MDLP—线性异构网络中多个可划分任务的资源分配策略

刘洪涛[1,2] 岳 鹏[1] 杨 娟[1] 邱玉辉[1]
(西南大学计算机与信息科学学院 重庆 400715)[1]
(重庆教育学院计算机科学与现代教育技术系 重庆 400067)[2]

摘 要 分布式系统中提高自治并发处理的能力的关键途径是提高任务资源分配的合理程度。可划分任务调度作为其中一个分支也受到众多关注。现已有许多针对可划分任务资源分配的研究,但它们要么是基于同构网络,要么是单一任务分配策略。本文提出了一个异构线性网络中可进行多个可划分任务资源分配的策略 MDLP。MDLP 通过构建一个线性规划问题,简化了复杂的求解过程,并且在 MDLP 的算术模型中,不必预先知晓前个任务的结束时间就可以求得最优解。

关键词 资源分配,可划分任务,线性网络

MDLP—The Resource Allocating Policy for Multi Divisible Loads in Heterogeneous Linear Networks

LIU Hong-Tao[1,2] YUE Peng[1] YANG Juan[1] QIU Yu-Hui[1]
(School of Computer and Information Science, Southwest University, Chongqing 4000715)[1]
(Department of Computer and Modern Education Technology, Chongqing Education Collese, Chongqing 400067)[2]

Abstract Reasonably allocating the resources to the loads can effectively improve the concurrency ability of the distributed systems. The resource allocating policy for the divisible loads is one kind of those polices. Many scholars has got their focus on the resource allocating problems about the divisible loads under distributed systems. However, most of them are under homogeneous systems, others are used to treat one load case. In this paper, we propose a resource allocating policy—MDLP, for multi divisible loads under heterogeneous linear system. MDLP model the optimize problem of the divisible loads into a linear programming problem, and solve it through a simple way. On the other hand, the finish time of the previous loads is no longer need to used to compute the current load's distribution.

Keywords Resource allocating, Divisible loads, Linear networks

1 前言

分布式系统中提高自治并发处理的能力的关键途径是提高任务资源分配的合理程度。最优的资源分配方案往往能获得较高的并发处理性能。而分布式系统中的资源分配策略的划分除了可以从资源类型的角度进行划分(典型的是任务的处理器调度[2],还包括网络带宽调度[1]和存储器调度等方面。而根据任务类型的不同则可划分为:1)事件驱动[3,4]和 2)数据驱动[5~7]两大类。本文提出的资源分配策略是资源类型为处理器的资源分配策略。且所处理的任务是可划分的任务(Divisible Loads),且划分后子任务间可独立执行,相互间无依赖关系。与数据驱动的 DAG(有向无环图)任务流不同,可划分任务流是事件驱动的。理论上说可划分任务是可划分为任意大小的。可划分任务的处理可以在各种拓扑结构网络中进行,如总线型[8]、树型[9],及 mesh 网和超立方网络结构[10]。这里我们假设网络环境是在线性网络中进行处理。

文[11]考虑了实时系统的时限的可划分任务调度策略,但是它没有涉及到多任务的调度和最优性问题。文[12]是线性网络中的可划分任务的最优调度策略,但是它假设的前提是处理器的可执行时间是任意的。文[13]则是在总线型网络中进行可划分任务的调度。文[14]是一个在线性网络中进行可划分任务最优化调度,但是它的缺点在于它在进行多任务调度时假设当前任务的调度已知道前一个任务的完成时间,并且具有复杂的求解过程。

本文提出了一个异构线性网络中可进行多个可划分任务调度的资源分配策略 MDLP(Multi Divisible Loads Policy)。在 MDLP 的算术模型中,我们将最优问题构建成为一个线性规划问题,使得复杂的求解过程简单化。与文[14]需要上个任务的执行完成时间我们将当前调度任务所需的数据与前一个任务的执行结束估计时间建立关联,通过对上个任务分布的迭代求得逐渐收敛的任务完成估计时间,再利用这个估计时间对当前任务的资源分布逐步求精。当上个任务的分布收敛时,可求得准确的最优计算完成时间,此时当前任务的资源分布也达到最优。

2 网络环境和计算环境构建

用一个 5 元组 $G=\langle L, LINK, P, \omega, Z\rangle$ 来描述一个线性网络,如图 1。其中 $L=\{L_n \mid n\in[1,N]\}$,表示边界节点 p_1 上加载的可划分任务集合。$LINK=\{l_i \mid i\in[1,m-1]\}$,表示节点间的链路集合。$P=\{p_i \mid i\in[1,m]\}$,表示网络中处理器集合。$\omega=\{\omega_i \mid i\in[1,m]\}$,表示处理器 p_i 的计算能力集合,ω_i 的具体定义见表 1。$Z=\{z_i \mid i\in[1,m-1]\}$,表示链路 l_i 的通信能力集合,z_i 的具体定义见表 1。

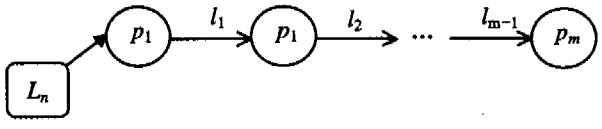


图 1 线性网络 G

假设 G 为单端口环境[15]，即 p_i 在接收或发送任务时可同时进行计算，但接收、发送任务则不能同时进行。

在进行计算环境构建之前先给出两个关于可划分任务的相关定义。

定义 1 使得可划分任务计算最优的充分必要条件是 G 中所有参与的处理器 $p_i\ \forall i\in[1,m]$，计算结束时刻必须在同一时刻，且 N 中没有空闲处理器。

定义 2 若一个任务 L_n 能保证在 L_{n-1} 完成之前传递到 G 中所有参与的处理器 $p_i\ \forall i\in[1,m]$ 上，并各自保留一部分进行计算，则称 L_n 是不用分割的。

在我们提出的 MDLP 中，我们假设所有的任务都是不用分割的。分割任务的目的在于当 L_n 过大时，L_n 直接分发的时间过长会导致处理器空闲，从而破坏定义 1。因此为避免这种情况可采用分割技术。L_n 可被分割为 $L_n^1,L_n^2,\cdots,L_n^k$，k 部分，且有 $\sum_{i=1}^{k}L_n^i=L_n$，$\forall L_n^i$ 满足下面提出的限制条件④。

本文中使用到的相关参数及其定义见表 1。

表 1 相关参数定义表

L_n	可划分任务 L_n，$n\in[1,N]$
N	加载在边界节点 p_1 上的可划分任务的总数
$F(L_n)$	任务 L_n 处理完成的时间，定义为 p_m 上计算完成的时刻
$T_p(L_n)$	标准处理器计算任务量为 L_n 的时间，标准处理器定义为 $\omega_i=1$ 的处理器
ω_i	处理器 p_i 的计算速度的倒数，即与标准处理器处理性能的比率
$T_c(L_n)$	标准链路传递任务量为 L_n 的时间，标准链路定义为 $Z_i=1$ 的链路
z_i	链路 l_i 的通信速度的倒数，即与标准链路通信性能的比率
α_n^i	任务 L_n 分配在 p_i 上的比例，即 p_i 上的任务量为 $L_n\cdot\alpha_n^i$
$t_D(L_n)$	L_n 在 p_1 开始向其他节点分发的时刻
m	网络 N 中处理器的个数

3 问题构建

我们的目的是在线性网络 G 中求的 L_n 的最优分布，即 $Optimize(L_n)$，满足定义 1，且满足下列限制条件：

①依照定义 1，有 p_1 上执行任务完成的时刻应与剩余任务传递到 p_m 后 p_m 执行 $L_n\cdot\alpha_n^m$ 部分完成的时刻相同，如图 2。有式(1)。

$$F(L_{n-1})+T_p(L_n)\cdot\omega_1\cdot\alpha_n^1=t_D(L_n)+T_c(L_n)\cdot\sum_{i=1}^{m-1}z_i(1-\sum_{j=1}^{i}\alpha_n^j)+T_p(L_n)\cdot\omega_m\cdot(1-\sum_{i=1}^{m-1}\alpha_n^i)\quad(1)$$

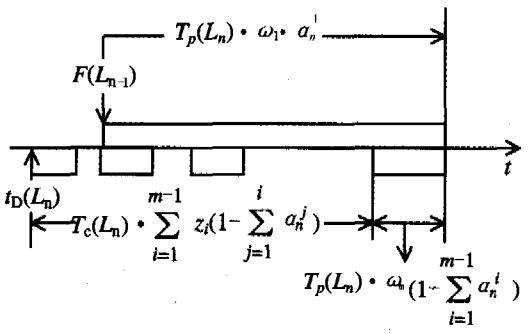


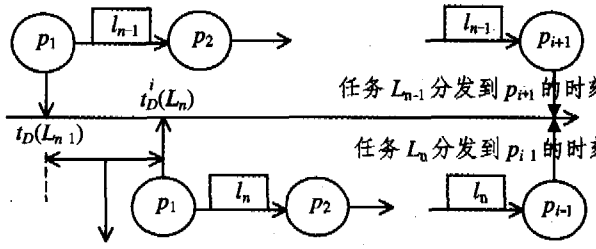
图 2 (1)式说明图

②任务加载在 p_1 上后开始分发的时间应满足(2)式，即在 L_{n-1} 完成之前，并在 L_{n-1} 开始分发之后。如图 2。

$$F(L_{n-1})>t_D(L_n)>t_D(L_{n-1})\quad(2)$$

③因为 G 是单端口模型，因此 L_n 的分发要避免与 L_{n-1} 任务分发的冲突。假设 L_{n-1} 总是比 L_n 先开始发送，因此有 L_{n-1} 分发到 p_i 的时刻要早于 L_n 分发到 p_i 的时刻。且 p_i 接收 L_n 的时间段里是不能分发 L_{n-1} 的。假设此刻 L_{n-1} 已从 p_i 分发出去，因此对于每个处理器 p_i 来说，有

$$(*)=(1*)+(2*)\quad(3)$$



对当前处理器 p_i 来说接收任务和发送剩余任务所需的最短时间 $t_D(L_n)$ 只有晚于 $t_D(L_n)$ 才能避免在 p_i 发生接收任务的冲突

图 3 (4)式的说明图

$(*)=t_D^i(L_n)+T_c(L_n)\cdot\sum_{p=1}^{i}z_p(1-\sum_{j=1}^{p}\alpha_n^j)$ 表示任务 L_n 分发到 p_{i+1} 的时刻；$(1*)=t_D^i(L_{n-1})+T_c(L_{n-1})\cdot\sum_{p=1}^{i}Zp(1-\sum_{j=1}^{p}\alpha_{n-1}^j)$ 表示任务 L_{n-1} 分发到 p_{i-1} 的时刻；$(2*)=T_c(L_n)\cdot\sum_{p=1}^{i}z_p(1-\sum_{j=1}^{p}\alpha_n^j)-T_c(L_n)\cdot\sum_{p=1}^{i-2}z_p(1-\sum_{j=1}^{p}\alpha_n^j)$，表示 p_i 接收和发送任务的时间耗费。如图 3 所示。由(3)式可求得

$$t_D^i(L_n)=t_D^i(L_{n-1})+T_c(L_{n-1})\cdot\sum_{p=1}^{i}z_p(1-\sum_{j=1}^{p}\alpha_{n-1}^j)-T_c(L_n)\cdot\sum_{p=1}^{i-2}z_p(1-\sum_{j=1}^{p}\alpha_n^j)\quad(4)$$

令 $t_D(L_n)=\max\{t_D^i(L_n)\},i\in[1,m-1]\quad(5)$

即 L_n 的分发开始时间应取一个最大延迟才能保证所有 p_i 都不发生接收冲突。

④为保证所有任务都不用分割，即所有处理器都不会空闲，则有

$$T_c(L_n)\cdot\sum_{p=1}^{m-1}z_p(1-\sum_{j=1}^{p}\alpha_n^j)\leqslant F(L_{n-1})+t_D(L_n)\quad(6)$$

(6)式表示 L_n 必须在 L_{n-1} 完成之前传递到所有处理器，否则就会有处理器空闲。而 $F(L_{n-1})=L_{n-1}$ 到达 p_m 的时刻 $+p_m$ 处理 $L_{n-1}\cdot\alpha_{n-1}^m$ 部分的时间

$$=t_D(L_{n-1})+T_c(L_{n-1})\cdot\sum_{p=1}^{m-1}z_p(1-\sum_{j=1}^{p}\alpha_{n-1}^j)+T_c(L_{n-1})\cdot\omega_m\cdot(1-\sum_{p=1}^{m-1}\alpha_{n-1}^p)\quad(7)$$

将(7)式代入(6)式可得 $t_D(L_n)$ 关于 $t_D(L_{n-1})$ 的函数：

$$t_D(L_n)=f_a^n[t_D(L_{n-1})]\quad(8)$$

⑤为保证定义 1，令 p_i 计算本节点上任务量的时间为 p_i 将剩余任务传递到 p_{i+1} 上与 p_{i+1} 计算本节点上任务量的时间和，即

$$T_c(L_n)\cdot\omega_i\cdot\alpha_n^i=T_c(L_n)\cdot z_i\cdot(1-\sum_{j=1}^{i}\alpha_n^j)+\alpha_n^{i+1}\cdot\omega_i\cdot T_c(L_n)\quad(9)$$

⑥$\sum_{i=1}^{m}\alpha_n^i=1$。

⑦$t_D(L_1)=0$，即任务 L_1 在处理器 p_1 上分发任务的开始时间为初始时刻 0。

(下转第 61 页)

研究工作包括：基于具体应用的预测模型的建立和分析；节点运动数学模型的建立和分析；网络链路信息获取与预测机制的关系；基于预测的网络通信协议等。

参 考 文 献

1 Mobile Ad-hoc Networks (MANET). http://www.ietf.org/html.charters/manet-charter.html
2 McDonald A B, Znati T. A Path Availability Model for Wireless Ad-Hoc Networks. In: He D J, Jiang S M, Rao J Q, et al. eds. Proceedings of IEEE Wireless Communications and Networking Conference 1999 (WCNC'99), Link availability prediction model for wireless ad hoc networks. Proc. 2000 International Conference on Distributed Computing System Workshop, Taipei, Taiwan, (April 10-13, 2000). D7～D11
3 Jiang Shengming, He D J, Rao J Q. A prediction-based link availability estimation for mobile Ad Hoc networks. IEEE Infocom, 2001
4 Su W, Lee Sung-Ju, Gerla M. Mobility prediction and routing in ad hoc wireless networks. International Journal of Network Management, 2001, 11: 3～30
5 Shah H, Nahrstedt K. Predictive Location-Based QoS Routing in Mobile Ad Hoc Networks. In: Proceedings of IEEE International Conference on Communications (ICC 2002), New York, NY, April 2002
6 Su W, Gerla M. Ipv6 flow handoff in ad-hoc wireless networks using mobility prediction. In: Proceedings of IEEE GLOBECOM'99, Rio de Janeiro, Brazil, Dec. 1999
7 Zonoozi M, Dassanayake P. User mobility modeling and characterization of mobility paterns. IEEE Journal on Selected Areas in Communications, 1997, 15(7)
8 McDonald A B, Znati T. A mobility based framework for adaptive clustering in wireless ad-hoc networks. IEEE Journal on Selected Areas in Coummunications, Aug. 1999. Special Issue on Ad-Hoc Networks, (in press)
9 Wang Jianxin, Deng Shuguang, Chen Songqiao, et al. QoS Routing with Mobility Prediction in MANET. In: 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM' 01). Victoria, Canada. Aug 2001, 357～360
10 McDonald A B, Znati T. Link availability models for mobile ad-Hoc networks: [Technical Report]. TR99-07. Department of Computer Science, University of Pittsb Mayurgh, 1999

(上接第 42 页)

4 求解算法 MDLP_Algorithm()

下面对线性规划问题 Optimize(L_n)进行求解过程描述，即算法 MDLP_Algorithm()。

MDLP_Algorithm(){

Begin：给 α_n^i 赋初值为 $\alpha_n^i=\dfrac{\frac{1}{\omega_i}}{\sum_{j=1}^{m}\frac{1}{\omega_j}}$

(1)将 α_n^i 值代入(4)和(7)式得出 $t_D(L_n)$的值。

(2)由(9)式可得出 α_n^i 是一个关于 α_n^1 的函数，$\alpha_n^i=f^i(\alpha_n^1)$ (*)，将(*)代入(1)式，求得新的 α_n^1 及 α_n^i。

(3)若 $t_D(L_n)$收敛，则结束。否则转向 1 继续迭代。}

MDLP_Algorithm()是一个迭代的过程，在此期间 α_n^i 和 $t_D(L_n)$会有震荡，但始终会收敛。并且这个求解过程中由于 $t_D(L_n)$是一个关于 $t_D(L_{n-1})$的函数，因此 $t_D(L_n)$的计算可以不用如文[12]中一样预先知道 $t_D(L_{n-1})$的准确值，可以在 $t_D(L_{n-1})$的计算过程中逐步求精。

5 模拟试验

我们用分布式系统资源分配的传统度量手段 AOR 来显示 MDLP 的性能，如图 4。AOR＝ACT/OCT 是任务真实完成时间与最优计算时间的比率(ACT：Actual Complement Time，OCT：Optimal Computing Time)。通过多台计算机的多端口 Socket 通信模拟线性网络。理论上由于 L_n 是可以划分为任意大小的，因此结果有一定的误差。从图 4 可以看出，随着任务总量增加，ACT 与 OCT 的逼近度有一定震荡，这是由于模拟环境不准确的原因。

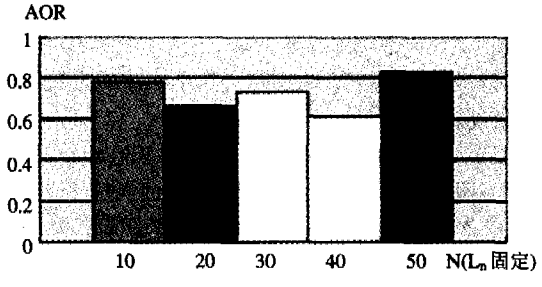


图 4

结论 本文提出了一个异构线性网络中可进行多个可划分任务资源分配的策略 MDLP。MDLP 通过构建一个线性规划问题，简化了复杂的求解过程，并且在 MDLP 的算术模型中，不必预先知晓前个任务的结束时间就可以求得最优解。但是 MDLP 的问题在于，我们没有对 MDLP 的最优解进行形式化的描述，也没有证明 MDLP 始终会找到最优解。这将会在我们将来的工组中完成。

参 考 文 献

1 Hou Y T, Panwar S S. On Generalized Max-Min Rate Allocation and Distributed Convergence Algorithm for Packet Networks. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(5): 401～416
2 Yang Juan, Baiyun, et al. DJSM—A Generally Dynamic Job scaling Mathematic Model for Parallel Applications. In: Proceedings of The 2005 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'05), Las Vegas, NV, June, 2005. 1099～1105
3 Georgiadis L, Nikolaou C, et al. A fair workload allocation policy for heterogeneous systems. Journal of Parallel and Distributed Computing, 2004, 64(1): 507～519
4 Rotaru T, Hans-Heinrich N. Dynamic load balancing by diffusion in heterogeneous systems. Journal of Parallel and Distributed Computing, 2004, 64(4): 481～497
5 Ali Shoukat, Kim Jong-Kook, et al. Greedy Heuristics for Resource Allocation in Dynamic Distributed Real-Time Heterogeneous Computing Systems. In: 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02), Las Vegas, NV: CSREA Press, 2002. 519～530
6 Ravindran B, Li Peng, et al. Proactive resource allocation for asynchronous real-time distributed systems in the presence of processor failures. Journal of Parallel and Distributed Computing, 2003, 63(12): 1219～1242
7 Ranaweera S, Agrawal D P. A Task Duplication Based Scheduling Algorithm for Heterogeneous Systems. In: Proceedings of the 16th International Parallel and Distributed Processing Symposium, Florida: IEEE Computer Society Press, 2002. 445～450
8 Sohn J, Robertazzi T G. Optimal divisible job load sharing on bus networks. IEEE Transactions on aerospace Election Systems, 1996, 32: 34～40
9 Barlas G. Collection-aware optimum sequencing of operations and closed-form solutions for the distribution of a divisible load on arbitrary processor trees. IEEE Transactions on Parallel and Distributed Systems, 1998, 9(5): 429～441
10 Blazewicz J, Bharadwaj V, et al. Divisible task scheduling-concept and verification. Parallel Computing, 1999. 25: 87～98
11 Eshahin M M. Heterogeneous Computing. Artech House, Norwood, MA, 1996
12 Bharadwaj V, min W H. Scheduling divisible loads on heterogeneous linear daisy chain networks with arbitrary processor release times. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(3): 273～288
13 Bharadwaj V, Li H F, et al. Scheduling divisible loads in bus networks with arbitrary processor release times. Computing Math Application, 1996, 32(7): 57～77
14 Min W H, Veeravalli B, et al. Design and performance evaluation of load distribution strategies for muliple divisible loads on heterogeneous linear daisy chain networks. Journal of Parallel and Distributed Computing, 2005, 65: 1558～1577
15 Hsu W J, Chung M J, et al. Linear recursive networks and their applications in distributed systems. IEEE Transactions on Parallel and Distributed Systems, 1997, 8(7): 673～680