

基于 SAAS 模式的共享模型的数据 扩展技术研究

唐圣潘¹,周肆清¹,丁长松²

(1. 中南大学 信息科学与工程学院,湖南 长沙 410083;

2. 吉首大学 数学与计算机科学学院,湖南 吉首 416000)

摘要:提出一种优良的基于 SAAS 模式的共享数据模型的多租户数据扩展方法。首先简单地介绍了 SAAS 模式及其数据模型,并详细阐述了目前基于 SAAS 模式的共享数据模型的多租户数据扩展技术的优缺点;最后,提出一种以 XML 为基础的共享数据模型的多租户数据扩展方法,并且详细介绍了在此方法下灵活的数据操作及其性能分析,并将其与目前常用的基于 SAAS 模式的共享数据模型的多租户数据扩展方法进行对比分析,可得在 SAAS 系统中应用基于 XML 的共享数据模型的多租户数据扩展方法是性能优良的。

关键词:多租户; SAAS; 数据模型; XML

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2011)08-0063-04

Data Extension Technology Research Based on Shared Model of SAAS Pattern

TANG Sheng-pan¹, ZHOU Si-qing¹, DING Chang-song²

(1. School of Information Science and Engineering, Central South University, Changsha 410083, China;

2. Mathematics and Computer Science College, Jishou University, Jishou 416000, China)

Abstract: It aims to propose a preferred multi-tenant data extension method based on shared data model of SAAS pattern. It first explains simply the SAAS pattern and data model of SAAS pattern, describes in detail advantages and disadvantages of multi-tenant data extension technology based on the current shared data model of SAAS pattern; Finally propose a XML-based multi-tenant data extension method based on shared data model of SAAS pattern, and describe in detail flexible data operation and performance evaluation under this method. It's preferred to apply XML-based multi-tenant data extension method based on shared data model, compared with normal multi-tenant data extension method based on shared data model of SAAS pattern at present.

Key words: multi-tenant; SAAS; data model; XML

0 引言

随着 Internet 技术的迅猛发展,将软件作为一种新的服务形式提供给客户的需求量逐年增加,而 SAAS^[1,2] (Software as a Service, 软件即服务)作为一种新型软件服务形式的出现正是顺应了这个需求。它是一种颠覆传统的软件服务形式,将使软件供应商与客户之间的关系发生彻底转变,从简单的售卖关系转变为服务关系。

SAAS 服务面向互联网上的所有租户,每个租户

根据自身的实际情况,所要求的数据结构各不相同,SAAS 软件开发商们在搭建基于共享模型的 SAAS 架构^[3]时就不得不考虑各租户之间数据结构的差异性问题。针对各租户所要求的数据结构的不同,在进行系统数据库设计时就要对各租户不同的数据结构实施相应的数据扩展策略。

1 SAAS 数据模型

在设计基于 SAAS 模式^[4]的系统数据模型^[5,6]时,出于降低开发成本和接受服务的客户量等考虑,在数据的隔离、共享之间取得一定的平衡是一个必须考虑的重要因素。就一般而言,SAAS 系统的数据模型有如下三种形式:

(1) 独立数据库。

将每个客户的数据单独存放在一个独立数据库

收稿日期:2011-01-08;修回日期:2011-04-14

基金项目:湖南省软件学课题(2009ZK3046)

作者简介:唐圣潘(1984-),男,硕士研究生,研究方向为中间件技术、数据库应用技术、SAAS 应用;周肆清,副教授,研究方向为中间件技术、数据库应用技术、SAAS 应用。

中,使各客户间的数据完全隔离,最大限度地保证了客户数据的安全。

(2) 共享数据库、单独模式。

所有客户共享数据库,但各自有一套独有的数据表来存放各个客户的数据。这在各客户数据的隔离和共享之间取得了一定的平衡。

(3) 共享数据库、共享模式。

所有客户共享一个数据库和同一套数据表。该模式下的一个数据表里可以包含多个客户的数据,由客户 ID 来区分数据归属于哪个客户。该模型具有投入成本低等特点,而且每台数据库服务器可以支持最大的客户量;但是由于所有客户的数据存放在同一个数据表中,因此可能需要花费更多额外的成本来保证客户之间的数据安全^[7]。

2 共享模型的多租户数据扩展方案

在这种模式下,所有的客户共享数据库、共享表结构,所有客户的数据存放在同一个数据表中,通过客户 ID 来区分不同客户的数据。该模式的数据库服务器硬件和数据备份成本最低,它允许每个数据库服务器所支持的客户量达到最多。但是由于所有客户共同使用一个表,在可扩展性、可配置性上产生了瓶颈。

为了解决这个问题,通常有以下三种办法。

2.1 定制列

即使用固定扩展集^[5],指在表中除了各租户共有的一些字段外,还包括各租户各自独有的一些字段。如:可设计 Table (TenantID, FixedCol1, ExtendColu1, ExtendColu2... ExtendColun), 其中 TenantID, FixedCol1 是固定字段;ExtendColu1, ExtendColu2... ExtendColun 是扩展字段。

这种方法不需要处理复杂的数据扩展跟踪,但随着租户的增加,每个租户要求添加的列就很多,但特定租户扩展的数据列对于其他租户是没有任何意义的,严重地破坏了表的结构,并且提供的扩展有限,有时扩展字段中的字段可能为空,给表空间带来了巨大浪费。

2.2 预分配字段

该方法在表格中提供一定预设数量的预设^[8]字段,当客户需要扩展数据时,从表中选取适量的预设字段来扩展数据,但每个客户选取的预设字段的含义可能不同。如表 1 中 F1、F2、F3 就是预分配的字段。

在表 1 中, TenantID 字段将每条记录与租户相关联。除了一组标准字段外,还提供一些预设字段,预设

字段的使用由租户决定,预设字段的数据类型可以不同,一般采用字符串数据类型,并使用元数据^[9]来跟踪其真实数据类型,如表 2 所示。

表 1 预分配字段

TenantID	...	Name	BirthDate	F1	F2	F3
784	...	李四	1980-05-15	123	Null	null

表 2 元数据跟踪表

TableID	F1_Label	F1_Data Type
001	“金额”	Int

该方案虽然满足客户数据的可扩展、可配置性,但在给定的数据表中,预设字段的数量具有不确定性,有的客户需求多,有的客户需求少。如果预设字段的数量设得过大,就会浪费空间;设得过小,又不满足所有客户的需求。

2.3 名称值对

名称值对将所有用户异构的数据(扩展的数据)放在一个扩展表中,并在主数据表中有一个字段与扩展表相关联,并且用元数据表中的元数据来跟踪相应扩展字段的标记和数据类型。该方法使客户自己能够对数据模型进行延伸。元数据表存储各个客户定制字段的信息,包括字段名称和数据类型。数据表、扩展表和元数据表间关系如图 1 所示。

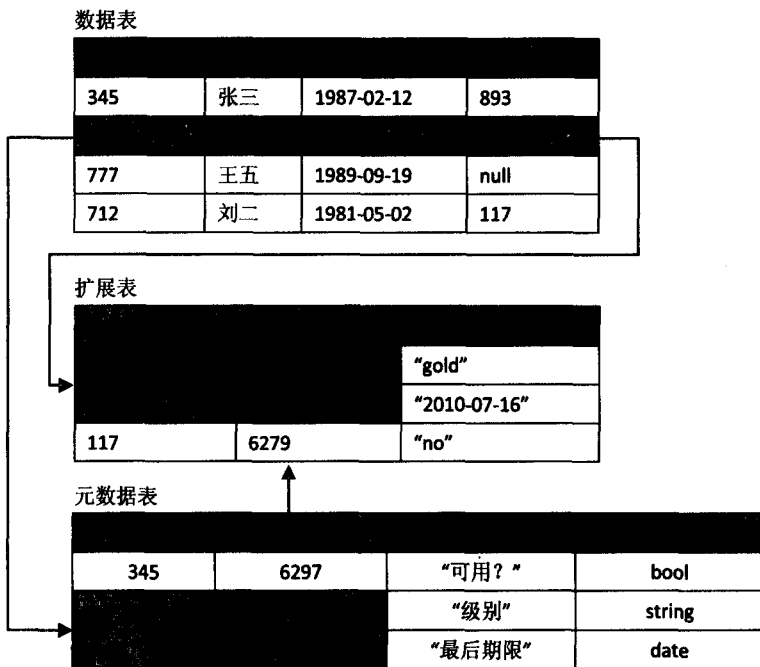


图 1 数据表、扩展表、元数据表间关系

该方法能最大限度地满足所有客户的无限扩展需求,客户能够自行决定数据的可扩展、可配置性,又保持了在该数据模型下的成本优势。虽然这种结构可以方便地扩展无数个字段,但增加诸如索引、查询以及更新记录等数据库功能的复杂程度。

3 基于 XML 的共享模型的多租户数据扩展方案

在共享模型的 SAAS 系统中,所有客户共享表结构,但租户间的数据结构是不同的,将异构的数据存入到固定的表,需要对异构的部分数据进行处理。而 XML^[10] 数据中数据结构是不尽相同的、是异构的,但可以通过 XML 技术灵活地管理 XML 数据中的节点。因此,可利用 XML 数据的特性并采用相应的技术来处理各租户间异构的数据。

该方案是指在表中采用一种基于 XML 的数据模型字段来处理各客户间异构的数据。现在主流大型关系数据库系统都支持对 XML 数据的存储和管理。Oracle 从 9.2 开始就支持一种新的数据类型 (XMLType),用于存储和管理 XML 数据,并提供了很多的函数,用来直接读取 XML 文档和管理节点。

下面以 Oracle 数据库系统为例。表结构如:TableName (TenantID, Colu1, Colu2, Colu3, XMLDataField),其中 TenantID, Colu1, Colu2, Colu3 字段是所有客户共用的字段;XMLDataField 字段存储各客户独有的数据,其格式遵循 XML 的格式要求。设计 XMLDataField 字段格式为:

```
<UserExtendColS>
<UserExtendCol1 colName=" " dataType=" ">value1
</UserExtendCol1>
<UserExtendCol2 colName=" " dataType=" ">value2
</UserExtendCol2>
.....
<UserExtendColn colName=" " dataType=" ">valueN
</UserExtendColn>
</UserExtendColS>
```

在上面 XMLData 字段格式中,每个节点代表一个扩展列,该扩展列可以有多种属性,如:colName、dataType 等,以及该扩展列的值。通过这种方法可以实现灵活的数据查询、更新等操作,满足客户数据的无限扩展需求。

1) 利用关系数据库系统 (ORACLE) 支持对 XML 数据的存储和管理,实现对 XML 类型数据的增加、删除、查找、修改功能。建立一个含 XMLType 类型字段 (XMLData) 的数据表 User,下面详细介绍对该字段及字段里面的节点数据进行增加、删除、查找、修改操作。

(1) 向 XMLData 字段插入数据。

```
Insert into User (XMLData) value (
sys.xmlType.createXML('
<UserExtendColS>
<UserExtendCol colName="age" dataType="int">20
</UserExtendCol>
<UserExtendCol colName="sex" dataType="bool">true
```

```
</UserExtendCol>
</UserExtendColS>
'));
```

注:用 createXML 函数往 XMLData 字段里插入 XMLType 类型数据。

(2) 向 XMLData 字段里面的数据节点里加入节点。

```
Update User set XMLData = insertChildXML ( XMLData, '/
UserExtendColS/UserExtendCol',XMLType(' <UserExtendCol colName="birthday" dataType="datetime">2001-01-05 </UserExtendCol>'));
```

注:XMLType 提供了 insertChildXML 函数在 XML 数据节点最后追加一个节点,还提供了 insertChildXMLBefore, appendChildXML 两个函数分别表示在某个节点前追加节点以及在节点末尾追加多个节点。

(3) 查询 XMLData 字段里面的内容。

```
Select u. XMLData. Extract ( '/UserExtendColS/UserExtendCol
[ @ colName="age" ]/text() ) . getStringVal() as age from User
u;
```

注:Extract 函数返回一个 XML 文档的一个节点树,或者某一节点下所有符合条件的节点。

(4) 更新 XMLData 字段里面的数据。

```
Update User set XMLData = updateXML ( XMLData, '/UserEx-
tendColS/UserExtendCol[ @ colName="age" ]/text()', "20")
Where existsNode ( XMLData, "/UserExtendColS/UserExtendCol[ @
colName="age" ]") = 1;
```

注:使用 updateXML 函数来更新 XMLData 字段里面某个节点的数据;existsNode() 函数来判断是否存在扩展列名为 age 的列,该函数的返回值只有 1 (存在) 和 0 (不存在)。

(5) 删除 XMLData 字段里面的数据节点。

```
Update User set XMLData = deleteXML ( XMLData, '/UserEx-
tendColS/UserExtendCol[ @ colName="age" ]/') Where exists-
Node ( XMLData, '/UserExtendColS/UserExtendCol[ @ colName="
age" ]') = 1;
```

注:使用 deleteXML 函数来删除 XMLData 字段里面某个节点。

2) 性能分析。

在基于 XML 的共享模型的多租户数据扩展方案中,利用 XML 来处理多租户各用户之间异构的数据,对整个 SAAS 应用的体系架构和性能将产生很大的影响。

(1) 在基于共享数据模型的 SAAS 模式下,其必须具备处理多租户异构数据的能力,满足多租户数据无限扩展的要求。在此模式下,用户量非常高,业务数据为海量级^[11],并发压力也非常高。因此在如此海量数据和高并发压力的情况下,性能将会是做数据检索、统计计算的瓶颈。在上面提到的各种共享模型扩

展方法中,定制列、预分配字段不满足多租户无限扩展的需求,而名称值对虽然能满足多租户无限扩展的要求,但是在进行数据检索时,至少进行三表(数据表、扩展表、元数据表)的连接操作,在海量级的数据检索时,性能的低下很难满足用户的需求。而基于 XML 共享模型的多租户数据扩展方案,在数据检索时不必进行连接多表操作,其数据检索跟普通的 WEB 应用的数据检索差不多,只是必须对检索得到的数据进行相应的处理后呈现给用户;并且该方案可以满足多租户无限扩展的要求,以最大限度地利用表空间。

基于 XML 的共享模型的多租户数据扩展方案与定制列、预分配字段、名称值对从可扩展性、性能、灵活性等方面进行对比分析,如表 3 所示。

表 3 各方案比较表

比较项	定制列	预分配字段	名称值对	XML 存/取
可扩展性	低	中	高	高
性能	高	中	低	高
灵活性	低	中	高	高
实现复杂度	低	中	高	低
空间利用率	低	中	高	高

(2) 在软件设计模式中, MVC 模式^[12] (Model-View-Controller) 作为经典的设计模式,将模型、视图、控制器相互独立分开,模型层主要负责业务逻辑的处理和完成与数据库的交互;视图层主要负责与用户的交互;控制器的作用是从客户端接受请求,并选择执行相应的业务逻辑,把响应结果送回到客户端,它与模型层和视图层整合在一起完成用户的请求。

对于普通的 WEB 应用, MVC 模式完全可以作为搭建其系统的设计模式,不必考虑软件使用者之间的冲突,只需关心自身的功能逻辑。而 SAAS 系统,是由不同用户共享一个软件系统,就必须需要考虑软件不同使用者之间的冲突,系统就必须需要完成对不同用户异构数据的处理。

基于 XML 的共享模型的多租户数据扩展方案,只需在 MVC 模式中加入一个对 XML 数据的处理层,完成对 XML 数据的拆、封处理。其作用是在模型层完成对业务逻辑的处理后,将相应的异构数据转换成 XML

数据以便完成与数据库的交互;在模型层完成与数据库的交互后,将数据库中 XML 数据转换成相应的业务数据以便进行相应的业务逻辑处理。

引入 XML 数据处理层后的四层模型如图 2 所示。

(3) 现在大型关系数据库系统基本都具有 XML 文档的导入、导出功能。用基于 XML 共享模型的多租户数据扩展方案,可以将异构的数据(XML 数据)导出到本地磁盘,也可以将本地磁盘的 XML 文档导入到相应数据库表的 XML 数据列,增强了系统数据维护人员对异构数据的维护。

4 结束语

从 SAAS 模式的数据模型入手,从多个方面分析了现行多租户数据扩展技术的优缺点,在此基础上提出一种基于 XML 的多租户数据扩展方案,并探讨了其对整个 SAAS 应用体系架构和性能所产生的影响,从各方面与当前常用的多租户数据扩展方案进行对比分析,可得在 SAAS 系统中应用基于 XML 的共享数据模型的多租户数据扩展方案是性能优良的。

参考文献:

- [1] Dan Ma. The Business Model of Software-As-A-Service [C]//IEEE International Conf on Services Computing. [s. l.]: IEEE Computer Society, 2007.
- [2] Liao Hancheng. Design of SAAS-Based Software Architecture [C]//2009 International Conference on New Trends in Information and Service Science. [s. l.]: [s. n.], 2009: 277-281.
- [3] 王舜燕, 黄芬, 刘万春. 基于 SAAS 模式的软件设计方法探讨[J]. 计算机与数字工程, 2008, 12(10): 102-105.
- [4] 王文明, 曹宝香. 支持 SAAS 模式的消息中间件实时性的研究[J]. 计算机技术与发展, 2010, 20(7): 76-79.
- [5] 邓伟华. SAAS 应用的数据模型研究与设计[J]. 电脑编程技巧与维护, 2009, 24(8): 5-9.
- [6] Chong F, Carraro G. 多用户数据体系结构[M]. 北京: 人民邮电出版社, 2006.
- [7] 赵玉霞. 基于 SAAS 模式下的系统数据安全策略研究[J]. 软件导刊, 2010, 9(1): 143-145.
- [8] 吕中作, 徐悦, 戴钢. 基于 SAAS 模式公共服务平台多用户数据结构的研究[J]. 计算机系统应用, 2008(2): 7-11.

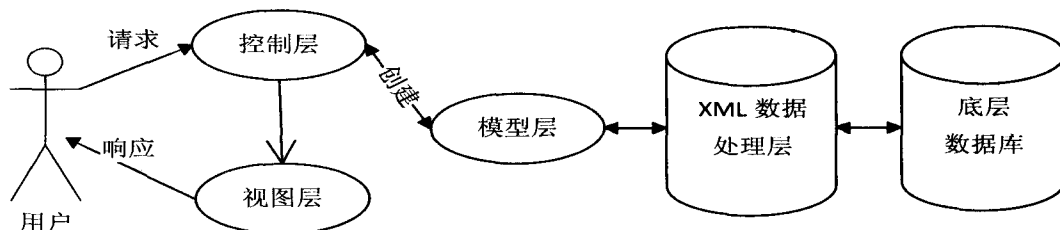


图 2 SAAS 服务模式下的四层模型

于实际错误数。另外,新模型在对未来错误数的预测方面体现出了一定的优势。通过对预计错误数的比较和拟合曲线的比较可以得出新模型的预测效果要好于 G-O 模型和原模型。各模型相应的评测标准数据如表 4 所示。

表 3 模型参数 II

参数名	a_0	Λ	p	q	α	λ
参数值	128.970	0.173	0.904	0.102	0.59	0.0973

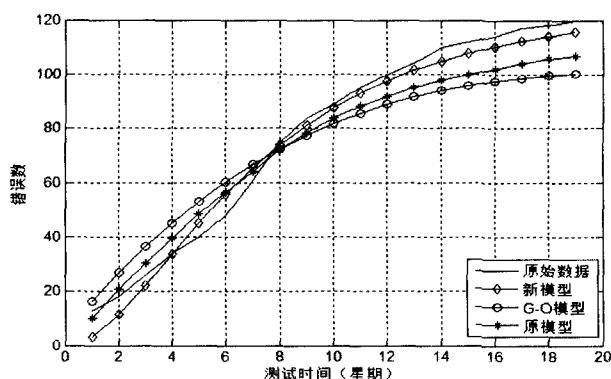
图 2 G-O 模型、原模型和改进后模型
对实际数据的拟合曲线 II

表 4 G-O 模型、原模型和改进后模型比较结果 II

模型	SSE	R-Square	AE
G-O 模型	2976.7	0.5208	0.1878
原模型	1406.8	0.6978	0.1115
新模型	404.8399	0.9826	0.1054

同样,新模型的拟合性能强于其它两个模型。

通过以上两组数据的实例仿真,可以很明显地看出新模型在各项评测指标上都优于其它两种模型,尤其在对未来错误数的预测上,有着一定的优势。综上,文中提出的新模型是成功的。

3 结束语

在深入研究已发表的 NHPP 模型的基础上,本节对软件可靠性模型的假设条件进行了修改和完善。在测试过程中,投入的资源(人力、物力、财力等)、测试环境以及测试人员的努力程度对测试的结果有一定的

影响。基于此,结合错误发现率函数,并考虑了软件排错过程中新错误的引入和错误的不完全排除等情况,提出了一种考虑测试效率的可靠性模型。最后,引用两组已发表的数据进行实例仿真,结果显示,新模型有着良好的拟合能力和预测性能。综上,文中提出的新模型是成功的。

参考文献:

- [1] 邹丰忠,刘海青,王 林. 软件可靠性综合模型[J]. 武汉大学学报(工学版),2003,36(1):86-88.
- [2] 蔡开元. 软件可靠性工程基础[M]. 北京:清华大学出版社,1995.
- [3] 徐仁佐. 软件可靠性工程[M]. 北京:清华大学出版社,2007.
- [4] 黄锡滋. 软件的可靠性与完全性[M]. 北京:科学出版社,1995.
- [5] Huang Chin-Yu, Kuo Sy-Yen, Lyu M R. An Assessment of Testing-Effort Dependent Software Reliability Growth Models [J]. IEEE Transaction on Reliability, 2007, 56(2):198-211.
- [6] Huang Chin-Yu, Kuo Sy-Yen, Chen Ing-Yi. Analysis of a Software Reliability Growth Model with Logistic Testing-Effort Function[C]//Proc of 8th International Symposium on Software Reliability Engineering. [s. l.]: [s. n.], 1997:378-388.
- [7] Yamada S. Software-Reliability Growth with a Weibull Test-Effort: A Model & Application [J]. IEEE Transactions on Reliability, 1993, 42(1):100-106.
- [8] Kuo Sy-Yen, Huang Chin-Yu, Lyu M R. Framework for Modeling Software Reliability, Using Various Testing-Efforts and Fault-Detection Rates [J]. IEEE Transaction on Reliability, 2001, 50(3):310-320.
- [9] Malaiya Y K, Karunanithi N, Verma P. Predictability measures for software reliability models [C]//Proc. 1990 IEEE Int. Computer Software and Applications Conference, Chicago, IL: [s. n.], 1990:7-12.
- [10] 顾林君. 一种错误发现率为时间函数的软件可靠性模型 [D]. 南京:南京邮电大学,2009.
- [11] 丁晓光. 基于测试覆盖的软件可靠性模型 [D]. 南京:南京邮电大学,2008.

(上接第 66 页)

- [9] Thomas K, Thao N, Linh L. A Software as a Service with Mtenancy Support for an Electronic Contract Management Application [C]//IEEE International Conf on Services Computing. [s. l.]: IEEE Computer Society, 2008:179-186.
- [10] 周 健,孙丽艳. 面向对象 XML 的存储模式的研究[J]. 计

算机技术与发展, 2009, 19(3):114-122.

- [11] 王安全. SAAS 模式下大数据量统计框架的研究和实现 [J]. 计算机技术与自动化, 2009, 28(2):128-130.
- [12] 潘海兰, 吴翠红, 葛晓敏. XML 及其在 MVC 模式中的应用 [J]. 计算机技术与发展, 2010, 20(2):202-205.

《计算机技术与发展》欢迎投稿, 欢迎订阅!