



《大数据技术及应用》

第六章 云数据库



提纲

6.1 云数据库概述

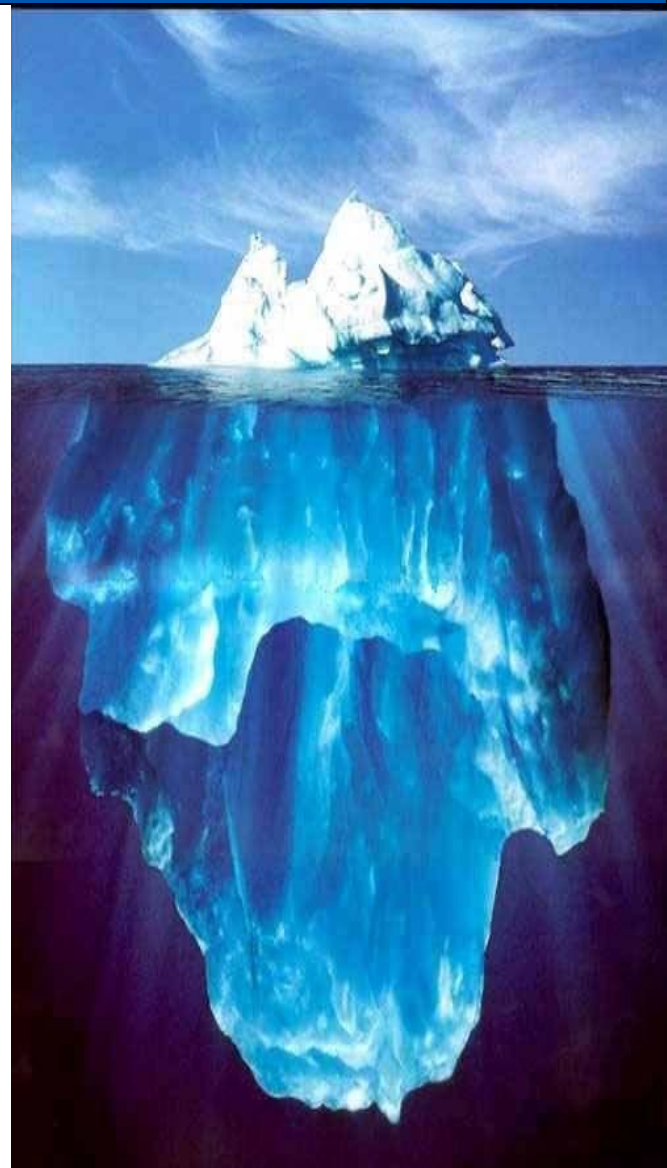
6.2 云数据库产品

6.3 云数据库系统架构

6.4 Amazon AWS和云数据库

6.5 微软云数据库SQL Azure

6.6 云数据库实践



6.1 云数据库概述

6.1.1 云计算是云数据库兴起的基础

6.1.2 云数据库概念

6.1.3 云数据库的特性

6.1.4 云数据库是个性化数据存储需求的理想选择

6.1.5 云数据库与其他数据库的关系

6.1.1 云计算是云数据库兴起的基础



云计算示意图

云计算概念

•通过整合、管理、调配分布在网络各处的计算资源，通过互联网以统一界面，同时向大量的用户提供服务

云计算特点

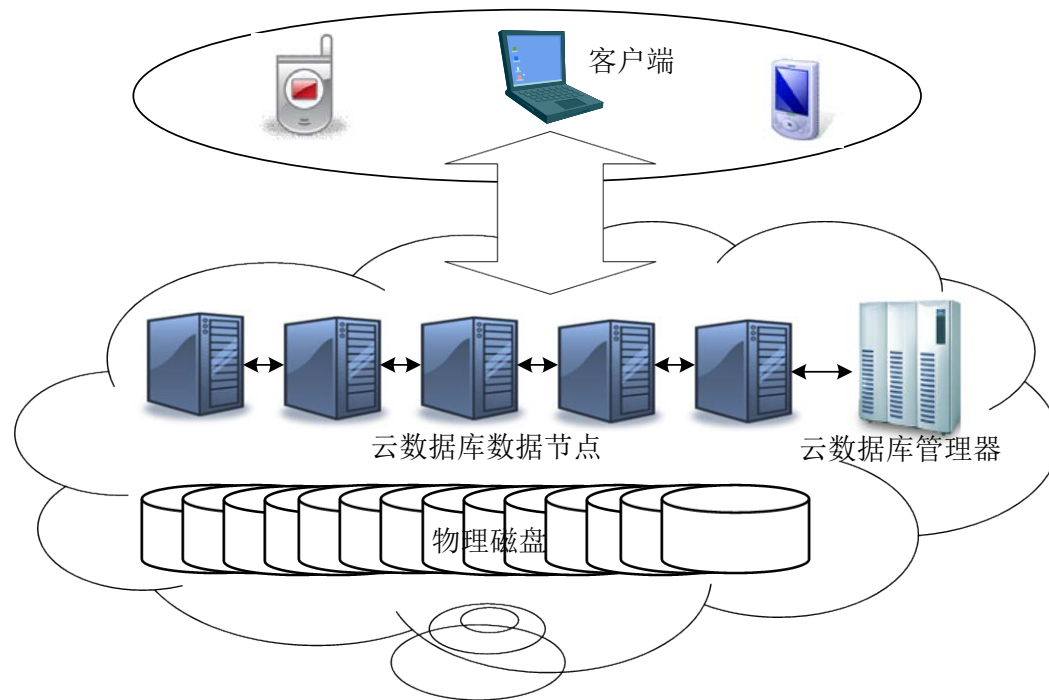


云，优化
计算资源

云计算八大优势



6.1.2云数据库概念



云数据库是部署和虚拟化在云计算环境中的数据库。云数据库是在云计算的大背景下发展起来的一种新兴的共享基础架构的方法，它极大地增强了数据库的存储能力，消除了人员、硬件、软件的重复配置，让软、硬件升级变得更加容易。云数据库具有高可扩展性、高可用性、采用多租形式和支持资源有效分发等特点。

6.1.3云数据库的特性

云数据库具有以下特性：

- (1) 动态可扩展
- (2) 高可用性
- (3) 较低的使用代价
- (4) 易用性
- (5) 高性能
- (6) 免维护
- (7) 安全

腾讯云数据库和自建数据库的比较

	自建数据库	腾讯云数据库
数据安全性	开发者自行解决，成本高昂	15种类型备份数据，保证数据安全
服务可用性		99.99%高可靠性
数据备份		0花费，系统自动多时间点数据备份
维护成本		0成本，专业团队7x24小时帮助维护
实例扩容		一键式直接扩容，安全可靠
资源利用率		按需申请，资源利用率高达99.9%
技术支持		专业团队一对一指导、QQ远程协助开发者

6.1.4 云数据库是个性化数据存储需求的理想选择

企业类型不同，对于存储的需求也千差万别，而云数据库可以很好地满足不同企业的个性化存储需求：

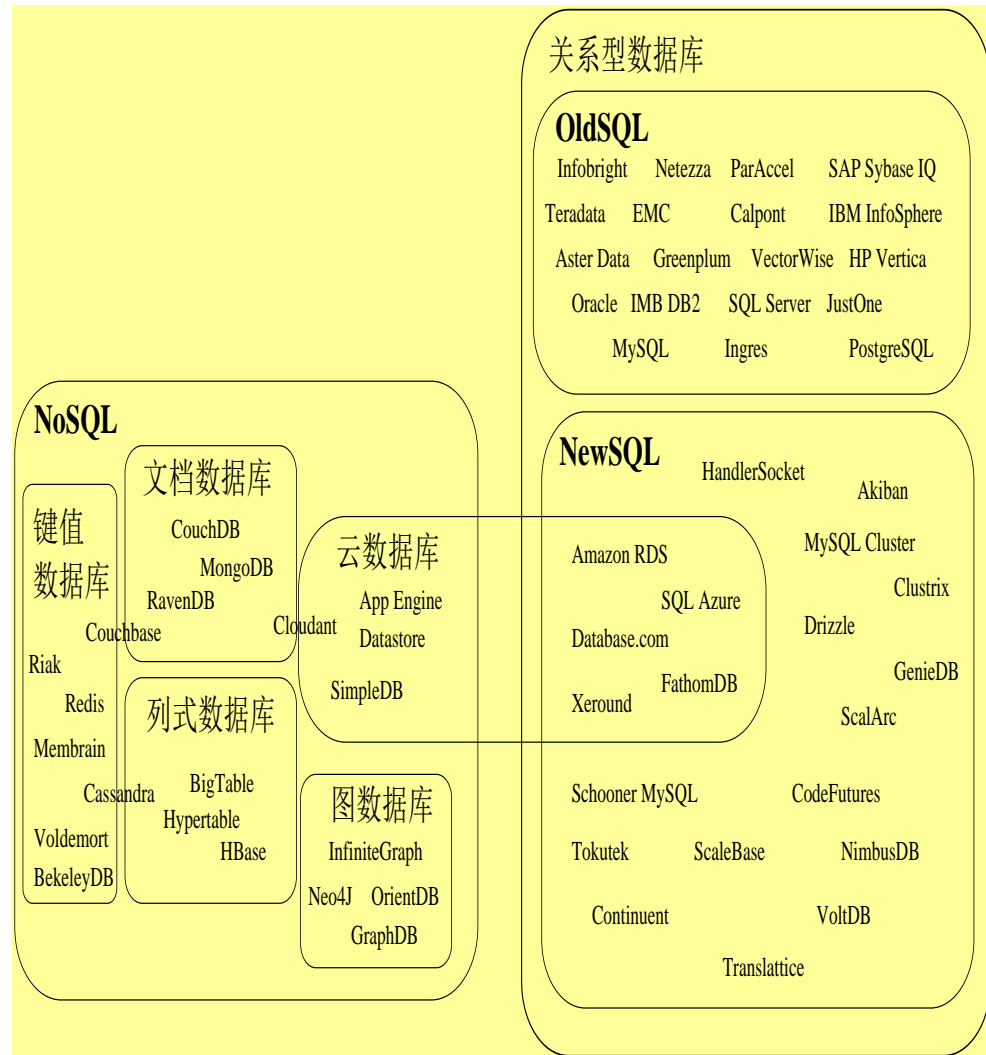
- 首先，云数据库可以满足大企业的海量数据存储需求
- 其次，云数据库可以满足中小企业的低成本数据存储需求
- 另外，云数据库可以满足企业动态变化的数据存储需求

到底选择自建数据库还是选择云数据库，取决于企业自身的具体需求

- 对于一些大型企业，目前通常采用自建数据库
- 对于一些财力有限的中小企业而言，IT预算比较有限，云数据库这种前期零投入、后期免维护的数据库服务，可以很好满足它们的需求

6.1.5云数据库与其他数据库的关系

- 从数据模型的角度来说，云数据库并非一种全新的数据库技术，而只是以服务的方式提供数据库功能
- 云数据库并没有专属于自己的数据模型，云数据库所采用的数据模型可以是关系数据库所使用的关系模型（微软的**SQL Azure**云数据库、阿里云**RDS**都采用了关系模型），也可以是**NoSQL**数据库所使用的非关系模型（**Amazon Dynamo**云数据库采用的是“键/值”存储）
- 同一个公司也可能提供采用不同数据模型的多种云数据库服务
- 许多公司在开发云数据库时，后端数据库都是直接使用现有的各种关系数据库或**NoSQL**数据库产品



6.2 云数据库产品

6.2.1 云数据库厂商概述

6.2.2 Amazon的云数据库产品

6.2.3 Google的云数据库产品

6.2.4 Microsoft的云数据库产品

6.2.5 其他云数据库产品

6.2.1 云数据库厂商概述

云数据库产品

企业	产品
Amazon	Dynamo、SimpleDB、RDS
Google	Google Cloud SQL
Microsoft	Microsoft SQL Azure
Oracle	Oracle Cloud
Yahoo!	PNUTS
Vertica	Analytic Database v3.0 for the Cloud
EnerpriseDB	Postgres Plus in the Cloud
阿里	阿里云RDS
百度	百度云数据库
腾讯	腾讯云数据库

6.2.2 Amazon的云数据库产品

➤ **Amazon**是云数据库市场的先行者。**Amazon**除了提供著名的**S3**存储服务和**EC2**计算服务以外，还提供基于云的数据库服务：

— **Amazon RDS**：云中的关系数据库

— **Amazon SimpleDB**：云中的键值数据库

— **Amazon DynamoDB**：云中的**NoSQL**数据库

— **Amazon Redshift**：云中的数据仓库

— **Amazon ElastiCache**：云中的分布式内存缓存

6.2.3 Google的云数据库产品

- **Google Cloud SQL**是谷歌公司推出的基于**MySQL**的云数据库
- 使用**Cloud SQL**，所有的事务都在云中，并由谷歌管理，用户不需要配置或者排查错误
- 谷歌还提供导入或导出服务，方便用户将数据库带进或带出云
- 谷歌使用用户非常熟悉的**MySQL**，带有**JDBC**支持（适用于基于**Java**的**App Engine**应用）和**DB-API**支持（适用于基于**Python**的**App Engine**应用）的传统**MySQL**数据库环境，因此，多数应用程序不需过多调试即可运行，数据格式对于大多数开发者和管理员来说也是非常熟悉的
- **Google Cloud SQL**还有一个好处就是与**Google App Engine**集成

6.2.4 Microsoft的云数据库产品

SQL Azure具有以下特性：

- 属于关系型数据库：支持使用TSQL（Transact Structured Query Language）来管理、创建和操作云数据库
- 支持存储过程：它的数据类型、存储过程和传统的SQL Server具有很大的相似性，因此，应用可以在本地进行开发，然后部署到云平台上
- 支持大量数据类型：包含了几乎所有典型的SQL Server 2008的数据类型
- 支持云中的事务：支持局部事务，但是不支持分布式事务

6.3 云数据库系统架构

6.3.1 UMP系统概述

6.3.2 UMP系统架构

6.3.3 UMP系统功能

6.3.1 UMP系统概述

UMP系统是低成本和高性能的MySQL云数据库方案。

总的来说，**UMP**系统架构设计遵循了以下原则：

- 保持单一的系统对外入口，并且为系统内部维护单一的资源池
- 消除单点故障，保证服务的高可用性
- 保证系统具有良好的可伸缩，能够动态地增加、删减计算与存储节点
- 保证分配给用户的资源也是弹性可伸缩的，资源之间相互隔离，确保应用和数据安全

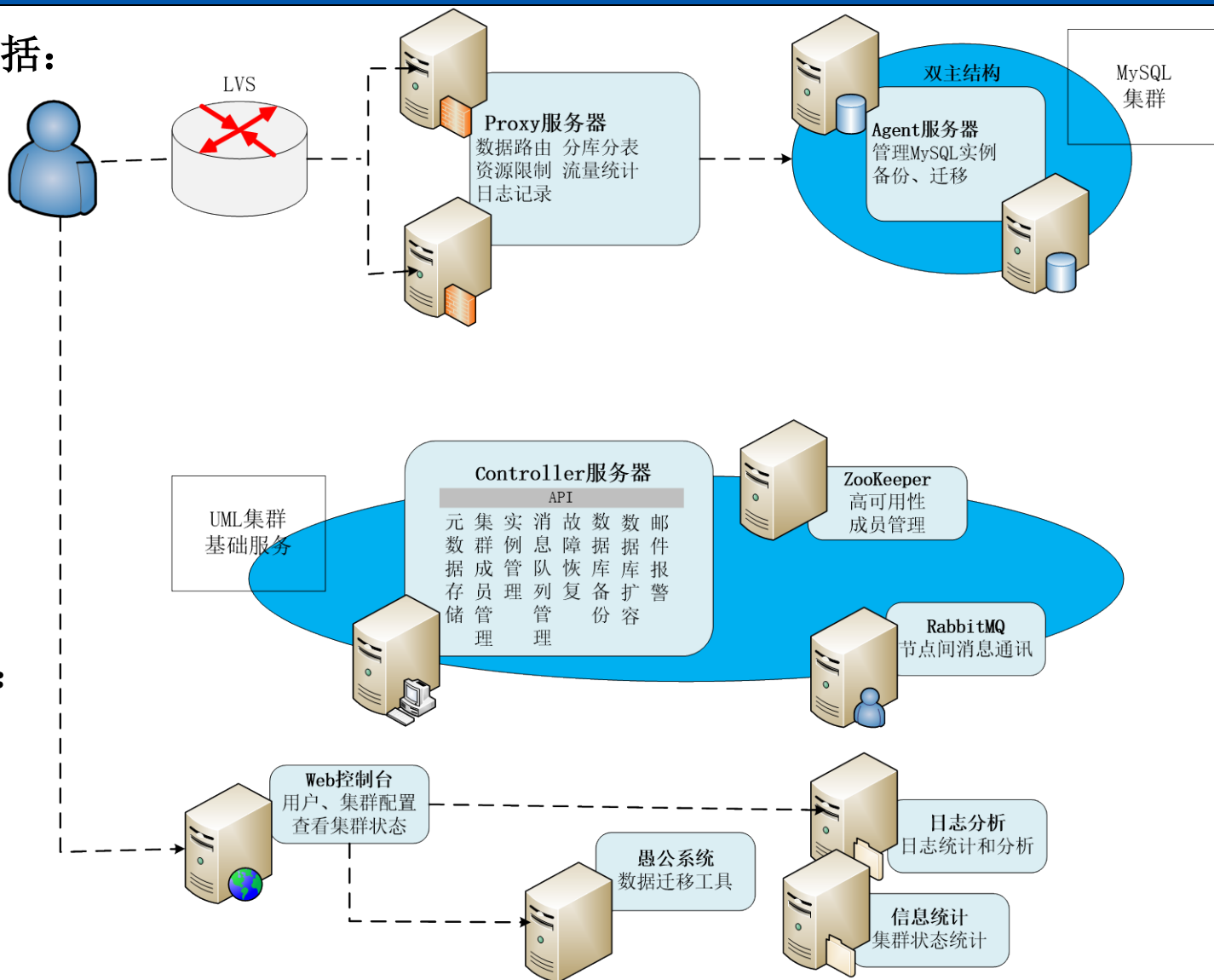
6.3.2 UMP系统架构

UMP系统中的角色包括:

- Controller服务器
- Proxy服务器
- Agent服务器
- Web控制台
- 日志分析服务器
- 信息统计服务器
- 愚公系统

依赖的开源组件包括:

- Mnesia
- LVS
- RabbitMQ
- ZooKeeper



6.3.2 UMP系统架构

1. Mnesia

- **Mnesia**是一个分布式数据库管理系统
- **Mnesia**支持事务，支持透明的数据分片，利用两阶段锁实现分布式事务，可以线性扩展到至少**50**个节点
- **Mnesia**的数据库模式(**schema**)可在运行时动态重配置，表能被迁移或复制到多个节点来改进容错性
- **Mnesia**的这些特性，使其在开发云数据库时被用来提供分布式数据库服务

6.3.2 UMP系统架构

2. RabbitMQ

- **RabbitMQ**是一个工业级的消息队列产品（功能类似于**IBM**公司的消息队列产品**IBM Websphere MQ**），作为消息传输中间件来使用，可以实现可靠的消息传送
- **UMP**集群中各个节点之间的通信，不需要建立专门的连接，都是通过读写队列消息来实现的

6.3.2 UMP系统架构

3. Zookeeper

Zookeeper是高效和可靠的协同工作系统，提供分布式锁之类的基本服务（比如统一命名服务、状态同步服务、集群管理、分布式应用配置项的管理等），用于构建分布式应用，减轻分布式应用程序所承担的协调任务

在**UMP**系统中，**Zookeeper**主要发挥三个作用：

- 作为全局的配置服务器
- 提供分布式锁（选出一个集群的“总管”）
- 监控所有**MySQL**实例

6.3.2 UMP系统架构

4. LVS

- **LVS(Linux Virtual Server)**即Linux虚拟服务器，是一个虚拟的服务器集群系统
- **UMP**系统借助于**LVS**来实现集群内部的负载均衡
- **LVS**集群采用**IP**负载均衡技术和基于内容请求分发技术
- 调度器是**LVS**集群系统的唯一入口点，调度器具有很好的吞吐率，将请求均衡地转移到不同的服务器上执行，且调度器自动屏蔽掉服务器的故障，从而将一组服务器构成一个高性能的、高可用的虚拟服务器
- 整个服务器集群的结构对客户是透明的，而且无需修改客户端和服务器的程序

6.3.2 UMP系统架构

5. Controller服务器

- **Controller**服务器向**UMP**集群提供各种管理服务，实现集群成员管理、元数据存储、**MySQL**实例管理、故障恢复、备份、迁移、扩容等功能
- **Controller**服务器上运行了一组**Mnesia**分布式数据库服务，其中存储了各种系统元数据，主要包括集群成员、用户的配置和状态信息，以及用户名到后端**MySQL**实例地址的映射关系（或称为“路由表”）等
- 当其它服务器组件需要获取用户数据时，可以向**Controller**服务器发送请求获取数据
- 为了避免单点故障，保证系统的高可用性，**UMP**系统中部署了多台**Controller**服务器，然后，由**Zookeeper**的分布式锁功能来帮助选出一个“总管”，负责各种系统任务的调度和监控

6.3.2 UMP系统架构

6. Web控制台

Web控制台向用户提供系统管理界面

7. Proxy服务器

Proxy服务器向用户提供访问MySQL数据库的服务，它完全实现了MySQL协议，用户可以使用已有的MySQL客户端连接到Proxy服务器，Proxy服务器通过用户名获取到用户的认证信息、资源配额的限制(例如QPS、IOPS (I/O Per Second)、最大连接数等)，以及后台MySQL实例的地址，然后，用户的SQL查询请求会被转发到相应的MySQL实例上。除了数据路由的基本功能外，Proxy服务器中还实现了很多重要的功能，主要包括屏蔽MySQL实例故障、读写分离、分库分表、资源隔离、记录用户访问日志等

6.3.2 UMP系统架构

8. Agent服务器

Agent服务器部署在运行**MySQL**进程的机器上，用来管理每台物理机上的**MySQL**实例，执行主从切换、创建、删除、备份、迁移等操作，同时，还负责收集和分析**MySQL**进程的统计信息、慢查询日志（**Slow Query Log**）和**bin-log**

9. 日志分析服务器

日志分析服务器存储和分析**Proxy**服务器传入的用户访问日志，并支持实时查询一段时间内的慢日志和统计报表

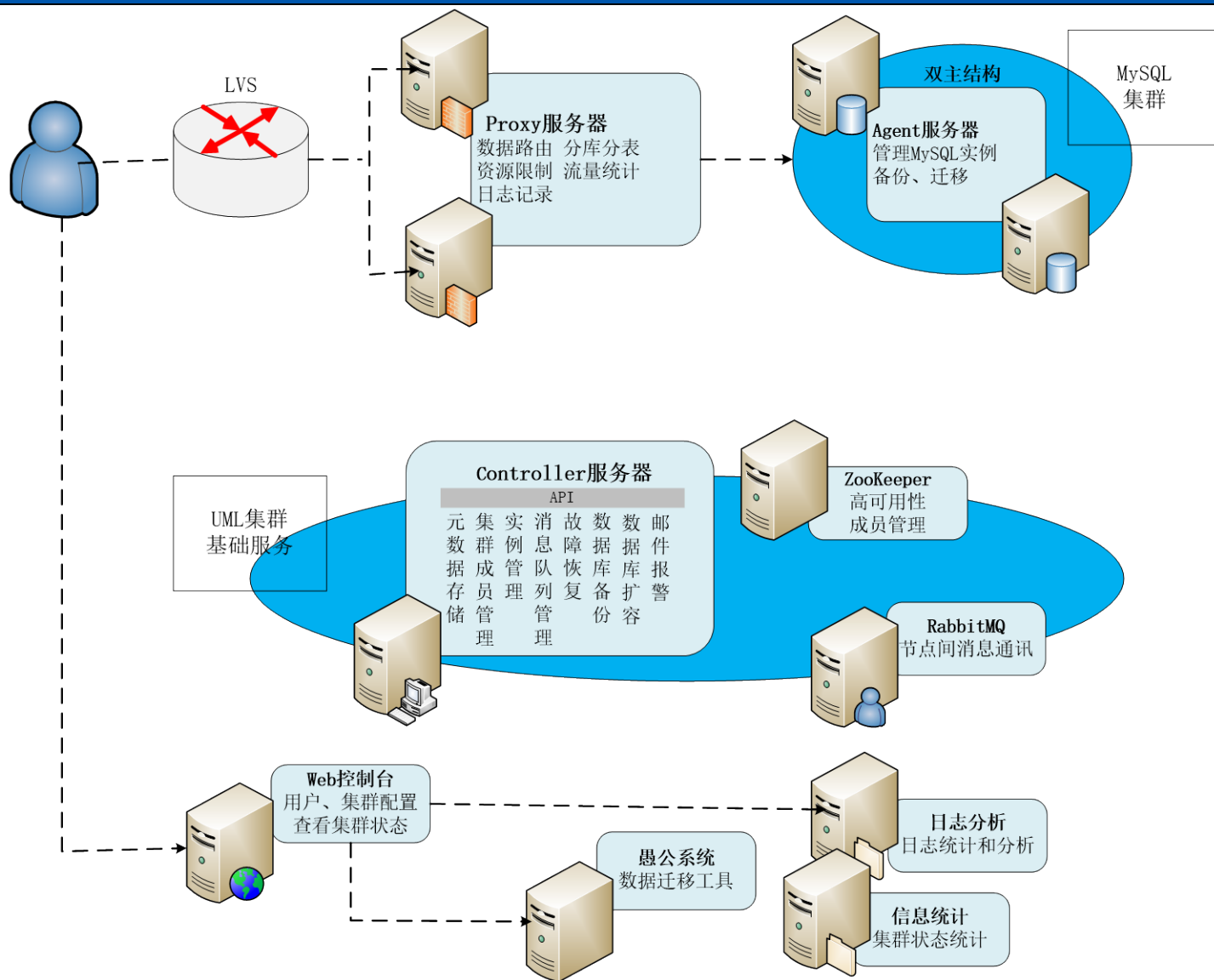
10. 信息统计服务器

信息统计服务器定期将采集到的用户的连接数、**QPS**数值以及**MySQL**实例的进程状态用**RRDtool**进行统计，可以在 **Web**界面上可视化展示统计结果，也可以把统计结果作为今后实现弹性的资源分配和自动化的**MySQL**实例迁移的依据

11. 愚公系统

愚公系统是一个全量复制结合**bin-log**分析进行增量复制的工具，可以实现在不停机的情况下动态扩容、缩容和迁移

6.3.2 UMP系统架构



6.3.3UMP系统功能

UMP系统是构建在一个大的集群之上的，通过多个组件的协同作业，整个系统实现了对用户透明的各种功能：

- 容灾
- 读写分离
- 分库分表
- 资源管理
- 资源调度
- 资源隔离
- 数据安全

6.3.3UMP系统功能

1. 容灾

- 为了实现容灾，**UMP**系统会为每个用户创建两个**MySQL**实例，一个是主库，一个是从库
- 主库和从库的状态是由**Zookeeper**负责维护的
- 主从切换过程如下：
 - Zookeeper**探测到主库故障，通知**Controller**服务器
 - Controller**服务器启动主从切换时，会修改“路由表”，即用户名到后端**MySQL**实例地址的映射关系
 - 把主库标记为不可用
 - 借助于消息中间件**RabbitMQ**通知所有**Proxy**服务器修改用户名到后端**MySQL**实例地址的映射关系
 - 全部过程对用户透明

6.3.3UMP系统功能

1. 容灾

- 宕机后的主库在进行恢复处理后需要再次上线，过程如下：
 - 在主库恢复时，会把从库的更新复制给自己
 - 当主库的数据库状态快要达到和从库一致的状态时，**Controller**服务器就会命令从库停止更新，进入不可写状态，禁止用户写入数据
 - 等到主库更新到和从库完全一致的状态时，**Controller**服务器就会发起主从切换操作，并在路由表中把主库标记为可用状态
 - 通知**Proxy**服务器把写操作切回主库上，用户写操作可以继续执行，之后再把从库修改为可写状态

6.3.3UMP系统功能

2. 读写分离

- 充分利用主从库实现用户读写操作的分离，实现负载均衡
- UMP**系统实现了对于用户透明的读写分离功能，当整个功能被开启时，负责向用户提供访问**MySQL**数据库服务的**Proxy**服务器，就会对用户发起的**SQL**语句进行解析，如果属于写操作，就直接发送到主库，如果是读操作，就会被均衡地发送到主库和从库上执行

6.3.3UMP系统功能

3. 分库分表

UMP支持对用户透明的分库分表（shard / horizontal partition）

当采用分库分表时，系统处理用户查询的过程如下：

- 首先，**Proxy**服务器解析用户**SQL**语句，提取出重写和分发**SQL**语句所需要的信息
- 其次，对**SQL**语句进行重写，得到多个针对相应**MySQL**实例的子语句，然后把子语句分发到对应的**MySQL**实例上执行
- 最后，接收来自各个**MySQL**实例的**SQL**语句执行结果，合并得到最终结果

6.3.3UMP系统功能

4. 资源管理

- **UMP**系统采用资源池机制来管理数据库服务器上的**CPU**、内存、磁盘等计算资源，所有的计算资源都放在资源池内进行统一分配，资源池是为**MySQL**实例分配资源的基本单位
- 整个集群中的所有服务器会根据其机型、所在机房等因素被划分多个资源池，每台服务器会被加入到相应的资源池中
- 对于每个具体**MySQL**实例，管理员会根据应用部署在哪些机房、需要哪些计算资源等因素，为该**MySQL**实例具体指定主库和从库所在的资源池，然后，系统的实例管理服务会本着负载均衡的原则，从资源池中选择负载较轻的服务器来创建**MySQL**实例

6.3.3UMP系统功能

5. 资源调度

- **UMP**系统中有三种规格的用户，分别是数据量和流量比较小的用户、中等规模用户以及需要分库分表的用户
- 多个小规模用户可以共享同一个**MySQL**实例
- 对于中等规模的用户，每个用户独占一个**MySQL**实例
- 对于分库分表的用户，会占有多个独立的**MySQL**实例

6.3.3 UMP系统功能

6.资源隔离

UMP采用的两种资源隔离方式

方法	应用场合	实现方式
用Cgroup限制MySQL进程资源	适用于多个MySQL实例共享同一台物理机的情况	可以对用户的MySQL进程最大可以使用的CPU使用率、内存和IOPS等进行限制
在Proxy服务器端限制QPS	适用于多个用户共享同一个MySQL实例的情况	Controller服务器监测用户的MySQL实例的资源消耗情况，如果明显超出配额，就通知Proxy服务器通过增加延迟的方法去限制用户的QPS，以减少用户对系统资源的消耗

6.3.3 UMP系统功能

7. 数据安全

UMP系统设计了多种机制来保证数据安全：

- **SSL数据库连接：** **SSL(Secure Sockets Layer)**是为网络通信提供安全及数据完整性的一种安全协议，它在传输层对网络连接进行加密。**Proxy**服务器实现了完整的**MySQL**客户端/服务器协议，可以与客户端之间建立**SSL**数据库连接
- **数据访问IP白名单：**可以把允许访问云数据库的**IP**地址放入“白名单”，只有白名单内的**IP**地址才能访问，其他**IP**地址的访问都会被拒绝，从而进一步保证账户安全
- **记录用户操作日志：**用户的所有操作记录都会被记录到日志分析服务器，通过检查用户操作记录，可以发现隐藏的安全漏洞
- **SQL拦截：** **Proxy**服务器可以根据要求拦截多种类型的**SQL**语句，比如全表扫描语句“**select ***”

6.4 Amazon AWS和云数据库

6.4.1 Amazon和云计算的渊源

6.4.2 Amazon AWS

6.4.3 Amazon AWS平台上的云数据库



6.4.1 Amazon和云计算的渊源

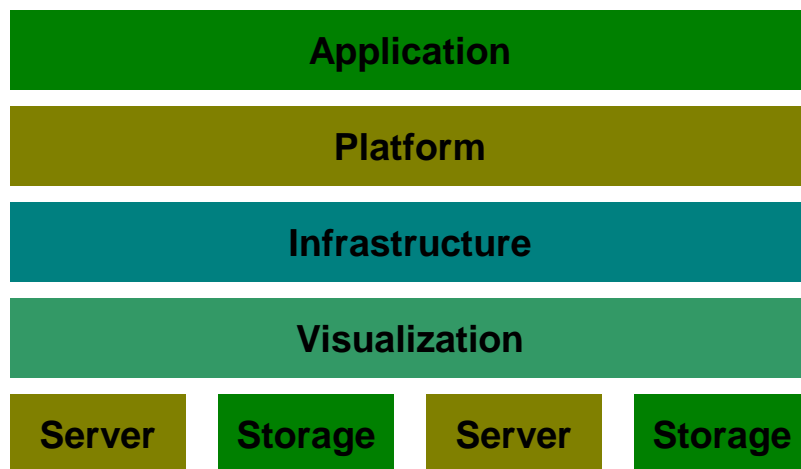


亚马逊云计算平台满**10岁** 彻底改变科技界

2016年3月14日，亚马逊网络服务（**AWS**）十岁了

- **Amazon Web Services**业务相当于紧随其后的**4大**竞争对手的总和
- 亚马逊在全球拥有**12个**区域性数据中心
- **Amazon Web Services**提供的多个亚马逊数据库都在与甲骨文（**Oracle**）激烈竞争，其中**Amazon RDS**有**10万**多个活跃用户
- 亚马逊数据库**Aurora**，是**Amazon Web Services**历史上增长最快的服务

6.4.1 Amazon和云计算的渊源



SaaS

Software as a Service

PaaS

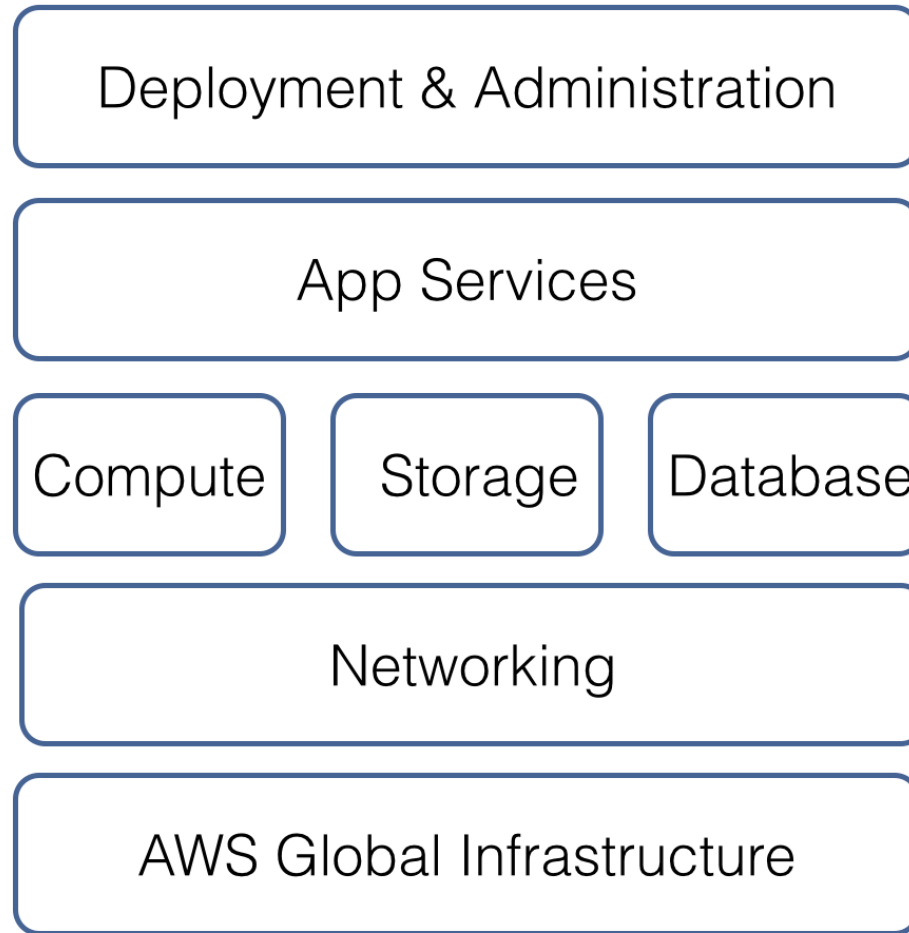
Platform as a Service

IaaS

Infrastructure as a Service

亚马逊的云服务提供了多达几十种服务，涵盖了**IaaS**、**PaaS**、**SaaS**这三层

6.4.2 Amazon AWS

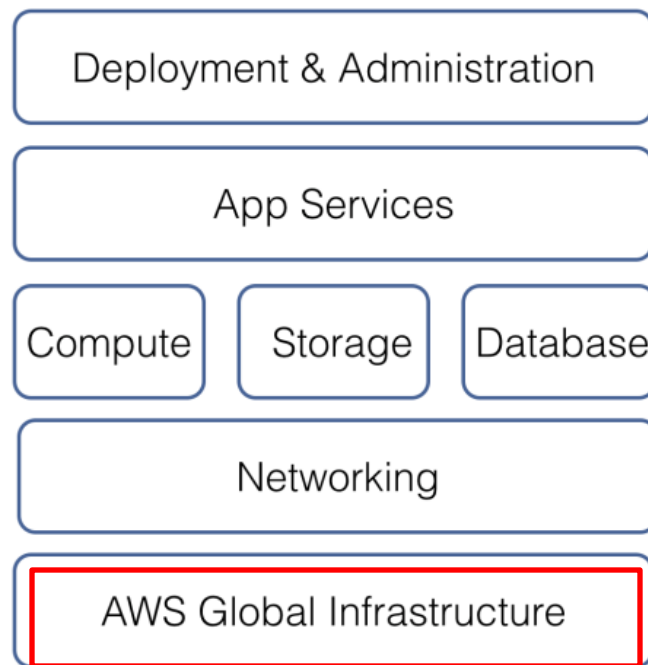


Amazon AWS架构图

6.4.2 Amazon AWS

AWS Global Infrastructure(AWS全局基础设施)

- 在全局基础设施中有**3**个很重要的概念。第一个是**Region**（区域），每个**Region**是相互独立的，自成一套云服务体系，分布在全球各地。目前全球有**10**个**Region**（比如 北京）
- 第二个是**Availability Zone**(可用区)，每个**Region**又由数个可用区组成，每个可用区可以看做一个数据中心，相互之间通过光纤连接
- 第三个是**Edge Locations**（边缘节点）。全球目前有**50**多个边缘节点，是一个内容分发网络（**CDN**, **Content Distribution Network**），可以降低内容分发的延迟，保证终端用户获取资源的速度

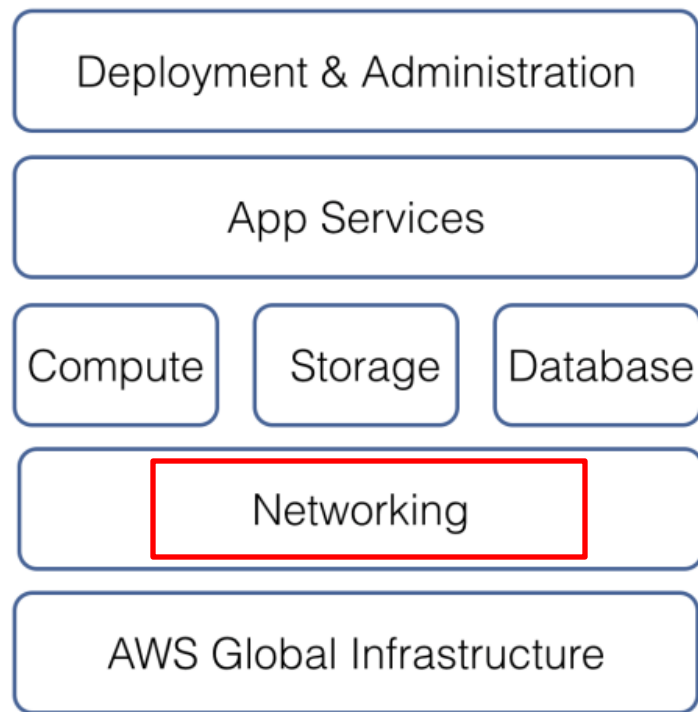


6.4.2 Amazon AWS

Networking（网络）

AWS提供的网络服务主要有：

- **Direct Connect**：支持企业自身的数据中心直接与**AWS**的数据中心直连，充分利用企业现有的资源
- **VPN Connection**：通过**VPN**连接**AWS**，保证数据的安全性
- **Virtual Private Cloud**：私有云，从**AWS**云资源中分一块给你使用，进一步提高安全性
- **Route 53**：亚马逊提供的高可用的可伸缩的云域名解析系统。**Amazon Route 53** 高效地将用户请求连接到 **AWS** 中运行的基础设施，例如 **Amazon EC2** 实例、**Elastic Load Balancing** 负载均衡器或 **Amazon S3** 存储桶



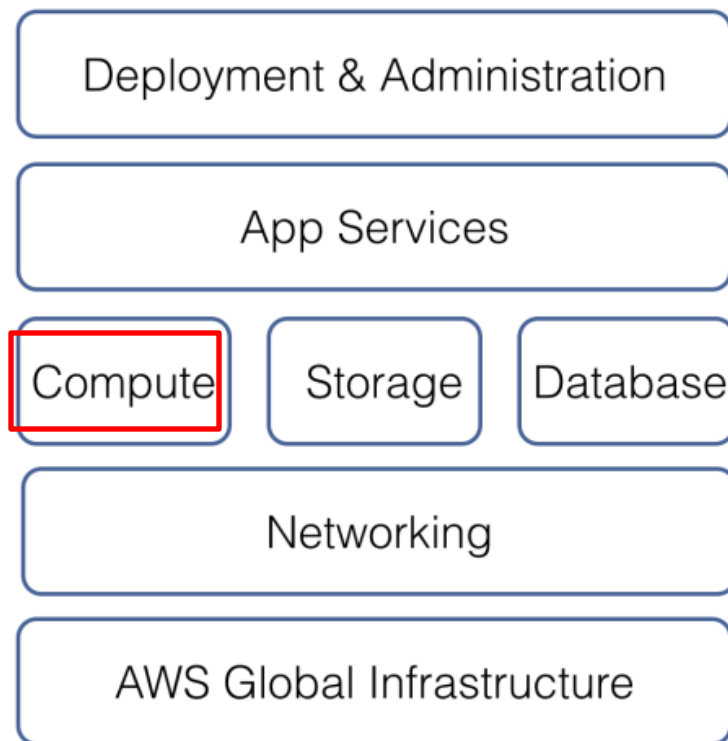
6.4.2 Amazon AWS

Compute（计算）

亚马逊的计算核心，包括了众多的服务

- **EC2: Elastic Compute Cloud**, 亚马逊的虚拟机，支持**Windows**和**Linux**的多个版本，支持**API**创建和销毁，有多种型号可供选择，按需使用。并且有自动扩展功能(**5分钟**即可新建一个虚拟机)，有效解决应用程序性能问题

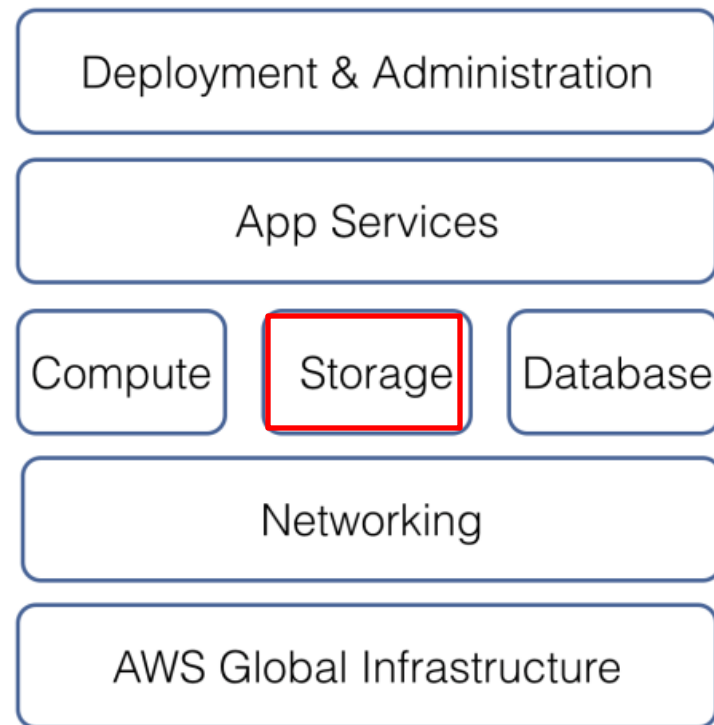
- **ELB: Elastic Load Balancing**, 亚马逊提供的负载均衡器，可以和**EC2**无缝配合使用，横跨多个可用区，可以自动检查实例的健康状况，自动剔除有问题的实例，保证应用程序的高可用性



6.4.2 Amazon AWS

Storage（存储）

- **S3: Simple Storage Service**, 简单存储服务, 是亚马逊对外提供的对象存储服务。不限容量, 单个对象大小可达**5TB**, 可实现高达**99.999999999%**的可用性
- **EBS: Elastic Block Storage**, 专门为**Amazon EC2** 虚拟机设计的弹性块存储服务, **Amazon EBS**可以为**Amazon EC2**的虚拟机创建卷 **volumns**。 **EBS**相当于一个分布式块设备, 可以直接挂载在**EC2**实例上, 用于替代**EC2**实例本地存储, 从而增强**EC2**可靠性
- **Glacier**: 主要用于较少使用的存储存档文件和备份文件, 价格便宜量又足, 安全性高

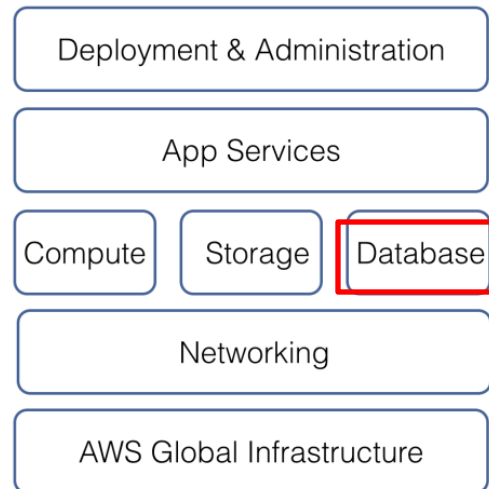


6.4.2 Amazon AWS

Database（数据库）

亚马逊提供关系型数据库和**NoSQL**数据库，以及一些**cache**等数据库服务

- **SimpleDB**: 基于云的键 / 值数据存储服务
- **DynamoDB**: **DynamoDB**是亚马逊自主研发的**No SQL**数据库，性能高，容错性强，支持分布式
- **RDS: Relational Database Service**，关系型数据库服务。支持**MySQL**，**SQL Server**和**Oracle**等数据库
- **Amazon ElastiCache**: 数据库缓存服务



6.4.2 Amazon AWS

Application Service（应用程序服务）

- **Cloud Search**: 一个弹性的搜索引擎，可用于企业级搜索
- **Amazon SQS**: 队列服务，存储和分发消息
- **Simple Workflow**: 一个 workflow 框架
- **CloudFront**: 世界范围的内容分发网络（**CDN**）
- **EMR: Elastic MapReduce**, 一个 **Hadoop** 框架的实例，可用于大数据处理



6.4.2 Amazon AWS

Deployment & Admin (部署和管理)

- **Elastic BeanStalk**: 一键式创建各种开发环境和运行时
- **CloudFormation**: 采用**JSON**格式的模板文件来创建和管理一系列亚马逊云资源
- **OpsWorks**: **OpsWorks**允许用户将应用程序的部署模块化，可以实现对数据库、运行时、服务器软件等自动化设置和安装
- **IAM**: **Identity & Access Management**, 认证和访问管理服务。用户使用云服务最担心的事情之一就是安全问题。亚马逊通过**IAM**提供了立体化的安全策略，保证用户在云上的资源绝对的安全



6.4.2 Amazon AWS

总体而言，**Amazon AWS**的产品分为几个部分：

- 计算类**

- 弹性计算云EC2**：EC2提供了云中的虚拟机
- 弹性MapReduce**：将Hadoop MapReduce搬到云环境中，大量EC2实例动态地成为执行大规模MapReduce计算任务的工作机

- 存储类**

- 弹性块存储EBS**
- 简单消息存储SQS**
- Blob对象存储S3**
- NoSQL型数据库**：SimpleDB和DynamoDB
- 关系数据库RDS**

- 工具支持**

- AWS支持多种开发语言**，提供Java、Rupy、Python、PHP、Windows &.NET 以及Android和iOS的工具集
- 工具集中包含各种语言的SDK**，程序自动部署以及各种管理工具
- AWS通过CloudWatch系统**提供丰富的监控功能

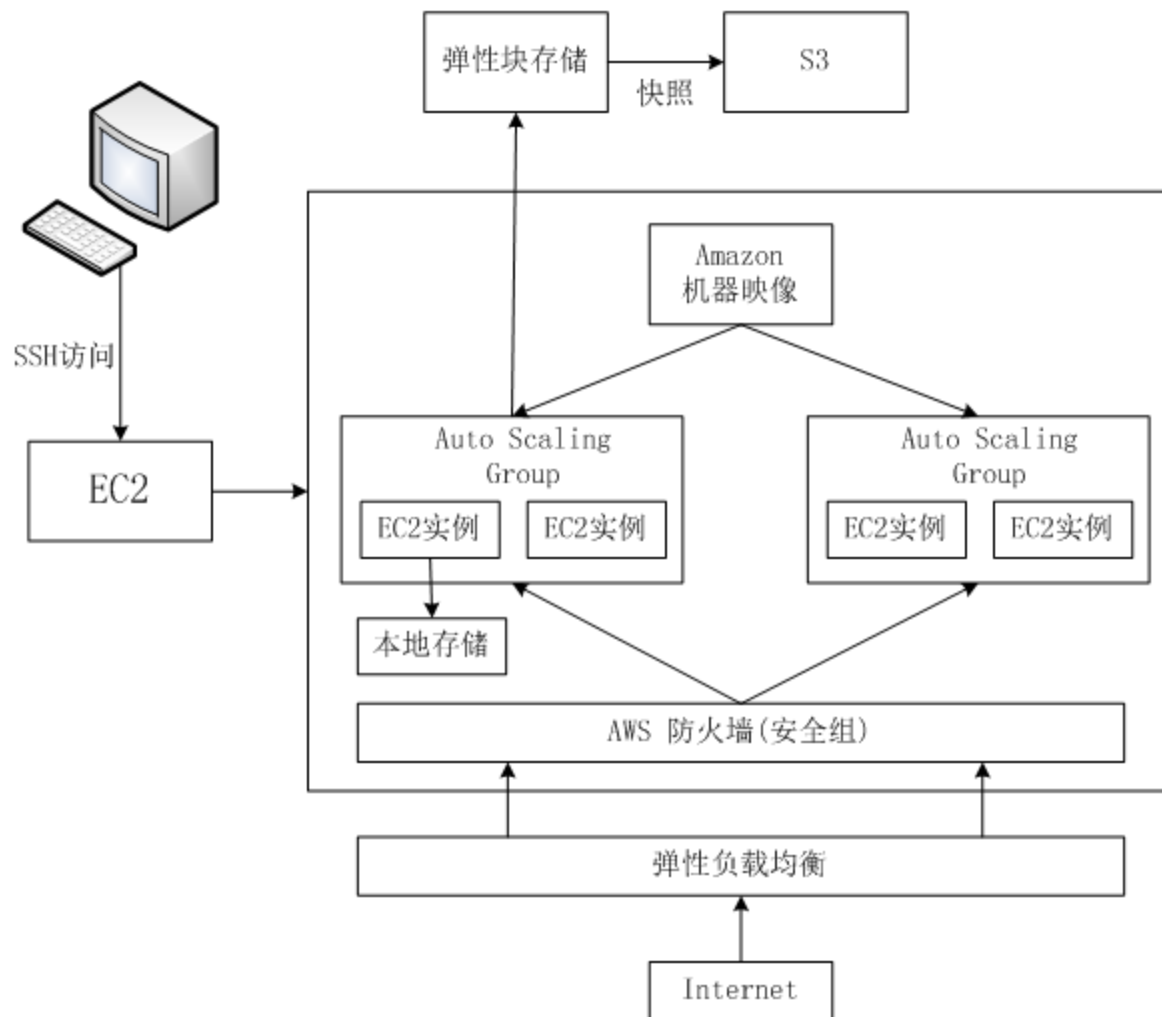
6.4.2 Amazon AWS

Amazon EC2架构

相比传统的虚拟机托管，**EC2**的最大特点是允许用户根据需求动态调整运行的实例类型和数量，实现按需付费

Amazon EC2平台主要包含如下部分：

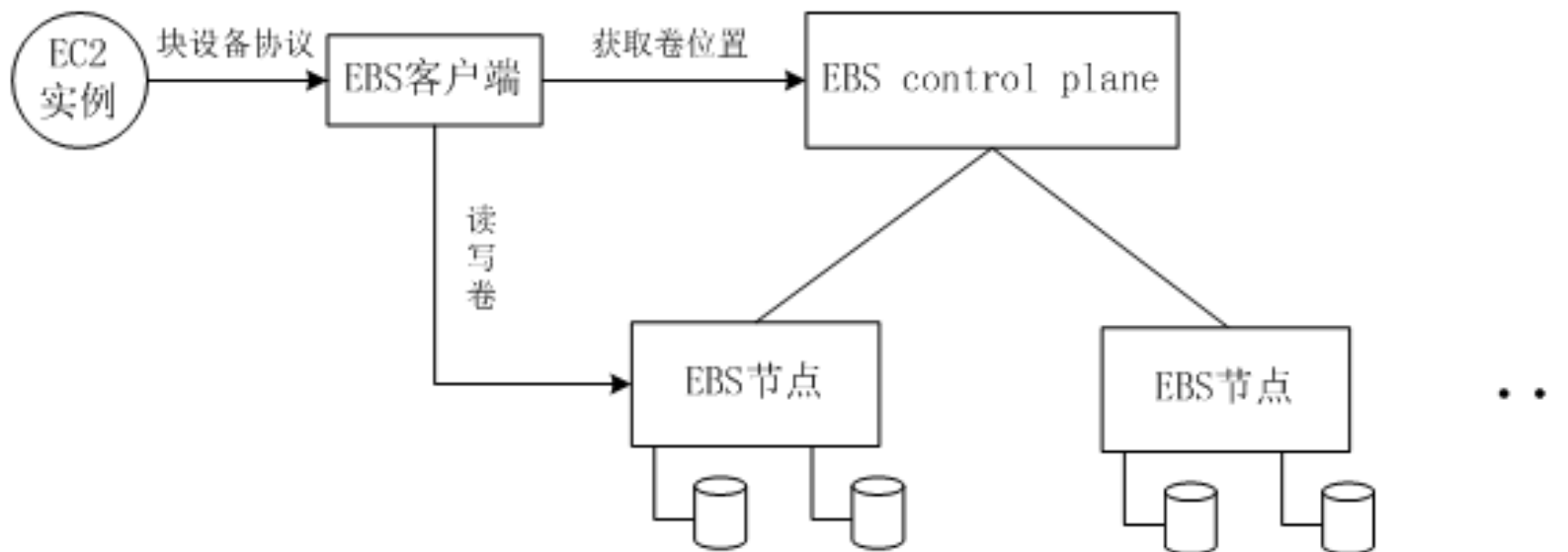
- **EC2实例（AMI）**
- 弹性块存储
- 弹性负载均衡（自动缩放）



6.4.2 Amazon AWS

EC2存储

- **EC2本地存储**是实例自带的磁盘空间，但它并不是持久的，也就是说这个实例所在的节点出现故障时，相应的磁盘空间也会随之清空
- 为了解决本地存储不可靠问题，**EC2推出了EBS**
- **EBS通过卷来组织数据**，每个**EBS卷**只能挂载到一个**EC2实例**
- **EBS卷并不与实例绑定**，而是与用户帐号绑定



6.4.2 Amazon AWS

Amazon S3和EBS的区别

	EBS	S3
服务对象	系统管理员	系统管理员/最终用户
服务场景	<ul style="list-style-type: none">•作为虚拟机硬盘，在虚拟机看来就像EBS就像本地的硬盘；当EC2实例失效时，EBS卷可以自动解除与该实例的关联，从而可以关联到新的实例	<ul style="list-style-type: none">•用户可通过Http/REST API 存取文件•网站可将静态文件存放到S3中，通过CDN网络分发到不同的区域中以提升性能•可作为虚拟机EBS卷的备份或快照
服务机制	块设备，可格式化为任何 OS 可以识别的格式	<ul style="list-style-type: none">•对象存储，桶--对象二级结构•无需在其上建文件系统

6.4.2 Amazon AWS

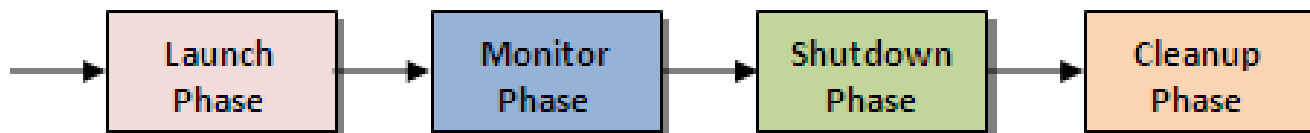
在**EC2**中创建虚拟机实例时，会提示选择镜像（**Images**）的类型：

- **S3-Hosted images**: 镜像需从**S3**存储中拷贝到**EC2**实例的本地存储。完成虚拟机镜像拷贝后启动**EC2**实例
- **EBS-backed images**: 虚拟机启动要快得多，当关闭虚拟机后，虚拟机的数据还在**EBS**上

6.4.2 Amazon AWS

AWS云管理平台

云平台负责根据客户的需求（并发数、吞吐量、数据存储空间等）来弹性地分配资源，然后将不用的资源收回



Amazon AWS云管理平台运行过程

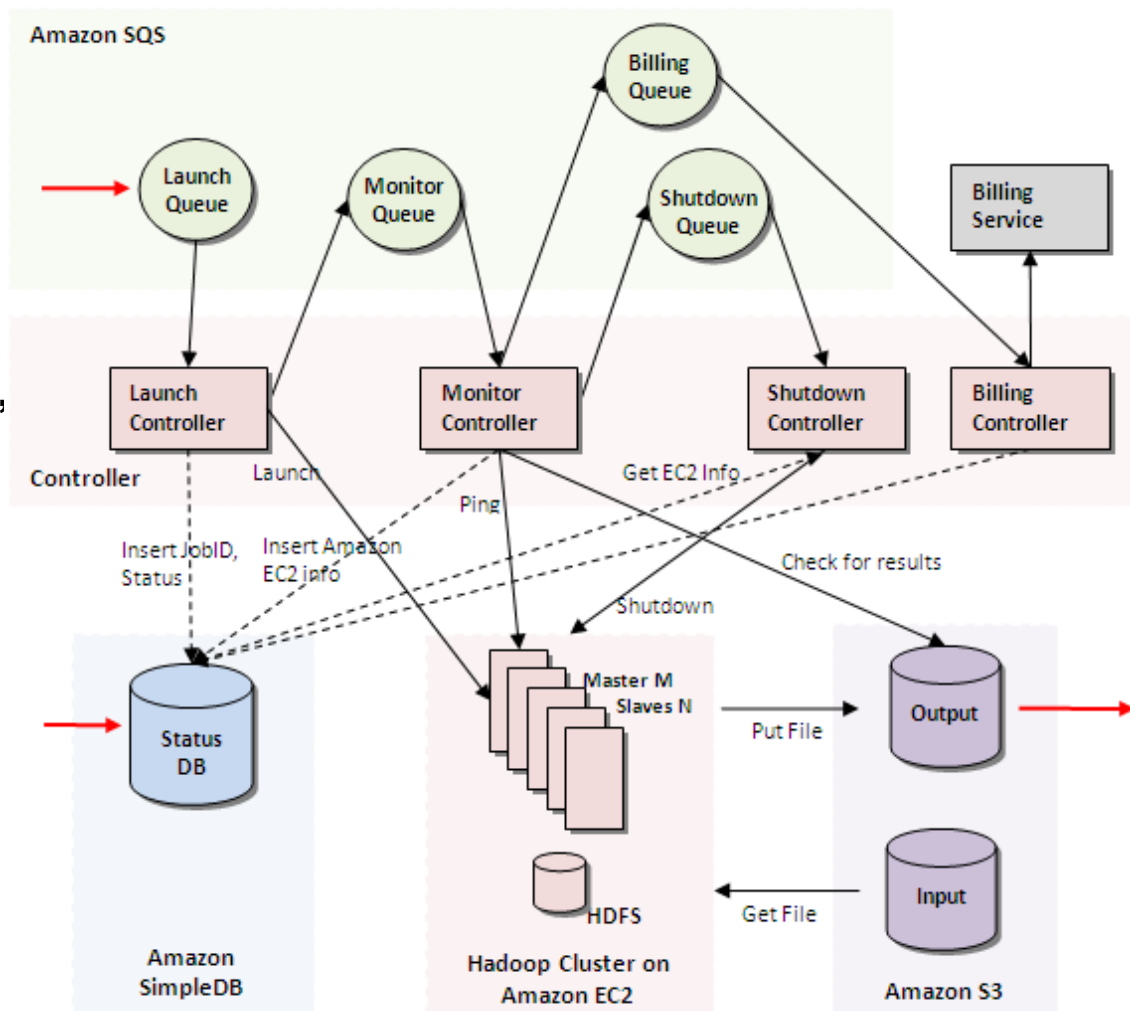
任何一个**SaaS**在提供服务的时候，云平台都会通过4个阶段对服务进行资源的分配及调整：

1. 首先启动服务，当有客户进行服务操作时，云平台会启动服务
2. 启动后监控服务的需求情况
3. 当无人访问时，停止服务
4. 回收不被使用的资源

6.4.2 Amazon AWS

一个典型的**Hadoop**作业执行时，**AWS**具体的操作流程：

- 消息平台首先发送服务启动的命令给启动控制器，由启动控制器首先将启动信息放在**SimpleDB**的缓冲区里
- 分配**EC2**的计算资源，启动**Hadoop**等操作，将计算数据从**S3**中导入**EC2**，开始进行计算和分析
- 监控控制器接收到监控信息后，对应用中的所有资源和错误进行监控，更新**SimpleDB**的缓冲区中的状态，并且根据用户的需要随时增减资源（计算节点和存储节点）
- 关闭控制器在收到关闭消息后，会停止**EC2**、**Hadoop**等资源，将运算结果放入**S3**或者客户指定的存储目标并发消息给结算控制器



6.4.3 Amazon AWS平台上的云数据库

时至今日，所有**Amazon Web Services**数据库服务都已走上正轨，成为亚马逊数十亿美元业务的组成部分。这些数据库服务包括：

- **Amazon RDS**：云中的关系数据库
- **Amazon SimpleDB**：云中的键值数据库
- **Amazon DynamoDB**：云中的**NoSQL**数据库
- **Amazon Redshift**：云中的数据仓库
- **Amazon ElastiCache**：云中的分布式内存缓存

6.4.3 Amazon AWS平台上的云数据库

SimpleDB

- **SimpleDB**是**AWS**上的第一个**NoSQL**数据库服务（键值数据库）
- 记录由主键和多个属性组成
- 可以把数据进行多副本存储，支持高并发读取
- 更新操作只能针对主副本进行，但可以快速传播到其他副本，提供最终一致性
- **SimpleDB**更适合存储小型、碎片化的零散数据

缺陷如下：

- **SimpleDB**有单表限制。**SimpleDB** 数据模型由域、项目、属性和值组成，每个域最多只能保存**10GB**的数据，所以得自己分区以免超过此限制
- 性能不稳定。**SimpleDB**以简单为设计目标，**SimpleDB**并不需要用户指定主键，也不需要用户创建索引，会默认对所有属性创建索引。然而这种简洁性也带来了一些副作用
- 一致性问题。**SimpleDB**设计时采用的是最终一致性模型

6.4.3 Amazon AWS平台上的云数据库

Amazon DynamoDB

- 采纳了**SimpleDB**中成功的托管服务形式及灵活的数据模型
- 记录由主键和多个属性组成，这一点类似于**SimpleDB**与**BigTable**，这比简单的**KV**模型更易用
- 提供了一致性读功能
- 限制了系统的功能，只能通过主键去操作记录，不能进行批量更新，这使得系统可以保证可伸缩性及任何时候的高性能
- 全面使用**SSD**来提升系统性能

6.4.3 Amazon AWS平台上的云数据库

Amazon RDS

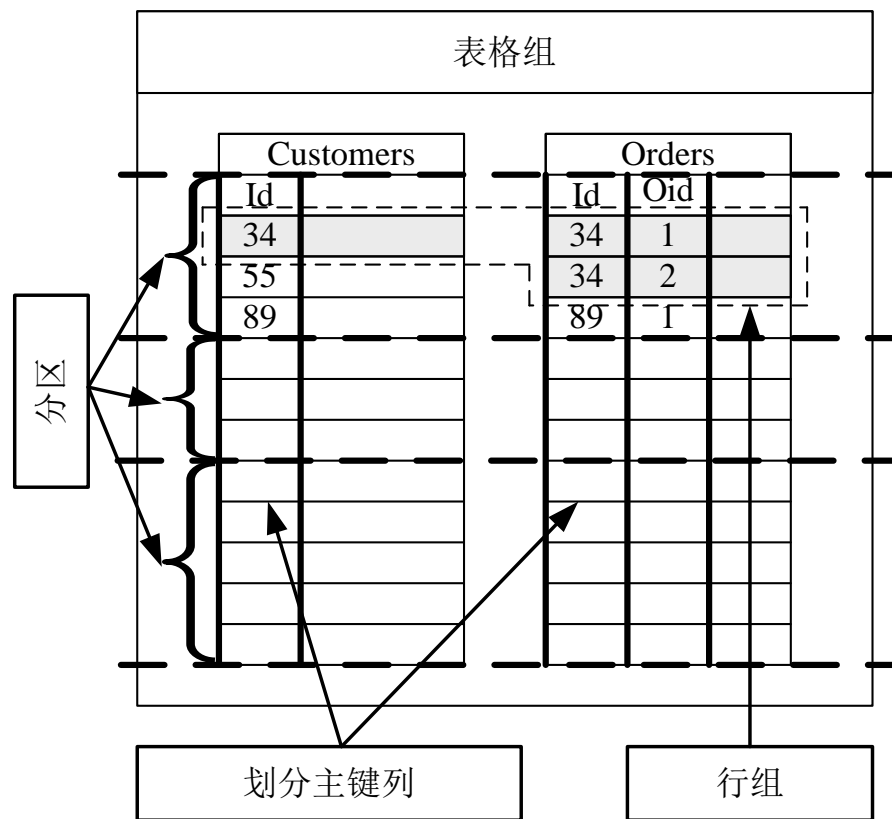
- **Amazon RDS** 有超过 **10 万** 活跃客户和 多个数据库引擎可供选择，已成为云中运行关系数据库的新常态
 - **MySQL**
 - **Oracle**
 - **SQL Server**
 - **PostgreSQL**
 - **MariaDB**
 - **Aurora**
- 借助 **AWS** 数据库迁移服务及其附带模式转换工具，客户可选择从本地部署向 **AWS** 迁移相同数据库引擎
- **RDS** 可以建立 **3TB** 和 **3万** 的 **DB** 实例

6.5 微软云数据库SQL Azure

- **SQL Azure**是微软的云关系型数据库，后端存储又称为“**云SQL Server**”
- 构建在**SQL Server**之上，通过分布式技术提升传统关系数据库的可扩展性和容错能力

1. 逻辑模型

- 一个逻辑数据库称为一个表格组
- 表格组中所有划分主键相同的行集合称为行组（**row group**）
- 只支持同一个行组内的事务，同一个行组的数据逻辑上会分布到一台服务器，以此规避分布式事务
- 通过主备复制将数据复制到多个副本，保证高可用性

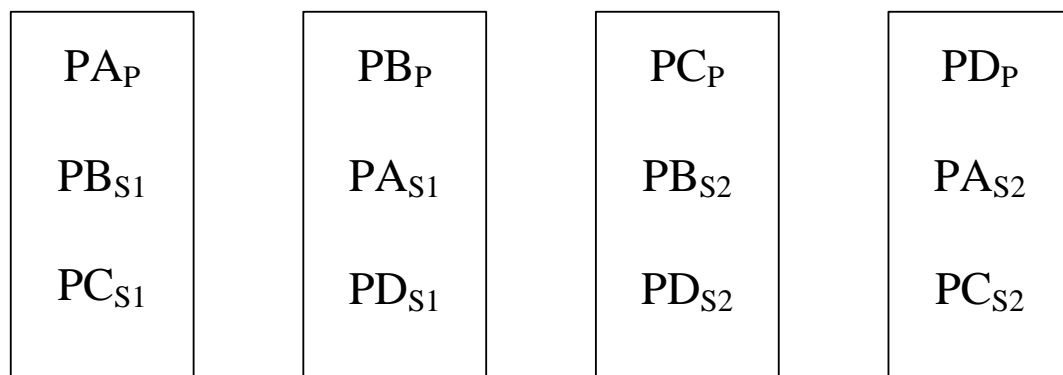


云SQL Server数据模型

6.5 微软云数据库SQL Azure

2.物理模型

- 在物理层面，每个有主键的表格组根据划分主键列有序地分成多个数据分区。每个行组属于唯一分区
- 分区是**SQL Azure**复制、迁移、负载均衡的基本单位。每个分区包含多个副本（默认为3），每个副本存储在一台物理的**SQL Server**上
- SQL Azure**保证每个分区的多个副本分布到不同的故障域。每个分区有一个副本为主副本（**Primary**），其他副本为从副本（**Secondary**）。主副本处理所有的查询、更新事务，并以操作日志的形式，将事务同步到从副本，从副本接收主副本发送的事务日志并应用到本地数据库



每台物理**SQL Server**数据库混合存放了来自不同逻辑分区的主副本和从副本

6.5 微软云数据库SQL Azure

3.体系架构

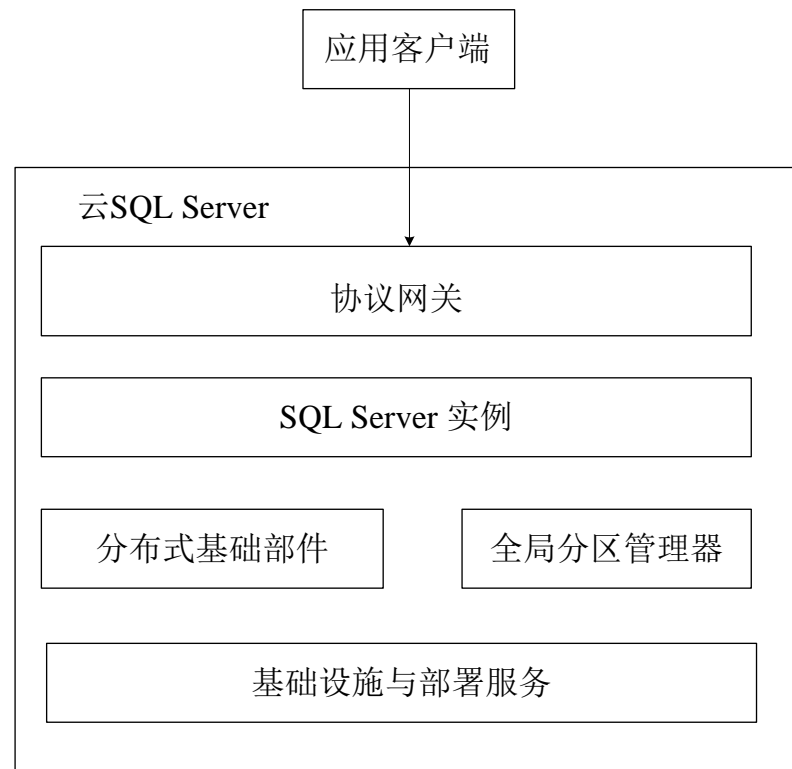
•**SQL Azure**分为四个主要部分：**SQL Server**实例、全局分区管理器、协议网关、分布式基础部件。

•每个**SQL Server**实例是一个运行着**SQL Server**的物理进程。每个物理数据库包含多个子数据库，它们之间相互隔离。子数据库是一个分区，包含用户的数据以及**schema**信息

•全局分区管理器维护分区映射表信息

•协议网关负责将用户的数据库连接请求转发到相应的主分区上

•分布式基础部件（**Fabric**）用于维护机器上下线状态，检测服务器故障并为集群中的各种角色执行选取主节点操作

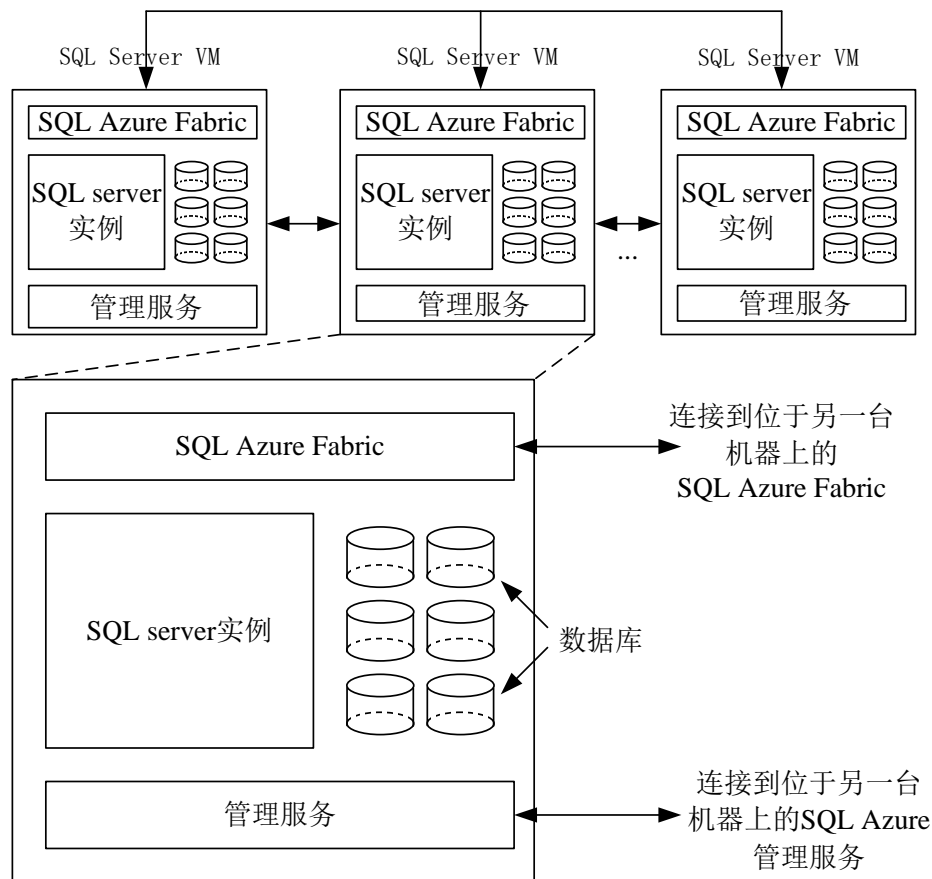


云SQL Server的分层架构

6.5 微软云数据库SQL Azure

3.体系架构

- SQL Azure**的体系架构中包含了一个虚拟机簇，可以根据工作负载的变化，动态增加或减少虚拟机的数量
- 每台虚拟机**SQL Server VM(virtual machine)**安装了**SQL Server** 数据库管理系统，以关系模型存储数据
- 通常，一个数据库会被散存储到**3~5**台**SQL Server VM**中



SQL Azure的体系架构

6.6 云数据库实践

6.6.1 阿里云RDS简介

6.6.2 RDS中的概念

6.6.3 购买和使用RDS数据库

6.6.4 将本地数据库迁移到云端RDS数据库

6.6.1 阿里云RDS简介

- **RDS**是阿里云提供的关系型数据库服务，它将直接运行于物理服务器上的数据库实例租给用户，是专业管理的、高可靠的云端数据库服务
- **RDS**由专业数据库管理团队维护，还可以为用户提供数据备份、数据恢复、扩展升级等管理功能，相对于用户自建数据库而言，**RDS**具有专业、高可靠、高性能、灵活易用等优点，能够帮助用户解决费时费力的数据库管理任务，让用户将更多的时间聚焦在核心业务上
- **RDS**具有安全稳定、数据可靠、自动备份、管理透明、性能卓越，灵活扩容等优点，可以提供专业的数据库管理平台、专业的数据库优化建议以及完善的监控体系

6.6.2 RDS中的概念

- **RDS实例**，是用户购买**RDS**服务的基本单位。在实例中：
 - 可以创建多个数据库
 - 可以使用常见的数据库客户端连接、管理及使用数据
 - 可以通过**RDS**管理控制台或**OPEN API**来创建、修改和删除数据库

- **RDS数据库**，是用户在一个实例下创建的逻辑单元
 - 一个实例可以创建多个数据库，在实例内数据库命名唯一，所有数据库都会共享该实例下的资源，如**CPU**、内存、磁盘容量等
 - **RDS**不支持使用标准的**SQL**语句或客户端工具创建数据库，必须使用**OPEN API**或**RDS**管理控制台进行操作

6.6.2 RDS中的概念

❑ 地域指的是用户所购买的**RDS**实例的服务器所处的地理位置

RDS目前支持杭州、青岛、北京、深圳和香港五个地域，服务品质完全相同。用户可以在购买**RDS**实例时指定地域，购买实例后暂不支持更改

❑ **RDS**可用区是指在同一地域下，电力、网络隔离的物理区域，可用区之间内网互通，可用区内网络延时更小，不同可用区之间故障隔离

RDS可用区又分为单可用区和多可用区

- 单可用区是指**RDS**实例的主备节点位于相同的可用区，它可以有效控制云产品间的网络延迟
- 多可用区是指**RDS**实例的主备节点位于不同的可用区，当主节点所在可用区出现故障（如机房断电等），**RDS**进行主备切换后，会切换到备节点所在的可用区继续提供服务。多可用区的**RDS**轻松实现了同城容灾

6.6.2 RDS中的概念

- 磁盘容量是用户购买**RDS**实例时，所选择购买的磁盘大小
实例所占用的磁盘容量，除了存储表格数据外，还有实例正常运行所需要的空间，如系统数据库、数据库回滚日志、重做日志、索引等
- **RDS**连接数，是应用程序可以同时连接到**RDS**实例的连接数量
 - 任意连接到**RDS**实例的连接均计算在内，与应用程序或者网站能够支持的最大用户数无关
 - 用户在购买**RDS**实例时所选择的内存大小决定了该实例的最大连接数

6.6.3 购买和使用RDS数据库

- ❑ 进入阿里云首页，阿里云官网（<http://www.aliyun.com/>）
- ❑ 使用支付宝账户登录阿里云，账户登录成功后，点击“云数据库RDS”，即可进入云数据库RDS页面。点击“立即购买”，即可获得RDS服务。新用户可以免费体验半年的RDS服务。购买成功后，可以通过管理控制台对RDS实例进行使用
- 购买RDS实例
- 管理RDS
- 管理RDS实例
- 新建RDS账号
- 新建RDS数据库
- 连接RDS数据库
- 操作RDS数据库

6.6.3 购买和使用RDS数据库

1.购买RDS实例

进入RDS页面后，点击“立即购买”，即可跳到下图的购买页面

如果已经购买阿里云服务器ECS（Elastic Compute Service），若选择和ECS所在地域相同，ECS和RDS之间可以以内网方式访问

关系型数据库RDS

包年包月

按量付费

推荐搭配云服务器使用，内网流量免费！

地域：

青岛

杭州

北京

香港

深圳

不同地域之间的产品内网不互通；订购后不支持更换地域，请谨慎选择 教我选择>>

基本配置

可用区：

单可用区

请选择可用区

数据库类型：

MySQL

版本：

5.5

5.6

存储空间：

250GB

500GB

1000GB

5 GB

步长为5GB

内存：

240MB

最大连接数:60 IOPS:150

购买时长：

1个月

2

3

4

5

6

7

8

9

1年

2年

3年

数量：

1 台

6.6.3 购买和使用RDS数据库

2.管理RDS

购买**RDS**实例成功后，可以通过管理控制台，查看已开通的产品与服务。点击云数据库**RDS** 进入管理界面如下图。我们可以创建新实例、对已购买实例进行管理、续费和升级操作

云数据库 RDS

云数据库管理

全部

青岛

刷新

新建实例

常规实例

请输入实例名称或实例ID进行搜索

搜索

实例名称	运行状态	实例类型	数据库类型	可用区类型	所在可用区	付费类型	操作
<div><div></div><div>rdsmnbrjq7z3izz</div><div>rdsmnbrjq7z3izz</div></div>	运行中	专享	MySQL5.6	单可用区	青岛可用区B	包月 30天后到期	<div>管理</div> <div>续费</div> <div>升级</div>

批量续费

共有1条，每页显示：20条

«

<

1

>

»

6.6.3 购买和使用RDS数据库

3.管理RDS实例

- 点击已购买RDS实例的管理操作，可以查看该实例的基本信息如下图
- 一个实例可以创建多个数据库，在实例内数据库命名唯一，所有数据库都会共享该实例下的资源

The screenshot displays the RDS console interface for a specific instance. At the top, the instance name 'rdsmnbrjq7z3iz...' is shown with a status of '(运行中)' and a '返回实例列表' button. Navigation links for '功能指南', '登录数据库', and '内外网切换' are also present. A left sidebar contains a menu with options like '基本信息', '帐号管理', '数据库管理', '系统资源监控', '备份与恢复', '参数设置', '日志管理', '性能优化', '阈值报警', and '安全控制'. The main content area is divided into several sections: '基本信息' (Basic Information) with fields for instance ID, name, region (青岛), availability zone (可用区B), and endpoint; '运行状态' (Running Status) showing '运行中' and '可用性: 100.0%'; '配置信息' (Configuration) detailing memory, space, instance type (专享), database type (MySQL5.6), and IOPS; '运行周期' (Running Cycle) showing creation time and billing type; and '可维护的时间段' (Maintenance Window) showing the current status as '未设置'.

基本信息	
实例ID: rdsmnbrjq7z3iz	地域: 青岛
名称: rdsmnbrjq7z3iz	可用区: 可用区B [迁移可用区]
外网地址: yuqianli.mysql.rds.aliyuncs.com	端口: 3306 如何连接RDS 如何设置白名单

运行状态	
运行状态: 运行中	锁定模式: 正常
可用性: 100.0%	已用空间: 501M

配置信息	
数据库内存: 240M	数据库空间: 5120M
实例类型: 专享	数据库类型: MySQL5.6
最大连接数: 60	最大IOPS: 150

运行周期	
创建时间: 2015-01-25	付费类型: 包月 30 天后到期

可维护的时间段	
时间段: 未设置	设置

6.6.3 购买和使用RDS数据库

4.新建RDS账号

- 点击“创建新账号”按钮后，可创建新的**RDS**账号，并选定需绑定的数据库，以及输入账号密码和账号类型（读写权限）等信息
- **MySQL**实例支持最多创建**50**个账号，**SQL Server**实例支持最多创建**20**个账号
- 创建完**RDS**账号后，还可以对**RDS**账号进行重置密码和修改操作

6.6.3 购买和使用RDS数据库

5.新建RDS数据库

- 点击“数据库管理”按钮后，可查看数据库基本信息，并可对数据库进行创建、删除（需至少有1个数据库）的操作
- 点击“增加数据库”后，在弹出的界面中填写数据库相关信息，提交后即可生效
- 此外，**RDS**数据库还可以是自建数据库迁移来的或是从其他**RDS**实例中迁入的

6.6.3 购买和使用RDS数据库

6.连接RDS数据库

- 如果是在阿里云服务器**ECS**上连接**RDS**数据库，就选择内网模式；
- 如果是在其他服务器上连接**RDS**使用，就选择外网模式，在控制台的右上角有切换方式
- 从本地对云端的**RDS**数据库进行远程访问时使用外网模式，需要在“安全控制->白名单设置”位置填入本地机器的外网**IP**，从而让**RDS**数据库允许我们的本地机器访问

6.6.3 购买和使用RDS数据库

6.连接RDS数据库

- ✓ 使用客户端**MySQL-Front** 访问
- ✓ 使用数据库管理工具**Navicat_MySQL**
- ✓ 使用**MySQL**命令登录

命令格式如下：

```
mysql -u user_name -h yuqianli.mysql.rds.aliyuncs.com -P3306 -pxxxx
```

- ✓ 使用阿里云控制台**iDB Cloud**访问

RDS连接地址以及端口不需要再输入，
只需在“用户名”中输入数据库的账号，
在“密码”栏中输入数据库账号的密码，
便可以登录**RDS**。



iDB Cloud

yuqianli.mysql.rds.aliyuncs.com:3306

mysqladb

输入密码

☐ 记住密码 [关于iDB Cloud](#)

登录

Copyright © iDBCloud All Rights Reserved

6.6.3 购买和使用RDS数据库

7.操作RDS数据库

- 连接RDS数据库后，对数据库的操作与直接对本机MySQL数据库操作无异。iDB Cloud登录数据库后的界面如下图所示：

- 在“iDB Cloud登录数据库界面”（如右图）的顶端可以看到iDB Cloud提供以下三种创建表的方法：

- 可视化界面
- SQL窗口
- 命令窗口



6.6.4将本地数据库迁移到云端RDS数据库

- 假设我们有一个本地应用程序，它使用本地的**MySQL**数据库存取和管理数据。现在，我们打算把本地**MySQL**数据库中的数据全部迁移到远程的阿里云**RDS**数据库中，本地应用程序不迁移（依然运行在本地），但是，我们希望本地应用程序使用云端的**RDS**数据库服务进行数据存取和管理。为此，需要执行以下两步操作：

第一步：把本地数据库迁移到云端的**RDS**数据库

第二步：修改本地应用程序配置，使用**RDS**数据库服务

6.6.4将本地数据库迁移到云端RDS数据库

如何把本地数据库迁移到云端的**RDS**数据库？

第1步：在本地数据库中创建一个迁移账号

第2步：设置迁移账号权限

第3步：确认本地数据库中的配置文件是否正确，需要确认本地数据库中的**MySQL**配置文件**my.cnf**

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
log bin = mysql-bin
binlog format = row
log-error = /tmp/error.log
server_id = 2
# Disabling symbolic-links is recommended to prevent assorted security risks
```

第4步：登录本地数据库，通过命令查看是否为“**ROW**”模式

第5步：在**RDS**管理控制台对应的实例页面，点击“将数据迁移至**RDS**”按钮，在弹出的页面中，填写待迁移的本地数据库连接地址、数据库连接端口、数据库账号、数据库密码，即可完成从本地迁移到云端

本章小结

- 云数据库是在云计算兴起的大背景下发展起来的，在云端为用户提供数据服务，用户不需要自己投资建设软硬件环境，只需要向云数据库服务供应商购买数据库服务，就可以方便、快捷、低成本地实现数据存储和管理功能
- 云数据库具有动态可扩展、高可用性、低成本、易用性、大规模并行处理等突出优点，是大数据时代企业实现低成本的大规模数据存储的理想选择
- 云数据库市场有很多代表性的产品可供选择。**Amazon**是云数据库市场的先行者，谷歌和微软公司都开发了自己的云数据库产品，都在市场上形成了自己的影响力
- 以阿里云**RDS**为例，介绍了云数据库的具体使用方法