网格资源调度算法的负载均衡及性能分析

王 琴[1,2]　曾文华[1,2]　史文翀[1,2]

(1 厦门大学软件学院，福建 厦门 361005)

(2 智能信息技术福建省重点实验室，福建 厦门 361005)

摘　要：网格系统由大量的异构资源组成，其目的是要实现资源的全面共享和协同工作，因此资源调度问题已经变得越来越重要。文章对各类经典的静态调度算法和动态调度算法进行资源调度的仿真，并对各算法的运行结果进行负载均衡和性能的比较分析。

关键词：网格，资源调度算法，负载均衡，资源释放时间

中图分类号：TP303　　**文献标识码**：A　　**文章编号**：1000-7180(2006)10-0201-03

Analysis Load Balance and Performance for Grid Resource Schedule Algorithms

WANG Qin[1,2], ZENG Wen-hua[1,2], SHI Wen-chong[1,2]

(1 School of Software, Xiamen University, Xiamen 361005, China)

(2 Key Laboratory for Intelligent Information Technology of Fujian province, Xiamen 361005, China)

Abstract: Grid system consists of a lot of heterogeneous resources, and its aim is to achieve a comprehensive resource sharing and cooperation, therefore resource schedule has become more and more important. Various classic static and dynamic schedule algorithms are integrated and simulation also is implemented for resource schedule, then the results of the algorithm implements are analyzed in the aspect of load balance and performance.

Key words: Grid, Resource schedule algorithm, Load balance, Resource release time

1　引言

一般而言，网格资源是由多个资源提供者提供的，通常面向网格资源的调度者没有为面向应用的调度提供必要的接口，以帮助实现一个面向应用的调度过程。研究并提出一个好的资源调度策略将在很大程度上提高网格资源的利用率，推动网格技术的进一步发展。

图 1 是一个简单的资源调度流程图，其中 MDS 服务(Monitoring and Discovery Service)[1]的作用是收集和发布系统状态信息，该服务主要用于获得主机节点的信息；NWS 服务 (Network Weather Service)能够周期性地监视、动态地预测各种网络性能和计算资源，并且该服务可以在给定时间内把这些信息发送出去。网格资源中的信息采集模块 MDS 先收集必要信息，然后对资源按照一定的策略进行分发。

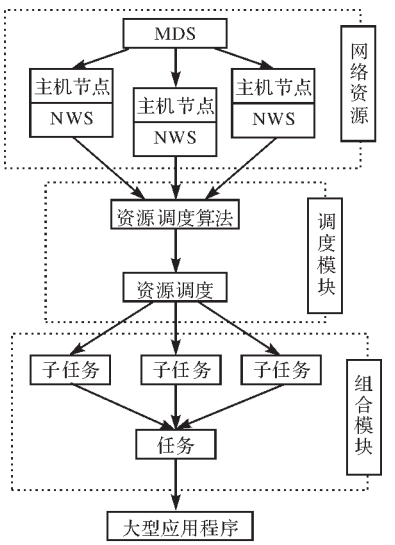

图1　资源调度流程图

2　静态算法

静态调度算法[2,3]是指所有的机器-任务映射策略在执行资源调度前就已经全部确定。网格中常见的静态调度算法有：OLB (Opportunistic Load Bal-

收稿日期：2006-04-22

基金项目：“985 工程”智能化国防信息安全技术科技创新平台项目(0000-X07204)

ancing)、MET (Minimum Execution Time)、MCT (Minimum Completion Time)、Min_Min、Max_Min、Duplex、GA(Genetic Algorithms)、SA(Simulated Annealing)、GSA、Tabu、A* 等。本文在研究实现这些算法的基础上，重点对静态的 OLB、MET、MCT、Min_Min、Max_Min 五种经典调度算法的执行结果进行了对比和分析。

OLB 算法是随机把一个可用机器分配给一个待执行任务，而不考虑任务在该机器上的资源释放时间，其目的是尽量使所有机器处于工作状态。MET 算法则以任意顺序把每一个机器分配给在其上运行具有最短资源释放时间的任务，而不考虑该机器的可用性。而 MCT 算法是将 MET 算法的执行时间(资源释放时间)用最小完成时间(机器可用时间+资源释放时间)代替。

静态 Min_Min 算法的思想是尽可能把每一个机器分配给在其上最早执行且具有最短资源释放时间的任务，是基于最小完成时间(MCT)的，该算法在每一次映射中考虑的是全部未分配的机器。而静态 Max_Min 算法的目的是为了最小化由于执行需要较长资源释放时间的任务而导致的后果。假设元任务是由许多需要短执行时间的任务和一个需要长执行时间的任务组成，这时，使用 Max_Min 算法会比使用 Min_Min 调度算法得到更好的匹配效果。

3 动态算法

动态调度算法[3,4]是指一些机器-任务映射策略在执行资源调度期间根据实际情况进行确定。现有的动态调度算法可以分为两类：在线模式(On-Line mode)和批模式(Batch mode)两种。在线模式是指任务一旦到来就启用资源调度模块将其映射到机器，该模式对每一个任务的映射只考虑一次。而在批模式下，任务一到达并不立即映射到机器，而是把任务收集起来组成一个任务集合，等映射事件到来后才启用资源调度模块对该集合中的任务进行集中映射。

3.1 线模式动态调度算法

在线模式下常见的启发式调度算法有：OLB、MCT、MET、SA(Switching Algorithm)、KPB(K-Percent Best)。其中，OLB、MCT 和 MET 三种算法的思想与静态调度算法类似，只是在资源可用时间、任务完成时间和资源释放时间等方面的计算加入了动态因素。KPB(K 最优调度算法)在映射时，只考虑将机器的一个子集(非全部机器)的某一机器映射到某一任务，该机器子集是由对该任务有最好任务执行时间的 n×k 个机器组成，其中 $1/n \le k \le 1$。然后把该子集中具有最小完成时间的机器分配给等待执行的任务。

3.2 批模式启发式调度算法

在批模式下的调度算法中，资源调度模块是每经过一个预定义的映射事件后对任务开始进行映射，映射事件的定义主要有两种方式：规则时间间隔策略和固定任务计数策略。

本文采用规则时间间隔策略。批模式下常见的调度算法有：Min_Min、Max_Min 和 Sufferage。动态的 Min_Min、Max_Min 是在静态 Min_Min、Max_Min 的基础上加入动态因子，以此来体现资源可用的动态性。Sufferage 算法的基本思想是通过分配一个特定机器给一个特定的任务(如果该机器没有匹配给该任务的话，那么这个任务将会具有更大的资源释放时间)，从而产生好的映射策略。

4 资源调度算法仿真

本文在 Linux 平台上用 JAVA 实现各类算法。采用非一致 ETC 矩阵模拟任务的运行来接近真实网格环境中的资源调度。然后取其结果的平均执行时间作为该算法在矩阵上的 release_time 值，其中 release_time 表示资源被占用后的释放时间，本文在仿真时用机器来代替资源。

4.1 静态资源调度算法性能的比较

从图 2、图 3 可知，由于 OLB 算法是随机把一个可用机器分配给一个待执行任务，因此机器负载均衡效果很好，但是因为没有考虑资源释放时间而导致其 release_time 值比较大。相反，使用 Min_Min 算法是将全部未分配的机器分配给在其上最早执行且具有最短资源释放时间的任务，因此具有最好的 release_time 但机器负载均衡效果不理想。MET 算法的目标是把每一个机器分配给具有最短资源释放时间的任务，牺牲机器负载不均衡来换取较好

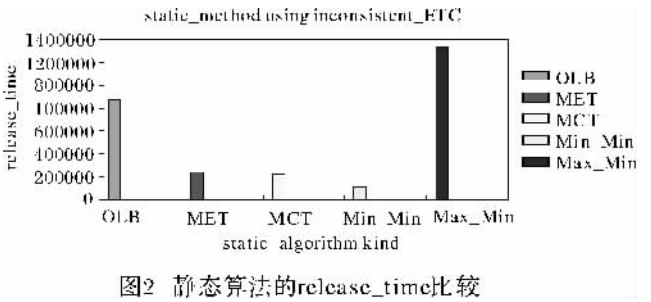

图2 静态算法的release_time比较

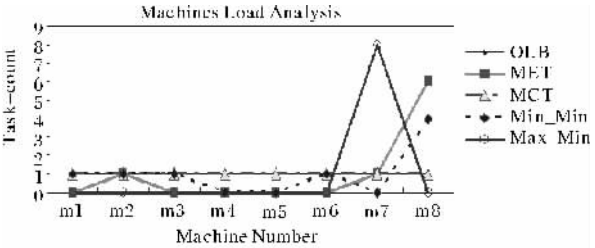

图3 静态算法的机器负载均衡效果比较

release_time 值。而 MCT 算法结合了 OLB 和 MET 的优点，机器负载均衡效果与 OLB 一样，并且由于充分考虑机器可用时间和资源释放时间，其 release_time 比 OLB 要好。Max_Min 算法的映射策略较大地改变给定资源的可用时间状态，因此该算法导致一个较大的 release_time 和不均衡的机器负载。

4.2 动态批模式资源调度算法的性能比较

从图 4、图 5 可以看出，由于 Sufferage 调度算法在确定映射策略时对某一机器如果没有分配给其第一选择的任务而带来的 release_time 的损失进行了考虑，减少了 release_time 值，所以具有较好的 release_time 和机器负载效果。动态 Min_Min 算法具有最好的 release_time 但机器负载效果最差，而使用 Max_Min 算法得到一个较大的 release_time 和不均衡的机器负载，其原因与资源调度使用静态的 Min_Min 算法、静态的 Max_Min 算法时的原因是类似的。

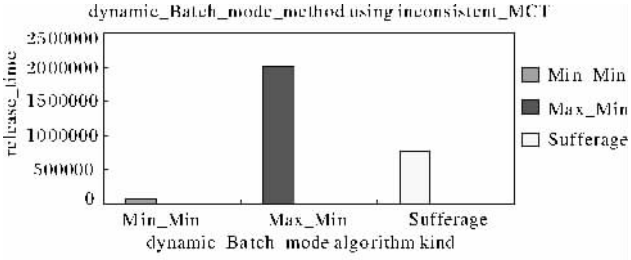

图4 动态批模式算法的release_time 比较

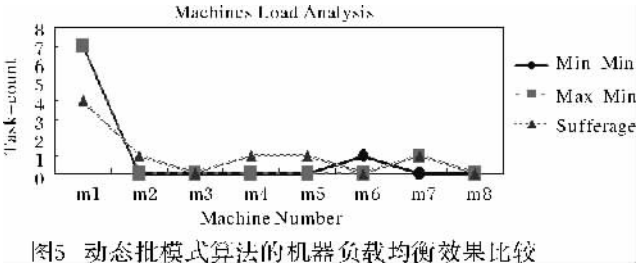

图5 动态批模式算法的机器负载均衡效果比较

4.3 动态在线模式资源调度算法的性能比较

从图 6、图 7 可知，资源调度使用四种动态在线模式算法的机器负载平衡情况及 release_time 效果的波动性不是很大。其中 KPB 算法由于每次仅考虑全部机器的一个子集来映射任务，因此该算法将这个子集中的机器映射到所有的任务上引起了负载不均衡，而该算法并不是要匹配机器给当前具有最短资源释放时间任务，这样做可以保留当前某机器以便分配给随后到达的在其上具有更短资源释放时间的任务，因此具有比较好的 release_time。

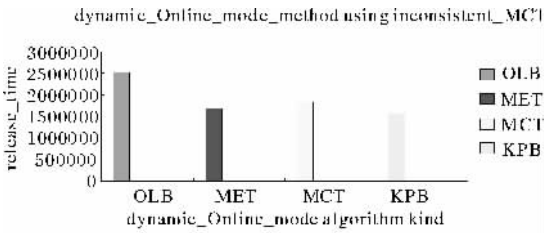

图6 动态在线模式算法的release_time比较

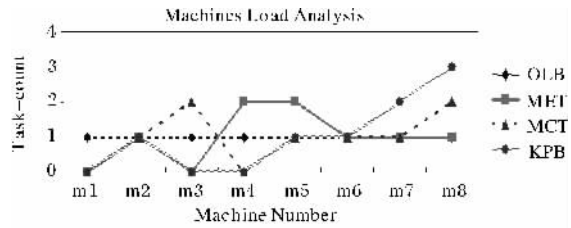

图7 动态在线模式算法的机器负载均衡效果比较

5 结束语

资源调度在网格环境中起着举足轻重的作用，由仿真结果可知，资源的负载均衡效果与资源的期望释放时间的大小是无法同时达到最好的。网格环境中，对于某一具体应用应根据具体情况来选择合适的某一算法以充分利用网格系统的处理能力，从而提高应用程序的性能。

参考文献：

[1] L Wang, H J Siegel, V P Roychowdhury, et al. Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach. Journal of Parallel and Distributed Computing. 1997, 47(1): 1~15

[2] Tracy D Braun, Howard Jay Siegel, Noah Beck. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed Computing, 2001: 810~837

[3] T Braun, H Siegel, N Beck. et al. A comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing systems. In 8th IEEE Heteroge-

(下转第 210 页)

文 Ct2 并负责本流程的事务处理，SC1 和 SC2 自动加入该事务，并通过 Ct2 与其关联。另外，SC1 和 SC2 负责各自域的协调。最后，事务结束，若事务成功完成，各个域协调器给所在域每个参与者返回成功完成的响应。在流程执行过程中如果遇到失败，则由相应的域协调器发起事务补偿。如果失败发生在付款流程，则补偿由付款流程向旅行流程传递，依次反序进行补偿，使事务恢复到一致状态。另外，在发生系统故障的情况下，事务通过事务日志进行恢复，以在系统恢复正常后继续事务的执行。

根据上述分析，可以看到通过使用加入域协调器的 WSBPEL 代替 Web 服务业务活动协调分布式域，具有许多优点：① 不必区分本地作用域和分布式作用域；② 减少代码冗余；因为补偿业务逻辑已经体现在各自作用域的域协调器中，没必要复制到调用活动；③ 协调服务不必硬编码；④ 单个活动的补偿可以由本地流程引擎外的实体触发；⑤ WS-AtomicTransaction 同样可以加入 WSBPEL，只需要在域协调器中加入原子事务规范相关消息集。

7 结束语

本文通过分析 Web 服务组合中的事务处理需求，在相关工作基础上，扩展 WSBPEL 的事务处理能力以支持流程的分布式协调，并给出了分布式协调原型系统结构，最后提出 WSBPEL 的分布式协调模型进行了分析。将来的工作将对此模型进一步进行分析和完善，并对故障恢复进行分析研究以及 Web 服务事务放宽隔离性问题等。

参考文献：

[1] OASIS WSBPEL. http: //www.oasis-open.org/ committees/wsbpel. 2006

[2] F Curbera, R Khalaf, N Mukhi. The next step in web services[J]. Communications of the ACM, 2003, 46(10)

[3] OASIS BTP. http: //oasis-open.org/ committees/business-transaction. 2004

[4] OASIS WS-TX. http: //www.oasis-open.org/ committees/ws-tx. 2006

[5] OASIS WS-CAF. http: //www .oasis-open.org/ committees/ws-caf. 2005

[6] P Sauter, I Melzer. A comparison of WS-business activity and BPEL4WS long-running transaction. KiVS 2005: 115~125

[7] W3C WSCDL. http: //www.w3.org/ TR/ws-cdl-10/. 2005

作者简介：

刘 波 男，(1981-)，硕士研究生。研究方向为 Web Services 技术。

吴家铸 男，副教授。研究方向为 Web Services 技术、视景仿真技术。

(上接第 203 页)

neous Computing Workshop, Apr. 1999: 15~29

[4] H Singh, A Youssef. Mapping and scheduling heterogeneous task graphs using genetic algorithms. In: 5th IEEE Heterogeneous Computing Workshop (HCW'96), 1996: 86~97

作者简介：

王 琴 女，(1984-)，硕士研究生。研究方向为网格资源调度算法。

曾文华 男，(1964-)，博士，教授，博士生导师。研究方向为人工智能、网格计算、嵌入式系统、计算机体系结构、智能控制。

史文翀 男，(1981-)，硕士研究生。研究方向为网格体系结构，网格资源调度。

(上接第 206 页)

出的一种满足时延、带宽等约束条件的基于遗传算法的选播 QoS 路由算法的收敛速度较快，较好地平衡了网络负载，提高了网络的利用率和服务质量。

参考文献：

[1] S Deering, R Hinden. Internet protocol version 6 (IPv6) specification. RFC 2460, Dec. 1998

[2] S Vegesna. IP 服务质量 [M]. 北京：人民邮电出版社，2001

[3] Weijia Jia, D Xuan, W Zhao. Integrated routing algorithms for anycast messages [J]. IEEE Communications Magazine, 2000, 38(1): 48~53

[4] Chor Ping Low, Choor Leng Tan. On anycast routing with bandwidth constraint [J]. Intl. Journal on Computer Communications, 2003, 26: 1541~1550

[5] B Waxman. Routing of multipoint connections [J]. IEEE J. Select. Areas Commun., 1988, 6(9): 1617~1622

作者简介：

李陶深 男，博士研究生。研究方向为网络路由算法、网络与信息安全、分布式数据库系统。