

机群并行计算中负载共享的关键问题

The Key Issue of Load Sharing in Cluster of Parallel Computing

胡凯 王强 胡建平

(北京航空航天大学计算机科学与工程系 北京 100083)

Abstract In this paper, we analyze some basic characteristics of Load Sharing under network parallel computing environment and provide a basic architecture and arithmetic. We stress that it is key issue how to select migratable process in Load Sharing and discuss a discriminant in detail. We also discuss some problems in implementation.

Keywords Load sharing, Network parallel computing, Cluster, Migrate

1 引言

利用一组工作站并行工作解决粗粒度的较大型计算问题,即 NOW 系统一直是计算机界一个重要的研究领域。随着 Internet 的迅猛发展,网上已有数十万的各种计算机,但实际的网络资源利用率是很低的,据有关统计,系统平均使用率仅为 30% 左右,有的空闲率竟达 91%,其功能扩展和利用率已成为人们日益关注的课题。显然,信息并不是网上唯一能获得的资源,网上计算机如果能被组织加以协调工作,将会形成具有巨大潜力的分布式并行计算环境,直至形成一能力无比的全球计算环境。

实际上也可把 Internet 看成是一个大的 NOW 机群,国际上 90 年代初就有人明确提出“基于网络的多计算机是构造并行系统的一个新方向”。(Scientific American)1997 年五月称:“未来世界最快的计算机将是 INTERNET 或它的子集”。随着网上并行最大的障碍——网络带宽和延迟的不断突破,以及支持异构的分布式对象技术标准和 Java 等跨平台语言的应用,利用网上闲散资源形成相当强大的并行计算能力成为一个重要研究方向。在我国高性能计算机较少、微机为主的情况下,网络并行计算环境技术的研究具有很重要的战略和现实意义。

网络并行计算群(Cluster)应尽可能遵循以下原则:

(1)在现有网络环境下,不附加和改变现有软硬件配置。以保证通用性。

(2)不影响网上用户的工作。

(3)能组织利用匿名网上用户加入计算群。

网络并行计算的一个基本问题是如何将工作负载有效地分布到各个处理机并根据情况动态地实现负载的共享,这方面的技术和策略是实现网络并行计算的核心。

2 负载共享技术

2.1 负载平衡和负载共享

负载平衡和负载共享都是研究如何在—组处理机间优化分配系统工作负载问题,面临许多同样的基本问题,但对网络并行计算环境而言,二者之间存在目标上的差别。

负载平衡,目的是为了充分发挥工作组中所有成员的能力,使每个成员尽量保持负载均衡,不出现忙闲不均的情况,从而改进系统整体效率。

为区别目前对负载平衡和负载共享不同的认识,方便研究,我们给出如下定义。

负载共享:在网络并行计算环境下,将已划分为一组并行子任务的任务负载按某种策略再分布到各处理机上并行计算,以尽可能加速任务的执行。

根据这个定义可以看出,负载平衡是追求系统的利用率和效率,而负载共享追求的是并行任务的加速比,单纯追求系统负载的绝对平衡会造成系统各类开销的增大,从而降低加速比。因此,二者在实现技术上是相通的,但在策略和算法上体现追求目标的不同。

2.2 负载共享的实现方式

胡凯 博士生,主要研究领域为分布式并行计算。王强 博士生,主要研究领域为分布式和图像处理。胡建平 教授,博士生导师,主要研究领域为分布式系统和知识工程。

负载共享按实现方式可分为

(1)静态方式:忽略系统当时状态,基于对某些计算量和通讯量的先验知识形成可迁移进程表。仅在任务未投入运行前可迁移。

动态方式:可根据当前运行状态自适应决定负载共享策略,又可分为:

(2)动态非剥夺式:仅允许在任务未运行前进行迁移。

(3)动态剥夺式:可挂起正运行进程进行迁移。

三种方式中,静态方式不考虑运行状态的变化,实现简单,但先验知识难以确定或可能作出错误决定,动态方式能自适应系统状态变化,但实现较复杂。静态和动态非剥夺式仅在任务未执行时使用,进程一旦在某处理机上运行须执行到结束,其间不能进行再分配,运行实际上是进程的远程执行(如用 rsh 等工具),开销小。动态剥夺式有利于并行任务的灵活配置,但须进程迁移的支持,开销大。

学术界对这几种方式一直争议较大,Eager 等人认为剥夺式开销太大,不如非剥夺式带来的利益大,许多系统也都遵循这一结论(如 Utopia),但 Barak 等人认为经过负载指标的选择限制及算法的制定,剥夺式能取得很好的实效且对系统重构等有不可替代的作用。

2.3 技术现状

在上述方式的实现上,传统的负载平衡通常有在用户层和在系统层实现二种形式。

(1)大部分系统仅在用户层实现动态方式,系统没有自己的策略,主要由用户决定,如在系统:Accent, Locus, Utopia, DEMOS/MP, V, NEST, RHOLIOS, MIST 等中。

(2)部分系统在系统层实现,可由系统提供策略,但大多是非剥夺式的,如在系统:Amoeba, Sprite, Conder, Mach 等中。

(3)在系统层实现动态剥夺式的系统较少,典型的是 MOSIX 系统。

2.4 负载共享的基本问题

负载共享主要研究以下问题:

(1)选择策略(who):选择哪一个进程进行迁移。

(2)定位策略(which):定位什么样的空闲处理机作为目的机。

(3)迁移策略(when):决定负载迁移的时机。

(4)负载共享的激发方式:有发送者激发(招标法)和接收者激发(征兵法)二种方式。

(5)控制方式:有集中式和分布式实现负载共享算

法二种方式。

3 关键技术

3.1 网络并行计算环境特点

在研究负载共享时,要考虑网络并行 Cluster 的如下特点:

(1)网络用户的自主性:网络用户由于地理位置的透明和匿名性,具有对本地资源的绝对控制权。因此,加入 Cluster 的用户须遵守一定的规则,一般遵循本地用户优先原则。

(2)网上资源是丰富的:可认为能利用的空闲处理机足够多,且可能是动态变化的。

(3)网络的不确定性:当前网络的流量和可靠性都是不确定的,负载共享算法必须包含一定的安全和容错处理。

3.2 基本实现结构

由以上讨论可导出网络并行计算环境下负载共享算法应采用以下策略:

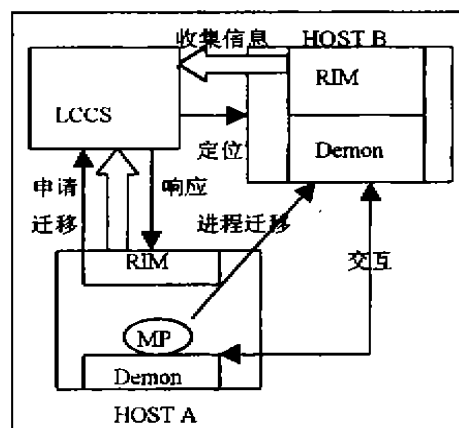
(1)由于可利用空闲处理机足够多,因此,空闲处理机的定位策略相对简单,激发方式宜采用发送者激发(招标法)。

(2)集中和分布式控制方式各有优缺点,本方案主张采用集中式,可减少网络通讯开销,方便 Cluster 的扩展。

(3)由于网络不确定性,负载共享应能提供动态剥夺方式,可支持负载的多次迁移。

(4)一般假定没有先验知识可利用,选择哪一个进程迁移,何时迁移,即选择策略和迁移策略,是实现的关键,直接关系到负载共享的收益。

一个基本实现结构如图 1 所示。



LCCS:负载共享集中控制服务器

Demon:守护进程 RIM:资源信息监视器

图1 处理机A向处理机B迁移进程

工作原理,各处理机周期性向 LCCS 传送本地资源信息,HOST A 选择可迁移进程集 MP,当满足迁移条件时,向 LCCS 提出申请,LCCS 定位空闲目的机 HOST B,然后应答 HOST A,HOST A 迁移进程到 HOST B。

3.3 负载共享的关键策略

我们强调进程迁移的时机和选择可迁移进程的标准和策略是实现动态剥夺式负载共享的关键问题。否则选择了不该迁移的进程可能会造成错误或开销大于收益情况。

• 采用以下迁移策略:

(1)规定的负载测量指标达到某个阈值时,负载指标和阈值是可动态调整的。

(2)发现有高优先权的进程需在本地运行时。

(3)发现有更合适新处理机加入 Cluster 时。

(4)发现某异常情况须重新配置进程分布时。

• 选择可迁移进程策略:所谓可迁移进程是其迁移后能使系统的平均响应时间减少的进程。但当进程未执行完前是很难作出定量评估的,因此,确定一个简单实用的选择可迁移进程的近似标准是十分重要的。

研究表明,根据运行信息预测性评估进程的生命周期是一种较为可行的方法。一般是迁移剩余运行时间较长的进程,才能体现迁移的好处。

对不考虑 I/O 的进程,统计表明通常只有约 4% 的进程运行时间超过 2 秒,但它们占据了约 60% 的 CPU 时间,是可迁移的主要对象。

设 age_i 表示进程已使用的 CPU 时间;设 life_i 表示进程的生命周期, MOR 等人研究指出 age_i > 1 秒的进程,其生命周期超过 T 秒的概率为:

$$P(\text{life}_i > T) \approx T^k \text{ 其中 } -1.3 < k < -0.8$$

进程的剩余生命周期和进程的 age_i 成线性正比关系,因此,选择迁移老进程是基本策略。

进程的迁移性还与迁移开销有关,一般要求:

$$\text{age}_i > \text{迁移开销} / (n - m)$$

其中 n, m 分别为源、目的处理机上进程数。

综上,我们可取经验公式,若进程 i 满足:

$$\text{age}_i > A * \text{迁移开销}, 0.1 < A < 1$$

则进程 i 是可迁移的,参数 A 的选择与处理机的环境有关,是可调的。

负载共享的算法如下:

```
for(i);
{收集运行信息报 LCCS
while(i ∈ 负载进程表)do
  if (age(i) > 0.5 * 迁移开销) then
    加入 i 到可迁移进程表 MPTAB 中
if 当前状况 ∈ {迁移策略} then
  向 LCCS 申请迁移
```

• 10 •

```
wait (LCCS 回答);
获取合适目的机地址
查 MPTAB 选择可迁移进程
执行进程迁移;
```

4 存在的问题

(1)异构问题:因为用户作业可能要在不同类型机器和操作系统中运行,必须使用一种与平台无关的语言,目前,这方面的解决技术较成熟,如 Java 平台、中间件语言和可移植的语言等,但也存在相当多的未解决问题。

(2)安全问题:是网络环境下 Cluster 最为敏感的问题,包括:

• 认证机制:要有良好的认证机制,保证迁移的目的处理机是遵循规则的可靠用户。

• 保护机制:保证本地非共享部分(如文件系统)的安全。保证各处理机获得的任务不是虚假任务(如病毒或死锁任务),新技术有软件容错隔离(SFI)技术,在机器码中插入保护检查等。

• 保密机制,不能使执行任务代码被随意拷贝,这方面目前还是难点。

(3)正确性和效率问题:须确认分配出去的任务确实被执行过,可采用在程序中加入校验和、检验计算结果等方法。

(4)组织问题:须决定本地用户优先还是并行任务优先原则,确定在多大程度上限制网络用户自主性。匿名用户自愿、合作地登录加入 Cluster 是一个重要前提。

(5)代码传递问题:匿名用户之间,程序执行代码必须预先或实时传递。存在着代码兼容性和重定位等诸多问题。

结束语 本文分析了网络并行计算环境下负载共享的基本特点,给出了基本实现结构和算法,强调指出选择可迁移进程集是负载共享技术的关键点,较详细讨论了其判别标准公式,最后讨论了实现中的难点。

参考文献

1. Osser W. Automatic Process Selection for Load Balancing. www.nestrl.com 1992
2. Zhou SongNian, et al. Utopia: A Load Sharing Facility for Large, Heterogeneous Distributed Computer System Software-Practice and Experience, 1993, 23(11)
3. Harchol M, et al. Exploring Process Lifetime Distributions for Dynamic Load Balancing. ACM Trans. on Computer System, 1997, 8
4. Zhang Y, et al. Comparison of dynamic and static load-balancing strategies in heterogeneous distributed systems. IEEE Proc. Comput. Digit. Tech., 1997, 144(2)
5. Svensson A. History, an Intelligent Load Sharing Filter. 16th Interl. Conf. on Distributed Computing System