

云计算中面向 SLA 的作业分层优先级调度策略

林清滢 陆锡聪 徐 林

(韩山师范学院计算机应用与技术系 潮州 521041)

摘 要 为了满足云计算中服务提供商和服务消费者双方协商的服务等级协议(SLA),提出在云计算环境下采用元调度和本地调度两层框架结构;在每层上采用了多级反馈队列调度算法,设置了基于截止时间、短作业优先等不同优先级的队列,并设计元调度器完成资源的分配和监控。实验仿真结果表明,该调度策略可以满足用户对作业处理时间的需求,保障了云计算 SLA。

关键词 云计算,分层调度,服务等级协议,作业优先级

中图法分类号 TP393 文献标识码 A

Hierarchical Priority Scheduling Strategy of Job for SLA in Cloud Computing

LIN Qing-ying LU Xi-cong XU Lin

(Department of Computer Application and Technology, Hanshan Normal University, Chaozhou 521041, China)

Abstract In order to meet service level agreements between the service provider and the service consumer in cloud computing, this paper proposed a framework with meta-level scheduling and local-level scheduling in the cloud environment. It uses multi-level feedback queue scheduling algorithm in each level. The scheduling algorithm with job prioritization is used to give more priority for deadline based and shortest jobs. Also, the meta-scheduler is designed to allocate and monitor resources. Finally, the experiments simulated show that our strategy can satisfy the consumer on response time of job and can meet the SLA.

Keywords Cloud computing, Hierarchical scheduling, Service level agreement, Job priority

云计算主要是以服务消费者为中心的“按需使用、按量付费”的商业服务模式,在这种商业服务模式,作业的处理时间和执行费用就成了服务消费者最关心的问题,通常情况下,云服务提供商和服务消费者需要通过协商谈判签订 SLA (Service Level Agreement, 服务等级协议)来保证自身利益和服务质量^[1]。但是在云计算这样大规模集群环境下,系统几乎时刻都在处理海量的作业,所以如何对云资源进行合理的分配,对作业进行高效的调度,以满足用户对作业处理时间的需求,这成了云计算研究中的重点和难点。因此,本文采用分层调度模式,面向 SLA 提出了基于截止期限为最高优先级的多级反馈队列的作业调度策略,以满足用户实际的服务质量需求,在一定程度上保障了云计算 SLA。

1 相关工作

目前,各大云计算提供商会根据自己已有的架构搭建云计算平台,并根据自己资源的特点采用不同的调度策略。其中较为著名的是 IBM、Google、Yahoo 等云计算平台,另外,OpenNebula 和 Eucalyptus 也是比较有影响的开源云计算平台。IBM、Google、Yahoo 等云平台主要采用了主从式调度模式,该调度模式是由主节点控制整个系统的作业调度^[2]。早期主节点的调度采用的是 FIFO 调度算法,FIFO 调度算法中

用户作业都被提交到唯一的一个队列内,从节点在它们空闲时向主节点请求分配任务^[3]。FIFO 的优点是实现非常简单、调度过程快和占用的资源较少,缺点是资源的利用率不高,并且会导致短作业的响应时间超出容忍范围的问题。针对 FIFO 资源的利用率不高的问题,Yahoo 公司开发了计算能力调度算法(Capacity Scheduler),该算法采用多个队列,每个队列分配一定的系统容量,空闲资源可以动态分配给负荷重的队列,支持作业优先级,优点是支持多作业并行执行,提高资源利用率,动态调整资源分配,提高作业执行效率,缺点是用户需要了解大量系统信息,才能设置和选择队列^[4]。此外,研究人员针对主从式调度模式,设计了适应于用户特殊需求的调度算法,如适用于实时作业的调度算法 Deadline Scheduler^[5]和 Constraint-based Scheduler^[6],Deadline Scheduler 是根据作业的运行进度和剩余时间动态调整作业获得的资源值,以便作业尽可能在截止期限内完成。Constraint-based Scheduler 则根据作业的期限和当前系统中的实时作业运行情况,预测新提交的作业能不能在规定期限内完成,如果不能,则将作业反馈给用户,让他重新调整作业的截止时间。这类调度算法虽然能满足用户的基于截止期限的服务需求,但与 FIFO 的调度算法一样,存在短作业的响应时间超出容忍范围的问题。

本文受广东省科技计划项目(2012B010100036),韩山师范学院科研团队项目(LT201102)资助。

林清滢(1970—),女,硕士,副教授,主要研究方向为云计算、调度理论,E-mail:lqying@hstc.edu.cn;陆锡聪(1973—),男,硕士,高级实验师,主要研究方向为云计算;徐 林(1975—),女,博士,讲师,主要研究方向为网络安全、云安全。

2 作业分层优先级调度策略

2.1 分层调度框架结构

如图 1 所示,在云计算平台上采用了层次式调度模式。作业调度过程分为两层:元调度和本地调度,每层调度都由相应的调度器完成。即在云控制器中采用元调度器,在集群控制器中采用本地调度器。

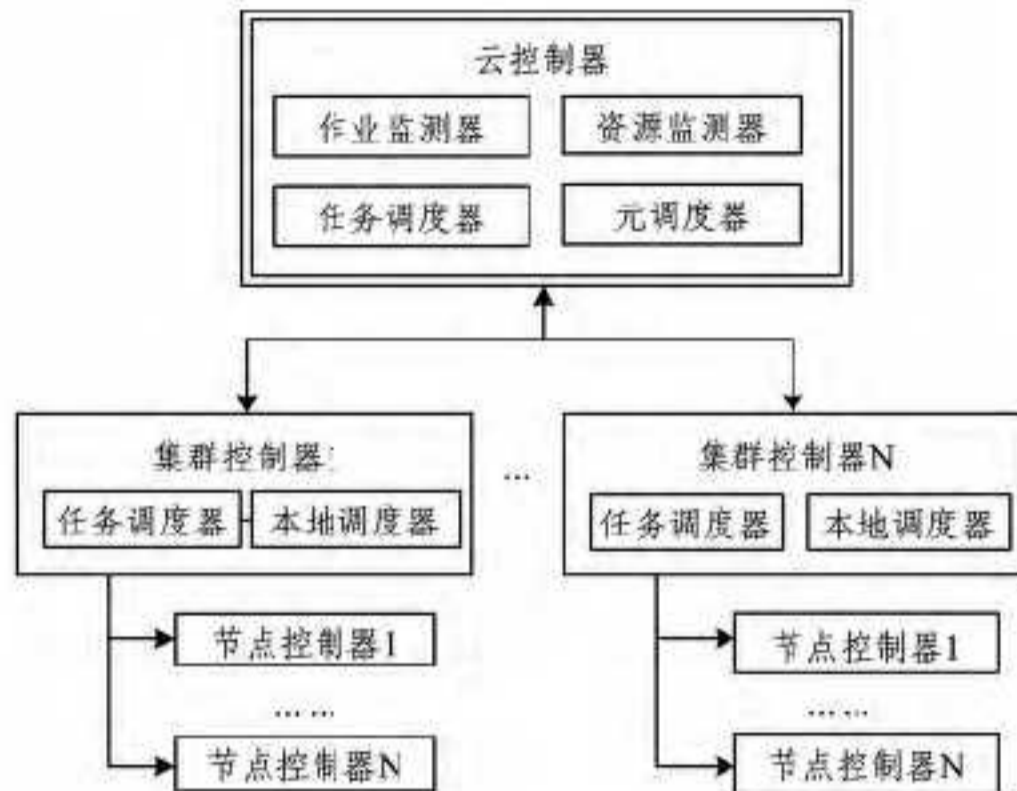


图 1 云计算环境下分层调度的框架结构图

2.2 优先级调度策略

云控制器的主要工作是管理和控制其下的所有集群控制器,它主要是由作业监测器、任务调度器、资源监测器、元调度器 4 大模块组成。这 4 大模块的工作流程如图 2 所示。

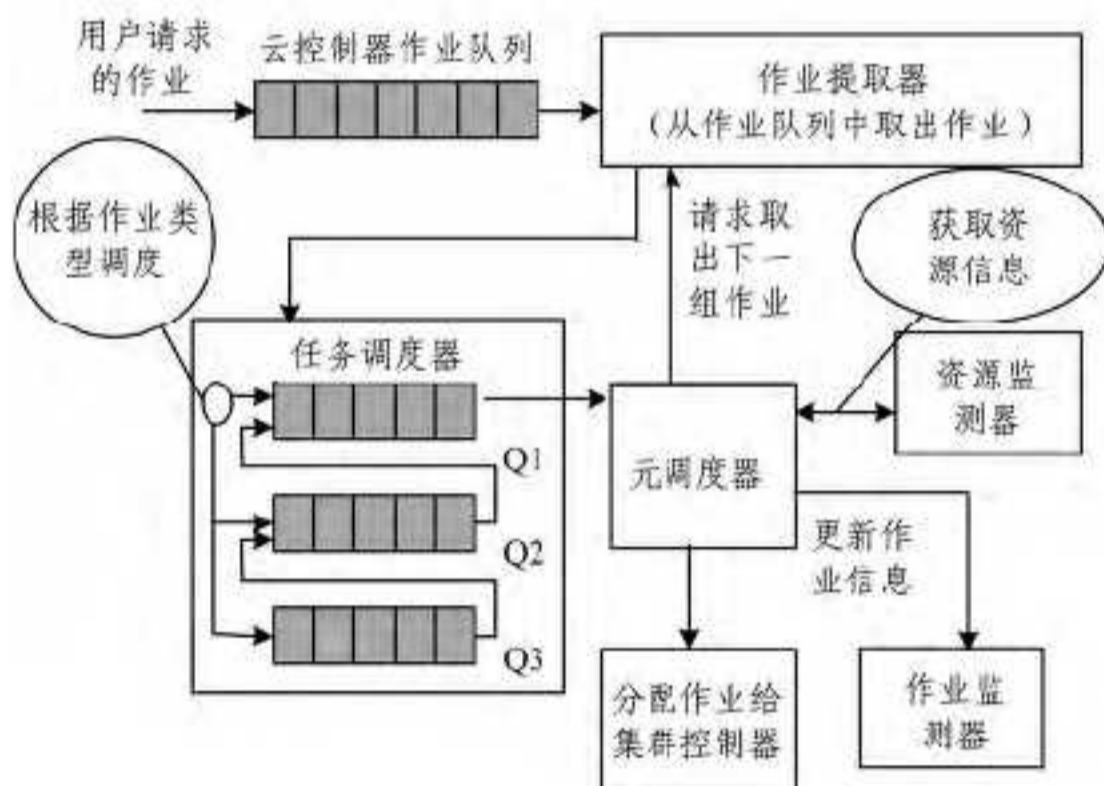


图 2 云控制器的工作流程图

用户的作业被推送进云控制器的作业队列中,每隔一段时间作业提取器会从作业队列中取出作业,然后发送给任务调度器,任务调度器将根据作业优先级类型安排作业给调度进程,如果作业是基于截止期限的,那么任务调度器就把作业压进最高优先级的队列 Q_1 ,如果作业是短作业,那么任务调度器就把作业压进第二优先级的队列 Q_2 ,其余的作业就直接被压进最低优先级的队列 Q_3 中。由于队列 Q_1 、 Q_2 、 Q_3 是首尾相连的,因此,假设有一个作业从 Q_1 队列中移入元调度器,那么 Q_2 中的作业会自动移进 Q_1 队列尾,同样 Q_3 中的作业会自动移进 Q_2 队列尾。

元调度器一方面从 Q_1 中得到作业后,运用负载调度算法分配作业给集群控制器,另一方面,从资源监测器中获取资源信息,然后在调度进程中更新作业监测器的作业信息,假如元调度器在任务调度器中找到空队列,那么它将通知作业提取器从作业队列中取出下一组作业。其中元调度器使用的负载调度算法描述如下:

- Step 1 从 Q_1 队列中取出作业。
- Step 2 确定作业的请求信息。
- Step 3 查询资源监测器并更新资源信息。
- Step 4 计算所有资源或集群控制器的资源负载量。
- Step 5 找出具有最少负载的资源。
- Step 6 把作业分配给匹配的资源。
- Step 7 转向 Step 1。

该算法将使用式(1)和式(2)计算资源负载量(Load Cost of Resource, $LC(R)$):

$$LC(R_i) = QL(R_i) / [\sum NC(R_i)] \quad (1)$$

其中, $NC(R_i)$ 表示节点控制器的数量, $QL(R_i)$ 表示资源 R_i 中的队列负载,它是运用式(2)计算出来的。

$$QL(R_i) = \sum [NVM(J_1), NVM(J_2), \dots, NVM(J_n)] \quad (2)$$

其中, $NVM(J_1)$ 表示的是由作业 1 请求的虚拟机的数量, n 表示的是等待资源 R_i 的作业数量。

集群控制器主要是由任务调度器和本地调度器组成。集群控制器的工作流程如图 3 所示,从云控制器发送过来的作业被推送进集群控制器中的作业队列。这里的任务调度器与云控制器中的任务调度器类似,将根据作业优先级安排作业给本地调度器,然后本地调度器将按照 FIFO 算法进行调度,最后把作业分配给节点控制器中相应匹配的虚拟机。

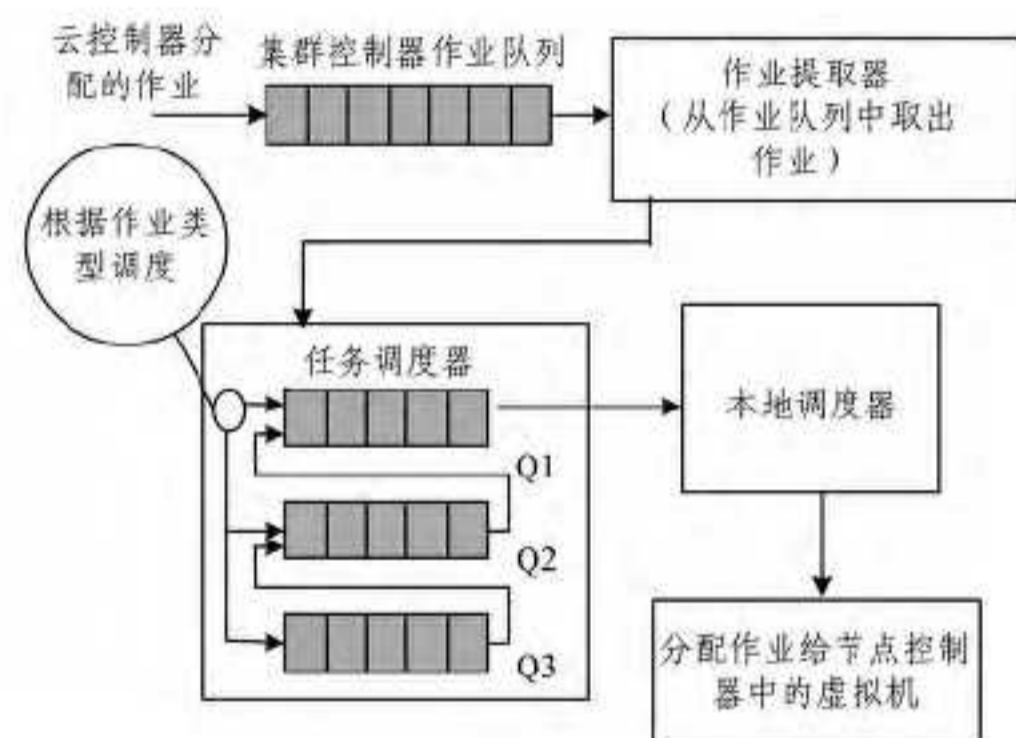


图 3 集群控制器的工作流程图

3 实验结果及分析

利用 CloudSim^[7] 云计算模拟平台来模拟作业分层优先级调度(Hierarchical Priority Scheduling, HPS)。在本模拟实验中,设置了 50 个作业,其中只有 5 个是基于截止期限的,其余 45 个没有截止期限要求,如表 1 所列。这 50 个作业提交给 5 个空闲的资源节点。如果调度策略采用现有的 FIFO 调度算法,即按照先来先服务的原则,作业 1—作业 25 提交给 5 个空闲的资源节点,按照表 1 中每个作业的执行时间,那么至少需要经过 150min 后,作业 26—作业 30 才开始执行,而这几个作业的截止期限是 120min,显然用户的服务请求都得不到满足,从而降低了用户对所使用云服务的满意程度,同样地,如果采用 SJF(Shortest Job First, 最短作业优先)调度算法,作业 26—作业 30 的用户服务请求也得不到满足,从而导致 SLA 违约。

采用文中提出的作业分层优先级调度,使用多级队列优先保证基于截止期限的作业能在规定时间内完成,同时也兼顾了短作业优先级别。实验仿真结果表明,HPS 能满足用户对作业处理时间的需求,而且与现有的 FIFO 和 SJF 调度算

(下转第 351 页)

另一方面,PE 文件作为可执行文件,文件压缩处理时会采用无损压缩,这种操作不会影响到嵌入 PE 文件冗余空间的解密指令;而且,在嵌入解密指令后,可修改 PE 文件的校验值,避免产生文件的校验错误。因此,该空间多态技术具有很好的鲁棒性。

另外,在具体的空间多态实现中,可以根据宿主 PE 文件的冗余空间情况,采取灵活的空间多态方案。如采用块嵌入,以减少跳转指令;采用选择性嵌入,解决冗余空间较小的问题。

结束语 本文在分析传统的加密、多态、变形技术的基础上,结合 PE 文件中存在冗余的特点,提出了空间多态的概念,详细阐述了空间多态技术的实现原理以及空间多态引擎的设计实现。空间多态技术弥补了传统多态技术的一些缺点,给恶意代码的自我保护提供了一种新思路。本文方案利用了 PE 文件节块之间的冗余空间,如何利用 PE 文件中的其它冗余空间实现空间多态,甚至人为构造冗余,将整个恶意代码主体嵌入宿主 PE 文件,实现空间变形技术,是值得进一步深入研究的内容。

参考文献

[1] 肖英,邹福泰. 计算机病毒及其发展趋势[J]. 计算机工程, 2011, 37(11):149-151
 [2] 吴伟民,范炜锋,王志月,等. 基于特征码的 PE 文件自动免杀策

略[J]. 计算机工程, 2012, 38(12): 118-121
 [3] 范吴平. Win32 PE 文件病毒的检测方法研究[D]. 成都: 电子科技大学, 2011
 [4] 吴丹飞,王春刚,郝兴伟. 恶意代码的变形技术研究[J]. 计算机应用与软件, 2012, 29(3): 74-77
 [5] 周梅红,刘宇峰,胡晓雯,等. 恶意代码多态变形技术的研究[J]. 计算机与数字工程, 2008, 36(10): 149-153
 [6] Holloway R. University of London. Metamorphic Virus: Analysis and Detection[R]. Konstantinou E, Wolthusen S: Royal Holloway, University of London, 2008
 [7] 王清,等. 0day 安全: 软件漏洞分析技术(第 2 版)[M]. 北京: 电子工业出版社, 2011
 [8] 汪洁,王建新,刘绪崇. 基于近邻关系特征的多态蠕虫防御方法[J]. 通信学报, 2011, 32(8): 150-158
 [9] Bashari B, Masrom M. Metamorphic Virus Detection in Portable Executables Using Opcodes Statistical Feature[C]// Proceeding of the International Conference on Advanced Science, Engineering and Information Technology. 2011
 [10] Corporation M. Microsoft Portable Executable and Common Object File Format Specification[EB/OL]. Revision 6.0, 1999, 2
 [11] 白金荣,王俊峰,赵宗渠. 基于 PE 静态结构特征的恶意软件检测方法[J]. 计算机科学, 2013, 40(1): 122-126
 [12] 段钢. 加密与解密(第 2 版)[M]. 北京: 电子工业出版社, 2006
 [13] 威利. Windows PE 权威指南[M]. 北京: 电子工业出版社, 2011

(上接第 317 页)
 法相比,其调度性能更优,作业执行时间较短,提高了用户的满意度,从而满足服务提供商和服务消费者双方协商的 SLA,如图 4 所示。

表 1 作业信息表

作业 ID	每个作业大小	截止期限	每个作业执行时间
1-10	30MB	没有	30min
11-25	20MB	没有	30min
26	30MB	120min	30min
27	20MB	120min	40min
28	30MB	120min	50min
29	20MB	120min	40min
30	30MB	120min	40min
31-50	10MB	没有	30min

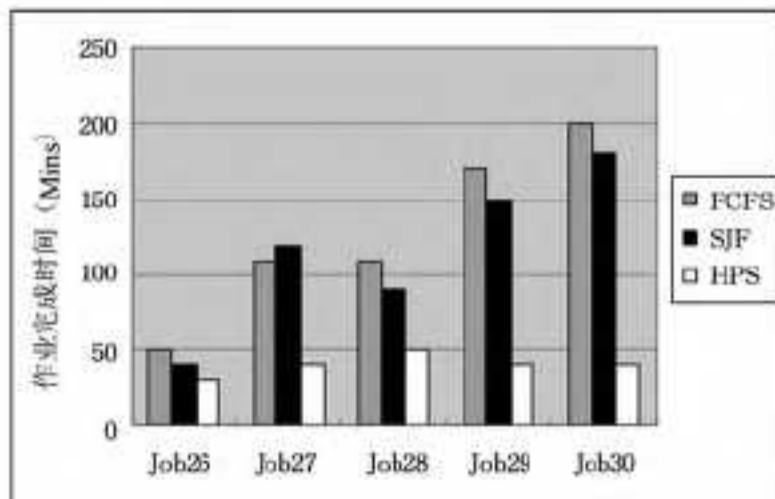


图 4 FIFO、SJF、HPS 调度算法性能分析结果

结束语 作业以分层的模式进行调度,并且使用多级队列优先保证了基于截止期限的作业能在规定时间内完成,同时也兼顾了短作业优先级别,最终作业的执行和管理是由各自节点中的节点控制器所完成。目前的实践中,只是面向 SLA 满足了用户对作业处理时间的需求。下一步的工作可

以在云控制器中集成不同的任务调度器,从而实现不同用户更多类型的服务质量需求。

参考文献

[1] Fito J O, Goiri I, Guitart J. SLA-driven Elastic Cloud Hosting Provider [C]// Proceeding of the 18th Euromicro Conference on Parallel, Distributed and Network-based Processing. Pisa, Italy, 2010: 111-118
 [2] 左利云,曹志波. 云计算中调度问题研究综述[J]. 计算机应用研究, 2012, 29(11): 4023-4027
 [3] White. Hadoop 权威指南[M]. 曾大刚,周傲英,译. 北京: 清华大学出版社, 2010
 [4] The Apache Software Foundation. Capacity Scheduler Guide [EB/OL]. http://hadoop.apache.org/docs/r1.1.1/capacity_scheduler.html, 2012-03-12
 [5] Polo J, Carrera D, Becerra Y, et al. Performance-driven task co-scheduling for MapReduce environments[C]// Network Operations and Management Symposium (NOMS). Osaka, Japan, 2010: 373-380
 [6] Kc K, Anyanwu K. Scheduling hadoop jobs to meet deadlines [C]// 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom). Indiana, USA, 2010: 388-392
 [7] Rodrigo N C, Rajiv R, Anton B. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [J]. Software: Practice and Experience (SPE), 2011, 41(1): 23-50