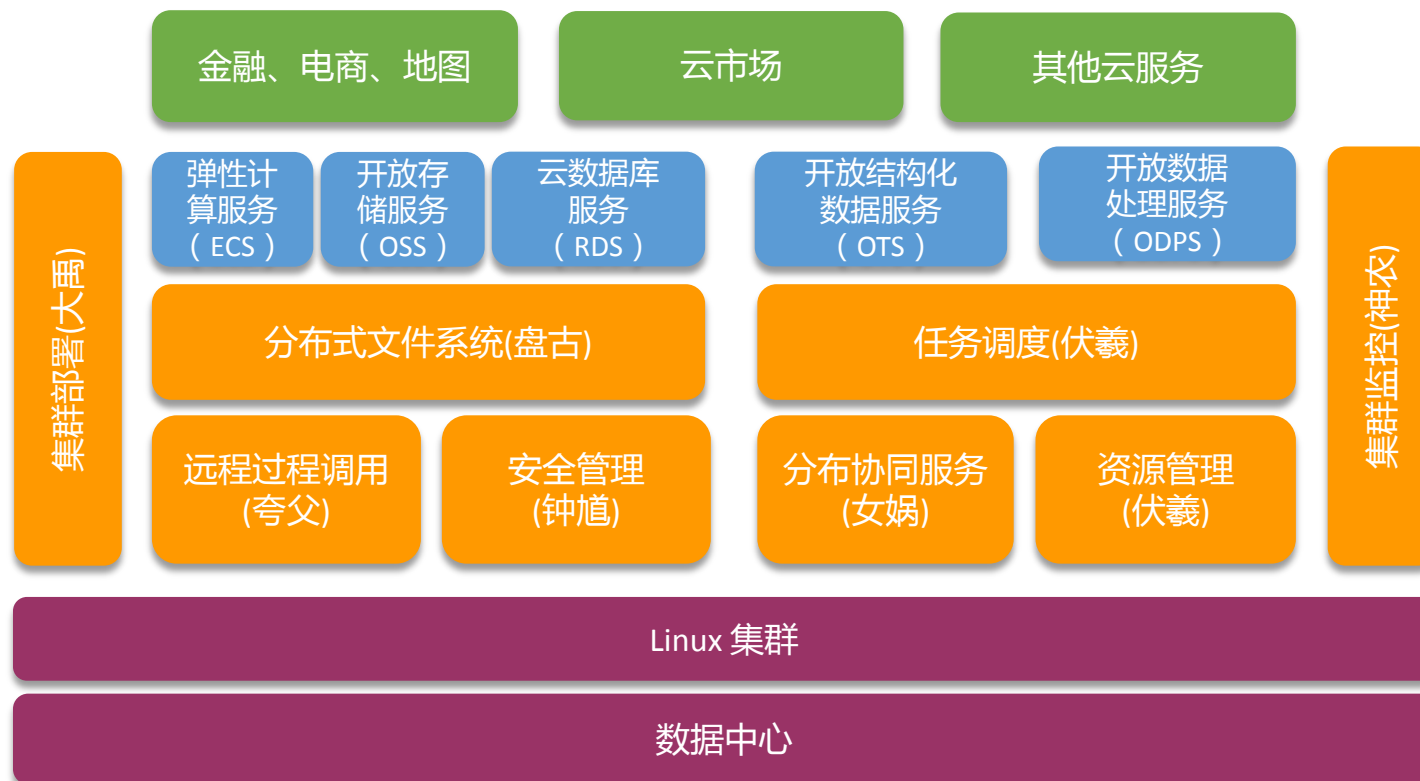
用

- 由于数据不集中，导致数据被重复存储和计算，光淘宝商品类目表就有**70**多张.....

阿里大数据平台发展历程

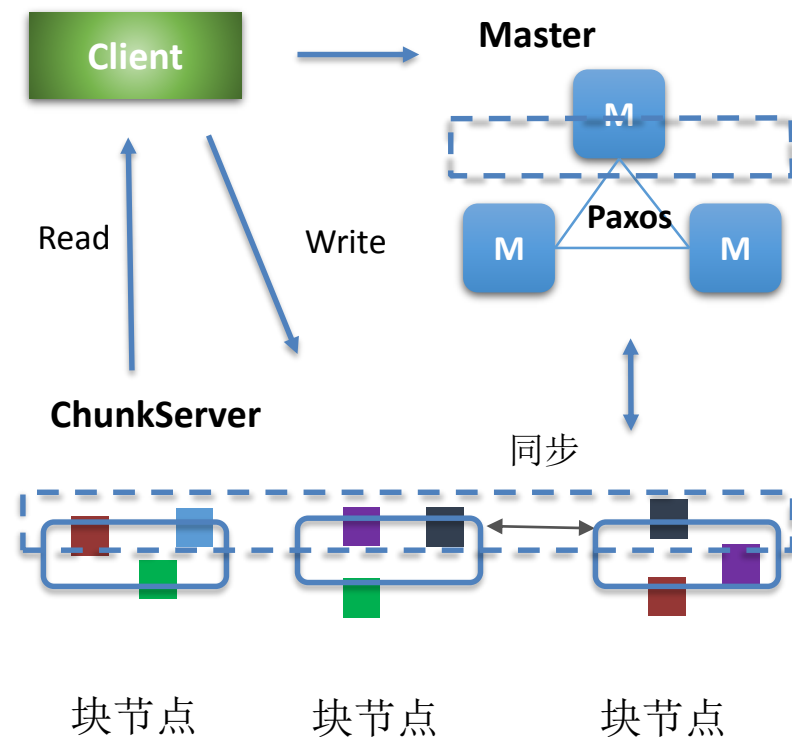


飞天开放平台



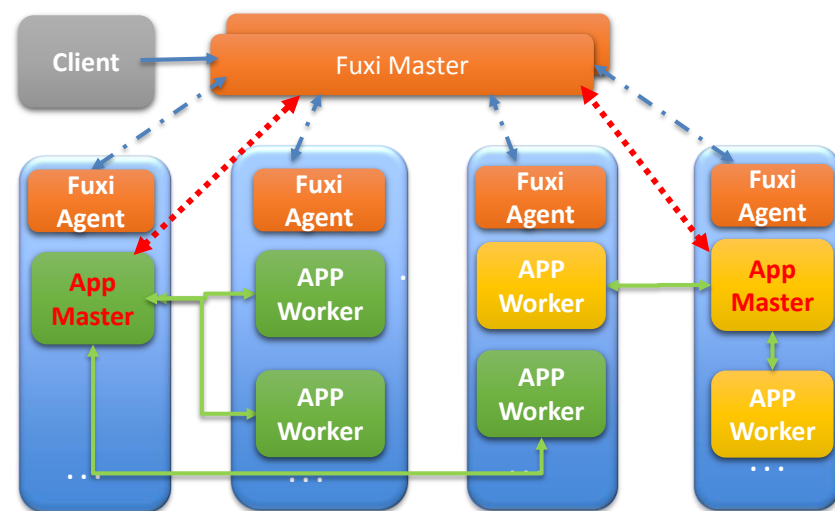
分布式文件系统（盘古）

- **Master/Slave架构**
- 稳定性增强
 - 基于**Paxos**的多**Master**架构
 - 自动故障恢复时间小于**1分钟**
 - 透明热升级
- 多租户增强
 - **Capability**与目录配额
 - 流控、优先级与公平性
 - 离线/在线混布
- 性能增强
 - 混合存储，原生**RaidFile**支持
 - 锁优化，读写分离
- 规模增强
 - 文件数无限制，单集群 **> 5K**

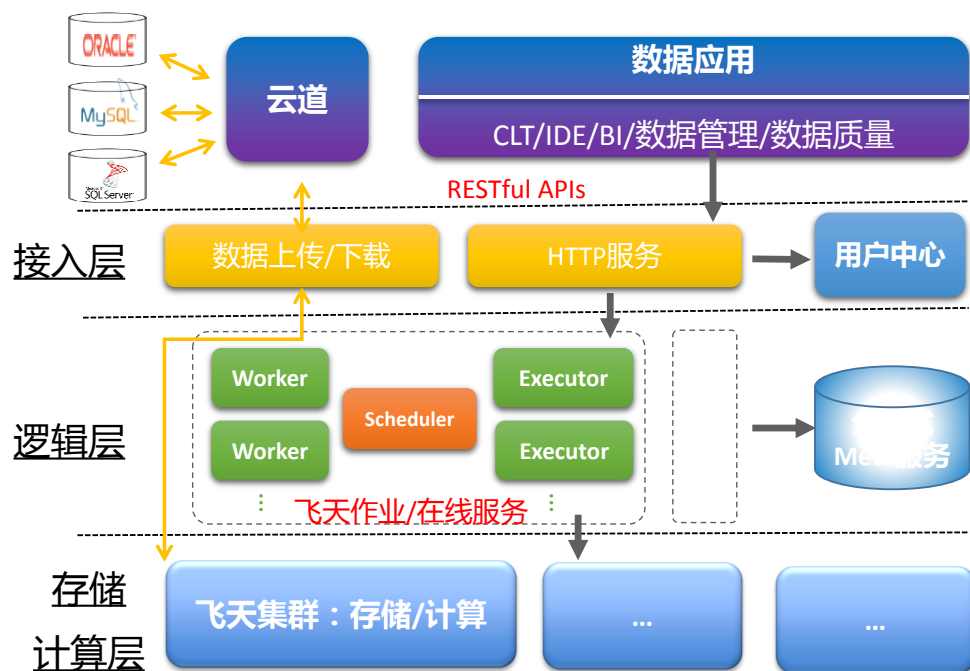


资源管理与任务调度（伏羲）

- 稳定性增强
 - Fuxi Master HA
 - AppMaster Failover
- 多租户增强
 - 多维度资源：CPU/内存...
 - 配额组管理，弹性（min/max）
 - Cgroup隔离
 - 进程沙箱
 - 离线/在线混布
- 规模增强
 - 增量调度，10K+规模
- 编程模型：Job/Service



ODPS



- 单一集群规模可以达到**10000+**服务器（保持**80%**线性扩展）
- 单个**ODPS**部署可以支持**100**万服务器以上，无限制（线性扩展略差），支持同城、异地多数据中心模式
- **10000+**用户数，**1000+**项目应用、**100+**部门（多租户）
- **100**万以上作业（目前单日平均提交任务），**20000**以上并发作业

ODPS的丰富功能

- 海量数据存储
 - 多份拷贝，突破单一集群限制，增加存储利用率
- 丰富的计算工具和编程模型
 - **SQL**: 语法兼容**Hive**，函数语义与传统关系数据库更兼容
 - 流计算
 - **MapReduce**
 - 图计算: **PageRank**, **K-均值聚类**, 金融风控,
 - 算法平台: **SVD**分解, 逻辑回归, 随机森林, 朴素贝叶斯,



鲍尔默之问

- 利用**Web**软件收发电子邮件、处理文档和电子表格、进行协作很方便吗？
- 高速宽带连接会像断言的那样普及和可靠吗？
- 企业、大学、消费者会让某家公司保存他们的资料吗？



谢 谢