第16卷 第5期
2006年5月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol. 16 No. 5
May 2006

分布式计算机系统负载平衡研究

陈 涛[1],陈启买[2]

(1.安徽财经大学 信息工程学院,安徽 蚌埠 233041;
2.华南师范大学 计算机学院,广东 广州 510631)

摘 要:负载平衡是分布式系统中的一个研究热点。为了实现和充分利用这种能力,需要优良的负载平衡分配方案。对负载问题进行了数学化描述,研究了分布计算机系统中基本负载平衡策略,在此基础上提出了一个基于人工智能的负载平衡方案。利用在线跟踪技术,获得作业的行为特征(资源需求和执行时间等),从而筛选出那些不值得转移的短作业。性能测试的结果表明,文中所提出的方法能够较好地缩短作业的平均响应时间和提高系统的资源利用率,实现了动态负载平衡的目的。

关键词:负载平衡;分布计算机;智能调度

中图分类号:TP333.8 **文献标识码**:A **文章编号**:1673-629X(2006)05-0033-03

Study on Load Balancing of Distributed Computer System

CHEN Tao[1], CHEN Qi-mai[2]

(1. School of Info. Eng., Anhui Univ. of Finance & Economics, Bengbu 233041, China;
2. Institute of Computer, South China Normal University, Guangzhou 510631, China)

Abstract: Load balancing is a hot reaearch point in distributed systems, but the good load balancing strategies is needed. Based on mathematics model description, the basic load balancing strategies and their characters are studied. In this groundworks, an intelligence load balancing strategy and its algorithm are addressed. Using on-line tracing technique, this paper can predicate the behavior of a job, such as its resource requirements and its approximate execution time. Therefore, we can recognize those short-lived jobs which are not worth transferring. Experiment measurement results show that it is able to reduce mean response time and improve resource utilization of systems.

Key words: load balancing; distributed computer system; intelligence dispatch

0 引 言

在分布式系统中,经常出现某些处理机负载过重而另外一些处理机负载很轻甚至空闲的情况。为了提高处理机的利用率和系统并行计算的效率,人们试图把负载过重处理机上的一部分负载转移到空闲或轻负载处理机上,从而导致了负载平衡问题的研究。负载平衡是分布式系统的资源管理模块,它的主要功能是合理和透明地在处理器之间分配系统负载,以达到系统的综合性能最优。

总的来说,负载平衡策略可以分为静态负载平衡和动态负载平衡。静态调度算法是根据系统的先验知识做出决策,运行时负载不能重新分配[1]。静态调度算法的目标是调度一个任务集合,使它们在各个目标结点上有最短的执行时间。设计调度策略时要考虑的3个主要因素是处理机的互连、任务的划分和任务的分配[2]。而动态负载平衡是根据计算机过程中数据项的变化情况,交换系统的状态信息来决定系统负载的分配。它具有超过静态算法的执行潜力,能够适应系统负载变化情况,比静态算法更灵活、有效。动态算法利用系统状态的短期波动来提高性能。由于它必须收集、储存并分析状态信息,因此动态算法会产生比静态算法更多的系统开销,但这种开销常常可以被抵消掉。这里集中讨论动态负载平衡算法。

收稿日期:2005-09-29
基金项目:安徽财经大学自然科学科研基金(04AC059);安徽省教育厅自然科学科研基金(2005KJ051)
作者简介:陈 涛(1972-),男,安徽太和人,硕士,讲师,研究方向为并行计算和数据挖掘;陈启买,硕士生导师,研究方向为数据挖掘和分布式系统。

1 负载平衡问题的描述

设系统由 n 台处理机组成,顺序标记为 $P_0,P_1,P_2,\cdots,P_{n-1}$,处理机之间通过通信线路加以连接,用每台处理机拥有的任务数表示其负载,记为 $W(i)(0\leqslant i\leqslant n-1)$。整个系统的总负载 $W=\sum W(i)$,系统的平均负载为 $W^*=w/n$。定义负载上界 $W_1{}^*=w^*+\mathcal{E}$,定义负载下界 $W_2{}^*=w^*-\mathcal{E}$($\mathcal{E}$ 为一参数,其值大小根据具体的多处理机系统而定)。

某一处理机结点的负载 $W(i)<W_2{}^*$ 时,该处理机结点定义为轻载结点;当负载 $W_2{}^*<W(i)<W_1{}^*$ 时,

该处理机结点定义为适载结点;当负载 $W(i) > W_1^*$ 时,该处理机结点定义为重载结点;当负载 $W(i) = 0$ 时,该处理机结点定义为空载结点。

2 动态负载平衡策略

典型的动态调度算法有 5 个策略:启动策略、转移策略、选择策略、定位策略和信息交换策略。

2.1 启动策略

启动策略的责任是决定谁应该激活负载平衡活动,有 3 种方法:发送者发动、接收者发动和对称发动[3]。

在发送者发起的方法中,负载分配活动由重负载结点发动,它力图把一个进程发送到轻负载结点,其性能在系统的整体负载较轻的情况下比较有效。迄今为止,这种方法研究的也比较多。在接收者发起的方法中,轻负载结点向重负载结点请求获得一个进程,其性能在系统的整体负载较重的情况下比较有效。对称发动的方法使用兼有接收者发动的和发送者发动的方案,可以根据当前负载情况自动进行切换。发送者发动适用于系统低负载情况,而接收者发动适用于系统高负载情况。使用何种策略应根据平均负载值进行切换。

2.2 转移策略

转移策略决定一个结点是否在合适的状态参与负载转移。多数转移策略是使用静态门限策略(Threshold Policy)。门限用负载单元数表示,当一个结点的工作负载超过某个门限值时,该结点的工作负载可以转移到网络中的其它结点上。如果结点的负载小于某个门限值时,转移策略就认为它是一个远程任务的接受者。Lin 和 Keller[4]使用两个门限将结点的工作负载分类成轻的、中等的和重的,认为仅当超过重负载门限时,才需要转移负载。

某些转移策略决定进程迁移的主要标准是使用两机负载差值。Stankovic, Krueger, Finkel 的超平均算法使用基于负载差的转移策略,即若两机的负载差值超过某值则进行转移。另一个使用负载差方法的是,根据对远程机负载的当前估计值决定其是否迁移,周期地检查本地进程在远程其它机上可能的响应时间,若有很大的改进(考虑了迁移开销),则希望此进程迁移。

尽管有上面的选择策略,但都没有给出门限和负载差值大小的选择方法,所以都采用了固定值,一旦选定就长期不变。实际上不同的作业在不同的系统下应动态调整。

2.3 选择策略

源处理器选择最适合转移、最能起平衡作用的任务,并发送给合适的目标处理器。最简单的方法是选择最新生成的任务,这个任务导致处理器工作负载超出门限值。这些任务相对来说转移的代价不大,特别是对于非抢先式的负载转移来说,更是如此。另一种方法是选择一个已经运行的任务,然而,这时可能的结果是转移运行任务的代价抵消了作业运行时间的减少。因此,转移的作业运行的时间应该足够长,否则,相应时间的改进被转移的开销所抵消。Svensson[5]根据作业的过去的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,运行一个作业前先查表,若其平均执行时间小于某个规定值,则只能在本地执行。Wang[6]使用人工神经网络,通过学习作业过去的执行特征知识来指导下次的作业选择。Stealth 和 Utopia 用查表方法支持作业的自动选择,表中列出了以前执行过的作业名及转移建议。

作业选择,特别是用于非抢先的作业选择是非常困难的问题,至今成果非常少,因为这要求在作业执行以前预测其性质(如资源要求及执行时间)[7]。

2.4 定位策略

按照作业定位的范围可分为局部定位和全局定位。局部定位是在局部范围内为作业寻找合适的执行结点,而全局定位是在全局范围内为作业寻找合适的执行结点。

(1)局部定位策略有以下两种:a. 成对方法:每个处理机力求与负载差极大的一个邻居处理机成对,负载的转移是在成对的处理机之间发生。b. 负载向量方法:在每个机器上维持一个负载向量,它给出最近收到的网络中有限数目的机器的负载值。负载平衡决策是根据一个机器的负载与此机器上保持的负载向量指出的其它机器的负载相对差作出的。Barak 等人和 Ni 等也使用这种方法,不同的是他们以估计的作业响应时间表示负载轻重[8]。

(2)全局定位策略也有两种:a. 广播方法:是仅当机器成为空闲时广播一个报文,通知它要接受迁移进程。使用广播请求会接收到大量的回答报文,如果不追求最佳的,可以只接收第一个回答或有限几个回答报文的轻载机。b. 全局系统负载方法:这时每个机器应力求计算全系统的负载,并相对于此调整其自身的负载,而不是交换本地负载值。这个方法能检测系统是否处于全面的重负载和轻负载,当一个机器的本地负载与此平均值相差很大并且不能找到其负载处于互补状态的另一个机器时,就修改它所保持的全局平均值并向所有其它机器广播这一事实。例如,如果某机超载并且找不到一个轻载的机器时,那么就应增加全局平均值。要适当设置,允许一个进程从某机器上迁移所使用的该机负载与全局平均值之差额量。太小会使机器花费很多时间进行进程迁移,太大会漏掉很多迁移的机会。

2.5 信息交换策略

在自适应型负载平衡策略中为了对到达某结点的请求服务的进程决定如何布局,必须有个机构在网中传播有关处理机负载的信息。因为网络结构是耦合的,所以此信息与真实的系统状态在精确程度上有所偏离(过时)。但在具有足够的精度的同时必须避免不稳定性。信息交换策略有以下 3 种[9]:

(1)按需驱动。某结点仅当成为发送者或接收者时,才收集其它结点状态信息,如文献[10]的信息策略是仅当某个处理机根据本地负载状态确信超载时才请求网络中其

它处理机的负载信息。

⑵周期进行。各结点定期交换负载信息。这个周期值必须仔细地选择。周期太短会产生不稳定性,并且频繁的负载交换将产生附加的开销;周期太长则不精确。

⑶状态改变时驱动。结点当其状态改变到某种程度时发布其状态信息。注意它是发布本结点的信息,而不象按需驱动那样收集其它结点状态信息。可以将机器状态分为轻的、中等的和重的,只有在负载由重负载变成轻负载状态时才进行信息交换。

3　基于智能的负载平衡系统

根据上面所提到的策略,课题组研究了一个基于智能的负载平衡系统。该系统利用在线跟踪技术,获得作业的行为特征(资源需求和执行时间等),从而筛选出那些不值得转移的短作业。

3.1　组　成

该系统由主机状态服务员、远程执行服务员和调度服务员3个部分组成,如图1所示。用户将作业提交给调度服务员,调度服务员对用户所提交的作业进行在线跟踪,获得此作业的性质,判断其是否可调度到远程去执行。主机状态服务员向其它的主机状态服务员发广播查询报文,为可以远程执行的作业寻找一个最佳机。最佳机上的远程执行服务员派生一进程执行分派到此最佳机上的远程作业。

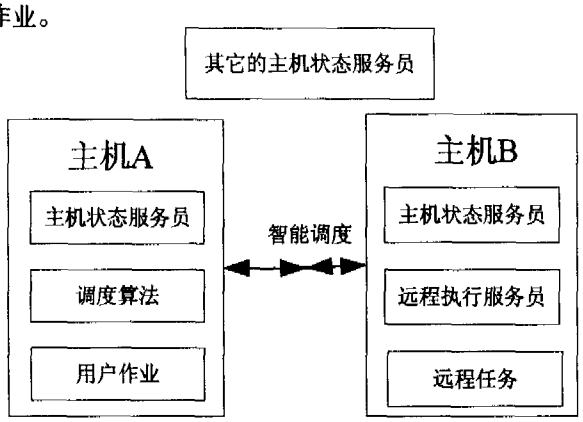

图1　基于智能的负载调度的组成

3.2　调度算法

该系统使用动态的源发动,即由作业进入结点启动,并根据作业当前状态决定作业迁移策略,只允许转移一次。

当用户作业提交给调度服务员时启动如下算法:

(1) 获得本地结点的CPU利用率X、本地CPU队列长度Y和本地I/O利用率Z。

(2) 确定调度的阈值,如果属于轻负载工作站,则在本地执行,转(5)。

(3) 启动作业的执行,进行在线跟踪1s,获得CPU的利用率和I/O利用率。如果跟踪期内作业执行完毕,则转(5)。

(4) 在网络中寻找一个负载最轻的主机,作业迁移到最佳机上运行。

(5) 等待作业执行完毕,利用作业实际执行时间来校正和估计算法的相关参数并修正。此类作业下次执行时可直接利用该知识。

4　结束语

负载平衡调度是分布计算机系统有效利用处理器资源的一种途径,它能让多台服务器或多条链路共同承担一些繁重的计算或I/O任务,从而以较低成本消除系统瓶颈,提高系统络的灵活性和可靠性。采用负载平衡技术可使分布式系统的整体吞吐量有所提高,特别是在分布式系统提供的服务程序所访问的资源多样化的情况下,效果尤其明显。把人工智能理论引进负载调度,为这类研究开辟了一个新的方向。

参考文献:

[1] Kunz T. The influence of different workload description on a heuristic load balancing scheme[J]. IEEE Trans. on Software Engineering, 1991, 17(7): 725－730.

[2] Wu J, Fernandez E. A scheduling scheme for communicating tasks[A]. In: Proc. of the 22nd Southeastern Symposium on System Theory[C]. Cookeville, TN, USA: IEEE Computer Society Press, 1990. 91－97.

[3] Eager D, Lazowska E, Zahorjan J. A comparison of reciever－initiated and sender－initiated adaptive load sharing[J]. Performance Evaluation, 1985, 10: 1－3.

[4] Lin F C, Keller R M. The Gradient Model Load Balancing Method[J]. IEEE Trans. on Software Engineering, 1987, 13(1): 32－38.

[5] Svensson A. History, an intelligent load sharing filter[A]. In: Proceedings of 10th International Conference on Distributed Computing Systems[C]. Paris, France: IEEE Computer Society Press, 1990.

[6] Wang C J, Krueger P, Liu M T. Intelligent Job Selection For Distributer Scheduling[A]. In: Proceedings of 13th International Conference on Distributed Computing Systems[C]. Pittsburgh, Pennsylvania, USA: IEEE Computer Society Press, 1993.

[7] Wu Jie. 分布式系统设计[M]. 高传善等译. 北京: 机械工业出版社, 2001.

[8] Eager D, Lazowska E, Zahorjan J. Adaptive load sharing in homogeneous distributed systems[J]. IEEE Trans. on Software Engineering, 1986, 12(5): 662－775.

[9] 鞠九滨. 机群计算[M]. 长春: 吉林大学出版社, 1999.

[10] Hac A, Jin X. Decentralized algorithm for dynamic load balancing with file transfer[J]. Journal of Systems and Software, 1991, 16(1): 37－52.