

任意可分负载的多轮调度算法

康雨, 闫相国, 郑崇勋, 申广杰

(西安交通大学生物医学信息工程教育部重点实验室, 710049, 西安)

摘要: 为了提高并行计算中具有负载任意可分特性的大规模应用的任务响应速度, 提出了一种针对带传输和计算延迟的三阶段多轮调度模型求解近似最优调度轮数的算法(DCMR). 通过对特定的调度时序分析, 得出闭合式方程组, 然后利用二分法快速搜索并结合回溯调整法求解近似最优调度轮数, 使计算时间尽可能多地与传输时间重叠, 从而缩短了整个应用的执行时间. 算法经仿真表明: 在多种参数变化的情况下, 可以求解出近似最优的调度方案; 与经典的 FIFO 和 LIFO 算法相比具有更强的自适应能力; 在计算时间明显大于传输时间的情况下, 能够稳定地保持任务响应时间为理想时间的 1.1 倍左右.

关键词: 任意可分负载; 多轮调度; 并行计算

中图分类号: R319; TH772.2 **文献标志码:** A **文章编号:** 0253-987X(2009)08-0125-05

A Multi-Round Scheduling Algorithm of Data-Collection for Divisible Workload

KANG Yu, YAN Xiangguo, ZHENG Chongxun, SHEN Guangjie

(The Key Laboratory of Biomedical Information Engineering of Ministry of Education,

Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A multi-round scheduling algorithm, data-collection multi-round (DCMR), is presented to minimize the makespan of divisible workloads in parallel computing. A three-stage model is proposed and takes communication latency and computation start-up time into consideration. The algorithm provides a method to generate a near-optimal number of scheduling rounds. Close-form equations are given through analyzing a specific time sequence of load distribution, and then the bisection method, combined with back-forward adjustment, is used to get an asymptotically optimal number of scheduling rounds, which make the computation time overlap the communication time as much as possible and reduce the makespan. Simulation results show that the algorithm can find a near-optimal number of scheduling rounds under different network parameters. Compared with the classical algorithms such as FIFO and LIFO, the DCMR has higher adaptability. When the computation time dominates the communication time, the algorithm can keep the makespan at a rather low level which is about 1.1 times of the ideal time.

Keywords: divisible workload; multi-round scheduling; parallel computing

在并行计算领域, 处理的任务类型主要分为不可任意划分负载和任意可划分负载. 与不可任意划分的任务相比, 后者又可划分为相互独立的子任务, 这些子任务之间没有相互依赖的关系, 因此各个节

点在处理任务的过程中不会产生额外的通讯开销. 在图像处理、数据挖掘、信号处理等领域, 很多应用^[1-3]都具有任意可划分特性(在实际中存在最小的划分单元).

针对可任意划分负载的多轮调度方法,是通过将任务划分成许多块(chunk)分多次由主节点(master)交给工作节点(worker),再通过适当的时序安排使得各个 worker 上的计算时间能够尽量多地覆盖传输时间. 最初的固定调度轮数下的多轮调度方法研究模型未考虑传输延迟和计算准备时间^[4-5],后续出现的多轮调度方法^[6-8]的研究模型考虑了网络传输延迟和计算准备时间,尤其是文献^[8]提出的 UMR(Uniform Multi-Round)算法,分别针对同构和异构网络的计算环境对多轮调度进行了详细的讨论,并给出了求解近似最优调度轮数的方法. 以上所述方法都是研究二阶段模型的,即只考虑了任务分发和任务处理阶段,却忽略了结果数据返回阶段.

总体来说,目前针对三阶段模型的研究结果比较少,除了以二阶段模型为研究目标的 UMR 算法外,尚未见到求解最优或者近似最优调度轮数的方法. 针对以上问题,本文提出了一种针对三阶段模型的多轮调度算法(DCMR),它在考虑了传输延迟和计算延迟的情况下,能够快速求解近似最优调度轮数,使得各个 worker 上的计算时间能够尽可能多地覆盖传输时间,同时还讨论了节点各个性能参数变化对 DCMR 算法的影响.

1 计算平台模型

本文研究的是由一个 master 和 N 个 worker 构成的星型网络拓扑计算平台,如图 1 所示. 对于一个可任意划分的任务,worker 在处理完本轮所接收的任务之后,会给 master 返回结果数据. master 以串行传输的方式发送、接受数据,即同一时间内发送、接收操作不可重叠,不参与任务处理. 对于任意一个 worker,计算时间可以与数据的发送、接收时间重叠.

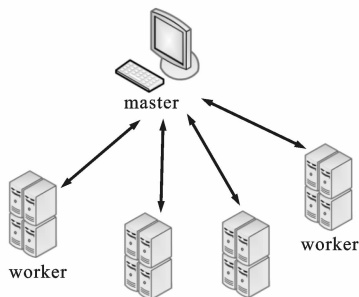


图 1 星型网络拓扑计算平台

2 DCMR 算法

2.1 调度模型

为了使计算时间尽可能地覆盖传输时间,最佳情形是 master 不间断地进行传输、接收操作,并且在任务全部执行完成之前,每一个 worker 不会出现空闲.

假设给定的一个 worker,其参数恒定不变,即 master 向第 i 个 worker 发送数据时的传输延时 t_i^m 、第 i 个 worker 向 master 返回数据时的传输延时 t_i^r 、第 i 个 worker 的计算延时 t_i^c 、master 到第 i 个 worker 的传输速率 B_i 、第 i 个 worker 向 master 返回数据的传输速率 B'_i 和第 i 个 worker 的计算能力 S_i 等恒定不变,则本文以 3 个节点的 4 轮调度时序(见图 2)为例进行 DCMR 算法推导.

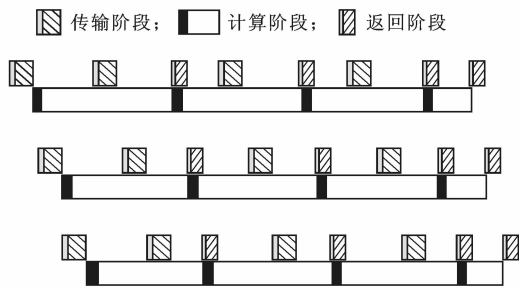


图 2 4 轮调度时序图

如图 2 所示,worker 接收到的任务顺序是从 1 到 N 排列的,其假设第 j 轮从第 i 个 worker 上返回给 master 的数据量 $f_{j,i}$ 与本轮调度时从第 i 个 worker 上获得的数据块的大小 $c_{j,i}$ 成正比,即

$$f_{j,i} = \delta c_{j,i} \quad (1)$$

式中: δ 为信息返回比,是一个给定的值.

对于第 1 轮调度应该满足

$$\frac{c_{1,i}}{S_i} + t_i^c + \frac{c_{1,i}}{B_i} + t_i^m = \frac{c_{1,i-1}}{S_{i-1}} + t_{i-1}^c + \frac{f_{1,i-1}}{B'_{i-1}} + t_{i-1}^r \quad 2 \leq i \leq N \quad (2)$$

$$\frac{c_{1,1}}{S_1} + t_1^c = \sum_{k=2}^N \left(\frac{c_{1,k}}{B_k} + t_k^m \right) + \sum_{k=1}^N \left(\frac{c_{2,k}}{B_k} + t_k^m \right) \quad (3)$$

从第 2 轮开始到 $M-1$ 轮调度应该满足

$$\frac{c_{j,i}}{S_i} + t_i^c + \frac{f_{j-1,i-1}}{B'_{i-1}} + t_{i-1}^r = \frac{c_{j,i-1}}{S_{i-1}} + t_{i-1}^c + \frac{f_{j,i-1}}{B'_{i-1}} + t_{i-1}^r \quad 2 \leq i \leq N \quad (4)$$

$$\frac{c_{j,1}}{S_1} + t_1^c = \sum_{k=1}^N \left(\frac{f_{j-1,k}}{B'_k} + t_k^r \right) + \sum_{k=1}^N \left(\frac{c_{j+1,k}}{B_k} + t_k^m \right) \quad (5)$$

对于最后1轮调度应满足

$$\begin{aligned} \frac{c_{M,i}}{S_i} + t_i^{\text{cp}} + \frac{f_{M-1,i-1}}{B'_{i-1}} + t_{i-1}^{\text{re}} = \\ \frac{c_{M,i-1}}{S_{i-1}} + t_{i-1}^{\text{cp}} + \frac{f_{M,i-1}}{B'_{i-1}} + t_{i-1}^{\text{re}} \\ 2 \leq i \leq N \end{aligned} \quad (6)$$

2.2 调度轮数的确定

文献[9]中的CAMR(Collection Aware Multi-Round)算法研究的是三阶段模型,但其忽略了计算准备时间和传输延迟,则 $c_{j,i}$ 与 $c_{1,1}$ 呈正比关系,即 $c_{j,i} = ac_{1,1}$,因此CAMR可以先令 $c_{1,1} = 1$,依照推导出的比例关系依次计算 $c_{j,i}$,直到 $c_{j,i}$ 趋近于0,或者达到预先给定的最大轮数,最后按照计算出来的比例对总任务进行划分.这种方法忽略了延迟,理论上最优调度轮数是无限大,但忽略了延迟参数的三阶段调度模型与实际应用仍然存在着差距.

对于一个给定的调度轮数 M ,master 分配到各个 worker 上的任务总量为

$$W_{\text{total}} = \sum_{j=1}^M \sum_{i=1}^N c_{j,i} \quad (7)$$

式中: M 为待求的总调度轮数; N 为总的 worker 数量.设 N 为已知量,将式(1)代入式(2)~式(6),则其可以简化为含有 $MN+1$ 个变量的 $MN-1$ 个线性方程,加之式(7),总计有 MN 个相互独立的方程.如果给定一个 M ,便可求出一组 $c_{j,i}$.另外,参照图1,采用DCMR进行调度时的任务完成时间(makespan)为

$$\begin{aligned} T(M) = \sum_{i=1}^N \left(\frac{c_{1,i}}{B_i} + t_i^{\text{cm}} \right) + \\ \sum_{j=1}^M \left(\frac{c_{j,N}}{S_N} + t_N^{\text{cp}} \right) + \frac{f_{M,N}}{B'_N} + t_N^{\text{re}} \end{aligned} \quad (8)$$

因此,对于一个给定的任务总量和 worker 数,则存在一个最优的调度轮数,使得式(8)最小.本文采用二分法搜索最佳的调度轮数.

DCMR 步骤如下.

- (1)随机选取一个调度轮数 $M > 2$.
- (2)求解由式(1)~式(7)构成的方程组.
- (3)如果在步骤(2)的结果中出现 $c_{j,i} \leq 0$,转步骤(5).
- (4)如果在步骤(2)中计算结果均满足 $c_{j,i} > 0$,则 $M \leftarrow 2 \times M$,转步骤(2).
- (5)用二分法求解 $[M/2, M]$ 范围内 $c_{j,i}$ 的分配情况,选择的原则是:找到一个调度轮数 M' ,其恰好满足所有的 $c_{j,i} > 0$.
- (6)以最小的 makespan 为目标,对最终的分配

方式进行调整.

2.3 优化调整

由于在DCMR的步骤(5)中找到了一个 M' ,恰好使得所有的 $c_{j,i} > 0$,也就是说当 $M' + 1$ 时,解中会出现 $c_{j,i} < 0$ 的情况,但出现的数量和大小都不确定.在极端情况下,如果只有一个 $c_{j,i} < 0$,此时若不要求所有节点在最后一轮都有任务,则可能出现采用 $M' + 1$ 比采用 M' 所获得的响应时间更短.因此,需做以下调整:将出现的负值依次累加到紧邻的非负 $c_{j,i}$ 之上,直到最后的负值被消除为止,其伪代码如图3所示.

```

1.  $j \leftarrow M, i \leftarrow N$ 
2. while  $c_{j,i} < 0$ 
3.     find the nearest  $c_{a,b} > 0$ 
4.      $c_{a,b} \leftarrow c_{a,b} + c_{j,i}$ 
5.     if  $(i = 1) < 0$ 
6.          $j \leftarrow j - 1$ 
7.          $i \leftarrow i + N$ 
8.     endif
9. endwhile

```

图3 负值消除伪代码

对于 $M', M' + 1, M' + 2, \dots$,做上述调整后计算对应的响应时间,如果发现响应时间开始上升则停止计算,并取响应时间最小的作为最终结果(在多组实验中发现,在到达 $M' + 3$ 之前,响应时间就开始上升).另外,对于采用多轮调度策略进行任意可分负载的分配,都要先保证每轮中的计算时间明显大于该轮中的最大数据传输时间,因此要采用多轮调度,对于三阶段模型在同构环境下应满足

$$B > N(1 + \delta)S \quad (9)$$

显然,式(9)是一个很宽松的条件,在这个条件下DCMR求出的轮数至少大于3.对于异构环境,选取 worker 的方式需要就具体情况而定,总的原则是要保证计算时间明显大于传输时间.根据文献[4]中的证明得知,在异构环境下,按照传输速率递减顺序排列进行任务分发能够获得更好的响应时间.

3 实验结果与分析

利用 SimGrid 工具进行了仿真,并在同构环境下进行了测试.由于除本文方法外,尚未见到针对带延迟参数的三阶段模型求解近似最优调度轮数的算法,因此仅与经典的调度算法,即 FIFO 法、LIFO 法进行了比较.在仿真实验中,取总任务量 $W_{\text{total}} = 200$ 个,信息返回比 $\delta = 1$,worker 的任务计算能力 $S = 1$ 个/s,接收和发送速率 $B = B' = 20$ 个/s,计算延迟

以及发送、返回延迟 $t_i^p = t_i^m = t_i^r = 0.03$ s. 图4显示了同构环境下任务的响应时间与 N 、 M 之间的关系, 其中每一条曲线的终节点是给定 N 情况下用 DCMR 在未做优化调整时求出的 M' 所对应的响应时间. 可以看出, 参与计算的 worker 数量越多, 响应速度越快. 在固定 worker 各项性能参数的情况下, 随着 worker 数量的增加, 最优调度轮数快速减少. 更重要的是, 图示显示当给定调度轮数对应的所有解 $c_{j,i} > 0$ 时, 响应时间是随着轮数的增加而持续递减, 这个结果是采用二分法的一个重要依据, 同时也解释了为什么在 2.2 节中未计算 $M' - 1$ 的响应时间的原因.

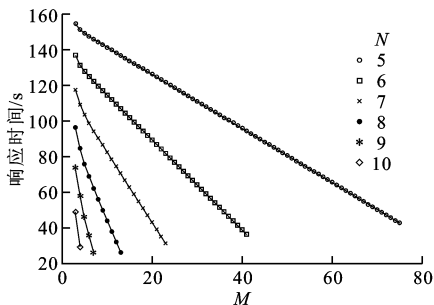


图4 响应时间与 M 及 N 之间的关系

在同构并行计算环境下, 将实际完成时间与理想完成时间的比值作为规范化的完成时间 $T = T_{\text{makespan}} / T_{\text{ideal}}$, 其中理想完成时间 $T_{\text{ideal}} = W_{\text{total}} / (NS)$. 仿真环境中, 取 $N=10$, $B=B'=30$ 个/s, 仿真结果如图5、图6所示. 从图5可以看出, 对于三阶段模型, DCMR 在满足式(9)的情况下明显优于经典的 FIFO、LIFO 法. 图5中, 随着信息返回比的增加, 在计算总量不变的情况下, 总的传输量增加, 但 DCMR 的 T 仍然保持在 1.1 左右. 图6为调整后的结果, 可以看出, 随着传输能力的加强, DCMR 可以划分更多更小的块去覆盖传输时间, 从而使得 T 逐渐逼近 1. 另外, 从 2.1 节的各组方程可以看出, 用 DCMR 求得的最小调度轮数是 3. 在 NS 与 B 接近的情况下, 受 DCMR 求解轮数的限制, 结果会导致响应时间偏高, 图6中的当传输速率计算能力比小于 20 时, DCMR 的相应时间曲线加快上升. 这个缺陷可以通过略微修改求解过程加以弥补, 即将 2 轮(由式(1)、式(2)、式(3)、式(6)、式(7)构成方程组)和 1 轮(FIFO)的情况同时结合 2.3 节中的优化调整一起考虑, 在这种情况下, 只要 $B > \max(NS, \delta NS)$, 就可能无解. 从图6还可以看出,

相比于 FIFO、LIFO 法, 由于 DCMR 能够利用计算时间去尽可能多地覆盖传输时间, 因此具有更高的效率. 总体来说, 在满足式(9)的条件下, DCMR 在不同参数变化时, 均能够保持稳定的 T , 使之在 1.1 左右.

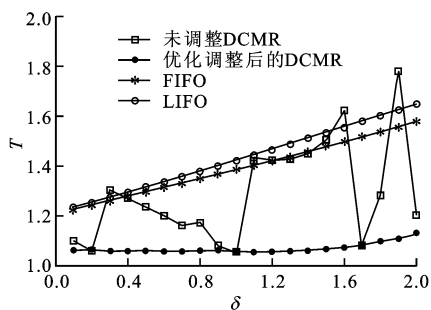


图5 信息返回比 δ 与 T 的关系

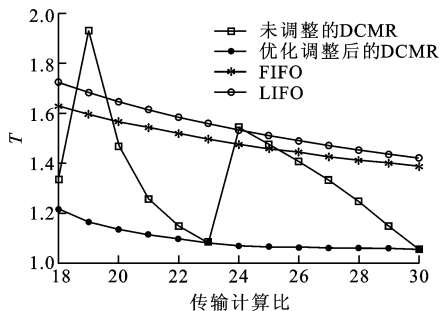


图6 传输速率计算能力比与 T 的关系

4 结 论

DCMR 同时考虑了计算延迟和传输延迟, 并针对任意可分负载在星型网络拓扑的并行计算平台进行了多轮调度. DCMR 利用二分法快速搜索出近似最优的调度轮数, 并做优化调整, 使得任务的响应时间最小. 实验证明了, DCMR 可以获得近似最优的调度方案, 其任务响应时间接近理想完成时间.

参考文献:

- [1] 康雨, 闫相国, 郑崇勋, 等. 医学可视化网格平台的设计与实现 [J]. 西安交通大学学报, 2007, 41(8): 1000-1002.
KANG Yu, YAN Xiangguo, ZHENG Chongxun, et al. Design and realization for a medical image visualization grid platform [J]. Journal of Xi'an Jiaotong University, 2007, 41(8): 1000-1002
- [2] KWANGIL K, ROBERTAZZI T G. Signature search time evaluation in flat file databases [J]. IEEE Trans

on Aerospace and Electronic Systems, 2008, 44(2): 493-502.

[3] HUNG T G, ROBERTAZZI T G. Scheduling nonlinear computational loads [J]. IEEE Trans on Aerospace and Electronic Systems, 2008, 44(3): 1169-1182.

[4] BHARADWAJ V, GHOSE D, MAN V. Multiinstallment load distribution in tree networks with delays[J]. IEEE Trans on Aerospace and Electronic Systems, 1995, 31(2): 555-567.

[5] BHARADWAJ V, GHOSE D, MAN V, et al. Scheduling divisible loads in parallel and distributed systems [M]. Los Alemitos, USA: IEEE Computer Society, 1996.

[6] HAGERUP T. Allocating independent tasks to parallel processors: an experimental study [J]. Journal of Parallel and Distributed Computing, 1996, 11(17): 1-33.

[7] BEAUMONT O, LEGRAND A, ROBERT Y. Scheduling divisible workloads on heterogeneous platforms [J]. Parallel Computing, 2003, 29(9):1121-1152.

[8] YANG Y, RAADT K, CASANOVA H. Multiround algorithms for scheduling divisible loads [J]. IEEE Trans on Parallel and Distributed Systems, 2005, 16(11): 1092-1102.

[9] 赵明宇,张田文. 三段可任意划分负载应用的多次数据分配 [J]. 哈尔滨工业大学学报, 2008, 40(5):745-749.

ZHAO Mingyu, ZHANG Tianwen. A collection-aware multi-round scheduling algorithm [J]. Journal of Harbin Institute of Technology, 2008, 40(5):745-749.

(编辑 苗凌)