移动云计算中时延保证的任务分配方法

蒋维成，李兰英，刘华春，侯向宁

（成都理工大学工程技术学院 电子信息与计算机工程系，四川 乐山 614000）

摘 要：为在移动云计算中给任务提供实时保障，设计任务窗口对虚拟机中的任务进行分配，根据任务截止期和任务窗口大小进行调度；监控任务的执行过程，对窗口尺寸进行动态调整和修正，采取反馈机制保障后续任务分配不受影响，确保窗口内的任务时延达标。根据任务的变化情况，建立相应的虚拟机扩展或收缩策略，保障任务能够实时完成。实验结果表明，任务能在规定的时间里得到提交，保证了实时任务可用性，系统中资源利用率高，其性能得到了很好的验证。

关键词：移动云计算；虚拟化；实时任务；调度；任务截止期

中图法分类号：TP393 **文献标识号**：A **文章编号**：1000-7024（2020）05-1214-05

doi：10.16208/j.issn1000-7024.2020.05.003

Task scheduling with delay guarantee in mobile cloud computing

JIANG Wei-cheng, LI Lan-ying, LIU Hua-chun, HOU Xiang-ning

(Department of Electronic Information and Computer Engineering, The Engineering & Technical College of Chengdu University of Technology, Leshan 614000, China)

Abstract: To provide real-time guarantee in mobile cloud computing, task windows were designed in virtual machines and tasks were scheduled according to deadlines and task window sizes. The process of tasks was monitored. The size of window was dynamically adjusted and revised, and the feedback mechanism was used to ensure that the subsequent task allocation was not affected and the delay was up to standard. According to the change of task, the methods of virtual machine expansion and contraction were proposed to meet the deadline. Experimental results show that the task can be submitted on time and the resource utilization in the system is very high. The performance of the algorithm is validated.

Key words: mobile cloud computing; virtualization; real-time task; scheduling; task deadline

0 引 言

移动云计算中的服务请求受环境影响，许多任务要求在规定的时间内完成，如过程控制、环境监测中的数据处理，以及移动手机中的各种实时信息的获取等。若时延超过某一范围将导致不利影响，或任务的失败。这些具有时延要求的任务，对服务器中的虚拟机提出了实时要求。由于这些任务种类繁多，变化很大，增加了创建虚拟机的困难，若根据任务类型创建虚拟机，将给系统带来巨大开销。另外，管理这些虚拟机也是一个复杂的问题。创建虚拟机需要时间，将推迟任务的执行，影响任务的提交，对实时任务造成不利影响。

为了缩短实时任务的响应时间，可以将任务分配到已经创建的虚拟机中。在任务指派过程中，需要考虑任务量的大小，任务量大的任务需要分配到计算能力强的虚拟机中，才能满足时延要求。此外，虚拟机中指派的任务不能过多，不然造成任务的响应时间过长，导致任务失败。为了对移动云计算中的实时任务提供时延有保障的服务，本文根据虚拟机计算能力来构建任务窗口，将任务调度到能在截止期之前完成的虚拟机中，对任务执行过程进行监控，确保任务的有效完成。

1 相关工作

云计算中任务的分配引起了学者的广泛关注和深入研

收稿日期：2019-02-26；**修订日期**：2019-11-26

基金项目：四川省教育厅科研基金项目（16ZB0412）；乐山市科技局基金项目（18JZD060）；成都理工大学工程技术学院基金项目（C122019001）

作者简介：蒋维成（1977-），男，湖南永州人，硕士，副教授，研究方向为计算机网络；李兰英（1979-），女，山东菏泽人，硕士，副教授，研究方向为计算机网络；刘华春（1966-），男，四川泸州人，硕士，副教授，研究方向为网络安全；侯向宁（1972-），男，陕西西安人，硕士，讲师，研究方向为云计算。E-mail：da2009yin@163.com

究。文献［1］提出了一种任务调度算法，把任务分为不同种类，每类任务具有相似的属性（用户类型、任务类型、任务大小和任务延迟），根据类别，选择最小执行时间的任务进行执行。文献［2］根据任务在计算节点的处理频度、等待时间和执行时间等参数构建数学模型，采用优先级来进行任务分配。为了减少结点间的数据传输，文献［3］采用数据复制方法来提高任务调度数据的效率。

云服中管理系统的服务对象包含企业生产经营的一系列活动的流程。这些流程构成信息流动和资源交换的重要载体，可以划分为多个任务单元[4]。分配和调度这类任务单元保障整个流程的顺利完成成为研究的热点问题[5,6]。通过合并低效率的处理器，降低资源使用数量，文献［7］提出了一种云工作流任务调度能效优化算法，在不违背任务顺序和截止时间约束前提下降低工作流执行总能耗。根据任务平均运行时间进行调度的算法，很难最小化任务的完成时间，建立任务的运行时间和能耗模型，文献［8］提出能耗感知的调度算法。任务复制的调度算法中存在过度复制任务造成资源浪费的现象，文献［9］最小化任务的复制量，以求达到最小化应用的完成时间和能量消耗。文献［10］提出了基于 MapReduce 的能量感知多作业调度模型，设计遗传算法，对云计算中大规模任务调度进行优化，以提高服务器的能源效率。文献［11］建立资源总租赁成本的调度优化数学模型，构造插入邻域和交换邻域的迭代局部搜索算法，以一定概率的插入和交换操作实现扰动当前解，增加群体多样性，选择算法性能最优的参数组合，构建云环境下多模态工作流调度方法。文献［12］提出了资源动态增加与收缩策略，并设计了实时任务的一种节能调度算法 EARH。文献［13］设计了基于仿生自主神经系统的云调度管理系统，利用最优性分析和自主触发机制实现局部资源管理，采用启发式算法来获取面向用户请求分发的全局最优调度策略，以实现兼顾性能和能耗的云调度管理机制。

但是，这些调度算法存在如下问题，对实时任务无法提供时延保证的服务，使得任务能够在有效截止期内完成，保障任务实时可用性，尤其是数据中心服务器负载突然增加或执行过程的某些不确定因素造成任务响应时间过大，导致实时任务的提交延误或无效。本文针对这些问题，提出一种基于任务窗口时延确保的调度方法（scheduling method for task window delay guarantee，TWDG），以提高移动云计算数据中心实时任务的可用性。

2 算法框架

2.1 任务模型

在移动云计算中，用户提交的任务具有很大的随机性，本文针对的应用是非周期的实时性要求较高的独立任务。任务执行若超时将造成不良影响。这些任务表示为 $P=\{p_1,p_2,p_3\cdots p_n\}$。这些任务的到达时间和截止期在任务到达之后才能得知。对于某任务集合 P 中的任何一个任务 $p_i \in P$，可以采用三元组来表示。$p_i=(A_i,S_i,D_i)$。这里 A_i 表示任务 p_i 的到达时间，S_i 表示 p_i 的任务量大小，D_i 表示 p_i 的截止期。

2.2 调度模型

移动云计算系统是由多个物理服务站点组成，每个站点有若干台服务器。物理服务器构成集合 $B=\{b_1,b_2,b_3\cdots b_n\}$，n 为服务器的数量。这些服务器构成为用户提供计算服务的基础设置平台（IaaS）。其中某台服务器 b_i 可以表示为 $Q=(F,E,R)$，这里 F 表示服务器的 CPU 频率（MHz），E 表示内存大小（MB），R 表示网络带宽（Mbps）。

每台服务器上创建若干个虚拟机，这些服务器上的虚拟机集合表示为 $VM=\{VM(1),VM(2),VM(3),\cdots VM(m)\}$，使用 f_i、e_i 和 r_i 表示虚拟机 VM(i) 中该虚拟机的 CPU 频率、内存大小和网络带宽。

创建虚拟机需要一定的时间[14]，这将影响实时任务的执行。为了减少因创建虚拟机导致任务时延的增加。在系统中维持一个备用虚拟机。通常备用虚拟机处于休眠状态，仅分配极少必要的资源，而当系统资源不够时，唤醒备用虚拟机，立即分配资源，并为其指派任务。此时系统中就不存在备用虚拟机，立即创建一个备用虚拟机，始终保持系统中存在一个备用虚拟机，以减少创建虚拟机的时间，确保实时任务得到及时响应。

在系统中增加一个实时任务监控器，用于监测实时虚拟机中实时任务的执行情况，监测和记录实时任务超时发生的任务个数，发生的时间，实时任务比计划时间提前完成的时间。

本文的调度模型如图 1 所示，新到达的任务经任务分析器进行分析，得到任务量的大小和截止期要求。交给调度器，调度器根据任务窗口的大小和截止期的要求，将任务分配到满足条件的虚拟机中执行。任务监测器对虚拟机中实时任务执行进行监测，对任务执行时间增加的超时现象和执行时间大幅减少的情况进行处理。并对任务窗口进行调整。

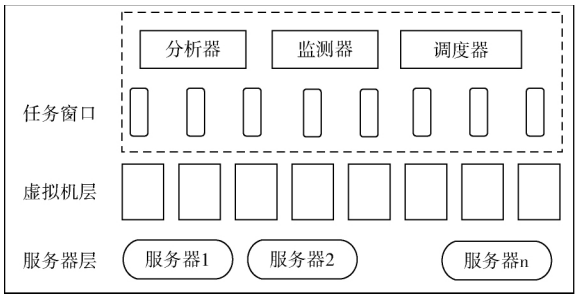

图 1 调度模型

定义 1 标准任务，用来衡量实时任务的规模和任务量大小，抽象出来的量，大小为一个单位任务量，用 S_0 表示。具体任务可以通过标准任务来衡量任务量的大小。

定义 2 虚拟机计算能力，表示虚拟机的计算处理能力，用 C 表示。它与 CPU 频率、内存大小和网络带宽有关。通常虚拟机的 CPU 频率越高，内存越大，网络带宽较高条件下，虚拟机的计算能力就越强。用式（1）表示

$$C = k_1 \cdot F + k_2 \cdot E + k_3 \cdot R \tag{1}$$

式中：k_1，k_2，k_3 为比例系数，表示虚拟机的计算能力 C 同 CPU 频率、内存大小、网络带宽之间的关系。

定义 3 任务处理速率，指单位时间内完成标准任务的数量，用 V 表示。它反应虚拟机对任务处理的快慢，与虚拟机的计算能力有关。计算能力越强，单位时间内完成标准任务越多，任务处理速率也就越大。

定义 4 任务窗口，表示分配给虚拟机任务量多少的度量，用 W 表示。计算能力越强的虚拟机其任务窗口越大，在相同的截止期条件下，该虚拟机可以分配的任务就越多。反之，则越少。每个虚拟机根据其计算处理能力构建相应的任务窗口，它是任务指派的依据和度量。虚拟机受自身计算能力的限制，超过任务窗口的任务将不再分配给该虚拟机，只有任务量小于任务窗口尺寸的任务才能指派给虚拟机。每分配一个任务，任务窗口尺寸减小相应的值。当任务窗口尺寸减少到 0 时，不再给虚拟机分配任务。相应的，完成一个任务则对任务窗口增加相应的量。它是一个动态变化的量。

对于任务量大小为 S_i 的任务，在指派任务后虚拟机任务窗口变为 $W = W - \delta \cdot \frac{S_i}{S_0}$，式中 δ 为比例系数，表示任务与任务窗口尺寸之间的关系。相应的，虚拟机每完成一个任务，则将任务窗口的值增加一定的量。任务 S_i 完成后，虚拟机任务窗口变为 $W = W + \delta \cdot \frac{S_i}{S_0}$，$S_i$ 为该任务的任务量。增大任务窗口，以便虚拟机可以接纳新的任务。

假设实时任务 p_i 的任务量为 S_i，那么任务 p_i 在虚拟机 VM(j) 中的运行时间为 $T_a = \frac{S_i}{V_j}$，式中 V_j 为虚拟机 VM(j) 的任务处理速率。任务 p_i 的截止期为 D_i，那么任务 p_i 可以推迟执行的时间为 $T_d = D_i - A_i - T_a$。

若虚拟机 VM(j) 中已经分配的任务队列中的待处理任务集合为 $P = \{p_1, p_2, p_3 \cdots p_m\}$，这些任务相应的任务量为 $\{S_1, S_2, S_3 \cdots S_m\}$，那么等待时间为式（2）

$$T_w = \sum_{i=1}^{m} \frac{S_i}{V_j} = \frac{1}{V_j} \sum_{i=1}^{m} S_i \tag{2}$$

任务分配时，首先查找任务窗口尺寸大于任务所需尺寸大小的虚拟机，若存在，则在这些虚拟机集合中，进一步查找等待时间 $T_w < T_d$ 的虚拟机集合，若找到一个则进行任务的分配；若所有虚拟机均不满足条件，则激活备用虚拟机，为它分配相应的资源，并把任务分配到虚拟机上。

2.3 窗口调整和资源伸缩

受运行环境的影响，虚拟机中实时任务的执行时间可能增加或缩短，若变化的时间很短，对实时任务的影响不大，而当某一任务在截止期到来时仍未完成，将影响后续任务的执行，可能导致后续任务超时。为了避免这种现象，立即减少该虚拟机的任务窗口尺寸。设发生超时时，虚拟机所处理的任务为 p_k，发生一次超时任务窗口减小的量为 $\Delta W = 2\delta \cdot \frac{S_k}{S_0}$，式中 S_k 为发生超时的任务量大小。这样考虑的原因是，发生超时的最早时间点为此任务刚开始执行的起点。如果仅减去处理 S_i 所需时间，难以保证后续任务在截止期内完成。另外，超时造成的相关处理，也将增加时间开销。因此，为了确保后续任务能够在截止期内完成，选取处理 S_i 所需时间的 2 倍。虚拟机的任务窗口变为式（3）

$$W = W - 2\delta \cdot \frac{S_k}{S_0} \tag{3}$$

当任务提前较多时间完成时，或较多的任务均提前完成时，表明虚拟机处理能力有较多剩余，可以分配更多的任务。若监测到某任务 p_j 的执行提前的时间为 T_b 时，增大的任务窗口量为 $\Delta W = \frac{1}{2}\delta \cdot T_b \cdot V_j / S_0$。这里仅仅只增加提前时间的一半任务窗口量，是确保增加任务窗口后的任务能够满足时延要求，窗口增加速度是比较慢的。如果任务窗口的尺寸仍小于虚拟机的计算能力，那么将有更多的任务提前完成，从而继续导致任务窗口增加。虚拟机的任务窗口变为式（4）

$$W = W + \frac{1}{2}\delta \cdot T_b \cdot V_j / S_0 \tag{4}$$

虚拟机中发生任务超时采用 2 倍任务窗口量进行减少，而对虚拟机中任务提前完成仅增加 1/2 倍任务窗口量进行增加，这样来对任务窗口进行修正，使得任务窗口的大小动态地与任务量大小相平衡，符合虚拟机计算机能力的要求，减少超时的发生，保证实时任务能够在截止期到来之前完成。

若系统中任务窗口长期处于最大值，表明该虚拟机较长时间处于空闲状态。如果系统中存在较多的虚拟机处于闲置状态，可以将这类虚拟机中的任务进行合并，从而关闭一部分虚拟机，减少系统资源的浪费。

当出现某任务的提交超时时，对任务窗口中的任务执行造成不利影响，此刻需要对已分配的任务进行检查，设超时造成的时间增加量为 $\Delta t'$，若后续任务均可以在截止期内完成，则表明超时造成的时间增加 $\Delta t'$ 是允许的，不会造成后续任务的超时，只需把虚拟机的任务窗口减少；若可能造成后续任务超时时，则将这些任务按新的到来时间，重新分配到计算能力强且负载又低的虚拟机中。

3 结果分析

为了检验本文提出的 TWDG 算法的性能，与任务分配方法 FIFO 算法和 RR 算法进行了比较实验。实验过程采用 Cloudsim 仿真平台和 Opennet 软件来模拟移动云计算中的基础设置。采用 Google 云服务系统中任务数据和仿真数据源来对随机任务进行分配。根据文献［14］来设置任务周期和截止期等信息。任务周期（T_c）由任务开始时间（t_s）和结束时间（t_e）、主机 CPU 的平均利用率来确定，如式（5）所示

$$T_c = (t_e - t_s) \times U_a \times C' \tag{5}$$

式中：C'为主机 CPU 的处理能力。任务的执行时间由任务周期和虚拟机 CPU 频率决定。如式（6）所示

$$T_t = T_c / f_i \tag{6}$$

采用截止期基准来控制任务的截止期，具体计算公式如式（7）所示

$$D_k = A_i + (t_e - t_s) \times \beta_d \tag{7}$$

式中：β_d 为截止期基准。

任务和符合时延要求的任务数结果如图 2 所示。

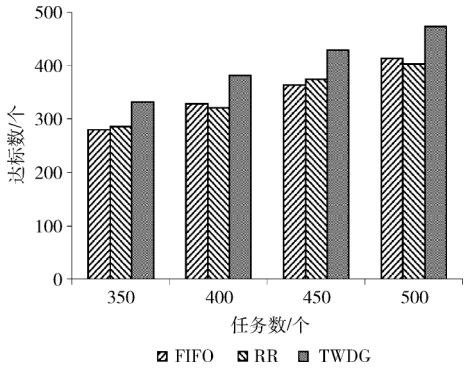

图 2 时延符合要求任务

从图 2 可以看出，随着任务规模的增加，符合时延要求的任务数不断增加。在任务数为 350～500 这 4 种不同情况中，TWDG 算法时延达标的数量是相当高的，达标任务数接近于任务数。而 FIFO 算法和 RR 算法中时延达标的数量却比 TWDG 算法少。这主要是这两个算法缺少对任务截止期的考虑，另外，FIFO 算法和 RR 算法中时延达标数也不稳定，偶然性比较大，存在时高时低的现象。TWDG 算法采用基于任务窗口的方式来对任务进行分配，把任务分配给能在截止期之前完成的虚拟机中执行，保证了任务的实时可用性。

图 3 是在截止期基准率不断增大情况下，提交任务中符合时延要求的任务数。随着截止期基准率的增大，任务时延增加，实时性要求降低。从图 3 可用看出，TWDG 算法受截止期基准变化影响较小，时延达标任务都很高。而 FIFO 算法和 RR 算法则受截止期基准率变化影响较大，FIFO 算法和 RR 算法在截止期允许范围增大时，达标任务数增加较多。TWDG 算法中备用虚拟机减少了创建虚拟机时延增加，保证任务突然增多时能及时地进行响应，从而增加了达标的数量。因此，无论是在截止期基准较小的高实时要求下，还是在截止期基准较大的低实时情况下，达标的任务数都是很高的，表明 TWDG 算法的应用范围广，既适合时延要求高的环境，也适合时延要求低的环境。

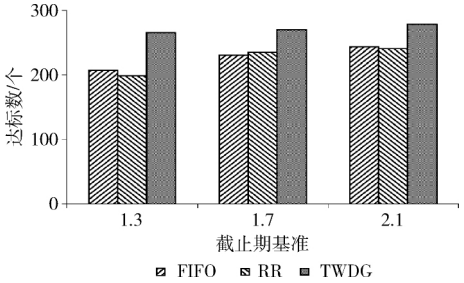

图 3 截止期影响

图 4 是任务截止期相同情况下，三算法的吞吐量的比较。从图 4 可用看出，在同等任务条件下，TWDG 算法所耗时间要少，因而吞吐量要比 FIFO 算法和 RR 算法要大。TWDG 算法根据虚拟机的计算能力来构建任务窗口大小，对于计算能力存在差异的异构型系统来说，有利于保持负载均衡，从而提高了任务的吞吐量。

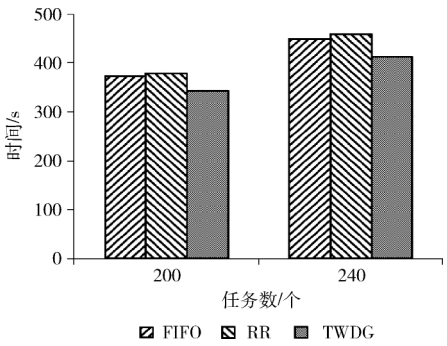

图 4 耗时

4 结束语

移动云计算中终端设备种类繁多，服务请求多种多样，很多任务具有实时要求，增加了服务器处理的复杂性，为这些任务提供实时保障的服务具有挑战性。本文根据虚拟机的计算能力来构建任务窗口，按任务截止期和任务窗口大小来进行实时任务的调度，保证分配到虚拟机的任务都符合时延要求，同时，对虚拟机中任务执行进行监测，进一步减少超时发生，并建立相应的反馈机制来保障后续任务的分配不受超时影响，提高实时任务分配的可用性。为了降低因创建虚拟机而延误实时任务的执行，在系统中增

加备用虚拟机，备用虚拟机在普通情况下处于休眠状态，占用极少资源，在需要时激活，能够及时进行响应，减少时延。建立服务器中资源的伸展和收缩策略，以提高资源的利用率。本文算法能有效地应对任务量的变化，满足时延要求，适用范围广。

参考文献：

[1] Ali HGEDH，Saroit IA，Kotb AM. Grouped tasks scheduling algorithm based on QoS in cloud computing network [J]. Egyptian Informatics Journal，2017，18 (1)：11-19.

[2] YUAN Youwei，YU Jia，ZHENG Hongsheng，et al. Cloud workflow scheduling algorithm based on novelty ranking and multi-quality of service [J]. Journal of Zhengjiang University：Engineering Science，2017，51 (6)：1190-1196 (in Chinese). [袁友伟，余佳，郑宏升，等. 基于新颖性排名和多服务质量的云工作流调度算法 [J]. 浙江大学学报（工学版），2017，51 (6)：1190-1196.]

[3] Casas I，Taheri J，Ranjan R，et al. A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems [J]. Future Generation Computer Systems，2017，74 (9)：168-178.

[4] Ahmed W，Wu Y，Zheng W. Response time based optimal web service selection [J]. IEEE Transactions on Parallel & Distributed Systems，2015，26 (2)：551-561.

[5] WEI Wei，LIU Yang，YANG Weidong. A fast approximation algorithms for the general resource placement problem in cloud computing platform [J]. Journal of Computer Research and Development，2016，53 (3)：697-703 (in Chinese). [魏蔚，刘扬，杨卫东. 一种通用云计算资源调度问题的快速近似算法 [J]. 计算机研究与发展，2016，53 (3)：697-703.]

[6] Mao C，Chen J，Towey D，et al. Search-based QoS ranking prediction for web services in cloud environments [J]. Future Generation Computer Systems，2015，50 (C)：111-126.

[7] WANG Guohao，LI Qinghua，LIU Anfeng. Energy-efficiet optimization algorithm of cloud workflow task scheduling [J]. Computer Engineering and Applications，2018，54 (10)：90-98 (in Chinese). [王国豪，李庆华，刘安丰. 一种云工作流任务调度能效优化算法 [J]. 计算机工程与应用，2018，54 (10)：90-98.]

[8] Li K，Tang X，Li K. Energy-efficient stochastic task scheduling on heterogeneous computing systems [J]. IEEE Transactions on Parallel & Distributed Systems，2014，25 (11)：2867-2876.

[9] Mei J，Li K，Li K. Energy-aware task scheduling in heterogeneous computing environments [J]. Cluster Computing，2014，17 (2)：537-550.

[10] Wang X，Wang Y，Cui Y. An energy-aware bi-level optimization model for multi-job scheduling problems under cloud computing [J]. Soft Computing，2016，20 (1)：303-317.

[11] WANG Hongxin，ZHANG Yue. Scheduling method for multi-mode cloud workflows with deadlines [J]. Journal of Chinese Computer Systems，2016，37 (11)：2437-2442 (in Chinese). [王宏欣，张跃. 带截止期约束的多模态云服务工作流调度 [J]. 小型微型计算机系统，2016，37 (11)：2437-2442.]

[12] Zhu X，Yang LT，Chen H，et al. Real-time tasks oriented energy-aware scheduling in virtualized clouds [J]. IEEE Transactions on Cloud Computing，2014，2 (2)：168-180.

[13] QIU Xiwei，DENG Zixuan，SUN Peng，et al. Research on energy-efficient cloud scheduling based on bionic autonomic nervous systems [J]. Application Research of Computers，2016，33 (10)：3011-3016 (in Chinese). [邱曦伟，邓紫璇，孙鹏，等. 基于仿生自主神经系统的节能高效云调度研究 [J]. 计算机应用研究，2016，33 (10)：3011-3016.]

[14] CHEN Huangke，ZHU Jianghan，ZHU Xiaomin，et al. Resource-delay-aware scheduling for real-time tasks in clouds [J]. Journal of Computer Research and Development，2017，54 (2)：446-456 (in Chinese). [陈黄科，祝江汉，朱晓敏，等. 云计算中资源延迟感知的实时任务调度方法 [J]. 计算机研究与发展，2017，54 (2)：446-456.]