文章编号：2096-1472(2016)-01-44-04

SaaS多租户数据管理及实现策略

殷伟凤
（浙江传媒学院电子信息学院，浙江 杭州 310018）

摘 要：云计算是通过互联网以服务的形式为客户提供企业级计算资源的技术。最普遍使用的服务是软件即服务(SaaS)。许多SaaS提供者利用多租户模式来托管应用。多租户是一种架构方法，软件应用程序的单实例服务多个租户，因此多租户设计关键要解决的是租户数据的共享与隔离。文章论述了多租户架构中数据存储管理的三种方式，详细描述了共享表时多租户数据库的具体实现策略，指出了各种模式映射技术的优缺点，最后给出了将来的研究目标。

关键词：多租户；模式映射技术；软件即服务；数据管理
中图分类号：TP391 **文献标识码**：A

Data Management and Implementation Strategy for Multi-tenant Application

YIN Weifeng
(*Dept.of Electronic Information,Zhejiang University of Media and Communications,Hangzhou* 310018,*China*)

Abstract: Cloud computing is a technology which provides enterprise-grade computing resources as services to customers through internet.One of the popularly available services is software as a service (SaaS).Many of the SaaS providers make use of a multitenant model to host their applications.Multitenancy is an architectural approach in which a single instance of a software application serves multiple tenants.Data sharing and isolating issues is a challenge for implementation of multi-tenant application.In this paper a short survey of Multi-Tenant Data Architecture was presented.The paper introduced three approaches to managing multi-tenant data and concluded some techniques on design and implement multitenant database schema,finally,it identifies a goal of on-going work.

Keywords:multi-tenant;schema-mapping techniques;software as a service;data management

1 引言(Introduction)

云计算已经成为最重要的计算技术。云计算是一个能够方便地按需对可配置计算资源(例如网络、服务器、存储、应用程序和服务)的共享池进行网络访问的模型[1]。目前在云计算范式中最重要的服务是软件即服务(SaaS)，SaaS是软件的一种新型的云计算服务交付模式，可通过互联网以“按需服务”的形式为多个用户提供应用程序。企业通过订购SaaS服务，无需购买和维护自己的IT设施就可使用各类IT服务，减少了软硬件、网络、系统维护的费用。而对于服务提供商，则通过发挥SaaS的规模效应来降低综合使用成本。如Saleforce.com、Goole.com、Alisoft.com等都是SaaS应用的典型成功案例。SaaS应用程序最重要的需求是多租户的支持[2]，为了最大化SaaS的规模效应，一般采用的都是多个租户共享一个运行实例的架构(Multi-Tenant架构，即多租户架构)。

多租户是SAAS业务模型领域一种较新的软件架构，在此架构模式中，允许多个租户共享硬件资源以及应用程序和数据库，并可根据租户需求配置应用程序[3]。可配置性是多租户模式的重要特征。在多租户应用中，所有的租户都使用同一个数据库实例，因此必须确保每个租户只能访问它们各自的数据，因此数据隔离成为多租户应用中最为关键的问题[4]。

2 多租户数据存储方案(Data storage schema for muti-tenant)

SaaS区别于传统技术的重要差别就是多租户模式，多租户架构是SaaS应用的基本特性，也是实现SaaS规模效应的基本要素。多租户就是多个租户共用一个实例，租户的数据既有隔离又有共享，因此多租户设计的关键是如何解决数据存储问题。

2.1 多租户数据存储方案

目前管理多租户数据主要有三种方法[5]：独立数据库、共享数据库和独立数据模式、共享数据库和共享数据模式。

独立数据库方式是指一个租户一个数据库，如图1所示，这是进行数据隔离最简单的方法，该方案用户数据隔离级别最高，安全性最好，但成本也高。该方案的优点是：为不同的租户提供独立的数据库，有助于简化数据模型的扩展设

计，满足不同租户的独特需求；如果出现故障，恢复数据比较简单。该方案的缺点是：增大了数据库的安装数量，随之带来维护成本和购置成本的增加。

图1 独立数据库

Fig.1 Separate database

共享数据库和独立数据模式是指多个或所有租户共享数据库，但一个租户一个数据模式。如图2所示。该方案的优点是：为安全性要求较高的租户提供了一定程度的逻辑数据隔离；每个数据库可以支持更多的租户数量。缺点是：如果出现故障，数据恢复比较困难，因为恢复数据库将牵扯到其他租户的数据；如果需要跨租户统计数据，存在一定困难。

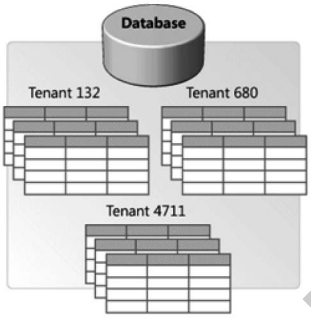


图2 共享数据库，独立模式

Fig.2 Shared database,separate schema

共享数据库和共享数据模式是指租户共享同一个数据库，同一个模式，但在表中通过租户ID区分租户的数据，如图3所示。这是共享程度最高、隔离级别最低的模式。该方案的优点是：维护和购置成本最低，允许每个数据库支持的租户数量最多。缺点是：隔离级别最低，安全性最低，需要在设计开发时加大对安全的开发量；数据备份和恢复最困难，需要逐表逐条备份和还原。

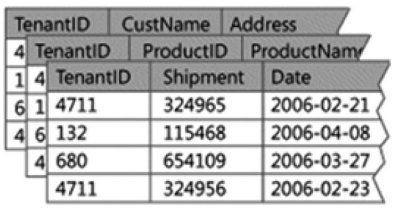
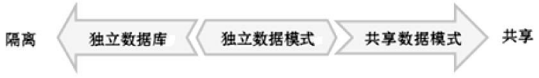


图3 共享数据库，共享模式

Fig.3 Shared database,shared schema

2.2 三种方案的比较及选择

从隔离和共享两个相反的方向比较，依次是独立数据库、共享数据库和独立数据库、共享数据库和共享数据模式，三种方案比较如图4所示，三种方法都有各自的优缺点。选用时可以从经济性、安全性、租户以及技能等因素去考虑。



模式	隔离级别	共享级别	安全性	成本
独立数据库	高	低	高	高
共享数据库，独立数据模式	中	中	中	中
共享数据库，共享模式	低	高	低	低

图4 三种数据存储模式比较

Fig.4 Compare for three kinds of dada storage schema

因为开发共享结构相对比较复杂，在初始开发时需比使用独立方法设计应用程序付出更大的开发工作，但每个服务器可支持更多的租户，持续的运营成本更低。因此如果无法为构建一个共享模式应用提供足够的开发支持或需要尽快使应用上市而不能进行大规模开发，那么必须更多的考虑独立的方法。

如果应用程序存储了敏感的租户数据，客户都会对安全性有较高的预期，需要提供强大的数据安全性保障服务水平协议(SLA)。通常依靠物理隔离可以提供较好的安全级别。使用共享方法存储数据也可以提供强大的数据安全，但需要使用更复杂的设计模式。

服务的租户数量、属性和需求也是确定不同方式数据架构的因素。租户越多越要多考虑共享方法。如果各租户需存储大量的数据，独立数据库方法可能会更好。需要支持每个租户的并发终端用户数量越大，独立方法将越适合满足终端用户的需求。如果希望为每个租户提供增值服务，如每个租户的备份和恢复能力，这样的服务通过独立的方法更容易提供。

设计单实例多租户架构仍然是很新的技能，缺乏现成的专业技能。如果设计师和职员没有足够的构建SaaS应用的经验，他们需要获得必要的知识，或者必须雇佣有经验的人员。在某些情况下，独立方法比共享方法可更多地利用传统软件开发的现有知识。

3 多租户数据库实现(Realizing multi-tenant database)

与上述三种数据存储管理相对应的有三种实现多租户数据库的方法[6]：共享主机、共享进程和共享表。

在共享机器的方法中，每个客户都有自己的数据库进程并且多个客户共享同一个主机。此方法无需修改数据库的实现，基本上不会降低客户隔离度。但该方法不是池式内存，每个数据库在每个应用服务器上都需要有自己的连接池，套接字无法在客户间共享。

在共享进程方法中，每个客户都有自己的表并且多个客户共享同一个数据库进程。此方法更有利于池式内存，可方便进行每个服务器的客户数扩展。客户间可以共享连接池。

共享表方法对于池式资源是最合适的。扩展能力仅受限于数据库支持的行的数目，比共享进程方法要提高几个数量级。客户能共享连接池，可以成批执行管理操作。但该方法由于在磁盘上的文件是来自多个租户的混合数据，迁移比较困难。另外混合数据分布在许多页会影响访问客户数据的性能，共享连接池和数据使安全性成为最关键的问题。

为了实现多租户，大多数托管服务使用查询变换把应用程序中的多个单租户逻辑模式映射到数据库中一个多租户物理模式。这种方法会降低服务器的性能，改进的方法是在租户间共享表，但这种技术可能会影响租户扩展应用程序的能力。最灵活的解决方案是将逻辑表映射到固定的通用结构，如通用表和透视表。

实现共享表的多租户数据库可采用多种存储模型，这些模型称为模式映射技术[7]，主要有基本布局、私有表布局、扩展表布局以及通用表布局、透视表布局、Chunk表布局等通用结构。

(1)基本布局

实现多租户的最基本的技术是每个表增加一个租户ID列(Tenant)，所有的租户共享此表。这种方法是从服务提供者的角度而不是租户角度来看待数据，提供了较好的合并但不具有扩展性，传统Web应用程序大都采用此方法。

(2)私有表布局

支持扩展性最基本的方法是每个租户设置各自的私有表。在此方法中，查询转换层只需要重命名表名，非常简单。此方法每个租户都有不同的业务需求，需要有大量的表来满足每个租户的需求，因此该技术适用于较少数量的租户。

(3)扩展表布局

可以结合上述两种方法，扩展成不同的表。将源表分成基表和扩展表两部分，多个租户可以使用同样的基表，扩展表和基表都需要一个Tenant列，还要增加一个Row列。这种方法比私有表提供了更好的合并性，但表的数量也会与租户数成比例增加。

(4)通用表布局

通用结构允许创建任意形状任意数量的表。通用表是一个有Tenant列、table列和许多通用数据列的通用结构表。数据列设置成一个灵活的类型，如VARCHAR类型，其他的类型可转换为该类型。每个租户的每个逻辑源表的第n列映射到通用表的第n个数据列，因此不同的租户可以以不同的方式扩展同一个表。该种方法的缺点是数据表较宽，数据库必须要处理许多空值，另外对索引支持不是太方便。

(5)透视表布局

透视表是一个可选的通用结构，在透视表中，逻辑源表中的每一行的每个字段都对应着一行。除了上述描述的Tenant、Table和Row列，透视表还有一个Col列指明了这一行所表示的是源表中的哪个字段以及一个数据列表示出那个字段的值。数据列可以给定一个灵活的类型，如VARCHAR，其他类型也可以转换为此类型。此方法无需处理许多空值，可以较好地支持索引，但增加了元数据，增加了运行开销，连接操作较费时。

(6)Chunk表布局

第三类通用表称之为Chunk表。Chunk表类似于透视表，但有一组不同类型的数据列，col列被Chunk列取代，一个逻辑源表被划分为多组列，每组设置一个chunk_ID，将一组列映射到一个Chunk表中。相比透视表，该方法减少了存储元数据，降低了重构逻辑源表的开销。

(7)Chunk Folding

这是一种将源表垂直划分成块放入不同物理多租户表的一种技术，在需要的时候可进行连接。该技术将最常使用的租户列映射到传统表中，而其他列放在Chunk表中，数据库的“元数据开销”分摊给了专用的传统表和一组称为CHunk表的通用结构。应用Chunk Folding方法，参考文献[8]中提出了基于Chunk Folding的自适应多租户缓存管理机制，该机制以租户的SLA需求作为驱动，依据租户当前访问模式，动态生成缓存单元集并计算缓存单元集的I/O效益，通过贪婪算法来选择缓存单元集，使得租户SLA得到满足的同时最小化缓存资源的消耗。

4 XML支持实现多租户数据库(XML for realizing multi-tenant database)

XML和关系数据库是两种完全不同的技术集，XML支持层次数据模型，而数据库支持关系数据模型。关系数据库的缺点是在数据库层缺乏对租户概念的支持，因此，必须在通用表的行存储租户ID。在参考文献[9]中提出了一种针对多租户应用的混合模式共享技术，该方法由两张表组成，一张用于租户的通用内容，如ID、名称、联系方式等，如表1所示。

表1 通用表

Tab.1 Universal table

Manager ID INT	Username VARCHAR	Full Name VARCHAR	Contact INT	County ID INT
1	Ramachan	Ramachandra Nayak	0831456721	2
2	Ashutosh	AshutoshAgarwal	084178654	1
3	priya123	Priyanka Choudary	0832567842	3

另一张扩展表针对每一个租户。扩展表由两行组成：一是租户的ID，另一行包含了一个描述某一个租户的其他信息的XML文档，如表2所示。每个租户都有其存储空间存储私有数据。该方法采用了将扩展表与XML文档相结合的基本思想。

表2 XML扩展表

Tab.2 XML extension table

Manager ID	Extension XML
1	<Parameters><Hospital name=" Apollo" > <Beds>120</Beds> </Hospitalname> </Parameters>

当租户进行查询时，在运行时都通过附加一个租户ID来执行，以XML形式存储数据的优点是：特定租户数据的查询不会因要避免空值而受到干扰，可对租户字段进行索引因而可对每一个租户进行查询优化。

5 结论(Conclusion)

多租户是SaaS软件的一种架构模式，可在多个租户间共享单应用实例，最大化SaaS的规模效应。多种模式映射技术应用于多租户应用中可解决数据共享和数据隔离，但如何更好地构建多租户架构，设计数据管理服务，提高性能并保证安全性，仍然是一个很大的挑战。下一步的主要工作是设计适合实际应用环境的模式映射技术算法。

参考文献(References)

[1] Peter Mell,Timothy Grance.The NIST definition of cloud computing[J/OL].http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf,September 2011.

[2] Guo J.,Sun W.,Huang Y.,Wang Z.,Gao B.A Framework for Native Multi-Tenancy Application Development and Management[C].in Proceedings of The 9th IEEE International Conference on E-Commerce Technology and the 4 th IEEE International Conference on Enterprise Computing,Tokyo,Japan,July 23-26,2007:551-558.

[3] Bezemer C.,Zaidman A.Multi-tenant SaaS applications: maintenance dream or nightmare?[C].Proceedings of the Joint ERCIM Workshop on Software Evolution(EVOL)and International Workshop on Principles of Software Evolution (IWPSE):88-92.

[4] Chang Jie Guo,Wei Sun,Ying Huang,Zhi Hu Wang,Bo Gao. A framework for native multi-tenancy application development and management[C].In Proc. Int.Conf.on E-Commerce Technology(CEC)&Int.Conf.on Enterprise Computing, E-Commerce,and E-Services(EEE):551-558.

[5] G.C.Frederick Chong,R.Wolter.Multi-Tenant Data Architecture[J/OL].https://msdn.Microsoft.com/en-us/library/aa479086.aspx,2006.

[6] Dean Jacobs,Stefan Aulbach.Ruminations on multi-tenant databases[C].In Datenbanksysteme in Business,Technologie und Web (BTW),12.Fachtagung des GI-Fachbereichs Datenbanken und Informationssysteme (DBIS),Proc.7.-9.März,volume 103 of LNI,GI,2007:514-521.

[7] S.Aulbach,T Grust,D.Jacobs,A.Kemper,J.Rittinger.Multitenant databases for software as a service:Schema mapping techniques[C]. Proceedings of the 34th Intetnational Conference on Management of Data(SIGMOD),2008:1195-1206.

[8] 姚金成,等.基于Chunk Folding的多租户数据库缓存管理机制[J].计算机学报,2011,34(12):2319-2331.

[9] F.S.Foping,I.M.Dokas,J.Feehan,S.Imran.A new hybrid schema-sharing technique for multitenant applications[C].ICDIM, 2009:1-6.

作者简介：

殷伟凤(1967-)，女，硕士，副教授.研究领域：计算机应用.