Master/Slave 结构下可保证边竞争 QoS 限制的资源分配策略

杨 娟 赖祥伟 邱玉辉

(西南大学计算机与信息科学学院 重庆 400715)[1] (西南大学智能软件与软件工程实验室 重庆 400715)[2]

摘 要 任务调度作为分布式系统中提高系统并发处理的关键一直受到很多关注,随着分布式系统规模的扩大以及分布式系统中所处理任务数的增多,这个 NP 问题很多只能依靠启发式搜索技术获得近似最优解,然而这些算法中大都忽略了分布式系统中的一些实际问题,如通信竞争问题。已有的关注通信竞争的理论要么不适用于多任务的实时分布式系统,要么最终的任务分配无法实现整体 makespan 值最小(min-max)。本文提出了一个以 master/slave 为支撑结构,以最小化系统整体响应时间为代价函数的算术模型 MMP。MMP 既考虑了通信竞争问题,也考虑了多任务事实分布式系统的特征:任务数量多,任务间相互独立,任务以一定频率到达,任务执行受 QoS 限制。

关键词 master/slave,min-max,通信竞争,QoS 限制

A Resource Allocation Policy Which can Guarantee the Communication Contention in QoS Limitations under Master/Slave Architecture

YANG Juan LAI Xiang-Wei QIU Yu-Hui

(Faculty of Computer and Information Science, Southwest University Chongqing 400715)[1]

(Intelligent Software and Software Engineering Laboratory, Southwest University Chongqing 400715)[2]

Abstract An effective way to enhance the concurrency capability of the distributed system is to appropriately allocate the tasks to the processors. However, it is harder and harder to get the optimal solutions along with the expanding scale of the systems. Although many heuristic researching based policies are occurred to get the approximating results instead of the optimal ones, few of them are considered under the reality situations, many specifications have been omitted, such as the communication contention. Some theory backgrounds have been built up to support the communication contention which is not suitable for the real time systems handling with large number of tasks, or they can't minimize the maximum makespan. We built up a mathematical model MMP to solve the problems refered before, which not only get the communication contention into considering, but also get the min-max results without breaking the QoS limitations.

Keywords Master/Slave, Min-max, Communication contention, QoS limitation

1 引言及研究背景

异构环境下如何提高系统的并发处理性能已成为分布式系统领域的一个重要课题,而任务调度作为并行编程的一个基础方面更是受到了广大学者的关注。随着分布式系统规模的扩大以及分布式系统中所处理任务数的增多,使得原本就是 NP 难度问题[1,2]的经典问题更加难以获得一个最优解,因而出现大量以启发式搜索技术为支撑的算法。这些算法期望在一个可接受的时延内获得一个较为满意的近似解来取代最优解。这些算法中虽然很多可以得到一个较为理想的结果,但大都基于一个较为理想的计算环境,而忽略了一些实际问题,例如大都忽略了网络中通信竞争问题[3~8]。只有少数算法将通信竞争考虑在其问题构建中[9~11]。通信竞争的理论在文[12]中被 Oliver Sinnen 等人所提出。

文[12]提出了一个既考虑了通信竞争又可以处理 DAG 任务图的集中式任务调度策略,然而这种策略是不适用于多任务实时分布式系统(如 minigrid)环境的。

我们在文[3]中提出了一种事实系统中统一任务调度的通用算术模型,这个模型同样只能应用于小型分布式系统且也未考虑竞争问题。

文[4]中提出了一种基于扩散收敛的调度策略。通过设置收敛因子进行邻居节点间的任务传递控制,当收敛因子收敛时说明系统整体已获得最大收益即负载均衡。这种收敛策略虽然适用于多任务实时分布式系统,但这种基于 master/slave 结构的任务调度策略同样也没有考虑通信竞争问题。

文[5]提出了一个 minigrid 环境下的分布式任务调度策略,这种策略是一种收敛扩散策略的改进。该策略将每个需要处理的任务封装在一个具有学习机制的移动 agent 内,实现其自动迁移和登录。这种策略使得任务均衡调度在一个真正异构环境下成为可能,但它同样忽略了边竞争问题,且若要使得所有任务在系统中真正均衡分布,移动 agent 必须准确掌握全局信息,否则会导致其学习过程出现偏差,最终无法实现稳定。

文[13]是一个应用于 master/slave 结构的任务均衡模型,它既考虑了边竞争问题,也使得节点在只掌握局部信息的前提下获得全局稳定。该文构建了一个代价函数为最大化系统单位时间吞吐量的模型。

综上所述,我们构建了以 master/slave 为支撑结构,以最小化系统整体响应时间为代价函数的算术模型 MMP。master/slave 结构是在支撑网络中考虑多个不同的节点分别扮演

调度器的角色，进行最优调度。在我们提出的算术模型中既考虑了文[12]提出的边竞争问题，也考虑了多任务事实分布式系统的特征：任务数量多，任务间相互独立，任务以一定频率到达，任务执行受 QoS 限制。本文的主要贡献在于：(1)在 master/slave 结构下充分考虑了边竞争问题；(2)在满足支撑网络结构 QoS 限制的前提下实现了目标函数的最优(QoS 为任务流量控制在系统最大吞吐量内且整体任务完成时间不会超过系统能忍受的最大时延)；(3)目标函数为任务流整体响应时间 makespan 最小(通常称为 Minimize the Maximum makespan，Min-Max)，而不是传统的最大化系统单位时间吞吐量或是获得系统整体稳定状态。另外，我们给出了完备的数学证明，并且给出了一个实例进行说明。

2 MMP 策略(Min-Max Policy in Master/Slave Structure)

2.1 构建 MMP 策略的网络环境模型

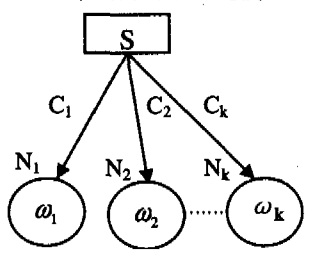


图 1 Fork 结构图

MMP 策略应用的任务性质为任务间无依赖关系的单任务，且任务规模一致。因此任务在一段时间内的任务到达数量变化情况可用任务到达频率 λ 的变化来模拟。单位时间(Unit)系统能处理的最大任务数为 R，因此 $\lambda \leqslant R$。MMP 策略所依赖的 master/slave 网络结构我们用单端口有向 Fork 结构图 G 模拟，如图 1 所示。$G=(S,E,C,N,\omega)$，其中 S 代表交换节点，$E=\{e_i \mid 0<i<k\}$ 代表边集合，$C=\{c_i \mid 0<i<k\}$ 代表边通信耗费集合，$N=\{N_i \mid 0<i<k\}$ 代表处理器集合，$\omega=\{\omega_i \mid 0<i<k\}$ 代表处理器 N_i 的处理性能指标集合。

N_i	$N_i \in N, i \in [1,k]$，表示处理器节点
S	表示交换节点
e_i	$e_i \in E, i \in [1,k]$，表示从交换节点 S 指向处理节点 N_i 的通信链路
c_i	$c_i \in C, i \in [1,k]$，表示从交换节点 S 指向处理节点 N_i 的通信链路的通信耗费
R_i	分配到节点 N_i 上的任务总数
R	系统单位时间能处理的任务总数(QoS)，$R=\text{maximize}\sum_{i=1}^{k} R_i$
t_i	为处理节点 N_i 上进行计算的时间片
α_i	从 S 传递任务处理节点 N_i 上的时间片，且 $\alpha_i=\frac{t_i}{\omega_i} \cdot c_i$

选用 Fork 结构作为本文研究拓扑结构背景的原因在于 Fork 结构是大部分拓扑结构的基础，无论是规则的拓扑结构(如有向无环图)还是不规则的拓扑结构(mesh 网络)，我们都可以将其拆分成 Fork 结构。以立方体 mesh 网为例，节点 N_{ijk} 的邻居节点为 N_{ijk}，$N_{i'jk}$，$N_{ij'k}$，k' 为与 k 仅有 $\pm l \bmod k$ 差别的地址，i' 为与 i 仅有 $\pm l \bmod i$ 差别的地址，j' 为与 j 仅有 $\pm l \bmod j$ 差别的地址。因此可构建以 N_{ijk} 为 master 节点的 Fork 结构图。但如何将拓扑结构图拆分成 Fork 结构图本身也是一个 NP 难度问题[13]且已超出本文讨论的范围，这里不再详述。

2.2 MMP 策略的问题构建

引入辅助问题 $\text{maximize}\sum_{i=1}^{k} R_i$，$R_i$ 是 N_1-N_k 计算节点上的任务总数。有限制条件：

$$R_i \leqslant \frac{1}{\omega_i}, 0 \leqslant i \leqslant k \tag{1}$$

$$\sum_{i=1}^{k} R_i \cdot c_i \leqslant 1 \tag{2}$$

可由文[13]得知，其解为 $R=\sum_{i=1}^{p}\frac{1}{\omega_i}+\frac{\varepsilon}{C_{p+1}}$，为单位时间内系统所能承受的最大任务量。其中 p 为使得 $\sum_{i=1}^{p}\frac{c_i}{\omega_i} \leqslant 1$ 满足的最大索引值，即有 $\sum_{i=1}^{p+1}\frac{c_u}{\omega_i}>1$。具体过程见文[13]。

现假设任务流量频率为 λ，有 $\lambda \leqslant R$，且在如 2.1 节所描述的网络拓扑结构中，N 个计算节点总是保证运行，则系统整体响应时间为 t。为求得任务流频率为 λ 的任务整体响应最短时间，可定义线性规划问题 minimize(t)。限制条件如下：

$$t=\max\{t_1,t_2,\cdots,t_k\} \tag{3}$$

$$\frac{t_i}{\omega_i}=\frac{\alpha_i}{c_i} \tag{4}$$

$$\sum_{i=1}^{k}\frac{t_i}{\omega_i}=\lambda \tag{5}$$

$$\sum_{i=1}^{k}\alpha_i \leqslant t \tag{6}$$

$$0 \leqslant \alpha_i \leqslant t, 0 \leqslant i \leqslant k \tag{7}$$

其中 t_i 是各个节点的执行时间，α_i 是 N_0 向 N_i 节点发送任务的时间片。可得出该线性规划问题的一个解为 λ/R，且此时对应的资源分布为 $(\frac{\lambda}{R}\cdot\frac{1}{\omega_1},\cdots\frac{\lambda}{R}\cdot\frac{1}{\omega_p},\frac{\lambda}{R}\cdot\frac{\varepsilon}{C_{p+1}})$，其中 p 为使得 $\sum_{i=1}^{p}\frac{c_i}{\omega_i} \leqslant 1$ 满足的最大索引值，$\varepsilon=1-\sum_{i=1}^{p}\frac{c_i}{\omega_i}$。

3 证明

现证明 λ/R 是该问题的最优解。

(1)首先证明它是一个解，即 λ/R 满足上述 5 个限制条件。

①证明 $t=\lambda/R$ 满足限制条件③

因为 $t_i=\lambda/R, 1 \leqslant i \leqslant p, t_{p+1}=\frac{\lambda}{R}\cdot\varepsilon$，且有 $t_i=0$，当 $p+1 \leqslant i \leqslant k$ 且 $p+1<k$。所以，$t=\max\{t_1,\cdots,t_k\}=\max\{\frac{\lambda}{R},\cdots\frac{\lambda}{R}\cdot\varepsilon,0,0,\cdots 0\}=\frac{\lambda}{R}$。因此满足限制条件③。

②证明 $t=\lambda/R$ 满足限制条件④

假设④不成立，即 N_0 传递到 N_1 的任务数补等于 N_1 执行的任务数，$\alpha_i \neq \frac{t_i}{\omega_i}\cdot c_i$ (8)

而根据 p 的定义可知 $\sum_{i=1}^{p}\alpha_i+\varepsilon\cdot t=1$ (9)

将 $\alpha_i=\frac{t_i}{\omega_i}\cdot c_i$ 代入(9)式，根据假设则应得

$$\sum_{i=1}^{p}\frac{t_i}{\omega_i}\cdot c_i+\varepsilon\cdot t \neq t \tag{10}$$

而(10)式左边 $\sum_{i=1}^{p}\frac{t_i}{\omega_i}\cdot c_i+\varepsilon\cdot t=\sum_{i=1}^{p}\frac{t_i}{\omega_i}\cdot c_i+(1-\sum_{i=1}^{p}\frac{c_i}{\omega_i})\cdot t=t$，与假设(3)式矛盾。所以，$t=\lambda/R$ 满足限制条件④。

③证明 $t=\lambda/R$ 满足限制条件⑤

因为 $\sum_{i=1}^{k}\frac{t_i}{w_i}=\sum_{i=1}^{p}\frac{t_i}{w_i}+\frac{\epsilon}{C_{p+1}}=t\cdot(\sum_{i=1}^{p}\frac{1}{\omega_i}+\frac{\epsilon}{C_{p+1}})=t\cdot R=\frac{\lambda}{R}\cdot R=\lambda$，满足限制条件⑤。

④证明 $t=\lambda/R$ 满足限制条件⑥

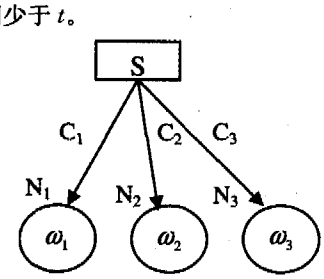
因为 $\sum_{i=1}^{k}\alpha_i=\sum_{i=1}^{p}\alpha_i+\epsilon\cdot t=t$，所以满足限制条件⑥。

⑤证明满足限制条件⑥

因为 $\alpha_i=t_i/\omega_i$，当 $1\leqslant i\leqslant p$ 时有 $\alpha_i=t\cdot\frac{c_i}{\omega_i}$。根据 p 的定义有 $\sum_{i=1}^{p}\frac{c_i}{w_i}\leqslant 1$，即 $\frac{c_i}{w_i}\leqslant 1$，即 $t\cdot\frac{c_i}{w_i}\leqslant t$，所以 $\alpha_i\leqslant t$。当 $i>p$ 时，若 $p+1<k$ 有 $\alpha_i=t\cdot(1-\epsilon)\leqslant t$，$i=p+1$ 或 $\alpha_i=0$，$k\geqslant i>p+1$，因此满足限制条件⑦。

(2)证明 $t=\lambda/R$ 是最优解。

假设存在一种资源分布 $R'=(R'_1,R'_2,\cdots R'_k)$，使得系统整体响应时间 $t'<t=\lambda/R$。因为 $t'\neq t$，所以 $R'=(R'_1,R'_2,\cdots R'_k)$ 与 $(\frac{\lambda}{R}\cdot\frac{1}{\omega_1},\cdots\frac{\lambda}{R}\cdot\frac{1}{\omega_p},\frac{\lambda}{R}\cdot\frac{\epsilon}{C_{p+1}})$ 不尽相同，但仍满足 $\sum_{i=1}^{k}R'i=\lambda$。假设其中有 $R'_i=\frac{\lambda}{R}\cdot\frac{1}{\omega_i}-\delta$，则必有 $R'_j=\frac{\lambda}{R}\cdot\frac{1}{\omega_1}+\delta$，则有 $\alpha_j=R'_j\cdot c_j=(\frac{\lambda}{R}\cdot\frac{1}{\omega_j}+\delta)\cdot c_j=\frac{\lambda}{R}\cdot\frac{1}{\omega_j}\cdot c_j+\delta\cdot c_j$，而 $\alpha_j=\frac{t'_j}{\omega_j}\cdot c_j$，$t'_j\leqslant t'$，即 $\alpha_j=\frac{\lambda}{R}\cdot\frac{1}{\omega_j}\cdot c_j+\delta\cdot c_j=\frac{t'_j}{\omega_j}\cdot c_j$，则有 $t'_j-t=\delta\cdot\omega_j>0$，即 $t'_j>t$ 与假设 $t'_j\leqslant t'<t$ 矛盾。因此，不存在一个分布 $R'=(R'_1,R'_2,\cdots R'_k)$ 可以使得其整体响应时间少于 t。



示例拓扑结构图

4 实例

在 120 个 unit 内，若有 40 个任务到达系统，求系统的最小整体响应时间(min-max)。

为简单起见，假设系统中处理器只有 3 个(理论上处理器个数可以有无限多个，但在系统 QoS 限制下，只需要使用 3 个)，S 代表没有处理能力的交换设备。3 个处理器的处理性能分别是 $\omega_1=8,\omega_2=5,\omega_3=4$，而 3 个节点的通信性能恰好与其计算性能相反，分别是 $c_1=1,c_2=2,c_3=3$。根据 Master-Slave 算法，任务分配以通信性能为基础，因此可求得 $R=\sum_{i=1}^{p}\frac{1}{\omega_i}+\frac{\epsilon}{C_{p+1}}=\frac{1}{8}+\frac{1}{5}+\frac{1-\sum_{i=1}^{p}\frac{c_i}{\omega_i}}{3}=\frac{29}{60}$，即每 60 个 unit 系统最多可处理 29 个任务。当 $\lambda<R$ 时，即每 120 个 unit 系统最多可处理 58 个任务。此时 $t=\lambda/R=20/29$，其对应于 N1、N2、N3 三个节点的分布为(5/58,4/29,19/174)。其中 $\lambda=1/3$，表示 120 个 unit 内只有 40 个任务到达，求此时系统的 Min-Max 解。因为已求得分布(5/58,4/29,19/174)，即对应于 120 个 unit 有 $T_{N1}=15\cdot\frac{\lambda}{R}\approx 10$，$T_{N2}=24\cdot\frac{\lambda}{R}\approx 17$，$T_{N3}=19\cdot\frac{\lambda}{R}\approx 13$，此时 $\sum_{i=1}^{3}T_{Ni}=10+17+13=40$。可求得实际计算时间为 $\max\{10\times 8,17\times 5,13\times 4\}=85$，而理论值为 $\frac{\lambda}{R}\cdot 120\approx 83$，两者基本相等。

结论 本文以线性规划的方式构建了一个可应用于多任务实时分布式系统的算术模型，这个模型中既考虑了支撑环境的边竞争问题，也实现了系统整体响应时间 makespan 最小的代价函数(Min-Max)，且保证了支撑网络结构的 QoS 限制即任务流量控制在系统最大吞吐量内且整体任务完成时间不会超过系统能忍受的最大时延下求得最优解。另外我们对这个算术模型给出了完备的数学证明，且通过一个实例对其进行了说明。

参 考 文 献

1 Sarkar V. Partitionning and Scheduling Parallel Programs for Execution on Multiprocessors. MIT Press,1989

2 Garey M R, Johnson D R. Computers and Intractability, A Guide to the Theory of NP-Completeness. Freeman W H and Company, 1991

3 Yang J, Bai Y, 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, 2005. 1099~1105

4 Rotaru T, Nageli H H. Dynamic Load balancing by Diffusion in Heterogeneous Systems. J Parallel and Distributed Computing, 2004,64:481~497

5 Liu J, Jin X L, et al. Agent-Based Load Balancing on Homogeneous Minigrids: Macroscopic Modeling and Characterization. IEEE Transactions on Parallel and Distributed Systems, 2005,16(7): 586~598

6 Ali S, Kim J K, et al. Greedy Heuristics for Resource Allocation in Dynamic Distributed Real-time Heterogeneous Computing Systems. PDPTA'02 International Conference

7 Ravindran B, Li P, et al. Proactive Resource Allocation for Asynchronous Real-Time Distributed Systems in the Presence of Processor Failures. J Parallel and Distributed Computing, 2003, 63:1219~1242

8 Georgiadis L, Nikolaou C, et al. A Fair Workload Allocation Policy for Heterogeneous Systems. J Parallel and Distributed Computing,2004, 64:507~519

9 Kwok Y K, Ahmad I. Link Contention-Constrained Scheduling and Mapping of Tasks and Messages to a Network of Heterogeneous Processors. Cluster Computing: Networks, Software Tools, and Applications,2000,3(2):113~124

10 EI-Rewini H, Lewis T G. Scheduling Parallel Program Tasks onto Arbitrary Target Machines. J Prallel and Distributed Computing, 1990,9(2)138~153

11 Sinnen O, Sousa L. List Scheduling: Extension for Contention Awareness and Evaluation of Node Priorities for Heterogeneous Cluster Architectures. Parallel Computing,2004,30(1):81~101

12 Sinnen O. Sousa L Communication Contention in Task Scheduling. IEEE Transactions on Parallel and Distributed Systems, 2005,16(6):503~515

13 Banino C, Beaumont O, et al. Scheduling Strategies for Master-Slave Tasking on Heterogeneous Processor Platforms. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(4): 319~330