

异构云计算体系结构及其多资源联合公平分配策略

王金海 黄传河 王 晶 何 凯 史姣丽 陈 希
(武汉大学计算机学院 武汉 430072)
(wangjinhai@whu.edu.cn)

A Heterogeneous Cloud Computing Architecture and Multi-Resource-Joint Fairness Allocation Strategy

Wang Jinhai, Huang Chuanhe, Wang Jing, He Kai, Shi Jiaoli, and Chen Xi
(Computer School, Wuhan University, Wuhan 430072)

Abstract Resource allocation strategies are an important research hotspot about cloud computing at present. The most fundamental problem is how to fairly allocate the finite amount of resources to multiple users or applications in complex application under heterogeneous cloud computing architecture, at the same time, to achieve maximize resource utilization or efficiency. However, tasks or users are often greedy for classical resource allocation problems, therefore, under the condition of finite amount of resource, the fairness of resource allocation is particularly important. To meet different task requirements and achieve multiple types resource fairness, we design a heterogeneous cloud computing architecture and present an algorithm of maximizing multi-resource fairness based on dominant resource(MDRF). We further prove the related attributions of our algorithm such as Pareto efficiency, and give the definition of dominant resource entropy (DRE) and dominant resource weight (DRW). DRE accurately depicts the adaption degree between the resource requirement of user and the resource type of server allocated for user tasks, and makes the system more adaptive and improves the system resource utilization. DRW guarantees the priority of users obtaining resource when cooperating with the adopted Max-Min strategy guaranteeing fairness, and makes the system resource allocation more ordered. Experimental results demonstrate that our strategy has more higher resource utilization and makes resource requirements and resource provision more matching. Furthermore, our algorithm makes users achieve more dominant resource and improves the quality of service.

Key words multi-resource fairness allocation; dominant resource entropy(DRE); dominant resource weight(DRW); Max-Min Fairness; heterogeneous cloud

摘 要 资源分配策略是当前云计算研究领域中的一个重要研究热点,异构云计算体系结构下的复杂应用问题研究中,最基本的问题在于如何将总体有限的资源分配给多个租户或应用,以达到效率或收效最大化.但是,在经典的资源分配问题中,任务或者用户往往是“贪婪”的;因此,在总体资源有限的前提下,资源分配的公平性就显得尤为重要.为了满足不同的任务需求,达到多种资源分配的公平性,设计了一个虚拟化的异构云计算体系结构,提出了该体系结构下基于占优资源的多资源联合公平分配算法(maximizing multi-resource fairness based on dominant resource, MDRF),并且证明了算法的帕累托等相关属性;给出了占优资源熵(dominant resource entropy, DRE)和占优资源权重(dominant resource weight, DRW)的定义,占优资

源熵更加精确地刻画了用户资源请求与任务所调度到的服务器资源之间的适应程度,使系统的自适应能力更强同时提高了资源利用率. 占优资源权重保障了用户优先获取资源的优先次序,协同所采用保障公平性的 Max-Min Fairness 策略,使资源的分配更加有序. 实验表明,我们的策略有更高的系统资源利用率,并且使需求与供给更加匹配,进而使用户的占优资源获取更多,提高了服务质量.

关键词 多资源公平分配;占优资源熵;占优资源权重;最大最小公平性;异构云

中图法分类号 TP316

云计算环境下复杂应用问题的研究中,最基本的问题在于如何将总体有限的资源分配给多个租户或应用,以达到效率或收效最大化,在经典的资源分配问题中,任务或者用户往往是“贪婪”的,即获得越多的资源对于任务的完成就有利. 因此,在总体资源有限的前提下,特别是资源多样化的异构云计算系统中,资源分配的公平性就显得尤为重要.

以前的公平性问题研究中,大多着眼于单一资源的分配公平性问题,提出了诸如 Max-Min Fairness 等诸多公平性理论,由此衍生出各个工程方向上成熟的资源共享或调度机制,诸如网络队列调度中的 Fair Queuing, Round-Robin Scheduling 等调度策略就是对于 Max-Min Fairness 的实现方案. 而在云计算环境下,资源并不是单一的一种,包括计算资源(CPU)、内存资源、存储资源、网络资源(带宽)等,特别是面向复杂工程和科学应用的云计算环境,甚至包括 GPGPU, SSD 等异构计算、网络和存储资源. 不同的工程应用或者作业需要的资源也不尽相同,表现为不同的资源组合. 如何进行多资源类型的联合调度和控制以满足不同需求组合就成为云计算环境下的一个重要问题. 相对单一资源类型的分配问题而言,多资源类型分配中更加难以做到完全的公平性;同时,异构云计算体系结构下资源供给的多样性,更增加了资源公平分配的难度,尤其是多种资源联合的公平分配.

已有的研究成果中,占优资源公平分配(dominant resource fairness allocation, DRF)^[1]与异构环境下的占优资源公平分配(dominant resource fairness allocation in heterogeneous cloud, DRFH)^[2]的资源分配模型都是基于用户无限的任务请求数量这一假设,分别追求占优资源数量上的公平分配或者各用户任务数量上的公平分配. 但是,由于云计算中心的负载通常是具有很大浮动性的^[3-4],并且在不同时间片段内每个用户请求的任务数也是不同的,所以这些占优资源的公平性分配策略在很大程度上限制了资源公平分配策略的自适应性. 并且他们在资源

分配时仅仅考虑了占优资源和瓶颈资源(即约束任务分配数量的资源),而没有考虑其他资源. 我们的算法则综合评估了所有资源相对于占优资源的占优程度,给出了具体的量化方法,更具优越性.

已有机制缺少对包括占优资源在内的所有资源的综合评估,可操作性和适应性都较差,同时还可能造成较多较大的非占优资源和非瓶颈资源残片. DRFH 改进了 DRF 在云环境中的应用,但是文献[2]只是基于 Max-Min Fairness 作了简单的改进,虽然研究者也提出了分配过程中要遵循资源不浪费原则(non-wasteful),但是没有提出有效的度量方法,怎样做才算更 non-wasteful,作者并没有给出相应的准确度量方法,尤其是当占优资源在 2 个异构服务器上相同时并没有提出行之有效的方法. 此外,DRFH 算法对于 CPU 资源的分配是按占用的 CPU 时间;但是,异构云环境中 CPU 的处理能力差异很大,比如一个 10 GHz 处理能力的 CPU 与一个 6 GHz 处理能力的 CPU 在相同的时间内处理能力差别是相当大的,因此,这样的分配方法是不合适的,也是很不公平的.

虚拟化技术的成熟及应用提高了云计算数据中心的资源灵活部署和服务迁移的能力,虚拟机平台能够动态地将服务映射到所需要的物理资源,而且正在运行的应用服务程序不必停止或重启^[5-6]. 同时,虚拟机技术提供了性能和错误隔离的能力,以更细粒度的方式多维度按需配置资源^[7-9].

针对上述问题,本文设计了一个虚拟化的异构云计算体系结构,提出了该体系结构下基于占优资源的多资源联合公平分配算法(maximizing multi-resource fairness based on dominant resource, MDRF),并且证明了算法的帕累托等相关属性;给出了占优资源熵和占优资源权重的定义,占优资源熵保障了用户资源请求与任务所调度到的服务器资源的适应程度,使系统的自适应能力更强,提高了多个维度上的资源利用率. 占优资源权重保障了用户优先获取资源尤其是占优资源源的次序,协同所采用保障公

平性的 Max-Min 策略,使多用户对于资源的分配更加有序.实验表明,我们提出的策略具有更高的系统资源利用率,并且使用户的资源需求与系统的资源供给更加匹配,进而最终使用户获取的占优资源份额更多;同时,由于不同用户在同一服务器上获得的占优资源更加互补,减少了不同用户对占优资源竞争,提高了服务质量.此外,我们的策略比前面提到的 DRF 和 DRFH 策略更具灵活性,即可以满足不同时间片段内的各种需求,并且分配策略使所有用户最终达到帕累托最优(Pareto efficiency).DRF 与 DRFH 的前提是无限多的任务量,而实际情况是一个用户的任务请求通常是有限的并且具有很大的时变波动特性,比如说一个公共云中一个用户一次请求耗尽所有资源池中的占优资源通常是不现实的,在不时刻同一个用户对资源的请求量也具有很大的波动性.而我们提出的系统资源配置策略,是基于用户实际请求任务的资源分配,更加切合实际应用场景.

1 相关工作

尽管已有大量的关于资源公平分配的计算系统文献,但是这些工作大都局现于同构系统下讨论单一类型的资源分配,例如 CPU 时间^[10]、链路带宽^[11-14]等.近年来,研究者们提出了各种各样的有关公平性的概念,从特殊应用的分配^[15-16]到常用的公平性规则^[17-20]等.至于多资源分配,使用最先进技术(state-of-the-art)云计算系统也是采用天真的单一资源抽象技术.诸如,在当前的 Hadoop^[21-22]系统中,2 个公平的共享调度器支持将一个节点的任务以固定的资源比例划分到多个时间片段并且以时间片段为粒度联合分配资源;Quincy^[23],一个为 Dryad^[24]开发的公平调度器,建模公平调度问题为调度任务到时间片段的最小成本流(min-cost flow)问题.近来的工作^[25],虽然将工作部署约束考虑其中,但其仍是一个基于时间的单资源任务调度问题.

Ghods 等人^[1]最早系统地研究云计算系统中的多资源公平分配问题,研究者提出了多用户的 DRF,并且显示出诸多的令人满意的公平属性.DRF 快速地吸引了大量研究者的关注,并得到广泛推广.比较显著的成果,Joe-Wong 等人^[26]推广 DRF 量化并使其成为一个一体化的框架,以权衡分配公平和分配效率.Dolev 等人^[27]提出了另外一个多资源公平分配概念,也就是众所周知的基于瓶颈资源的公平分配(bottle-based fairness, BBF),在 BBF 中实现了

DRF 具有的 2 种公平性属性.Gutman 和 Nisari^[28]将 DRF 推广到更通用的用户实用工具,并且显示了其具有 BBF 机制的性质.另一方面,Parkes 等人^[29]用多种方法扩展 DRF,包括某些资源的零需求、权重用户倾斜以及特殊任务的不可划分情形等.近几年,他们都研究了 DRF 规则下的社会福利损失问题,Kash 等研究者们扩展了 DRF,允许用户随时可以加入系统但却不离开的情形.然而,所有这些相关的工作,都是或明或暗地指出所有待分配的资源都被集中在一个超级计算机上(仍可看作是一个单机系统),与目前广泛探讨的流行的云计算数据中心系统有着本质的不同.针对上述问题,Wei 等人^[2]提出一个基于 DRF 的异构云环境中的多资源公平分配算法 DRFH,对上述算法作出了较大改进,使其适应于异构的云环境中;但是,研究者工作是基于用户无穷多的任务假设,并且在任务调度过程中只关注用户的占优资源份额及调度中的服务器瓶颈资源而忽略了除此之外的其他资源,并且缺少行之有效的资源评估方法,操作性和用户资源需求与服务器资源供给的自适应性都比较差,从而导致了因瓶颈资源的限制而产生的较多资源空闲.

其他相关工作,包括经济学中的公平划分问题,尤其是 Leontief 的偏好平均主义(egalitarian division under Leontief preferences)^[30]和 cake-cutting 问题^[31-32],所有这些工作也都是假定一体化(all-in-one)的资源模型.因此,这些理论及策略都不能直接应用于异构的云计算系统中.

2 问题描述及相关定义

在本文提出的虚拟化异构云计算系统中,用虚拟机响应用户的服务请求,这里做一个约束,任务请求等同于虚拟机的请求,为讨论方便,在下文中约束任务请求即为虚拟机请求,任务的响应过程即为虚拟机部署过程也即资源分配过程.

2.1 用户请求及服务器资源模型

在云计算系统中,资源池(resource pool)由一系列的异构服务器组成,定义服务器集合: $S = \{1, 2, \dots, m\}$,每个服务器包含 k 种硬件资源,例如 CPU、内存(memory)、磁盘(disk)等.资源的种类集合定义为 $R = \{1, 2, \dots, k\}$,记 $r \in R$.对于每一个服务器 l ($l \in S$),记 $c_l = (c_{l1}, c_{l2}, \dots, c_{lk})$ 为该服务器的资源向量.这里, c_{lr} 为服务 l 中可用资源 r 的数量.不失一般性,对于每一种资源 r ,我们正则化所有服务器的

整个资源数量为 1,那么有:

$$\sum_{l \in S} c_{lr} = 1, \forall r \in R. \tag{1}$$

记 $U=\{1,2,\cdots,n\}$ 为共享云系统资源的用户集合,对于每一个用户 $i(i \in U)$,记其资源需求向量为 $\boldsymbol{d}_i=(d_{i1},d_{i2},\cdots,d_{ik})$,这里, d_{ir} 表示用户 i 的每个任务对资源 r 的需求量占整个资源池中资源 r 的比例(共享比例).简单起见,我们假设所有用户的资源请求都是正向的,即:

$$d_{ir} > 0, \forall i \in U, r \in R.$$

如果 $r_i^*=\arg \max_{r \in R} d_{ir}$, 我们说 r_i^* 为用户 i 的全局占优资源(global dominant resource, GDR). 换句话说, r_i^* 是整个资源池中用户 i 的任务中最重的资源需求.

作为一个简单的例子,我们考虑图 1 中 3 个异构服务器的情况. Server 1 是高 CPU 的,包含 60 GHz CPU 和 20 GB 内存;Server 2 和 Server 3 是高内存的,分别含有 20 GHz CPU 和 60 GB 内存、30 GHz CPU 和 60 GB 内存.由于系统一共包含了 110 GHz CPU 和 140 GB 内存,因此,3 个服务器正则化后的资源向量分别为

$$\begin{aligned} \boldsymbol{c}_1 &= (\text{CPU}, \text{memory})^\top = (6/11, 1/7)^\top, \\ \boldsymbol{c}_2 &= (\text{CPU}, \text{memory})^\top = (2/11, 3/7)^\top, \\ \boldsymbol{c}_3 &= (\text{CPU}, \text{memory})^\top = (3/11, 3/7)^\top. \end{aligned}$$

假定某段时间有 3 个用户共享系统,用户 1 是 CPU 密集型的任务,每个任务需要 5 GHz CPU 和 2 GB 内存;用户 2 和用户 3 是内存密集型的任务,分别需要 2 GHz CPU 和 5 GB 内存、3 GHz CPU 和 5 GB 内存.正则化后,用户 1 的资源需求向量为 $\boldsymbol{D}_1=(\text{CPU}, \text{memory})^\top=(1/22, 1/20)^\top$,CPU 为全局占优资源;用户 2 的资源需求向量为 $\boldsymbol{D}_2=(\text{CPU}, \text{memory})^\top=(1/55, 1/28)^\top$,内存为全局占优资源;用户 3 的资源需求向量为 $\boldsymbol{D}_3=(\text{CPU}, \text{memory})^\top=(3/110, 1/28)^\top$,内存为全局占优资源.

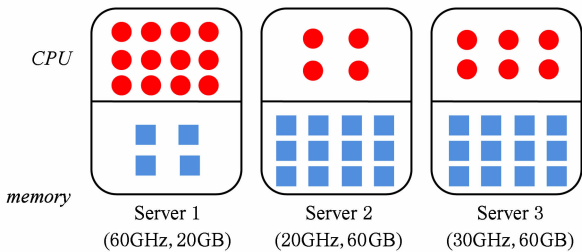


Fig. 1 An example of a system consisting of three heterogeneous servers shared by three users.

图 1 3 个用户共享的 3 个异构服务器的例子

在我们的系统中,所有的任务都用虚拟机来响应,每个虚拟机响应一个任务,由于云计算虚拟机加载可扩展特性即加载位置(虚拟机被加载到的服务器)无关特性,因此,每个用户的任务可在不同的服务器上执行.

2.2 DRF 与 DRFH 的无效性

引言提到的 DRF 分配策略考虑并解决的问题是在单机系统上(所有资源看成一个整体的集群)将占优资源等比例分配给用户.但是,在云计算中心,当资源分布到许多异构的服务器上时,DRF 分配只能是将服务器一个个地隔离对待,由于服务器是异构的,用户任务的资源需求也各种各样,因此,一个用户在不同的服务器上将会有不同的占优资源.例如,在图 1 中用户 1 与用户 2 和用户 3 有不同的占优资源.

在文献[2]中,研究者针对上述问题进行了探讨,提出在异构云计算中的 DRFH 分配策略,只是从占优资源和瓶颈资源 2 个角度进行资源分配,同 DRF 一样都是基于用户有无穷多的任务这一假设;同时,也没有考虑当用户有相同占优资源时的情况,比如用户 2 和用户 3 有相同的占优资源(内存)时如何处理,缺乏可操作性.此外,在真实的基于虚拟机技术的云环境中,虚拟机为任务分配的基本单位,涉及的资源也是多样的,不只是占优资源和瓶颈资源.因此,在进行资源分配时应该综合考虑各种资源以及各异构服务器的资源构成,将任务分配到最适合的服务上,减少各种不可分配的资源残片,提高利用率.以图 1 为例,在图 2 中图 2(a)和图 2(b)分别显示了 DRF 及 DRFH 在异构云环境中的具体分配情况.

从图 2(a)不难看出,DRF 分配算法的资源浪费情况相当严重,可以说该算法是失效的.虽然 DRFH 分配算法作了较显著的改进,但是每个服务器仍有较多的资源没用被充分利用.原因在于 DRFH 分配的独占性,事实上,随着任务分配的进程各服务器的可用资源也在发生动态变化,而 DRFH 分配算法没有考虑这一变化——影响资源利用率及最终可分配任务数的重要因素.

很显然,在相同的条件下(相同的输入、相同的系统可用资源),系统的资源利用率越高,其所承载的任务将越多.不幸的是,DRF 和 DRFH 策略在其假设的前提下做到了资源分配的公平性,但却违背了帕累托最优的高效性,原因是二者都缺乏对用户

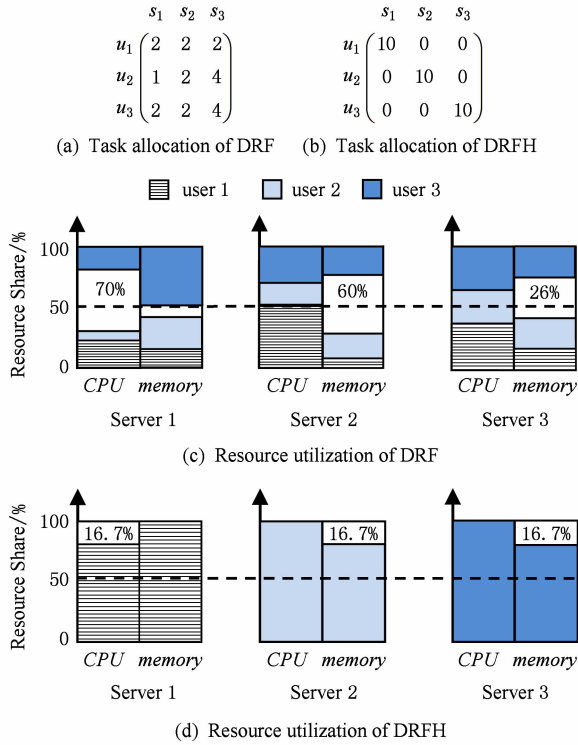


Fig. 2 DRF allocation and DRFH allocation.

图2 DRF分配及DRFH分配

任务资源需求及系统可用资源的精确量化. 在下文中, 我们从对资源(用户任务占优资源和服务器占优资源以及之外的所有资源)精确量化入手, 解决上述分配策略不能实现问题, 具有更强的操作性和更高的自适应性: 1) 用户占优资源与服务器占优资源的自适应; 2) 分配任务随意性, 打破了无穷多任务的假设.

2.3 资源分配机制及其属性

对每个用户 i 和 server l , 记 $\mathbf{A}_{il} = (A_{il1}, A_{il2}, \dots, A_{ilk})^T$ 为其资源分配向量, 这里 A_{ilr} 为用户 i 在 server l 上资源 r 的分享比例. 记 $\mathbf{A}_i = (\mathbf{A}_{i1}, \mathbf{A}_{i2}, \dots, \mathbf{A}_{in})$ 为用户 i 的资源分配矩阵. 记 $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n)$ 为所有用户的整个分配. 我们说一个分配 $\mathbf{A}$ 是可行的, 当且仅当没用一个服务器被用超任意一种资源, 即:

$$\sum_{i \in U} A_{ilr} \leq c_{ilr}, \forall l \in S, r \in R. \quad (2)$$

对于所有用户 i , 给定一个分配 $\mathbf{A}_{il}$, 则在 server l 上可被调度的最大任务数量按如下公式计算:

$$N_{il}(\mathbf{A}_{il}) = \min_{r \in R} \{A_{ilr}/d_{ir}\}. \quad (3)$$

在分配方案 $\mathbf{A}_i$ 下, 用户 i 被调度的任务总数可计算为

$$N_i(\mathbf{A}_i) = \sum_{l \in S} N_{il}(\mathbf{A}_{il}). \quad (4)$$

在分配任务的时候, 不仅需要提高用户的占优资源或者调度的任务数; 同时, 占优资源或者调度的任务数还取决于任务调度到一个服务器的瓶颈资源 $r_{il}^{\#}$ 并且有 $d_{r_{il}^{\#}} = \arg \min_{r \in R} \{A_{ilr}/d_{ir}\}$. 在异构云系统中, 由于用户需求是多样的, 因此任务的加载(消耗资源)和任务的离开(比如任务完成, 系统将资源释放到资源池)常常使得服务器的可用资源多样化. 为了减少资源浪费, 使任务调度到用户需求和供给更加吻合的服务器上, 下面我们借用信息论中最大熵的原理来刻画用户任务及服务器整个资源相对于占优资源的占优度.

定义 1. 占优资源熵(dominant resource entropy, DRE). 用来衡量用户任务资源需求相对于所调度到的服务器资源供给的占优资源适应度(dominant resource fitness rank, DRFR), 熵值越大则适应度越高, 说明虚拟机部署到该服务器的用户任务资源需求包括占优资源在内的所有资源与该服务器的可供资源越吻合. 记 r_i^* 为用户 i 的占优资源, 则用户 i 部署到 server l 上时, 用户 i 的任务资源需求相对于 server l 的资源占优度可计算为

$$E_i(\mathbf{A}_{il}) = - \sum_{r \in R} h(\mathbf{A}_{il}^r) \ln h(\mathbf{A}_{il}^r), \quad (5)$$

其中, $h(\mathbf{A}_{il}^r)$ 为用户的资源 $r(r \in R)$ 在分配方案 $\mathbf{A}_{il}$ 中相对于整个服务器 l 的资源(尤其是占优资源)的比重, 计算为

$$h(\mathbf{A}_{il}^r) = \frac{A_{ilr}/d_{ir}}{\sum_{r \in R} (A_{ilr}/d_{ir})}. \quad (6)$$

占优资源熵不但可以精细地评估用户与其所调度到的服务器各资源供给包括占优资源在内的整个资源的比重, 而且还同时刻画了二者的吻合程度. 熵值越大, 说明任务的资源需求与服务器的供给越吻合, 当各种资源都完全吻合时, 每种资源(包括占优资源)都成为瓶颈资源, 这时熵值最大, 这种任务调度到该服务器时不会造成任何资源的浪费. 因此, 占优资源熵为任务调度提供了更加精确的刻度, 使各服务器的资源残片不但更少而且更小. 当每种资源都成为分配中的瓶颈资源时, 熵值达到极大值.

定义 2. 占优资源权重(dominant resource weight, DRW). 用来衡量基于占优资源的用户在整个资源的综合权重, 可计算为

$$\omega_i = \sum_{r \in R} (d_{ir}/d_{r^*}), \quad (7)$$

其中, $d_{r^*} = \arg \max_{i \in U} d_{ir}$, 为所有用户在资源 r 上的最大占优资源.

在进行任务调度时,权重越高的用户任务得到越优先的调度.由于云计算采用按需分配服务计价模式,根据用户请求的资源数量(用虚拟机将多维资源打包出售)和服务时间长度的实际情况进行收费.因此,DRW 越高,说明用户单个任务请求的资源量越大,故而服务费用也将越高;从服务提供商的角度考虑,DRW 越高,为其贡献的收益也越多,因此,其得到优先服务也正好吻合云计算的商业运营模式.

此外,权重的定义为具有相同占优资源的任务提供了量化指标,保证了任务分配的有序性,尤其是具有相同占优资源的任务分配有序性.有时两个或多个任务具有相同的占优资源,但是,其次要占优资源可能差别较大,这个时候再依据占优资源确定优先顺序就有失公平性.

定义 3. 公平性(fairness). 每个用户的需求得到满足或等比例满足即是公平的.

资源分配的公平性应该根据请求的实际情况而定,并不一定是必须调度相同的任务数或者获得相同比例的资源.我们强调的是等效的公平分配,由于用户的请求不同(任务数可能不同,每个任务需要的资源也可能不同),主要体现在占优资源上,因此用户获得相同比例的任意一种资源都是等效的公平,比如 10% 的 CPU 与 10% 的 memory 或者 10% 的 disk 具有等效的公平性,或者说 2 个用户都有相同比例的任务得到的服务也是公平的.此外,用户请求的资源数量是依自己情况而定(不同的占优资源、不同任务数等),只要需求得到了满足我们也认为是公平的,这也体现了云应用的按需分配(on-demand)特性.

系统的资源配置机制不但影响系统的资源利用率,也影响用户对分配资源的满意程度.在本文中,主要采用 3 种分配机制(allocation mechanism):

- 1) 权重越大的用户任务越优先得到资源分配;
- 2) 为了担保用户任务调度的公平性,我采用最大最小公平性(Max-Min Fairness)分配的思想,在每一轮资源分配过程中,已经获取资源最少的用户优先得到调度;
- 3) 用户的任务总是优先调度到 DRE 最高(占优资源最适应)的服务器上,也即在调度任务时必须满足如下公式:

$$\mathbf{A}_{il} = \arg \max_{l \in S} E_{il}(\mathbf{A}_{il}). \tag{8}$$

定义 4. 资源有效利用原则(non-wasteful rule). 对于用户 i 和服务器 l ,如果分配 $\mathbf{A}_{il}$ 是 non-

wasteful 的,那么,减少任意一种资源的数量都将减少调度任务的数量.例如,对于所有的 $\mathbf{A}'_{il} < \mathbf{A}_{il}$ (也即满足 $E_i(\mathbf{A}'_{il}) < E_i(\mathbf{A}_{il})$),我们都有:

$$N_{il}(\mathbf{A}'_{il}) < N_{il}(\mathbf{A}_{il}).$$

对于服务器 l 而言,调度到其上的任务 DRE 越高,说明按此份额分配时各种资源都越接近瓶颈资源,用户任务资源请求量与服务器资源越吻合,因此资源浪费也最少,可加载的任务也越多.同时,对于任意一个任务在调度时都要满足 $\mathbf{A}_{il} = \arg \max_{l \in S} \lambda(\mathbf{A}_{il})$,否则任务将不会调度到该服务器.

通常情况下,一个好的分配机制应该包含如下基本属性,其中,无论是在云计算系统^[1-2,26]还是经济学^[30-31]中,公平性和资源有效可度量被广泛认为是最重要的属性.

1) 无嫉妒性(envy-freeness). 如果没有用户相对于自己的分配份额而更喜欢别人的分配,那么这种分配机制是无嫉妒性的.例如,对于任意 2 个用户 $i, j \in U$,都有 $N_i(\mathbf{A}_i) \geq N_i(\mathbf{A}_j)$. 这是保障公平性的重要因素.

2) 帕累托最优(Pareto efficiency). 在一次分配中,如果返回的分配方案 $\mathbf{A}$ 使得其相对于所有可行的分配方案 $\mathbf{A}'$ 而言,如果对于用户 i 有 $N_i(\mathbf{A}_i) < N_i(\mathbf{A}'_i)$ 成立,那么将存在一个用户 j 使得 $N_j(\mathbf{A}_j) > N_j(\mathbf{A}'_j)$ 成立.此属性担保了分配方案的有效性以及资源的高利用率.

3) 激励共享(sharing incentive). 每个用户应该更好地共享云资源池中的资源,而不是独占地利用他自己分到的资源份额或者独自享用资源池资源.一方面,随着任务的完成应将资源释放到资源池中;另一方面,多个用户根据自己每个任务需要的资源份额更好地在一个服务器或者多个服务器上共同利用资源.

4) 防止策略性操纵(strategy-proofness). 用户不能通过撒谎以获取更多的资源,此属性提供激励兼容.

除此之外,还有一些重要属性:瓶颈公平(bottleneck fairness)、任务单调(task monotonicity)、资源单调(resource monotonicity)等^[1,30,33-34].

任务单调与资源单调是指用户的离开、任务的完成及系统资源的增加都不会产生现有的方案使用户减少或者获得资源的状况^[1].

2.4 多资源联合公平分配问题定义

2.3 节已经对多资源联合公平分配机制及其相

关属性作了比较详细的阐述. 我们基于这些属性, 形式化在异构云计算环境下最大化全局占优资源的 MDRF 问题.

设 ρ_{il} 是用户 i 在服务器 l 上获得的占优资源份额, 则有:

$$\rho_{il} = d_{r_{il}^*} \times N_i(\mathbf{A}_{il}) = d_{r_{il}^*} \times (A_{r_{il}^{\#}}/d_{r_{il}^{\#}}) = d_{r_{il}^*} \times \min_{r \in R} \{A_{ilr}/d_{ir}\}, \quad (9)$$

其中, r_{il}^* , $r_{il}^{\#}$ 分别为占优资源和瓶颈资源. 用户 i 在整个云计算中心获得的占优资源份额可计算为

$$G_i(\mathbf{A}_i) = \sum_{l \in S} \rho_{il} = \sum_{l \in S} (d_{r_{il}^*} \times \min_{r \in R} \{A_{ilr}/d_{ir}\}). \quad (10)$$

因此, 我们的 MDRF 分配方案可形式化为

object

$$\max_{\mathbf{A}} \sum_{l \in S} (d_{r_{il}^*} \times \min_{r \in R} \{A_{ilr}/d_{ir}\}), \quad (11)$$

s. t.

$$\sum_{i \in U} A_{ilr} \leq c_{ilr}, \forall l \in S, r \in R, \quad (12)$$

$$\sum_{l \in S} N_{il} \leq \mathbf{D}_i(N_i), \forall i \in U. \quad (13)$$

目标是使每个用户都获取最大化的全局占优资源, 第 1 个约束条件为每个服务器上调度的任务资源需求不能超过服务器供给的资源容量; 第 2 个约束条件为所有服务器上调度的任务数不超过用户请求的总任务数. 但是, 这一目标在分配过程中操作性不够强, 分配过程中对资源的量度也过于粗糙, 因此, 源于我们的占优资源熵定义, 可将上述问题等价于:

object

$$\max_{\mathbf{A}} \sum_{i \in U, l \in S} E_{il}(\mathbf{A}_{il}), \quad (14)$$

s. t.

$$\sum_{i \in U} A_{ilr} \leq c_{ilr}, \forall l \in S, r \in R, \quad (15)$$

$$\sum_{l \in S} N_{il} \leq \mathbf{D}_i(N_i), \forall i \in U. \quad (16)$$

目标是最大化全局占优资源熵, 约束条件同上.

很显然, 当每个服务器上都至少有一种资源被耗尽时, 我们的 non-wasteful 分配方案使得占优资源熵值达到最大, 资源利用率也最高, 最终系统输出的最优解矩阵为 $\mathbf{A}$.

3 多资源联合公平分配异构云计算体系结构

3.1 多资源联合公平分配体系结构设计

本文所设计的体系结构如图 3 所示, 资源调度中心由多个部件组成: 任务管理器(task manager)、任务调度器(task scheduler)、策略管理器(strategy manager)等. 在系统中, 前端代理服务器负责用户请求的收集、任务的分析等, 并将用户请求的任务上传给自动控制中心的任务管理器. 任务管理器负责用户任务请求及响应用户请求任务的管理; 任务调度器负责响应用户请求任务的虚拟机调度(包括虚拟机的部署和迁移); 策略管理器协助任务调度器进行策略选择等.

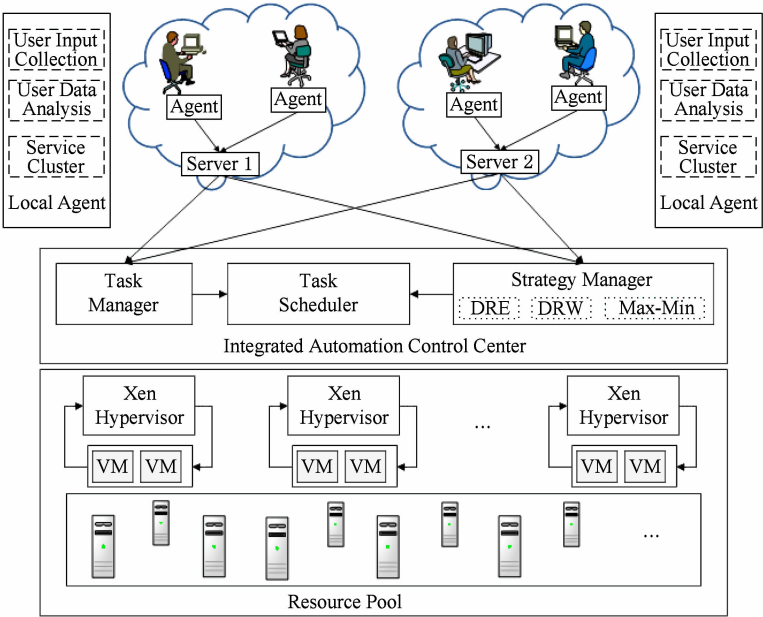


Fig. 3 A multi-resource fair scheduling architecture in heterogeneous cloud.

图 3 异构云中多资源联合公平调度体系结构

在我们的系统中,每个服务器运行 Xen Hypervisor (VMM),包含一个特权域 Domain 0(包括 Domain 管理和控制工具)和其他域 Domain U(包括 Domain U PV 客户系统和 Domain U HVM 客户系统);每个 VM 位于 Domain U,封装一个或多个应用,如 Web Server, Remote Desktop, DNS, Mail, Map/Reduce 等. 资源池由异构的服务器资源组成,资源种类包括 CPU, Memory, Disk, Network 等,当服务器上仍有未利用完的资源时(即仍然可以加载新的虚拟机时),服务器提供用来创建虚拟化资源的基础设施以满足用户任务的需求. 多个虚拟机可以根据其相应的资源需求在同一个物理服务器上动态地启动或停止,因此,在同一个服务器上提供了动态配置各种各样的资源划分的便利性以满足用户不同的任务请求. 在同一服务器上,多个虚拟机可以根据不同的需求基于不同的操作系统环境并发地运行不同的应用. 任务管理器根据服务器不同的运行状态决定是否进行虚拟机的迁移以改善服务性能.

在我们的系统中,采用虚拟机技术,将资源池中的各种异构资源实现两层资源映射,即:用户请求到虚拟机的映射和虚拟机到资源池物理资源的映射. 用户请求由具有不同资源配置的虚拟机进行响应. 资源调度也即任务调度或者虚拟机调度采用时间片段(time-slot)模型,将一天(24 h)划分为不同的时间片段(例如 5 min, 10 min, 20 min 等),在每个时间片段开始之初对用户请求的各种虚拟机按照系统提供的公平调度策略将虚拟机调度到最合适的服务器上. 因此,系统设计的重点是 3.2 节提出的 MDRF 算法.

3.2 MDRF 分配算法

渐进填充(progressive filling)^[1-2,25]是一种实现 Max-Min Fairness 的理想算法,它最早被应用在在网络场景中的资源分配. 我们将这一思想引入我们的算法以解决异构云计算环境中的基于占优资源的多资源联合公平分配问题,基于我们定义的占优资源熵,每次都将在任务调度到各种资源分配比例最合适的服务器上,直到每个用户的全部任务都分配完毕或者每台服务器都至少有一种资源被耗尽不能再调度任务.

在本节中,我们介绍求解 2.4 节所定义的带约束条件的最优化问题. 为此,我们基于 3.1 节所设计的异构云计算体系结构提出了基于启发式算法的 MDRF 多资源联合公平分配算法,其主要思想为:

1)根据用户请求任务的 DRW 将用户任务降序排列;2)在每一轮资源分配过程中挑选一个任务,DRW 越大的用户任务得到优先调度;3)为了保障公平性,使每个用户的任务都能得到调度,我们采用了 Max-Min Fairness 思想,在一轮分配中还要保障当前获取资源最少的用户优先得到任务调度,当出现多个用户获取资源同时最少时 DRW 大的在本轮中优先调度;4)用户任务被调度时,任务应该调度到占优资源最适合的服务器上,即 DRE 最大的服务器上. MDRF 算法的操作过程如算法 1 描述:

算法 1. MDRF 算法

```
输入:用户需求  $d_i, i \in U$ ; 服务器集群  $S$ ;  
输出:任务分配矩阵  $\mathbf{A}$ .  
① 按式(7)计算每个用户的占优资源权重  $\omega_i$ ;  
/* 降序排列用户权重 */  
② sort, DecreaseUserWeight( $\omega_i$ );  
③ 挑出权重最大的用户  $i$ ;  
/* 调用 MDRF_fittest() 函数将用户  $i$  调度到最适合的服务器上 */  
④ call MDRF_fittest( $i, S$ );  
⑤ for all user  $i \in U$   
⑥ if  $N_1 \parallel N_2 \parallel \cdots \parallel N_i \neq 0$  then  
/* 还有用户任务未分配完,从未分配的任务中挑出分配资源或者任务数最少但权重最大的进行分配,  $\parallel$  表示“或运算” */  
⑦ 从任务未分配完的用户中挑选出已分配任务数最少并且具有最大占优资源权重的用户  $i'$ ;  
⑧ call MDRF_fittest( $i', S$ );  
⑨ endif  
⑩ endfor  
/* 函数 MDRF 将用户  $i$  调度到最适合的服务器上 */  
⑪ MDRF_fittest( $i, S$ );  
⑫ for each server  $l \in S$   
⑬ 计算 user  $i$  相对于 server  $l$  的占优资源熵  $E(\mathbf{A}_{il})$ ,根据式(5)~(6)找到满足式(8)的 server;  
⑭ 修改资源分配矩阵  $\mathbf{A}$  中对应用户  $i$  的向量  $\mathbf{A}_i = (\mathbf{A}_{i1}, \mathbf{A}_{i2}, \cdots, \mathbf{A}_{im})$ ;  
⑮ 修改用户的资源请求向量  $\mathbf{D}_i$ ;  
⑯ end for  
⑰ return 资源分配方案  $\mathbf{A}$ .
```


在 MDRF 中,任务是调度的基本对象,在每一轮分配过程中,MDRF 都采用最佳适应算法,寻求任务可调度的最佳目标服务器(best-fit MDRF),用最少的全局占优资源来响应用户任务资源需求(投入同样的资源可以响应更多的任务),同时考虑全局占优资源的状态,保障任务分配到使全局资源残片最小的服务器上.采用我们提供的分配算法,图 1 中给出的请求例子将产生如图 4 所示的资源分配方案及资源利用率.

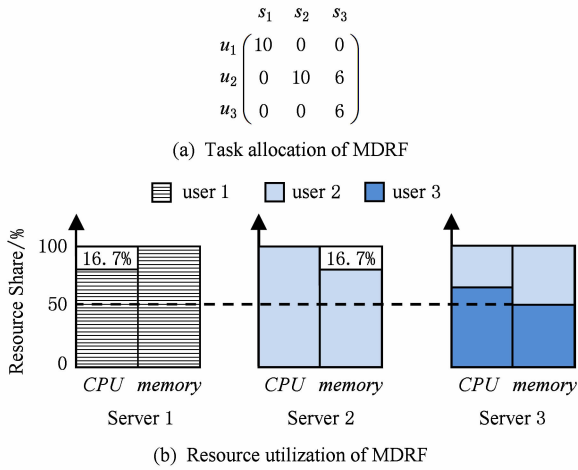


Fig. 4 MDRF allocation.

图 4 MDRF 分配

无论相对于 DRFH 还是 DRF,算法 MDRF 都有着更高的系统资源利用率(server 3 的占优资源利用率提高了 16.7%),响应总的用户任务的个数也更多,虽然算法对每个用户调度的任务数不同,但是系统中有大量的异构服务器存在,用户 1 和用户 3 少于用户 2 的用户任务由这 3 个服务器之外的其他服务器完成,否则我们的算法将自适应产生图 2 中的 DRFH 算法分配结果.

3.3 MDRF 基本属性分析

我们的算法满足下列属性,首先介绍最本质的 3 个基本属性:无嫉妒性、帕累托最优和可信性(truthfulness).

定理 1. 无嫉妒性:解问题式(14)的 MDRF 分配算法是无嫉妒性的(envy-free).

证明. 记 $G_i(\mathbf{A}_{il}), \forall l \in S$ 为用户 i 在解问题式(14)的输出解矩阵 $\mathbf{A}$ 中对应的解向量 $\mathbf{A}_i$ 中的对应分量.对所有用户 i 而言,他们在所有服务器上的 MDRF 分配都要满足式(8).下面,我们证明 $G_i(\mathbf{A}_j) \leq G_i(\mathbf{A}_i)$ 对于任意 2 个用户 i 和 j 都成立.

因为 MDRF 的分配原则即满足式(8) $\mathbf{A}_{il} =$

$\arg \max_{l \in S} E_{il}(\mathbf{A}_{il})$,所以在每一轮分配过程中都有:

$$E_i(\mathbf{A}_{il}) \geq E_i(\mathbf{A}_{jl}), \forall l \in S$$

成立,所以:

$$E_j(\mathbf{A}_{jl})/E_i(\mathbf{A}_{il}) \leq 1, \forall l \in S.$$

记 $\sigma_{il} = \max_{l \in S} E_{il}(\mathbf{A}_{il}), \forall i \in U$,为任意用户 i 使得熵值最大时在 server l 分得的资源比例;同理,对于用户 j 也有 $\sigma_{jl} = \max_{l \in S} E_{jl}(\mathbf{A}_{jl}), j \in U$. 因此,对于任意 2 个用户 i 和 j ,则有:

$$\begin{aligned} G_i(\mathbf{A}_j) &= \sum_l G_{il}(\mathbf{A}_{jl}) = \sum_l G_{il}(\max_{l \in S} E_{il}(\mathbf{A}_{jl})) = \\ &= \sum_l \max\{\sigma_{jl}(E_{jl}(\mathbf{A}_{jl})/E_{il}(\mathbf{A}_{il}))\} \leq \\ &= \sum_l \sigma_{jl} = \sum_l G_{il}(\mathbf{A}_{il}) = G_i(\mathbf{A}_i). \quad \text{证毕.} \end{aligned}$$

定理 2. 帕累托最优. 解问题式(14)的 MDRF 分配算法是帕累托最优的(Pareto efficient).

证明. 记 $\{\sigma_{il}\}$ 和相应的 σ_i 是问题式(14)的解用户 i 获得的资源比例, $\sigma_i = \sum_l \sigma_{il}$. 对于所有用户 i ,他的 MDRF 分配在 server l 上都有 $\sigma_{il} = \max_{l \in S} E_{il}(\mathbf{A}_{il})$, 因此, σ_i 是目标式(14)的极大值.

MDRF 分配算法是迭代执行的,在每一轮分配中用户增加自己的全局占优资源到可用的服务器上,直到所有的任务被调度(用户达到最满意状态),此后的调度过程中用户的该任务将不再被考虑. MDRF 再开始重复上面的分配过程,直到没有用户仍有可调度的任务或者所用的可用资源耗尽. 如果分配 $\mathbf{A}$ 使得每个用户的全部任务都得到满足,显然是帕累托最优的:没有用户更羡慕别人的分配.

下面我们证明分配 $\mathbf{A}$ 耗尽所有资源仍然没有分配完所有用户任务的情况.

反证法,假设分配 $\mathbf{A}$ 不是帕累托最优的,那么,将存在一些分配 $\mathbf{A}'$ 使得 $G_i(\mathbf{A}'_i) \geq G_i(\mathbf{A}_i)$ 对于所有用户成立,并且,对于一些用户 j 有严格的不等式 $G_j(\mathbf{A}'_j) > G_j(\mathbf{A}_j)$ 成立. 不失一般性, $\mathbf{A}'$ 也是 non-wasteful 的. 对于用户 i 和 server l ,存在一些 $\{\sigma'_{il}\}$ 和 $\sigma'_i, \sigma'_i = \sum_l \sigma'_{il}$, 有 $\sigma'_i = \max_{l \in S} E_{il}(\mathbf{A}'_{il})$ 成立.

MDRF 在每一轮分配中,都严格遵照 Max-Min Fairness 原则以保障资源共享的公平性,即获得资源最少的用户按照自己的占优权重获得优先分配的次序,并且将其任务调度到使得占优资源熵 DRE 最大的服务器上,即 $\mathbf{A}_{il} = \arg \max_{l \in S} E_{il}(\mathbf{A}_{il})$. 对于任意一个服务器 l ,其资源耗尽时都有 $\sum_{i \in U} E_{il}(\mathbf{A}_{il})$ 达到

极值成立,此时,记分配 $\mathbf{A}$ 使得服务器 l 上的空闲资源向量为 $\boldsymbol{\varepsilon} = (\varepsilon_{l1}, \varepsilon_{l2}, \dots, \varepsilon_{lk})$, 分配 $\mathbf{A}'$ 使得服务器 l 上的空闲资源向量为 $\boldsymbol{\varepsilon}' = (\varepsilon'_{l1}, \varepsilon'_{l2}, \dots, \varepsilon'_{lk})$. 由于 $\mathbf{A}'$ 也是 non-wasteful 的, 所以 $\mathbf{A}'_{il} = \arg \max_{i \in S} E_{il}(\mathbf{A}'_{il})$. 对于任意服务器 l , 记 $\mathbf{o} = (o_{l1}, o_{l2}, \dots, o_{lk})$ 为零空闲资源向量, 即在分配结束时服务器 l 的全部资源都耗尽(如图 4 中 server 3 的空闲资源向量即为 $\mathbf{o}$ 向量). 根据占优资源熵的定义, 熵值越大, 服务器各类型资源越接近瓶颈资源, 因此最终的空闲资源也就越少, 那么有:

$H'(\boldsymbol{\varepsilon}', \mathbf{o}) = \|\boldsymbol{\varepsilon}' - \mathbf{o}\|_1 \geq \|\boldsymbol{\varepsilon} - \mathbf{o}\|_1 = H(\boldsymbol{\varepsilon}, \mathbf{o})$, 这里 $\|\cdot\|_1$ 是 L^1 范数, 用来衡量 2 个向量之间的距离. H 越小说明 2 个向量的距离越近. 故而, 对于分配 $\mathbf{A}'$, 有:

$$\sum_{l \in S} H'(\boldsymbol{\varepsilon}'_l, \mathbf{o}) \leq \sum_{l \in S} H(\boldsymbol{\varepsilon}_l, \mathbf{o}),$$

进而可以得出:

$$\sum_{i \in U, l \in S} G_i(\mathbf{A}') \geq \sum_{i \in U, l \in S} G_i(\mathbf{A}).$$

因为总体空闲资源越少, 系统的资源利用率也就越高, 最终全部用户获取的总资源量也越多. 由于我们的分配原则严格遵照 Max-Min Fairness, 所以, 对于任意用户 i 和 j ($i \neq j$), 都有 $\sigma'_i = \sum_l \sigma'_{il} \geq \sigma_i = \sum_l \sigma_{il}$, 并且 $\sigma'_j = \sum_l \sigma'_{jl} \geq \sigma_j = \sum_l \sigma_{jl}$ 成立, 然而这与我们最初的解目标式(14)中的解 σ_i ($\forall i \in U$) 是最优相矛盾. 因此, MDRF 分配满足帕累托最优特性. 证毕.

定理 3. 可信性: 解问题式(14)的 MDRF 分配算法是可信的(truthful).

证明. 在我们的云系统中, 虚拟机是以出租方式按需分配的收费服务, 用户撒谎提供更大的资源请求, 将会付出更多的预算成本, 从而导致自己更大的经济损失, 因此用户是不会说谎的, 即 truthful 的. 证毕.

其他重要属性如下:

定理 4. MDRF 满足激励共享和瓶颈公平属性.

证明. 考虑一个系统有 n 个用户请求的情况, 假定资源 k 是采用我们的基于占优资源熵的渐进填充算法第一种被耗尽的资源, 记用户 i 获得资源 k 的最大占优份额为 $g_{i,k}$, 由于资源 k 已经被耗尽, 因此, 易得 $g_{i,k} \geq 1/n$. 进一步来讲, 由于我们占优资源熵的定义及渐进填充的特性, 用户的任务总是优先调度到占优资源更匹配的服务器上, 因此, 用户获得

的占优资源 $g_{i,r^*} \geq g_{i,r^\#} = g_{i,k} \geq 1/n$. 用户 i 至少获得 $1/n$ 的占优资源份额. 所以, MDRF 满足激励共享属性. 如果所有用户有相同的占优资源, 那么每个用户都将获得 $1/n$ 的占优资源份额, 故 MDRF 也同样满足瓶颈公平属性. 证毕.

定理 5. MDRF 满足任务单调及资源单调属性.

证明. 在我们的方案中, 资源分配是以虚拟机为单位(包含完成任务请求的各种资源)的, 由虚拟机的特性, 即虚拟机部署以后任务在完成过程中将以资源隔离的形态被用户独占, 因此, 在任务得到调度时获得资源, 在用户离开时虚拟机将资源释放到资源池, 所有这些操作都不会影响到其他用户, 故而满足任务单调性. 同时, 当系统的资源增加时资源进入资源池, 等待以虚拟机的形式打包调度, 并不会影响到其他用户正在执行任务的虚拟机(资源独占并且隔离特性), 所以 MDRF 也满足资源单调属性. 证毕.

3.4 时间复杂度分析

假定系统在某一时刻共有 n 个用户向系统提交任务请求, 每个用户请求的任务数为 t_i ($i \in U$), 则用户请求的任务总数:

$$T = \sum_{i \in U} t_i,$$

此时, 系统中可提供资源分配的服务器(该服务器的所有资源均未被耗尽)总数为 m . 根据我们所提出的资源分配算法即: 每一次都根据占优资源权重(DRW)从获取资源最少的用户中挑选出一个任务, 然后根据占优资源熵将该任务分配到最适合的服务器(即使该服务器产生的资源碎片最少)上. 所以 MDRF 算法的时间复杂度为 $O(mT)$. 在数据中心的规模比较大且用户请求的任务数量比较多的情况下, 可以将计算任务分派给多个服务器计算, 如果分派给 k 个相同的服务器计算时, 只需要做一个简单的服务器资源使用状态同步, 算法的时间复杂度就可以降低为 $O(mT/k)$. 因此, 我们提出的算法完全可以适应大规模云计算数据中心.

4 实验评估

实验建立: 我们基于 Google 的集群使用情况数据集^[35-36]评估了 MDRF 的性能, 表 1 为 Google 的异构服务器集群及其组成情况, CPU 和 memory 都是基于最大处理能力的服务器容量正则化处理后的值. 我们基于 900 个用户对 12 000 台服务器的资源

需求和使用本文提出的算法进行仿真. 每一个用户提交的请求都可以进行任务的划分, 并且请求一定数量的资源去完成. 我们根据计算需求信息——任务请求的资源数量和任务运行时间作为我们的分配算法评估的需求输入.

Table 1 A normalized Configuration of Server in Google Cluster

Cluster		
表 1 一个 Google 集群正则化处理后的 Server 配置		
Number of Servers	CPU	memory
6 732	0.50	0.50
3 863	0.50	0.25
1 001	0.50	0.75
795	1.00	1.00
126	0.25	0.25
52	0.50	0.12
5	0.50	0.03
5	0.50	0.97
3	1.00	0.50
1	0.50	0.06

4.1 资源动态分配

在用户动态加入或者离开系统时, 我们采用最佳适应的 MDRF(best-fit MDRF) 分配算法, 保障用户任务及资源分配的公平性. 首先, 我们在一个包含了 60 个服务器的异构集群中仿真了 3 个用户提交不同资源需求的任务, 服务器集群的配置是根据表 1 中 Google 的服务器集群分布随机生成的, 形成了一个总共包含 40.75 CPU 和 38.34 memory 的共享资源池. 用户 1 在一开始就加入系统, 提交的每个任务的资源需求为 0.2 CPU 和 0.3 memory. 如图 5 所示, 由于一开始只有用户 1 提交任务资源请求, 所以用户 1 被分配到 41% 的 CPU 和 61% 的 memory, 这一分配一直持续到第 200 s, 此时用户 2 加入系统, 并且提交的每个任务都有一个 0.5 CPU 和 0.1 memory 的资源需求. 由于用户 2 的加入, 两者开始竞争计算资源, 此时 MDRF 算法开始发挥作用, 在这种情况下, 用户 1 获得了一个 45% 的内存资源共享, 用户 2 获得了 46% 的 CPU 资源共享. 在第 500 s, 用户 3 加入系统, 开始提交内存密集型的任务, 每个

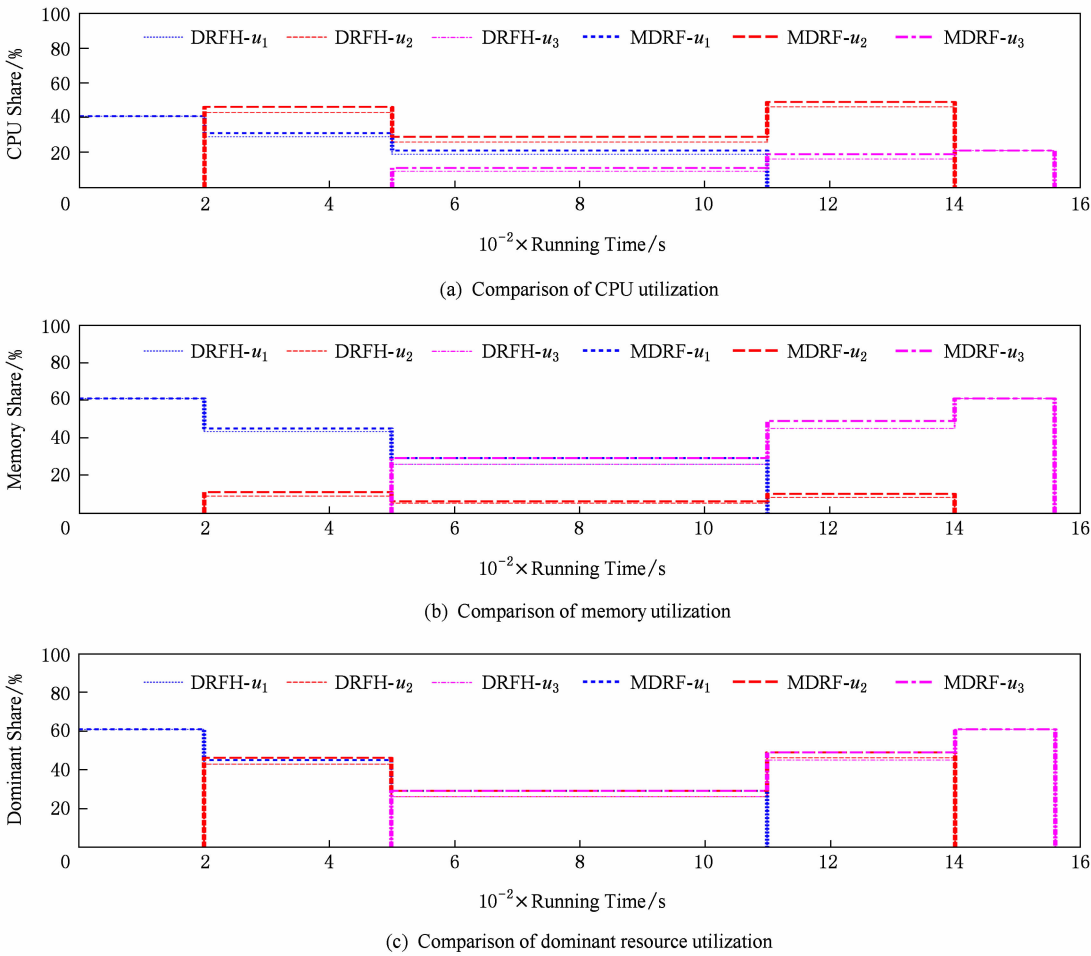


Fig. 5 CPU, memory and global dominant sharing for three users under MDRF and DRFH strategies.

图 5 MDRF 和 DRFH 策略下的 CPU, memory 以及全局占优资源分配

任务需要 0.1 CPU 和 0.3 *memory* 的资源,此时,MDRF 算法为 3 个用户都分配了 29% 的全局占优资源.直到第 1100 s,用户 1 离开系统.此后,用户 2 与用户 3 共享系统资源,分别获得 49% 的全局占优资源.由图 5 可以很明显地看出,与 DRFH 相比,MDRF 具有较明显的占优资源共享率的提升约 3%,这里的用户较少,VM 的种类较少;然而,在实际的大规模环境中,用户的资源需求种类是多样的,故而虚拟机请求的种类也是多样的,所以资源利用率还将有更大的提高.从图 5 也可以容易地看出,用户的资源种类需求越多,请求的种类越互补(如 CPU 密集型的服务与内存密集型的服务均可以看作资源请求互补型的任务),MDRF 算法的效果就越明显,这是因为我们的算法更大程度上减少了资源分配过程中产生的资源碎片,从而提高了系统的整体资源利用率.

4.2 资源利用率

我们评估采用 MDRF 算法的系统资源利用率.我们采样了一个来自 Google 的 24 h 的计算需求数据,然后在一个由 1 500 个服务器节点组成的小型异构云计算系统上进行仿真,每个服务器的资源配置是根据表 1 中的服务器分布随机抽取的.我们比较了 MDRF 与其他算法比如 Best-Fit DRFH^[2] 和 Slots^[22] 的性能差别.

图 6 显示了 3 种算法的 CPU 和 Memory 利用率的时间序列,从图 6 可以很清晰地看出,我们提出的算法在资源利用率方面有显著提高,这主要是因为我们的算法充分考虑并且精确量化响应任务的虚拟机资源需求与服务器资源供给间的各种资源差异.通过仿真实验,我们的算法在同构云计算环境下一样具有同样的效率.

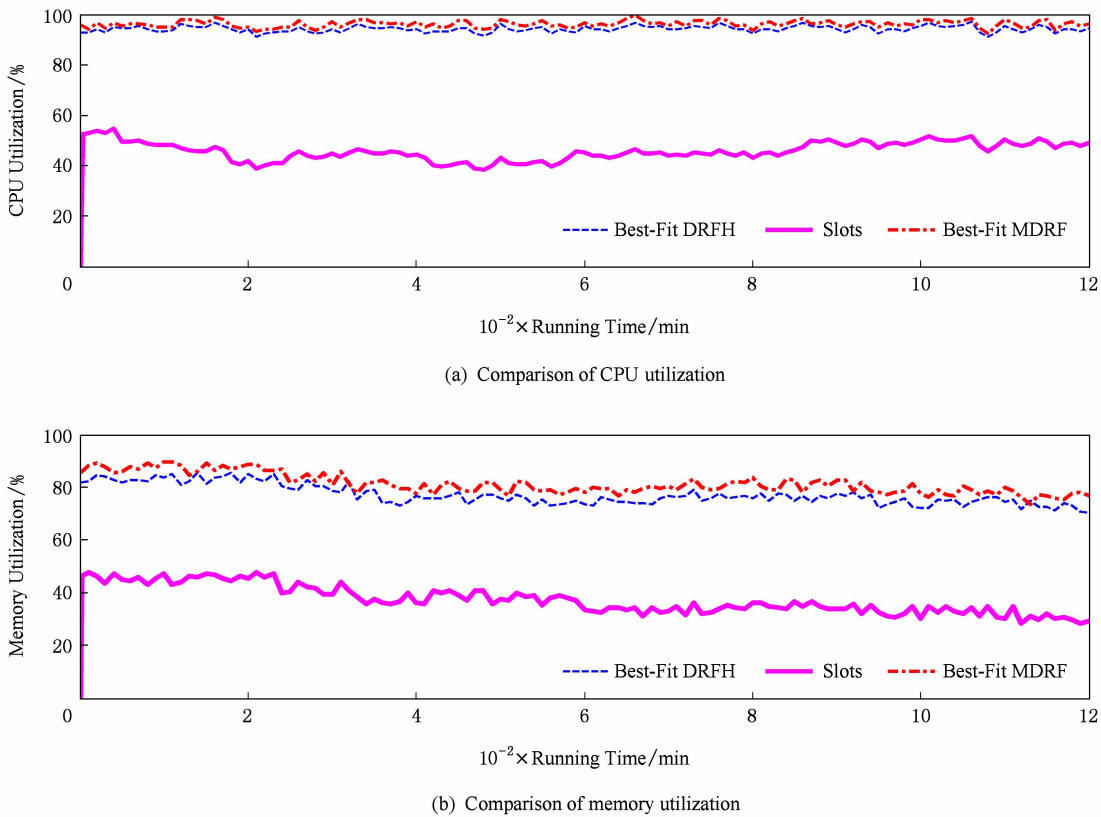


Fig. 6 CPU and memory utilization.

图 6 CPU 和内存的资源利用

高的资源利用率是因为各种资源碎片的减少.为了进一步验证我们所提出算法的资源效率提升,我们进一步增加资源分配的维度,将磁盘容量和网络带宽添加进来.我们通过类比采样,产生更多样化的资源请求类型,用户资源请求模型参照亚马逊

EC2 虚拟机类型,请求的资源向量为

$\mathbf{D} = (\text{CPU}, \text{memory}, \text{disk}, \text{network bandwidth})^T$, 同时,我们对服务器的资源参数进行相应地调整.重复与上面类似的实验,观察算法的效率.图 7 显示了增加资源维度后各类资源利用率有进一步的提升.

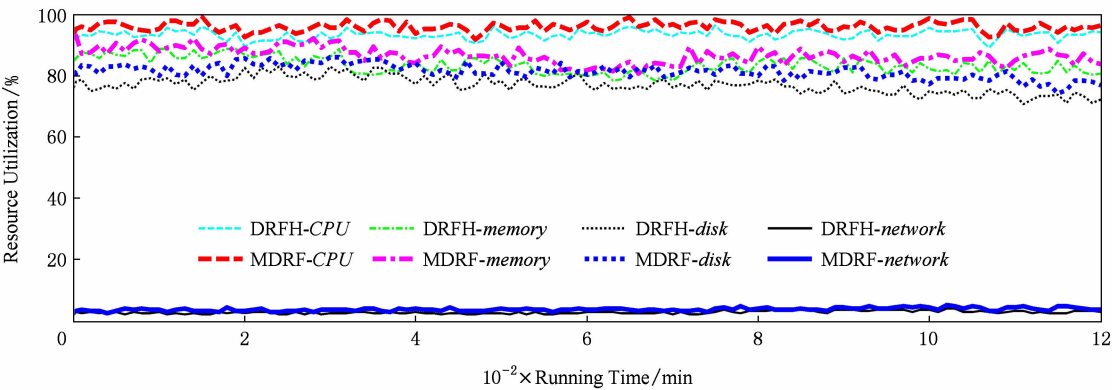


Fig. 7 Comparison of resource utilization for multiple request types.

图7 多请求类型下的资源利用率比较

此外,高的资源利用率是由于用户任务完成时间的减少.记 $p(t)$ 为任务完成时间为 t 的任务完成概率,记 $F(t)=P(p(t)=1)$ 为任务完成时间为 t 的用户的任务都完成的累积分布函数.图8显示了2种算法任务完成时间的累积分布.从图8(a)中可以很容易看出,我们的算法有更好的效果;而图8(b)更加详细地展示了MDRF相对于DRFH在不同种类任务情况下完成时间的提升.实验中,我们根

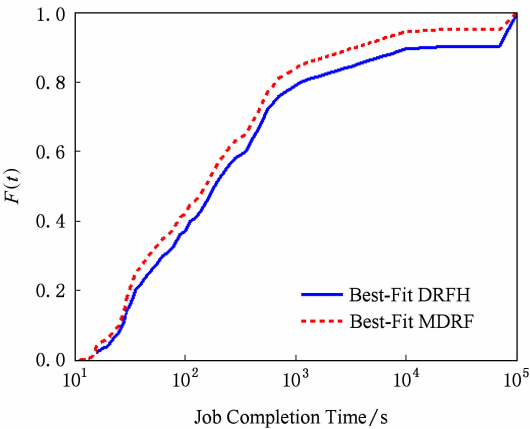
据任务数的多少将任务分为5种,对于每一种任务我们计算平均完成时间的减少量.从实验结果看,对于小任务,任务完成时间提升并不明显;但是,对于包含更多任务的大任务而言,提升比较明显.从一般意义上讲,用户的任务量大时,反而期望完成时间得到更大的改善.

5 结束语

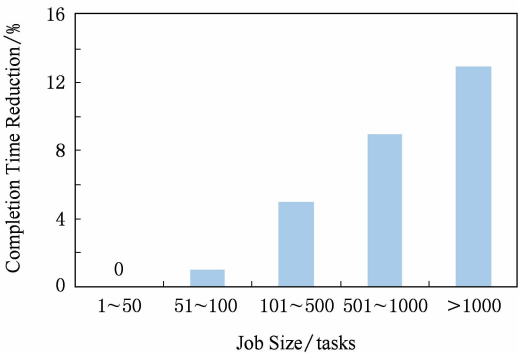
本文中,在总体资源量有限的前提下,尤其是资源组成多样化的异构云计算系统中,为了满足用户不同的任务需求,达到多种资源分配的公平性,我们设计了一个虚拟化的异构云计算体系结构,并提出了该体系结构下基于占优资源的多资源联合公平分配算法MDRF,同时,给出了占优资源熵和占优资源权重的定义.占优资源熵用来精确地刻画用户任务各资源的请求量与任务被调度到服务器各资源供给的适应程度,增强了系统的自适应能力,提高了整体资源的利用率;占优资源权重保障了不同用户优先获取资源的次序,协同所用的保障资源分配公平性的Max-Min Fairness策略,使资源分配更加有序.此外,还证明了算法的帕累托最优等相关属性.实验表明,我们提出的策略具有更高的系统资源利用率并且使需求与供给更加匹配,进而在总体资源不变的情况下使用户的占优资源获取更多,提高了服务质量.

参 考 文 献

[1] Ghodsi M Z A, Hindman B A, Konwinski, et al. Dominant resource fairness: Fair allocation of multiple resource types [C] //Proc of FAST'11. Berkeley, CA: USENIX Association, 2011: 1-14



(a) CDF of job completion time



(b) Job completion time reduction

Fig. 8 MDRF improvements on job completion time.

图8 任务完成时间的提升

- [2] Wei W, Baochun L, Ben L. Dominant resource fairness in cloud computing systems with heterogeneous servers [C] // Proc of the 33rd IEEE INFOCOM. Piscataway, NJ: IEEE 2014; 583-591
- [3] Xiao Z, Song W, Chen Q. Dynamic resource allocation using virtual machines for cloud computing environment [J]. IEEE Trans on Parallel and Distributed Systems, 2013, 24(6): 1107-1117
- [4] Jinhai W, Chuanhe H, Kai H, et al. An energy-aware resource allocation heuristics for VM scheduling in cloud [C] //Proc of the 10th IEEE Int Conf on High Performance Computing and Communications & 2013 IEEE Int Conf on Embedded and Ubiquitous Computing (HPCC_EUC). Berlin: Springer, 2013; 587-594
- [5] Zuo Liyun, Cao Zhibo, Dong Shoubin. Virtual resource evaluation model based on entropy optimized and dynamic weighted in cloud computing [J]. Journal of Software, 2013, 24(8): 1937-1946 (in Chinese)
(左利云, 曹志波, 董守斌. 云计算虚拟资源的熵优化和动态加权评估模型[J]. 软件学报, 2013, 24(8): 1937-1946)
- [6] Fei X, Fangming L, Hai J, et al. Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions [J]. Proceedings of the IEEE, 2014, 102(1): 11-31
- [7] Sheng D, Wang C L. Dynamic optimization of multiattribute resource allocation in self-organizing clouds [J]. IEEE Trans on Parallel and Distributed Systems, 2013, 24(3): 464-478
- [8] Shi Xuelin, Xu Ke. Utility maximization model of virtual machine scheduling in cloud environment [J]. Chinese Journal of Computers, 2013, 36(2): 252-261 (in Chinese)
(师雪霖, 徐格. 云虚拟机资源分配的效用最大化模型[J]. 计算机学报, 2013, 36(2): 252-262)
- [9] Tang Zhuo, Zhu Min, Yang Li, et al. Random task-oriented user utility optimization model in the cloud environment [J]. Journal of Computer Research and Development, 2014, 51(5): 1120-1128 (in Chinese)
(唐卓, 朱敏, 杨黎, 等. 云环境中面向随机任务的用户效用优化模型[J]. 计算机研究与发展, 2014, 51(5): 1120-1128)
- [10] Baruah S K, Gehrke J E, Plaxton C G. Fast scheduling of periodic tasks on multiple resources [C] //Proc of the 9th Int Parallel Processing Symp. Piscataway, NJ: IEEE, 1995; 280-288
- [11] Mo J H, Walrand J. Fair end-to-end window-based congestion control [J]. IEEE/ACM Trans on Networking, 2000, 8(5): 556-567
- [12] Blanquer J M, Ozden B. Fair queuing for aggregated multiple links [C] //Proc of ACM SIGCOMM Computer Communication Review. New York: ACM, 2001; 189-197
- [13] Liu Y, Knightly E. Opportunistic fair scheduling over multiple wireless channels [C] //Proc of the 22nd Annual Joint Conf on the IEEE Computer and Communications Societies. Piscataway, NJ: IEEE, 2003; 1106-1115
- [14] Khan N, Martini M G, Staehle D, et al. Opportunistic proportional fair downlink scheduling for scalable video transmission over LTE systems [C] //Proc of the 78th Vehicular Technology Conf (VTC Fall). Piscataway, NJ: IEEE, 2013
- [15] Koksai C E, Kassab H, Balakrishnam H, et al. An analysis of short-term fairness in wireless media access protocols [C] //Proc of ACM SIGMETRICS. New York: ACM, 2000; 1-13
- [16] Bredel M, Fidler M. Understanding fairness and its impact on quality of service in IEEE 802.11 [C] //Proc of the 28th Conf on Computer Communications (IEEE INFOCOM 2009). Piscataway, NJ: IEEE, 2009; 1098-1106
- [17] Uchida M, Kurose J. An information-theoretic characterization of weighted α -proportional fairness in network resource allocation [J]. Information Sciences, 2011, 181(18): 4009-4023
- [18] Lan T, Kao D, Chiang M, et al. An axiomatic theory of fairness in network resource allocation [C] //Proc of IEEE INFOCOM 2010. Piscataway, NJ: IEEE, 2010
- [19] Pla A, Lopez B, Murillo J. Multi-dimensional fairness for auction-based resource allocation [J]. Knowledge-Based Systems, 2015, 73(6): 134-148
- [20] Pao W, Lou W, Chen Y, et al. Resource allocation for multiple input multiple output-orthogonal frequency division multiplexing-based space division multiple access systems [J]. Iet Communications, 2014, 8(18): 3424-3434
- [21] Hadoop. Hadoop capacity scheduler [EB/OL]. [2014-03-09]. http://hadoop.apache.org/docs/r0.20.2/capacity_scheduler.html
- [22] Hadoop. Hadoop fair scheduler [EB/OL]. [2014-03-09]. http://hadoop.apache.org/docs/r0.20.2/fair_scheduler.html
- [23] Isard M, Prabhakaran V, Currey J, et al. Quincy: Fair scheduling for distributed computing clusters [C] //Proc of ACM SOSP 2009. New York: ACM, 2009; 1-20
- [24] Isard M, Budiu M, Yu Y, et al. Dryad: Distributed data-parallel programs from sequential building blocks [C] //Proc of the 2nd European Conf on Computer Systems (EuroSys 2007). New York: ACM, 2007; 59-72
- [25] Ghodsi A, Zaharia M, Shenker S, et al. Choosy: Max-min fair sharing for datacenter jobs with constraints [C] //Proc of the 8th ACM European Conf on Computer Systems. New York: ACM, 2013; 365-378
- [26] Carlee J W, Sen S, Tian L, et al. Multi-resource allocation: Fairness-efficiency tradeoffs in a unifying framework [C] //Proc of 2012 IEEE INFOCOM. Piscataway, NJ: IEEE, 2012; 1206-1214

[27] Dolev D, Feitelson D G, Halpern J Y, et al. No justified complaints; On fair sharing of multiple resources [C] //Proc of the 3rd Conf on Innovations in Theoretical Computer Science (ITCS 2012). Piscataway, NJ: IEEE, 2012: 68-75

[28] Gutman A, Nisarr N. Fair allocation without trade [C] //Proc of the 11th Int Conf on Autonomous Agents and Multiagent Systems 2012; Innovative Applications Track (AAMAS 2012). Piscataway, NJ: IEEE, 2012: 816-823

[29] Parkes D C, Procaccia A D, Shah N. Beyond dominant resource fairness; Extensions, limitations, and indivisibilities [C] //Proc of the 13th ACM Conf on Electronic Commerce (EC'12). New York: ACM, 2012: 808-825

[30] Li J, Xue J. Egalitarian division under Leontief preferences [J]. Economic Theory, 2013, 54(3): 597-622

[31] Procaccia A D. Cake cutting; Not just child's play [C] //Proc of 2013 Communication ACM. New York: ACM, 2013: 78-87

[32] Barbanel J B, Brams S J. Two-person cake cutting; The optimal number of cuts [J]. Mathematical Intelligencer, 2014, 36(3): 23-35

[33] Varian H. Equity, envy, and efficiency [J]. Journal of Economic Theory, 1974, 9(1): 63-91

[34] Pazner E A, Schmeidler D. Egalitarian equivalent allocations: A new concept of economic equity [J]. Quarterly Journal of Economics, 1978, 92: 671-687

[35] Reiss C, Wilkes J, Hellerstein J L. Google Cluster-UsageTraces [EB/OL]. [2014-07-09]. <http://code.google.com/p/googleclusterdata>

[36] Reiss C, Tumanov A, Ganger G R, et al. Heterogeneity and dynamicity of clouds at scale; Google trace analysis [C] //Proc of the 3rd ACM Symp on Cloud Computing. New York: ACM, 2012



Wang Jinhai, born in 1982. PhD candidate in Wuhan University. Lecturer of Xinjiang Vocational University. His main research interests include cloud computing and high performance computing.



Huang Chuanhe, born in 1963. Professor and PhD supervisor in Wuhan University. Senior member of China Computer Federation. His main research interests include the mobile Internet, mobile Ad Hoc network, wireless sensor network, wireless mesh network, WDM network, the Internet of things, network security and parallel distribution processing.



Wang Jing, born in 1986. PhD candidate in Wuhan University. Her main research interests include network security (WJing@whu.edu.cn).



He Kai, born in 1987. PhD candidate in Wuhan University. His main research interests include network security(kaihe@whu.edu.cn).



Shi Jiaoli, born in 1979. PhD candidate in Wuhan University. Associate professor. Her main research interests include network security(shijiaoli@whu.edu.cn).



Chen Xi, born in 1988. PhD main in Wuhan University. His current research interests include wireless network MAC protocols(chenxi_cs@whu.edu.cn).