

基于帕累托最优的隐私保护服务定价模型

史玉良 邵雅丽 周中民 张宏磊 陈 玉 崔立真

(山东大学计算机科学与技术学院 济南 250101)

摘 要 在云计算环境下,租户向云服务提供商租赁云产品和服务,那么云服务商如何规定一个合理的服务定价使得多租户与提供商都满意成为一个亟待解决的问题.针对该问题,该文提出一种基于帕累托最优思想的服务定价模型,综合考虑多租户与提供商的利益,使用多目标粒子群优化算法得到全局最优的资源分配与服务定价结果.由于迭代算法的效率与服务定价实时性的需求存在冲突,所以该文提出两阶段定价策略:独立定价与集中定价.独立定价阶段,参考历史同需求或相似需求的定价,结合当前资源使用占比给出实时报价;集中定价以一定时间为周期,根据已知的该周期内多租户提出的不同需求,提前使用粒子群算法求出最优服务定价与资源分配策略.实验表明,通过该文提出的定价模型,可以得到一个使多租户和云服务提供商都满意的定价,并且在定价过程中,采用隐私保护技术,有效地保护了租户的数据安全.

关键词 隐私保护;帕累托;粒子群优化算法;独立定价;集中定价;云计算;网络空间安全

中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2016.01267

Pricing Model of Privacy Preserving Service Based on Pareto Optimization

SHI Yu-Liang SHAO Ya-Li ZHOU Zhong-Min ZHANG Hong-Lei CHEN Yu CUI Li-Zhen

(Department of Computer Science and Technology, Shandong University, Jinan 250101)

Abstract In the circumstance of cloud computing, tenants lease cloud products and services from cloud service provider. How to give a reasonable price satisfying both provider and tenants is a challenge for provider. This paper proposes a pricing model based on Pareto Optimization thought with application of Multi-Objective Particle Swarm Optimization Algorithm. It gives a global optimal resource allocation and price result, considering comprehensively profits of both provider and tenants. In order to avoid conflict between iterative algorithms' low efficiency and the service pricing's real-time demand, the thesis separates the pricing process into two stages, referring to Solitary Pricing and Centralized Pricing. In the Solitary Pricing stage, we give a real-time price referring to the history data with the same demand or similar demand and current resource usage ratio. In the Centralized Pricing stage, we give an optimal price and resource allocation strategy with Pareto Optimization thought based on different known demands proposed by multi-tenants in a certain period. The experimental results show that the proposed pricing model can get a price satisfying both provider and tenants, and in the pricing process, using the privacy protection technology, effectively protects the data security of tenants.

Keywords privacy preserving; pareto; particle swarm optimization; solitary pricing; centralized pricing; cloud computing; cyberspace security

收稿日期:2015-01-31;在线出版日期:2015-12-28. 本课题得到国家自然科学基金(61272241,61572295)、科技部创新方法工作专项(2015IM010200)、山东省泰山产业领军人才工程专项经费、山东省自然科学基金(ZR2014FM031,ZR2013FQ014)、山东省科技重大专项(2015ZDXX0201B03,2015ZDXX0201A04,2015ZDJQ01002)、山东省科技攻关计划(2015GGX101015)和山大基础科研经费(2015JC031)资助. 史玉良,男,1978年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为云计算、数据库、隐私保护. E-mail: shiyuliang@sdu.edu.cn. 邵雅丽,女,1987年生,硕士,主要研究方向为云计算、隐私保护. 周中民,男,1989年生,硕士研究生,主要研究方向为云计算、隐私保护. 张宏磊,男,1987年生,硕士研究生,主要研究方向为云计算、隐私保护. 陈 玉,男,1990年生,硕士研究生,主要研究方向为云计算、隐私保护. 崔立真,男,1976年生,博士,教授,中国计算机学会(CCF)会员,主要研究领域为数据科学与工程、服务计算、智能协同计算.

1 引言

随着云技术的飞速发展,云产品或云服务被企业及个人广泛应用.在云环境下,租户的数据需要上传到云服务提供商处,如果对上传的数据不加处理就直接存储,将不可避免地存在隐私泄漏的可能,比如数据库管理员窥探库中数据时造成无意识泄漏或有意将信息卖给竞争对手公司等.因此,提供商不仅会提供云产品或云应用服务,还会提供相应的隐私保护服务.

本文前期工作^[1-2]中提出数据分块作为隐私保护服务的一种策略,将租户的身份信息与隐私敏感属性分割到不同数据分块中,然后混淆数据分块间的对应关系以保护租户的隐私.该方法较传统的隐私加密与数据泛化方法有着良好的性能并且可以重构原始信息.文献^[3-4]给出了不同的数据分块策略生成算法,算法均根据租户个性化提出的隐私约束计算策略.隐私约束越复杂,数据分块个数越多,其上的应用操作需要的连接次数越多,性能相对就越低.想要在隐私安全的同时提高应用性能,就需要获得更多的云计算资源(CPU、内存、缓存等).资源分配的多,服务定价就会越高,但是超过租户承受能力的价格又会造成用户的流失,服务提供商如何规定一个合理的定价使得租户与提供商都满意成为一个亟待解决的问题.

在本文前期工作^[5]中,提出采用基于纳什均衡的思想对隐私保护服务进行定价,但在纳什均衡点处,参与者虽然对自身状态都满意,但此状态不一定是全局最优的,因此本文采用帕累托最优的思想建立多目标粒子群模型寻求多目标优化的平衡点.帕累托最优思想适合于解决多目标矛盾性问题,而粒子群算法(PSO)较一般的优化算法来说,需要调整的参数少且能较快收敛于最优解.然而,当种群规模较大,迭代次数较多的情况下,粒子群算法耗时依然较大,比如,由于数据迁移、资源的重新分配等因素,50个租户迭代50次所花费的时间数量级以小时计.考虑到现实情况,租户不可能等待数小时再拿到一份定价合同,因此本文将定价分为两个阶段:独立定价与集中定价.独立定价参考同需求(或需求相近)的上一周期历史定价数据,根据资源占比情况快速给出价格;集中定价以一个月为周期,将本月内加入的租户进行博弈,求出最佳资源分配与定价.由于

集中定价阶段已知租户的需求,可以在该周期内租户签订正式租用协议之前预先一天或几天进行计算,因此算法的效率对服务影响不大.

本文第2节是相关工作;第3节给出隐私保护服务定价模型的相关概念与问题描述;第4节介绍隐私保护服务定价模型的具体实现;第5节是相关实验和分析;最后对本文进行了简要的总结.

2 相关工作

大部分云服务提供商都是按照虚拟机配置以固定比例定价,比如 Amazon EC2^①和 Microsoft Windows Azure^②对计算资源的使用、存储和数据传输均采用此种策略.不可否认,该策略对于租户来说可以更直观的理解.文献^[6]也是基于“按使用付费(pay-as-you-use)”的思想,提出一种虚拟企业模型的概念,租户对云资源的需求稳定时,模型为其分配公有云资源,当租户在短时间内对云资源的需求大量增加时,模型为其分配私有云资源,以此尽可能的降低提供商的开销.然而此种策略只针对中小企业有效,普适性较低.文献^[7]综合考虑租户的折扣心理与时间需求因素,给出一个基准模型,为云服务动态地定价.该模型使得服务提供商可以从非理性租户和时间需求紧的租户处获得较大收益.文献^[8]利用动态赌博理论建立了一个面向资源提供商的定价模型,该模型既可以吸引大型专业IT资源提供商,又可以吸引个人计算机拥有者加入云银行.以上定价模型着重考虑提供商的收益,而忽略了租户的利益.

Amazon EC2后期又提出了一种竞价机制,它根据资源的供求关系来确定价格,这种竞价机制综合考虑提供商与租户双方的利益.文献^[9]提出一种自适应的多轮组合拍卖竞价方式,竞标者可以根据历史投标数据即时的调整自己的边际收益率,借以最大化期望值.文献^[10]对首轮-二轮竞标算法进行修改以满足动态分配多媒体资源的需求.文献^[11]提出一种基于组合双向拍卖的动态资源定价模型,给出云资源分配与定价算法,用户通过响应时间出价,资源提供商根据负载情况要价.竞价机制综合考虑了提供商与多租户的利益,然而忽略了用户的服

① Web portals of EC2. <http://aws.amazon.com/cn/ec2/>

② Web portals of Azure. <http://azure.microsoft.com/en-us/>

务质量需求. 文献[12]利用经济学中的金融选择理论给出一个资源定价模型, 该模型将云资源当作潜在资产计价, 动态计算资产在不同时间的价格, 既能满足用户的服务质量需求, 同时最大化租户的利益.

上述方法均没有将安全和隐私保护服务作为对资源定价的影响因素来考虑. 在云计算环境下, 由于不同租户对云安全保护需求和隐私关注程度不同, 因此对同一种云安全隐私保护技术的收费也不尽相同. 当前不少机构和公司根据不同云安全方案和客户需求制定了各自的云安全保护方案和定价标准^[13], 例如 Zero-Knowledge & Privada 公司利用匿名代理技术对客户身份信息进行匿名隐藏, 保证客户的隐私信息在上网过程中不会发生隐私泄露, 并根据保护级别的不同, 收费在每月 30 美元到 40 美元之间不等. 知名的隐私服务公司 TRUSTe 则根据客户的需求, 发现客户的系统或数据潜在的隐私安全风险, 并制定可行隐私保护方案, 隐私服务定价的标准取决于客户系统或数据的复杂性. 科力斯卫士 (QualysGuard)^① 是全世界覆盖范围最广, 部署用户最多的按需定制安全解决方案, 它在短短几小时内就可在全世界内任何地方、任何时候部署 SaaS 模式服务方案, 保护用户的隐私. 国内云服务商中, 阿里云在对外提供基本的云产品和云服务的同时, 也推出了一系列的云安全服务 (如云盾) 供租户按需租赁, 并对各种不同的安全服务按保护等级进行定价收费^②. 上述这些国内外研究和应用成果对本文的研究内容提供了很好的借鉴意义.

本文前期研究成果^[1-4]证明, 在 SaaS 模式中, 基于分块混淆的隐私保护技术可以同时保证应用性能和数据可用性, 然而, 隐私保护服务等级的高低, 隐私约束的复杂度直接影响到服务质量, 若想要满足租户对服务的性能需求, 必须分配更多的资源, 相应收取更高的费用. 本文的前期工作^[5]提出一种隐私保护策略定制框架, 综合考虑隐私保护需求、SaaS 应用的性能和租户与 SaaS 服务提供商的利益, 基于纳什均衡思想建立博弈模型, 最终通过多租户与 SaaS 服务提供商之间的博弈获得隐私保护策略, 并给出各个租户的资源定价.

纳什均衡^③又称非合作博弈均衡, 指参与者为了自身利益的最大化, 都不愿意单方面改变其策略的状态, 也就是说, 如果参与者单方面改变策略, 只会使自己收益更低. 可以证明, 在纳什均衡点处, 参

与者虽然对自身状态都满意, 然而此状态不一定是全局最优的, 可能参与者选择了一个较坏策略的均衡点, 但在帕累托最优思想下选择的一定是最优策略的均衡点, 从而达到全局最优, 即在帕累托最优下租户和服务供应商实现了双赢, 利益最大.

在定价上本文提出的基于帕累托最优的隐私保护服务定价模型将定价分为两个阶段: 独立定价与集中定价. 独立定价参考同需求 (或需求相近) 的上一周期历史定价数据, 根据资源占比情况给出价格; 集中定价根据多租户提出的资源申请量与隐私保护服务需求建立效用函数, 并使用多目标粒子群算法根据最优解的帕累托支配关系进行求解. 该模型通过降低资源定价或应用响应时间 (即提升服务质量) 提升租户满意度的同时最大化资源利用率, 以此增加提供商的收益.

3 隐私保护服务定价模型

3.1 SaaS 应用隐私保护基本模型

如图 1 所示.

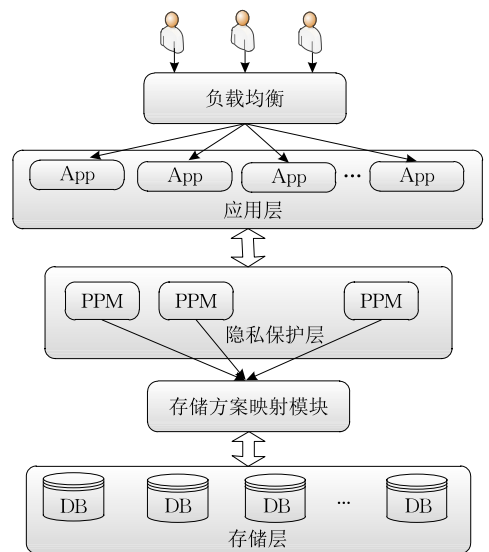


图 1 SaaS 应用隐私保护基本模型

在 SaaS 模式下, 云服务提供商提供多个应用实例 (App) 供租户进行定制和租赁, 并通过负载均衡模块来平衡服务压力. 为了防止租户隐私的内部泄露, 在应用层和存储层之间增加了隐私保护层, 该层负责对租户的隐私数据进行分块混淆, 并将混淆后

① <https://www.qualys.com/>

② <http://www.aliyun.com/>

③ http://en.wikipedia.org/wiki/Nash_equilibrium

的数据进行存储.此外,当租户需要访问隐私数据时,调用相应的重构算法对分块数据进行重构并返回给租户.随着租户数量和请求压力的增加,单一的隐私保护模块会导致数据分块和重构时间的增加,进而使服务总的响应时间增加,影响租户 SLA 约束的实现,因此需要在隐私保护层,部署多个隐私保护模块(PPM)来并行处理多租户数据的分块和重构操作.

通常情况下,SaaS 模式可以采用 3 种存储方案存储不同租户的数据,即独立数据库、共享数据库独立数据模式和共享数据库共享数据模式.其中独立数据库方案隔离程度最高,但共享程度最低,使用成本也最高,而共享数据库共享数据模式方案则具有隔离程度最低共享程度最高的特点,适合以最少的服务器支持最多租户的应用场景,使用成本相对较低.由于 SaaS 应用具有可个性化定制的特点,使不同租户对数据的隔离程度以及应用性能的要求也会不同,而不同的隐私保护需求则使不同租户的分块策略有所不同,最终需要使用不同的存储模式来存储数据.因此,当隐私保护模块对租户数据分块完成后,存储方案映射模块需要根据该租户的分块策略映射到合适的存储模式上,并将分块数据在存储层进行存储.

在上述 SaaS 应用隐私保护基本模型中,应用实例和隐私保护模块具有多实例和多共享的特性,数据存储模式则具有多样性的特点,且因租户隐私需求和应用性能要求的不同,对于不同的存储模式,其共享程度也会不同.共享程度越高,服务商获得相同的利润需要付出的成本就越低,对应租户的定价也就越低,反之,要保持一定的收益成本比,则只能提高单位租户的定价.下面将首先对隐私保护过程中的不同数据存储模式及其共享程度进行介绍,然后详细描述帕累托资源分配的思想,最后通过结合资源分配结果和资源共享程度以及帕累托最优思想,总体描述隐私保护服务定价过程.

3.2 SaaS 数据存储方案

传统的 SaaS 模型采用的是 multi-tenancy 架构,也是单实例多租户.在这种模型下,服务提供商部署一个应用实例即可满足多个客户的要求.在数据存储方面则采用共享数据库共享数据模式的存储方案,以实现通过最大程度的共享来支持最多的租户.从用户的角度考虑,则需要进行特殊的改造和过

滤,使每一个租户感受不到所使用的应用实例和存储模式也在同一时间为其他客户所共享,提升用户的功能体验.

在 SaaS 应用隐私保护基本模型中,由于不同租户的隐私保护需求不同,其最终使用的分块策略也会有所不同,此时,对于相同的租户数据逻辑视图,采用单一的共享数据模式将无法实现所有租户的需求.因此本文在原有 SaaS 存储方案的基础上进行了扩展,首先根据租户定制的隐私保护策略对数据进行分块,然后根据租户 SLA 协议和分块策略将其映射到合适的存储模式,最后将分块后的数据在所映射存储模式中进行存储,以实现 SaaS 模式下的数据隐私保护,同时为隐私服务的定价提供了硬件基础.

如图 2 所示,假设有 3 个租户 $\{A_1, A_2, A_3\}$,他们具有相同的逻辑表结构,字段 *tenantID* 用来记录数据所属的租户,租户 $\{A_1, A_2\}$ 使用相同的分块策略 $\{(user, age), (zipcode, disease)\}$,租户 A_3 使用的分块策略为 $\{(user), (age, zipcode, disease)\}$.首先根据 3 个租户的分块策略对租户的隐私数据进行分块,由于租户 $\{A_1, A_2\}$ 使用相同的分块策略,因此在数据存储时可以共享相同的存储模式 (C_AX, C_BX) ,模式中的每一个分块中都需记录数据所属租户的 *TenantID*.由于租户 A_3 的分块策略和租户 $\{A_1, A_2\}$ 不同,因此无法与租户 $\{A_1, A_2\}$ 共享存储模式 (C_AX, C_BX) ,此时需要为租户 A_3 单独创建新的存储模式 (C_EX, C_FX) 对数据进行存储.创建一个新的存储模式,需要付出一定的计算、存储以及后续维护代价,而随着一个存储模式共享程度的提高,其创建和维护的单位代价也会越来越低,因此,对应租户的服务定价也应相应降低.本文提出使用因子的概念对资源共享程度进行描述,用 $f(0 < f \leq 1)$ 表示, f 越大,共享程度越低,定价相应越高.

在上述数据存储模式中,为了混淆具有相同隐私约束的不同数据分块中同一条记录的关联关系,采用非完全可信的服务运营商与可信第三方协作的方式,利用乘法同态加密函数生成数据分块间的映射关系,并为同一租户下的同一条记录在不同数据分块上生成相应的 *CID*,用来标识租户数据之间的关联关系.由于攻击者不能获得分块的加密函数,租户数据的隐私获得了良好的隐藏保护.

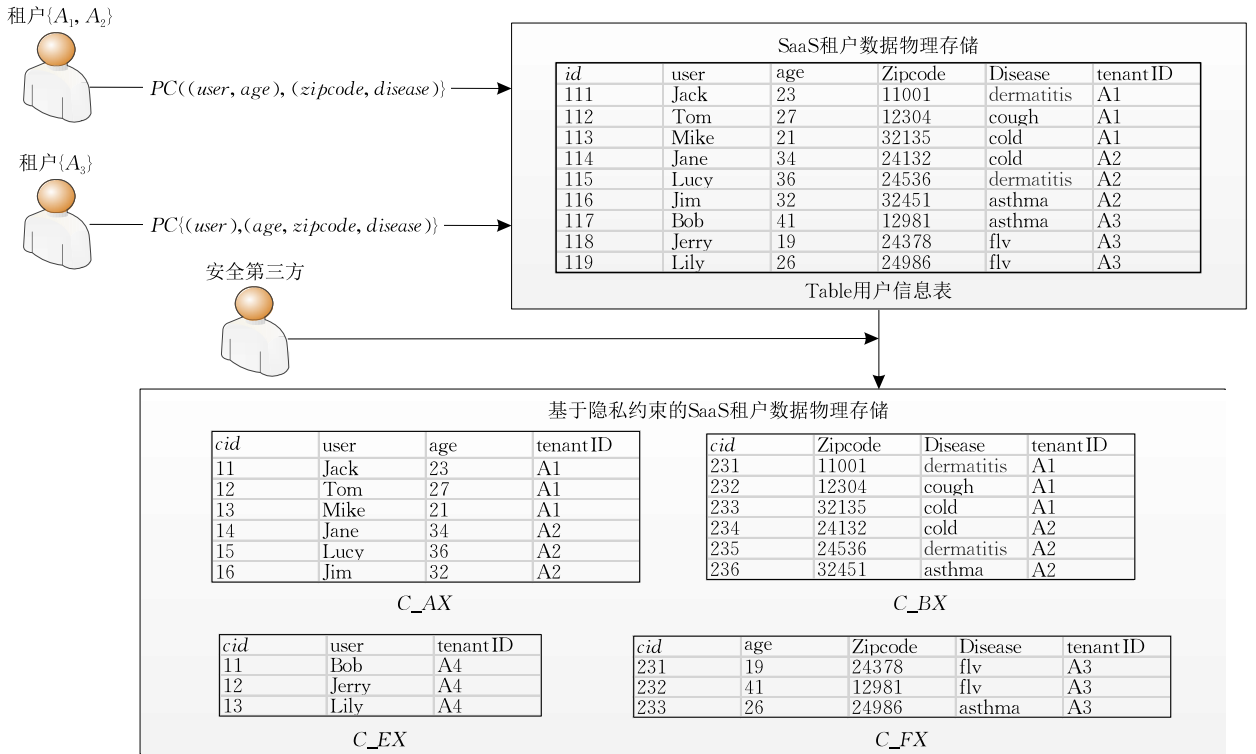


图2 隐私保护数据存储方案

3.3 帕累托最优状态

文献[5]提出通过纳什均衡的思想进行服务提供商与租户之间的一对一博弈,博弈时在保证纳什均衡状态的前提下,使得本次参与博弈的双方利益最大化,但最终得到的只是一个局部最优解.纳什均衡适合双寡头博弈的情况,而帕累托最优,综合考虑所有参与博弈的对象,是资源分配的一种理想状态.在帕累托最优状态下,不存在能使某一方变得更好而所有博弈方都没有变坏的调整方式,总体上达到一种资源分配的最优状态,即取得全局最优解.

定义 1. 帕累托最优^①(Pareto Optimality)是资源分配的一种理想状态,假定固有的一群人和一些可分配的资源,从一种分配状态到另一种分配状态的变化中,在不使其他人的利益减少的情况下使得至少一个人变得更好称作“帕累托改进”.不存在任何帕累托改进的状态称为帕累托最优状态.

定义 2. 帕累托支配(Pareto Dominate, PD),一个向量 $u=(u_1, \dots, u_m)$ 支配(或非劣于)另一个向量 $v=(v_1, \dots, v_m)$,当且仅当 $S(u_i) \geq S(v_i)$, ($\forall i \in (1, \dots, m)$)且 $P(u) \geq P(v)$ (其中至少有一个严格不等式成立),表示为 $u < v$.其中 $S(u_i)$ 表示根据 u_i 计算得到的租户满意度, $P(u)$ 表示根据 u 计算得到的服务商总利润.

由于需要不断进行迭代测试以判断当前状态是

否还存在可用的帕累托调整,因此选择一种合适的迭代算法对提高整个定价过程的效率就尤为关键.粒子群优化算法^②(Particle Swarm Optimization)是近些年发展起来的一种新的进化算法,也是一种随机搜索算法,该算法模拟鸟群捕食从无序到有序的特点,找出可行解中的最优解.其中,每个粒子代表一个潜在解,给粒子赋予速度向量,让其在可行空间中飞行,同时,根据自己与同伴的经验时刻调整自己飞行的方向,最终到达平衡状态.由于粒子群优化算法操作简单、收敛速度快,在函数优化、图像处理、大地测量等众多领域都得到了广泛的应用.因此,本文采用该算法来探寻帕累托最优状态.

3.4 隐私保护服务定价

在 SaaS 模式中,租户需要根据自身情况对应用(组件)进行按需定制,签订 SLA 协议并提交个性化的隐私保护策略对隐私信息进行保护.基于此模式,本文提出了隐私保护服务定价过程,下面首先给出该过程相关的定义.

定义 3. 云服务需求(Cloud Service Demand, CSD),用来表示租户对 SaaS 应用租赁情况的形式化描述,用四元组 $\langle ACD, PPS, PR, SD \rangle$ 表示,其中

^① http://en.wikipedia.org/wiki/Pareto_Optimality

^② http://en.wikipedia.org/wiki/Particle_Swarm_Optimization

ACD 为应用定制请求, PPS 为隐私保护策略, PR 表示应用性能要求, SD 表示存储空间需求。

定义 4. 应用定制请求 (Application Customization Demand, ACD), 表示租户对各应用组件的租赁情况, 用 R 表示, $R = \{r_i | 1 \leq i \leq M, r_i \in \{0, 1\}\}$, 其中 M 表示应用组件数量, r_i 取 0 时表示没有租赁基础应用 App_i , 取 1 表示租赁了 App_i 。

定义 5. 隐私保护需求 (Privacy Preserving Demand, PPD), 租户存储在提供商处的隐私数据不希望被窃取或泄露的需求, 也可以称为隐私约束, $PC\{(a1, a2), (a3, a4, a5), \dots\}$. 隐私约束中的各属性集合如 $(a1, a2), (a3, a4, a5), \dots$ 中的属性不能同时出现在同一个数据分块中, 否则会泄漏租户的隐私。比如, {姓名, 疾病} 是一条隐私约束, 如果姓名李雷与所患疾病癌症出现在同一个数据分块中, 数据库管理员就可以获知, 李雷患了癌症这条信息, 而这是租户不希望泄漏的隐私信息。

定义 6. 应用性能要求 (Performance Requirement, PR), 表示租户对所租赁应用提出的如正确性、响应时间等性能的最低要求。为便于算法描述, 本文只使用响应时间作为性能要求的评价指标。

定义 7. 存储空间需求 (Storage Demand, SD), 表示租户所持有的数据存储到云服务提供商处需要占用的存储空间大小。

在图 3 所示的隐私保护服务定价过程中, 云资源分配过程包括基础应用资源分配、隐私保护资源分配和存储资源分配。其中对各类资源的共享程度,

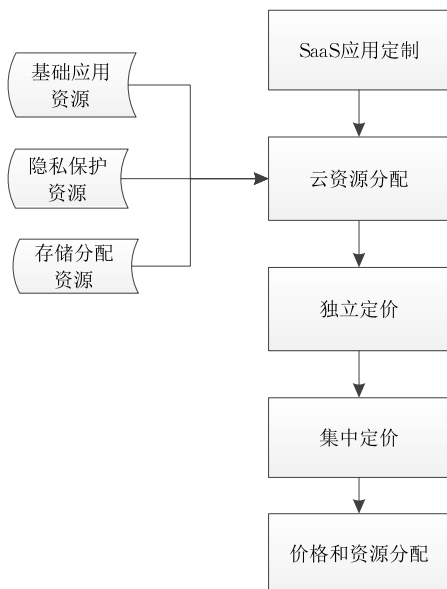


图 3 隐私保护服务定价过程

用使用因子 $f(0 < f \leq 1)$ 进行描述, 共享程度越高, 使用因子小, 反之当只有一个租户使用该资源时, 使用因子最大为 1。

基础应用资源分配主要负责计算租户租赁各应用组件所需要的计算资源量, 用 R_c 表示。设 R_{App_i} 表示服务运营商部署应用组件 $App_i (1 \leq i \leq m)$ 所需要的资源, f_i 表示租户对 App_i 的使用因子, r_i 表示租户的应用定制情况, 与定义 4 中描述一致, 则租户的基础应用资源需求: $R_c = \sum_{i=1}^M R_{App_i} \times r_i \times f_i$ 。

隐私保护资源分配负责计算要同时满足租户的隐私保护需求和应用性能所需要的计算资源, 用 R_p 表示。设 R_{PPM_i} 表示服务运营商部署隐私保护模块 $PPM_i (1 \leq i \leq n)$ 所需要的计算资源, f_i 表示租户对 PPM_i 的使用因子, r_i 表示是否分配 PPM_i 为该租户提供隐私保护服务, 则租户的隐私保护资源分配量:

$$R_p = \sum_{i=1}^N R_{PPM_i} \times r_i \times f_i.$$

存储资源分配负责为租户数据分配存储空间, 用 R_s 表示。设租户数据量大小用 $dataSize$ 表示, 则 R_s 与 $dataSize$ 成正比, 表示为 $R_s = dataSize \times l \times f_i$, 其中 $l (l \geq 1)$ 为比例系数, 用于校正因分块引起的空间消耗, f 为数据架构使用因子。

设隐私保护过程所用时间为 t_p , 则 t_p 随着分块数和数据量的增加会线性增加, 随隐私保护资源的增加而减少, 即 $t_p = \frac{k_1 \times constraints \times dataSize}{k_2 \times R_p}$, 其中 $constraints$ 为分块数, $dataSize$ 为数据量, k_1, k_2 为两个调整因子。设不进行隐私保护时应用的响应时间为 t_0 , 则在增加隐私保护处理过程后的总响应时间 $T = t_0 + t_p$ 。

上述模型中, 租户向 SaaS 应用定制平台提交云服务需求 CSD, 然后将需求信息分发给各资源分配模块计算满足租户需求所需要的最低资源分配; 资源分配完成后, 由独立定价模块收集资源分配结果并参考相似需求在上一周期的定价实施独立定价; 最后在集中定价阶段, 根据租户的服务定价与应用性能建立效用函数, 并使用多目标粒子群算法根据最优解的帕累托支配关系求解最终的服务价格以及资源分配。

4 隐私保护服务定价模型的实现

本文提出的隐私保护服务定价模型分为两个阶

段：独立定价与集中定价。独立定价参考同需求（或需求相近）的历史定价数据，根据资源占比情况给出价格；集中定价根据多租户提出的资源申请量与隐私保护服务需求建立效用函数，并使用多目标粒子群算法根据最优解的帕累托支配关系进行求解。

4.1 独立定价策略

供应商根据相似需求的历史定价数据以及当前资源使用占比进行估价。

$$P_{ins} = (P_{his} - cost)\alpha + cost \quad (1)$$

其中： P_{ins} 表示实时价格； P_{his} 表示上一周期相同或相似需求的定价； $cost$ 为需求对应的资源成本，见式(5)。 α 是资源使用占比，即当前使用掉的资源量与供应商持有的资源总量的比值，公式如下：

$$\alpha = \sqrt{1 + \frac{used_resource}{total_resource}} \quad (2)$$

其中： $use_resource$ 表示当前已分配出去的资源量； $all_resource$ 表示供应商的资源总量。由式(1)与(2)可见， α 越大，证明供应商已分配出去的资源越多，也就是当前持有的可用资源越稀缺，价格相应要更高。

算法 1. 基于资源使用占比的独立定价算法 (HPSPA).

输入： $round$ ； CSD_RA ； $margin$ ； $stack$

输出： P_{ins}

1. IF ($round = 0$)
2. $P_{ins} = margin \times \alpha + cost$;
3. ELSE {
4. $similarity = MAX$;
5. $P_{his} = 0$;
6. $list = stack.peek()$;
7. $iterator = list.iterator()$;
8. WHILE ($iterator.hasNext()$) {
9. $tenant = iterator.next()$;
10. $distance = comparetoHistory(tenant, CSD_RA)$;
11. IF ($distance < similarity$) {
12. $similarity = distance$;
13. $P_{his} = tenant.price$;
14. }
15. }
16. $P_{ins} = (P_{his} - cost) \times \alpha + cost$;
17. }

算法 1 为独立定价策略，算法输入参数 $round$ 表示集中定价的轮数， CSD_RA 为云资源分配平台根据租户服务需求为其分配的最低资源量， $margin$ 为第一次独立定价无历史数据参考时提供商规定的

最低利润， $stack$ 用以存储每个周期给多租户的定价，其元素为 $list$ 类型，每个节点存储一个 $tenant$ 数据， $tenant$ 类中有 3 个成员变量，包括 id ， CSD_RA ， $price$ 。其中 id 用以标记租户身份， $price$ 为集中定价算法给出的结果。之所以使用栈结构存储周期数据，是根据栈结构后进先出的特点，栈顶元素永远存储上一周期最新的数据，独立定价算法需要搜索上一周期数据时只需获取栈顶元素即可。

第 1 行判断集中定价进行过的轮数， $round$ 等于 0，表示尚未进行集中定价，此时使用提供商给定的最低利润 $margin$ 代替式(1)中的 $(P_{his} - cost)$ 计算独立定价(第 2 行)。否则使用式(1)根据历史数据 P_{his} 计算独立定价(第 15 行)。

算法 1 中 4~15 行为计算 P_{his} 的过程。第 4 行变量 $similarity$ 表示当前租户的需求与历史需求的相似度，用一个最大常量初始化，这里可以赋值 65536， $similarity$ 值越小，证明需求的相似度越高。第 5 行变量 P_{his} 表示历史定价。算法第 6 行获取栈顶元素，第 7 行定义链表遍历指针，8~13 行遍历链表，调用算法 2 比对当前租户需求与历史需求的相似度，赋值给变量 $distance$ (第 10 行)，如果当前搜索的节点相似度高于 $similarity$ 记录的值，则用当前数据赋值给 $similarity$ 变量，并将租户的历史定价赋给变量 P_{his} 。

算法 1 只需遍历一次 $list$ 链表，复杂度为 $O(n)$ ，实时性较好。

算法 2. 历史数据比对算法 $comparetoHistory$ 。

输入： $tenant$ ； CSD_RA

输出： $distance$

1. $R_current = CSD_RA.R_c$;
2. $R_history = tenant.CSD_RA.R_c$;
3. $R_p_current = CSD_RA.R_p$;
4. $R_p_history = tenant.CSD_RA.R_p$;
5. $R_s_current = CSD_RA.R_s$;
6. $R_s_history = tenant.CSD_RA.R_s$;
7. $distance = k_1 \times Math.abs(R_current - R_history) + k_2 \times Math.abs(R_p_current - R_p_history) + k_3 \times Math.abs(R_s_current - R_s_history)$;
8. return $distance$;

算法 2 用以比对当前租户需求与历史需求的相似度。输入参数 $tenant$ 表示上一周期参与集中定价的租户，其中保存了其资源分配和定价，参数 CSD_RA 为本轮独立定价中云资源分配平台为租户分配的资源量。算法 1~6 行分别取出当前和历史的基础

应用资源、隐私保护资源和存储资源分配量,第 7 行根据当前和历史的资源分配量计算其相似度,参数 k_1, k_2, k_3 分别表示服务提供商对各类资源的重视程度,一般来说基础应用和隐私保护资源如(CPU、内存和缓存)的价格是要比存储资源(如硬盘)贵很多的,此时在计算需求近似度时要优先考虑基础应用和隐私保护资源的近似度。

4.2 集中定价策略

4.2.1 基于帕累托最优的隐私保护服务定价算法

集中定价策略根据多租户提出的资源申请量与隐私保护服务需求建立效用函数,并使用多目标粒子群算法根据最优解的帕累托支配关系进行求解.下面首先给出效用函数:

$$Satisfaction_i(p, t) = S_0 - \alpha p - \beta t \quad (3)$$

式中: $Satisfaction_i(p, t)$ 表示租户满意度,它由 p 跟 t 两个参数决定,其中, p 表示租赁资源与服务的单位价格, t 表示应用的响应时间; α, β 分别为价格跟响应时间的系数,表示租户对价格和响应时间的偏好度, $\alpha + \beta = 1$. 微观经济学中, α/β 为边际替代率,它表示租户在满意度不变的条件下,租户每提高一单位价钱所得到的性能提升. S_0 表示租户的初始满意度.

对于提供商来说,提供商的效用函数如下:

$$profit(n) = \sum_{i=1}^n (p_i - cost_i - x_i \times penalty_i) \quad (4)$$

式中: $Profit(n)$ 表示服务提供商的收益; p_i 表示租赁资源与服务的价格,是提供商的收入; $cost_i$ 表示资源的成本,是提供商的支出; $Penalty_i$ 表示提供商提供的服务性能没有满足 SLA 协议中的标准所缴纳的罚金. $x_i \in \{0, 1\}$, 取 0 表示给第 i 个租户提供的服务满足 SLA, 不需缴纳罚金, 取 1 表示需要缴纳罚金.

$cost$ 包含计算资源(CPU、memory)的成本与存储资源(disk)的成本.

$$cost = num_cmp \times cmp_price + num_storage \times storage_price \quad (5)$$

其中: $num_storage$ 表示分配给租户的存储资源量; num_cmp 表示分配给租户的计算资源数量,它受 SLA 协议与数据分块数影响.

cmp_price 为每单位计算资源的价格:

$$cmp_price = \frac{RC/unit_time}{cores} \quad (6)$$

$storage_price$ 为每单位存储资源的价格:

$$storage_price = \frac{RC/unit_time}{capacity} \quad (7)$$

其中: $unit_time$ 表示租赁周期; $cores$ 表示计算资源数量,一般为 CPU 核数; $capacity$ 表示存储资源数量,一般以 G 为单位; RC 表示相对成本:

$$RC = L \times MP \times S\% \quad (8)$$

新的生产商加入云, CRA 会对其提供的资源质量进行估分,用 S 来表示, MP 表示相似产品的市场价格, L 表示调整系数.

由于服务提供商的资源有限,分配给各个租户的资源不能超过提供商拥有的资源总量.

$$\begin{cases} \sum_{i=1}^n num_cmp_i \leq num_cmp_{total} \\ \sum_{i=1}^n num_storage_i \leq num_storage_{total} \\ t_i \leq t_{SLA} \end{cases} \quad (9)$$

粒子群算法中,每次迭代粒子的速度 v_i 与位置 x_i 更新公式如下:

$$\begin{cases} v_i^{t+1} = \omega v_i^t + c_1 r_1 (pBest_i - x_i^t) + c_2 r_2 (gBest - x_i^t) \\ x_i^{t+1} = x_i^t + v_i^{t+1} \end{cases} \quad (10)$$

其中:上标 t 表示迭代次数; ω 表示惯性权重,一般取 $0.9 \sim 0.4$ 线性递减值; c_1, c_2 为加速因子,通常取 $c_1 = c_2 = 2$; r_1, r_2 在 $(0, 1)$ 之间随机取值; $pBest$ 表示个体极值,记录单个粒子发现的最好的位置; $gBest$ 表示全体极值,记录整个种群经历过的最好的位置.

算法 3. 基于帕累托最优的集中定价算法 (POCPA).

输入: 种群规模 L , 最大迭代次数 T , 租户个数 N , 租户需求 $CSD(n)$

输出: 每个租户的资源分配 CSD_RA 与定价 P_opt

```

1. WHILE ( $t < T$ ) {
2.   IF ( $t = 0$ ) {
3.     FOR ( $i = 0; i < L; i++$ ) {
4.       initializing( $particle[i]$ );
5.        $pBest[i] = particle[i]$ ;
6.       IF ( $i = 0$ )
7.          $gBest = particle[i]$ ;
8.       ELSE{
9.         IF ( $gBest < particle[i]$ )
10.           $gBest = particle[i]$ ;
11.       }
12.     }
13.   }ELSE{
14.     FOR ( $i = 0; i < L; i++$ ) {
15.       IF ( $particle[i] < pBest[i]$ )
16.          $pBest[i] = particle[i]$ ;
17.       IF ( $pBest[i] < gBest$ )

```



```

18.   gBest = pBest[i];
19.   }
20.   }
21.   IF(gBest = gBestold)
22.     Break;
23.   changeVelocity();
24.   changePosition();
25.   }

```

算法 3 为集中定价策略, 算法输入 L 为种群的规模, 即设置多少个粒子来搜索最优解, 每一个粒子都表示成一个 W 维向量, 向量的每一个元素代表参与定价的某一租户的定价和资源分配. T 表示最大迭代次数, N 表示参与集中定价的租户个数, CSD 表示租户提出的服务需求. 算法 2~12 行为第一次迭代, 考虑到独立定价结果的合理性, 集中定价的最优解必然会出现在独立定价结果附近, 因此粒子的初始位置在独立定价结果附近随机选取(算法第 4 行). 算法 13~20 行根据定义 2 帕累托支配关系搜索个体极值和全局最优值, 算法第 10~16 行计算某一粒子的个体极值, 若根据当前粒子位置得到的计算结果优于根据当前的个体极值的计算结果则更新个体极值. 算法第 17~18 行用来探索全局最优解. 算法第 21 行表示, 若本轮迭代得到的全局最优解与上轮相同, 则此解即为最优解. 算法 23~24 行根据式(10)计算下一轮粒子搜索的速度与位置. 算法需要两次循环, 因此时间复杂度为 $O(n^2)$, 较独立定价算法增加一个数量级.

4.2.2 帕累托最优均衡点存在性证明

由帕累托最优的定义, 不存在任何帕累托改进的状态下达到帕累托均衡, 即无法使得其他人的利益不减少的情况下至少一个人变得更好. 下面通过边际替代率的定义给出帕累托最优均衡点存在的证明.

定义 8. 边际替代率(Marginal Rate of Substitution, MRS)在维持效用水平或满足程度不变的前提下, 消费者增加一单位一种商品的消费可以代替的另一种商品的消费数量.

由边际替代率的定义可知, 当两种商品的边际替代率相等时, 增加一单位一种商品的消费必然会减少另一种商品的消费, 即不存在任何帕累托改进, 也就是达到了帕累托均衡.

定理 1. 两种资源分配到 N 个租户的问题存在帕累托最优状态.

证明. 为 N 个租户分配两种资源 C, D , 分别代表计算资源 CPU、存储资源 disk, 已知计算资源

与存储资源的总量为 C_0, D_0 .

设效用函数为

$$U_i = C_i^{\alpha_{1i}} D_i^{\alpha_{2i}} (i = 1, \dots, n).$$

其中: α_{1i}, α_{2i} 分别代表 C, D 的分配弹性系数, 在 $\alpha_{1i} > 0, \alpha_{2i} > 0$ 范围内可随机分配.

已知

$$C_1 + \dots + C_n \leq C_0,$$

$$D_1 + \dots + D_n \leq D_0.$$

设租户 2 到租户 N 的平均效用需求为 $\bar{U}_i (i = 2, \dots, n)$, 求 $MAX(U_1)$.

令

$$L = U_1(C_1, D_1) + \sum_{i=2}^n \lambda_i [U_i(C_i, D_i) - \bar{U}_i] + \lambda_0 (C_1 + \dots + C_n - C_0) + \lambda_1 (D_1 + \dots + D_n - D_0),$$

分别对 C_i, D_i 求偏导

$$\frac{\partial L}{\partial C_1} = \frac{\partial U_1}{\partial C_1} + \lambda_0 = 0,$$

$$\frac{\partial L}{\partial C_2} = \lambda_2 \frac{\partial U_2}{\partial C_2} + \lambda_0 = 0,$$

$$\frac{\partial L}{\partial C_n} = \lambda_n \frac{\partial U_n}{\partial C_n} + \lambda_0 = 0,$$

$$\frac{\partial L}{\partial D_1} = \frac{\partial U_1}{\partial D_1} + \lambda_1 = 0,$$

$$\frac{\partial L}{\partial D_2} = \lambda_2 \frac{\partial U_2}{\partial D_2} + \lambda_1 = 0,$$

$$\frac{\partial L}{\partial D_n} = \lambda_n \frac{\partial U_n}{\partial D_n} + \lambda_1 = 0.$$

由上可得

$$\frac{\partial U_1 / \partial C_1}{\partial U_1 / \partial D_1} = \dots = \frac{\partial U_n / \partial C_n}{\partial U_n / \partial D_n},$$

即

$$MRS_{CD}^1 = \dots = MRS_{CD}^n. \quad \text{证毕.}$$

上式表示不同租户对计算资源、存储资源的边际替代率均相等, 即如果增加一个租户的收益必定会降低另一个租户的收益, 于是该状态达到帕累托最优均衡状态.

但有时可能会有分配多种资源到多个租户的情况, 下面进行推广, 针对此情况给出以下定理并证明.

定理 2. 多种资源分配到 N 个租户的问题存在帕累托最优状态.

证明. 假设存在 n 种资源 R_1, \dots, R_n , 分别代表计算资源 CPU、存储资源 disk 和带宽等, 已知 n 种资源各自的总量分别为 R_{1D}, \dots, R_{nD} .

对于 N 个租户偏好既定,效用函数为

$$U_i = R_{1i}^{\alpha_{1i}} \cdots R_{ni}^{\alpha_{ni}} \quad (i = 1, \dots, N),$$

其中 $\alpha_{1i} > 0, \dots, \alpha_{ni} > 0$ 代表各种资源的分配弹性系数,在范围内可随机分配.

提供商提供的各种资源量有限,所以分配给各租户的各种资源总和不大于提供商持有的各种资源量:

$$R_{i1} + \cdots + R_{iN} \leq R_{iD} \quad (i = 1, \dots, N).$$

设租户 2 到租户 N 的平均效用需求为 $\bar{U}_i \quad (i = 2, \dots, n)$, 求 $MAX(U_1)$.

$$L = U_1(R_{1L}, \dots, R_{nL}) +$$

$$\sum_{i=2}^N \lambda_i [U_i(R_{1i}, \dots, R_{ni}) - \bar{U}_i] +$$

$$\delta_1 (R_{1D} - (R_{1L} + \cdots + R_{1N})) + \cdots +$$

$$\delta_n (R_{nD} - (R_{nL} + \cdots + R_{nN})).$$

对各种资源变量 R_1, \dots, R_n 求偏导,如下:

$$\frac{\partial L}{\partial R_{11}} = \frac{\partial U_1}{\partial R_{11}} - \delta_1 = 0$$

$$\frac{\partial L}{\partial R_{12}} = \lambda_2 \frac{\partial U_2}{\partial R_{12}} - \delta_1 = 0$$

...

$$\frac{\partial L}{\partial R_{1N}} = \lambda_N \frac{\partial U_N}{\partial R_{1N}} - \delta_1 = 0$$

...

$$\frac{\partial L}{\partial R_{n1}} = \frac{\partial U_1}{\partial R_{n1}} - \delta_n = 0$$

$$\frac{\partial L}{\partial R_{n2}} = \lambda_2 \frac{\partial U_2}{\partial R_{n2}} - \delta_n = 0$$

...

$$\frac{\partial L}{\partial R_{nN}} = \lambda_N \frac{\partial U_N}{\partial R_{nN}} - \delta_n = 0.$$

由上,得出

$$\frac{\partial U_1}{\partial R_{11}} = \lambda_2 \frac{\partial U_2}{\partial R_{12}} = \cdots = \lambda_N \frac{\partial U_N}{\partial R_{1N}} = \delta_1,$$

$$\frac{\partial U_1}{\partial R_{n1}} = \lambda_2 \frac{\partial U_2}{\partial R_{n2}} = \cdots = \lambda_N \frac{\partial U_N}{\partial R_{nN}} = \delta_n,$$

即

$$1 = \frac{\partial U_1 / \partial R_{1L}}{\partial U_1 / \partial R_{iL}} = \cdots = \frac{\partial U_N / \partial R_{1N}}{\partial U_N / \partial R_{iN}} \quad (i = 2, \dots, n),$$

$$MRS_{R_1, \dots, R_n}^1 = \cdots = MRS_{R_1, \dots, R_n}^N. \quad \text{证毕.}$$

上式表示不同租户对 n 种资源的边际替代率均相等,即增加一个租户的收益必定会降低另一个租户的收益,表明该状态达到帕累托最优均衡.

4.2.3 帕累托最优均衡与纳什均衡

前期工作^[5]中,我们使用了纳什均衡思想对隐私保护服务定价建立博弈模型,最终确定资源分配

与定价,从而生成隐私保护策略.本文使用帕累托最优的思想建立服务定价模型,并使用多目标粒子群优化算法求解出资源分配与定价结果.本文的贡献与创新点在于以下几个方面.

(1) 文献[5]给出的博弈模型更适合双寡头情况,即两方博弈.因为现实世界中,我们不能强制不同租户与它们的隐私保护服务需求同时到来,因此一个时间点上,只能进行服务提供商与单一租户的博弈.这种情况下,定价服务需耗费大量的计算资源与时间.本文创造性地提出两阶段定价机制,对于异步到来的租户与隐私保护服务请求,我们使用独立定价算法根据历史定价数据快速计算出初始服务价格,一定周期后,再对积累的多租户建立集中定价模型,使用多目标粒子群优化算法根据帕累托支配关系确定最优的服务定价.而文献[5]所采取的定价策略只是根据当前需求确定的租户隐私保护等级、应用性能、租赁费用.

(2) 资源管理是云计算的核心部分,资源定价是其合理使用的其中一种管理方法.文献[5]给出的博弈模型主要考虑的是资源定价的问题.服务提供商希望资源定价越高越好,而租户希望定价越低越好,通过双方博弈来解决服务提供商与租户双方利益最大化的矛盾问题.本文提出的服务定价模型同样解决了双方利益最大化的问题,同时保证在帕累托最优均衡状态下,资源分配最合理,资源利用率最高.

(3) 纳什均衡思想是保证在均衡状态下,博弈双方认为当前自己是利益最大化的.而帕累托最优均衡状态下,不仅参与方都达到利益最大化,同时达到全局最优分配.举例如图 4 所示.

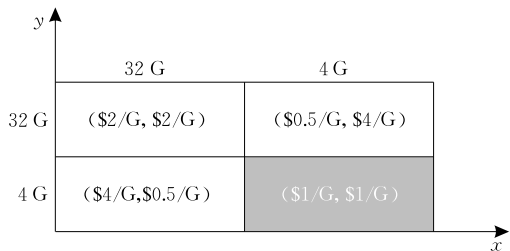


图 4 帕累托最优均衡与纳什均衡下双方的收益

横轴纵轴分别表示不同的租户 x, y . 对于租户 y 来说,如果他申请 32 GB 内存,对方也申请 32 GB 内存,他们的价格都是 2 元/GB,如果对方只申请 4 GB 内存,他将按 0.5 元/GB 收费;如果 y 申请 4 GB 内存,对方申请 32 GB 内存,他将按 4 元/GB 收费,

如果对方也申请 4GB 内存,他们都只需 1 元/GB.

在不知道对方如何选择的情况下, y 认为自己申请的多比较划算,同理, x 也选择 32GB,最终他们将按(2 元/GB, 2 元/GB)收费,此时达到纳什均衡点,参与者都满意,然而这不是帕累托最优的.帕累托最优允许参与者互相传递信息,在 x 确定 y 只购买 4GB 的情况下自己选择 4GB 收益最高(4GB 内存可以满足需求的情况下),此时的均衡状态为全局最优状态,对于租户来说收益最大,对于提供商来说资源利用率最高.

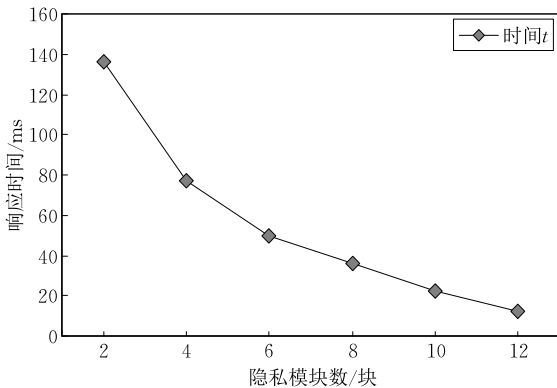
5 实验

为了验证本文提出的基于帕累托最优的隐私保护服务定价模型的有效性和性能,我们搭建了一台服务器,配置为 8 核 CPU Inter® Xeon® 1.80 GHz, 8GB 内存,1TB 硬盘.系统采用 Red Hat Enterprise

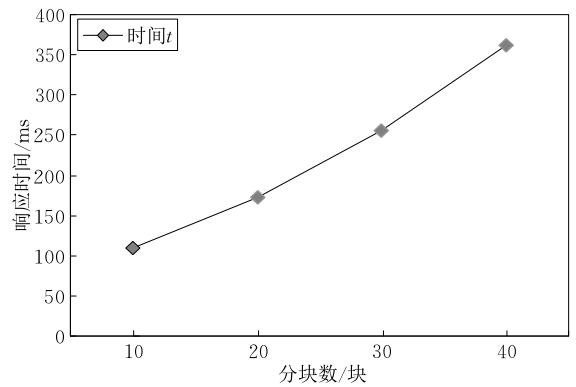
Linux6.2 版本,Apache Tomcat 作为应用服务器,测试数据库采用 Oracle Database 11g Release 2,编程环境选用 Eclipse-SDK-3.4.3-win32,编程语言为 Java6.

为了测试响应时间与隐私保护模块、数据分块之间的关系,在应用服务器端模拟了 300 次/s 服务请求,不断增大隐私保护模块、数据分块的数量观察响应时间的变化,为了简化实验,假设每一个隐私保护模块隐私保护需求和应用性能要求是不变的.

从图 5(a)中可以看出随着隐私保护模块的不断增多,响应时间不断减小,两者近似线性关系,因为随着隐私保护模块的增加,分配的资源越多,应用处理速度越快.图 5(b)中假设数据量不变,很明显分块数与响应时间呈线性正相关.这是因为,租户数据的分块数越多,数据块之间的产生平均连接次数就越多,从而增加了应用的响应时间.



(a) 响应时间与隐私约束的关系



(b) 响应时间与分块数的关系

图 5 响应时间分别与隐私约束和分块数的关系

图 6 模拟了提供商对 10 个租户的服务定价,10 个租户的数据量从 1T 到 25T 等量递增,隐私约束个数从 10 个到 100 个等量递增.为了方便计算,实验中的资源分别为 CPU、内存和磁盘,定价范围参考阿里云^①,如表 1 所示.

表 1 资源价格表

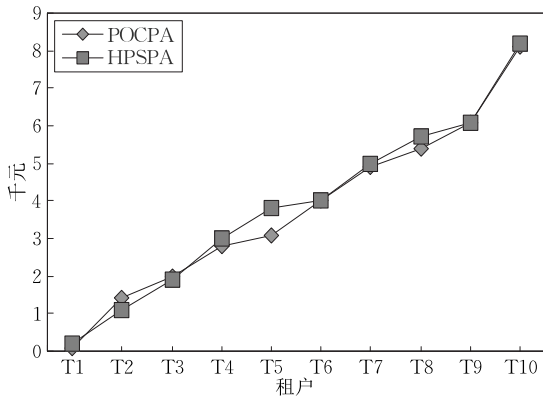
CPU/(元/核)	磁盘/(元/GB)	内存/(元/TB)
50	25	300

根据式(1)求出对应的响应时间,再根据式(2)求出满足租户提出的性能需求所需的计算资源.分别使用独立定价算法 HPSPA、集中定价算法 POCPA 给出服务定价如图 6(a)所示,求出租户的满意度 *Satisfaction* 与提供商的收益 *Profit* 并以文献[5]提出的纳什均衡算法 CIA 进行对比.其中,先进行

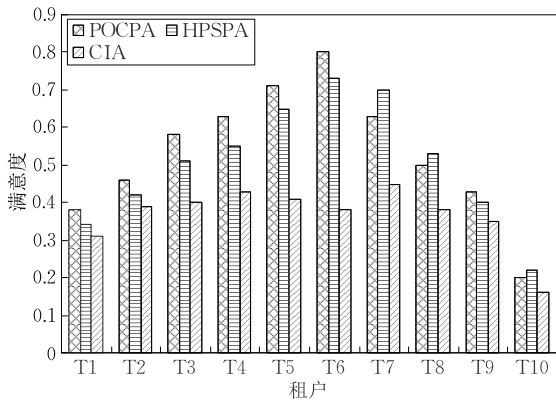
集中定价算法,得到的数据作为独立定价的历史参考数据.假设最高收益对应的满意度为 80%(实验图形随该值的变化等比例伸缩,数据变化趋势不受影响),以此为基准将全部结果转化成百分值,如图 6(b)和(c)所示.

图 6(a)为独立定价和集中定价阶段不同租户的服务定价,因为 10 个租户的数据量和隐私保护等级是等量递增的,所以两种算法的曲线整体呈上升趋势.从图中可以看到,独立定价和集中定价阶段对于租户的定价差别并不是很大,原因在于独立定价阶段给出的价格本身就是一个相对合理的价格,在集中定价阶段只是根据帕累托最优对部分租户的价格和资源分配做了调整,保证租户满意度和服务商

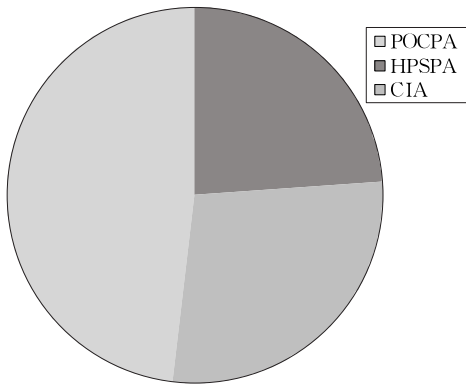
① [Http://www.aliyun.com/](http://www.aliyun.com/)



(a) 2种算法下不同租户的定价



(b) 3种算法下不同租户的满意度



(c) 3种算法下提供商的收益

图 6 不同算法下租户的定价、满意度和提供商的收益的情况

利润不会降低的同时实现了资源分配的最优。

从图 6(b)可以看出,随着隐私需求和资源需求量的增加,3 种算法下不同租户的满意度呈钟形曲线走势变化,该走向较明显的是 HPSPA 算法和 POCPA 算法.对于服务提供商来说,减少资源闲置收益最高,因此对于资源申请较多的租户定价较优惠,因此前期租户的满意度随着需求的增加而增加.由于提供商的资源有限,当租户提出的隐私需求与资源需求过大时,提供商会提高服务定价以避免剩余资源不足分配给其他用户的情况,因此后期租户

的满意度随着需求的增加反而呈下降趋势,且下降幅度比前期上升幅度更大. HPSPA 算法是参考历史同需求的价格跟当前资源使用占比进行的定价,这里参考的历史数据是 POCPA 算法得出的,因此前期走势同 POCPA 算法,随着需求的增加租户满意度也相应增加,后期由于资源使用占比越来越大,而服务定价与资源使用占比的增加而相应提高,因此租户的满意度逐渐下降. POCPA 算法下,租户的满意度,整体看,要比 HPSPA 算法下的高,这是因为 POCPA 算法下的结果是全局最优的. CIA 算法的租户满意度大致相同,同时低于 HPSPA 算法跟 POSPA 算法,这是由于 CIA 算法中,各租户和提供商只考虑自身利益最大化,多方参与者互相牵制,当达到纳什均衡点处,参与者认为当前价格下自己的收益最大,然而并不是一个全局最优的结果.

图 6(c)为 3 种算法下提供商的收益比例,很明显,POCPA 算法下提供商的收益最高,CIA 算法次之,HPSPA 算法下提供商的收益最低.正如上段分析的,HPSPA 算法给出的价格比较偏重于租户的收益,以此来吸引租户对服务的申请,因此提供商的收益相对较低. POCPA 算法综合考虑多方利益,合理分配资源,因此有一个全局最优的收益. CIA 算法参与者多方互相牵制,因此提供商的收益相应减少.

为了验证本文提出的基于帕累托最优思想的多目标粒子群优化算法的有效性,我们采用 CloudSim 模拟器对云计算环境进行仿真.实验中模拟 500 个租户跟 1 个提供商,每个租户的隐私需求与资源需求随机分配.图 7 为 POCPA 算法跟 CIA 算法的时间开销随租户数增加的变化趋势.租户数目从 16 个开始,每次增加 32 个.开始,POCPA 算法的时间开销增长缓慢,到 112 个租户左右,算法的时间开销呈线性增长趋势.当租户数目达到 400 左右时,POCPA 算法开始趋于收敛,即租户数再增加时,多目标粒子

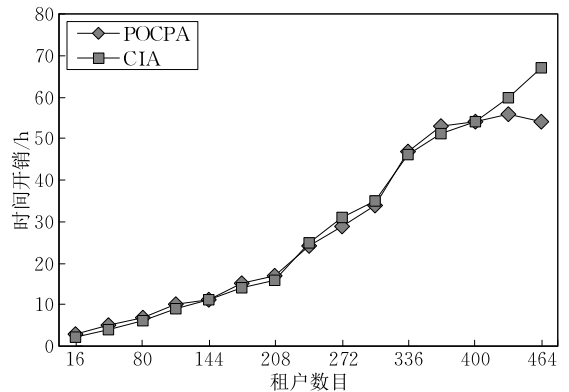


图 7 POCPA 与 CIA 算法的性能对比分析

群算法的时间开销不会继续加大,因此云服务提供商不必担心因业务规模扩大造成的全局缓存策略生成效率不可控的问题.此时,算法的执行时间是55个小时左右,不到3天,在一个集中定价周期内完全可以接受.可以看到,当租户增加到500个时,CIA算法依然呈线性增长趋势,没有任何收敛现象,可见,粒子群优化算法能够较快收敛于最优解.

6 总 结

本文首先分析了当前的服务定价模型,发现现有的模型给出的定价,并不能使租户和云服务提供商同时满意.而且,在定价时,也没有考虑到租户数据的隐私保护问题.鉴于此,本文基于帕累托最优的思想,综合考虑租户基础应用需求和隐私保护需求,提出一种可以使租户和服务提供商都满意的两阶段定价模型,并且对模型最优解的存在性给出了证明.实验结果证明,相比独立定价算法HPSPA和纳什均衡算法CIA,基于帕累托最优思想和粒子群算法的POCPA算法具有最优的租户满意度和最高的服务商利润,实现了租户和服务商的共赢,并且随着租户规模的增加,其算法执行效率也会逐渐收敛,具有更好的实用性.

下一步将重点研究租户需求发生改变时,对服务定价和资源分配的最优调整策略.

参 考 文 献

- [1] Li Lin, Li Qing-Zhong, Shi Yu-Liang, Zhang Kun. SAPS—A single attribute protection scheme for SaaS. *Information*, 2012, 15(1): 275-282
- [2] Zhang Kun, Li Qing-Zhong, Shi Yu-Liang. Research on data combination//Proceedings of International Conference on Tools with Artificial Intelligence. Dayton, USA, 2008: 423-430
- [3] Shi Yu-Liang, Jiang Zhen, Zhang Kun. Policy-based customized privacy preserving mechanism for SaaS applications. *Grid and Pervasive Computing*, 2013, 7861(0302-9743): 491-500
- [4] Shao Ya-Li, Shi Yu-Liang. A novel cloud data fragmentation cluster-based privacy preserving mechanism. *IJGDC: International Journal of Grid and Distributed Computing*, 2014, 7(4): 21-32
- [5] Yu Chao, Shi Yu-Liang. Game-theoretic strategy for personalized privacy protection. *International Journal of Grid and Distributed Computing*, 2014, 7(4): 123-138
- [6] Promise M, George S, Nomusa D, et al. A Comparative analysis of pricing models for enterprise cloud platforms//Proceedings of the African. Bambous, Mauritius, 2013: 1-7
- [7] Tayfun K, Nazim T. A pricing model for cloud computing service//Proceedings of the 47th Hawaii International Conference on System Science. Hawaii, USA, 2014: 699-707
- [8] Li Hui-Xi, Li Hao. A research of resource provider-oriented pricing mechanism based on game theory in cloud bank model //Proceedings of the International Conference on Cloud and Service Computing. Hong Kong, China, 2011: 126-130
- [9] Sui Xin, Leung Ho-Fung. An adaptive bidding strategy in multi-round combinatorial auctions or resource allocation//Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence. Dayton, USA, 2008: 423-430
- [10] Raouf O, Brunel Univ, Al-Raweshidy H S. Pricing-based resource allocation in ad-hoc networks using auction theory//Proceedings of the Future Network and Mobile Summit. Florence, Italy, 2010: 1-8
- [11] Hu Zhi-Gang, Liu Yan. Dynamic resource pricing based on combinatorial double auction in cloud environment. *Computer Engineering*, 2012, 38(8): 19-21(in Chinese)
(胡志刚, 刘艳. 云环境下基于组合双向拍卖的动态资源定价. *计算机工程*, 2012, 38(8): 19-21)
- [12] Sharma B, Thulasiram R K, Thulasiraman P, et al. Pricing cloud compute commodities: A novel financial economic model//Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Washington, DC, USA, 2012: 451-457
- [13] Rust R T, Kannan P K, Peng N. The customer economics of Internet privacy. *Journal of the Academy of Marketing Science*, 2002, 30(4): 455-464



SHI Yu-Liang, born in 1978, Ph.D., associate professor. His research interests include cloud computing, database and privacy preserving.

SHAO Ya-Li, born in 1987, M.S. Her research interests include cloud computing and privacy preserving.

ZHOU Zhong-Min, born in 1989, M.S. candidate. His research interests include cloud computing and privacy preserving.

ZHANG Hong-Lei, born in 1987, M.S. candidate. His research interests include cloud computing and privacy preserving.

CHEN Yu, born in 1990, M. S. candidate. His research interests include cloud computing and privacy preserving.

CUI Li-Zhen, born in 1976, Ph. D., professor. His

research interests include data science and engineering, service computing, and intelligent collaborative computing.

Background

The topic this paper researched belongs to the problem of resource pricing based on privacy preserving service in the cloud computing environment. The popular international solution to this kind of problem is a fixed proportion of pricing model used by Amazon EC2 and Microsoft Windows Azure etc. Besides, there are pricing models based on the cloud bank framework or dynamic gambling theory. However, all the methods above have only considered the profit of service provider. The paper proposed a resource pricing strategy based on Pareto optimum theory. It considers the profit of both service provider and the multi tenants, allocating resources rationally in the premise of both the provider and the tenants' satisfaction to the price. In the overall optimal state, the waste of resources is the least and the profit of both provider and tenants is the highest. The previous studies of our team in the direction of privacy preserving under the cloud computing environment include A New Privacy-Preserving Scheme DOSPA for SaaS, Data Combination Privacy Preservation Mechanism for SaaS, Policy-Based Customized Privacy Preserving Mechanism for SaaS Applications, A Novel Cloud Data Fragmentation Cluster-based Privacy Preserving Mechanism, Document-oriented Database-based Privacy Data

Protection Architecture, and Game-Theoretic Strategy for Personalized Privacy Protection.

The research work was supported by the National Natural Science Foundation of China under Grant Nos. 61272241, 61572295, the Innovation Methods Work Special Project No. 2015IM010200, the Taishan Industry Leading Talent of Shandong Province, the Natural Science Foundation of Shandong Province under Grant Nos. ZR2014FM031, ZR2013FQ014, the Shandong Province Independent Innovation Major Special Project Nos. 2015ZDXX0201B03, 2015ZDXX0201A04, 2015ZDJQ-01002, the Shandong Province Key Research and Development Plan No. 2015GGX101015, the Fundamental Research Funds of Shandong University No. 2015JC031.

The result of this paper showed the applications are different. Besides, the willing payments are also different. The existing SLA model is lack of support to the definition and description of personality and fine-grained privacy preserving demands. Hence, this paper Comprehensive considered the conditions of privacy preserving content, privacy preserving level, application performance and the service price, motivating and balancing the problem of data privacy through the pricing method.