一个基于收益与开销的作业选择策略*

胡 亮 徐高潮 鞠九滨

(吉林大学计算机科学系 长春 130023)

摘要 本文介绍了作者研制的负载平衡系统ILBOT(intelligent load balancer based on on-line tracing)的作业选择策略,该策略考虑了不同的负载环境对不同类型的作业响应时间的影响,并以此为依据来估算作业转移的收益与开销,将一个基于收益与开销的新的选择策略用在负载平衡算法中.性能测试的结果表明,使用该策略能较好地缩短作业的平均响应时间和提高资源的利用率.

关键词 负载平衡,工作站群,选择策略.

中图法分类号 TP393

网络中负载平衡就是要将网络中重负载主机上的作业转移到轻负载的主机上执行,使得网络中所有主机的负载趋向均等,目的是要缩短作业平均响应时间和提高整个系统的资源利用率.负载平衡算法分成静态(Static)和动态(Dynamic)两种.动态负载平衡算法使用系统状态信息(各节点上的负载)进行负载分配决策.静态算法不使用这种信息,而是使用预先知道的系统知识.动态负载平衡算法比静态的优越,因为它能利用系统状态的短期起伏改进性能.动态负载平衡算法的组成包括4个部分.[1,2]

(1) 转移策略决定节点是否处于适合参加任务转移的合适状态,即决定某节点是个任务发送者还是远程任务的接收者.

(2) 选择策略决定哪一个任务应该转移.选择一个任务进行转移的基本判据是,转移此任务的开销比起它的响应时间的减少是合算的.

(3) 定位策略决定把所选择的作业转移到哪个节点上.

(4) 信息策略负责收集系统状态信息.

选择策略指应选择哪个(些)进程进行转移.一个简单且易实现的方法是只考虑迁移新到达的进程.V系统[3]和NEST[4]使用的选择策略是只选择新到达的作业进行转移,而不管此作业的类型.这种方法是盲目的,例如,短作业转移后作业响应时间的改进抵消不了转移的开销.选择策略中所使用的主要判据是,把一个进程迁移到远程机可改进响应时间.Kruger和Finkel[5]概括了迁移进程时应考虑的7个因素,虽然这些因素很全面,包括了评价选择策略的两个重要方面,即"开销"和"收益",但没有给出如何度量这些因素,以保证被选择远程执行的作业能改进响应时间.Kuhl和Casavant[6]指出今后的负载平衡系统应从这两方面考虑,这要求对作业响应时间的改进和转移开销进行估算,从而必须获得作业的性质和不同系统环境对不同作业响应时间的影响.这是非常困难的课题,至今成果非常少,因为这要求在作业执行以前预测性质(如资源要求及执行时间).

Barak,Shiloh[7]和Kruger等人[5]在选择策略中考虑了作业响应时间的改进和开销,但无法定量估计,所以并未用于实际的系统中.Svenson[8]根据作业的过去执行时间进行作业选择,即对一个命令事先测量其平均执行时间,把所有测量过的作业列个表,运行此作业时先查表,若大于某个门限,则在本地执行,否则转移.这个被称为智能筛选的方法只考虑了作业执行时间因素,未考虑作业性质.此外,未执行过的作业无法处理.Koch[9]和Wang等人[10]使用人工神经网,通过学习作业过去的执行特征的知识来指导下次的作业选择.这个方案的优点是:决定作业是否转移时用了比较精确的知识,可适应各类作业和系统配置的变化.其缺点是:对某个作业的转移作出正确决定前必须已执行过很多次,次数愈多愈正确;每个命令带有某个参数是一类,不同参数的相同命令不属于一类,所以限制很严.这种方法不适

* 本文研究得到国家自然科学基金和吉林省青年基金资助.作者胡亮,1968年生,讲师,主要研究领域为分布式系统与计算机网络.徐高潮,1966年生,博士,讲后,主要研究领域为分布式系统与计算机网络.鞠九滨,1935年生,教授,博士导师,主要研究领域为分布式系统与计算机网络.

本文通讯联系人:胡亮,长春 130023,吉林大学计算机科学系

本文1996-10-15收到原稿,1997-05-12收到修改稿

© 中国科学院软件研究所 http://www.jos.org.cn

合用于实际系统的实现.

Sprite 系统[11]和 Condor[12]由用户选择作业进行转移,不支持自动选择. Stealth[13]和 Utopia[14]用查表方法支持作业的自动选择,表中列出以前执行过的作业名及转移建议.

1　ILBOT 的作业选择策略

ILBOT(intelligent load balancer based on on-line tracing)的作业选择策略可如下简单描述:(1) 使用在线跟踪估算出该作业在完全空闲的主机上的执行时间 T_0,并确定作业的性质;(2) 使用环境对不同作业的影响估算出该作业在当前执行环境下的主机上的执行时间 T_1;(3) 估算出作业转移的总开销 ΔT. 当 $T_1-T_0>\Delta T$ 时,说明此作业远程执行的收益大于开销,可以被转移;否则,作业留在本地执行. ILBOT 系统只处理单个进程的作业,未考虑由多个进程组成的作业的情况. 另外,ILBOT 系统的空闲机并不是绝对的空闲,而是指它的负载指标低于某一特定值(例如 CPU 的利用率低于 30%).

文献[15]详细论述了如何利用在线跟踪技术获得作业的性质. 在线跟踪就是事先运行作业一段时间(如 1s),对其进行跟踪,估算出它在完全空闲的主机上的执行时间和资源需求(它的 CPU 利用率和 IO 利用率). 实际上在线跟踪的过程又是短作业的自动筛选的过程. Leland 和 Ott[16]在 VAX750 和 780 上分析了日常运行的 950 万个 UNIX 进程,分析结果表明 98%的小进程占用 35%的 CPU 时间,而 0.1%的大进程却占用了 50%的 CPU 时间. Cabrera[17]对实际的在若干台 VAX11/750(780)上运行的 12.2 万个进程的寿命分布进行了实验测量和分析,发现进程的平均寿命为 0.4s,78%的进程寿命小于 1s,97%的进程可在 8s 内结束. 我们在 SUN4/65 上统计了 1 万多个进程的行为特征,统计时使用了系统本身提供的 acct()系统调用,编写了一个统计程序,在网络上对各类不同用户命令进行自动统计,统计的结果表明 80%以上的进程可以在 0.5s 以内完成,这些作业无疑应该留在本地执行. 如果我们选择跟踪时间为 1s 的话,那么 80%以上的作业可以在跟踪时间内完成,所以不需调度到远程执行.

2　环境负载对作业响应时间的影响

环境负载是指作业进入系统时的背景,即当前机器的忙闲程度,例如,CPU 环境负载是指当时 CPU 的忙闲程度. 我们对不同类型作业的响应时间同各种不同负载环境之间的关系进行了大量的实验研究,得出以下结论[15]:(a) 不同忙碌程度的 CPU 对 I/O 类作业的响应时间影响甚微,而对 CPU 类作业影响很大;(b) 环境负载的 IO 利用率对 CPU 类作业的响应时间影响很小,但对 IO 类作业影响很大;(c) 当 CPU 利用率达 100%时,不同 CPU 队列长度对 CPU 类作业有不同影响. 进一步的研究表明环境负载对作业响应时间的影响可用下面的表达式进行较精确的描述.

(1) CPU 利用率和 IO 利用率对作业响应时间的影响

下面讨论不同 CPU 利用率的环境对不同 CPU 负载的作业的影响. 在 UNIX 系统下,某一节点内进程的调度是时间片轮换的. 假设所有的进程具有相同的优先级. 假设环境负载为 X,刚进入的作业负载为 Y,设每个作业的优先级是完全相同的,通过大量的实验得出 CPU 的使用概率为 $X+Y-XY$,所以作业进入后 CPU 的利用率为

$$X+Y-XY \tag{1}$$

因为作业的优先级是完全相同的,所以环境负载的作业占用 CPU 利用率为

$$(X+Y-XY)X/(X+Y)=X-X^2Y/(X+Y) \tag{2}$$

同样,刚进入的作业占用 CPU 利用率为

$$(X+Y-XY)Y/(X+Y)=Y-Y^2X/(X+Y) \tag{3}$$

设刚进入的作业在完全空闲的主机上的执行时间为 T,则该作业要求 CPU 的服务时间为

$$YT \tag{4}$$

设刚进入的作业在环境负载为 X 的主机上的执行时间为 T_1,则该作业要求 CPU 的服务时间由式(3)可得,为

$$T_1Y-Y^2XT_1/(X+Y) \tag{5}$$

由于作业在不同的执行环境下要求 CPU 的服务时间应该是一样的,所以式(4)和式(5)是相等的,可得出

$$T=T(X-Y)/(X+Y-YX) \tag{6}$$

所以,该作业在环境负载为 X 下执行时间在完全空闲的主机上执行时,响应时间增加的百分比为

$$100\%*(T_1-T)/T=XY/(X+Y-XY) \tag{7}$$

图 1 中实线是实际测试出的不同%CPU 负载环境对不同%CPU 作业的响应时间的影响,测试的硬件环境为 Sun4/65 工作站,测试程序是我们编制的一个应用程序,它占用 CPU 负载 40%. 虚线是根据公式(7)计算出的在不同

© 中国科学院软件研究所 http://www.jos.org.cn

%CPU负载环境下分别运行不同%CPU 作业的响应时间的变化情况.我们从图中可以看出,实际测试的结果和理论计算出的结果符合得相当好.

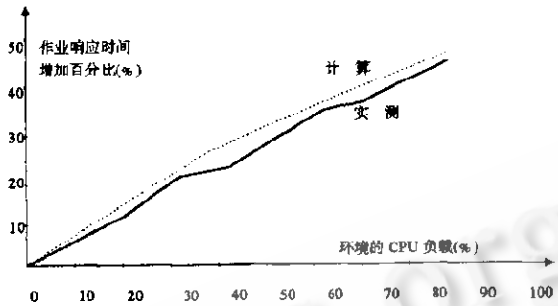

图1 CPU 负载为40%的作业在不同环境下响应时间增加百分比

IO 的情况应与 CPU 类似,在没有 IO 缓存的情况下有着与 CPU 一样的计算公式.

(2) CPU 队列对作业响应时间的影响

当 CPU 利用率达 100%时,CPU 队列长度是影响作业响应时间的决定因素.假设环境负载为 L,刚进入的作业负载为 X,每个作业的的优先级是完全相同的,那么刚进入的作业可以看作 X 个作业,这时,系统中争用 CPU 的作业数可看成是$(X+L)$个,设刚进入的作业在空闲机上的执行时间为 T,则不占用 CPU 的时间为$(1-X)T$,而占用 CPU 的时间 XT.由于这时系统中争用 CPU,而争用 CPU 的作业数是$(X+L)$,故系统分时的结果是,这部分处理时间延长到$(X+L)$倍,成为$(X+L)XT$,所以刚进入的作业在此环境下的执行时间为$(1-X)T+(X+L)XT$.所以同空载相比,作业执行时间增加的百分比为

$$X^2+XL \tag{8}$$

图2中的虚线是根据此公式计算出的在不同 CPU 队列负载环境下分别运行不同%CPU 作业的响应时间的变化情况.我们从此图中可以看出,实际测试的结果和理论计算出的结果符合得相当好.

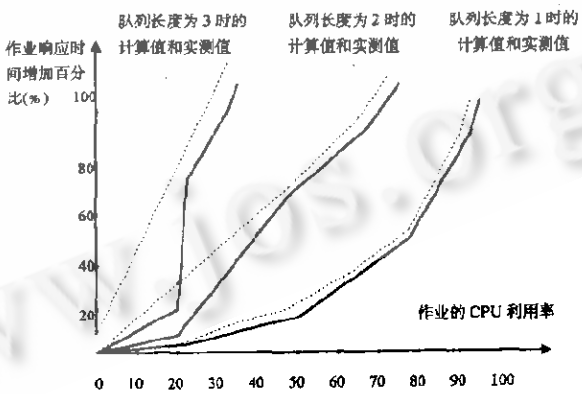

图2 CPU 队列长度对不同%CPU 的作业的响应时间的影响

3 作业转移所能获得的收益与开销

选择什么样的作业转移,主要依据此作业转移后是否可能缩短响应时间.只有在缩短的响应时间大于开销的情况下,作业转移才是有效的.

(1) 作业在某执行环境下的执行时间的估算

执行环境对作业响应时间的影响包括3个方面:环境的 CPU 利用率;环境的 CPU 队列长度;环境的 IO 利用率.

设作业在完全空闲的主机上的执行时间为 T_0,CPU 利用率为 X,IO 利用率为 Y;环境的 CPU 利用率为 R,CPU 队列长度为 L,IO 利用率为 S,则作业在当前执行环境下的本地机上的执行时间 T_1 通过以下步骤估算出来:

© 中国科学院软件研究所 http://www.jos.org.cn

(a) 环境IO对作业执行时间的影响,假设相对于完全空闲的主机,响应时间增加的百分比为I,则有

$$I=SY/(S+Y-SY);$$

(b) 环境CPU对作业执行时间影响,假设相对于完全空闲的主机,响应时间增加的百分比为C,当$R>95\%$时,使用CPU队列计算CPU对作业执行时间影响,则有

$$C=X^2-X+XL;$$

当$R<95\%$时,使用CPU利用率计算CPU对作业执行时间影响,则有

$$C=RX/(R+X-RX);$$

(c) 作业在当前执行环境下的本地机上的执行时间

$$T_1=T_0+T_0(C+I);$$

那么作业从当前执行环境转移到空闲机上执行所能获得的收益为$B=T_1-T_0$.

(2) ILBOT中作业转移的总开销的估算

对于所有的作业,转移的额外开销包括3个部分:(1) 作业的跟踪时间(我们选为1s);(2) 跟踪后恢复时间;(3) 远程执行的代价,即远程启动一个作业的时间(在我们的ILBOT系统中为1.5s).其中作业的跟踪时间和远程执行代价对所有的作业来说都是一样的,不同类型的作业有不同的恢复时间.由于恢复时间不会超过跟踪时间,所以转移的额外开销不会超过3.5s.

作业远程执行时,由于需要访问基地节点上的文件,必然会造成一定的性能损失.这部分损失是可以估算出来的,估算方法如下:

(1) 作业跟踪完成之后,跟踪服务程序形成了一个被访问文件的情况表,表中记录了哪些文件需要回到基地节点访问及在跟踪的这段时间内对这些文件的操作量.设跟踪的这段时间内远程访问文件的总字节数为D,那么作业访问远程文件的总的字节数可估算为T_0D.

(2) 设系统访问远程文件的速度为S,由于远程访问文件要比本地访问文件慢得多,所以远程访问文件的时间损失可估算为$F=T_0D/S$.

所以作业转移的总开销为$\Delta T=F+3.5$(s).当$B>\Delta T$时,说明此作业可以被转移;否则,作业留在本地执行.

ILBOT的选择策略同Svenson[8]的方法不同,Svenson的方法是确定一个门限T,根据作业以前的执行时间来决定作业是否转移,如果作业以前的执行时间大于T则转移,否则,留在本地执行.Deriche[38]早在1989年就指出这种固定门限的方法不能适应系统状态的变化,不同类型的作业在不同的执行环境下应选择不同的门限.实际上,我们基于"开销"和"收益"的选择策略是一种动态的门限策略.

4 性　能

我们将ILBOT的选择策略同固定门限的选择策略LBST(load balancing with scheduled threshold)进行了对比测试.测试是在由Ethernet连接起来的6台Sun4/20无盘工作站和1台Sun4/65服务器上进行的.我们分别在无负载平衡系统NoLB(no load balancing),LBST和ILBOT上测量了每种情况下的平均作业响应时间、平均CPU利用率和平均CPU队列长度.为了构成不同的负载,在不同的主机上运行不同的命令流(Script),其组成包括各类作业:大计算量的(如求大质数、积分)、编译类的(如cc,f77)、I/O类的(如cp)、交互式的(如ls,ps,df),以便模拟实际情况.6台工作站的负载分布属于典型情况,即轻载(%CPU<30%)、中等(%CPU在40%和60%之间)、重载(%CPU>70%)各两台.LBST的门限选为10s.

测试结果见表1.由表1可看出,采用基于收益与开销的作业选择策略的ILBOT系统在平均作业响应时间及均方差、平均CPU利用率均方差等主要指标上均明显优于LBST.

表1　各种LB算法比较

算法	平均作业响应时间				平均CPU利用率			平均CPU队列长度		
	测量值(s)	改进(%)	均方差(s)	改进(%)	测量值(%)	均方差(%)	改进(%)	测量值(%)	均方差(%)	改进(%)
NoLB	28.56	0	82.38	0	59.64	23.35	0	1.91	1.28	0
LBST	17.51	38.7	50.26	40.2	65.32	9.48	59.4	1.73	0.43	66.7
ILBOT	12.96	54.6	34.35	58.3	72.83	6.00	74.3	1.32	0.38	70.4

© 中国科学院软件研究所 http://www.jos.org.cn

ILBOT 充分估计了转移的开销，以不同的作业在不同执行环境下的执行时间来估计作业所可能带来的收益，进而判断作业是否应该转移，所以，ILBOT 的选择策略适用于不同的作业和不同的负载环境. 我们建议将基于"收益"与"开销"的作业选择策略应用于动态负载平衡系统中.

参考文献

1 Benmohammed-Mahieddine K, Dew P M. A periodic symmetrically-initiated load balancing algorithm for distributed systems. ACM Operating System Review, January 1994,28(1):66～79

2 Xu J, Hwang K. Heuristic methods for dynamic load balancing in a message passing multicomputer. Journal of Parallel and Distributed Computing, May 1993,18(2):1～13

3 Stumm M. The design and implementation of a decentralized scheduling facility for a workstation cluster. In: Proceedings of the 2nd Conference on Computer Workstations. March 1988. 12～22

4 Ezzat A K. Load balancing in NEST: a network of workstations. In: Proceedings of the Fall Joint Computer Conference. USA, November 1986

5 Kruger P, Finkel R. An adaptive load balancing algorithm for a multicomputer. Computer Science Technical Report #539, University of Wisconsin-Madison, USA, April 1984

6 Casavant T L, Kuhl J G. Effects of response and stability on scheduling in distributed computing systems. IEEE Transactions on Software Engineering, November 1988,14(11):1578～1588

7 Barak A, Shiloh A. A distributed load-balancing policy for a multicomputer. Software Practice and Experience, 1985,15(8): 901～913

8 Svenson A. History, an intelligent load sharing filter. In: Proceedings of the 10th International Conference on Distributed Computing Systems. Paris, France, IEEE Computer Society Press, 1990

9 Koch T, Rohde G, Kramer B. Adaptive load balancing in a distributed environment. In: Proceedings of the 1st Workshop on Service of Distributed and Networked Environments. June 27～28, 1994. 115～121

10 Wang C J, Kruger P, Liu M T. Intelligent job selection for distributed scheduling. In: Proceedings of the 13th International Conference on Distributed Computing Systems. Pittsburgh, Pennsylvania: IEEE Computer Society Press, 1993

11 Douglis F, Ousthout J. Transparent process migration: design alternatives and the sprite implementation. Software Practice and Experience, August 1991,21(8):757～785

12 Litzkow M T. Condor——a hunter of idle workstation. In: Proceedings of the 8th International Conference on Distributed Computing Systems. IEEE Computer Society Press, 1988. 104～111

13 Krueger P, Chawla R. The stealth distributed scheduler. In: Proceedings of the 11th International Conference on Distributed Computing Systems. Arlington, Texas, USA: IEEE Computer Society Press, 1991. 336～343

14 Zhou S, Wang J, Zheng X *et al*. UTOPIA: a load sharing facility for large, heterogenous distributed computer system. Software Practice and Experience, December 1993,23(12):1305～1336

15 鞠九滨，杨鲲，徐高潮. 使用资源利用率作为负载平衡系统的负载指标. 软件学报，1996,7(4):238～243
(Ju Jiu-bin, Yang Kun, Xu Gao-chao. Using resource utilization as load index in dynamic load balancing. Journal of Software, 1996,7(4):238～243)

16 Leland W E, Ott T J. Load balancing heuristics and process behavior. In: Proceedings of the ACM SIGMETRICS Conference on Measurement and Modelling of Computer Systems. May 1986. 54～68

17 Cabrera L F. The influence of workload on load balancing strategies. In: Proceedings of the Summer USENIX Conference'86. June 1986. 446～458

18 Deriche M, Huang N K, Tsai W T. Dynamic load balancing in distributed heterogeneous systems under stationary and bursty traffics. In: Proceedings of the 32nd Midwest Symposium on Circuits and Systems. August 14～16, 1989. 669～672

A Job Selection Strategy Based on Benefit and Overhead

HU Liang XU Gao-chao JU Jiu-bin

(*Department of Computer Science Jilin University Changchun* 130023)

Abstract In this paper, the authors introduce the job selection strategy in the load balancing system——ILBOT (intelligent load balancer based on on-line tracing). In this strategy, the authors consider the influence upon response time of various kind of job caused by the various load environment. According to it, they estimate the benefit and overhead of transferring a job. They use a new selection strategy based on benefit and overhead in load balancing algorithm. The measurement results of the experiment show that it can improve mean response time of jobs and resource utilization of systems substantially.

Key words Load balancing, workstation cluster, selection strategy.

© 中国科学院软件研究所 http://www.jos.org.cn