一种同构机群系统中的处理机分配算法*

温钰洪 王鼎兴 沈美明

(清华大学计算机系 北京 100084)

摘要 机群系统的分布式计算环境为并行处理技术带来了新的研究与应用问题,正成为并行计算的热点问题.如何合理、有效地将并行任务划分到机群系统的结点上,将直接影响系统的执行性能.本文分析影响系统执行效率的执行开销因素,同时提出一个启发式的处理机分配算法.

关键词 处理机分配,工作站机群系统,系统性能评价函数.

随着计算机体系结构以及通讯技术的发展,在并行处理技术中,采用高带宽(High Bandwidth)、低延时(Low Latency)的通讯网络,以高性能的工作站机群系统(Workstation Cluster System)作为并行计算的平台已越来越多地引起了人们的重视.同时,也相应出现了支持这一模式工作的一系列软件工具环境,如Express[1], PVM[2]等.然而,如何利用机群系统的特点,有效地完成并行任务计算,提高并行计算加速比,却是机群系统中有待解决的关键问题.

在并行计算系统中,如何将并行任务划分到各处理机结点上执行是最基本,也是最关键的一步,它直接影响到并行任务计算的效率.尽管动态的处理机分配策略有着执行性能上的优势,但静态的分配策略却由于操作控制上的简单性,在系统中具有易实现的优点.[3,4]

静态处理机分配及在分布式并行系统上的任务映射的主要解决方法是通过对于并行子任务计算量以及子任务间通讯量的预测,以系统性能最优作为目标,通过计算任务分配后的系统执行开销的办法,求得任务与处理机的映射,使系统开销最小.[5-7]然而,在机群系统中,由于工作站结点通过传统的网络进行拓扑连接,它区别于 MPP 系统,因而需要研究针对于机群系统特点的处理机分配策略与算法.

本文首先分析并行任务中子任务间的通讯对于整个并行执行效率的影响,指出处理机通讯是影响机群系统并行执行性能的主要因素,然后提出针对于机群系统并行计算的处理机分配策略及算法,以此作为提高机群系统并行计算性能的基础.

---

* 本文研究得到国家 863 高科技项目基金资助.
作者**温钰洪**,1967 年生,博士,讲师,主要研究领域为并行程序设计环境.**王鼎兴**,1937 年生,教授,主要研究领域为分布/并行处理技术.**沈美明**,1938 年生,教授,主要研究领域为并行/分布和智能计算机系统.

本文通讯联系人:王鼎兴,北京 100084,清华大学计算机系

本文 1996-03-07 收到修改稿

1 机群系统结构及处理机分配目标

机群系统的构造主要分为2种类型:①利用现有的高性能局域网络,将工作站连成并行计算环境,以并行程序环境加以支持,如利用Ethernet网进行连接(如图1所示);②提供机群系统连接的硬件支持,以提高工作站结点间的通讯效率,如SCI[8],Bit3[9],ATM[10]等.其主要目的在于通过低延时的网络通讯,提高其并行计算性能,结构如图2所示.从表1中可看出2类实现方法在通讯性能上的差别.

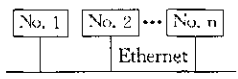

图1 基于传统局域网的机群系统

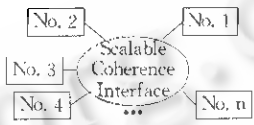

图2 SCI连接的机群系统

表1 机群系统中网络实现性能比较

	Latency (ms)	Transmision Speed (Mb/s)
RS/6000－Ethernet	3.5	0.05～0.1
RS/6000－FDDI	<1	4
RS/6000－Bit3	0.24	10
RS/6000 IBM V7	0.14	3～5
NEctar / ATM	0.35	100

利用机群系统对信号处理、数值天气预报以及地震信号处理的应用中,我们发现,对于同一应用程序,随着处理机结点数的增加,其加速比效率下降较大.这主要是由于将并行任务划分到多个结点上执行之后,处理机之间的通讯量迅速增加所引起的.[11,12]设一个并行应用程序的计算量为M,其处理速度为S,则单处理机上的时间是$T=M/S$.

如果将其划分在N个处理机上执行,则各处理机的计算量为$M/N+C$,其中C为处理机间的通讯量,其执行时间为

$$T=\frac{\frac{M}{N}+C}{S}=\frac{M+C\times N}{S\times N}=\frac{M}{S\times N}+\frac{C}{S}$$

从机群系统并行计算的可扩展性上分析,根据ISO-efficiency的定义[13],P个处理机完成同一任务时

$$pTp=\sum_{i=0}^{p-1}te^i+\sum_{i=0}^{p-1}to^i=Te+To$$

其中Te是串行执行时间,To为处理机间的通讯时间,并行计算的$Speedup$:

$$Speedup=Te/Tp=((Te/(Te+To))*p)=p/(1+To/Te)$$

Efficiency定义为: $E=Speedup/P=1/(1+To/Te)$

ISO-efficiency关心的是要保持系统的E的前提下,当处理机数增加时,应该加大多少的任务量;相反地,实际应用的情况往往是应用程序任务量一致,而采用多个处理机来并行计算,即Te一定.因此,处理机分配的目的则在于如何通过合理的处理机与并行子任务的映射,从而减少To,提高E的值.

2　处理机分配问题

2.1　处理机分配问题描述

对于机群系统中的处理机分配问题，可以描述为：

对给定的含有 m 个并行子任务 $t_1,t_2,\cdots,t_m$ 的应用程序，在有 n 个处理机的分布并行机群系统中，对每个执行子任务 t_i，分配一个处理机 p_i，t_i 在 p_i 上加载执行，要求整个子系统并行执行效率最佳.

定义 2.1. 定义映射 $\pi:v\to n$ 为并行子任务到处理机结点上的分配，

$$v_i\in\{t_1,t_2,\cdots,t_m\},\ \pi(v)\in\{p_1,p_2,\cdots,p_n\}$$

定义 2.2. $X(u,i)$表示将子任务 u 分配到结点 i 上执行时的执行开销，$1\leqslant i\leqslant n$.

定义 2.3. $c(u,v,i,j)$表示子任务 u 分配到结点 i 中，v 分配到结点 j 中时，两个处理机执行时的通讯开销，$1\leqslant i,j\leqslant n$.

对于含有 m 个并行子任务，在有 n 个处理机结点的机群系统上的分配映射，$\pi:v\to[n]$，各个处理机上的执行开销为：

$$\begin{aligned}cost1&=\sum_{u\in V}(X(u,p(u))+\sum_{u\neq v}C(p(u),p(v),u,v))\\&=\sum_{u\in V}X(u,p(u))+\sum_{u\in V}\sum_{u\neq v}C(p(u),p(v),u,v)\end{aligned}$$

在整个机群系统中，执行一个并行应用程序，则要求系统开销：

$$cost2=\max\{\sum_{\pi(u)=k}X(u,k)+\sum_{\pi(u)=k}\sum_{u\neq v}C(\pi(u),\pi(v),u,v)\}\quad 1\leqslant k\leqslant n$$

为最小，以保证将应用程序的并行子任务映射于处理机中，其执行的效率最佳，并行计算加速比最大.

2.2　机群系统处理机分配问题分析

在并行应用程序的执行中，任务的完成取决于各子任务分别在各处理机上的执行时间 $X(u,i)$(任务 u 在结点 i 上的执行时间)以及子任务执行时与其它处理机结点上的任务间的通讯时间 $c(u,v,i,j)$.

在我们研究的同构机群系统上的处理机分配问题中，由于各处理机结点都是采用同一类型的处理机的工作站结点，如 RS6000，SPARC 10，SPARC 2 等. 它们具有处理能力一致的特点(在考虑各站点负载计算都一样的前提下)，因而，各子任务的执行时间就可以由子任务中的程序代码的计算量来衡量计算. 这可以通过编译器在用户代码的编译过程中得到，是一个静态的参量.

另一部分程序执行时间为子任务间的通讯开销. 应用程序中，各子任务间的通讯量是由应用程序问题本身决定的. 但在处理机分配中，对各子任务在处理机上的映射，则可能会得到完全不同的结果. 特别是在对于象具有 SCI(scalable coherence interface)硬件支持的机群系统中，由于它能实现相邻结点间的局部通讯与网络本身通讯的相容性，因而如何将并行子任务分配映射于处理机结点中，将大大影响到应用程序的并行计算效率.

在应用程序的执行中，子任务间通讯以及对执行性能的影响主要分为如下几类：①同步等待：应用程序中的同步点设置，等待各任务都执行到相应的程序点；②数据依赖：一处理机

上的任务执行需要另一处理机上任务计算所产生的结果及中间数据;③执行完成等待:当一处理机上子任务完成计算后,等待其它处理机上子任务的计算结束,以结束整个并行应用程序的执行.因而,这需要在处理机分配中,一方面考虑系统开销最小,同时考虑处理机上负载的平衡.

定义 2.4. 对于并行应用程序中的 m 个并行子任务,定义其计算量分别为 $Q_1, Q_2, \cdots, Q_m$.

定义 2.5. 同构机群系统中,各个处理机执行速度一致,因而各子任务的计算时间分别为 $t_i = Q_i / Speed$.

定义 2.6. 矩阵 $C_{m \times m}$ 表示各并行子任务间的数据通讯量.

$$C = \begin{bmatrix} 0 & c12 & . & . & c1m \\ c21 & 0 & . & . & c2m \\ . & & . & & \\ . & & & . & \\ cm1 & cm2 & . & . & 0 \end{bmatrix}_{m \times m}$$

在处理机分配中,如果子任务 Ts_i 与 Ts_j 分配于同一处理机中,为简便起见,设其通讯量为 0,即 Ts_i 与 Ts_j 对处理机 PE_k 只产生计算量开销 $f = t_i + t_j$.

定义 2.7. 矩阵 $Sn \times n$ 代表各处理机之间的 Message Passing 通讯处理速度.

$$S = \begin{bmatrix} 0 & s12 & . & . & s1n \\ s21 & 0 & . & . & s2n \\ . & & . & & \\ . & & & . & \\ sn1 & sn2 & . & . & 0 \end{bmatrix}_{n \times n}$$

在并行子任务的处理机分配映射算法执行中,首先对各子任务按其开销 $cost_i = t_i + \sum_{j=1}^{m} c_{ij}$ 进行递减排序,以保证先处理系统开销较大的子任务.

2.3 处理机分配策略

2.3.1 无通讯任务

我们首先考虑子任务不相关的处理机分配问题.这类问题中没有通讯开销,因而 $c(i,j) = 0$,这样处理机分配即简化为寻找一个分配矩阵 A,使:

$$cost = \max_{j=1,2,\cdots,n} \left(\sum_{i=1}^{n} X(i) A_{ij} \right)$$

最小,其中

$$A_{ij} = \begin{cases} 1, & \text{如果任务 } i \text{ 分配于处理机 } j \text{ 中} \\ 0, & \text{如果任务 } i \text{ 不分配于处理机 } j \text{ 中} \end{cases}$$

这类简化问题的处理机分配,在同构机群系统中,由于各处理机处理能力相同,因而只要按各子任务的计算量评估,在各处理机结点中,尽可能平均地分配映射,满足结点上负载的平衡即能使并行应用程序的并行计算获得最佳的执行效率.

2.3.2 具有通讯的任务

以下讨论具有子任务间通讯开销的并行应用程序.在同构机群系统上的处理机分配问题,以使整个系统中具有最小的执行开销.

对于具有通讯开销的任务分配，处理机除了完成所分配任务的计算之外，还需要向相关结点发送与接收数据的通讯开销. 每个子任务 t_i，对于所分配的处理机结点将包含如下 2 方面的开销：

①子任务的执行时间 $X(i)$；

②子任务与所有其它子任务间的通讯执行时间(没有分配在同一处理机结点中)，

$$\sum_{j=1}^{n} C(i,j)A_{ik}(1-A_{jk})$$

因而，对于所有分配于某一处理机结点中的子任务，其总的执行开销为：

$$W=\sum_{i=1}^{n}(X(i)+\sum_{j=1}^{n}C(i,j)A_{ik}(1-A_{jk}))$$

并行子任务在处理机上的分配过程中，对于处理机分配矩阵 A，当分配到第 t 步时，设为 $A(T)$，则对于处理机结点 K 上的执行开销可计算为：

$$f_1(k,A^{(T)})=f_1(k,A^{(T-1)})+X(i,j)+\sum_{j=1}^{N}\sum_{m=1}^{K}C(i,j)A_{jm}^{(T)}(1-A_{jk}^{(T)})$$

f_1 表示结点 k 中的执行开销下限，只包含第 t 步分配时结点上的执行开销以及它与已经分配的子任务在其它结点上之间的通讯开销，不包括未分配任务与它的影响.

$$f_u(k,A^{(T)})=f_u(k,A^{(T-1)})+X(i,j)+\sum_{j-1}^{N}C(i,j)(1-A_{jk}^{(T)})$$

f_u 表示结点上的执行开销上限，其中包括与所有未分配的任务间的可能的通讯开销.

在以下的处理机分配算法中，为简化处理机分配的实现过程(处理机分配是一个 NP 完备问题[5,9])，我们将采用：

$$f(k+1,A^{(T)})=f(k+1,A^{(T-1)})+X(i,j)+\sum_{j=1}^{N}\sum_{m=1}^{K}C(i,j)A_{jm}^{(T)}(1-A_{jk}^{(T)})$$

作为算法实现时结点开销计算的启发式评价函数，来完成子任务在同构机群系统上的处理机分配及映射.

3　处理机分配算法

3.1　算法设计

依据前述的处理机分配策略，我们得到以下的处理机分配算法：

Begin

1. 将各子任务按其执行开销评估的递减排序，排序依据为：$S=X(i)+\sum_{j-1}^{N}C(i,j)$
2. 初始化各处理机结点的执行开销，$f_i=0$；
3. For $i=1$ To N Do　　/* 处理每个子任务 */

Begin

(1) 预置任务 t_i 分配的开销最大值，$f=$MAXAM；

(2) For $j=1$ To M Do　　/* 评价各个处理机 */

Begin

①假设任务 t_i 分配于处理机中，计算结点开销：

$$f(k+1,A^{(T)})=f(k+1,A^{(T-1)})+X(i,j)+\sum_{j-1}^{N}\sum_{m=1}^{K}C(i,j)A_{jm}^{(T)}(1-A_{jk}^{(T)})$$

②if $f_1(k+1,A^{(T)})<f$ then {

(a) $f=f_1(k+1,A^{(T)})$；

(b) 记录 $i,j,p=j$；

}

End

(3) 将任务 t_i 分配于处理机 P 中;

(4) 记录处理机 P 的执行开销;

End

4. 得到所有子任务在处理机上的分配映射矩阵 A,以及各处理机的系统执行开销,其中

$$cost=\max\{f_i(k+1,A)\quad i=1,2,\cdots,m\}$$

即为应用程序在机群系统上并行执行所需要的时间.

End /* 算法结束 */

3.2 处理机分配算法示例

设一并行应用程序中,其并行子任务的计算量分别为:

$$t_i=\{10,\ 12,\ 8,\ 10,\ 9,\ 6,\ 3,\ 6\}$$

任务间的通讯量矩阵为:

$$C=\begin{bmatrix}0&7&5&4&4&2&4&2\\7&0&4&0&2&3&2&0\\5&4&0&4&2&2&0&5\\4&0&4&0&0&0&3&2\\4&2&2&0&0&3&2&3\\2&3&2&0&3&0&4&3\\4&2&0&3&2&4&0&0\\2&0&5&2&3&3&0&0\end{bmatrix}_{8\times 8}$$

为计算简便,设 $C_{ij}=C_{ji}$.

这时,TA_i 已经按 $cost=t_i+\sum_{j=1}^{n}C_{ij}$ 进行了排序.

$$TS_i=\{38,\ 30,\ 24,\ 22,\ 22,\ 21,\ 21,\ 21\}$$

在算法的执行中,假设各处理机之间的通讯处理速度一致,则可以不必考虑 S_{ij} 的不同,使之进一步简化.

如果要将上述并行应用程序划分于 4 个处理机的机群系统中,则其处理机分配过程为:

① 将 TS_1 分配于处理机 $P1$ 中,则:$f_1=t_1=10$

② 分配 TS_2,依算法可得,将 TS_2 分配于处理机 $P2$ 中:$f_1=f_1+c_{12}=17$ $f_2=t_2+c_{21}=19$

(若将 TS_2 也放于 $P1$ 中,$f_1=f_1+t_2=22$)

③ 分配 TS_3,同理,若将其置于 $P1$ 或 $P2$ 中,开销为:

$f_1=f_1+t_3=25$ $f_2=f_2+c_{23}=23$ $f_{\max}=25$

或 $f_1=f_1+c_{13}=22$ $f_2=f_2+t_3=27$ $f_{\max}=27$

若置 TS_3 于 $P3$ 中,则:

$f_1=f_1+t_{13}=22$ $f_2=f_2+t_{23}=23$ $f_3=t_3+c_{13}+c_{23}=17$

$f_{\max}=\max(f_1,f_2,f_3)=23$

∴ 将 TS_3 置于 $P3$ 中.

④ 同理分配 TS_1,若置于 $P1$,$P2$,或 $P3$ 中:

$f_1=f_1+t_1=32$ 或 $f_2=f_2+t_1=33$ 或 $f_3=f_3+t_1=27$

若不在 $P1,P2,P3$ 中:

$f_1=f_1+c_{14}=26$　$f_2=f_2+c_{24}=23$　$f_3=f_3+c_{34}=21$　$f_4=t_4+c_{14}+c_{24}+c_{34}=18$

∴ 选择将 TS_4 置于 $P4$ 中:　$f_{\max}=26$

⑤ 以下简略了分配过程,TS_5 分配于 $P4$ 中:

$f_1=f_1+c_{15}=30$　$f_2=f_2+c_{25}=25$　$f_3=f_3+c_{35}=23$

$f_4=f_4+t_5+c_{51}+c_{52}+c_{53}=35$

⑥ 将 TS_6 分配于 $P3$ 中:

$f_1=f_1+c_{16}=32$　$f_2=f_2+c_{26}=28$

$f_3=f_3+t_6+c_{61}+c_{62}+c_{64}+c_{65}=37$　$f_4=f_4+c_{46}+c_{56}=38$

⑦ 将 TS_7 分配于 $P2$ 中:

$f_1=f_1+c_{17}=36$　$f_2=f_2+t_7+c_{71}+c_{73}+c_{74}+c_{75}+c_{76}=44$

$f_3=f_3+c_{37}+c_{67}=41$　$f_4=f_4+c_{47}+c_{57}=43$

⑧ 将 TS_8 分配于 $P3$ 中:

$f_1=f_1+c_{18}=38$　$f_2=f_2+c_{28}+c_{78}=44$

$f_3=f_3+t_8+c_{81}+c_{82}+c_{84}+c_{85}+c_{87}=54$　$f_4=f_4+c_{48}+c_{58}=48$

这样,我们得到分配方案为:

$P1$:　$\{TS_1\}$

$P2$:　$\{TS_2,TS_7\}$

$P3$:　$\{TS_3,TS_6,TS_8\}$

$P4$:　$\{Ts_4,TS_5\}$

而系统开销 $f=\max(f_1,f_2,f_3,f_4)=54$

3.3　处理机分配算法分析

在上述的启发式处理机分配算法中,以

$$f(k+1,A^{(T)})=f(k+1,A^{(T-1)})+X(i,j)+\sum_{j=1}^{N}\sum_{m=1}^{K}C(i,j)A_{jm}^{(T)}(1-A_{jk}^{(T)})$$

作为主要的执行开销评价函数,其中计算了当第 t 步分配中,对于 t_i 子任务的分配,与前面已分配子任务间的计算开销及其通讯开销.由于在算法的最初,已将各子任务按其计算量与通讯量之和进行了排序,先分配开销大的子任务,而在每一步分配中,都将其分配于当前开销最小的处理机结点中,因而算法本身对于并行任务的分配,能获得满足的处理机映射关系,减少了问题本身由于 NP 完备性所带来的回溯性.

在算法中,以各个子任务的总的执行开销作为考察评价的参数,这一作法一方面有其算法执行的简单性.另一方面,在数值计算、信号处理等 DSP 应用中,都能获得满意的处理机分配结果.

从算法的执行中可以看出,对于各个子任务的处理机分配,都在各个处理机结点上对其所带来的执行开销进行评价函数计算,并置于一执行开销最小的结点中.这一机制,保证了整个系统中处理机任务分配在各个结点上的负载平衡问题,减少了由于结点负载不平衡对于应用程序执行时的总的性能影响,以获得较好的并行计算效率.

从算法执行的复杂性上分析,算法本身的执行开销在于处理每个子任务的分配,评价子

任务在各个处理机结点上的执行开销评价函数的计算以及对各子任务开销的排序过程.执行开销的排序过程复杂性为$O(n\log n)$,由于评价函数的计算只包含已分配的处理机与当前分配任务的通讯开销计算,因而算法的执行复杂性为:

$$O(n\log n)+O(m\times n\log n)=O(mn\log n)$$

在这一处理机分配算法中,子任务与其它结点间的通讯开销由2个子任务间的通讯量来进行评价计算,但在实际的应用程序中,处理机之间的通讯不单纯是一个数据通讯量的问题,它还受到各处理机的忙闲状态、程序的执行状态、通讯链路的占用情况以及中间处理机结点间的执行状态等多方面因素的影响.往往实际的通讯开销会比评价估算的处理机间通讯量计算要大,并且是一个动态执行的影响参数,因而,更进一步的子任务执行开销应以如下的评价函数进行计算:

$$f(k+1,A^{(T)})=f(k,A^{(T-1)})+B_0X(i)+\sum_{j=1}^{N}\sum_{m=1}^{K}B_jC(i,j)A_{jm}(1-A_{mk})$$

其中 $B_0<B_j,1\leqslant j\leqslant k$ $\qquad B_0+\sum_{j=1}^{K}B_j=1$

这一方面考虑了通讯开销计算时的动态影响因素,另一方面,还考虑了不同子任务分配时,所分配结点在网络拓扑结构中,所处位置不同,其通讯动态影响因素也不同的关系.这是一个很复杂的问题,需要更进一步地研究.同时,与机群系统中网络拓扑结构连接以及网络通讯的性能与效率有直接的关系.

4 结束语

该处理机分配算法已实现于PGR并行图归约智能工作站系统上的并行程序设计环境中,通过大量的实际应用例子的运行测试可看出,该算法将并行任务有效地划分到工作站结点中,提高了并行计算效率,有着重要的作用.[12]

处理机任务分配是机群系统完成并行任务计算、获得高并行执行性能的关键.在我们的分析与算法设计中,充分考虑了并行子任务对并行计算过程的计算开销以及处理机间通讯开销的影响.通过启发式的性能评价函数计算,使算法能获得处理机分配满意的任务映射,同时减小其执行的复杂性.由于并行应用程序的执行中,通讯开销不仅受任务间的通讯量的影响,同时还与系统结构本身有着直接的关系,因而还需要进一步采用静态分配与动态执行平衡的研究方法,解决同构机群系统中的处理机分配问题.

参考文献

1 Parasoft. Express: user's guide. Parasoft Co. Paradena, CA, 1988.

2 Geist Al, Beguelin Adam, Dongarra Jack *et al*. PVM: parallel virtual machine — a users guide and tutorial for network parallel computing. MIT Press, Cambridge, Massachusetts, 1994.

3 Chu W W *et al*. Task allocation in distributed data processing. IEEE Comput. Mag., Nov. 1980, **13**(11): 57~69.

4 Lo V M. Heuristic algorithms for task assignment in distributed systems. IEEE Trans. Computer, Nov. 1988. 1384~1397.

5 Coffman E G, Jr Garey M R, Johnson D S. An application of bin-packing to multi-processor scheduling. SIAM J. Comput., Feb. 1978, **7**: 1~17.

6 Efe E. Heuristic models of task assignment scheduling in distributed systems. IEEE Computer, June 1982. 50～56.
7 Coffman E G. Computer and job shop scheduling theory. New York: Wiley, 1976.
8 IEEE Computer Society. IEEE standard for scalable coherence interface (SCI). IEEE Std., 1596～1992.
9 Bit3 Co. Bits link workstations.
10 CCITT. Recommendation I. 150: B-ISDN ATM functional characteristics. Revised version, Geneva: ITU 1992.
11 温钰洪. 分布并行计算机系统中并行程序设计环境的研究与设计[博士论文]. 北京：清华大学，1994.
12 Wen Yuhong, Wang Dingxing, Shen Meiming *et al*. A parallel programming environment based on message passing. In: The Proceedings of ICPADS'94, Taibei, Dec. 1994. 724～729.
13 Grama A, Gupta A, Kumar V. Isoefficiency function: a scalability metric for parallel algorithms and architectures. IEEE Parallel Distributed Technology, 1993, **3**: 12～21.

A PROCESSOR ALLOCATION ALGORITHM IN PARALLEL COMPUTING OF HOMOGENEOUS WORKSTATION CLUSTER SYSTEMS

WEN Yuhong　WANG Dingxing　SHEN Meiming

(*Department of Computer Science　Tsinghua University　Beijing*　100084)

Abstract　Network computing of workstation cluster systems has brought about a lot of new research problems to the applications in the distributed parallel computing environments, and has become a hot-spot research problem in the parallel processing. How to allocate the workstation processors and map the parallel tasks onto the allocated processors will greatly influence the parallel computing performance of the applications. This paper analyzes the system performance parameters in homogeneous network computing and then introduces a fast efficient heuristic processor allocation algorithm.

Key words　Processor allocation, workstation cluster systems, system performance evaluation function.