一种动态优先级排序的虚拟机I/O调度算法

郭松辉[1,2] 龚雪容[1] 王 炜[1,2] 李清宝[1,2] 孙 磊[1]

(解放军信息工程大学 郑州 450001)[1] (数学工程与先进计算国家重点实验室 郑州 450001)[2]

摘 要 I/O任务调度是影响I/O密集型虚拟机性能的重要因素。现有调度方法主要是针对虚拟机整机I/O带宽的优化,较少兼顾各虚拟域与全局性能,也无法满足域间差异化服务的要求。针对现有方法的不足,提出了一种动态优先级排序的虚拟机I/O调度算法DPS。该算法基于多属性决策理论,以离差最大化方法计算I/O任务的优先级评估属性权重,对I/O任务优先级进行综合评估;通过引入任务所在虚拟域价值,体现云计算环境下虚拟域重要性差异。在Xen系统中通过实验评测DPS调度虚拟化网卡的性能,结果表明,DPS能够有效提高指定域与全局的I/O任务截止期保证率、整机I/O带宽,并能为不同虚拟域的I/O应用提供差异化服务。

关键词 云计算,虚拟化,I/O调度,差异化服务,多属性决策

中图法分类号 TP316 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.003

I/O Scheduling Algorithm Based on Dynamic Prioritization in Virtual Machines

GUO Song-hui[1,2] GONG Xue-rong[1] WANG Wei[1,2] LI Qing-bao[1,2] SUN Lei[1]

(PLA Information Engineering University, Zhengzhou 450001, China)[1]

(State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China)[2]

Abstract I/O scheduling is a key factor for the performance of I/O-intensive virtual machine. The existing scheduling methods mainly focus on optimizing the global I/O bandwidth of virtual machine, and few taking the performance of each virtual domain and the entire system synchronously into account. According to the deficiencies of the existing methods, I/O scheduling algorithm DPS was proposed based on dynamic prioritization. The algorithm takes advantage of multi-attribute decision making theory to calculate attribute weights dynamically for assessing I/O task priority comprehensively, and effectively characterize virtual domain by its value. The performance of DPS is evaluated by scheduling virtualized NIC in Xen. The results show that DPS improves the deadline guarantee ratios of specified domain and the entire system, improves global I/O bandwidth, and it can provide differentiated services for each virtual domain on demand.

Keywords Cloud computing, Virtualization, I/O scheduling, Differentiated service, Multi-attribute decision making

1 引言

云计算依托虚拟化技术动态组织各种硬件资源,构建透明化可伸缩计算系统架构,实现对资源的按需访问,有效提高资源利用率,降低能源消耗[1,2]。在虚拟化时对资源的共享会引起性能上的相互干扰[3]。宿主在同一物理主机上的多个虚拟机(Virtual Machines, VMs)对CPU、内存、磁盘和网卡等硬件资源都会产生竞争,导致不可预期的响应延迟和性能降低。云计算环境下的大量应用,如文件存储、流媒体和科学运算[1)]等,需要对I/O资源进行频繁操作,属于I/O密集型应用[4]。I/O密集型应用由于数据类型、虚拟化策略配置和共享I/O资源的虚拟机数量等参数的影响,在性能上存在较大的差异。在云计算环境中,通常依据系统能力目标和应用需求来确定虚拟机的宿主物理机部署。一方面,当多个虚拟机同时对I/O资源发起访问时,资源竞争会导致各虚拟机性能不可预期[5];另一方面,由于云计算环境中应用的多样性,应用具有不同的I/O负载特征,对聚合在同一物理主机中的各虚拟机的响应时间、吞吐率等性能具有不同的要求。同时,对于I/O密集型的负载,大量CPU运行周期浪费在CPU自旋等待数据或空闲循环上,降低了系统整体性能和可扩展性。因此,通过对虚拟机I/O调度问题进行研究,确保应用获得合理、可控的I/O带宽资源,才能为云用户提供既能保障服务质量(Quality of Service, QoS)又能最小化资源成本的云计算服务。

传统的虚拟机调度器仅关注虚拟机之间CPU资源的公平共享,将I/O调度放在次要位置,导致虚拟机系统的I/O资源难以得到充分利用[6]。近来,围绕提高虚拟机I/O性能,借助硬件辅助、软件优化等措施取得了大量研究成果。Noorshams Q等人基于统计回归分析[3]实现了虚拟机的I/O自动性能建模,并基于随机Petri网[7]实现了对复杂I/O系统的建模,为虚拟机I/O性能优化提供了有效的分析方法。Muench D[8],

1) http://aws.amazon.com/cn/solutions/case-studies/dummi-studio/

到稿日期:2015-12-19 返修日期:2016-03-07 本文受国家自然科学基金(61072047),国防预研基金(910A26010306JB5201)资助。

郭松辉(1979—),男,博士生,主要研究方向为云计算、虚拟化、性能建模,E-mail:guo_song_hui@163.com;龚雪容(1975—),女,博士,讲师,主要研究方向为网络与服务计算、虚拟化;王 炜(1975—),男,博士,副教授,主要研究方向为云计算可靠性、虚拟化、系统结构;李清宝(1967—),男,博士,教授,博士生导师,主要研究方向为云计算、系统结构;孙 磊(1973—),男,博士,副研究员,硕士生导师,主要研究方向为云计算安全、虚拟化。

Peleg O[9]等人基于硬件的 I/O 虚拟化方法，借助内存映射 I/O(Memory-mapped I/O)、存储管理单元(Memory Mana-gement Unit，MMU)和输入输出管理单元(Input/Output Memory Management Unit，IOMMU)等硬件资源，实现了高性能的 I/O 虚拟化。Guan H B 等人[5]基于双目标优化的负载高敏感模型进行 I/O 调度，将 CPU 密集域的额外信用分享给 I/O 密集域，确保系统中 I/O 密集域和 CPU 密集域之间的公平调度，提高 I/O 总带宽并减少应用响应时间。

云计算中 I/O 密集型应用根据不同的应用目标，对延迟时间、响应速度和 I/O 带宽等要求各不相同[10]。例如，在线音乐服务 Spotify，借助于 Amazon 云服务[1)]，具有严格的流媒体传输时延要求。基于即时互动服务的应用，如在线游戏公司 bwin. party 的在线博彩业务，要求具有尽可能短的玩家请求响应时间[2)]，Windows Azure 对其用户的标准系统响应低至 2～3ms。而现在广泛兴起的基于云平台的分布式计算[11]，利用互联网上计算机的 CPU 闲置能力来解决大型计算问题，对网络数据交换的延迟时间、响应速度等则没有严格的要求。因此，根据虚拟机 I/O 应用的特征，按照任务属性对虚拟机 I/O 设备进行调度[12]，才能获得理想的性能。Tan H L 等人[13]研究了基于虚拟链路动态映射的 I/O 带宽动态分配方案，依据各个请求的数据传输流程，为每个虚拟机映射一条到 I/O 设备的独立逻辑数据通路，具有理想的 I/O 性能隔离效果，但该方案从各独立应用的需求出发，不能对全局资源进行统一协调与分配。Bourguiba M 等人[14]采用包聚合的方式，减少了 I/O 调用过程中的上下文切换、中断场景保存等开销，提高了系统整体吞吐率，但未引入应用属性、任务处理时间作为调度依据，在高负载应用等环境下会出现关键应用因超时而无法满足服务等级协议(Service Level Agreements，SLAs)的情况。Jain N[10]等人根据每个任务预期延迟、任务截止期和历史任务完成情况，在每次 I/O 带宽分配前进行优先级排队，能够有效提高应用在截止期前获得 I/O 资源的概率，但其未考虑任务本身时间的紧迫性、价值等固有属性和应用的 I/O 速率等统计特征。王永炎等人[15]通过综合考虑任务的两个特征参数建立优先级表，再结合拉格朗日插值法获得相应任务的唯一优先级作为调度依据，具有成功率高、调度速率快的优点，但难以扩展到更多属性任务的调度。

因云计算平台中应用的多样性，虚拟机系统产生的I/O任务具有多种特征，对 I/O 服务也相应具有多种性能要求。不考虑任务特征的绝对公平的 I/O 调度算法，会导致低 I/O 带宽和高时间延迟，因此，虚拟机 I/O 调度算法必须充分考虑任务的实时特征和系统整体性能要求。本文依据任务的价值与时间、虚拟域的价值等特征参数，提出了一种综合任务价值密度与时间紧迫性的动态优先级虚拟机 I/O 调度算法 DPS(Dynamic Priority Scheduling)，该算法使用多种指标多角度对虚拟机 I/O 调度进行分析，并引入多属性决策理论，将待处理任务的动态选取转化为多属性决策问题加以解决。由于多属性决策方法的结果受评估属性的重要性权重的影响很大，而现有的重要性权重多依赖专家的主观赋值，因此本文结合虚拟机 I/O 调度提出权重的客观赋值方法，通过分析影响虚拟机 I/O 任务优先级的指标，根据任务属性权重未知且属性值以实数形式给出的特点，基于离差最大化[16-18]方法得到评估属性的重要性权重，实现了任务优先级的客观分派，从而为虚拟机应用提供差异化服务，并有效提高虚拟机 I/O 性能，确保了云计算平台的 QoS。本文的主要贡献如下：

1) 提出了一种任务静态映射排队方法，在特权域中为各个虚拟机中的应用建立访问 I/O 资源的静态任务队列，支持 I/O 调度器按照一定的策略进行调度；

2) 提出了一种新的基于任务优先级的 I/O 动态调度算法 DPS，该算法能够根据任务价值密度、时间紧迫性和任务所在虚拟域价值等特征，为虚拟机 I/O 应用提供差异化服务；

3) 提出任务优先级分派模型 TPAM，使用多种指标从多角度对任务进行评价，通过引入多属性决策理论，将任务的遴选转化为多属性决策问题加以解决。

本文第 2 节首先对 Xen 的 I/O 虚拟化技术进行介绍；第 3 节详细阐述 DPS 的设计，包括任务模型和调度框架；第 4 节构建 DPS 的任务优先级分派模型，详细描述属性指标权重的计算方法；第 5 节通过模拟云计算环境中虚拟机 I/O 调度的场景，基于 Xen 平台对提出的方法进行验证；最后总结全文并提出未来工作。

2 Xen I/O 虚拟化技术

Xen[19]是一种在科研领域和云计算应用领域广泛使用的虚拟机监控器(Virtual Machine Monitors，VMMs)。本文基于 Xen 进行虚拟机 I/O 调度的相关研究，研究成果同样适用于其他虚拟机平台。Xen 同时支持全虚拟化和半虚拟化两种工作模式[20]，当采用半虚拟化模式时，虚拟域能够获得更高的 I/O 处理能力[21]。Xen 的 I/O 虚拟化结构如图 1 所示。

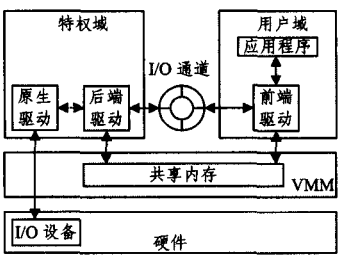

图 1 Xen I/O 虚拟化结构

Xen 的半虚拟化设备模型采用前后端分离的设备驱动结构，虚拟设备驱动包含 3 部分：用户域(User Domain，DomU)中的前端驱动(Front-end Driver)、能够访问真实 I/O 设备的特权域(Domain 0，Dom0)中的后端驱动(Back-end Driver)和原生驱动(Native Driver)。前端驱动为用户域提供虚拟设备，将用户域的 I/O 请求转发给后端驱动，后端驱动通过本地原生驱动控制硬件设备(I/O Device)完成 I/O 操作。在整个 I/O 设备访问的过程中，后端设备驱动是所有前端访问真实物理设备的代理。DomU 与 Dom0 通过 I/O 环(I/O Ring)传递 I/O 请求，通过共享内存(Shared Memory)传递 I/O 数据，通过事件通道(Event Channel)实现域间异步事件通知机制。

当用户域中的上层应用需要执行 I/O 发送任务时，前端驱动将请求信息写入 I/O 环，通过超级调用(Hypercall)通知后端驱动；后端驱动接收到 I/O 请求后，将请求提交给原生驱动。当硬件设备完成 I/O 请求后，后端驱动通知前端向用户域报告 I/O 操作完成。当执行 I/O 接收任务时，硬件设备在接收到数据包后，向 VMM 产生中断，由 VMM 通知特权域进行接收，采用与发送相反的流程完成接收操作。

1) http://aws. amazon. com/cn/solutions/case-studies/spotify/

2) http://www. microsoft. com/china/casestudies/details. aspx?CompanyProfileID=237

3 DPS 设计

3.1 虚拟机 I/O 任务模型

设虚拟机系统中用户应用产生的 I/O 任务集合为 $T=\{T_1,T_2,\cdots,T_n\}$，I/O 任务用五元组表示为 $T_i=\langle a_i,e_i,DL_i,v_i,c_i\rangle$，其中各参数表述如下：

1）a_i 表示任务的到达时间，即任务被启动并准备执行的时间；

2）e_i 表示任务的最坏情形执行时间，即任务在最坏情况下无中断执行所需的 I/O 处理时间，由虚拟机根据 I/O 设备性能和任务的数据量预估得到；

3）DL_i(DeadLine)表示任务 T_i 的截止期，即任务在该时刻应该完成执行并产生一个有价值的结果，由 I/O 应用随任务输入；

4）v_i 表示任务的价值，即任务的关键性，是该任务相对于其他任务的重要程度，本文约定该值越大，表示重要程度越高；

5）c_i 表示任务的完成时间。

3.2 相关定义与假设

定义 1(任务的评估属性，Attribute Assessment Task)

为便于给任务分派优先级，通过对任务特征参数进行处理，获取到对任务优先级具有重要影响的属性指标[1)]，以对 I/O 调度进行量化分析。这部分用于评估分析的典型属性即为任务的评估属性(在后面的讨论中，如果没有特别说明，任务的属性均指评估属性)。使用向量形式表示的任务属性称为任务的属性向量。

定义 2(任务 T_i 的空闲时间 ST_i，Spare Time)

$ST_i=DL_i-(e_i+a_i)$

ST_i 表示任务 T_i 的空闲时间长度，反映了任务的时间紧迫性，由任务的截止期 DL_i、最坏情形执行时间 e_i 和到达时间 a_i 决定，DL_i，e_i，a_i 具有相同的量纲及量纲单位。空闲时间越小，则紧迫性越高。

定义 3(任务 T_i 的价值密度 VD_i，Value Density)

$$VD_i=\frac{v_i}{e_i}$$

VD_i 表示任务 T_i 的预期价值与任务理论执行时间的比率，反映了任务的重要性，由任务的价值 v_i 和最坏情形执行时间 e_i 决定。式中 v_i 与 e_i 都是任务自身的属性，即价值密度只与任务自身的属性相关，与任务的执行过程无关。

定义 4(任务 T_i 所在虚拟域的价值 DO_i，DOmain value)

云计算平台按照 SLA 为用户提供差异化服务，因此虚拟机中承载不同类型应用的虚拟用户域的重要性各不相同。系统将虚拟用户域的重要性量化为价值 DO_i，通过 DO_i 数值体现重要性差别。若 T_i 在截止期 DL_i 前完成，则 T_i 对系统累积价值的贡献为 DO_i，否则为 0。

定义 5(截止期保证率 DGR，Deadline Guarantee Ratio)

$$DGR=100*\frac{\sum_i T_G}{\sum_i T_C}$$

DGR 表示所有虚拟域的 I/O 任务调度满足截止期的情况，它体现了 I/O 调度算法的健壮性。i 为虚拟域的序号，T_G 为虚拟域满足截止期的任务数，T_C 为虚拟域调用 I/O 的任务总数。

假设 1　不同域中 I/O 应用间相互独立，即除 I/O 资源外，I/O 应用无冲突资源及相互依赖关系。

假设 2　虚拟用户域可以在一个时间段内运行不同的 I/O应用，但在任意给定的时刻，每个虚拟用户域只运行一种 I/O 应用。

假设 3　I/O 应用的任务属性在执行期间保持不变，且任务不会自动挂起。如网络应用对数据传输速率和数据量等要求，在提取任务评估属性时属性量化值保持不变。

3.3 DPS 框架

DPS 框架的目标是为有时间约束的上层 I/O 应用提供优质的服务，以使设备性能满足云计算环境下虚拟机的 SLA 要求，同时为非关键应用提供可接受的 QoS。DPS 框架同时将物理 I/O 设备带宽和用户应用程序性能作为优化的目标，在虚拟域之间有效调度资源。本文以网卡作为典型 I/O 设备示例分析对象。

虚拟机中各虚拟域承载的业务与虚拟域本身密切相关，为提高域间的性能隔离能力，便于对任务进行管理与调度，在特权域中为各用户域建立对应的静态任务队列。相比于动态队列，静态队列能够减少 I/O 任务频繁调度时初始化和上下文保存的开销。来自各用户域的任务或发送到各用户域的任务，到达特权域后进入与用户域对应的任务队列排队等候处理，由特权域统一管理和调度。DPS 框架的详细设计如图 2 所示。DPS 由属性提取模块、优先级分派模块和调度模块组成。DPS 在提交当前任务后，并不阻塞执行，而是对所有在各队列头的下步待执行 I/O 任务进行预处理，提取各队列头任务的评估属性，依据属性量化值计算得到待调度任务的优先级值，待当前任务执行完成后，立即提交下一任务，从而提高处理效率。DPS 任务的处理流程如图 3 所示。因此，不需要对物理主机或虚拟用户域进行修改，将 DPS 作为特权域中任务调度功能的扩展，就能将 DPS 集成到虚拟机系统中。

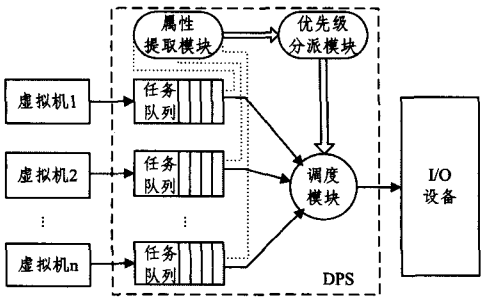

图 2　DPS 框架

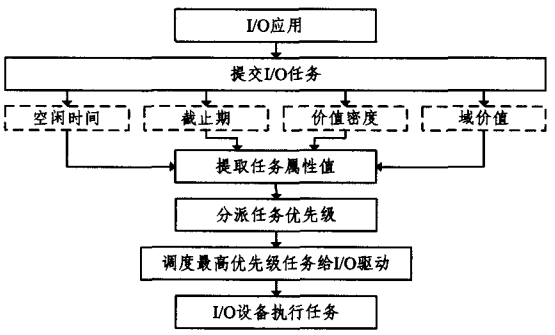

图 3　DPS 任务处理流程

4 任务优先级分派

4.1 分派原则

针对任务优先级分派问题，本文提出 DPS 任务优先级分

1) https://en.wikipedia.org/wiki/Attribute

派模型(DPS Task-Priority Assignment Model, TPAM)。TPAM 模型使用多种指标从多角度对任务进行评价,并引入多属性决策理论,将任务的遴选转化为多属性决策问题加以解决。多属性决策包含决策方案集和评估属性两个要素。决策方案集是候选决策方案的集合,评估属性是指决策过程中需要参考的各项指标[22]。确定任务优先级的多属性决策过程,即依据任务的多个属性,按照某种决策准则进行多个任务的优先级评定与选择。其中,任务的评估属性对应决策属性,待调度任务集合对应决策方案集。对于 I/O 任务的优先级排序,选取任务的空闲时间、截止期和价值密度作为任务优先级评估属性。同时,为避免只考虑任务属性值差异引起的可变权重,却忽略了虚拟用户域重要性对任务排序的影响,造成任务排序结果偏离真实预期,在任务优先级排序过程中引入任务所在域的价值作为评价指标。

TPAM 设计目标是:在 I/O 任务调度时综合考虑任务的空闲时间、截止期、价值密度和任务所在虚拟域的价值 4 个评估属性,使得虚拟机在 I/O 设备过载时的服务能力能够平缓地降级,在 I/O 设备轻载时能够具有最优异的服务能力。优先级分派原则是:

1)空闲时间越少、截止期越早、价值密度越高、虚拟域价值越高,任务的优先级越高;

2)对于任意两个任务,在空闲时间、截止期、价值密度和虚拟域价值 4 个特征参数中,只要有一个参数不同,那么它们就应被分配不同的优先级值,即任务的优先级分配是唯一的;

3)任务所在虚拟域的价值可以根据系统要求进行设置,体现虚拟域重要程度对 I/O 任务处理的影响,满足域间差异化服务要求。

4.2 分派模型

设图 3 中各虚拟域对应任务队列头的任务构成任务集合 $TASK=\{t_i \mid i\in[1,n]\}$,其中 n 为承载 I/O 应用的虚拟域数量。虚拟机 I/O 任务调度,即在任务集 $TASK$ 中选择最优先的任务执行。现有评估属性集合 $ATTR=\{A_1,A_2,\cdots,A_m\}$,分别对应任务的空闲时间、截止期、价值密度和任务所在虚拟域价值,对于任意任务 $t_i\in TASK$,有从不同角度得到的 m 个评价结果,即 t_i 的属性$(a_{i1},a_{i2},\cdots,a_{im})$,向量$(a_{i1},a_{i2},\cdots,a_{im})$称为任务的属性向量。根据评估属性集合 $ATTR$ 收集任务集合 $TASK$ 中各个任务的属性,形成如下决策矩阵:

$$ATTR=\begin{matrix}\\ t_1\\ \vdots\\ t_n\end{matrix}\begin{matrix}\begin{matrix}A_1 & \cdots & A_m\end{matrix}\\ \begin{bmatrix}a_{11} & \cdots & a_{1m}\\ \vdots & \ddots & \vdots\\ a_{n1} & \cdots & a_{nm}\end{bmatrix}\end{matrix} \tag{1}$$

本文中使用任务的空闲时间、截止期、价值密度和任务所在虚拟域的价值 4 个指标作为评估属性,决策矩阵如式(2)所示:

$$C=\begin{bmatrix}ST_1 & DL_1 & VD_1 & DO_1\\ ST_2 & DL_2 & VD_2 & DO_2\\ \vdots & \vdots & \vdots & \vdots\\ ST_n & DL_n & VD_n & DO_n\end{bmatrix} \tag{2}$$

其中,矩阵 C 的第 i 行是任务 t_i 的属性向量,即(ST_i,DL_i,VD_i,DO_i)。ST_i,DL_i,VD_i,DO_i 分别是任务 t_i 的空闲时间、截止期、价值密度和域价值,$i\in\{1,2,\cdots,n\}$。

基于上述定义,本文提出一种基于多属性决策的任务优先级分派模型 TPAM,表示如下:

$\{ATTR,TASK,\delta,C,\rho,w,\varphi,RANK\}$

其中:

(1)$ATTR$ 是任务评估属性集合。$ATTR=\{A_1,A_2,\cdots,A_m\}$,其中 $A_j,j\in\{1,2,\cdots,m\}$是具体的评估属性。本文中:

$ATTR=\{$空闲时间(ST),截止期(DL),价值密度(VD),虚拟域价值(DO)$\}$

即,本文使用任务空闲时间(ST)、截止期(DL)、价值密度(VD)和虚拟域价值(DO)对待调度任务进行评价。

(2)$TASK$ 是各队列头任务的集合。$TASK=\{t_1,t_2,\cdots,t_n\}$,其中 $t_i,i\in\{1,2,\cdots,n\}$是具体的任务。

(3)δ 是映射关系 $\delta:t_i\rightarrow(a_{i1},a_{i2},\cdots,a_{im})$,其中 $t_i\in TASK$ 是待调度任务,$(a_{i1},a_{i2},\cdots,a_{im})$是 t_i 的属性向量。映射关系表示对根据评估属性集合 $ATTR$ 产生的任务 t_i 进行多角度的评价。

(4)C 是决策矩阵。$C=\begin{bmatrix}c_{11} & \cdots & c_{1m}\\ \vdots & \ddots & \vdots\\ c_{n1} & \cdots & c_{nm}\end{bmatrix}$,$c_{ij}$ 表示第 i 个任务的第 j 个评估属性的属性值,$i\in\{1,2,\cdots,n\},j\in\{1,2,\cdots,m\}$。$C$ 中第 i 行是任务 t_i 的属性向量,$(\delta(t_1),\delta(t_2),\cdots,\delta(t_n))^{\mathrm{T}}=C$(此处 T 表示矩阵的转置)。

(5)ρ 是映射关系。$\rho:ATTR\rightarrow w$ 表示利用离差最大化法,通过评估属性值计算权重向量 w。

(6)w 是重要性权重向量。$w=(w_1,w_2,\cdots,w_m)^{\mathrm{T}}$ 表示评估属性的重要性,与决策矩阵一起参与多属性决策过程。

(7)φ 是映射关系。$\varphi:(w,C)\rightarrow RANK$,表示决策过程,即根据重要性权重向量 w 及决策矩阵 C 进行任务遴选,最终得到集合 $TASK$ 中任务的排序结果。

(8)$RANK$ 是任务的排序结果。$RANK=\{rank_1,rank_2,\cdots,rank_n\}$,其中,$rank_k=\{t_k,No_k\},k\in\{1,2,\cdots,n\}$,$t_k$ 是任务,No_k 是任务经过排序之后的序号。

TPAM 模型中各元素的关系如图 4 所示。

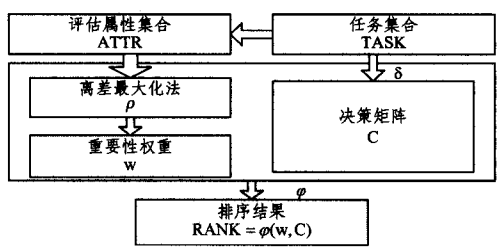

图 4 TPAM 模型元素关系图

4.3 基于离差最大化的优先级评估属性权重计算方法

本节介绍 TPAM 模型框架下的基于多属性决策的任务优先级评估属性权重计算方法,并给出基于多属性决策的任务优先级排序算法。任务属性的重要性权重对多属性决策方法的结果具有重要影响,任务属性权重的合理性直接影响任务优先级评定的准确性,因此需要采用权重计算速度快、权重表征意义与系统目标一致的赋权方法。现有多任务动态调度算法的任务属性权重依赖于专家主观赋值[23],原始数据由专家根据经验主观判断得到,存在随意性大、准确性与可靠性差的缺点。多属性离差最大化[17]决策方法是一种利用客观信息(属性值)赋权的最优化决策方法,能够有效放大评估属性之间的差异,便于任务的优选和排序。因此,本文基于离差最大化的优先级权重计算方法,实现任务的动态优先级分派。离差最大化权重计算方法具有计算速度快、实时性强、不依赖于历史数据的优点[16],相比于其他多属性决策方法,能够更好地满足 I/O 实时任务调度的需求。

评估属性指标通常分为效益型和成本型两类。效益型指

标值越大越优，成本型指标值越小越优，本文中任务的 4 种属性均属于效益型属性。由于各属性具有不同的量纲和量纲单位，为消除其不可公度性，在进行优先级分派前需要将任务的各属性指标作无量纲化处理。属性间的不可公度性是指属性间没有统一的度量标准，不能直接进行比较。

$$r_{ij}=\frac{a_{ij}-a_j^{\min}}{a_j^{\max}-a_j^{\min}},i=1,2,\cdots,n;j=1,2,\cdots,m \tag{3}$$

其中，$a_j^{\max}$，$a_j^{\min}$ 分别为第 j 个属性指标 A_j 中元素 a_{ij} 的最大值和最小值，$r_{ij}\in[0,1]$。

记无量纲化处理后的决策矩阵为 $R=(r_{ij})_{n\times m}$，r_{ij} 越大对优先级的贡献越大。设评价指标的权重 $w_j\geqslant 0,j=1,2,\cdots,m$，且满足单位化约束条件 $\sum_{j=1}^{m}w_j^2=1$。

在权重向量 w 的作用下，构造加权规范化决策矩阵：

$$D=\begin{array}{c} \\ t_1 \\ \vdots \\ t_n \end{array}\begin{array}{c} \begin{array}{ccc} A_1 & \cdots & A_m \end{array} \\ \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix} \end{array} \tag{4}$$

其中，$x_{ij}=w_jr_{ij},i=1,2,\cdots,n;j=1,2,\cdots,m$。

根据离差最大化的决策原理[17]，基于简单加性加权法(Simple Additive Weighting Method，SAW)[24]，任务 t_i 的优先级多指标综合评价值可表示为：

$$D(x_i)=\sum_{j=1}^{m}w_jr_{ij},i=1,2,\cdots,n \tag{5}$$

$D(x_i)$ 越大则任务 t_i 优先级越高，在权重向量 w 一定时，任务优先级的排序方案对应属性 A_j 差异的大小，故 A_j 对排序贡献越大，则赋予的权重越大，反之亦然。当 A_j 不能产生排序贡献时，赋予权重 0。对于属性 A_j，任务 t_i 与其他所有任务的离差 $d_{ij}(w)$ 为：

$$d_{ij}(w)=\sum_{k=1}^{n}|w_jr_{ij}-w_jr_{kj}|,i=1,2,\cdots,n;j=1,2,\cdots,m$$

对于属性 A_j，各任务与其他任务之间的总离差 $d_j(w)$ 为：

$$d_j(w)=\sum_{i=1}^{n}d_{ij}(w)=\sum_{i=1}^{n}\sum_{k=1}^{n}|r_{ij}-r_{kj}|w_j,j=1,2,\cdots,m$$

对于整个评估属性集合 $ATTR=\{A_1,A_2,\cdots,A_m\}$，所有任务的总离差 $d(w)$ 为：

$$d(w)=\sum_{j=1}^{m}d_j(w)=\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{k=1}^{n}|r_{ij}-r_{kj}|w_j$$

由于权重向量 w 应使 $d(w)$ 最大，因此可将求解权重向量 w 转化为求解如下最优化问题：

$$\begin{cases}\max d(w)=\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{k=1}^{n}|r_{ij}-r_{kj}|w_j \\ \text{s.t.}\sum_{j=1}^{m}w_j^2=1\end{cases} \tag{6}$$

利用拉格朗日乘数法解此最优化模型，并进行归一化处理，可得目标函数的唯一极大值点 $\hat{w}=(\hat{w}_1,\hat{w}_2,\cdots,\hat{w}_m)^{\mathrm{T}}$ 为：

$$\hat{w}_j=\frac{\sum_{i=1}^{n}\sum_{k=1}^{n}|r_{ij}-r_{kj}|}{\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{k=1}^{n}|r_{ij}-r_{kj}|},j=1,2,\cdots,m \tag{7}$$

利用式(5)、式(7)，得到待调度任务的优先级多指标综合评价值，依据该值对任务进行排序，选择优先级最高的任务交由 I/O 设备处理。上述整个过程即为 TPAM 模型中的映射关系 φ。

在此基础上得到 TPAM 任务优先级排序算法，如表 1 所列。

表 1　TPAM 任务优先级排序算法

Algorithm 1　I/O tasks rank
Input: Set of tasks to be ranked
Output: Rank of tasks
1. Get the evaluation attributes of each task
2. Construct matrix C with evaluation attributes
3. Rank the values of each attribute for limit operation
4. Construct matrix R non-dimensional with limit according to C
5. $\hat{w}$=MaxDeviationMethod(R)
6. Construct the weighted-standardization decision matrix D with R and $\hat{w}$
7. Get the evaluation value $D(x_i)$ of each task by D
8. Rank tasks according to the evaluation value $D(x_i)$

该算法首先获取任务的评估属性，建立属性矩阵 C。按类别对每项属性进行排序得到其最大值(Maximum Value)和最小值(Minimum Value)，使用式(3)对属性进行无量纲化处理得到决策矩阵 R。使用离差最大化计算方法，得到使整个属性集离差最大化的属性权重归一化向量 $\hat{w}$，利用 $\hat{w}$ 可计算出各任务的多属性综合评价值，即任务的优先级，据此对任务排序，选择最优先的任务提交给 I/O 设备进行处理。

5　验证与分析

5.1　实验设置

实验环境如下：服务器物理主机配置为 1 颗 8-core 3GHz Intel Xeon E5-2690 CPU、128GB 内存、1GB 网络接口卡，Xen4.4.1 虚拟机监视器，CentOS 6.5 操作系统。建立 8 个虚拟用户域，各分配 4GB 内存、1 个虚拟 CPU、1 个虚拟网卡，采用半虚拟化模式调用物理网卡进行网络数据的传输。实验采用网络测试工具 Netperf[1] 作为 I/O 密集型应用测试工具，通过 Netperf 的 TCP_STREAM 流式通讯功能测试调度算法性能。Netperf 服务器端运行在测试服务器的虚拟用户域中，Netperf 客户端运行在客户端主机中。

实验首先对比任务的优先级排序时间与 I/O 执行时间，以验证优先级排序占用的 CPU 时间能够满足系统性能要求；然后分析 DPS 的基本性能；最后将 DPS 与 Xen4.4.1 原始调度算法(Xen-Prim)、PriDyn 算法[10] 进行性能比较分析。性能实验采用截止期保证率与 I/O 带宽两项衡量任务调度性能的关键指标[23] 来评价算法优劣。

5.2　优先级排序与 I/O 执行时间对比

相比于 Xen-Prim，DPS 增加了优先级排序的 CPU 计算开销。I/O 密集型应用主要占用 I/O 资源，CPU 在 I/O 设备执行任务期间主要处于自旋等待数据或空闲循环的状态，且服务器通常配置多颗多核 CPU，在 I/O 设备处理当前任务期间，即可利用闲置 CPU 对待执行任务进行优先级排队预处理。优先级排序与 I/O 执行是 I/O 任务处理过程中的两个串行步骤，I/O 任务采用流水化处理，因此虽然单个任务延迟增加，但当优先级排序时间不超过 I/O 执行时间时，不会影响系统整体性能。图 5 为单个任务的优先级排序时间与 I/O 执行时间的对比。

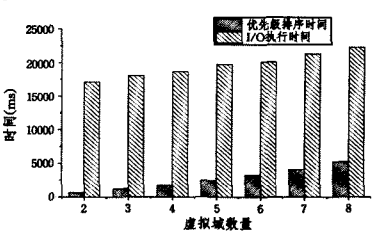

图 5　优先级排序时间与 I/O 执行时间对比

1) http://www.netperf.org/netperf/training/Netperf.html

对比实验采用随机生成的不同长度数据包的测试任务集进行发送和接收测试，每个虚拟域部署一个 I/O 应用。服务器用户域中的 Netperf 服务器端发送 TCP 包到客户端，当收到客户端发回的相同字节数的响应后，再继续发送下一包。分别启用 2—8 个虚拟用户域进行实验，每组实验重复 20 次，统计实验结果的平均值。如图 5 所示，优先级排序时间远少于 I/O 执行时间，完全符合 I/O 任务性能要求。

5.3 基本性能分析

基本性能实验通过改变典型任务参数从而引起任务属性发生变化的方法，分析 DPS 调度性能。各虚拟域中的 Netperf 服务器端分别发送相同字节数的 TCP 包到客户端，当收到客户端发回的相同字节数的响应后，再继续发送下一包，每个虚拟域部署一个 I/O 应用。任务价值和虚拟域价值取值范围为 1～100，由于域价值与任务价值对 I/O 调度的影响对属性变化的影响一致，因此只对变化任务价值时的性能进行实验分析。截止期按 10s 间隔设置。获取的截止期保证率包含测试域和系统全局两部分，I/O 带宽为系统全局带宽。

实验 1 固定数据长度、截止期、域价值，变化任务价值，分析任务价值对算法性能的影响。实验结果如图 6 所示，随着任务价值的增大，测试域的截止期保证率增加，全局的截止期保证率则逐渐下降。因为价值密度高的任务优先获得 I/O 调度执行，任务价值增加则价值密度也相应增加，系统 I/O 资源优先保障测试域的应用，提高了测试域任务成功执行的概率，截止期保证率增加。同时，可供分配给系统其他域的 I/O 资源则相应减少，系统全局截止期保证率下降，但降低幅度较为平缓，说明 DPS 对系统整体性能也能有效兼顾，充分利用了系统可用资源。任务价值的变化从全局来讲只是改变了任务处理的顺序，因此系统 I/O 带宽保持稳定，符合 4.1 节提出的 TPAM 设计目标。

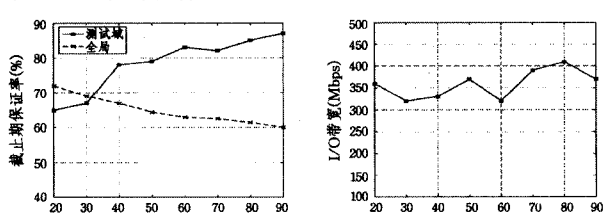

图 6 变化任务价值时的系统性能

实验 2 固定数据长度、任务价值、域价值，变化截止期，截止期测试间隔为 10s。分析由于截止期导致的时间紧迫性变化对算法性能的影响。实验结果如图 7 所示，随着截止期的加长，测试域的任务优先级降低，其他域中的任务得到执行的机会增加，各个域中任务得到执行的机会增加，因此系统全局的任务截止期保证率得到一定提高。系统总的任务处理量及速率保持稳定，因此 I/O 带宽仍基本保持稳定，同时，全局性能的平衡使 I/O 带宽波动幅度较小。

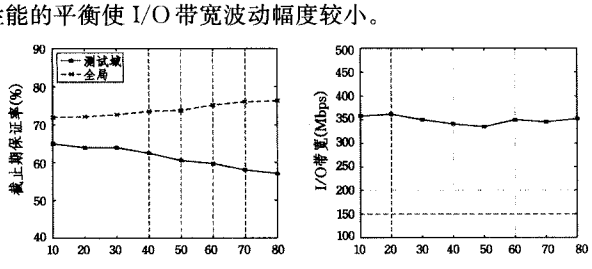

图 7 变化截止期时的系统性能

5.4 性能比较分析

性能对比实验将 DPS 算法与 Xen-Prim，PriDyn 进行对比分析。通过对不同长度的数据包进行发送和接收测试，比较分析各算法的 I/O 调度性能。实验结果如图 8 所示，其中横坐标表示任务数据长度。

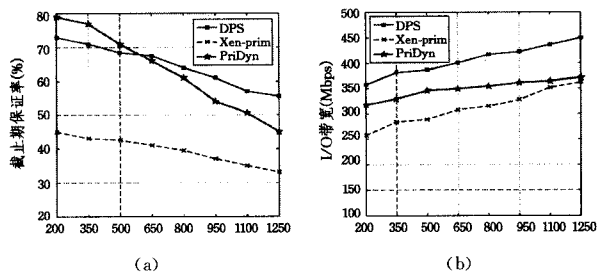

图 8 Xen-Prim，PriDyn 和 DPS 在不同数据长度时的性能对比

如图 8 所示，当系统传输的数据量较小时，DPS 截止期保证率、I/O 带宽与 PriDyn 算法相当。因为当系统负载较低时，只要能够合理安排截止期早的任务优先执行，就能保证绝大部分任务正常完成，PriDyn 算法在这方面做得最好。而基于多属性决策的 DPS 算法综合考虑任务的各种属性，会造成部分截止期早的任务错过截止期，导致截止期保证率降低。当系统负载较高时，由于系统资源的限制，通过综合考虑任务各属性，合理安排任务执行的顺序，让价值密度、时间紧迫性等较高的任务优先执行，才能获得最佳的系统性能，提高整个系统的截止期保证率。因此 DPS 算法在截止期保证率方面明显优于其他两种算法。在 I/O 带宽方面，DPS 优于其他两种算法，且随着单个任务数据量的增加，优势更加明显；PriDyn 优于顺序调度的 Xen-Prim，但在数据量较高时，优势不再明显。相比于 Xen-Prim，DPS 的 I/O 性能得到了约 20%～40%的提升；相比于 PriDyn，DPS 高出约 10%～15%。因此，DPS 在高任务负载时，通过合理调度优先级高的任务，保证了系统有序运行，获得了更高的 I/O 带宽。

结束语 针对现有虚拟机 I/O 调度算法指导性不强、对专家依赖程度高的问题，本文提出了动态优先级排序的虚拟机 I/O 调度算法 DPS，该算法采用基于多属性决策的任务优先级分派方法。通过从任务特征参数提取出任务的关键属性，获取到多角度的评估指标。在云计算环境中，虚拟机遵循 SLA 为用户提供服务，通过将任务所在域的价值也引入算法作为任务优先级的一项评估属性，体现了云计算环境中差异化服务的要求。基于提取到的评估属性，采用离差最大化方法对属性进行无量纲化处理，计算评估属性的重要性权重并进行任务优先级排序，根据计算结果遴选出优先级最高的任务提交至 I/O 设备处理。通过使用 Netperf 模拟常见的虚拟机 I/O 任务，对调度算法性能进行了评测，结果表明，DPS 能够综合全局任务调度需求，在全局性能和指定域性能上取得良好的平衡，提高系统综合性能，同时还能为云用户提供差异化服务。通过与 Xen 默认调度算法、文献[10]的调度算法进行对比，本文方法在高任务负载时有更优的截止期保证率，且 I/O 性能高出 PriDyn 10%以上、高出 Xen-Prim 20%以上，因此在截止期保证率、I/O 带宽这两项 I/O 任务调度的关键指标上性能更优。对于更广范畴的 I/O 设备，通过提取其 I/O 任务优先级的评估属性，就能将本文方法应用到 I/O 任务的调度。

相比于非虚拟化条件下的 I/O 任务执行性能，聚合在同一物理主机上的各个虚拟域存在较大的性能下降，这主要是

由 I/O 资源竞争、频繁的处理器模式切换和传输路径延长等原因引起的。本文仅针对 I/O 资源竞争问题进行虚拟机 I/O 调度的优化研究，实验结果也是基于对该问题的优化而获得，因此优化后的性能还不能接近非虚拟化条件下的 I/O 任务执行性能。后续将对减少处理器模式切换开销、缩短 I/O 任务传输路径等提高虚拟机 I/O 任务执行性能的优化措施进行研究。

参 考 文 献

[1] XIAO P, HU Z G, LIU D B, et al. Energy-efficiency enhanced virtual machine scheduling policy for mixed workloads in cloud environments[J]. Computers & Electrical Engineering, 2014, 40(5): 1650-1665.

[2] YANG Xing, MA Zi-tang, SUN Lei. Research on Extended Ant Colony Optimization Based Virtual Machine Deployment in Infrastructure Clouds[J]. Computer Science, 2012, 39(9): 33-37. (in Chinese)
杨星，马自堂，孙磊. 云环境下基于改进蚁群算法的虚拟机批量部署研究[J]. 计算机科学，2012，39(9)：33-37.

[3] NOORSHAMS Q, BUSCH A, RENTSCHLER A, et al. Automated Modeling of I/O Performance and Interference Effects in Virtualized Storage Systems[C]// Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW). Madrid, Spain: IEEE Press, 2014: 88-93.

[4] MYTILINIS I, TSOUMAKOS D, KANTERE V, et al. I/O Performance Modeling for Big Data Applications over Cloud Infrastructures[C] // Proceedings of the 2015 IEEE International Conference on Cloud Engineering (IC2E). Arizona, USA: IEEE Press, 2015: 201-206.

[5] GUAN H B, MA R H, LI J. Workload-Aware Credit Scheduler for Improving Network I/O Performance in Virtualization Environment[J]. IEEE Transactions on Cloud Computing, 2014, 2(2): 130-142.

[6] YU C, QIN L H, ZHOU J L. A multicore periodical preemption virtual machine scheduling scheme to improve the performance of computational tasks[J]. Journal of Supercomputing, 2014, 67(1): 254-276.

[7] NOORSHAMS Q, ROSTAMI K, KOUNEV S, et al. Modeling of I/O Performance Interference in Virtualized Environments with Queueing Petri Nets[C]// Proceedings of the 2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS). Paris, France: IEEE Press, 2014: 331-336.

[8] MUENCH D, ISFORT O, MUELLER K, et al. Hardware-Based I/O Virtualization for Mixed Criticality Real-Time Systems Using PCIe SR-IOV[C]// Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering (CSE). Sydney, Australia: IEEE Press, 2013: 706-713.

[9] PELEG O, MORRISON A, SEREBRIN B, et al. Utilizing the IOMMU scalably[C]// Proceedings of the 2015 USENIX Conference on Usenix Annual Technical Conference. California, USA: USENIX Association, 2015: 549-562.

[10] JAIN N, LAKSHMI J. PriDyn: Enabling Differentiated I/O Services in Cloud Using Dynamic Priorities[J]. IEEE Transactions on Services Computing, 2015, 8(2): 212-224.

[11] HWANG K, DONGARRA J, FOX G C. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things [M]. San Francisco: Margan Kaufmann, 2011.

[12] ONGARO D, COX A L, RIXNER S, et al. Scheduling I/O in Virtual Machine Monitors[C]// Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE'08). New York, USA: ACM Press, 2008: 1-10.

[13] TAN H L, HUANG L J, HE Z H, et al. DMVL: An I/O bandwidth dynamic allocation method for virtual networks[J]. Journal of Network and Computer Applications, 2014, 39(3): 104-116.

[14] BOURGUIBA M, HADDADOU K, El KORBI I, et al. Improving Network I/O Virtualization for Cloud Computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(3): 673-681.

[15] WANG Yong-yan, WANG Qiang, WANG Hong-an, et al. A Real-Time Scheduling Algorithm Based on Priority Table and Its Implementation[J]. Journal of Software, 2004, 15(3): 360-370. (in Chinese)
王永炎，王强，王宏安，等. 基于优先级表的实时调度算法及其实现[J]. 软件学报，2004，15(3)：360-370.

[16] WU Z B, FANG Y F. A Consensus and Maximizing Deviation based Approach for Multi-criteria Group Decision Making under Linguistic Setting[C]// Proceedings of the 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). Beijing, China: IEEE Press, 2014: 469-475.

[17] WANG Ying-ming. Using the Method of Maximizing Deviations to Make Decision for Multiindicies[J]. Journal of Systems Engineering and Electronics, 1998(7): 24-26. (in Chinese)
王应明. 运用离差最大化方法进行多指标决策与排序[J]. 系统工程与电子技术，1998(7)：24-26.

[18] LI X Y, MA H D, ZHOU F, et al. T-Broker: A Trust-Aware Service Brokering Scheme for Multiple Cloud Collaborative Services[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(7): 1402-1415.

[19] BARHAM P, DRAGOVIC B, FRASER K, et al. Xen and the Art of Virtualization[C] // Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03). New York, USA: ACM Press, 2003: 164-177.

[20] CHISNALL D. The Definitive Guide to the Xen Hypervisor [M]. New Jersey: Prentice Hall, 2007.

[21] NING F F, WENG C L, LUO Y. Virtualization I/O Optimization Based on Shared Memory[C] // Proceedings of the 2013 IEEE International Conference on Big Data. California, USA: IEEE Press, 2013: 70-77.

[22] HUANG Liang, FENG Deng-guo, LIAN Yi-feng, et al. Method of DDoS Countermeasure Selection Based on Multi-Attribute Decision Making[J]. Journal of Software, 2015, 26(7): 1742-1756. (in Chinese)
黄亮，冯登国，连一峰，等. 一种基于多属性决策的 DDoS 防护措施遴选方法[J]. 软件学报，2015，26(7)：1742-1756.

[23] XIA Jia-li, CHEN Hui, YANG Bing. A Real-Time Tasks Scheduling Algorithm Based on Dynamic Priority[J]. Chinese Journal of Computers, 2012, 35(12): 2685-2695. (in Chinese)
夏家莉，陈辉，杨兵. 一种动态优先级实时任务调度算法[J]. 计算机学报，2012，35(12)：2685-2695.

[24] WANG Y J. A fuzzy multi-criteria decision-making model based on simple additive weighting method and relative preference relation[J]. Applied Soft Computing, 2015, 30: 412-420.