

# 基于开放架构的SaaS服务平台数据管理技术研究

国艳群<sup>1,2</sup>, 韩敏<sup>1</sup>, 孙林夫<sup>1</sup>

(1. 西南交通大学CAD工程中心 成都 610031; 2. 西南电子设备研究所 成都 610036)

**【摘要】**为解决SaaS平台面临的数据安全及分布式数据管理等问题,提出了基于开放架构的SaaS服务平台系统应用框架;在此基础上,对数据应用进行了抽象建模,以优化缓存管理策略的思路解决网络环境下的数据访问问题。通过将传统LRU算法与SIZE算法相结合,提出了LRU-RSIZE算法,对开放SaaS服务平台的数据访问进行管理,并进行仿真分析。开放SaaS服务架构与LRU-RSIZE算法相结合,可有效地解决用户数据的按需存储与管理问题,并对增强用户的应用体验具有重要的意义。

**关键词** 缓存管理; 数据管理; LRU算法; LRU-RSIZE算法; SaaS

中图分类号 TP39

文献标志码 A

doi:10.3969/j.issn.1001-0548.2015.02.023

## Research on the Data Manage Technology of SaaS Service Platform Based on Open Architecture

GUO Yan-qun<sup>1,2</sup>, HAN Min<sup>1</sup>, and SUN Lin-fu<sup>1</sup>

(1. CAD Engineering Center, Southwest Jiaotong University Chengdu 610031;

2. Southwest Institute of Electronic Equipment Chengdu 610036)

**Abstract** To solve the problems of data security and distributed data management that existed in SaaS platform, the paper puts forward a system application framework based on open SaaS service platform architecture. On the basis of this architecture, the paper gives abstract modeling on data application, which uses the idea of optimizing cache management strategy to solve data access problem in network environment. Combining with traditional LRU arithmetic and SIZE arithmetic, the paper gives a LRU-SIZE arithmetic which is effective on the management of data access of the open SaaS service platform. The combination of open SaaS service architecture and LRU-SIZE arithmetic can solve the problems of data access and data management of enterprise effectively. Meanwhile, it can improve the application experience of users.

**Key words** cache management; data management; LRU algorithm; LRU-RSIZE algorithm; SaaS

随着云计算技术<sup>[1-2]</sup>的发展,软件即服务(SaaS)<sup>[3-4]</sup>作为云计算的典型代表,取得了广泛的应用。然而,在SaaS的建设中,数据存储的安全及管理问题一定程度上影响了SaaS的应用,使得SaaS始终难以走过“最后一公里”。一个典型的表现就是企业私有数据的存储问题,即把涉及企业敏感或私有数据托管给第三方平台进行虚拟存储,一直是企业用户难以接受的事情。除此之外,与之相关的分布式的数据存储与管理、基于网络的数据访问等机制,都是SaaS平台建设中的巨大挑战。

为解决数据的存储、管理与访问等问题,国内外学者做了大量的研究工作。文献[5]提出了海量数据的分布式协同处理机制,文献[6-7]提出了多租户

模式下的数据存储、迁移及虚拟化访问等策略,文献[8]则从数据安全的角度,开展了数据隐私保护机制的研究工作。上述研究均是基于SaaS服务平台,从运行效率、安全等方面开展的对平台的优化,未从根本上解决SaaS用户的数据物理托管安全及相应的分布式数据访问等问题。

本文在传统SaaS平台的基础上,提出了基于开放架构的SaaS服务平台数据管理技术。在开放架构下,企业用户既作为平台的使用者又可作为平台的建设者,企业的数据可根据实际需要存储于企业本地或SaaS平台提供的虚拟存储空间。可有效地解决平台建设中用户数据的存储与管理问题,同时,通过对传统LRU缓存算法进行改进,提出了LRU-

收稿日期: 2014-09-15; 修回日期: 2015-01-12

基金项目: 国家863项目(2013AA040606); 四川省制造业产业链协同与信息化支撑技术重点实验室项目(2014002); 四川省科技支撑计划(2014GZ0142)

作者简介: 国艳群(1980-),男,博士生,高级工程师,主要从事产业服务平台、系统工程等方面的研究。

RSIZE算法,对提高开放架构下SaaS的平台的数据访问进行了优化,对增强系统网络环境下的应用体验,具有重要意义。

## 1 系统应用架构

根据实体及业务的不同,系统分为平台端与企业端两类,其中平台基于网络对外提供存储、计算

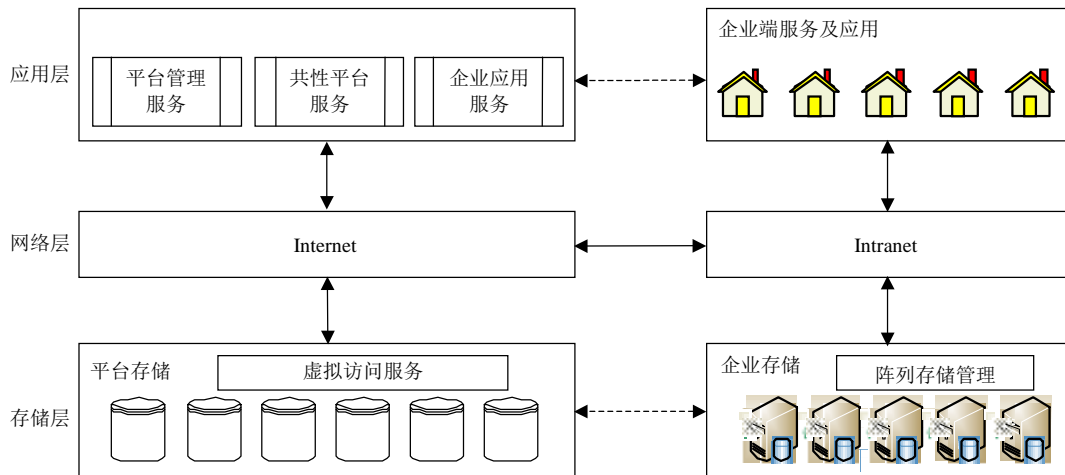


图1 系统应用架构设计

1) 存储层:包括SaaS平台的存储以及企业用户的存储阵列两部分,其中平台存储以虚拟的方式为企业租户提供存储空间,企业通过互连网络实现对虚拟存储的访问,企业存储阵列物理上位于企业内部,为企业私有。对于需存储的对象,从可公开性分类,可分为企业私有数据和企业共性应用数据;从动态性分类,可分为静态数据和动态数据;从数据的大小分类,则可按照数据所需的存储空间进行分级。对于数据的存储,由于在本文架构中,企业用户同时也作为平台的建设者,可以根据实际需要,将数据分类存储于平台或企业内部。

2) 网络层:包括互联网与企业网两类,企业内部的用户通过企业局域网络互联,企业则通过Internet与SaaS服务平台互联。

3) 应用层:平台端提供的服务主要包括平台管理服务、共性平台服务以及企业应用服务三类,其中平台管理服务实现平台自身的管理,共性平台服务为有平台提供的构件部署、发布、虚拟等服务,企业应用服务则为企业部署于平台的各类应用服务;企业端的服务及应用则为企业根据自身需要,部署于企业内部的服务及相应应用。

## 2 数据管理建模

对开放SaaS平台下数据访问机制进行抽象,可形成以下几种访问模式:

等服务,企业端则利用平台提供的服务开展企业相关业务,与传统SaaS不同的是,本文系统采用开放架构,即企业既作为系统的使用者使用平台的相关服务,又可作为建设者开发相应的服务,并选择将服务及其数据存储于平台端或企业内部。系统分为应用层、网络层与平台层,其应用架构如图1所示。

1) 企业数据及服务均部署于SaaS平台,企业用户通过网络访问;

2) 企业数据存储于SaaS平台,应用及服务部署于企业用户内部;

3) 企业数据存储与企业内部,企业服务部署于SaaS平台;

4) 企业数据及服务均部署于企业内部,平台仅提供基础管理服务及与多企业间的交互服务。

对于不同的企业应用,存在几种模式共存的情况,而在模式2、3均面临应用与数据分离的情况,本文主要针对这两种情况进行讨论。对这两种模式进一步抽象,则演化成网络化环境下对应用程序数据进行缓存优化的问题,综合考虑访问访问数据特性、网络带宽、缓存等因素,定义如下。

**定义 1** 数据,用三元组标识,  $D_i=(S_i, G_i, T_i)$ 。其中  $D_i$  表示第  $i$  个数据对象,  $S_i$  为  $D_i$  的唯一标识,  $G_i$  为数据  $D_i$  的大小,  $T_i$  为数据  $D_i$  的最近访问时间。

**定义 2** 带宽,用常量  $N$  表示,设企业的带宽保持稳定。

**定义 3** 缓存,用变量  $C_f$  表示。有两种计量策略,一种是缓存大小恒定情况下,计算不同缓存策略的性能指标,另一种是计算不同缓存策略缓存大小的需求指标。

目前,对于缓存的处理策略,典型的有LRU(以

最久未访问作为替换对象)、LFU(以访问频率最低作为替换对象)、SIZE(以访问代价最大作为替换对象)以及相应的组合及改进算法<sup>[9-10]</sup>,文献[11-12]基于分布式网络环境,开展了多级缓存、中心网络缓存等方面的替换策略研究,其本质仍然是对缓存策略与网络环境的综合应用,在具体实现方面,目前采用较多的为LRU算法。本文在SaaS应用环境下,从增强用户体验的角度,对LRU算法进行扩展,优先保留SIZE较大的对象可以增强用户的应用体验,本文在传统LRU及SIZE策略的,提出LRU-RSIZE交换算法(LRU reverse size),其评价因子计算为:

$$K_i = \delta_i \frac{B_i}{N} \quad (1)$$

式中,  $K_i$ 为“命中率-代价”因子,作为访问效率的综合评价;  $\delta_i$ 为命中率统计;  $B_i$ 为累计命中的访问数据大小与总访问数据量的比值,即在LRU替换的基础上,加入对象的访问代价作为计算权值。式(1)可有效地增强开放SAAS平台下的用户应用体验。

### 3 LRU-RSIZE算法描述与仿真分析

#### 3.1 算法描述

/\*LRU-RSIZE算法伪代码\*/

输入:

- 1) 随机生成的访问对象全集All-Set
- 2) 随机生成的对象访问顺序Seq
- 3) 栈形式的缓存列表Stack

输出:

- 1) 以 $K_i$ 为评价因子的访问命中率-代价统计
- 2) 算法对缓存空间的需求,即 $C_1$ 、 $C_2$ 的值

Begin

While( $i < N$ ) //N为循环访问次数

{

    随机生成访问对象 $A_i$ ;

    If( $A_i$ )in Stack //判断是否命中

        更新 $A_i$ 最近访问标志;

    else

    {

        if (Stack is not full) //如果栈尚未装满

            用 $A_i$ 更新栈顶;

        else

        {

            以 $T_i$ 作为第一因子,选出栈底元素 $D_i$ ;

            if( $D_i$ .Size $>\epsilon$ )// $\epsilon$ 为对象大小阈值

            //本文算法对 $\epsilon$ 取对象大小平均值

            以 $T_i$ 为因子在Stack中重新选择替换对象,直至选到满足 $D_i$ .Size $<\epsilon$ 的对象;

            //在实际应用中,该步骤加入随机因子,

            //避免出现栈中数据全部为大对象的情况;

            以 $A_i$ 替换栈顶;

        }

        以式(1)统计评价因子;

        统计Stack空间大小;

    }

    }

End

#### 3.2 算法特性分析

LRU算法为提高对象访问的命中率,采用优先置换最近未访问对象的策略,对最近未使用的对象进行替换,未考虑重新获取被替换对象的网络代价。LRU-RSIZE在考虑对象最近访问时间的时候,综合考虑被替换对象的大小(网络带宽一定的情况下,即为重新获取该对象的网络代价),并通过加入随机值的方式,避免出现大对象永远不会被替换的情况。LRU-RSIZE算法的主要适用环境为“具有一定本地缓存空间的网络应用访问”,对于开放SaaS平台而言,LRU-RSIZE的典型使用场景为企业应用与服务分离的情况,即服务部署于远端平台而数据存储于企业本地或者数据存储于远端平台而应用服务部署于企业本地。

#### 3.3 仿真结果分析

通过VC与Matlab对LRU-RSIZE算法与传统LRU算法进行仿真,在全集 $M$ 个对象、缓存 $N$ 个对象的情况下,随机生成 $X$ 次页面访问动作的条件下,得到对比曲线图像,其中 $M$ 取值200,  $N$ 取值100,  $X$ 取值5 000(示例,仿真分析表明在云计算环境下,同比增大 $M$ 、 $N$ 、 $X$ 的取值,并未影响相对趋势)。由仿真结果可得。

1) LRU-RSIZE的“命中率-代价”优于传统LRU的“命中率-代价”。从图2可知,仿真开始阶段由于缓存栈尚未装满,两种算法的“命中率-代价”均比较低,在第1 000次仿真后,“命中率-代价”逐步稳定,与LRU相比,LRU-RSIZE的“命中率-代价”稳定提升。

2) 因栈的大小按照固定对象的数目设定,从图3可以看出,在空间使用方面,LRU-RSIZE所用的空间比LRU也有所增加,两者的比例差距与图2中“命中率-代价”的比例差距相当。

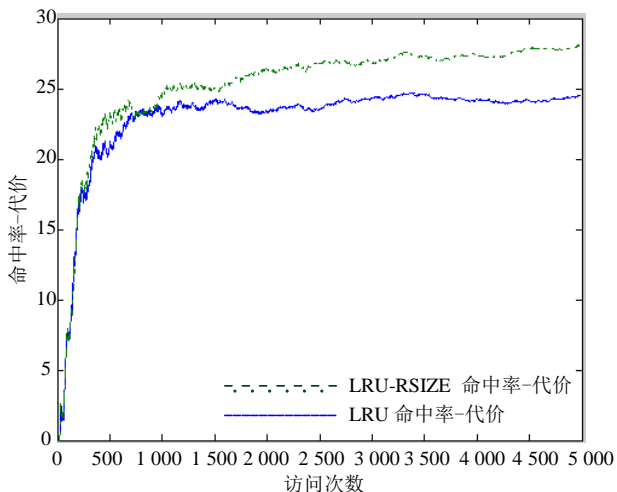


图2 “命中率-代价”仿真分析

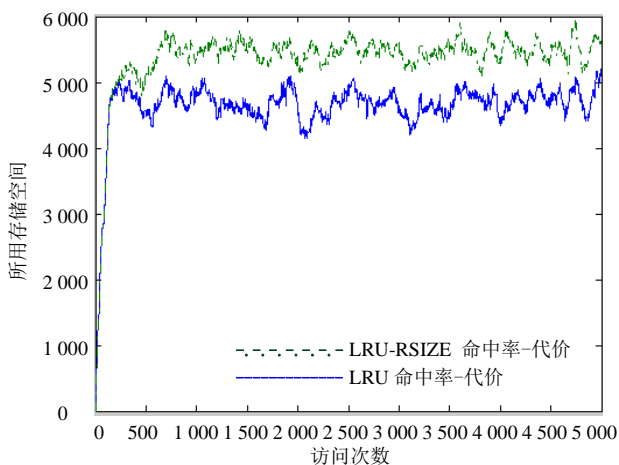


图3 缓存空间对比分析

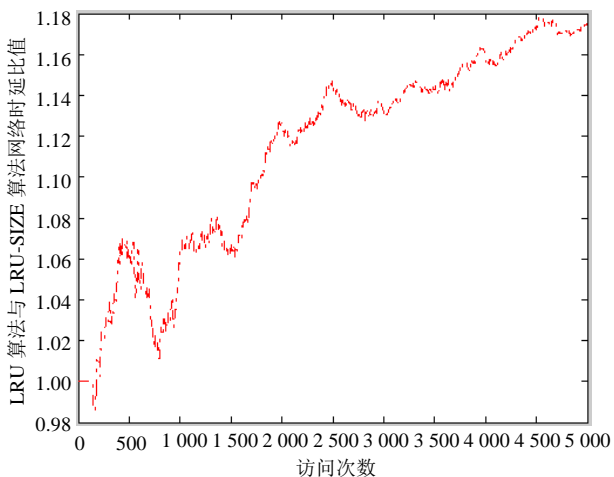


图4 网络时延分析

3) 网络时延与应用分析。从以上分析可以看出，采用LRU-RSIZE算法的同时，通过按照一定比例增大系统缓存空间可以获得比LRU更好的“命中率-代价”评估。进一步对比LRU算法与LRU-RSIZE算法的网络时延，如图4所示。与LRU-RSIZE算法相比，LRU算法的网络时延呈相对增大的趋势。在存

储代价不断降低的今天，存储容量已不再是网络应用的瓶颈。在用户网络带宽一定的情况下，开放的SaaS架构及相应的LRU-RSIZE缓存置换算法为解决SaaS的用户体验问题提供了一种新的解决思路，即“在考虑LRU的同时，优先替换较大的对象，减少用户对大对象的网络延时，增强用户的应用体验”。

### 4 小结

LRU-RSIZE算法以本地空间为代价，有效地增强了开放SaaS服务平台的用户应用体验，结合在SaaS数据管理体系架构，在平台的实际建设中，还可通过以下措施对数据管理进一步优化：

- 1) 建立专用的大文件服务器，实现对大块数据的专门存储；
- 2) 建立私有数据存储服务器，并通过数据的容灾备份机制，实现企业数据的安全存储；
- 3) 加强平台及企业用户的容灾备份及安全建设，进一步增强数据的可靠性与安全性。

另外，建立本地最小运行服务及数据存储子集，开展数据“断网-联网”情况下的数据同步研究，也是开放SaaS平台建设需进一步开展的重要工作。

### 参 考 文 献

[1] Wikipedia Cloud computing[EB/OL]. [2012-09-12]. [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing).

[2] 李伯虎, 张霖, 任磊, 等. 云计算典型特征、关键技术与应用[J]. 计算机集成制造系统, 2012, 18(7): 1345-1356.  
LI Bo-hu, ZHANG Lin, REN Lei, et al. Typical characteristics, technologies and applications of manufacturing[J]. Computer Integrated Manufacturing Systems, 2012, 18(7): 1345-1356.

[3] KWOK J, NGUYEN T, LAM L. A software as a service with multi-tenancy support for an electronic contract management application[C]//2008 IEEE International Conference on Service Computing. Honolulu, HI, USA: IEEE, 2008.

[4] 曹帅, 王淑营, 孙林夫. 面向产业链协同SaaS平台的业务流程定制技术[J]. 计算机应用, 2013, 33(5): 1450-1455.  
CAO Shuai, WANG Shu-ying, SUN Lin-fu. Workflow customization technology for collaborative saas platform of industrial chains[J]. Journal of Computer Applications, 2013, 33(5): 1450-1455.

[5] 戴炳荣, 宋俊典, 钱俊玲. 云计算环境下海量分布式数据处理协同机制的研究[J]. 计算机应用与软件, 2013, 30(1): 107-110.  
DAI Bing-rong, SONG Jun-dian, QIAN Jun-ling. Research on cooperation mechanism for mass distributed data processing in cloud computing environment[J]. Computer Applications and Software, 2013, 30(1): 107-110.