

将参数扫描应用分布于网格的负载平衡方法

王庆江, 刘培顺, 高 云

(中国海洋大学计算机科学系, 山东青岛 266100)

摘 要: 为降低高性能计算网格中参数扫描应用的响应时间, 基于负载平衡确定各网格节点分得任务的数量. 用处理机需求与处理机数的比值表示网格节点的作业负载, 用作业负载的均方差表示网格负载失衡度, 并按降低网格负载失衡度确定参数扫描应用的任务分配. 当参数应用的任务数不足以实现网格负载失衡时, 选择作业负载较低的一组节点, 并按负载平衡将该参数应用分布于这些节点. 实验表明, 基于负载平衡的调度可有效降低参数扫描应用的响应时间.

关键词: 计算网格; 参数扫描应用; 作业负载; 网格负载平衡

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2007) 11-2211-04

A Load Balancing Method for Distributing Parameter Sweep Applications on Grids

WANG Qing-jiang, LIU Pei-shun, GAO Yun

(Department of Computer Science, Ocean University of China, Qingdao, Shandong 266100, China)

Abstract: To reduce the response time of a parameter sweep application on high-performance computational grids, the number of tasks allocated to each grid node is determined based on load balancing. Job load of each grid node is represented by the ratio of processor requirements to processor number, the grid load imbalance degree is represented by the mean square deviation of job loads, and the task allocations of a parameter sweep application are determined by reducing the grid load imbalance degree. When the task number of a parameter application is not adequate to implement the grid load balancing, a set of nodes with lower job loads are selected, and the parameter application will be distributed on these nodes according to load balancing. Experiments show the load balancing-based scheduling can effectively reduce the response time of parameter sweep applications.

Key words: computational grid; parameter sweep application; job load; grid load balancing

1 引言

计算网格可汇聚地理分布的多个高端计算系统(又称网格节点), 有效执行计算密集型应用. 在流体动力学、生物信息学、粒子物理等领域, 参数扫描是一种重要的研究手段. 参数扫描应用(简称参数应用)对多维参数空间上的每一组参数完成指定的计算, 任务数巨大且任务间没有数据依赖, 非常适合在计算网格上运行.

Nimrod^[1]根据用户描述的应用特征, 将任务指派到负载较低的工作站, 并收集各任务的计算结果, 所采用的负载平衡方法主要适用于机群或工作站网络, 不能用于高性能计算网格, 因为后者的节点一般被建模为多处理机系统. Nimrod/G^[2]是网格环境下的 Nimrod, 包含资源的发现、获取、监视等机制, 根据参数扫描应用的完成期限、用户可支付费用、资源使用价格等确定任务指派. 文献[3]将并行化的三维图像重构分布在工作站群和超级计算机的空闲节点, 用计算网格加快三维图像重构的过程. 三维图像基于二维图像构建, 而二维图像是围绕单轴在不同角度下投影获得的, 故构建三维图像属于一维参数空间上的参数扫描应用. 时间共享的工作站总是可立即执行分配的计算请求, 而提交到空间共享的超级

计算机时, 计算请求的等待时间却往往难以预测, 文献[3]基于应用的性能预测自适应地只向能立即执行计算请求的超级计算机提交请求, 这种自适应分配方法可取得比单独在工作站或单独在超级计算机上执行时更短的响应时间. 文献[4]针对网格的动态和异构本质, 提出了调度参数扫描应用的异构自适应方法, 包括自动适应网格的动态特征、自动将正在运行的作业迁移到更好的资源、重用任务间共同的数据文件等. 自适应调度分为两个阶段, 先将参数应用的任务指派到当前发现的节点, 然后根据各节点作业取消或失败事件, 以及新发现计算节点等资源动态信息, 重新安排参数应用的任务指派, 即迁移某些任务, 这可进一步改善参数应用的执行. XSufferage^[5]算法考虑了参数扫描应用中任务的数据文件共享, 基于时空图确定任务到主机的映射, 并评估不精确的性能预测对调度的影响. 文献[6]将参数扫描应用的任务分成若干批, 每一批任务分布在各节点上运行, 根据各节点上任务执行情况确定下一批任务的数量和指派, 分批调度使参数扫描应用的执行可适应各节点的计算能力差异和波动.

与上述研究不同, 这里从多作业共享节点(或网格)的角度, 论述网格负载平衡, 并基于负载平衡确定参数

扫描应用的任务分配.

2 基于负载平衡的任务分配

网格作业可分为单节点和多节点作业^[7].单节点作业的运行局限于一个网格节点,多节点作业的运行跨越节点边界.为缩短响应时间,参数扫描应用应分布在多个节点运行,属于多节点作业.如不明确指出,下文中的作业指单节点作业或参数扫描应用中的任务.

假设网格节点分别为 $A_1, A_2, \dots, A_N$, A_j 的处理机数为 $\lambda_j, j \in [1, N]$. 对于同一个并行程序,不同节点会表现出不等的并行计算能力,使程序的运行时间不同.不同的并行程序可能有不同的并行性,使节点的并行计算能力表现为一个波动变化的值, NPB^[8]测试值是节点并行计算能力的粗略表示.为简单起见,假设 A_j 的并行计算能力为 c_j , 并行计算能力与作业运行时间成反比.若作业 J_i 在 A_j 上的运行时间为 t_{ij}^e , 上标 e 代表“运行”(execution), 则 $c_j \times t_{ij}^e$ 称为 J_i 的纯运行时间, 记为 l_i , 表示 J_i 不依赖指派的“纯计算需求”. J_i 在 A_j 上的处理机需求^[9]可表示为 $w_i \times \frac{l_i}{c_j}$, w_i 为 J_i 的并行度.在时空图中运行时间和并行度分别为长和宽,故一般用 l 代表运行时间, w 代表并行度.

定义 1 作业负载

在时刻 T , A_j 上作业的处理机需求 γ_j 可用式(1)计算, 这里, $J_i \rightarrow A_j$ 表示 J_i 在 A_j 上等待或运行, t_{ij}^s 是 A_j 调度 J_i 开始运行的时刻, 上标 s 代表“开始”(start). $t_{ij}^s \geq T$ 表示 J_i 尚未运行, $t_{ij}^s < T$ 表示 J_i 已在运行.

$$\gamma_j = \sum_{J_i \rightarrow A_j, t_{ij}^s \geq T} \left(w_i \times \frac{l_i}{c_j} \right) + \sum_{J_i \rightarrow A_j, t_{ij}^s < T} \left(w_i \times \frac{l_i}{c_j} - T + t_{ij}^s \right) \quad (1)$$

若 A_j 的处理机数为 λ_j , 则 $\frac{\gamma_j}{\lambda_j}$ 称作 A_j 的作业负载, 记为 ρ_j .

定义 2 网格负载失衡度

假设网格节点的作业负载集合为 $\{\rho_j | j \in [1, N]\}$, $\bar{\rho} = \frac{1}{N} \sum_{j=1}^N \rho_j$, 则 $\{\rho_j | j \in [1, N]\}$ 对 $\bar{\rho}$ 的方差称作网格负载失衡度, 记为 δ .

$\delta = 0$ 表示实现了整个网格的负载平衡, 简称网格负载平衡.

在网格上调度参数扫描应用, 就是确定各节点应分得任务的数目.

定义 3 任务分配向量

假设网格包含 N 个节点, 分别为 $A_1, A_2, \dots, A_N$, 参数扫描应用分配在 A_j 的任务数为 φ_j , 则 $(\varphi_1, \varphi_2, \dots, \varphi_N)$ 称作任务分配向量, 记为 Φ .

假设参数应用的任务数为 M , 任务并行度和纯运行时间分别为 w 和 l . A_j 上一个任务的处理机需求可表示为 $w \times \frac{l}{c_j}$. 若 A_j 分得 φ_j 个任务后的作业负载为 ρ_j' , 则 ρ_j' 可用式(2)计算.

$$\rho_j' = \rho_j + \varphi_j \times w \times \frac{l}{c_j} \div \lambda_j \quad (2)$$

假设 $\{\rho_1', \rho_2', \dots, \rho_N'\}$ 上计算的网格负载失衡度为 δ' . 要使 $\delta' = 0$, 不难推得式(3).

$$\varphi_i = c_i \times \lambda_i \times \left(\frac{\varphi_N}{c_N \lambda_N} - \frac{\rho_i - \rho_N}{w \times l} \right), \quad i \in [1, N-1] \quad (3)$$

由式(3)和约束条件 $\sum_{j=1}^N \varphi_j = M$, 可求出一组 $\varphi_j (j \in [1, N])$. 按四舍五入将 $\varphi_j (j \in [1, N-1])$ 变为整数, 再令 $\varphi_N = M - \sum_{j=1}^{N-1} \varphi_j$, 得到的 $(\varphi_1, \varphi_2, \dots, \varphi_N)$ 可近似实现网格负载平衡.

令 $\rho^{\max} = \max\{\rho_j | j \in [1, N-1]\}$, 由式(3)和 $\sum_{j=1}^N \varphi_j = M$ 可推得式(4), 这是式(3)有解的必要条件.

$$M \geq \sum_{j=1}^N c_j \times \lambda_j \times \frac{\rho^{\max} - \rho_j}{w \times l} \quad (4)$$

当式(4)不满足时, 从整个网格开始, 不断剔除作业负载最大的节点, 直到剩余的 X 个节点满足 $M \geq \sum_{i=1}^X$

$c_i \times \lambda_i \times \frac{\rho^{\max} - \rho_i}{w \times l}$, 这里, c_i, λ_i 和 ρ_i 分别为剩余节点中第 i 个节点的并行计算能力、处理机数和作业负载, $\rho^{\max}$ 为剩余节点中的作业负载最大值.

若最后得到的节点集为 R , 则向 R 中的节点分配任务, 可使这些节点的任务负载近似相等. 对于 $\forall R' \supset R$, R' 上不满足式(4), 故无论怎样分配任务, 都无法实现 R' 上的负载平衡(即 R' 中节点的任务负载相等). 而对于 $\forall R' \subset R$ 且 $R' \neq \Phi$, R' 上满足式(4), 故任务分配可实现 R' 上的负载平衡. 因此, R 是调度参数扫描应用时可实现负载平衡的最大的节点集, 且集合中的节点是从作业负载最低开始选取的.

3 实验

3.1 实验环境

JSSS^[10]是基于离散事件的调度模拟系统, 可模拟网络拓扑、资源相对性能、调度框架、网格层次和本地层次调度、网格工作负荷(workload)等, 可模拟单节点作业和参数扫描应用并存的情况.

假设网格包含 16 个节点(即 $N = 16$), 各节点都是高端计算系统, 本地调度采用装填法^[9]. 各节点的处理机数往往不等, 不妨令 $A_1 \sim A_{16}$ 的处理机数分别为 64、32、64、64、32、32、64、128、128、32、32、64、32、64、128 和

128. 各节点的并行计算能力往往不等,不妨令 $c_j = \frac{1}{j}$, $j \in [1, 16]$.

在用户提交一个参数扫描应用时,网格中会有多个单节点作业等待或运行. JSSS 包含一个工作负荷模型,可产生单节点作业的流. 要调整单节点作业形成的网格负载,可在各作业到达间隔上统一乘以一个因子,该因子称为到达间隔扩张因子,记为 ξ . ξ 越大,网格负载越轻.

通常,参数扫描应用中任务的并行度和纯运行时间较小,不妨令 $w = 4$, $l = 5$. 为使网格负载处于通常的状态,参数应用的提交不应过早,不妨令提交 J_{500} 时提交一个参数应用.

若不考虑网格负载,即隐含地假设调度参数应用前, $\rho_i = \rho_j$, 这里 $i, j \in [1, N]$ 且 $i \neq j$, 则式(3)可改写为式(5).

$$\varphi_i = c_i \times \lambda_i \times \frac{\varphi_N}{c_N \lambda_N}, \quad i \in [1, N-1] \quad (5)$$

由式(5)和 $\sum_{j=1}^N \varphi_j = M$ 可得, $\phi = \frac{M}{\sum_{j=1}^N c_j \lambda_j} \times (c_1 \lambda_1,$

$c_2 \lambda_2, \dots, c_N \lambda_N)$, 即 ϕ 仅依赖各节点的并行计算能力和处理机数,这种调度方法记为 Y. 若考虑网格负载,就按第 2 节提出的基于负载平衡的方法调度参数应用,这种调度方法记为 B. 参数应用调度前后的网格负载失衡度分别记为 δ 和 δ' .

3.2 结果分析

令到达间隔扩张因子 ξ 分别为 1, 0.5 和 0.25, 即网格负载由轻变重,在调度参数应用前各节点的作业负载如图 1. 由图 1, 当网格负载轻时,只有 A_1 和 A_2 上有单节点作业等待或运行;当网格负载适中或重时,多数节点的作业负载非零. 此外,计算能力较强或处理机较多的节点有较重的作业负载,这是单节点作业的节点选择采用了优化策略的结果. 图 1 说明,当网格负载不是很轻时,网格负载一般是不平衡的.

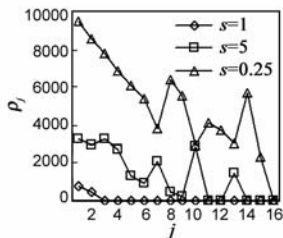


图1 各节点的作业负载

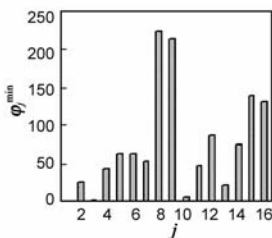


图2 各节点的最低任务数

要实现网格负载平衡,由式(4), A_j 至少应接纳一定数目的任务,该值记为 $\varphi_j^{\min}$. 以 $\xi = 0.5$ 为例,各节点至少需接纳的任务数如图 2. 由图 2, $\varphi_j^{\min}$ 与 A_j 的并行计算能力、处理机数、当前的作业负载有关,并行计算能

力较强、处理机数较大或当前的作业负载较轻,都会使 $\varphi_j^{\min}$ 较大. 这里, $\sum_{j=1}^N \varphi_j^{\min} = 1196$, 即 $M \geq 1196$ 时,参数应用的任务分配才可能实现整个网格的负载平衡;否则,按第 2 节方法选择一组节点,在这组节点上基于负载平衡调度参数应用.

令 $\xi = 0.5$, 当 M 分别为 512、1024 和 2048 时,按方法 B 确定各节点分得任务的数目如图 3, 各节点的作业负载如图 4. 由图 3, 当 $M = 512$ 时, $R = \{A_5, A_6, A_8, A_9, A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}\}$, 其他六个节点不参与运行参数应用;当 $M = 1024$ 时,除 A_1 和 A_3 外的其他 12 个节点都参与运行参数应用;当 $M = 2048$ 时,所有节点参与运行参数应用. 由图 4, $M < 1196$ 时,方法 B 不能实现整个网格上的负载平衡,只能使作业负载较低的一些节点实现负载平衡; $M \geq 1196$ 时,方法 B 可实现整个网格上的负载平衡,且调度后各节点的作业负载大于调度前作业负载的最大值.

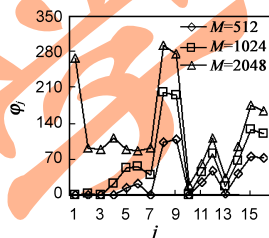


图3 方法B下各节点分得任务的数目

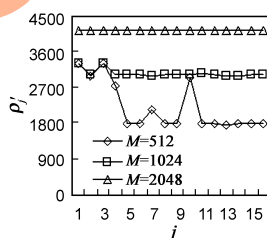


图4 方法B下各节点的作业负载

令 $\xi = 0.5$, M 分别为 512、1024 和 2048. 按方法 Y 调度参数应用后,各节点的作业负载如图 5, 方法 Y 和 B 在参数应用的响应时间 (t' , 上标 r 代表“响应”(response))和网格负载失衡度 (δ') 上的比较如图 6. 由图 5 和 6, 方法 Y 只根据并行计算能力和处理机数确定 ϕ , 没有考虑各节点的已有负载,故不会改变 ρ'_j ($j \in [1, 16]$) 之间的差别,即 δ' 不变. 采用方法 B 时, M 越大, R 越大,将有越多节点的作业负载相等,故 δ' 下降. 此外,与方法 Y 相比,方法 B 可使参数应用的响应时间更短.

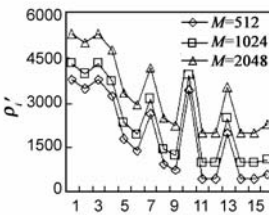


图5 方法Y下各节点的作业负载

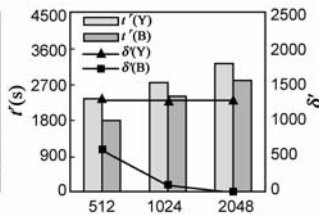


图6 在响应时间和网格负载失衡度上的调度比较

4 结论

为实现参数扫描应用在网格平台上的高性能执行,这里研究并提出了基于负载平衡的任务分布方法.

与机群负载平衡不同,高性能计算网格的负载平衡必须考虑节点是多处理机的事实,因此这里用处理机需求与处理机数的比值表示节点的作业负载.在基于负载平衡将参数扫描应用的任务分布于网格时,应该先分析单个任务的计算需求,以及任务总数是否足以实现整个网格上的负载平衡.如果任务数不足,则按作业负载由低到高,选取一个节点集,然后在这个节点集上基于负载平衡完成参数扫描应用的任务分配,这样做的原因是追求应用响应时间最短.在本文实验中,参数扫描应用的任务分配发生于网格中已有 500 个单节点作业等待、运行或完成的背景下,故实验可以评估多作业共享网格时该调度方法改善参数扫描应用执行的有效性.实验表明,这种任务分配可有效降低参数扫描应用的响应时间.

网格是整合地理分布的计算、存储、应用、数据等资源的复杂系统,中间件等系统软件负责屏蔽资源异构、匹配计算请求与资源性能、处理作业取消与失败等.为突出参数应用的调度研究,这里忽略了调度开销、异常处理等问题.尽管参数应用的网格调度和节点的本地调度是并发的,但网格负载很轻时网格调度开销不能忽略.作业取消或失败等异常情况发生时,节点本地调度一般会动态优化作业队列,这通常不会恶化参数应用的调度结果,但如果能在负载平衡调度方法基础上,引入动态优化,可能会进一步改善参数应用的执行.节点本地调度往往不能实现 100% 的处理机利用率,节点的作业负载用这里定义的作业处理机需求表示并不准确.此外,基于负载平衡的任务分配还需适应各节点多样化的本地调度,以及多个参数扫描应用并发执行的情况.下一步,将对上述问题进行一些研究和探讨.

参考文献:

- [1] Abramson D, Sosic R, Giddy J, et al. Nimrod: a tool for performing parametrised simulations using distributed workstations [A]. Proc of the 4th IEEE Symposium on High Performance Distributed Computing [C]. Washington: IEEE Computer Society, 1995. 112 – 121.
- [2] Buyya R, Abramson D, Giddy J. Nimrod/G: an architecture for a resource management and scheduling system in a global computational grid [A]. Proc of the 4th International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region [C]. Washington: IEEE Computer Society, 2000 (1): 283 – 289.
- [3] Shava S, Walfredo C, Jaime F, et al. Combining workstations and supercomputers to support grid applications: The parallel tomography experience [A]. Proc of the 9th Heterogeneous Computing Workshop [C]. Washington: IEEE Computer Society, 2000. 241 – 252.
- [4] Huedo E, Montero R S, Llorente I M. Experiences on adaptive grid scheduling of parameter sweep applications [A]. Proc of the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing [C]. Washington: IEEE Computer Society, 2004. 28 – 33.
- [5] Casanova H, Legrand A, Zagorodnov D, et al. Heuristics for scheduling parameter sweep applications in grid environments [A]. Proc of the 9th Heterogeneous Computing Workshop [C]. Washington: IEEE Computer Society, 2000. 349 – 363.
- [6] 王庆江, 桂小林, 郑守淇. 参数扫描应用的网格自适应调度 [J]. 西安交通大学学报, 2004, 38(2): 111 – 114.
Wang Qingjiang, Gui Xiaolin, Zheng Shouqi. Adaptive scheduling for parameter sweep applications in grid [J]. Journal of Xi'an Jiaotong University, 2004, 38(2): 111 – 114. (in Chinese)
- [7] Ernemann C, Hamscher V, Schwiegelshohn U, et al. On advantages of grid computing for parallel job scheduling [A]. Proc of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid [C]. Washington: IEEE Computer Society, 2002. 31 – 38.
- [8] Bailey D, Barszcz E, Barton J, et al. The NAS Parallel Benchmarks [R/OL]. <http://www.nas.nasa.gov/News/Techreports/1994/PDF/RNR-94-007.pdf>, 2006-04-12.
- [9] Mu'alem A W, Feitelson D G. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling [J]. IEEE Transactions on Parallel and Distributed Systems, 2001, 2(6): 529 – 543.
- [10] 王庆江. 计算网格在无中心式调度框架下的作业调度方法研究 [D]. 西安: 西安交通大学, 2005.

作者简介:



王庆江 男, 1968 年生于山东高青, 中国海洋大学计算机科学系讲师、博士. 研究方向为高性能计算、分布式计算. E-mail: qjwang@ouc.edu.cn



刘培顺 男, 1975 年生于山东菏泽, 中国海洋大学计算机科学系讲师、博士. 研究方向为网络安全、可信计算.

高云 女, 1977 年生于山东淄博. 中国海洋大学计算机科学系讲师, 硕士. 研究方向为图像处理.