

# 第六章 云数据库

## 一、云数据库概述

如何方便、快捷、低成本地存储海量数据？云数据库

### 1. 云计算是云数据库兴起的基础

- 云计算概念：

通过整合、管理、调配分布在各处的计算资源，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

高扩展性、虚拟化、超大规模、极其廉价、高可靠性、通用性、随时服务、按需服务

### 2. 云数据库概念

云数据库是部署在云计算环境中的虚拟化数据库。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。

在云数据库中，所有数据库功能都是在云端提供的，客户端可以通过网络远程使用云数据库提供的服务。客户端不需要了解云数据库的底层细节，所有底层硬件都已经被虚拟化，对客户端而言是透明的，同时可以获得理论上近乎无限的存储和处理能力。

### 3. 云数据库的特性

- 动态可扩展
- 高可用性
- 较低的使用代价
- 易用性
- 高性能
- 免维护
- 安全

### 4. 云数据库是个性化数据库存储需求的理想选择

- 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

是否自建数据库或者选择云数据库，取决于企业自身的需求

### 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的技术
- 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，也可以是NoSQL数据库所使用的非关系模型。

## 二、云数据库系统架构

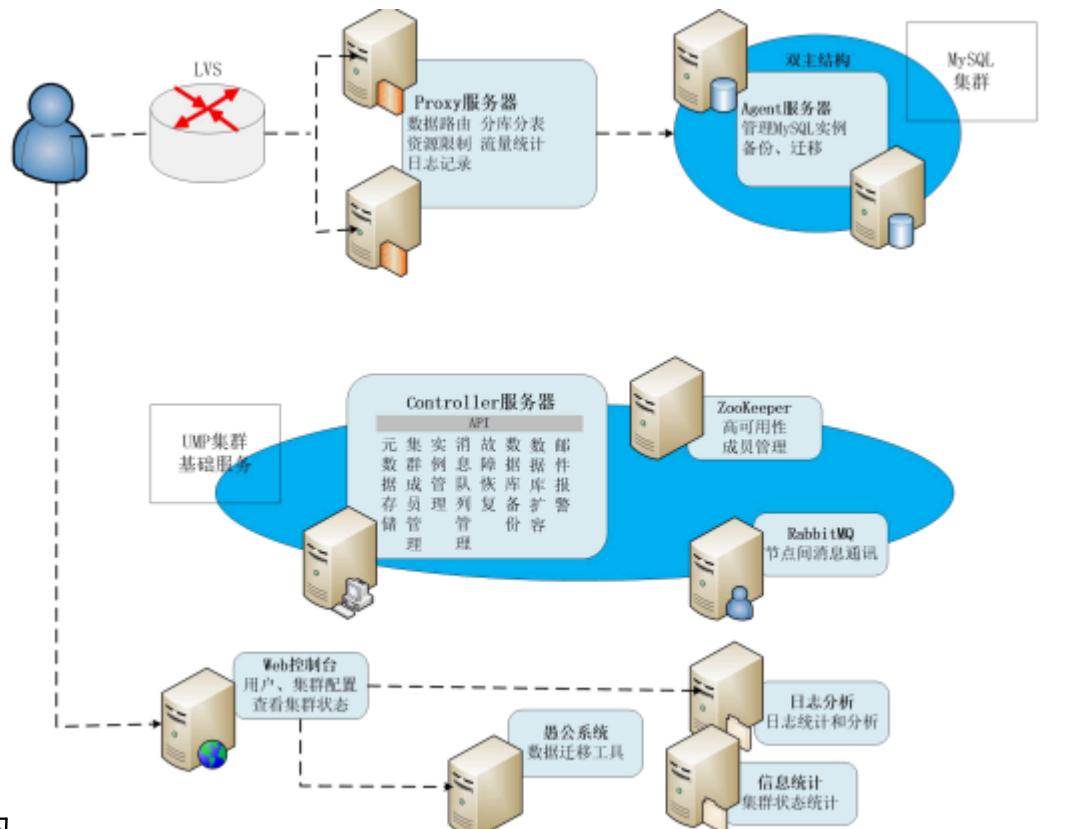
### 1. UMP系统概述

不同的云数据库产品采用的系统架构存在很大差异，这里以阿里巴巴核心系统数据库团队开发的UMP系统为例，进行介绍

UMP系统是低成本和高性能MySQL云数据库方案

总的来说UMP系统架构设计遵循以下原则：

- 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



### 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

### 3. 组件介绍

- Mnesia：Mnesia是一个分布式数据库管理系统，它的数据库模式可以在运行时动态重配置，表可以被迁移或复制到多个节点来改进容错性
- RabbitMQ：RabbitMQ是一个工业级的消息队列产品，作为消费传输中间件来使用，可实现可靠消息传送，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
- 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，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：UMP系统将多台服务器上运行的应用系统的相同配置保存在Zookeeper的某个目录节点，然后监控配置信息的状态
  - 提供分布式锁(选出一个集群的总管)：UMP集群中部署了多个Controller服务器，为了保证系统的正确运行，对于某些操作，在某一时刻只能由一个服务器执行，Zookeeper可以从多个Controller服务器中选择出这么一个总管，由总管负责发起各种系统任务
  - 监控所有的MySQL实例：集群中运行MySQL实例的服务器发生故障时，必须及时被监听到，然后使用其他正常服务器来替代故障服务器
- LVS：LVS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。
  - 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
  - Web控制台：Web控制台向用户提供系统管理界面
  - 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。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。
  - 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
  - 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，并且支持实时查询一段时间内的慢日志和统计报表
  - 信息统计服务器：定期将采集到的用户连接数以及MySQL实例的进程状态进行统计，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

- 愚公系统：实现在不停机的情况下动态扩容，缩容和迁移

#### 4. UMP系统功能

UMP系统建立在巨大集群之上，通过多个组件协同作业，整个系统实现了对用户透明的各种功能

- 容灾：为了实现容灾，UMP系统为每一个用户创建两个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的时候，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
- 读写分离：充分利用主从库实现读写分离功能，实现负载均衡

UMP系统实现了对用户透明的读写分离功能，当整个功能被开启的时候，复杂向用户提供MySQL数据库的Proxy服务器，就会对用户发起SQL语句进行解析，如果属于写操作，就直接发送到主库，如果是读操作就会均衡地发送到主库和从库上执行

- 分库分表：UMP支持对用户透明的分库分表，当采用分库分表的时候，系统处理用户的查询过程如下：

- 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。

- 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

- 资源调度：UMP系统中有三种规模的用户，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；

对于数据量非常大、需要分库分表的用户，会占有多个独立的MySQL实例。

- 资源隔离：UMP采用两种资源隔离的方式

方法	应用场景	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 (每秒查询数量)	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

- 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。
- 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 \*`，需要消耗非常多资源的SQL查询。