第3章 数据库安全

数据库作为企业资源发挥着越来越重要的作用，同时它也成为黑客攻击的主要目标。而如何保证数据库自身的安全，已成为现代数据库系统需要解决的主要问题之一。一个安全的系统需要数据库安全、操作系统安全、网络安全、应用系统自身安全共同完成。

3.1 数据库安全概述

如何有效地保证数据库系统安全，对于实现数据的保密性、完整性和有效性至关重要。主要原因在于：一方面，数据库系统承载关键业务数据，而这些数据牵涉企业各个方面的信息，具有重要的价值；另一方面，数据库系统通常比较复杂，其对连续性、稳定性有高标准的要求，安全管理人员在缺乏相关知识的情况下会使数据库安全管理工作滞后于业务需求。

3.1.1 数据库安全标准

1. 数据库安全问题的产生

数据库的安全性是指在不同层次保护数据库，防止未授权的数据访问，避免数据的泄露、不合法的修改或对数据的破坏。安全性问题不是数据库系统所独有的，它来自各个方面，其中既有数据库本身的安全机制，如用户认证、存取权限、视图隔离、跟踪与审查、数据加密、数据完整性控制、数据访问的并发控制、数据库的备份和恢复等，也涉及计算机硬件系统、计算机网络系统、操作系统、组件、Web 服务、客户端应用程序、网络浏览器等。只是由于在数据库系统中大量数据集中存放，而且为许多最终用户直接共享，从而使安全性问题更为突出，上述每一个方面产生的安全问题都可能导致数据库数据的泄露、意外修改、丢失等后果。

2. 常见的数据库的安全标准

目前，国内外均有数据库安全的等级标准。1991 年，美国国家计算机安全中心（NCSC）颁布了《可信计算机系统评估标准关于可信数据库系统的解释》（Trusted Database Interpretation，TDI）。1996 年，国际标准化组织 ISO 颁布了《信息技术安全技术——信息技术安全性评估准则》（Information Technology Security Techniques—Evaluation Criteria For It Security）。我国政府于 1999 年颁布了《计算机信息系统评估准则》。

目前，国际上广泛采用的是 TCSEC（TDI）标准。在此标准中，将数据库安全划分为 4 大类，由低到高依次为 D、C、B、A。其中，C 级由低到高分为 C1 和 C2，B 级由低到高分为 B1、B2 和 B3。每级都包括其下级的所有特性，各级指标如下。

(1) D 级：无安全保护的系统。

(2) C1 级：只提供非常初级的自主安全保护。能实现对用户和数据的分离，进行自主存取控制(DAC)，保护或限制用户权限的传播。

(3) C2 级：提供受控的存取保护，即将 C1 级的 DAC 进一步细化，以个人身份注册负责，并实施审计和资源隔离。很多商业产品为该级别。

(4) B1 级：标记安全保护。对系统的数据加以标记，并对标记的主体和客体实施强制存取控制(MAC)，以及审计等安全机制。若一个数据库系统符合 B1 级标准，则称为安全数据库系统或可信数据库系统。

(5) B2 级：结构化保护。建立形式化的安全策略模型并对系统内的所有主体和客体实施 DAC 和 MAC。

(6) B3 级：安全域。满足访问监控器的要求，审计跟踪能力更强，并提供系统恢复过程。

(7) A 级：验证设计，即提供 B3 级保护的同时给出系统的形式化设计说明和验证，以确信各安全保护真正实现。

我国国家标准的基本结构与 TCSEC 相似。我国标准分为 5 级，从第 1 级到第 5 级依次与 TCSEC 标准的 C 级(C1、C2)及 B 级(B1、B2、B3)一致。

3.1.2 数据库安全的特征

数据库安全包含两层含义：第一层是指系统运行安全，第二层是指系统信息安全。数据库系统的安全特性主要是针对数据而言的，包括数据独立性、数据安全性、数据完整性、并发控制、故障恢复等几个方面。

1. 数据独立性

数据库系统的数据独立性要靠 DBMS 来实现。到目前为止，物理独立性已经能基本实现，而逻辑独立性实现起来比较困难，数据结构一旦发生变化，一般情况下相应的应用程序都要做一些修改。这也是数据库系统结构复杂的一个重要原因。

2. 数据安全性

一个数据库能否防止无关人员越权访问，是数据库是否安全的一个重要指标。如果一个数据库对所有的人都公开，那么这个数据库就不是一个可靠的数据库。通常，比较完整的数据库会采取以下安全措施。

(1) 将数据库中需要保护的部分与其他部分相隔。

(2) 采用授权规则，如账户、口令和权限控制等访问控制方法。

(3) 对数据进行加密后存储于数据库。

3. 数据完整性

数据完整性包括数据的正确性、有效性和一致性。

(1) 正确性：是指数据的输入值与数据表对应域的类型一样。

(2) 有效性：是指数据库中的理论数值满足现实应用中对该数值段的约束。

(3) 一致性：是指不同用户使用的同一数据应该是一样的。

保证数据的完整性，需要防止合法用户向数据库中加入不合语义的数据。

4. 并发控制

如果数据库应用要实现多用户共享数据,就可能在同一时刻出现多个用户存取数据,这种事件称为并发事件。当一个用户取出数据进行修改,在修改存入数据库之前如有其他用户再读取此数据,那么读出的数据就是不正确的。这时就需要对这种并发操作施行控制,排除和避免这种错误的发生,保证数据的正确性。

5. 故障恢复

如果数据库系统运行时出现物理或逻辑上的错误,系统能尽快地恢复正常,这就是数据库系统的故障恢复功能。数据库管理系统应提供一套方法,及时发现故障和修复故障,从而防止数据被破坏。

3.1.3 数据库的安全层次

数据库系统的安全除了依赖自身的安全机制外,还与外部网络环境、应用环境、从业人员素质等因素息息相关,数据库的安全机制如图 3.1 所示。

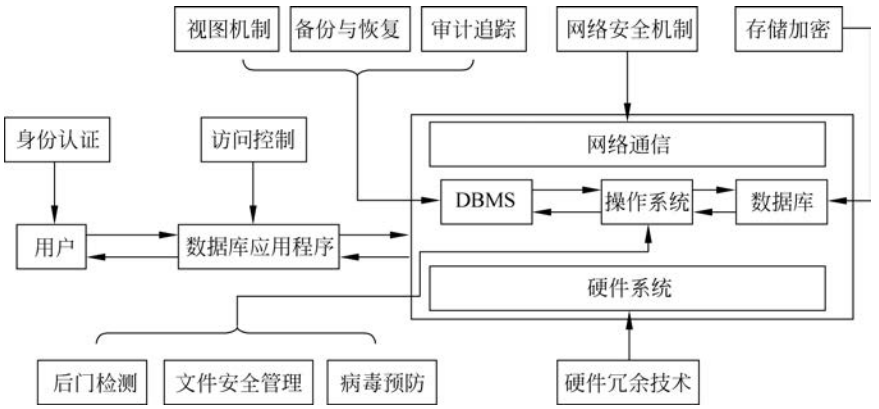

图 3.1 数据库安全机制

从广义上讲,数据库系统的安全框架可以划分为网络系统层、宿主操作系统层和数据库管理系统层三个层次。这三个层次构筑成数据库系统的安全体系,防范的重要性也逐层加强。为了实现三个层次的安全性,必须在物理层面实现:重要的计算机系统必须在物理上受到保护,以防止入侵者强行进入或暗中潜入;在人员层面实现:数据库系统的建立、应用和维护等工作,一定要由可信的合法用户来操作和管理;在操作系统层面实现:进入数据库系统,首先要经过操作系统,如果操作系统的安全性差,数据库将面临重大的威胁;在网络层面实现数据通信的网络安全,因为几乎所有网络上的数据库系统都允许通过终端或网络进行远程访问,所以网络的安全和操作系统的安全一样重要;在数据库系统层面应保证数据库系统有完善的访问控制机制,以防止非法用户操作。

1. 网络系统层次安全

从广义上讲,数据库的安全首先依赖于网络系统。随着 Internet 的发展和普及,越来越多的公司将其核心业务向互联网转移,各种基于网络的数据库应用系统面向网络用户提供各种信息服务。可以说,网络系统是数据库应用的外部环境和基础,数据库系统要发挥其强大作用离不开网络系统的支持,网络系统的安全是数据库安全的第一道屏障,外部入侵首先就是从入侵网络系统开始的。

网络入侵试图破坏信息系统的完整性、机密性和可用性。这些安全威胁无处不在，因此必须采取有效的措施来保障系统的安全。从技术角度讲，网络系统层次的安全防范技术主要有加密技术、认证技术、数字签名技术、防火墙技术、入侵检测技术等。

2. 操作系统层次安全

操作系统是大型数据库系统的运行平台，为数据库系统提供一定程度的安全保护。目前操作系统平台大多数集中在 Windows 和 UNIX，安全级别通常为 C2 级。一个安全的操作系统应该具有访问控制、内存管理、对象重用、审计、加密数据传送、加密文件系统、安全进程间通信机制等功能。主要安全技术有操作系统安全策略、安全管理策略、数据安全等方面。

操作系统安全策略用于配置本地计算机的安全设置，包括密码策略、账户锁定策略、审核策略、IP 安全策略、用户权利指派、加密数据的恢复及其他安全选项。具体可以体现在用户账户、口令、访问权限、审计等方面。

安全管理策略是指网络管理员对系统实施安全管理所采取的方法及策略。针对不同的操作系统和网络环境采取的安全管理策略也不尽相同，其核心是保证服务器的安全和分配好各类用户的权限。

数据安全主要体现在以下几个方面：数据加密技术、数据备份、数据存储的安全性、数据传输的安全性等。可以采用的技术主要有 Kerberos、IPSec、SSL、TLS、VPN 等技术。

3. 数据库管理系统层次安全

数据库系统的安全性很大程度上依赖于数据库管理系统。如果数据库管理系统安全机制强大，那么数据库系统的安全性能就较好。目前市场上流行的是关系模型数据库管理系统，其安全性功能较弱，这就导致数据库系统的安全性存在一定的威胁。

由于数据库系统在操作系统下都是以文件形式进行管理的，因此入侵者可以直接利用操作系统的漏洞窃取数据库文件，或者直接利用操作系统工具来非法伪造、篡改数据库文件内容。数据库管理系统层次安全技术主要是用来解决这一问题，保证在网络安全层次和操作系统安全层次被突破的情况下，仍能保障数据库数据的安全。其采用的主要技术就是通过数据库管理系统对数据库文件进行加密处理，保证即使数据不幸泄露或者丢失，也难以被破译和阅读。

3.2 数据库安全技术

3.2.1 容易忽略的简单漏洞

在所发现的漏洞中，有将近一半的漏洞或直接或间接地与数据库环境内不适当的补丁修复管理有关。在前三个月补丁修复周期内，据统计，只有 40%左右的管理员修复数据库，并且只有三分之一的管理员花费一年或者更长时间进行修复。下面是几种常见的简单漏洞。

1. 默认、空白和强度弱的用户名或者密码

跟踪数百甚至数千个数据库是很艰巨的任务，但是删除默认、空白及强度弱的登录凭证将是完善数据库安全非常重要的第一个步骤。攻击者们总是将注意力放在这些默认账户上。

2. SQL 注入攻击

SQL 注入攻击是黑客攻击数据库的常用手段之一。随着 B/S 模式应用开发的发展,使用这种模式编写应用程序的程序员也越来越多,但是由于程序员的水平及经验参差不齐,相当大一部分程序员在编写代码的时候,没有对用户输入数据的合法性进行有效判断,使应用程序存在安全隐患。用户可以提交一段数据库查询代码,根据程序返回的结果,获得某些他想得知的数据,这就是所谓的 SQL 注入。

如果数据库平台无法对输入内容进行审查,攻击者将能够执行 SQL 注入攻击,就像在 Web 攻击中所做的那样,SQL 注入攻击最终将允许攻击者提升权限,并且获取对更广泛功能的访问权限。

3. 广泛的用户和组特权

必须确保没有将特权分配给那些不必要的用户。只有将用户设置为组或者角色的一部分,然后通过这些角色来管理权限,这样将比向用户分配直接权利要更加易于管理。

4. 启用不必要的数据库功能

每个数据库都会附带很多辅助功能,并且大部分都不会被使用。数据库安全意味着减少攻击面,所以需要审查这些数据库功能,找出不必要或者不使用的功能,然后禁用或者卸载。这不仅能够降低通过这些载体发动攻击的风险,而且能够简化补丁修复管理。

5. 配置管理不完善

数据库有很多不同的配置可供选择,正确合适的配置将能够帮助数据库管理员提高数据库性能和加强数据库功能,而不完善的配置则会带来安全隐患。需要找出不安全的配置(默认情况下为启用状态是方便数据库管理员或者应用程序开发人员而开启的),然后重新进行配置。

6. 特权升级

数据库常常出现这样的漏洞,允许攻击者对鲜为人知或者低权限账号进行权限升级,然后获取管理员权限。

7. 拒绝服务攻击

拒绝服务攻击即攻击者想办法让目标机器停止提供服务,这是黑客常用的攻击手段之一。其实,对网络带宽进行的消耗性攻击只是拒绝服务攻击的一小部分,只要能够对目标造成麻烦,使某些服务被暂停甚至主机死机,都属于拒绝服务攻击。拒绝服务攻击问题一直得不到合理的解决,是因为这是由于网络协议本身的安全缺陷造成的,因此拒绝服务攻击也成为攻击者的终极手法。攻击者进行拒绝服务攻击,实际上让服务器实现两种效果:一是迫使服务器的缓冲区满,不接收新的请求;二是使用 IP 欺骗,迫使服务器把合法用户的连接复位,影响合法用户的连接。

3.2.2 数据库加密技术

对数据库安全性的威胁有时候是来自于网络内部,一些内部用户可能非法获取用户名和密码,或利用其他方法越权使用数据库,甚至可以直接打开数据库文件来窃取或篡改信息。因此,有必要对数据库中存储的重要数据进行加密处理,以实现数据存储的安全保护。数据库加密系统能够有效地保证数据安全,即使黑客窃取了关键数据,仍然难以得到所需的信息。另外,数据库加密以后,不需要了解数据内容的系统管理员不能见到明文,大大提高

了关键数据的安全性。

各用户(或用户组)的数据由用户用自己的密钥加密,数据库管理员无法进行正常解密,从而保证了用户信息的安全。另外,通过加密,数据库的备份内容成为密文,从而能减少因备份介质失窃或丢失而造成的损失。数据库加密对数据库系统效率有一定的影响,如果数据加/解密运算在数据库客户端进行,对数据库服务器的负载及系统运行几乎没有影响。

1. 加密的基本要求

一个良好的数据库加密系统应该满足以下基本要求。

(1) 合适的加密粒度。

不同的加密粒度,其作用和效果不同,常见的加密粒度有以下几种。

① 基于文件的数据库加密技术。

把数据库文件作为整体,对整个数据库文件加密,形成密文来保证数据的真实性和完整性。利用这种方法,数据的共享是通过用户用解密密钥对整个数据库文件进行解密来实现的,但多方面的缺点限制了这一方法的实际应用。首先,数据修改的工作将变得十分困难,需要进行解密、修改、复制和加密四个操作步骤,极大地增加了系统的时空开销;其次,即使用户只是查看某一条记录,也必须将整个数据库文件解密,这样无法实现对文件中不需要让用户知道的信息的控制。

② 基于记录的数据库加密技术。

一般而言,数据库系统中每条记录独立完整地存储了一个实体的数据,因此,基于记录的数据库加密技术是最常用的加密手段。这种方法的基本思路是:在不同密钥的作用下,将数据库的每一个记录加密成密文并存放于数据库文件中;记录的查找是通过将需要查找的值加密成密码文后进行的。然而基于记录的数据库保护有一个缺点,就是在解密一个记录的数据时,无法实现对这个记录中不需要的字段不解密,在选择某个字段的某些记录时,如果不对含有这个字段的所有记录解密就无法进行选择。

③ 基于字段的数据库加密技术。

在目前条件下,最好的加密/解密粒度是字段。如果以文件或列为单位进行加密,必然会造成密钥的反复使用,从而降低加密系统的可靠性和可用性。只有以记录的字段数据为单位进行加解密,才能适应数据库操作,同时进行有效的密钥管理并完成“一次一密”的密码操作。

(2) 密钥动态管理。

数据库客体之间隐含着复杂的逻辑关系,一个逻辑结构可能对应着多个数据库物理客体,所以数据库加密不仅密钥量大,而且组织和存储工作比较复杂,需要对密钥实现动态管理。

(3) 合理处理数据。

合理处理数据包括几方面的内容,首先要恰当地处理数据类型,否则 DBMS 将会因加密后的数据不符合定义的数据类型而拒绝加载;其次,需要处理数据的存储问题,实现数据库加密后,基本上不增加空间开销。在目前条件下,数据库关系运算中的匹配字段,如表间连接码、索引字段等数据不宜加密。

(4) 不影响合法用户的操作。

加密系统影响数据操作响应时间应尽量短,现阶段平均延迟时间不应超过 0.1s。此

外,对数据库的合法用户来说,数据的录入、修改和检索操作应该是透明的,不需要考虑数据的加密/解密问题。

2. 数据库加密层次

数据库数据的加密可在三个不同层次实现,这三个层次分别是 OS、DBMS 内核层和 DBMS 外层。在 OS 层,由于无法辨认数据库文件中的数据关系,从而无法产生合理的密钥,也无法进行合理的密钥管理和使用。所以,在 OS 层对数据库文件进行加密,对于大型数据库来说,目前还难以实现。在 DBMS 内核层实现加密是指数据在物理存取之前完成加密/解密工作,这种方式要求 DBMS 和加密器(硬件或软件)之间的接口需要 DBMS 开发商的支持。这种加密方式的优点是加密功能强,并且加密功能几乎不会影响 DBMS 的功能。其缺点是加密/解密运算在服务器端进行,会加重数据库服务器的负载。

比较实际的做法是在 DBMS 外层加密,DBMS 外层加密是将数据库加密系统做成 DBMS 的一个工具。采用这种加密方式时,加密/解密运算可以放在客户端进行,其优点是不会加重数据库服务器的负载并可实现网上传输加密,缺点是加密功能会受一些限制。

3.2.3 存取管理技术

存取管理技术主要包括用户认证技术和访问控制技术两方面。用户认证技术包括用户身份验证和用户身份识别技术。访问控制包括数据的浏览控制和修改控制。浏览控制是为了保护数据的保密性,而修改控制是为了保护数据的正确性和提高数据的可信性。在一个数据资源共享的环境中,访问控制非常重要。

1. 用户认证技术

用户认证技术是系统提供的最外层安全保护措施。通过用户身份验证,可以阻止未授权用户的访问,而通过用户身份识别,可以防止用户的越权访问。

2. 访问控制技术

访问控制技术是通过某种途径允许或限制用户访问能力及范围的一种方法。访问控制的目的是使用户只能对经过授权的相关数据库进行操作。访问控制从计算机系统的处理功能方面对数据提供保护,是数据库系统内部对已经进入系统的用户的访问控制。它是数据库安全系统中的核心技术,也是最有效的安全手段,限制访问者和程序可以进行的操作,以达到防止安全漏洞隐患的目的。DBMS 中对数据库的访问控制是建立在操作系统和网络的安全机制基础之上的。只有被授权的用户才拥有对数据库中的数据进行输入、删除、修改和查询等权限。通常有以下两种基本的访问控制方法。

(1) 按功能模块对用户授权。

每个功能模块对不同用户设置不同权限,如可设置为无权进入本模块、仅可查询、可更新可查询、全部功能可使用等,而且功能模块名、用户名与权限编码可保存在同一数据库。

(2) 基于角色的访问控制。

通常为了提高数据库的信息安全访问,用户在进行正常的访问前服务器往往都需要认证用户的身份、确认用户是否被授权。为了加强身份认证和访问控制,适应对大规模用户和海量数据资源的管理,通常 DBMS 主要使用的是基于角色的访问控制(Role Based Access Control,RBAC)。所谓“角色”就是一个或一群用户在组织内可执行操作的集合。角色可以根据组织中不同的工作创建,然后根据用户的职责分配,用户可以轻松地进行角色转换。基

于角色的访问控制技术根据用户在组织内所处的角色进行访问授权与控制，只有系统管理员有权定义和分配角色。用户与客体无直接联系，只有通过角色才享有该角色所对应的权限，从而访问相应的客体。

3.2.4　安全审计技术

企业内部人员的违规行为与传统的攻击行为不同，对内部的违规行为无法利用攻击机理和漏洞机理进行分析。因此，要防止内部的违规行为，就需要在内部建设审计系统，通过对操作行为的分析，实现对违规行为的及时响应和追溯。

1. 安全审计的概念

数据库安全审计系统主要用于监视并记录对数据库服务器的各类操作行为，通过对网络数据的分析，实时、智能地解析对数据库服务器的各种操作，并记入审计数据库中以便日后进行查询、分析、过滤，实现对用户操作的监控和审计。安全审计可以监控和审计用户对数据库中的数据库表、视图、序列、包、存储过程、函数、库、索引、同义词、快照、触发器等的创建、修改和删除等，分析的内容可以精确到 SQL 操作语句一级；还可以根据设置的规则，智能地判断出违规操作数据库的行为，并对违规行为进行记录、报警。

一个性能良好的审计系统必须具有以下基本特征。

(1) 能够制定确保系统安全审计策略正确实施的规章制度及措施。

(2) 能够对重要服务器的访问行为进行审计。

(3) 能够包括事件的日期、时间、类型、主体标识、客体标识和结果等。

(4) 能够定期对审计记录进行审查分析，对可疑行为及违规操作采取相应的措施，并及时报告。

2. 常见的安全审计技术

常见的安全审计技术主要有四类，分别是基于日志的审计技术、基于代理的审计技术、基于网络监听的审计技术、基于网关的审计技术。

(1) 基于日志的审计技术。

基于日志的审计技术通常是通过数据库自身功能实现，Oracle、DB2 等主流数据库均具有审计功能。通过配置数据库的自审计功能，即可实现对数据库的审计，其典型部署如图 3.2 所示。

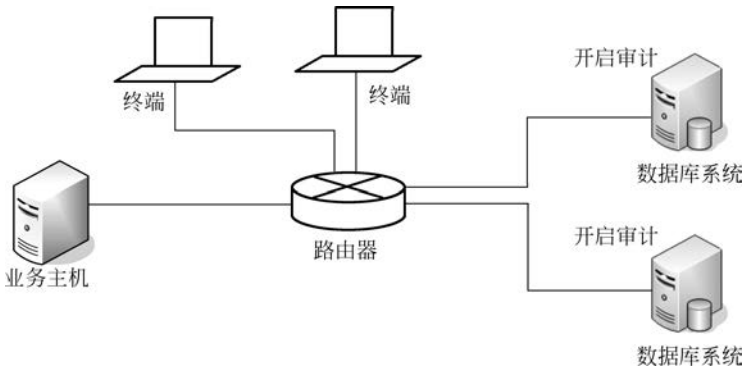

图 3.2　日志审计技术部署

该技术依托于现有数据库管理系统对网络操作及本地操作数据库的行为进行审计,具有很好的兼容性。但这种审计技术的缺点也比较明显,首先,开启自身日志审计对数据库系统的性能有影响,特别是在大流量情况下影响较大;其次,日志审计记录的细粒度差,缺少源 IP、SQL 语句等关键信息,审计溯源效果不好;最后就是日志审计需要到每一台被审计主机上进行配置和查看,较难进行统一的审计策略配置和日志分析。

(2) 基于代理的审计技术。

基于代理的审计技术是通过在数据库系统上安装相应的审计代理(Agent),在代理上实现审计策略的配置和日志的采集,常见的产品如 Oracle 公司的 Oracle Audit Vault,IBM 公司的 DB2 Audit Management Expert Tool,以及第三方安全公司提供的产品,其典型部署如图 3.3 所示。

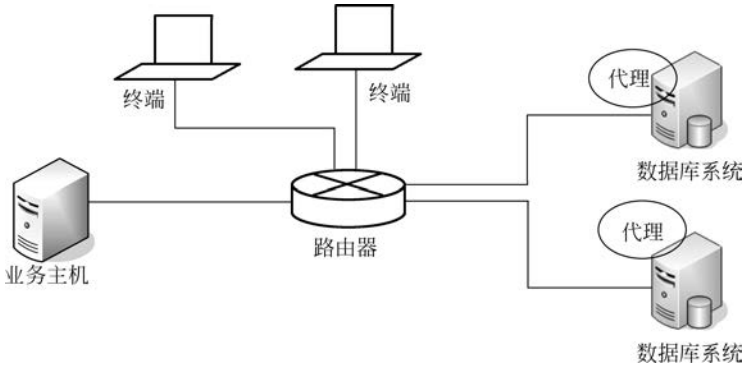

图 3.3　代理审计技术部署

代理审计技术与日志审计技术最大的不同是需要在被审计主机上安装代理程序,其在审计粒度上优于日志审计技术,但在性能上的损耗大于日志审计技术。由数据库厂商提供的代理审计类产品对自有数据库系统具有良好的兼容性,但是在跨数据库系统的支持上,存在一定的兼容风险。同时在引入代理审计后,对原数据库系统的稳定性、可靠性等方面会有一些影响。

(3) 基于网络监听的审计技术。

基于网络监听的审计技术是把对数据库系统的访问流镜像到交换机某一个端口,然后由专用硬件设备对该端口流量进行分析和还原,从而实现对数据库访问的审计,其典型部署如图 3.4 所示。

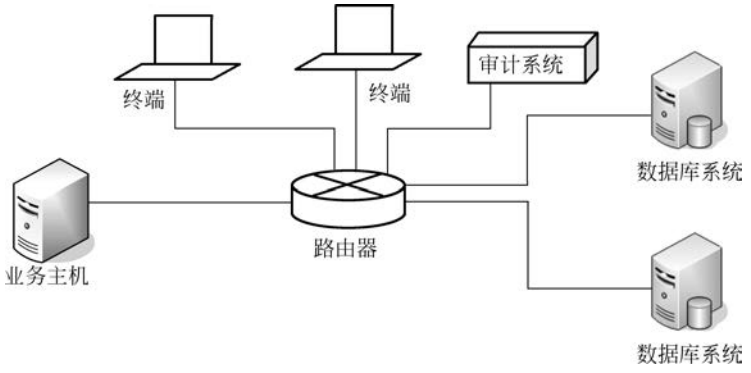

图 3.4　网络监听审计技术部署

虽然在针对加密协议时只能实现到会话级别审计(即可以审计到时间、源 IP、源端口、目的 IP、目的端口等信息),而无法对内容进行审计,但该技术最大的优点就是与现有数据库系统无关,易部署、无风险,部署过程不会给数据库系统带来性能上的负担,故网络监听审计技术在实际的数据库审计项目中应用非常广泛。

(4) 基于网关的审计技术。

基于网关的审计技术是通过在数据库系统前部署网关设备,通过在线截获并转发到数据库的流量而实现审计,其典型部署如图 3.5 所示。

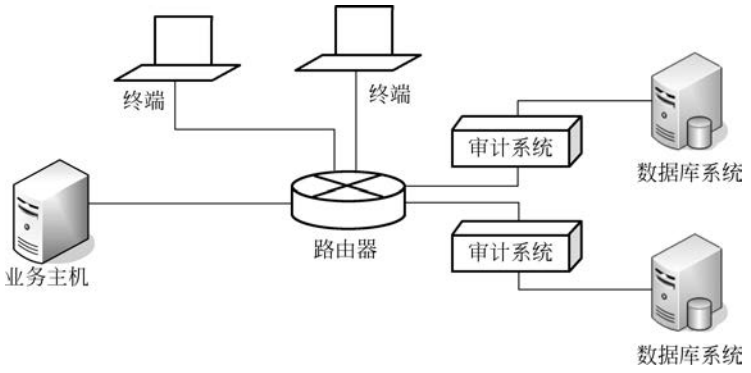

图 3.5　网关审计技术部署

该技术起源于安全审计在互联网审计中的应用,由于数据库环境存在流量大、业务连续性要求高、可靠性要求高的特点,与互联网环境大相径庭,所以网关审计技术主要应用于对数据库运维的审计,而不是对所有针对数据库访问行为的审计。

3.2.5　备份与恢复

计算机在运行过程中会出现多种异常现象,如磁盘故障、电源故障、软件故障、灾害故障及人为破坏等。一旦发生这些故障,就有可能造成数据的丢失。而数据库管理系统的备份和恢复机制能保证数据库系统出现故障时,可以将数据库系统还原到正确状态。

1. 故障的种类

数据库系统中发生的故障大致可以归结为以下几类。

1) 事务故障

所谓事务是用户定义的一个操作序列,这些操作要么全做要么全不做,是一个不可分割的工作单位。事务具有四个特性:原子性、一致性、隔离性和持续性。

原子性是指事务中包括的操作要么都做,要么都不做;一致性保证如果数据库系统在运行中发生故障,有些事务尚未完成就被迫中断,系统将事务中对数据库的所有已完成的操作全部撤销,回滚到事务开始时的一致状态;隔离性保证一个事务的执行不能被其他事务干扰,即一个事务内部的操作对其他并发事务是隔离的;持续性也称永久性,指一个事务一旦提交,它对数据库中数据的改变就应该是永久性的,接下来的其他操作或故障不应该对其执行结果有任何影响。

但事务的四个特性可能由于多个事务并行运行时,不同事务的操作交叉执行或者事务在运行过程中被强行停止等因素而受到破坏,这就是事务故障。事务故障意味着事务没有

达到预期的终点,因此数据库可能处于不正确状态。

2) 系统故障

系统故障是指造成系统停止运转,必须重新启动系统的事件。例如,特定类型的硬件故障、操作系统故障、DBMS 代码错误、数据库服务器出错及其他自然原因等。发生系统故障时,一些尚未完成的事务的结果可能已送入物理数据库,有些已完成的事务可能有一部分甚至全部留在缓冲区,尚未写回到磁盘上的物理数据库中,从而造成数据库可能处于不正确的状态。

3) 介质故障

系统故障常称为软故障,介质故障称为硬故障。硬故障指外存故障,如磁盘损坏、磁头碰撞、瞬时强磁场干扰等。这类故障将破坏数据库全部或部分内容,并影响正在存取这部分数据的所有事务。这类故障比前两类故障发生的可能性小得多,但破坏性最大,有时会造成数据的无法恢复。

4) 计算机病毒

计算机病毒是由一些人恶意编制的计算机程序。这种程序与其他程序不同,它可以像病毒一样进行繁殖和传播,并造成对计算机系统包括数据库系统的破坏。

5) 用户操作错误

在某些情况下,由于用户有意或无意的操作也可能删除数据库中的有用数据或加入错误数据,这同样会造成一些潜在的故障。

2. 数据恢复的基本原理

数据备份与恢复是实现数据库系统安全运行的重要技术。数据库系统免不了发生故障,一旦系统发生故障,重要数据就可能遭到损坏。为防止重要数据的丢失或损坏,数据库管理员应及时做好数据库备份,这样当系统发生故障时,管理员就能利用已有的数据备份,把数据库恢复到原来的状态,以便保持数据的完整性和一致性。

数据库恢复可以通过数据库备份文件、磁盘镜像和数据库在线日志三种方式来完成。

3. 数据备份

数据备份就是数据库管理员(DBA)定期地将数据库复制到其他存储介质上形成备用文件的过程。这些备用的数据文件称为后备副本或后援副本。当数据库遭到破坏后可以将后备副本重新装入,但重装后备副本只能将数据库恢复到转储时的状态,要想恢复到故障发生时的状态,必须重新运行自转储以后的所有更新事务。

例如,在图 3.6 中,系统在 T_a 时刻停止运行事务进行数据库转储,在 T_b 时刻转储完毕,得到 T_b 时刻的数据库一致性副本。系统运行到 T_c 时刻发生故障。为恢复数据库,首先由 DBA 重装数据库后备副本,将数据库恢复至 T_b 时刻的状态,然后重新运行自 T_b 时刻至 T_c 时刻的所有更新事务,这样就把数据库恢复到故障发生前的一致状态。

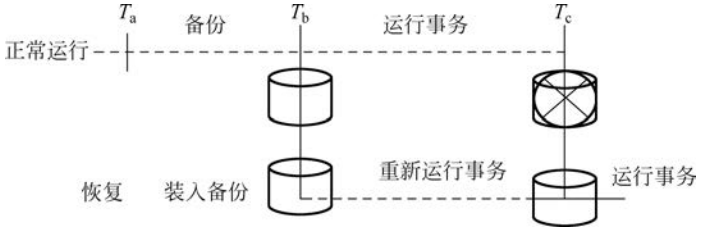

图 3.6 备份和恢复

数据备份十分耗费时间和资源，不能频繁进行。数据库管理员(DBA)应该根据数据库使用情况确定一个适当的转储周期和转储策略。

根据备份过程数据库是否关闭，备份可分为静态备份和动态备份。静态备份是指在备份过程中，系统不运行其他事务，专门进行数据转储工作。而动态备份是指在备份过程中，允许其他事务对数据库进行存取或修改的转储方式。由于动态备份可以动态地进行，这样后备副本中存储的就可能是过时的数据。因此，有必要把转储期间各事务对数据库的修改活动登记下来，建立日志文件(Log File)，使得后备副本加上日志文件能够把数据库恢复到某一时刻的正确状态。

根据备份数据量的多少，备份又可分为海量备份和增量备份。海量备份每次备份全部数据，海量备份能够得到后备副本，利用后备副本能够比较方便地进行数据恢复工作。但对于数据量大和更新频率高的数据库，不适合频繁地进行海量转储。而增量备份每次只备份上一次转储后更新过的数据，增量备份适用于数据库较大但是事务处理又十分频繁的数据库系统。

由于数据备份可在动态和静态、海量和增量下进行，因此数据备份方法可以分为 4 类：动态海量备份、动态增量备份、静态海量备份和静态增量备份。

4. 日志文件

日志文件是用来记录对数据库所进行的所有更新操作的文件。不同的数据库系统采用的日志文件格式基本相同。日志文件能够用来进行事务故障恢复、系统故障恢复，并能够协助后备副本进行介质故障恢复。当数据库文件毁坏后，可重新装入后备副本把数据库恢复到转储结束时刻的正确状态，再利用日志文件，把已完成的更新事务进行重做处理，而对于故障发生时尚未完成的事务则进行撤销处理，这样就可以把数据库恢复到故障前某一时刻的正确状态。

日志文件主要有以记录为单位的日志文件和以数据块为单位的日志文件。

以记录为单位的日志文件需要登记的内容包括：每个事务的开始标记、结束标记和所有更新操作，这些内容均作为日志文件中的一个日志记录。对于更新操作的日志记录，其内容主要包括：事务标识、操作的类型、操作对象、更新前数据的旧值及更新后数据的新值。

以数据块为单位的日志文件包括事务标识和更新的数据块。由于更新前后的各数据块都放入了日志文件，所以操作的类型和操作对象等信息就不放入日志记录。

为保证数据库的可恢复性，登记日志文件时必须遵循两条原则：一是登记的次序严格按事务执行的时间次序；二是必须先写日志文件，后写数据库。

5. 数据库恢复策略

当系统运行过程中发生故障时，利用数据库后备副本和日志文件就可以将数据库恢复到故障前的某个一致性状态。不同故障其恢复策略和方法也不一样。

1) 事务故障的恢复

当发生事务故障时，恢复子系统应利用日志文件撤销(UNDO)此事务已对数据库进行的修改。事务故障的恢复通常是由系统自动完成的，用户感知不到系统是如何进行事务恢复的。

事务故障的恢复步骤大致如下。

① 反向扫描文件日志(即从最后向前扫描日志文件)，查找该事务的更新操作。

② 对该事务的更新操作执行逆操作,即将日志记录中“更新前的值”写入数据库。若记录中是插入操作,则相当于做删除操作;若记录中是删除操作,则做插入操作;若是修改操作,则相当于用修改前的值代替修改后的值。

③ 重复执行①和②,恢复该事务的其他更新操作,直至读到该事务的开始标记,事务故障恢复就完成了。

2) 系统故障的恢复

系统故障恢复操作要撤销故障发生时未完成的事务,重做已完成的事务。系统故障的恢复是由系统在重新启动时自动完成的,不需要用户干预。

系统故障的恢复步骤大致如下。

① 正向扫描日志文件(即从头扫描日志文件),找出在故障发生前已经提交的事务(这些事务既有 BEGIN 或 TRANSACTION 记录,也有 COMMIT 或 ROLLBACK 记录),将其事务标记记入重做(REDO)队列。同时找出故障发生时尚未完成的事务(这些事务只有 BEGIN 或 TRANSACTION 记录,无相应的 COMMIT 或 ROLLBACK 记录),将其事务标记记入撤销(UNDO)队列。

② 对撤销队列中的各个事务进行撤销(UNDO)处理。

进行撤销处理的方法是:反向扫描日志文件,对每个事务的更新操作执行逆操作,即将日志记录中“更新前的值”写入数据库。

③ 对重做队列中的各个事务进行重做(REDO)处理。

进行重做处理的方法是:正向扫描日志文件,对每个重做事务重新执行日志文件登记的操作,即将日志记录中“更新后的值”写入数据库。

3) 介质故障的恢复

介质故障会破坏磁盘上的物理数据库和日志文件,这是最严重的一种故障。恢复方法是重装数据库后备副本,然后重做已完成的事务。

介质故障的恢复步骤大致如下。

① 装入最新的数据库后备副本,使数据库恢复到最近一次转储时的一致性状态。对于动态转储的数据库副本,还需要同时装入转储开始时刻的日志文件副本。利用恢复系统故障的方法(即重做+撤销的方法)将数据库恢复到一致性状态。

② 装入相应的日志文件副本(转储结束时刻的日志文件副本),重做已完成的事务。

利用日志技术进行数据库恢复时,恢复子系统必须搜索所有的日志,确定哪些事务需要重做。

6. 数据库镜像技术

随着磁盘容量越来越大,价格越来越便宜,为避免磁盘介质出现故障影响数据库的可用性,许多数据库管理系统提供了数据库镜像功能用于数据库恢复。数据库镜像技术是用来提高数据库可用性的主要软件解决方案。镜像基于每个数据库实现,并且只适用于使用完整恢复模式的数据库。简单恢复模式和大容量日志恢复模式不支持数据库镜像。

数据库镜像需要两个数据库,一个是主体数据库,另一个是镜像数据库,两个数据库驻留在不同的服务器上。在任何应用时间,客户端只能使用一个数据库,此数据库称为“主体数据库”。客户端对主体数据库进行的更新被同步到“镜像数据库”。

一旦出现介质故障,可由镜像磁盘继续提供服务,同时 DBMS 自动利用镜像磁盘数据

进行数据库的恢复，不需要关闭系统和重装数据库副本。在没有出现故障时，数据库镜像还可以用于并发操作，即当一个用户对数据加排他锁修改数据时，其他用户也可以读镜像数据库上的数据，而不必等待该用户释放锁。

由于数据库镜像是通过复制数据实现的，频繁地复制数据自然会降低系统运行效率，因此在实际应用中用户往往只选择对关键数据和日志文件镜像，而不是对整个数据库进行镜像。

3.3　云数据及其安全

3.3.1　云数据库概述

1. 云数据库

云数据库是在软件即服务(Software-as-a-Service，SaaS)成为应用趋势的大背景下发展起来的云计算技术，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，同时也虚拟化了许多后端功能。云数据库具有高可扩展性、高可用性、采用多组形式和支持资源有效分发等特点。

云数据库简称为“云库”，是部署和虚拟化在云计算环境中的数据库，它把各种关系数据库看成一系列简单的二维表，并基于简化版本的 SQL 或访问对象进行操作。云数据库解决了数据集中与共享的问题，剩下的是前端设计、应用逻辑和各种应用层开发资源的问题。

使用云数据库的用户不能控制运行原始数据库的机器，也不必了解它身在何处，如图 3.7 所示。客户端不需要了解云数据库的底层细节，所有的底层硬件都已经被虚拟化，对客户端而言是透明的。它就像在使用一个运行在单一服务器上的数据库一样，方便、容易，同时又可以获得理论上近乎无限的存储和处理能力。

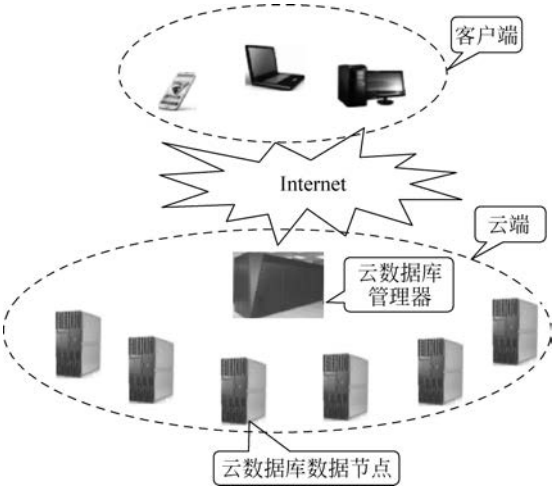

图 3.7　云数据库应用示意图

2. 云数据库的特性

云数据库具有以下特性。

1) 动态可扩展

理论上，云数据库具有无限可扩展性，可以满足不断增加的数据存储需求。在面对不断

变化的条件时,云数据库表现出很好的弹性,可以动态地调整资源以适应需求的变换。

2) 高可用性

在云数据库中,数据通常是复制的,在地理上也是分布的,所以不存在单点失效问题。如果一个节点失效了,剩余的节点就会接管未完成的事务。例如 Google、Amazon 和 IBM 等大型云计算供应商都具有分布在世界范围内的数据中心,通过在不同地理区间内进行数据复制,提供高水平的容错能力。Amazon SimpleDB 会在不同的区间内进行数据复制,因此,即使整个区域内的云设施发生失效,也能保证数据继续可用。

3) 低代价

用户使用云数据库时,通常采用多租户(Multi-Tenancy)的形式,这种共享资源的形式可以节省开销。而且用户采用按需付费的方式使用云计算环境中的各种软、硬件资源,不会产生资源浪费。另外,云数据库底层存储通常采用大量廉价的商业服务器,这也大幅度降低了用户开销。

4) 易用性

使用云数据库的用户不必控制运行原始数据库的机器,也不必了解它身在何处。用户只需要一个有效的链接字符串就可以开始使用云数据库。

5) 大规模并行处理

支持几乎实时的面向用户的应用、科学应用和新类型的商务解决方案。

3. 云数据库与传统的分布式数据库的区别

分布式数据库是计算机网络环境中各场地或节点上的数据库的逻辑集合。逻辑上它们属于同一系统,而物理上它们分散在用计算机网络连接的多个节点,并统一由一个分布式数据库管理系统管理。

云数据库和传统的分布式数据库具有相似之处,例如,都把数据存放到不同的节点上。但是,分布式数据库在可扩展性方面是无法与云数据库相比的。由于需要考虑数据同步和分区失败等开销,后者随着节点的增加会导致性能快速下降。而前者则具有很好的可扩展性,因为前者在设计时就已经避免了许多会影响到可扩展性的因素。另外,在使用方式上,云数据库也不同于传统的分布式数据库。云数据库通常采用多租户模式,即多个租户共用一个实例,租户的数据既有隔离又有共享,解决了数据存储问题,也降低了用户使用数据库的成本。

4. 云数据库的影响

1) 数据存储的变革

云数据库把以往数据库中的逻辑设计简化为基于一个地址的简单访问模型。但为了满足足够的带宽和数据容量,物理设计就显得更为重要。从应用成本和容错的角度分析,Google 和 Amazon 采用分散文件集群。分散文件既可能是运行在某个有完善管理数据中心的 SAN 集群,也可能是运行在某些老旧服务器上的磁盘塔。尽管存储效率不同,但对于云数据库而言,保存在它们之上的数据只要可以按照客户的要求保质保量交付就可以。

2) 极大地改变企业管理数据的方式

对于中小企业而言,云数据库可以允许他们在 Web 上快速搭建各类数据库应用,越来越多的本地数据和服务将逐渐被转移到云中。企业用户在任意地点通过简单的终端设备,

就可以对企业数据进行全面的管理。此外，云数据库可以很好地支持企业开展一些短期项目，降低开销，而不需要企业为某个项目单独建立昂贵的数据中心。但是对于大企业而言，云数据库并非首选，因为大企业通常会自己建造数据中心。

3）催生新一代的数据库技术

云模型提供了海量处理能力及大量的 RAM，因此，云模型将会极大地改变数据库的设计方式，将会出现第三代数据库技术。第一代是 20 世纪 70 年代的早期关系数据库；第二代是 20 世纪 80 年代至 20 世纪 90 年代的更加先进的关系模型；第三代的数据库技术，要求数据库能够灵活处理各种类型的数据，而不是强制让数据去适应预先定制的数据结构。事实上，从目前云数据库产品中的数据模型设计方式来看，已经有些产品（比如 SimpleDB、Hbase、Dynamo、BigTable）放弃传统的行存储方式，而采用键/值存储，从而可以在分布式的云环境中获得更好的性能。

3.3.2 现有的云数据库产品

就目前而言，虽然一些云数据库产品，如 Google BigTable、SimpleDB 和 HBase，在一定程度上实现了对于海量数据的管理，但是这些系统还不完善，只是云数据库的雏形。

1. Amazon 的云数据库产品

1）Dynamo

Dynamo 采用“键/值”存储非结构化数据，需要用户自己完成对值的解析。Dynamo 系统中的键（key）不以字符串的方式进行存储，而是采用 MD5_key（通过 MD5 算法转换后得到）的方式存储，因此，它只能根据 key 去访问，不支持查询。

2）SimpleDB

SimpleDB 是 Amazon 公司开发的一个可供查询的分布数据存储系统，它是 Dynamo “键/值”存储的补充和丰富，主要是服务于那些不需要关系数据库的 Web 开发者。顾名思义，SimpleDB 的目的是作为一个简单的数据库来使用，它的存储元素是由一个 id 字段来确定行的位置。这种结构可以满足用户基本的读、写和查询要求。SimpleDB 提供易用的 API 来快速地存储和访问数据。

3）Amazon RDS

Amazon RDS（Amazon Relational Database Service）是 Amazon 公司开发的一种 Web 服务，它可以让用户在云环境中建立、操作关系数据库（目前支持 MySQL 和 Oracle 数据库）。用户只需要关注应用和业务层面的内容，而不需要在烦琐的数据库管理工作上耗费过多的时间。

2. Google 的云数据库产品

1）Google BigTable

Google BigTable 是 Google 为了处理内部大量的结构化及半结构化数据而建立的一种满足弱一致性要求的大规模数据库系统。BigTable 构建在其他几个 Google 基础设施之上：首先，BigTable 使用了分布式 Google 文件系统 GFS（Google File System）来存储日志和数据文件；其次，BigTable 依赖一个高可用的、持久性的分布式锁服务 Chubby；再次，BigTable 依赖一个簇管理系统来调度作业，在共享机器上调度资源，监督机器状态。

目前，许多 Google 应用都是建立在 BigTable 上，如 Web 索引、Google Earth、Google

Finance、Google Maps 和 Search History。BigTable 提供的简单数据模型,允许客户端对数据部署和格式进行动态控制,并且描述了 BigTable 的设计和实现方法。

但是,与 Amazon SimpleDB 类似,BigTable 实际上还不是真正的 DBMS,它无法提供事务一致性、数据一致性。这些产品基本上可以被看成云环境中的表单。

2) Fusion Tables

Google 开发的另一款云计算数据库产品是 Fusion Tables。它采用了基于数据空间的技术。Fusion Tables 是一个与传统数据库完全不同的数据库,可以弥补传统数据库的很多缺陷。例如通过采用数据空间技术,它能够简单地解决 RDBMS 中管理不同类型数据的麻烦,以及排序整合等常见操作的性能问题。Fusion Tables 可以上传 100MB 的表格文件,同时支持 CSV 和 XLS 格式,并且具有处理大规模数据的能力。

3. Microsoft 的云数据库产品

2008 年,微软通过 SQL Data Service(SDS)提供 SQL Server 的 RDBMS 功能,这使得微软成为云数据库市场上的第一个大型数据库厂商。此后,微软对 SDS 功能进行了扩充,并且重新命名为 SQL Azure。微软的 Azure 平台提供了一个 Web 服务集合,可以允许用户通过网络在云中创建、查询和使用 SQL Server 数据库,云中的 SQL Server 服务器的位置对于用户而言是透明的。

SQL Azure 具有以下特性。

(1) 属于关系数据库。支持使用 TSQL(Transact Structured Query Language)来管理、创建和操作云数据库。

(2) 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。

(3) 支持大量数据类型。包含几乎所有典型的 SQL Server 2008 的数据类型。

(4) 支持云中的事务。支持局部事务,但是不支持分布式事务。

4. 其他云数据库产品

1) HBase 和 Hypertable

HBase 和 Hypertable 利用开源 MapReduce 平台 Hadoop,提供了类似于 BigTable 的可伸缩数据库实现。MapReduce 是 Google 开发的、用来运行大规模并行计算的框架。采用 MapReduce 的应用更像一个人提交的批处理作业,但是这个批处理作业不是在单个服务器上运行,应用和数据都是分布在多个服务器上。Hadoop 是由 Yahoo 资助的一个开源项目,是 MapReduce 的开源实现,从本质上来说,它提供了一个使用大量节点来处理大规模数据集的方式。

HBase 已成为 Apache Hadoop 项目的重要组成部分,并且已经在生产系统中得到应用。Hypertable 与 HBase 类似,不过,HBase 的开发语言是 Java,而 Hypertable 则采用 C/C++开发。相比于 HBase,Hypertable 具有更高的性能。

2) Relational Cloud

麻省理工学院研制的 Relational Cloud 可以自动区分负载的类型,并把类型近似的负载分配到同一个数据节点上,而且采用了基于图的数据分区策略,对于复杂的事务型负载也具有很好的可扩展性。此外,它还支持在加密的数据上运行 SQL 查询。

3.3.3 云数据库安全策略

1. 云数据库的缺陷和风险

1) 数据的传输问题

虽然在概念上云数据库与传统数据库的应用流程差别不大,但基于互联网的云数据库已远超出了用户的控制范围,因此在实际执行效率、服务响应质量方面增加了很多不确定因素。

2) 数据安全问题

用户对于云数据库安全最关心的是怎么相信云数据库提供商,以及云数据库提供商的内部工作人员会不会利用数据去干非法行为。比如个人隐私被泄露或者网上购物的购买行为被记录等,对于用户来讲,这都侵犯了用户的隐私。对于企业的核心数据来说,就绝对没那么简单。目前比较成熟的云服务商业模式大多数还是云服务提供商本身是内容提供商,企业把核心业务直接迁移至公共云端的成功的案例有限。这会成为制约未来云计算发展的一个重要障碍。

3) 云数据库安全问题

云计算的应用实践较短,技术上还不成熟,导致云环境数据库存在安全方面的缺陷。其安全问题主要分为数据访问控制与遵守规章制度两个基本类别。

2. 基本安全策略

1) 数据库审计

审计数据库产生的审计跟踪,可以指定哪些对象被访问或改变,它们是如何改变的,以及何时何人是否授权访问的时候,审计特别重要。当然,数据库审计的弱点在于它跟踪所有已经发生的行为。理想的情况下,基于云的数据库安全解决方案应具有入侵检测等功能。在出现数据丢失或者数据被盗前,对可疑的活动进行识别。围绕数据库审计的另一个值得关注的问题是性能下降。因为审计需要将有用的细节都以日志的形式记录下来并存储,会降低云数据库的性能。

除了用软件审计数据库外,还可以通过可信的第三方进行审核,发现数据库和环境的脆弱性,包括云环境。第三方审计人员可以用一些专门的工具,如 AppDetectivePro 来识别和处理一些常见的安全问题,如数据库软件是否已经打补丁,是否配置在最安全的方式;默认密码是否已经被修改;访问数据的用户是否是根据企业的安全策略进行访问;在同一个环境(开发、QA 和生产)下的所有机器都有相同的配置,是否有同级别的保护。

2) 访问控制

访问控制模型有 3 个类别:强制访问控制(Mandatory Access Control,MAC)、基于格的访问控制(Lattice-Based Access Control,LBAC)、基于角色的访问控制(Role-Based Access Control,RBAC)。

(1) 强制访问控制由系统的访问策略决定,不由雇主决定。MAC 在多层次的系统中用来处理敏感数据,如政府机密和军事情报。一个多层次的系统是单一的计算机系统,管理主体和客体之间的多个分类级别。

(2) 基于格的访问控制作为基于标签的访问控制限制能够应用复杂的访问控制决策,涉及较多的对象与科目。格模型是一个偏序集,描述了最大的上限下限,并至少是一对元

素,如主题和对象。格是用来描述一个对象的安全水平,可能有一个主题。若主体的安全级别大于或等于对象的主题时,一个主题只能允许有一个访问对象。

(3) 基于角色的访问控制是限制系统访问授权用户。RBAC 的核心概念是权限和角色相关联,用户被分配到合适的角色。在一个组织中不同的工作是由不同的角色来完成的,以用户职责为基础分配用户角色。用户能够实现从一个角色转换到另一个角色。角色可以授予新的权限,根据需要权限也可以从角色中撤销。

3) 隔离敏感数据库

有效的云数据库首先应隔离所有包含敏感数据的数据库。如 DBProtect 的数据库发现功能,生成部署云范围内的所有数据库的完整清单。它确定了所有的生产、测试和临时数据库。DBProtect 帮助组织和云供应商确保敏感数据位于授权与安全的数据库。

4) 数字水印技术

传统的密码技术由于对数据加密实施了置乱处理,因而容易引起攻击者的攻击,同时解密后无法提供有效方式保证数据不被非法复制、再次传播及恶意篡改。所以在云计算的条件下是不能采用传统安全方法的。数字水印技术通过在数字载体中植入可感知或者不可感知的信息来确定数字产品的所有权或检验数字内容所具有的原始性。因而,在云端的数据库安全中应用数字水印技术能够较好地解决云数据库中版权、泄密及可逆水印等问题,能够给予数据拥有者可靠的、鲁棒的云数据库安全解决方案。

5) 安全认证

双重安全机制包括对访问者进行身份验证,以及访问控制列表两项内容。身份验证通常包括密码认证、证物认证及生物认证 3 类,其中最常见的应用方式为密码认证,密码认证与后两种认证方式相比来说,其安全性较差,在破解后就失去了保护屏障。

生物认证在 3 类认证中成本最高,正因为如此,证物认证的使用率呈增加的趋势,具有代表性的例子就是 IC 卡与银行使用的 USB Key。除此以外,控制列表是确保网络资源不被非法用户访问的主要策略,其主要方式包括入网访问控制、网络权限控制与属性控制等诸多的方式,通过控制列表的应用实现对数据源地址、目的地址及端口号等特定指示条件拒绝非法数据,用户在对云环境数据库的资源实现访问前,都应通过登录认证的方式确保其具有的合法性。

习　　题

一、填空题

1. 数据库系统的安全特性主要是针对数据而言的,包括_____、数据安全性、数据完整性、并发控制、_____等几方面。

2. 存取管理技术主要包括_____和访问控制技术两方面。

3. 常见的安全审计技术主要有 4 类,分别是:_____、基于代理的审计技术、基于网络监听的审计技术、_____。

4. 事务具有 4 个特性:_____、_____、隔离性和_____。

5. _____就是数据库管理员(DBA)定期地将整个数据库复制到其他存储介质(如磁带或非数据库所在的另外磁盘)上保存形成备用文件的过程。

6. _____能够用来进行事务故障恢复、系统故障恢复，并能够协助后备副本进行介质故障恢复。

7. 数据库镜像需要两个数据库，一个是_____，另一个是镜像数据库。

8. _____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。

二、选择题

1. 数据的完整性是指(　　)。

A. 数据的存储与使用数据的程序无关　　B. 防止数据被非法使用

C. 数据的正确性、一致性　　D. 减少重复数据

2. 数据库系统运行过程中，由于应用程序错误所产生的故障通常称为(　　)。

A. 设备故障　　B. 事务故障　　C. 系统故障　　D. 介质故障

3. 一个事务的执行，要么全部完成，要么全部不做，一个事务中对数据库的所有操作都是一个不可分割的操作序列的属性是(　　)。

A. 原子性　　B. 一致性　　C. 独立性　　D. 持续性

4. 表示两个或多个事务可以同时运行而不互相影响的是(　　)。

A. 原子性　　B. 一致性　　C. 独立性　　D. 持续性

5. 事务的持续性是指(　　)。

A. 事务中包括的所有操作要么都做，要么都不做

B. 事务一旦提交，对数据库的改变是永久的

C. 一个事务内部的操作对并发的其他事务是隔离的

D. 事务必须是使数据库从一个一致性状态变到另一个一致性状态

6. 日志文件是数据库系统出现故障以后，保证数据正确、一致的重要机制之一。下列关于日志文件的说法错误的是(　　)。

A. 日志的登记顺序必须严格按照事务执行的时间次序进行

B. 为了保证发生故障时能正确地恢复数据，必须保证先写数据库后写日志

C. 检查点记录是日志文件的一种记录，用于改善恢复效率

D. 事务故障恢复和系统故障恢复都必须使用日志文件

7. 对数据库中的数据进行及时转储是保证数据安全可靠的重要手段。下列关于静态转储和动态转储的说法正确的是(　　)。

A. 静态转储过程中数据库系统不能运行其他事务，不允许在转储期间执行数据插入、修改和删除操作

B. 静态转储必须依赖数据库日志才能保证数据的一致性和有效性

C. 动态转储需要等待正在运行的事务结束后才能开始

D. 对一个 24h 都有业务发生的业务系统来说，比较适合采用静态转储技术

三、简答题

1. 什么是数据库的安全性？什么是数据库的完整性？两者之间的联系与区别是什么？

2. 数据库安全性保护通常采用什么方法？

3. 试述事务的概念及事务的 4 个特性。

4. 数据库运行中可能产生的故障有哪几类？哪些故障影响事务的正常执行？哪些故障破坏数据库数据？

5. 数据库恢复的基本技术有哪些？
6. 数据库备份的意义是什么？试比较常见数据备份方法。
7. 什么是日志文件？为什么要设立日志文件？
8. 针对不同的故障，试给出恢复的策略和方法。
9. 什么是数据库镜像？它有什么用途？
10. 云数据库具有哪些特点？
11. 为了保证云数据库的安全，可采取哪些基本的安全策略？