文章编号 1004-924X(2007)12-1969-05

异构多处理机系统的负载均衡与任务调度

童小念,舒万能,李子茂
(中南民族大学 计算机科学学院,湖北 武汉 430074)

摘要:分析了异构多处理机系统中的负载均衡和任务调度参数,讨论了异构集群任务调度模型,提出了一种负载均衡启发式优化算法(LBHOA)。LBHOA 采用启发式搜索策略,每次分配一个任务时,从不完全分配的结点中选择估计值最小的结点进行扩展搜索,直到找到完全分配的目标结点,且目标结点的时间开销估计值是所有完全分配结点中最小的。实验结果表明,与算法 WLCA 和 LTGA 相比,算法 LBHOA 的平均应答延迟时间的开销减少了 10%,任务完成时间的开销减少了 15%。LBHOA 降低了资源最优分配中的计算复杂度,能够满足异构集群系统中的负载平衡和优化调度的需要,使异构多处理机系统在系统资源均衡分配的同时使系统资源利用率最优。

关 键 词:异构集群;负载均衡;任务调度;负载均衡启发式优化算法
中图分类号:TP311.5 **文献标识码**:A

Load balancing and task scheduling of heterogeneous multiprocessor system

TONG Xiao-nian, SHU Wan-neng, LI Zi-mao

(*College of Computer Science, South-central University for Nationalities, Wuhan* 430074, *China*)

Abstract: In order to share various resources proportionally and to optimize the utility of system resource in a heterogeneous multiprocessor, a Load Balance Heuristic Optimization Algorithm (LBHOA) is proposed after the analysis of load balance and learning of task scheduling model. LBHOA uses heuristic search technique to choose the node of least estimation value and to spread search until finding a target note that has a least value of time spending in all complete distributive notes. The experimental results indicate that LBHOA can reduce 10% of average response delay and 15% of working time comparing with WLCA and LTGA and also can reduce computing complexity in resource allocating, which meets the requirements of load balance and optimization scheduling in heterogeneous multiprocessor system.

Key words: heterogeneous cluster; load balancing; task scheduling; Load Balancing Heuristic Optimization Algorithm(LBHOA)

收稿日期:2007-09-29;**修订日期**:2007-11-02.
基金项目:国家自然科学基金资助项目(No. 60603008);湖北省教育厅高等学校教学研究资助项目(No. 20050232)

1 引　言

在多处理机系统中，任务调度的优化是提高并行效率的重要因素，由此而获得的各个处理机的负载均衡是高性能集群系统并行性的基本保证[1]。特别是对于异构集群系统，由于各处理机的处理能力和通信速率都不相同，在分配作业和任务时，如何为处理机合理地分配负载以达到最优的并行计算性能，是异构多处理机系统研究中的一个热点问题。

负载均衡分为两类，一类是静态负载均衡，它是在任务确定的情况下，基于各个处理机的计算性能，调度一个任务集合，进行合理的任务分配。由于静态负载均衡算法主要是基于节点的处理能力而不考虑节点负载的动态变化，所以，实现静态负载均衡策略，其算法的复杂度低且效率较高，但其灵活性与针对性不足；另一类是动态负载均衡，它在任务不确定的情况下，基于节点负载的动态变化，根据系统当前的负载状态有针对性地进行负载分配，指派各个任务的执行过程。动态负载平衡策略具有灵活性与可伸缩性，但它在分析处理系统的状态时会产生额外开销。因此，在动态负载平衡算法的收益和代价方面需要权衡其利弊。

鉴于静态负载均衡技术具有确定的负载均衡规律，是分析与评价并行系统设计和计算性能的不可或缺的手段及策略，本文在分析异构多处理机系统中的负载均衡和任务调度参数的前提下，提出了一种基于任务计算量向量、任务通信量矩阵的静态负载均衡启发式优化算法 LBHOA，旨在研究并探讨异构多处理机系统中的负载均衡及任务优化调度问题。

2 异构集群任务调度模型

异构集群的任务调度模型应该考虑下述影响处理机负载均衡的 4 个主要因素：(1)任务的计算量不同；(2)任务之间的通信量不等；(3)处理机的计算速度不等；(4)处理机之间的通信率不等。为了便于分析问题，给出该问题的一般性定义：

$$TG=(T,C)$$

$$PG=(P,V)$$

$TG=(T,C)$定义为一个划分为 n 个任务的作业。结点集合 $T=\{t_k\mid k=1,2,\cdots,n\}$表示 n 个任务，用任务计算量向量 $\boldsymbol{R}=(r_k)_n$ 描述各任务的计算量，任务 t_k 的计算量为 r_k。$\boldsymbol{C}=[c_{kq}]_{n\times n}$表示任务间通信矩阵，$c_{kq}$为任务 t_k 同任务 t_q 之间的通信量。

$PG=(P,V)$定义为一个连接 m 个处理机的异构集群。结点集合 $P=\{p_i\mid i=1,2,\cdots,m\}$表示 m 个处理机，用处理机计算速度向量 $\boldsymbol{E}=(e_i)_m$ 描述可使用的 m 个处理机的计算能力，处理机 p_i 的计算能力为 e_i。$V=[\nu_{ij}]_{m\times m}$描述 m 个处理机之间的通信速率，ν_{ij}为处理机 p_i 同处理机 p_j 之间的通信速率。

异构集群的负载均衡任务优化调度问题可以描述为：寻找一个映射关系，将任务图 TG 映射到集群图 PG，使作业时间最短[2]。

为了更好地描述问题，引入任务分配矩阵 $\boldsymbol{A}=[a_{ki}]_{n\times m}$，表示 TG 到 PG 的映射关系，定义 $a_{ki}=\begin{cases}0 & \text{表示任务 } t_k \text{ 已分配到处理机 } p_i\\ 1 & \text{表示任务 } t_k \text{ 未分配到处理机 } p_i\text{。}\end{cases}$

对某个确定的分配矩阵 $\boldsymbol{A}$，可得出已分配到 p_i 的所有任务的计算量之和为 $\sum_{k=1}^{n}r_k(1-a_{ki})$，处理机 p_i 的计算速度为 e_i，因此，p_i 用于计算的时间开销为 $g_i=\frac{1}{e_i}\sum_{k=1}^{n}r_k(1-a_{ki})$。

处理机 p_i 上的任务 t_k 同其他处理机上的任务之间的通信量之和为 $\sum_{q=1}^{n}c_{kq}a_{qi}$，即通信量矩阵 $\boldsymbol{C}$ 的第 k 行同分配矩阵 $\boldsymbol{A}$ 的第 i 列的点积。在上述通信量的计算中，由分配矩阵的定义，将任务 t_k 与同在处理机 p_i 上的其他任务之间的通信量计算为 0。因为分配到处理机 p_i 上的所有任务在分配矩阵 $\boldsymbol{A}$ 的第 i 列中的元素 $a_{qi}=0$，因此，在计算任务 t_k 的通信量时，将 t_k 与同在一台处理机上的任务之间的通信量排除在外了。这并不是指同一处理机上的任务之间没有通信量，而是因为同一处理机上任务之间的通信时间开销相对于不同处理机上任务之间的通信时间开销可以忽略不计，因此，通过分配矩阵的定义和映射，将同一处理机上的任务之间的通信量计算为 0。

若分配到处理机 p_i 上有 x 个任务 $t_1,t_2,\cdots,t_x$，那么 p_i 同其他处理机的通信量为这 x 个任务的通信量的和 $h_i=\sum_{k=1}^{x}\sum_{q=1}^{n}c_{kq}a_{qi}$。

这里,通信量 h_i 作为处理机 p_i 用于通信时间开销的估计,如果忽略处理机之间通信服务的额外时间开销不等对实际通信时间开销的影响,那么,处理机 p_i 执行完分配给它的所有任务的时间开销的估计为:

$$f_i = g_i + h_i = \frac{1}{e_i}\sum_{k=1}^{n} r_k(1-a_{ki}) + \sum_{k=1}^{x}\sum_{q=1}^{n} c_{kq}a_{qi}$$

上述任务调度优化问题是一个 NP 完全问题。本文提出一个负载均衡启发式优化算法(Load Balancing Heuristic Optimization Algorithm,LBHOA)对任务优化调度问题的求解进行讨论。

3　基于 LBHOA 的求解

3.1　LBHOA 的设计思路

在 LBHOA 算法中,状态空间的结点状态用 $S=(s_1,s_2,\cdots,s_m)$ 表示,其中,s_i 为分配到处理机 p_i 的任务集合,初始结点状态 $S_0=(\phi,\phi,\cdots,\phi)$。一个任务能且只能分配到一个处理机上,如果全部 n 个任务都已经分配,称为完全分配,由分配矩阵定义,完全分配满足 $\sum_{i=1}^{m}\sum_{k=1}^{n}(1-a_{ki})=n$;否则,称为不完全分配,任何一个完全分配的状态都是搜索树的一个叶结点。对于不完全分配的结点,可将一个尚未分配的任务逐一添加到任务集合 $s_1,s_2,\cdots,s_m$ 中,由此产生该不完全分配结点的 m 个后裔结点。

对任何一个结点 u,采用估计函数 $f_i(u)=g_i(u)+h_i(u)$,$i=1,2,\cdots,m$,分别计算出 m 个处理机的时间开销的估计,由于 m 个处理机并行处理,所以,结点 u 的时间开销估计值为 $f(u)=\max\{f_i(u)\mid i=1,2,\cdots,m\}$。

搜索求解的终止条件是找到完全分配的目标结点。根据任务调度的优化求解要求,找到的目标结点的时间开销估计值 $f(u)$ 应是所有完全分配结点中最小的。为了减少搜索求解的时空开销,LBHOA 算法采用启发式搜索方式,即每次分配一个任务时,从不完全分配的结点中选择估计值 $f(u)$ 最小的结点 u 进行扩展搜索。

3.2　LBHOA 的求解过程

LBHOA 的任务优化调度算法如下:

(1)对 n 个任务建立任务计算量向量 $\boldsymbol{R}=(r_k)_n$,r_k 为任务 t_k 的计算量,建立任务间通信矩阵 $\boldsymbol{C}=[c_{kq}]_{n\times n}$,$c_{kq}$ 为任务 t_k 同任务 t_q 之间的通信量,$c_{kq}=C_{qk}$,$c_{kk}=0$。对可使用的 m 个处理机建立计算速度向量 $\boldsymbol{E}=(e_i)_m$,e_i 为处理机 p_i 的计算速度;

(2)把起始结点 s 放入 open 表中,且令 s 的状态为 $S(s)=\{s_i=\phi\mid i=1,2,\cdots,m\}$,$s$ 的分配矩阵 $\boldsymbol{A}(s)=[a_{ki}]_{n\times m}$ 为全 1 矩阵,$f(s)=0$;

(3)如果 open 表是一个空表,则没有解,失败退出;

(4)从 open 表中选择一个估计值 f 最小的结点 ν 移到 closed 表。如果有几个最小 f 值的结点,则从中选取正分配任务数最多的结点作为结点 ν;若有几个符合要求的结点,则从中任选一个结点作为结点 ν;

(5)如果由结点 ν 的分配矩阵 $\boldsymbol{A}(\nu)$ 计算有 $\sum_{i=1}^{m}\sum_{k=1}^{n}(1-a_{ki})=n$,即 ν 是一个完全分配的目标结点,则求得一个解,即是 $S(\nu)$,输出最优解;

(6)将一个尚未分配的任务 t_k 分别放入到 $S(\nu)$ 的 $s_1,s_2,\cdots,s_m$ 中,产生 ν 的 m 个后继结点。每个后继结点的分配矩阵由其父结点 ν 的分配矩阵 $\boldsymbol{A}(\nu)$ 得出,若任务 t_k 放入 $S(\nu)$ 的处理机 p_i 的任务集合 s_i 中,则令 $\boldsymbol{A}(\nu)$ 中的元素 $a_{ki}=0$。对每一个后继结点 u,计算它的时间开销估计值 $f(u)$;

(7)计算结点状态下的每一台处理机的时间开销估计值 $f_i(u)=g_i(u)+h_i(u)=\frac{1}{e_i}\sum_{k=1}^{n} r_k(1-a_{ki})+\sum_{k=1}^{x}\sum_{q=1}^{n} c_{kq}a_{qi}$,其中 $i=1,2,\cdots,m$;再取 m 台处理机的时间开销估计值的最大值为结点 u 的时间开销估计值 $f(u)=\max\{f_i(u)\mid i=1,2,\cdots,m\}$;

(8) 将结点的所有扩展生成的后继结点放入 open 表,转到第(3)步。

4　实验结果及分析

为了验证算法的有效性,完成了一个基于 JAVA 的实现,并对其进行了仿真测试。实验平台如表 1 所示:

表 1　多处理机配置

Tab. 1　Composition of multi-processors

多处理机	机器类型	操作系统	管理软件
处理机 1	Dell PowerEdge 2600	RedHat Linux 9.0	LSF 6.0
处理机 2	HP Superdome Server SMP	RedHat Linux 9.0	LSF 6.0
处理机 3	HP Rx2600 集群	RedHat Linux 9.0	LSF 6.0

图 1 为异构多处理机在 LBHOA 部署前的 CPU 利用率统计情况；图 2 为该算法部署后多处理机的 CPU 利用率统计情况。

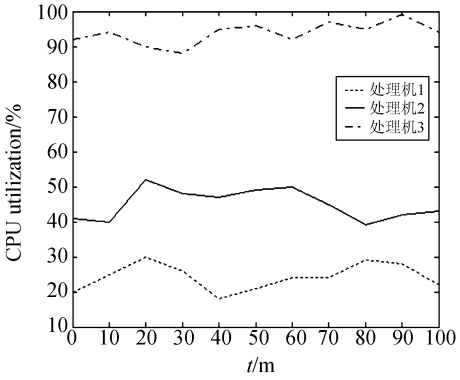

图 1　LBHOA 部署前 CPU 利用率统计

Fig. 1　CPU utilization rate before LBHOA

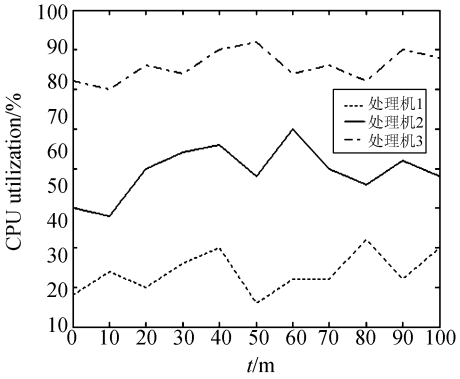

图 2　LBHOA 部署后 CPU 利用率统计

Fig. 2　CPU utilization rate after LBHOA

由图 1 可见，在 LBHOA 部署前，由于任务过多导致资源负载过重，处理机 3 的 CPU 利用率超过 92%，以致于用户无法继续提交任务；另一方面，处理机 1 和处理机 2 的利用率却过低，仅为 25%和 45%。图 1 中处理机 3 和处理机 1 的 CPU 利用率差别最大达到 67%，说明系统资源没有得到充分的利用。

由图 2 可见，在 LBHOA 部署后，处理机 3 的负载降低到 82%，同时处理机 1 和处理机 2 的利用率有了明显的提高，分别达到 60%和 72%，说明各个处理机的 CPU 负载趋于平衡。

图 3 和图 4 是 LBHOA 与 LTGA[3]（Linear Transformation Genetic Algorithm）、WLCA[4-5]（Weighted Least Connections Algorithm）等算法在平均应答延迟时间、平均任务完成时间等方面的对比。

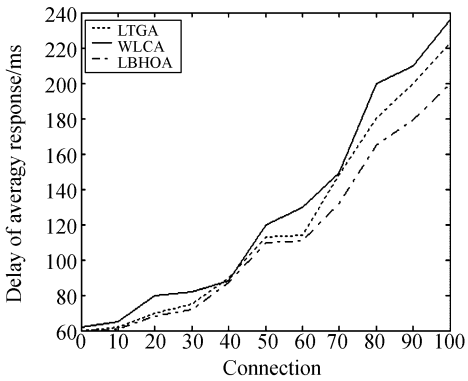

图 3　LBHOA、WLCA 与 LTGA 在平均应答延迟方面的比较

Fig. 3　Comparison of LBHOA, WLCA and LTGA in delay of average response

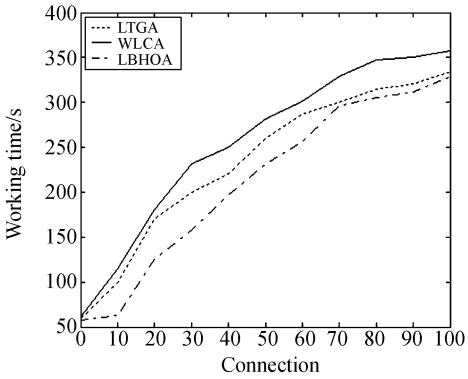

图 4　LBHOA、WLCA 与 LTGA 在任务完成时间方面的比较

Fig. 4　Comparison of LBHOA, WLCA and LTGA in working time of task

由图 3 可见,LBHOA 能有效地减少任务的平均应答延迟时间。随着客户端连接数量的不断增加,LBHOA 平均应答延迟时间比其它两种算法的应答延迟时间更小,说明它在处理异构多处理机之间的任务调度时,连接速度越来越快。

由图 4 可见,当数据规模增大时,LBHOA 在任务完成时间上一直表现出较好的性能,说明采用 LBHOA 带来的时间收益仍高于处理负载信息的时间开销。

5　结　论

为了使异构多处理机系统在系统资源均衡分配的同时使系统资源利用率最优,本文分析了异构多处理机系统中的负载均衡和任务调度参数,讨论了异构集群任务调度模型,提出了一个负载均衡启发式优化算法 LBHOA。LBHOA 采用启发式搜索策略,即每次分配一个任务时,从不完全分配的结点中选择估计值最小的结点进行扩展搜索,直到找到完全分配的目标结点,且目标结点的时间开销估计值是所有完全分配结点中最小的,满足了异构集群系统中的负载平衡和优化调度的需要。

在异构多处理机系统环境下的仿真实验数据表明:LBHOA 显著地降低了资源最优分配中的计算复杂度,使不同处理机利用率的差别不超过 22%,负载趋于均衡;与 WLCA 和 LTGA 相比,LBHOA 的平均应答延迟时间的开销减少了 10%,任务完成时间的开销减少了 15%,提高了异构多处理机系统的并行处理性能。下一步的研究工作中,将继续探讨基于并行任务的计算量、任务之间的通信量和异构处理机速率的动态负载均衡及任务调度问题。

参考文献:

[1] SINNEN O, SOUSA L A, SANDNES F E. Toward a realistic task scheduling model [J]. *Parallel and Distributed Systems, IEEE Transactions on Parallel and Distributed Systems*, 2006,17(3):263-275.

[2] 刘红,白栋,丁炜.应用于 MPLS 网络负载均衡的启发式自适应遗传算法研究[J].通信学报, 2003, 24(10): 39-45.
LIU H, BAI D, DING W. A heuristic adaptive genetic algorithm for load balancing in MPLS networks [J]. *Journal of China Institute of Communications*, 2003, 24(10):39-45. (in Chinese)

[3] SHU W N, ZHENG S J. A parallel genetic simulated annealing hybrid algorithm for task scheduling[J]. *Wuhan University Journal of Natural Sciences*,2006,11(5):1378-1382.

[4] BESTAVROS A, CROVELLA M E, LIU J, *et al*.. Distributed packet rewriting and its application to scalable server architectures [C]. *Proceedings of 6th IEEE International Conference on Network Protocols*, 1998: 290-297.

[5] DIAS M, KISH W, MUKHERJEE R, *et al*.. A scalable and highly available web servers[C]. *Proceedings of 41st IEEE Computer Society Intl. Conference*, 1997:85 - 92.

作者简介:童小念(1954—),女,湖北武汉人,中南民族大学计算机科学学院副教授,硕士生导师,主要从事计算机系统结构和多媒体技术等方面的研究。E-mail: tongxiaonian@yahoo.com.cn