文章编号：2095-4980(2015)03-0503-04

云计算中基于 QoS 的调配优化算法

甘 泉，郑均辉

(平顶山学院 计算机科学与技术学院，河南 平顶山 467000)

摘 要：研究和设计云计算环境下一种优化的基于 QoS 约束的调配算法，并验证其正确性。在 QoS 需求下采用具有服务等级协议(SLA)参数的约束条件，对任务划分优先级，形成优先级队列。SLA 服务参数采用二进制权值的方法进行标记，避免了作业分级重叠的情况。在对该任务分配计算资源时，采用资源等级队列的方法，分配合理的工作节点。通过对几种算法的比较实验，验证优化算法运转的鲁棒性和正确性。

关键词：云计算；服务等级协议；负载均衡；工作队列

中图分类号：TN915.09；TP312 **文献标识码：**A **doi：**10.11805/TKYDA201503.0503

Optimized scheduling algorithm in cloud computing based on QoS

GAN Quan，ZHENG Junhui

(College of Computer Science and Technology，Pingdingshan University，Pingdingshan Henan 467000，China)

Abstract：An optimized scheduling algorithm in cloud computing environment is designed based on the QoS constraints. Under the QoS requirements with the constraint conditions marked by the SLA(Service-Level Agreement) parameters, tasks are graded to form the priority queue. SLA service parameters are marked by the binary weights to avoid overlap of the Job Level. The method of resource level queue is adopted and the work is reasonably assigned to nodes when allocating computing resources to tasks. Correctness and robustness of the optimized algorithm are verified by experiments results comparison with several algorithms.

Key words：cloud computing；Service-Level Agreement(SLA)；load balancing；work queue

在云计算平台应用的各种技术中，资源调度算法是一个很关键的技术，因为它负责把任务分配到工作节点中运行，云计算平台性能的好坏主要体现在任务分配的合理程度。现有资源调度算法很多，一些经典的算法如 Min-min 算法、遗传算法、WQR(Workqueue with Replication)算法(WQ 的改进算法)和 LATE(Longest Approximate Time to End)算法等。这些算法有各自的特点，但也存在不足。

有学者提出一个优先级工作队列资源调度算法[1]。该算法是在 WQ 算法的基础上，加入工作节点模糊评级策略。该算法用运行任务的平均完成时间对工作节点进行评级，避免了根据硬件信息进行评价的困难。工作节点选择算法通过顺序查找等级队列，在找到的等级中随机选择工作节点，然后把任务分配给工作节点运行。工作节点等级迁移算法通过同级比较和异级比较的过程，使等级队列逐渐完成。

资源调度过程中，考虑到云计算的服务质量，云计算服务提供商通过与客户签订 SLA[2]将非本地的网络资源以更具有针对性、更高速、更廉价的方式提供给用户，并根据 SLA 中所签订的内容来保证各自利益。由于云计算环境下计算中心本身的大规模性，服务器之间可能存在的异构性，节点之间可能存在负载不均的现象，这将严重威胁到机器性能，从而用户的服务质量不能得到保证，服务提供商还需根据 SLA 内容缴纳违约罚金[3]。所以在云计算环境下如何建立有效负载均衡机制，合理利用网络资源以保证 SLA，成为必须考虑的研究内容。

1 负载均衡问题分析

SLA 是关于服务供应商与客户间签订的一份合同，其中定义了服务类型、服务质量和客户付款等方面的内

收稿日期：2014-11-10；修回日期：2014-12-09
基金项目：河南省科技厅攻关项目(KJT142102210226))

容，同时也包含了关于供应商提供的服务达不到服务等级协议时，供应商应对用户进行的赔偿等内容[4]。SLA 就是服务供应商和用户之间经过协商达成的一系列的目标，其主要目的是为了达到和维持一定的服务质量(Quality of Service，QoS)。

随着云计算技术的发展，越来越多的关于云计算的应用将投入使用，所以，如何保证云计算的安全性问题也得到了广泛关注，而如何保证云计算的服务质量，也将成为云计算继续发展的重要前提。因此，在用户与云供应商签订合同时，也更加关注 SLA。在 SLA 中，需要清晰地写明能够提供的各项服务参数指标，以及服务达不到要求时的违约责任，从而更好地保证用户的权益。SLA 中规定了服务的响应时间，数据的安全保障等 QoS 的内容，以及计费付费模式等等，因此 SLA 是评判云服务的标准，也是提供商最关注的环节之一[5]。所以提高云计算的服务质量尽量降低 SLA 违约风险是研究的重要内容。但是云计算这样大规模集群式的服务器模式，以及云计算基础设施的异构性和服务类型的多样性，导致系统中有些节点处于重载，与此同时，另一些节点处在轻载甚至空闲状态，重载节点响应速度急剧下降，系统无法提供服务，SLA 违约[6]。所以，部署一个有效的负载均衡策略对于保障云计算服务 SLA 来说，是至关重要的。

2 任务队列设计与分析

2.1 优先级任务队列设计

在作业分类器中设计从用户服务等级协议信息到作业优先级的映射机制。提取服务等级协议中 4 个主要的服务参数(可用性、响应时间、资源弹性和违约罚金)的指标值。对参数的指标值进行等级分类设计，将每个服务参数分为 3 个类别，具体的分类设计如表 1 所示。

表 1 SLA 服务权值表

Table1 SLA weight values

description	parameter	scope of agreement	weight(binary)
availability: in a certain time interval, according to the prescribed conditions, the ability of cloud computing providers to provide cloud computing services	*Ava*	>95%	0011
		90%–95%	0010
		<90%	0001
response: response time of cloud computing provider to the user's various business requests	*Res*	<60 s	0011
		60 s–180 s	0010
		>180s	0001
elasticity: the biggest elastic change scope of user for cluster resource dynamic application	*Ela*	>80 users	0011
		20–80 users	0010
		<20 users	0001
penalty: liquidated damages, of which can not meet the number of all kinds of service performance or maximum times that users and suppliers agreed after consultations	*Pen*	>100 000 ¥	0011
		30 000 ¥–100 000 ¥	0010
		<30 000 ¥	0001

在任务优先级队列设计中，对文献[1]中的优先级队列设计进行了改进。文献[1]中，SLA 服务参数采用等级值{A,B,C}的方法设计，在用户服务等级协议与作业等级的映射表 MappingTable 中，定义 SLA：$Ava_value=\mathrm{A}, Res_value=\mathrm{A}, Ela_value=\mathrm{A}, Pen_value=\mathrm{A}$，其中 1~2 个为真，则 $Job\ Level=\mathrm{High}$；SLA：$Ava_value=\mathrm{B}, Res_value=\mathrm{B}, Ela_value=\mathrm{B}, Pen_value=\mathrm{B}$，其中 1~2 个为真，则 $Job\ Level=\mathrm{Low}$。笔者经过演算认为，当 SLA：$Ava_value=\mathrm{A}, Res_value=\mathrm{A}, Ela_value=\mathrm{B}, Pen_value=\mathrm{B}$，则会出现 Job Level 既属于 high 又属于 low 的情况，因此改为采用二进制权值的方法来标记参数。

在基于 SLA 的作业优先级机制中，作业 i 的优先级设为 *NICE*，其取值为：

$$NICE=Ava[i]+Res[i]+Ela[i]+Pen[i],\ NICE\in[4-12] \tag{1}$$

式中 *NICE* 值越大，优先级越高；反之，优先级越低。作业按照优先级 *NICE* 值形成优先级队列。一段时间内，若有相同 *NICE* 的作业出现，则按照先进先出(First In First Out，FIFO)的算法排列在相同 *NICE* 的作业后。

2.2 作业结构设计

任务队列是待处理的任务的 FIFO 队列，此队列中的任务是最小不可再分的，是分配给工作节点处理的最小单位。任务定义为：$T(ID_T, U_T, IData_T, OData_T, S_T, P_T)$，其属性说明如下：$ID_T$ 为任务的编号；U_T 为任务的所有者标识，通过这个所有者标识可以区分调度的优先级；$IData_T$ 为需要处理的数据，调度算法可以通过把任务分配给有本地数据的工作节点，避免了网络传输数据带来的开销，从而优化云计算平台的网络负载；$OData_T$ 为处理完成后的数据；S_T 为任务状态，云计算中的工作节点的稳定性参差不齐，因而会出现任务不能正常运行的情况，通过标示任务的状态，从而为此类任务重新调度时，可以更先调度并赋予更高性能优先级的工作节点，

以便迅速完成；P_T 为任务费用，由于云平台中不同的工作节点需要不同的处理费用，通过表示任务可付的费用，当调度时可以分配一些成本与之相适应的工作节点，这对于减少云计算平台运行成本至关重要[7]。

3　队列资源调度算法设计

3.1 主要实体数据结构

1) 工作节点

工作节点是处理任务的单位，即任务分配的对象，是计算资源和存储资源的提供者。云计算中，大量采用虚拟化技术，每一台计算机可以虚拟出一个或多个工作节点。因此实际工作节点的数量可以大大多于实体机器的数量。工作节点定义为：$W(ID_W, S_W, B_W, MaxT_W, T_W, N_W, R_W, C_W)$，其属性说明：$ID_W$ 为工作节点编号；S_W 为工作节点可以提供的存储资源的大小，作为工作节点选择的一个重要依据；B_W 为工作节点网络带宽，工作节点选择的一个重要依据；$MaxT_W$ 为工作节点同时运行的最大任务数量，不同性能的工作节点设置不同的最大任务数量，工作节点的负载也是通过使用此属性来标识；T_W 是记录在工作节点上运行的任务的总时间，此值是通过累加任务运行的时间得出；N_W 为在工作节点上运行的任务的总数量，此值是通过累加任务的数量得出，以 T_W / N_W 得出的平均运行时间来表示工作节点的快慢，从而间接地评价了工作节点的性能；R_W 为工作节点的级别，工作节点的分级是以在此工作节点上运行任务的平均完成时间为基础，平均完成时间越短，级别越高，从而更优先调度。此值只能由工作节点迁移算法更改；C_W 为工作节点的使用费用，与任务中的 P_T 配合使用，为任务的分配提供选择依据，从而达到使用成本最小的目的。

2) 等级队列

等级队列为多个工作节点分级队列，队列中的工作节点根据平均任务完成时间来分级。等级队列的属性为 $RQ(N_{RQ}, MaxW_{RQ}, Q_{RQ})$，其属性说明如下：$N_{RQ}$ 为等级队列中分级数，分级数的多少直接影响着优先级工作队列算法的性能，分级数越多，本算法的性能越好，但算法的运行时间会越长。分级数由云计算平台的管理员来设置，平台中的工作节点性能差别越大，分级数应该越多，反之越少，当分级数为一时，本算法就退化为 WQ 算法[7]；$MaxW_{RQ}$ 为每一个分级中最多包含的工作节点数，这个属性是由算法自动设置，通常每一级都分配相同数量的工作节点，以达到最好的平均性能；Q_{RQ} 为保存工作节点的队列数组，数组的大小为 N_{RQ}，每一个数组项都为工作节点的列表。本文中，默认设定最高级别为 0 级，最低级别为 $(N_{RQ}-1)$ 级。

3.2 资源调度算法

3.2.1 工作队列选择算法

调度算法把在任务队列中的任务合理地分配给工作节点，从而优化任务的平均完成时间和负载均衡[8]。工作节点的选择是指在数量众多的工作节点中选择特定数量的工作节点，为任务分配计算资源和存储资源。工作节点选择算法在待调度任务队列中选择一个任务，分配给某一个工作节点。工作节点的选择会以工作节点的分级信息和数据的存储位置等信息为依据，而工作节点的分级信息是动态变化的，因此工作节点选择算法在调度每一个任务时都要读取一次分级信息。

3.2.2 算法流程设计

调度每一个任务前，都运行一次工作节点选择算法。时间复杂度为 $O(N_{RQ})$，N_{RQ} 为分级数。算法流程为：

1) 从任务队列中获取一个任务 t，转到流程 2)；如果队列为空，则退出算法；

2) 获取存储有任务输入数据的工作节点列表 L。如果列表 L 中有工作节点 w 存在于待分配任务的工作节点分级集合中，并且 w 为待分配任务的工作节点中最高级别的，则把任务 t 分配给工作节点 w，转到 1)；如果没有，则转到 3)；

3) 从 i (i 初始为 0)级别开始获取当前级别上待分配任务的工作节点列表 La，如果 La 为空，则设 $i=i+1$，转到 4)；如果 La 不空，转到 5)；

4) 如果 i 大于 $N_{rq}-1$，则等待，直至收到待分配任务的工作节点信息，设置 $i=0$，转到 3)；

5) 从待分配任务的工作节点列表 La 中随机选择一个工作节点 Wa，把任务 t 分配给工作节点 Wa，转到 1)。

4 实验及结果分析

为验证改进的算法的有效性，进行仿真实验。在实验室中用 PC 级搭建开源云计算平台 Hadoop 集群[8]，按照前述设计要求设置好集群中主控节点及其他节点相关配置。将 LATE、WQR 与本算法进行比较。实验结果如图 1 所示。

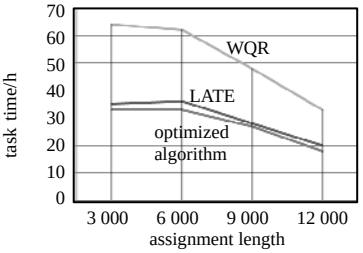

Fig.1 Comparison of algorithms experiment results
图 1 算法实验结果比较

这种算法优先把任务分配给存储有任务输入数据的工作节点，可以减少不必要的网络传输。该算法用运行任务的平均完成时间对工作节点进行评级，避免了根据硬件信息进行评价的困难。工作节点选择算法通过顺序查找等级队列，在找到的等级中随机选择工作节点，然后把任务分配给工作节点运行[9]。工作节点等级迁移算法通过同级比较和异级比较的过程，使等级队列逐渐完成。

5 结论

本文研究了在 SLA 约束下的云计算平台中，云计算任务基于约束条件的优先级队列设计和资源调度算法。在任务优先级队列设计中，对优先级队列设计进行了改进，采用二进制权值的方法来标记参数，避免了 Job Level 重叠的情况。在资源调度算法中，考虑到云计算平台的计算负荷，采用复杂度适中的算法设计，更符合云计算平台的实际情况。在任务执行过程中，面临 SLA 违约的危险时，就要采用任务迁移的方法，任务迁移算法将作为后续研究内容。

参考文献：

[1] 叶毅峰. 云环境下优先级工作队列资源调度算法研究[D]. 广州:暨南大学, 2013. (YE Yifeng. The priority work queue resource scheduling algorithm for cloud environment[D]. Guangzhou,China:Jinan University, 2013.)

[2] 王锐. 云计算环境中基于 SLA 的负载均衡机制研究[D]. 昆明:云南大学, 2012. (WANG Rui. Research on load-balancing mechanism based on SLA in cloud environment[D]. Kunming,China:Yunnan University, 2012.)

[3] 张明,李波,刘学全,等. 云计算下基于贝叶斯网络的多传感器目标识别[J]. 太赫兹科学与电子信息学报, 2014,12(5):740-744. (ZHANG Ming,LI Bo,LIU Xuequan,et al. Multi-sensor target recognition based on Bayesian classifier in cloud computing environment[J]. Journal of Terahertz Science and Electronic Information Technology, 2014,12(5):740-744.)

[4] 李建锋,彭舰. 云计算环境下基于改进遗传算法的任务调度算法[J]. 计算机应用, 2011,31(1):184-186. (LI Jianfeng, PENG Jian. Task scheduling algorithm based on improved genetic algorithm in cloud computing environment[J]. Journal of Computer Applications[J]. 2011,31(1):184-186.)

[5] 张建. 云计算服务等级协议 SLA 研究[J]. 电信网技术, 2012,2(2):7-10. (ZHANG Jian. Research on cloud computing service level agreements[J]. Telecommunications Network Technology, 2012,2(2):7-10.)

[6] LU Jing,SHAO Meifang. Relaxing data integrity in cloud computing environment[J]. International Journal of Digital Content Technology and its Applications, 2012,6(13):241-249.

[7] 郑湃,崔立真,王海洋. 云计算环境下面向数据密集型应用的数据布局策略与方法[J]. 计算机学报, 2010,33(8):1472-1480. (ZHENG Pai,CUI Lizhen,WANG Haiyang. A data placement strategy for data—intensive applications in cloud[J]. Chinese Journal of Computers, 2010,33(8):1472-1480.)

[8] Alain Tchana,Laurent Broto,Daniel Hagimont. Approaches to cloud computing fault tolerance[R]. Jeju Island:IEEE CITS 2012, 2012.

[9] CloudSim. A framework for modeling and simulation of cloud computing infrastructures and services[EB/OL]. Australia: CloudSim, (2011)[2014-11-10]. http://www.cloudbus.org/cloudsim/.

作者简介：

甘 泉(1980-)，男，安徽省灵璧县人，硕士，讲师，主要研究方向为操作系统、算法分析等.email:spring.gump@163.com.

郑均辉(1981-)，男，四川省叙永县人，硕士，讲师，主要研究方向为物联网工程.