

文章编号:1001-9081(2010)11-2861-03

考虑本地作业流时的网格资源调度算法

李荣胜¹,赵文峰²,徐惠民¹

(1. 北京邮电大学信息与通信工程学院,北京100876; 2. 北京邮电大学网络与交换技术国家重点实验室,北京100876)

(lrsbnu@sohu.com)

摘要:研究了网格资源上有和没有本地作业流两种情况下两种网格资源调度算法的性能优劣对比情况。建立了一个资源的本地随机作业流模型,提出了最快处理器可用资源优先(HRARF)和最适合作业并行度可用资源优先(MSNARF)两种网格资源调度算法,并对所提出的两种算法在资源有和没有本地作业流两种情况下调度网格作业的完工时间进行仿真。仿真结果显示,在资源负载较重时,在有和没有本地作业流两种情况下,HRARF和MSNARF两种算法的性能优劣对比正好相反。在网格中,两种算法在资源共享时和资源独占时的性能优劣对比可能不同。

关键词:完工时间;本地作业;动态负载;作业调度;网格计算

中图分类号:TP393;TP301.6 **文献标志码:**A

Scheduling algorithm on grid resources in consideration of local workloads

LI Rong-sheng¹, ZHAO Wen-feng², XU Hui-min¹

(1. School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Performances of two resource scheduling algorithms on grid resources with and without stochastic local workloads were studied. A stochastic local workload model of grid resources was established. Two scheduling algorithms on grid resources, Highest CPU-Rating Available Resource First (HRARF) and Most Suitable CPU-Number Available Resource First (MSNARF), were proposed. Makespans of grid workloads scheduled by the two proposed resource scheduling algorithms with and without stochastic local workloads were simulated. The simulation results show that when the loads of resources are heavy, the relative performance of MSNARF algorithm and HRARF algorithm on grid resources with and without stochastic local workload is reverse. In grid computing, relative performance of two scheduling algorithms on sharing resources and exclusive resources may be different.

Key words: makespan; local workload; dynamic load; job scheduling; grid computing

0 引言

网格是一个大规模、跨多个管理域、多用户的计算环境。网格资源分属于不同的组织或个人,被本地用户和网格用户共享,网格资源上同时运行着本地作业和网格作业^[1-3]。本地作业由本地作业调度系统调度,网格作业由网格作业调度系统调度。当网格调度系统把作业分派给某个资源时,该资源可能正在执行本地作业;资源在执行网格作业的过程中,本地作业随机到达。所以,网格作业调度系统面对的是负载动态变化的资源。关于网格作业调度已有大量的研究工作,研究的重点主要集中在网格作业集或者作业流的完工时间的优化、系统吞吐量的优化、网格资源的负载均衡等方面,主要考虑网格作业独占网格资源的情况。然而,考虑本地作业流与网格作业流竞争使用资源情况下的网格作业调度方面的研究文献还很少。文献[4]用马尔可夫链来预测资源的可用处理能力;文献[5]研究了在个人计算机组成的网格中,通过收集各个资源的空闲时间片来调度网格作业。

本文研究在网格资源有本地随机作业流(即资源负载动态变化情况下的随机网格作业流)完工时间的优化问题。首先,描述了网格随机作业流和网格资源,建立了一个资源本地的随机作业流模型,描述了网格作业调度问题;其次,针对到

达时间、计算量、并行度随机变化的网格作业流,提出最快处理器可用资源优先(Highest CPU-Rating Available Resource First, HRARF)和最适合作业并行度可用资源优先(Most Suitable CPU-Number Available Resource First, MSNARF)两种网格资源调度算法;最后,结合作业优先策略和所提的两种资源调度算法,在无本地作业流、有本地作业流且作业以不同的时间间隔到达情况下,对包含不同作业数量的网格作业流的完工时间进行仿真。

1 网格作业调度模型

1.1 网格作业流描述

设随机网格作业流如式(1)所示:

$$J = \{J_1, J_2, \dots, J_i, \dots, J_{n-1}, J_n\} \quad (1)$$

作业 J_i 用四元组来表示为:

$$J_i = \langle id_j, a_j, w_j, cpu_r_j \rangle \quad (2)$$

其中: id_j 表示作业 J_i 的编号; a_j 表示网格系统接收作业的时间,是某一时刻; $w_j (> 0)$ 表示作业的计算量; cpu_r_j 是一个正整数,表示作业的并行度。 a_j, w_j, cpu_r_j 都是随机量。

1.2 资源本地作业流模型

网格资源在执行网格作业的同时,还执行本地作业。作业类型不同,作业的各项属性也不一样。本文假设本地作业的到

收稿日期:2010-05-09;修回日期:2010-07-17。

基金项目:国家973计划项目(2007CB307103);贵州省重大科技专项计划项目(黔科合重大专项字[2007]6017)。

作者简介:李荣胜(1975-),男(壮族),广西大化人,博士研究生,主要研究方向:网格作业调度;赵文峰(1980-),男,山西夏县人,博士研究生,主要研究方向:语义Web、Web服务;徐惠民(1941-),男,上海人,教授,博士生导师,主要研究方向:网格计算。

达时间、计算量、并行度都是服从均匀分布的随机变量,即它们的概率密度函数具有式(3)的形式^[6]:

$$f_x(x) = \begin{cases} \frac{1}{b-a}, & 0 < a < x < b \\ 0, & \text{其他} \end{cases} \quad (3)$$

并假设有 m 个资源 R_i ($1 \leq i \leq m$),每一个资源都有一个本地作业流。资源 R_i 的本地作业的到达时间服从均匀分布 $U(a_i^b, a_i^e)$;计算量服从均匀分布 $U(w_i^l, w_i^h)$;并行度(正整数)服从均匀分布 $U(p_i^l, p_i^h)$ 。本文不考虑作业的内存大小要求、存储空间大小要求等属性。

1.3 网格资源描述

网格环境由 m 个资源组成,每个资源包含一到多台机器,每台机器包含若干个相同的 CPU。设资源的带宽、存储能力以及内存足够大,不影响作业的执行,则网格环境可以描述为:

$$\begin{aligned} Grid &= \{Grid_resource_{\xi} \mid \xi \in \mathbb{N}_+\} \\ Grid_resource_{\xi} &= \{Computer_{\eta} \mid \eta \in \mathbb{N}_+\} \\ Computer_{\eta} &= \{CPU_{\zeta} \mid \zeta \in \mathbb{N}_+\} \end{aligned} \quad (4)$$

其中 $1 \leq \xi \leq m$ 。

1.4 网格作业调度模型

本文研究式(1)和(2)描述的随机网格作业流在式(4)描述的网格环境中执行时完工时间的优化问题,同时,考虑资源有到达时间、计算量、并行度具有式(3)形式的本地作业流。

2 资源调度算法

网格作业调度是把 n 个作业分派给 m 个资源执行的过程,包括两个阶段:一个阶段是确定分派作业的顺序,如先来先服务(First Come First Served, FCFS)、最短作业优先(Shortest Job First, SJF)、最长作业优先(Longest Job First, LJF)等;另一个阶段是选择资源的优先顺序。

因为资源有本地作业流,对于网格调度系统而言,资源的负载动态变化,资源并不总是可用的。另外,网格作业有并行度要求,一个资源上必须有足够的空闲 CPU 才能执行一个作业。为了优化网格作业的完工时间和提高资源的利用率,本文提出了两种资源调度算法:HRARF 算法和 MSNARF 算法。

2.1 HRARF 算法

网格作业动态到达网格调度系统,这些作业可能属于不同的网格用户。网格用户只关心自己的作业的完成时间,不关心整个网格作业流的完成时间。从这个角度出发,本文提出了 HRARF 算法。其具体步骤如下。

1) 对网格资源进行排序。排序规则是:优先按 CPU 速度降序;CPU 速度相同时,按 CPU 数目降序。

2) 作业队列中的作业按某种优先策略排序。

3) while(还有作业提交或者作业队列 Q 非空)

① 读取下一个事件 e 。

② 如果 e 是新作业到达,按照 2) 中的优先策略加入 Q ,并调用 $schedule()$ 。

③ 否则如果 e 是一个作业执行完毕,调用 $schedule()$ 。

其中 $schedule()$ 内容为:

4) while(作业队列 Q 非空)

① 从 Q 取出队首作业 gl ,在资源队列中顺序寻找能立即执行 gl 的资源。

② 如果找到,把 gl 分派给该资源,并从 Q 中删除 gl 。

③ 否则,返回。

2.2 MSNARF 算法

因为作业有并行度要求(即同时在多个 CPU 上运行),所

以在调度作业时,只考虑资源的速度而不考虑资源的 CPU 数量,容易形成 CPU 碎片(运行当前作业后,剩余的 CPU 数量达不到下一个作业并行度的要求而闲置)。这是 HRARF 算法的缺点。为克服这个缺点提出了 MSNARF 算法,其原理是:优先把作业分派给 CPU 数量正好等于作业并行度的可用资源,当 CPU 数量正好等于作业并行度的资源不可用时,才把作业分派给 CPU 数量大于作业并行度的资源。该算法具体步骤如下。

1) 网格资源按 CPU 数 n_{CPU} 进行分类,把具有相同 CPU 数的资源放入同一个资源队列 R_queue_k 中, $k = n_{CPU}$ 。资源队列 R_queue_k 中的资源按 CPU 速度降序排列。

2) 作业队列中的作业按某种优先策略排序。

3) while(还有作业提交或者作业队列 Q 非空)

① 读取下一个事件 e 。

② 如果 e 是新作业到达,按照 2) 中的优先策略加入 Q ,并调用 $schedule()$ 。

③ 否则如果 e 是一个作业执行完毕,调用 $schedule()$ 。

其中 $schedule()$ 内容为:

4) while(作业队列 Q 非空)

① 从 Q 取出队首作业 gl ,根据其并行度 cpu_r_j 要求,先在 $k = cpu_r_j$ 的资源队列 R_queue_k 中顺序寻找能立即执行 gl 的资源。

② 如果找到,把 gl 分派给该资源,并从 Q 中删除 gl 。

③ 否则,按 A 升序依次在 $k > cpu_r_j$ 的资源队列 R_queue_k 中顺序寻找能立即执行 gl 的资源。

④ 如果找到,执行 4) 中的 ②。

⑤ 否则,返回。

3 仿真设计及结果

3.1 仿真设计

关于网格资源的假设:在仿真过程中,资源不退出网格且不出故障。

关于本地作业流的假设: m 个本地作业流的作业到达时间服从同一个均匀分布 $U(a^b, a^e)$,计算量服从同一个均匀分布 $U(w^l, w^h)$,作业的并行度(正整数)服从均匀分布 $U(p^l, p^h)$, $1 \leq i \leq m$ 。

在仿真中,取 $U(a^b, a^e) = U(0, 30000)$ (0 为仿真开始时刻); $U(w^l, w^h) = U(80000, 240000)$,单位为 MI(百万指令); $U(p^l, p^h) = U(0, R_i$ 的 CPU 数 + 1)。

本文在 GridSim^[7] 和 ALEA^[8] 软件包之上仿真上述的网格作业调度。网格作业流和网格资源分别使用了 ALEA 软件包中的 pisa.pwf 和 pisa.pwf.machines 文件。pisa.pwf 文件中的作业动态到达网格调度系统,计算量均匀分布,并行度在区间 [1, 8] 上均匀取整数值。pisa.pwf.machines 文件中有 150 个机群共 1416 个 CPU,每个机群的 CPU 数是 2^s , s 在区间 [1, 4] 上均匀取整数值,不同机群的 CPU 速度在区间 [344, 599] 上均匀取整数值,单位为 MIPS。

因为网格作业动态到达调度系统,所以,在仿真中,网格作业队列使用 FCFS 优先策略。资源调度使用本文提出的 HRARF 算法和 MSNARF 算法。

在仿真中,每个资源的本地作业流实现了软件包 GridSim 5.0 beta^[9] 中的接口 gridsim.parallel.util.WorkloadModel,本地作业调度系统按 FCFS 调度本地作业。本地作业和网格作业被分配给某个资源后,该资源按 FCFS 策略执行这些作业。

3.2 仿真结果及分析

在没有本地作业时,HRARF 算法和 MSNARF 算法的网

格作业流完工时间与作业数量的关系如图 1 所示。

每个资源的本地作业流分别包含 10, 20, 30 个作业时, 每种情况重复仿真 10 次取算术平均值, HRARF 算法和 MSNARF 算法的网格作业流完工时间与作业数量的关系如图 2 所示。

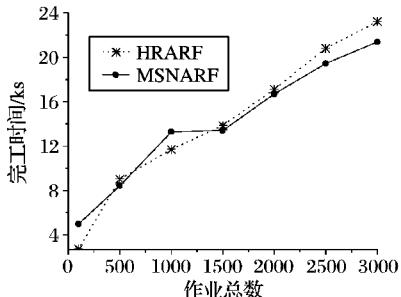


图 1 无本地作业流时网格作业流的完工时间与网格作业数量的关系

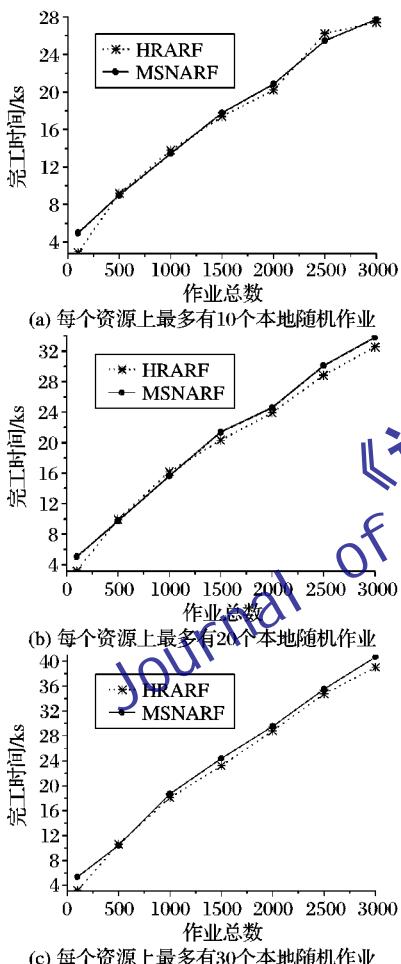


图 2 有本地作业流时网格作业流的完工时间与网格作业数量的关系

图 1 显示, 在资源没有本地作业流的情况下, 在资源负载较轻即网格作业数小于 1500 时, HRARF 算法优于 MSNARF 算法; 在资源负载较重即网格作业数大于 1500 时, MSNARF 算法优于 HRARF 算法。这是因为, HRARF 算法以 CPU 最快的资源优先, 在资源负载低时充分利用 CPU 快的资源, 在资源负载高时容易造成 CPU 碎片, 资源利用率低; MSNARF 算法以 CPU 数最接近作业并行度的资源优先, 在资源负载低时没有充分利用 CPU 快的资源, 在资源负载高时 CPU 碎片少, 资源利用率高。

图 2 显示, 在资源有本地作业流的情况下, 在资源负载较轻即网格作业数小于 500 时, HRARF 算法优于 MSNARF 算法; 在资源负载居中即图 2(a) 中网格作业数大于 500,

图 2(b) 中网格作业数在 500 至 1250、图 2(c) 中网格作业数在 500 至 1000 时, HRARF 和 MSNARF 两种算法性能的差别不明显; 在资源负载较重即图 2(b) 中网格作业数大于 1250, 图 2(c) 中网格作业数大于 1000 时, HRARF 算法优于 MSNARF 算法。

值得注意的是, 当资源的负载较重时, 在资源没有和有本地作业流两种情况下, HRARF 算法和 MSNARF 算法的性能优劣对比正好相反: 在资源没有本地作业流的情况下, MSNARF 算法优于 HRARF 算法; 而在资源有本地作业流的情况下, HRARF 算法优于 MSNARF 算法。这是因为, 在没有本地作业时, 网格作业独占资源, MSNARF 算法比 HRARF 算法的资源利用率高; 在资源有本地随机作业流时, 网格作业调度系统和本地作业调度系统在竞争使用资源, 此时, 如果网格作业调度系统不抢先占用 CPU 最快的资源, 则本地作业调度系统会占用, 故 HRARF 算法优于 MSNARF 算法。

4 结语

本文研究了在资源有本地作业流情况下的网格作业流完工时间的优化问题。本地作业调度系统以 FCFS 策略调度本地随机作业流, 网格调度系统以 FCFS 策略管理网格作业队列、以 HRARF 和 MSNARF 算法为队首作业分配资源, 本地作业和网格作业被分配给某个资源后, 该资源以 FCFS 策略执行这些作业。仿真结果显示, 在资源负载较重时, 资源无本地作业和有本地随机作业流两种情况下 HRARF 和 MSNARF 算法的性能优劣对比正好相反。下一步, 将研究其他算法在资源有无本地作业流两种情况下的性能优劣对比。

参考文献:

- [1] ANDERSON D P. BOINC: a system for public-resource computing and storage [C]// The Fifth IEEE/ACM International Workshop on Grid Computing. Washington, DC: IEEE Computer Society, 2004: 4–10.
- [2] BIRD I, ROBERTSON L, SHIERS J. Deploying the LHC computing grid — the LCG service challenges [C]// 2005 IEEE International Symposium on Mass Storage Systems and Technology. Washington, DC: IEEE Computer Society, 2005: 160–165.
- [3] CERN. WLCC worldwide LHC computing grid [EB/OL]. [2010-04-20]. <http://lcg.web.cern.ch/lcg/public/>.
- [4] DERBAL Y M. Decentralized probabilistic scheduling: application to computational grids [C]// Proceedings of the 2005 IEEE Conference on Control Applications. Washington, DC: IEEE, 2005: 1044–1050.
- [5] CELAYA J, ARRONATEGUI U. Distributed scheduler of workflows with deadlines in a P2P desktop grid [C]// Proceedings of the 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing. Washington, DC: IEEE Computer Society, 2010: 69–73.
- [6] 盛骤, 谢式千, 潘承毅. 概率论与数理统计 [M]. 4 版. 北京: 高等教育出版社, 2008: 44–45.
- [7] BUYYA R, MURSHED M. GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing [J]. The Journal of Concurrency and Computation: Practice and Experience, 2002, 14(13/15): 1175–1220.
- [8] KLUSACEK D. Alea2 — GridSim based grid scheduling simulator [EB/OL]. [2010-04-03]. <http://www.fi.muni.cz/~xklusac/alea/>.
- [9] The CLOUDS Laboratory. The CLOUDS lab: flagship projects — Gridbus and Cloudbus [EB/OL]. [2010-07-06]. <http://www.gridbus.org/gridsim/>.