

# 基于 Eucalyptus 的多租户水利系统应用研究<sup>①</sup>

王 锋, 张 璟, 张 彤, 马维纲

(西安理工大学 计算机科学与工程学院, 西安 710048)

通讯作者: 王 锋, E-mail: [wf@xaut.edu.cn](mailto:wf@xaut.edu.cn)

**摘 要:** 多租户作为云计算的核心计算, 它解决了在相同的系统资源或软件应用中实现多用户共同访问和使用, 提高了系统软硬件资源的利用率, 如何保障租户服务质量的前提下提高服务器资源利用率是一个挑战性问题. 本文在 Eucalyptus 平台基础上, 利用多租户技术, 实现了陕西农村饮水规划调查数据库系统. 数据的存储方式采用共享数据库共享模式, 提出了多租户改进遗传放置算法, 将租户合理的放置到虚拟机上, 在最少硬件资源使用的情况下保证了租户服务质量. 在此基础上进行了实验分析, 表明了上述算法的可行性和有效性.

**关键词:** 多租户; 共享模式; 遗传算法; 资源利用率

引用格式: 王锋, 张璟, 张彤, 马维纲. 基于 Eucalyptus 的多租户水利系统应用研究. 计算机系统应用, 2018, 27(9): 107-111. <http://www.c-s-a.org.cn/1003-3254/6553.html>

## Application Study on Multi-Tenant Water Resources System Based on Eucalyptus

WANG Feng, ZHANG Jing, ZHANG Tong, MA Wei-Gang

(Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China)

**Abstract:** Multi-tenant is the core of cloud computing, it solves the problem of sharing system resources and applications among multiple users, and improves the utilization of software and hardware of a system. How to improve server utilization without degrading the quality of tenant service is a challenging problem. This study suggests a rural drinking water survey database system based on Eucalyptus platform and multi-tenant technology. Data storage uses the sharing mode of shared database, and this paper discusses a Multi-tenant Improved Genetic Placement Algorithm (MIGPA) which places tenant in a proper location in a virtual environment with minimum consumption of hardware resources and guaranteed service quality. Experiment results show that the algorithm is feasible and effective.

**Key words:** multi-tenant; shared schema; genetic algorithm; resources utilization

## 1 引言

陕西省在 101 个县(区)已经建立了县级农村饮水安全水质监测中心, 每个水质监测中心配备了能够满足检测需要的设备、仪器和取样车, 实现了县级农村饮水工程水质监测网络全覆盖. 这些检测信息资源若

不能有效收集处理, 就会影响主管部门的决策. 为此, 建立全省范围的水质检测收集系统很有必要, 该系统能为相关部门对农村饮水安全问题做出快速科学的决策与工程实施提供有力支持. 要建成该系统, 需要解决以下问题: 1) 该系统同时传输大量数据有共同的特性, 若开发类似的系统需要耗费大量的人力和物力资

① 基金项目: 国家自然科学基金面上项目(51477144); 陕西省工业科技攻关项目(2016GY-140); 陕西省教育厅重点实验室科学研究计划项目(15JS078); 西安市科技计划项目(2017080CG, RC043(XALG027))

Foundation item: The General Program of National Natural Science Foundation of China (51477144); Programs for Industrial Science and Technology Development of Shaanxi Province (2016GY-140); Scientific Research Project of Key Laboratory of the Education Department of Shaanxi Province (15JS078); Science and Technology Planning Project of Xi'an Municipality (2017080CG, RC043(XALG027))

收稿时间: 2018-01-15; 修改时间: 2018-02-09; 采用时间: 2018-03-13; csa 在线出版时间: 2018-08-16

源,如何合理共享软硬件资源需要解决的问题.2)该系统有上千个用户,每个用户同时提交大量的数据集,同时提交的数据量达到TB级别,如何实现高效数据访问与资源分配问题.

近几年国内外云计算的发展可以用“风起云涌”来形容:各大传统IT运营商纷纷在其核心业务上推出云计算服务;新兴云计算服务提供商不断推出云平台基础架构和管理系统,例如OpenStack、CloudStack及Eucalyptus等.

Eucalyptus源于美国政府支持的加州大学圣巴巴拉分校研究项目,已转为Eucalyptus System公司商业化运作,被HP公司收购,但是其核心代码仍是开源,为服务提供商及企业用户创建私有云或混合云提供了基础设施服务解决方案.Eucalyptus平台的API(应用程序接口)可以全面兼容著名云计算平台Amazon的API,已拥有虚拟化环境的用户可使用Eucalyptus云平台增强自己的虚拟化环境.Eucalyptus平台是基于基础设施服务(IaaS)的,本文的云平台网络拓扑结构采用按树的EDGE模式.Eucalyptus提供IaaS级的私有云服务.它由云控制器(Cloud Controller, CLC)、集群控制器(Cluster Controller, CC)、节点控制器(Node Controller, NC)、存储控制器(Storage Controller, SC)以及存储服务(Walrus)组成.云控制器为用户提提供访问整个云平台的唯一接口,接受用户请求,并将请求转发给具体的服务组件;集群控制器负责管理集群内的所有节点以及所有节点的资源使用情况和网络通信情况;节点控制器部署在具体的计算节点上,负责监控和管理本节点上的虚拟机的生命周期和资源使用情况;存储控制器为每一个虚拟机提供虚拟磁盘,安装存储控制器的物理机被虚拟化成一个虚拟磁盘池,用户可为每个虚拟机挂载指定配额大小的虚拟磁盘,并进行远程网络IO访问;Walrus以桶、对象的方式存储大容量静态数据,为用户提供上传、下载、罗列等基本存储服务.网络拓扑结构如图1所示.

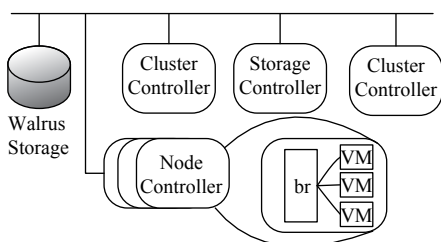


图1 EDGE模式网络拓扑结构

## 2 多租户技术

多租户的概念最早起源于软件领域,指一个软件实例服务于多个用户的架构.多租户作为云计算的核心计算,它解决了在相同的系统资源或软件应用中实现多用户共同访问和使用.同时应该保证系统安全性,并确保多用户之间数据的相互隔离.在数据层面,需要切割数据库实现多租户的数据管理<sup>[1-3]</sup>.

21世纪的今天,云计算已经成为IT领域一个新浪潮.各大云计算厂商纷纷推出了自己的云战略,SalesForce的force.com是面向特定领域(CRM),其领域客户需求共性明显,因此采用共享数据表和Shared Everthing架构非常合适并取得巨大成功.Windows Azure以云技术为核心提供了软件+服务的计算方法.SQL Azure采用共享数据库进程模式.Amazon推出弹性计算云(EC2)为企业提提供计算服务,和简单存储服务(S3)为企业提提供存储服务.Google搜索引擎建立分布在全球200多个国家和地区,超过100万台服务器的云计算环境下.在Google的众多应用中,多租户作为云计算的核心技术得到了更为广泛的应用,比如我们经常使用的Google Docs, Google App Engine等应用,都是多租户技术在实际应用中的应用体现.

国内主流的SaaS提供商有Xtools、阿里软件、800app、金算盘等.

随着云计算的兴起和推广应用,多租户技术也越来越成熟,借助多租户技术来架构农村饮水规划调查系统,是农村饮水规划调查快速提升信息化水平的捷径.本文正是基于这样的背景来研究如何在农村饮水规划化系统中引入多租户模式,提升农村饮用水检测信息化水平,降低农村信息化投入成本和维护成本,提高农村饮用水质量.“陕西省农村饮水规划调查数据库系统”利用Eucalyptus云计算平台,建立一个多租户的共享数据库共享schema水资源检测系统能解决目前存在的难题.

## 3 多租户关键技术研究

“陕西省农村饮水规划调查数据库系统”在Windows环境下实现了农村饮用水安全的工程调查信息在网上的传输、转换、处理、汇总、查询、分析,为省政府和相关部门在对农村饮用水安全问题做出快速科学的决策与工程实施提供了强有力的支持.系统主要包括数据转换、数据审核、数据对比等12个功能模块.多租户水利系统部署拓扑如图2所示.农村饮

水规划调查数据库系统的 Web 前端和数据库。

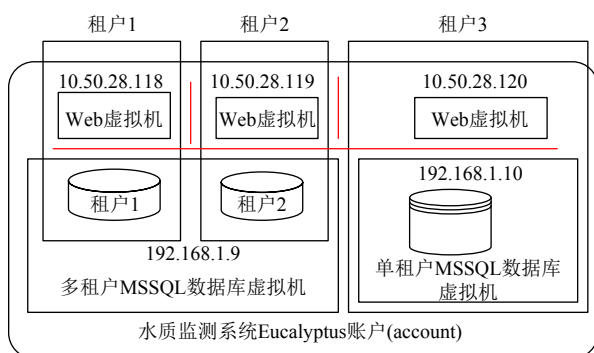


图2 数据库系统部署拓扑图

### 3.1 模块划分

单独部署, 每个租户都有自己 Web 虚拟机, 分配有公有 IP, 每个租户都有自己安全组和防火墙强规则, 租户的 Web 服务器放在自己的安全组里, 不同租户的 Web 虚拟机网络二层不通。

普通租户可以共享数据库服务器, 高级租户独享数据库服务器。所有数据库在同一安全组, 设定相应的防火墙规则。整个系统部署在一个 Eucalyptus Account 内。Web 虚拟服务器使用 instance-store 的镜像, 在启动虚拟机通过 user-data 自动把要使用的数据库信息传

给 Web server。MSSQL 虚拟机可以使用 EBS-base 的镜像 (20 G), 同时分配 EBS volume 用作数据盘。

### 3.2 多租户数据库设计

数据库设计是多租户实现上的最重要技术。租户与租户之间的数据需要相互隔离保证数据的安全性<sup>[4]</sup>。由于各租户数据之间有共同的数据特性, 所以多租户数据的存储方式采用共享数据库共享模式 (又称共享模式)。

在数据库表设计方面, 本系统采用多个租户使用相同的数据库和相同的表。为了将不同租户的数据进行隔离, 所有表包含一个 Water\_id 字段。在查询或存储表数据时, 需要应用 Water\_id 字段对数据进行强制隔离。数据库表举例: 以下两张表是本数据库用到的主要表。表 1 和表 2 有共同的租户字段 Water\_ID, 表 1 是村庄信息表, 查询该表信息需输入租户 Water\_ID; 表 2 为饮水点情况表, 所有租户的检测信息都存放在该表中, 租户之间的通过租户 Water\_ID 进行隔离。当租户需要对自己的业务数据操作时, 每条操作 SQL 条件语句需要增加 Water\_ID='?' 区分不同租户, 防止其它租户读取本租户信息。通过租户 Water\_ID 可以将表 1 和表 2 联系起来。

表 1 村庄信息表

Village_NO	Water_ID	city	county	town	country	Practice_Capacity	Source_Type	Manage_Type
040101	001	铜川市	耀州区	城关镇	东街村	27.6	地下水	镇管
040102	007	西安市	蓝田县	三里镇	罗李村	15.6	地表水	县管
040103	0010	榆林市	米脂县	印斗镇	沙坪	12.3	地下水	县管

表 2 饮水点情况表

ID	Village_NO	Water_ID	Test_time	Test_Result
1	040101	001	2015.8.10	优
2	040101	001	2015.9.1	优
3	040102	007	2015.4.4	优
4	040103	0010	2015.10.8	良

例如: 查询租户号 001 的隔离语句如下:

```
Select s1.city, s1.county, s1.town, s1.Source_Type,
s1.Manage_type s2.Test_time, s2.Test_Result from s1, s2
where s1.Water_ID=s2. Water_ID AND s1.Water_ID=
“001”
```

### 3.3 多租户改进遗传放置算法

当一个新租户加入时, 根据租户的用户数和服务

质量要求需要计算为租户分配的资源数量, 资源数量用于保障租户的服务质量, 租户放置算法将新加入的租户放置到资源节点上, 使之既满足租户的资源需求, 同时使资源供应商的资源利用率最大化。对于应用放置方面, 文献[5,6]对于多租户资源动态分配策略进行研究, 文献[7]对多租户安全资源分配算法和安全服务调度框架进行研究。文献[8]对多租户系统进行动态的扩展, 运用装箱策略管理多租户云应用程序中的存储资源。文献[9]应用遗传算法对 Android 设备驱动程序的未知安全漏洞进行检测。文献[10]以完工时间、机器能耗和工人操作机器的舒适度作为柔性作业车间调度问题的多目标函数, 利用改进遗传算法对其进行优化研究。本文在以往的研究基础上提出了多租户改进遗传放置算法 (Multi-tenant Improved Genetic Placement



Algorithm, MIGPA). 既满足租户的资源需求, 同时使资源利用率最大化。

通过 MIGPA 将租户放置到合适的虚拟机中, 生成租户放置到服务器的序列. 本文采用装箱问题中基于群体的编码方式, 对租户从 1, 2 到  $n$  进行编号, 采用字母 A-Z 来表示服务器的编号. 若有租户 1, 2, 3, 4, 5, 6, 7; 服务器 A, B, C, D, E, F. 若租户 2, 4, 6 放置到服务器 B 上, 则有  $B=\{2, 4, 6\}$ ; 租户 1, 3 放置到服务器 A 上, 则有  $A=\{1, 3\}$ ; 租户 5, 7 放置到服务器 C 上, 则有  $C=\{5, 7\}$ ; 则染色体可以记为  $BAC(B=\{2, 4, 6\}, A=\{1, 3\}, C=\{5, 7\})$ . 执行多次租户放置算法形成多个染色体, 用遗传算法实现租户占用的服务器资源最少及利用率最高. 多租户数据库系统主要由外部负载、数据库节点和租户数据库副本等几方面构成, 其参数主要包括用于描述节点、租户以及负载的基本参数使用的一些符号如表 3 所示.

表 3 基本参数

符号	含义
$T[i]$	表示需部署的租户, $T[i]$ 表示第 $i(i \in [0, n-1])$ 个租户, $n$ 表示租户个数
$S[i]$	表示第 $i$ 个服务器剩余的资源
$R[i]$	各租户放置所需资源, $R[i](i \in [0, m-1])$ 表示第 $i$ 个租户放置所需资源数
$F[i]$	第 $i$ 个服务器中所有租户的占用资源的和
K	表示了根据服务器资源使用情况对服务器的重视程度
C	表示服务器资源数量的限制, 资源包括 CPU, 内存硬盘等
Q	租户序列队列

本系统为实现服务器资源利用率最大化, 设计了多租户改进遗传放置算法 (MIGPA) 算法. 其中部分参数含义见表 3, 其余参数表示如下: Pc: 交叉发生的概率; Pm: 变异发生的概率; M: 种群规模; Gen: 终止进化的代数; Tf: 进化产生的任何一个个体的适应度函数超过 Tf, 则可以终止进化; 适应度函数  $f(i) = \sum_{i=1}^N (F/C)^k / N$ ,  $N$  是服务器数量; OldPop 表示旧种群; NewPop 表示新种群.

算法. 多租户改进遗传放置算法

Procedure BFD

初始化 Pc、Pm、M、Gen、M

0.8 =>Pc

0.09=>Pm

30=>M

20=>Gen

0=> $i=j$

初始化服务器剩余资源  $S[n]=\{S1, S2, \dots, Si, \dots, Sn\}$

X 个租户加入队列 Q

while (队列 Q 不为空)

租户出队  $T[i]=$  队列 Q 的队头出队

while ( $j < N$ ) 循环查找最适合的分配服务器  $S[j]$

if ( $R[j] < \min(S1, S2, \dots, Sj, \dots, Sn)$ )

服务器剩余资源  $S[j]-R[i] \Rightarrow S[j]$

end if

end while

end while

while(任何染色体得分超过 Tf Or  $i > Gen$ )

计算种群 OldPop 中个体的适应度  $f(i)$

初始化种群 NewPop

while( $M \leq 22$ )

根据个体适应度  $f(i)$  以比例选择算法从种群 OldPop 中选出 2 个个体

if random (0, 1) < Pc then

按交叉概率 Pc 对 2 个个体执行交叉操作

end if

if random (0, 1) < Pm then

按变异概率 Pm 对 2 个个体执行变异操作

end if

将 2 个新个体加入种群 NewPop 中

end while

newPop=>OldPop

end while

end Procedure

## 4 性能测试

本文研究租户放置算法目的是在保障租户服务质量的前提下提高服务器资源利用率. 采用虚拟化的方法, 在租户资源分配初期为租户分配合适的资源量, 并通过租户放置算法提高服务器使用率, 减少租户部署时期的物理服务器的使用数量, 降低服务供应商的成本. 本系统通过以下几个实验进行测试, 搭建的测试环境如下.

硬件: CC 和 SC 部署在一台 IBM X3850 服务器; CLC 和 Walrus 部署在一台 IBM X3850 服务器; NC 节点使用 30 台 IBM 3550 服务器; 所有服务器安装 centos6.5 操作系统; 系统运行于 NC 节点的虚拟机; 本次测试采用 LoadRunner.

实验一. 优化过程实验.

测试放置算法的优化过程. 用 LoadRunner 模拟注册 50 个租户, 每个租户用户数在 10-50 之间, 系统响应时间为 0.1 到 2 s, 农村饮水规划调查数据库系统产生 90 个虚拟机分配给 90 个租户. 放置程序需要将 90 个虚拟机放置到 30 台服务器中. 90 个租户的用户并发访问水利管理系统. 采用 Matlab 实现遗传算法, 设置交叉概率为 0.8, 变异概率为 0.09, 种群的个数为

30, 遗传代数数为 20. 图 3 显示了将 90 个租户放置到 30 台服务器的优化过程. 进化次数越多, 物理服务器占用的个数就越少. 在 20 代之前, 物理服务器数量变化较大, 之后就逐渐稳定. 租户放置算法很快得到了最优解.

#### 实验二. 占用服务器资源实验.

在满足租户服务质量要求的基础上, 采用 Next Fit, Best Fit 两种启发式算法同本文提出 MIGPA 进行比较, 使得需要的服务器数量最少.

Next Fit (NF) 算法: 将租户和服务器编序号, 按照租户序号依次放置到各个服务器中, 若低序号服务器资源不够, 则依次比较是否能放入下一个高序号服务器中, 直到放置完成.

Best Fit (BF) 算法: 将租户和服务器编序号, 依次将租户放置到剩余资源最少且能放置该租户的物理服务器中.

在 25 台服务器上放置租户. 四种租户放置算法的比较结果如表 4 所示. 本文提出的 MIGPA 同 NF、BF 和贪心算法相比, 使用的物理服务器数量最少, 当租户数量达到 90 个时, NF 算法的服务器已经不能继续放置租户了. 但是随着租户数量增多, MIGPA 优势越来越明显.

表 5 占用服务器资源实验

租户数	NF	BF	贪心算法	MIGPA
20	8	6	7	5
30	15	12	14	9
60	26	22	20	16
90	×	26	24	20

#### 实验三. 服务器资源利用率实验.

进一步测试 MIGPA 的 CPU 资源利用率. 选取 3 台服务器 Server1、Server2 和 Server3, 测试时间为 30 小时, 实验结果表明 MIGPA 可以为租户搜索到合适的服务器, CPU 资源利用率高, 满足租户的服务质量要求. 结果如图 3 所示.

## 5 结论

本文应用 Eucalyptus 的 EDGE 网络模式建立了云平台, 分析了多租户共享数据库模式系统的设计与实现, 提出了多租户改进遗传放置算法, 并和 Next Fit、Best Fit 和贪心算法进行比较, 本文提出的放置算法在随着租户数量增加, 使用的服务器数量会相对减少, 提高了资源利用率, 为租户提供了满意的服务质量. 通过

实验证明了本文提出的 MIGPA 可以较好地保证租户服务质量并提高服务器资源利用率.

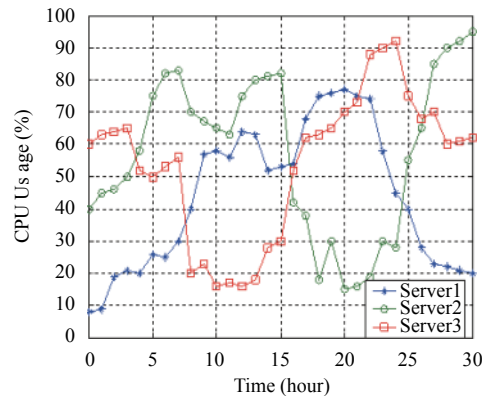


图 3 MIGPA 服务器 CPU 资源利用率实验

#### 参考文献

- 李响, 李彤, 谢仲文, 等. 一种面向 SaaS 多租户的多层模型. 计算机科学, 2017, 44(11): 56–63. [doi: 10.11896/j.issn.1002-137X.2017.11.009]
- Kuppusamy S, Kaniappan V, Thirupathi D. Design and development of multi-tenant Web framework. Procedia Computer Science, 2015, (48): 180–191. [doi: 10.1016/j.procs.2015.04.168]
- 李晓娜, 李庆忠, 孔兰菊, 等. 基于共享模式的 SaaS 多租户数据划分机制研究. 通信学报, 2012, 33(S1): 110–120.
- Zahid F, Gran EG, Bogdański B, et al. Efficient network isolation and load balancing in multi-tenant HPC clusters. Future Generation Computer Systems, 2017, (72): 145–162. [doi: 10.1016/j.future.2016.04.003]
- 谭瑛, 陈宁江, 黄汝维, 等. 一种多因素博弈的多租户资源动态分配策略. 小型微型计算机系统, 2016, 37(10): 2177–2182. [doi: 10.3969/j.issn.1000-1220.2016.10.007]
- 孟凡超, 周学权, 曹祖凤, 等. 基于成本优化的多租户 SaaS 应用优化放置算法. 计算机集成制造系统, 2014, 20(6): 1508–1518.
- 苑中梁, 陈兴蜀, 王毅桐. IaaS 环境下多租户安全资源分配算法和安全服务调度框架. 计算机应用, 2017, 37(2): 383–387, 396.
- Maenhaut PJ, Moens H, Volckaert B, et al. A dynamic tenant-defined storage system for efficient resource management in cloud applications. Journal of Network and Computer Applications, 2017, (93): 182–196. [doi: 10.1016/j.jnca.2017.05.014]
- 何远, 张玉清, 张光华. 基于黑盒遗传算法的 Android 驱动漏洞挖掘. 计算机学报, 2017, 40(5): 1031–1043.
- 王雷, 蔡劲草, 石鑫. 基于改进遗传算法的多目标柔性作业车间节能调度问题. 南京理工大学学报 (自然科学版), 2017, 41(4): 494–502.