

目录

3.1 基础存储架构Dynamo

3.2 弹性计算云EC2

3.3 简单存储服务S3

3.4 非关系型数据库服务SimpleDB和DynamoDB

3.5 关系数据库服务RDS

3.6 简单队列服务SQS

3.7 内容推送服务CloudFront

3.8 其他Amazon云计算服务

3.9 AWS应用实例

3.10 小结

3.3 简单存储服务S3

- ▶ 3.3.1 S3的基本概念和操作
- 3.3.2 S3的数据一致性模型
- 3.3.3 S3的安全措施

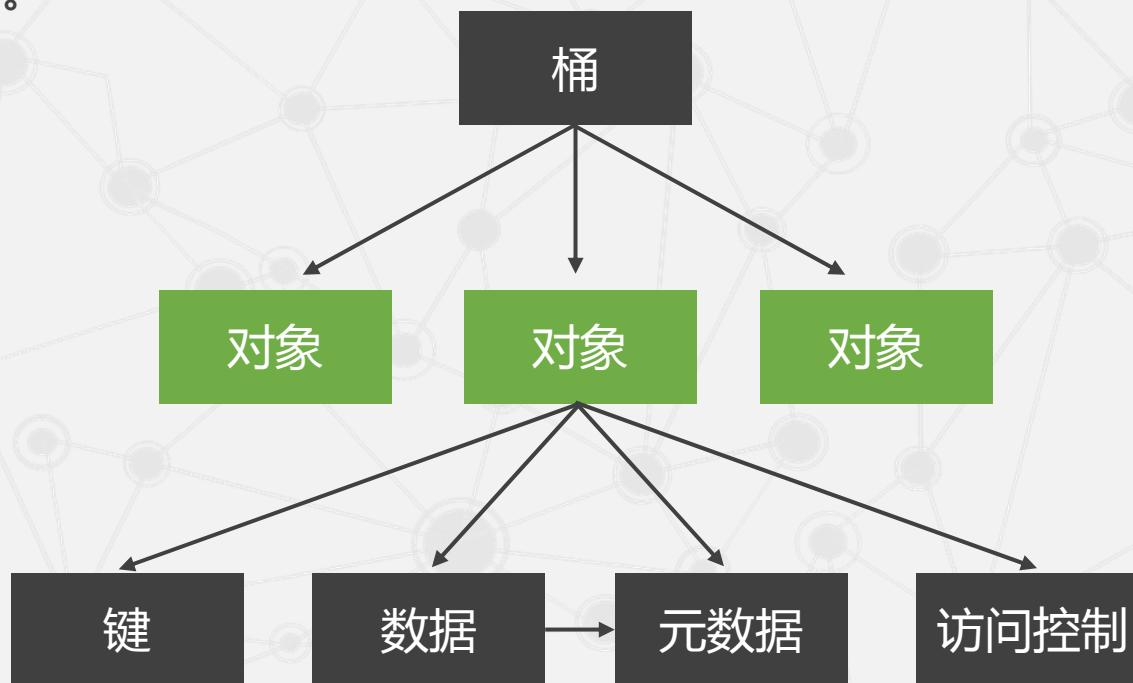
● S3的基本概念和操作

简单存储服务（Simple Storage Services，S3）构架在Dynamo之上，用于提供任意类型文件的临时或永久性存储。S3的总体设计目标是可靠、易用及低成本。和Google的GFS在一个层面。

S3存储系统的基本结构，
其中涉及两个基本概念：

桶
Bucket

对象
Object



• S3的基本概念和操作

桶

Bucket

桶是用于存储对象的容器，其作用类似于文件夹，但桶不可以被嵌套，即在桶中不能创建桶。

目前，Amazon限制了每个用户创建桶的数量，但没有限制每个桶中对象的数量。

桶的名称要求在整个Amazon S3的服务器中是全局唯一的，以避免在S3中数据共享时出现相互冲突的情况。

对象——S3的基本存储单元

对象

Object

数据

任意类型，但大小会受到对象最大容量的限制

元数据

数据内容的附加描述信息，通过名称-值（name-value）集合的形式来定义。

可以是系统默认的元数据或用户自定义的元数据

元数据名称	名称含义
last-modified	对象被最后修改的时间
ETag	利用MD5哈希算法得出的对象值
Content-Type	对象的MIME（多功能网际邮件扩充协议）类型，默认二进制/八位组
Content-Length	对象数据长度，以字节为单位

对象——S3的基本存储单元

对象

Object

数据

任意类型，但大小会受到对象最大容量的限制

元数据

数据内容的附加描述信息，通过名称-值（name-value）集合的形式来定义。

可以是系统默认的元数据或用户自定义的元数据

每个对象在所在的桶中有唯一的键（key）。通过将桶名和键相结合的方式，可以标识每个对象。

键在对象创建后无法被更改，即重命名对于S3中的对象是无效的。

• S3的基本概念和操作

S3中支持对桶和对象的操作，主要包括：**Get**、**Put**、**List**、**Delete**和**Head**。

下图列出了五种操作的主要内容。

操作目标	Get	Put	List	Delete	Head
桶	获取桶中对象	创建或更新桶	列出桶中所有键	删除桶	—
对象	获取对象的数据和元数据	创建或更新对象	—	删除对象	获取对象元数据

表3-3 S3的主要操作

3.3 简单存储服务S3

3.3.1 S3的基本概念和操作

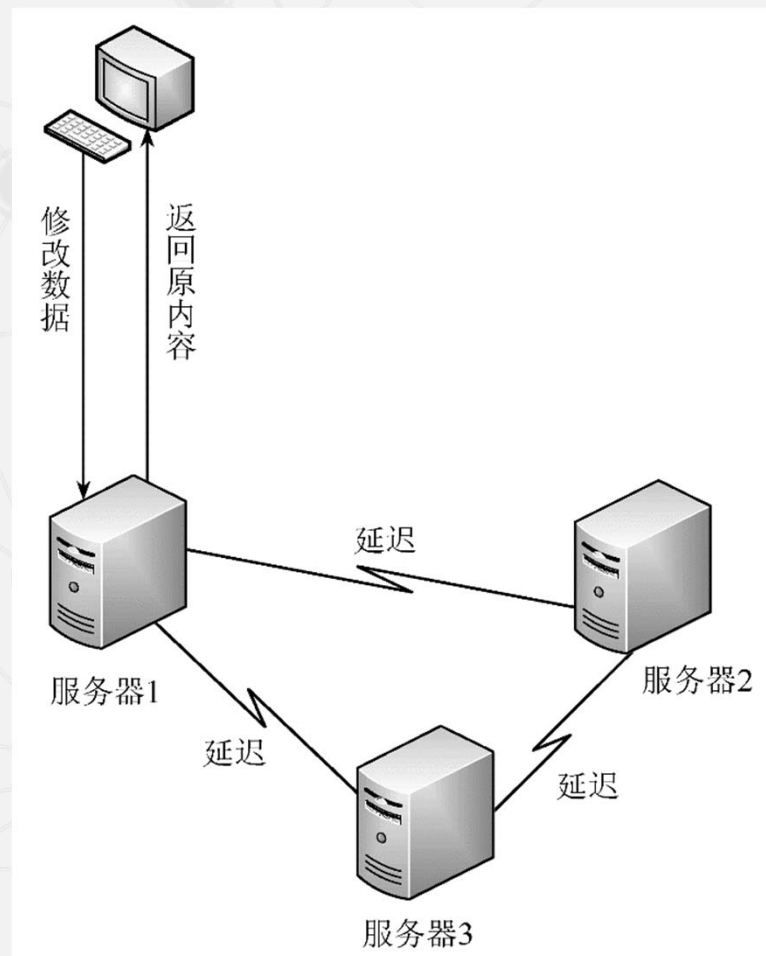
▶ 3.3.2 S3的数据一致性模型

3.3.3 S3的安全措施

- S3的数据一致性模型

与其构建的基础Dynamo相同，S3中采用了 最终一致性 模型

在数据被充分传播到所有的存放节点之前，服务器返回给用户的仍是原数据，此时用户操作可能会出现后面几种情况：



3.3 简单存储服务S3

《云计算》第三版配套PPT课件

• S3的数据一致性模型

	用户操作	结果
1	写入一个新的对象并立即读取它	服务器可能返回“键不存在”
2	写入一个新的对象并立即列出桶中已有的对象	该对象可能不会出现在列表中
3	用新数据替换现有的对象并立即读取它	服务器可能返回原有的数据
4	删除现有的对象并立即读取它	服务器可能返回被删除的数据
5	删除现有的对象并立即列出桶中的所有对象	服务器可能列出被删除的对象

3.3 简单存储服务S3

3.3.1 S3的基本概念和操作

3.3.2 S3的数据一致性模型

► 3.3.3 S3的安全措施

S3的安全措施

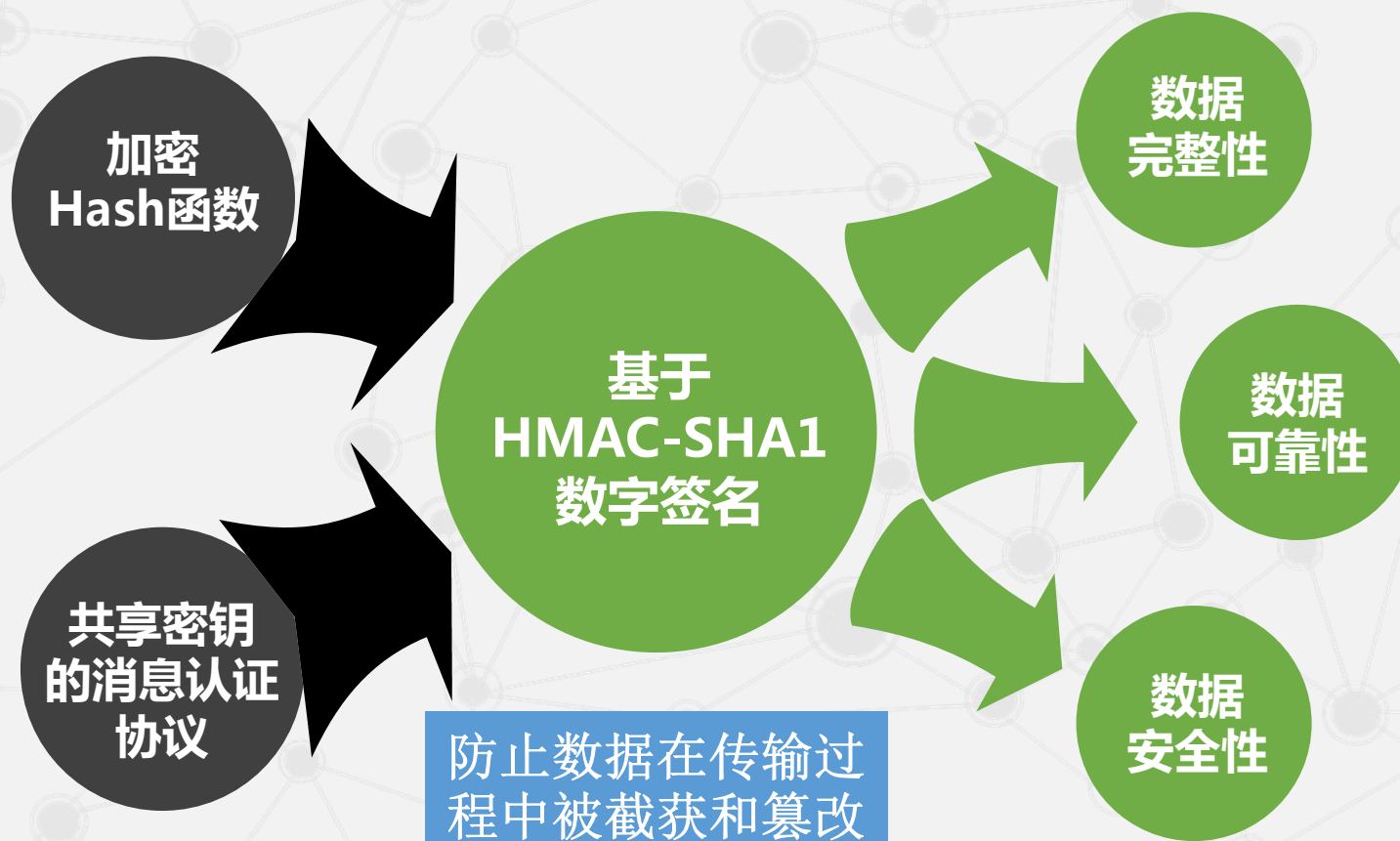
对于用户尤其是商业用户来说，系统的易用性是其考虑的一方面，但最终决定其是否使用S3服务的通常是S3的安全程度

S3向用户提供

身份认证 (Authentication) **访问控制列表 (ACL)**

身份认证 (Authentication)

一个加密的Hash函数
一个加密的随机密钥
一个安全的密钥交换机制



新用户注册→分配Access Key ID (20位) 和Secret Access Key (40位)

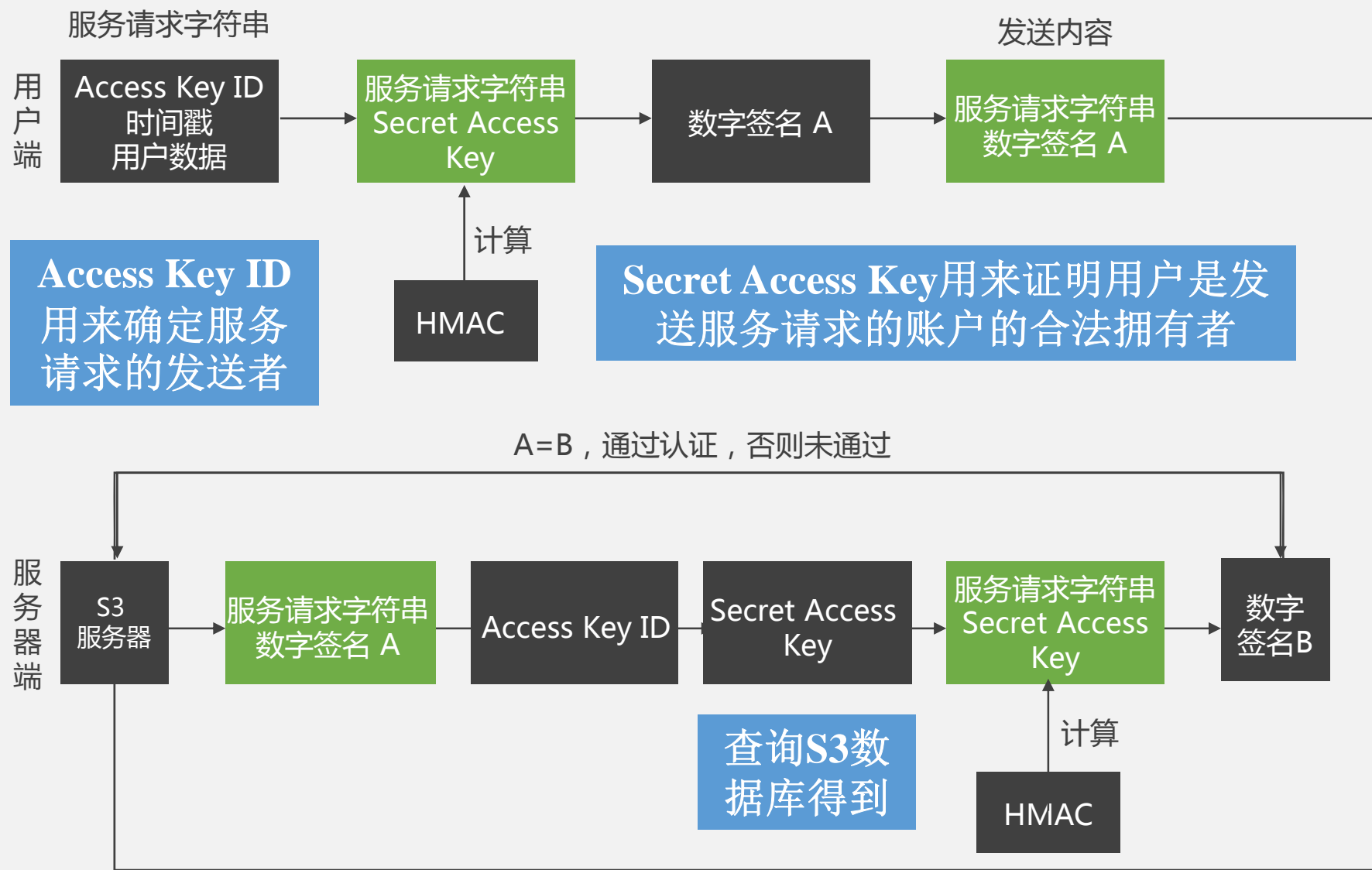


图3-17 S3数字签名具体实现过程

访问控制列表（Access Control List）

访问控制列表是S3提供的可供用户**自行定义**的访问控制策略列表。

S3的访问控制策略（ACP）提供如下所列的五种访问权限。

权 限	允许操作目标	具体权限内容
READ	桶	列出已有桶中的对象
	对象	读取数据及元数据
WRITE	桶	创建、覆写、删除桶中对象
READ_ACP	桶	读取桶的ACL
	对象	读取对象中的ACL
WRITE_ACP	桶	覆写桶的ACP
	对象	覆写对象的ACP
FULL_CONTROL	桶	允许进行以上所有操作，是S3提供的最高权限
	对象	



注意

S3的ACL不具有继承性

桶和对象的ACL是各自独立的，对桶有某种访问权限不代表对桶中的对象也具有相同的权限。

S3中有三大类型的授权用户

所有者 (Owner)

所有者是桶或对象的创建者，默认具有WRITE_ACP权限。
所有者默认就是最高权限拥有者。

所有者本身要服从ACL，如果该所有者没有READ_ACP，则无法读取ACL。
但所有者可以通过覆写相应桶或对象的ACP获取想要的权限，所以，从某种意义上说，所有者具有最高权限。

S3中有三大类型的授权用户

所有者 (Owner)

所有者是桶或对象的创建者，默认具有WRITE_ACP权限。
所有者默认就是最高权限拥有者。

个人授权用户 (User)

两种授权方式，

1. 通过电子邮件地址授权，即授权给和某个特定电子邮件地址绑定的AWS用户；
2. 通过用户ID进行授权，即直接授权给拥有某个特定AWS ID的用户。

通过电子邮件地址方式授权的方法最终还是在S3服务器内部转换成相应的用户ID进行授权

S3中有三大类型的授权用户

所有者 (Owner)

所有者是桶或对象的创建者，默认具有WRITE_ACP权限。
所有者默认就是最高权限拥有者。

个人授权用户 (User)

两种授权方式，

1. 通过电子邮件地址授权，即授权给和某个特定电子邮件地址绑定的AWS用户；
2. 通过用户ID进行授权，即直接授权给拥有某个特定AWS ID的用户。

组授权用户 (Group)

一种是AWS用户组，它将授权分发给所有AWS账户拥有者；
另一种是所有用户组，允许匿名访问，是一种有着很大潜在危险的授权方式。

目录

3.1 基础存储架构Dynamo

3.2 弹性计算云EC2

3.3 简单存储服务S3

3.4 非关系型数据库服务SimpleDB和DynamoDB

3.5 关系数据库服务RDS

3.6 简单队列服务SQS

3.7 内容推送服务CloudFront

3.8 其他Amazon云计算服务

3.9 AWS应用实例

3.10 小结

S3：提供任意类型文件的临时或永久性存储

非关系型数据库SimpleDB和DynamoDB：存储结构化数据，并为这些数据提供查找、删除等基本的数据库功能

3.4 非关系型数据库服务 SimpleDB和DynamoDB

- ▶ 3.4.1 非关系型数据库与传统关系数据库的比较
- 3.4.2 SimpleDB
- 3.4.3 DynamoDB
- 3.4.4 SimpleDB和DynamoDB的比较

• 非关系型数据库与传统关系数据库的比较

关系型数据库

优点

具有高一致性，在ACID方面很强，移植性很高

缺点

可扩展性方面能力较弱

非关系型数据库

优点

具有很高的可扩展性，具有很好的并发处理能力

缺点

缺乏数据一致性保证，处理事务性问题能力较弱
难以处理跨表、跨服务器的查询

3.4 非关系型数据库服务 SimpleDB和DynamoDB

3.4.1 非关系型数据库与传统关系数据库的比较

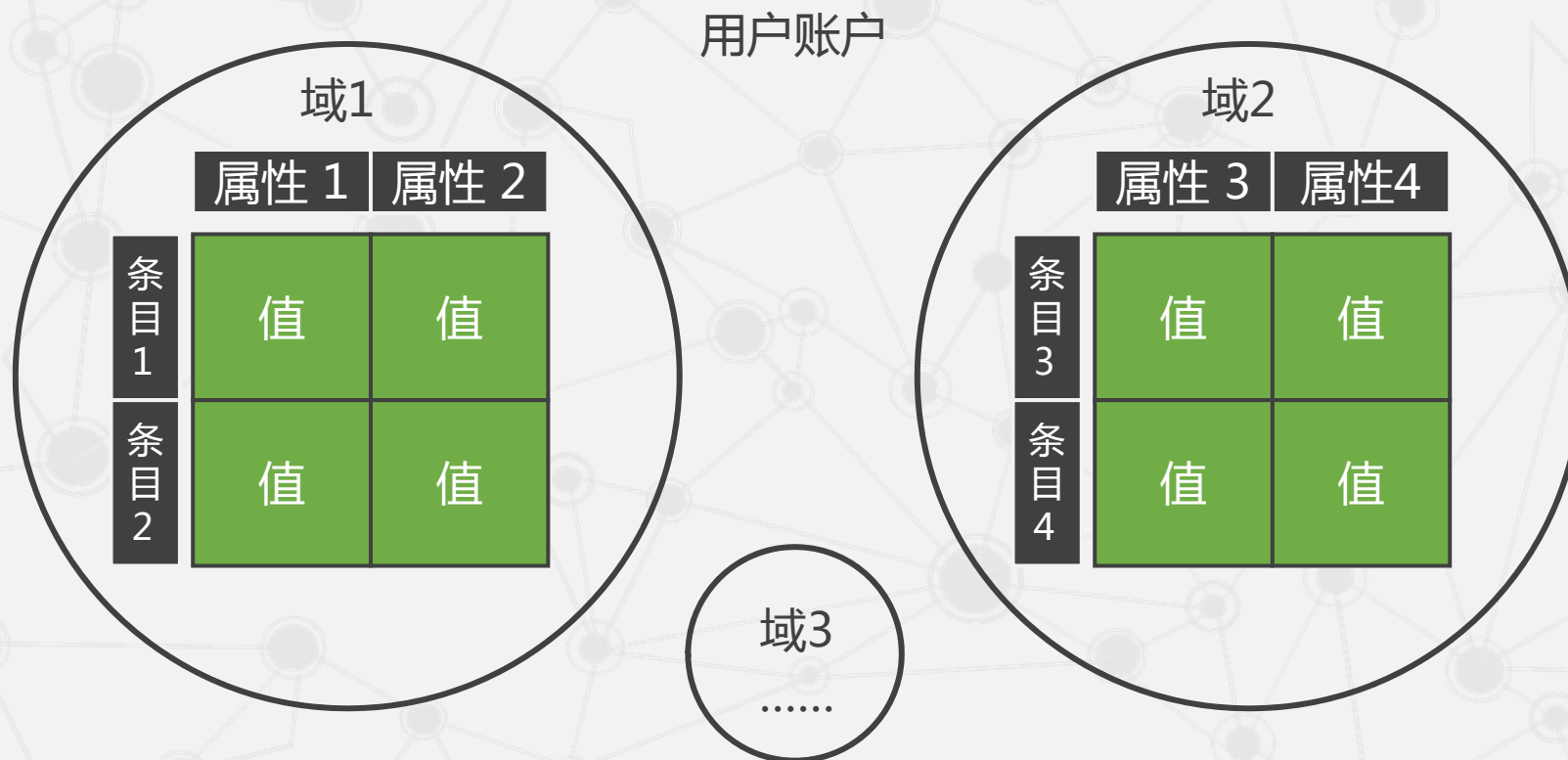
► 3.4.2 SimpleDB

3.4.3 DynamoDB

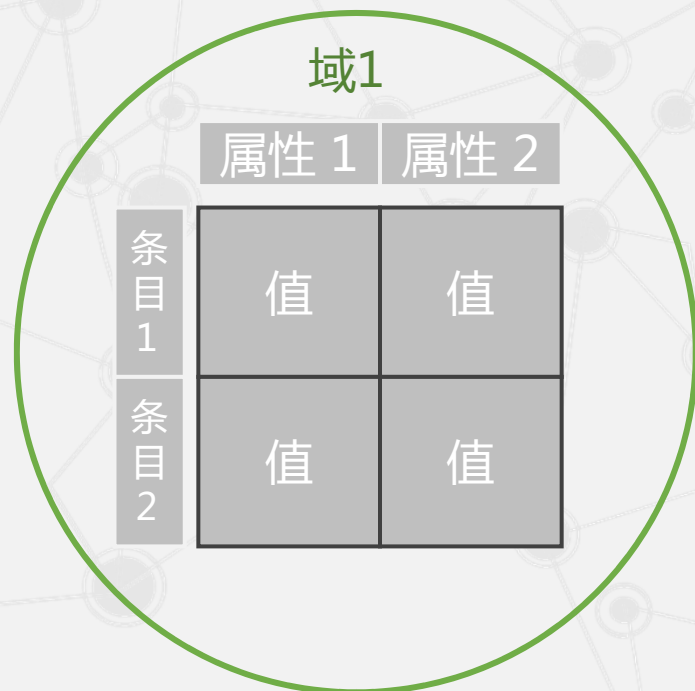
3.4.4 SimpleDB和DynamoDB的比较

- SimpleDB

SimpleDB基本结构图如下，包含了域、条目、属性、值等概念。



1.域 (Domain)



是否划分域需要综合
多种因素考虑

域是用于存放具有一定关联关系的数据的**容器**，其中的数据以**UTF-8编码**的字符串形式存储。

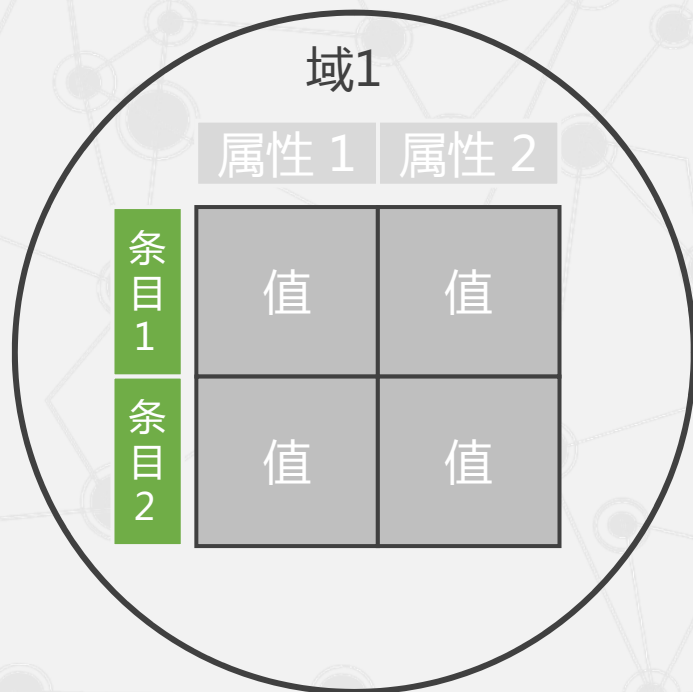
每个用户账户中的**域名**必须是**唯一**的，且域名长度为**3 ~ 255个字符**。

每个域中**数据的大小**具有一定的**限制**，所以通常将**不同特征**的数据放入**不同的域**中。

对于Web数据等不易划分的数据，可以利用**哈希函数**将其散列到不同的域中。

域的划分也会为**数据操作**带来一些**限制**，SimpleDB的数据库操作以域为基本单位，所有的查询只能在一个域内运行，不能在域间进行。

2. 条目 (Item)



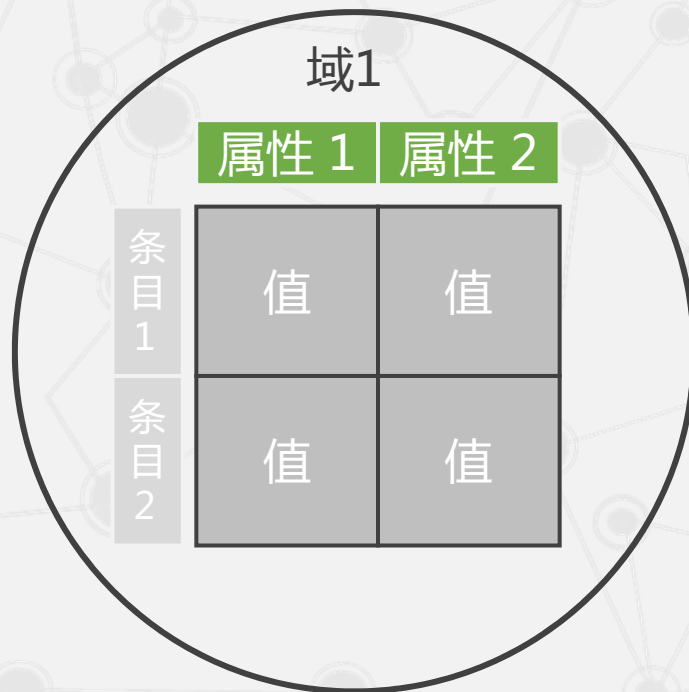
条目对应着一条**记录**，通过一系列属性来描述，即条目是**属性的集合**。

在每个域中，**条目名**必须是**唯一的**

与关系数据库不同，SimpleDB中**不需要事先定义条目的模式**，即条目由哪些属性来描述。

操作上具有极大的**灵活性**，用户可以随时**创建、删除以及修改条目**的内容

3.属性 (Attribute)

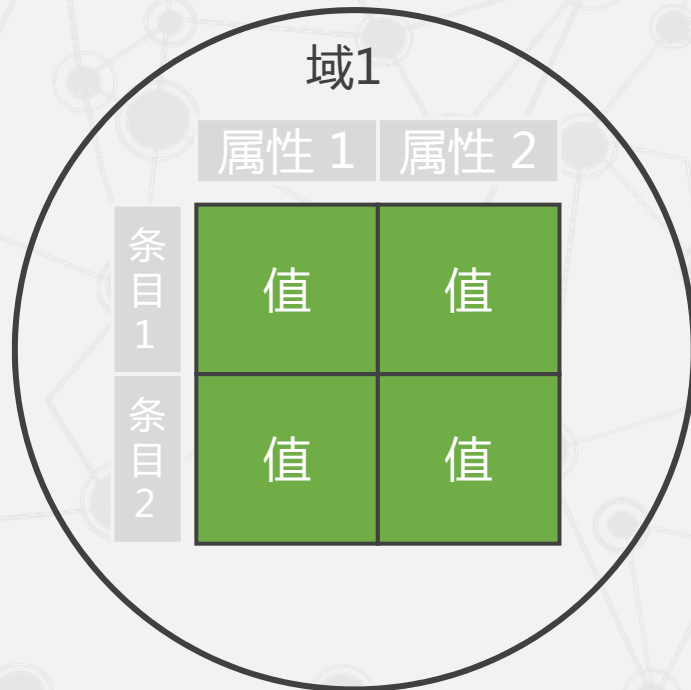


属性是条目的特征，每个属性都用于对条目某方面特性进行概括性描述。

每个条目可以有多个属性。

属性的操作相对自由，不用考虑该属性是否与域中的其他条目相关。

4. 值 (Value)

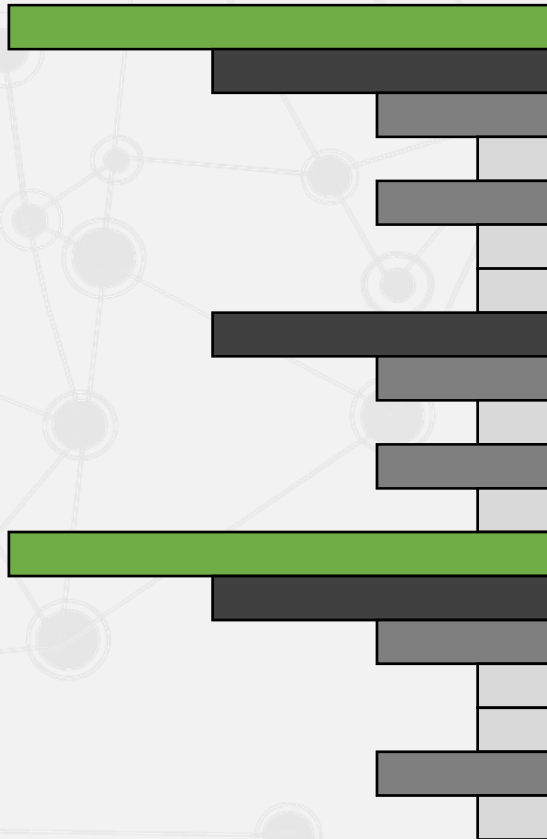


值用于**描述**某个条目在某个**属性上的具体内容**

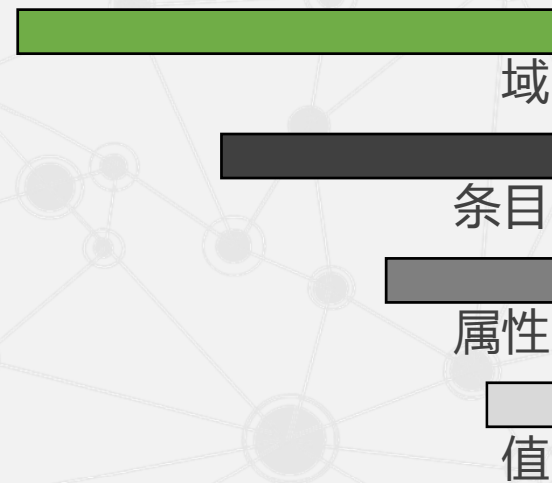
一个条目的一个属性中可以有**多个值**。

例如：

某类商品除颜色外其他参数完全一致，此时可以通过在颜色属性中存放多个值来使用一个条目表示该商品，而不需要像关系数据库中那样建立多条记录。



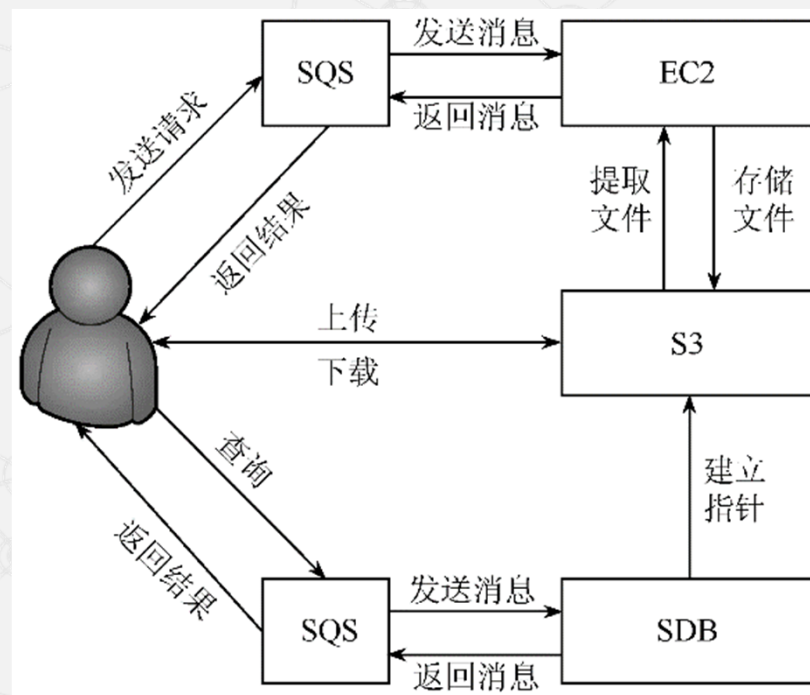
如图显示了SimpleDB的树状组织方式，
其中可以看出SimpleDB对多值属性的支持。



限制 SimpleDB中每个属性值的大小不能超过**1KB**

导致 SimpleDB存储的**数据范围**极其**有限**

解决 将相对大的数据存储在S3中，在SimpleDB中只保存指向某个特定文件位置的**指针**



SimpleDB与其他AWS组件综合使用的方式

3.4 非关系型数据库服务 SimpleDB和DynamoDB

3.4.1 非关系型数据库与传统关系数据库的比较

3.4.2 SimpleDB

► 3.4.3 DynamoDB

3.4.4 SimpleDB和DynamoDB的比较

• DynamoDB

DynamoDB的特点：

1

DynamoDB以**表**为基本单位，表中的条目同样不需要预先定义的模式。

2

DynamoDB中**取消**了对表中**数据大小的限制**，用户设置任意大小，并由系统自动分配到多个服务器上。

3

DynamoDB不再固定使用最终一致性数据模型，而是允许用户选择**弱一致性**或者**强一致性**。

4

DynamoDB还在硬件上进行了优化，采用**固态硬盘**作为支撑，并根据用户设定的**读/写流量**限制预设来确定数据分布的硬盘数量。

3.4 非关系型数据库服务 SimpleDB和DynamoDB

3.4.1 非关系型数据库与传统关系数据库的比较

3.4.2 SimpleDB

3.4.3 DynamoDB

▶ 3.4.4 SimpleDB和DynamoDB的比较

- **SimpleDB和DynamoDB的比较**

SimpleDB和DynamoDB都是Amazon提供的非关系型数据库服务。

SimpleDB

限制了每张表的大小，更适合于小规模复杂的工作。
自动对所有属性进行索引，提供了更加强大的查询功能。

DynamoDB

支持自动将数据和负载分布到多个服务器上，并且
未限制存储在单个表中数据量的大小，适用于较大
规模负载的工作。

目录

3.1 基础存储架构Dynamo

3.2 弹性计算云EC2

3.3 简单存储服务S3

3.4 非关系型数据库服务SimpleDB和DynamoDB

3.5 关系数据库服务RDS

3.6 简单队列服务SQS

3.7 内容推送服务CloudFront

3.8 其他Amazon云计算服务

3.9 AWS应用实例

3.10 小结

3.5 关系数据库服务RDS

► 3.5.1 RDS的基本原理

3.5.2 RDS的使用

非关系数据库在处理ACID（原子性、隔离性、一致性、持久性）类问题时存在一些先天性的不足，为了满足相关应用的需求，Amazon提供了相关数据库服务（Relational Database Service，RDS）

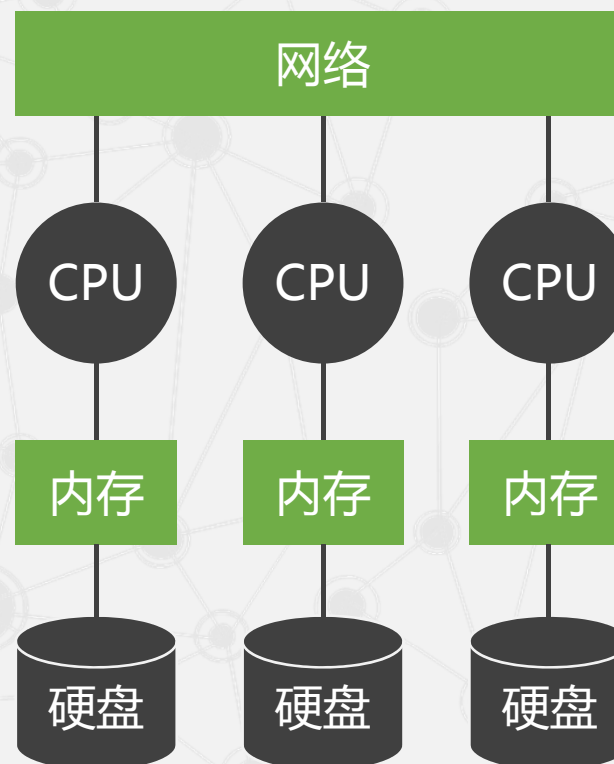
• RDS的基本原理

Amazon RDS将MySQL数据库移植到集群中，在一定的范围内解决了关系数据库的可扩展性问题。

MySQL集群方式采用了**Share-Nothing**架构。

每台数据库服务器都是完全独立的计算机系统，通过**网络相连**，**不共享任何资源**。

这是一个具有**较高可扩展性**的架构，当数据库处理能力不足时，可以通过**增加服务器数量**来提高处理能力，同时多个服务器也增加了**数据库并发访问的能力**。



3.5 关系数据库服务RDS

《云计算》第三版配套PPT课件

集群MySQL通过表单划分的方式将一张大表划分为若干个小表，分别存储在不同的数据库服务器上

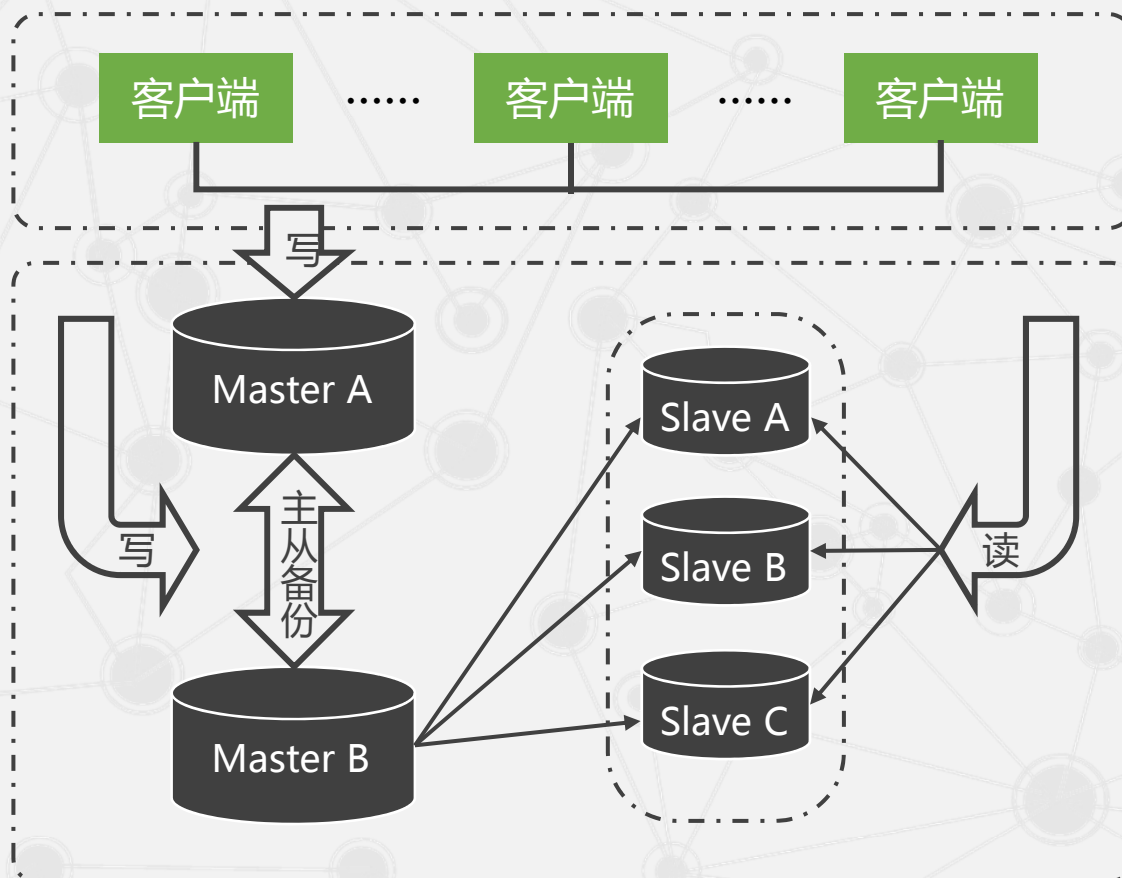
从逻辑上保证了数据库的可扩展性



表单主要根据业务的需要进行针对性的划分，对数据库的管理人员提出了非常高的要求，如果划分得不科学，则查询经常会跨表单和服务器的，性能就会严重下降。

3.5 关系数据库服务RDS

《云计算》第三版配套PPT课件



集群MySQL通过主从备份和读副本技术提高**可靠性**和**数据处理能力**。

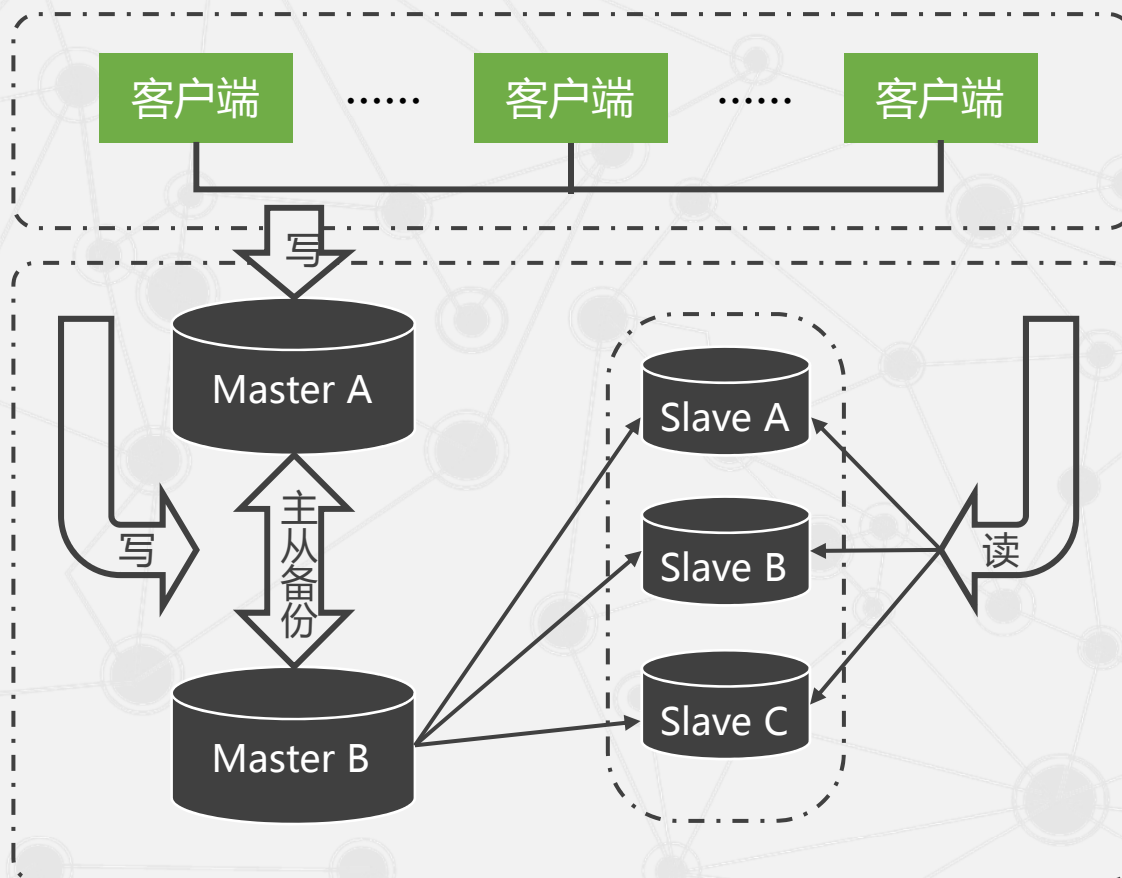
瘫痪

Master A为主数据库，Master B为从数据库，组成主从备份。

如果Master B检测到 Master A瘫痪，则立刻接替Master A的位置，成为主服务器，并会重新创建一台从服务器

3.5 关系数据库服务RDS

《云计算》第三版配套PPT课件



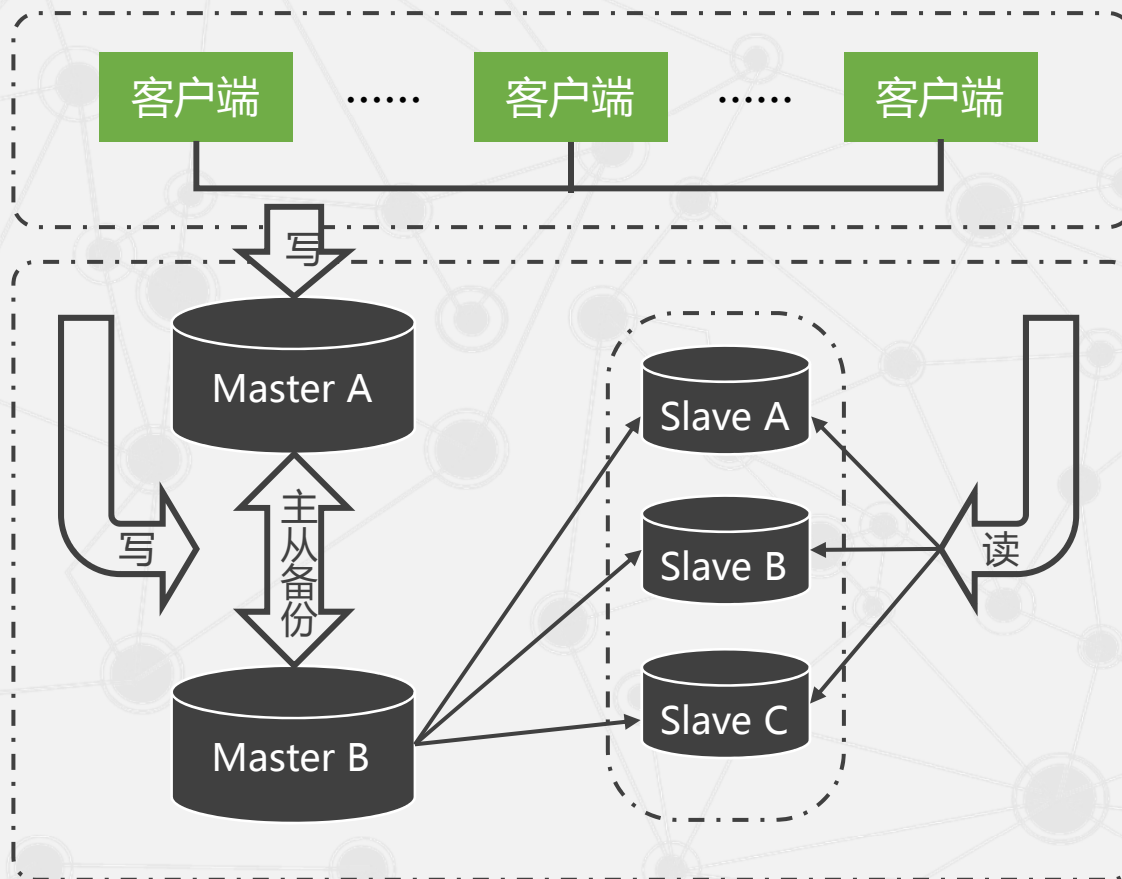
集群MySQL通过主从备份和读副本技术提高**可靠性**和**数据处理能力**。

升级

先对从数据库进行升级，然后将从数据库转变为主数据库，再对新的从数据库进行升级

3.5 关系数据库服务RDS

《云计算》第三版配套PPT课件



集群MySQL通过主从备份和读副本技术提高**可靠性**和**数据处理能力**。

并发处理

提供若干个读副本
(Slave) ——提高数据库的
并发处理能力，
写操作只能由主数据库来完成

3.5 关系数据库服务RDS

3.5.1 RDS的基本原理

▶ 3.5.2 RDS的使用

● RDS的使用

从用户和开发者的角度来看，RDS和一个远程MySQL关系数据库没什么两样。

Amazon将RDS中的MySQL服务器实例称做DB Instance，通过基于Web的API进行创建和管理，其余的操作可以通过标准的MySQL通信协议完成。

创建DB Instance时还需要定义可用的存储，存储范围为5GB到1024GB，RDS数据库中表最大可以达到1TB。

可以通过两种工具对RDS进行操作：命令行工具和兼容的MySQL客户端程序

命令行工具是Amazon提供的Javamazon网站下载。MySQL客户端是可以与MySQL服务器进行通信的应用程序



本章未完待续