

多架构数据库技术

云数据库介绍

ougd . ai . wangjie

- 1.什么是云数据库？
- 2.云数据库产品
- 3.云数据库架构
- 4.亚马逊云与数据库
- 5.微软云与数据库
- 6.阿里云与数据库
- 1.什么是云数据库？

基础概述

云计算是云数据库兴起的基础

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
云数据库是个性化数据存储需求的理想选择

有哪些特性

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

多架构数据库技术

有哪些优点

云数据库可以满足大企业的海量数据存储需求

云数据库可以满足中小企业的低成本数据存储需求

云数据库可以满足企业动态变化的数据存储需求

常见应用场景

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

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

2.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

多架构数据库技术

腾讯 腾讯云数据库

Amazon 的云数据库产品

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

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

Microsoft 的云数据库产品

SQL Azure 属于关系型数据库

SQL Azure 支持使用 TSQL (Transact Structured Query Language) 来管理、创建和操作云数据库 SQL Azure 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 SQL Azure 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

SQL Azure 支持云中的事务，支持局部事务，但是不支持分布式事务

3.云数据库架构

UMP 系统架构设计遵循的原则

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

多架构数据库技术

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 UMP 系统架构中的角色

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 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

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 Agent 服务器还负责收集和分析 MySQL 进程的统计信息、慢查询日志（Slow Query Log）和 bin-logWeb 控制台

Web 控制台向用户提供系统管理界面

多架构数据库技术

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

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

UMP 系统架构依赖的组件

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 LVSLVS(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端程序

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

多架构数据库技术

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）Zookeeper 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 UMP 系统架构的功能

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

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

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

多架构数据库技术

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

分库分表

UMP 支持对用户透明的分库分表

当采用分库分表时，系统处理用户查询的过程如下：

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

对 SQL 语句进行重写，得到多个针对相应 MySQL 实例的子语句

把子语句分发到对应的 MySQL 实例上执行

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

资源隔离

方法 应用场合 实现方式

多架构数据库技术

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

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 *”

4.亚马逊云与数据库

基础概述

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 这三层基础架构

多架构数据库技术

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。

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), 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 **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。

Compute(计算)

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

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,

多架构数据库技术

Amazon EBS 可以为 Amazon EC2 的虚拟机创建卷 `volumes`。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

Database(数据库)

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

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：一个工作流框架

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

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

核心产品

多架构数据库技术

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

工具类

AWS 支持多种开发语言，提供 Java、Ruby、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

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

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：

首先启动服务，当有客户进行服务操作时，云平台会启动服务启动后监控服务的需求情况

当无人访问时，停止服务

回收不被使用的资源

云数据库产品

多架构数据库技术

Amazon RDS

云中的关系数据库

Amazon RDS 有多个数据库引擎可供选择

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

Amazon SimpleDB

SimpleDB 是 AWS 上的第一个 NoSQL 数据库服务

优点

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

缺点

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

Amazon DynamoDB

云中的 NoSQL 数据库

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

多架构数据库技术

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

Amazon Redshift

云中的数据仓库

Amazon ElastiCache

云中的分布式内存缓存

5. 微软云与数据库

SQL Azure 是微软的云关系型数据库，后端存储又称为“云 SQL Server”，它构建在 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

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。

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。

多架构数据库技术

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

6. 阿里云与数据库

RDS 概述

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

RDS 概念

RDS 实例：是用户购买 RDS 服务的基本单位

RDS 数据库：是用户在一个实例下创建的逻辑单元

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 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

磁盘容量：是用户购买 RDS 实例时，所选择购买的磁盘大小

RDS 连接数：是应用程序可以同时连接到 RDS 实例的连接数量

RDS 使用

多架构数据库技术

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

RDS 迁移

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

把本地数据库迁移到云端的 RDS 数据库

在本地数据库中创建一个迁移账号

设置迁移账号权限

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` 登录本地数据库，通过命令查看是否为“ROW”模式在 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