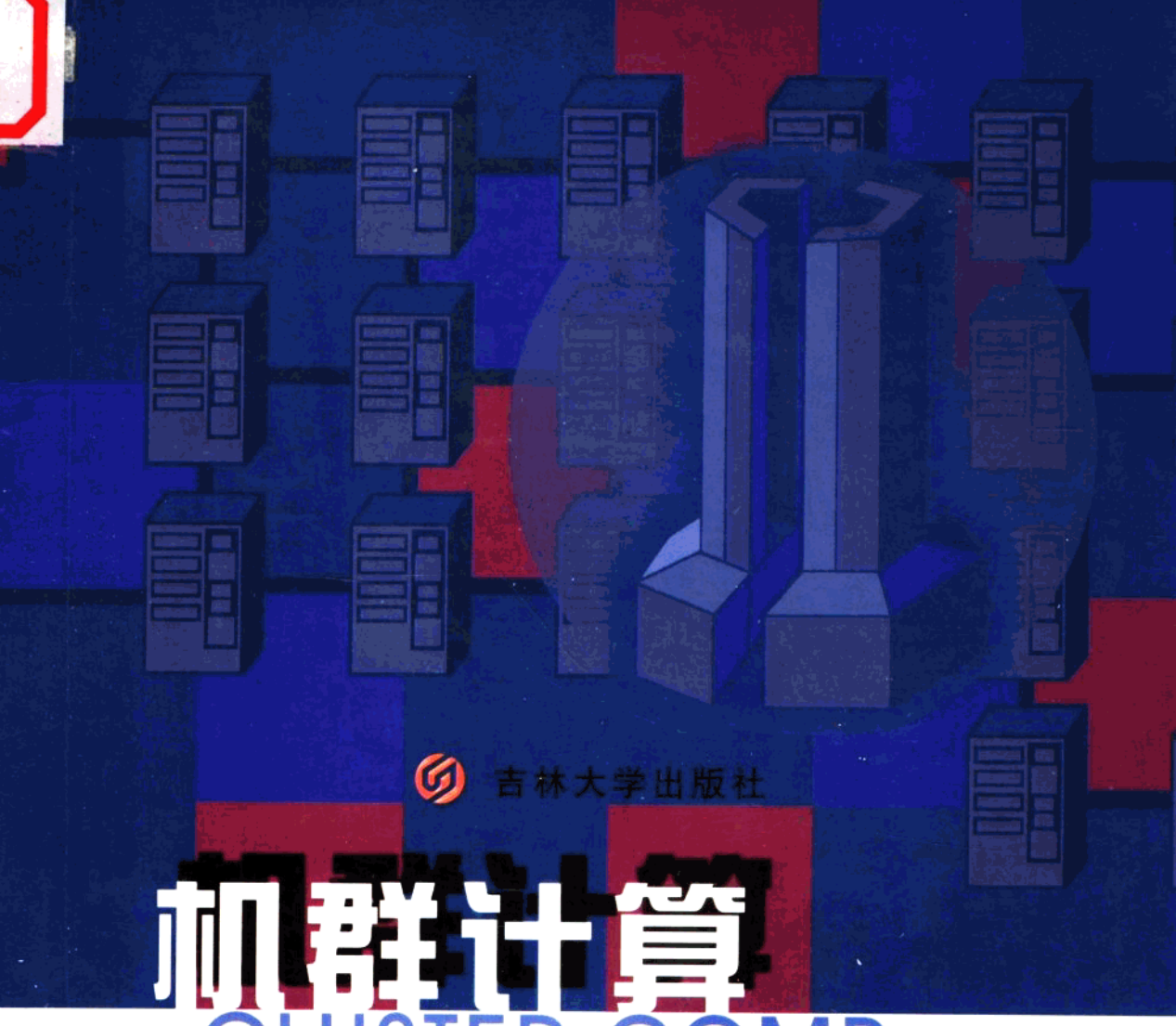

吉林大学出版社

机群计算

CLUSTER COMPUTING

鞠九滨等著

前　言

机群计算(Cluster Computing)有两个不同的含义：一个是指在工作站群(COW)或工作站网络(NOW)上并行执行应用程序(机群并行计算或工作站群计算)，另一个是指使用服务器群(Server Cluster)运行服务员程序(服务器群计算)．本书指的是前一种．

机群并行计算发展的历史可以大致划分为两个阶段．最早的机群并行计算研究工作的代表可认为是80年代初由J F Shoch和J A Hupp研制的蠕虫(Worm)程序(1982年3月发表在Communication of ACM 25(3):172－180)，他们把多段蠕虫程序分放在用户程序的各部分中，使其能自动寻找空闲工作站并行执行用户程序．这一阶段主要解决远程工作站上的资源共享问题，属于机群管理软件或机群操作系统的研究范畴．第二阶段从80年代末V S Sunderam开发出PVM(并行虚拟机，1990年12月发表在Concurrency: Practice and Experience 2(4):315－339)开始，使得基于机群的并行编程环境进入实用化阶段，机群管理软件也更加完善，从而机群并行计算得到迅速发展．

本书结合作者们的研究工作介绍近十年来在机群并行计算方面的研究成果．首先从作者的观点综述支持机群并行计算的系统软件，包括机群操作系统和并行编程环境，对国际上已发表的典型的商品和研究性机群并行软件进行比较和评述．在此基础上结合作者们研制的两个机群操作系统，说明在机群上共享作业负载的方法和负载平衡的原理和策略，以及我们在机群并行计算方面的几个研究成果，包括并行任务调度、并行应用函数库的设计、基于Linux和Windows95支持的PVM任务调度的并行编程环境、通用并行编程环境、可视化并行编程环境和利用远程网络在大规模机群组成的巨型机上进行并行计算的方法．检查点设备和基于检查点的进程迁移在机群并行计算中的应用(负载平衡和容错)是机群并行计算研究的关键和难点，本书给出对检查点算法的一个分类方法和作者的几个新算法，以及使用检查点实现进程迁移支持动态负载平衡和容错的方法．本书还介绍了机群并行计算的性能和改进机间通信的方法．最后介绍我们在分布式共享存储器机群方面的工作，阐述我们提出的一种能适应多种应用程序的分布式共享存储器集成系统的原理和方法．

本书是作者们自1990年开始承担的下列国家科研项目的研究成果：

(1) 分布式和并行操作系统(SIDLE)，“八六三”计划项目(863－306－206－2)，1990－1991；

(2) 工作站群负载平衡软件(ILBOT)，“八六三”计划项目(863－306－02－06－2)，1992－1993；

(3)工作站间的负载平衡，高等学校博士学科点专项科研基金资助项目(9218314)，1992－1994；

(4)工作站网络并行计算模型与算法，国家自然科学基金资助项目(69273016)，1993－1995；

(5)分布处理软件技术(DPVM)，“八六三”计划项目(863－306－02－06－2)，1994－1995；

(6)远程高性能计算环境，国家“九五”攻关重点科技项目(96－743－01－06－12)，1996－1998；

（7）工作站群并行计算的进程迁移和容错，国家自然科学基金资助项目（69673012），1997—1999.

本书是我和我的研究小组其他成员的集体成果. 小组其他成员对本书的主要贡献如下：

徐高潮（博士）：Sidle(2.2)，负载平衡(2.3 — 2.6)，DPVM(4.8)；

房至一（博士）：适应多种应用程序的分布式共享存储器系统(8.1—8.7)；

陶　杰（博士）：Prolog 的并行化与调度(2.2.4，3.4)，PESS(3.5)，NCSE(7.2)；

胡　亮（博士）：Linux 上的并行编程环境(5.1)，应用性能(6.2，6.3)；

魏晓辉（博士）：检查点(4.1—4.4，4.6—4.8)；

于秀峰（博士）：远程并行计算(3.6)；

张　钶（博士）：可视化并行编程环境(7.1，7.3)；

杨　鲲（硕士）：负载指标(2.4)；

郭　雷（硕士）：Condor 检查点系统分析和移植(4.5)；

王　勇（硕士）：PVM 任务调度(3.1)；

齐　红（硕士）：PVM 协作任务调度(3.2)；

尹　玉（硕士）：PVM 应用并行库(3.3)；

关晓阳（硕士）：Windows95 上的并行编程环境(5.2).

鞠九滨

1998 年 10 月于吉林大学

目　录

第一章 机 群 软 件

本章介绍机群并行计算的基本概念，系统软件和典型系统.

1.1 机群并行计算

过去十年，计算从以主架机为中心向分布式顾客-服务员方法明显转移. 今后几年这一趋势还要继续下去，并且进一步转向以网络为中心的计算. 产生这些趋势的动力是大约二十年前 Intel 的 Ted Hoff 发明了可大量生产的微处理器. 当代 RISC 微处理器现在在性能价格比方面远超过相应的主架机. 用 RISC 微处理器装配成并行机可以超过向量巨型机. 这种高性能并行机包含专用的互连网络支持低延迟高带宽的进程通信. 但对某些类型的应用这种互连最佳化是不必要的，使用通常的 LAN 技术已足够. 这就导致用高性能工作站群取代主架机、巨型机和并行机或者更好地管理已经安装的工作站群的各种应用. 机群并行计算是正在迅速成熟的技术，尽管还有某些局限，仍然在未来的以网络为中心的计算中占有重要地位.

1.1.1 机群与并行机

当今典型的并行计算主要有以下几种：基于向量和阵列机上的向量计算，主要用于大型科学计算和工程设计；基于共享存储器的多处理机上的计算，可用作服务器并同时进行并行处理；基于分布存储器的大规模并行机上的大规模并行计算；工作站群上的并行计算.

传统上，高性能计算需要计算科学家和工程师，他们熟悉向量或向量并行巨型机以及最近的大规模并行巨型机. 但传统的巨型机购置和维护都很昂贵，一个部门或单位承担不起，通常仅为政府支持的计算中心或大公司所拥有. 研究人员访问这样的系统通常很困难. 它们经常很忙，作业队列很长，全部的周转时间也很长.

最近几年台式 RISC 工作站性能提高很快，某些情况下相当于以前的巨型机. 即使便宜的系统也有相当的性能. 如果能让许多这种工作站一起工作，其功能总和相当于一个极大的计算资源. 由于商品化和大量生产，这些工作站非常便宜，比起低档巨型机还要便宜许多，所以很多单位可以有大批这种工作站并分布在整个单位内部. 观察表明，这些工作站在相当长的时间内处于空闲状态，特别是在夜间和周末. 这些空闲机时是巨大的计算资源，利用这些资源只需很少的代价. 这些工作站中的大部分已经在网络上连接起来，仅需要一种软件把它们组织起来用做并行系统. 很多单位提供这种软件，其技术相当成熟和可靠.

随着工作站性能的迅速提高和价格的日益下降，以及高速网络的发展，利用工作站群作为高速并行计算的平台已日益受到广泛的重视和欢迎. 工作站群也称为工作站网络，是把一群工作站用网络连接起来，配上相应的软件系统. 该系统具有工作站计算能力强、使用方便灵活、系统可扩充性好、有较好的性能价格比等优点. 基于网络的并行计算方法已经被证明是一种有生命力的、高效的方法. 松散耦合计算机上的群计算或并行处理，是一项迅速发展的技术，为实现高性能的应用提供了巨大潜力.

机群并行计算在 1992 年度的 ACM/IEEE 超级计算会议上受到极大的注意. 有人预言，机群或工作站群将成为提供高性能计算的主要手段，将在很多情况下取代传统向量机和并行

巨型机[1]. 例如美国 Lawrence Livermore 国家实验室用 18 个 IBM RS/6000 工作站取代了使用很久的 Cray X-MP[2]. 同年 Gordon Bell 奖(表彰在应用巨型机求解科学或工程问题方面作出重大贡献者)授给这个研究组，他们在由任意选择的分布在整个美国国内的异构型节点组成的虚拟机上运行大规模并行随机聚合物链模拟程序[3]

工作站群环境上的并行计算的研究最近几年在国际上尤为活跃，许多国外学者和厂商都预言它将具有与大规模并行处理一样的地位，主导今后并行计算的发展. 作为工作站群组成单元的工作站的性能以每年 50%的速度提高. 但价格不断下降，使人们看到工作站群的推广一定会越来越广泛.

最常见的机群是在局部网络上互连的一组工作站，但不必局限于此，机群可以由在一个机架上的紧密耦合的若干处理器组成. 例如流行的商品系统 IBM SP2 和 Cray T3D，它们使用由专用的高速互连网络连接的高性能 CPU 商品.

1.1.2 机群并行计算的系统软件

高性能工作站机群能从过载的巨型机分担一部分作业，提供批处理环境，以及当作便宜的并行机. 但是，必须有一个系统软件来管理机群，给用户提供一个单一的计算资源. 该软件功能包括机群管理、机群编程/调试、性能/执行分析、源代码分析/重构等. 支持机群并行计算的软件系统按其功能和实现方式可分成三类：机群管理软件(CMS)、分布式操作系统和机群计算环境(CCE). 这种划分不是绝对的，有些软件的功能界于它们之间. 机群并行计算系统的软件组成见图 1.1.1.

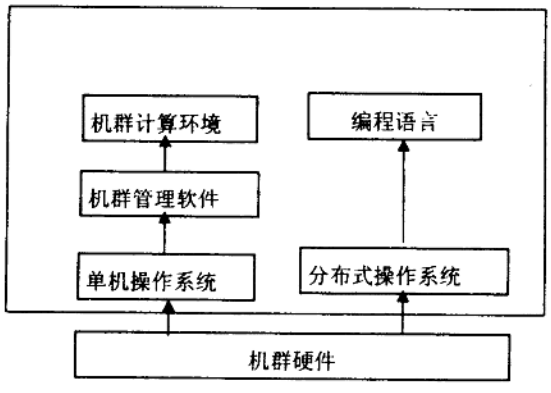

图 1.1.1 机群并行计算软件组成

机群管理软件

这类软件管理提交给工作站群的应用作业，包括传统的批处理和排队系统. 在此模型中，用户提交作业并说明需要什么样的机型和 CPU 以及内存数量，由机群管理软件调度到适当的机器上执行. 这类软件的使用越来越普遍，其功能包括负载平衡、提高 CPU 的利用率和容错等. 该软件保持机群中的所有机器处于忙碌状态并使这种作业的执行对交互式用户和其它非批量进程的影响减少到最低限度. 整体上能增加用户应用的可靠性和吞吐量. 用于管理这类机群的软件有 LSF、Condor 和 DQS 等. DQS 是一个网络排队系统，它可选择最合适的工作站运行一个给定的作业. Condor 与此稍许不同：当一个作业提交时，最初在一个空闲工作站上执行；当此工作站不再空闲时，该作业便被移往另一个合适的工作站.

在 CMS 批处理系统上运行一个作业时通常产生某一类资源说明文件，它一般是一个 ASCII 文本文件，用文本编辑器或在图形用户接口帮助下产生，包含一组关键字，由 CMS 解释. 可用的关键字的数量和性质依赖于 CMS 包，但至少包括作业名字、最长运行时间和要求的平台类型. 一旦完成，该作业说明文件由用户工作站上的顾客软件发送给主调度程序. 主调度程序是 CMS 的一部分，它保持该机群可用资源的全局状态(形成的队列和它管理的工作站上的计算负载)、监视作业的执行、保证作业能成功地执行完. 如果某个作业中途夭折不是因为应用程序运行时的差错造成，则主调度程序将调度它再次运行. 机群的每个工作站上有一个守护进程，每隔一定时间向主调度程序报告其状态. 主调度程序的一个任务是平衡它所管理的工作站上的负载. 所以当新作业被提交时，它不仅要满足其资源要求，而且要保证所用资源是负载平衡的.

典型的，批处理系统具有多个队列，每个对应于不同的作业类型. 例如，一个队列用于同构型机群，主要服务于并行作业；另一个队列可能在功能强大的服务员上用于大计算量作业；还可能有一个队列用于需要迅速周转的作业. 可能的队列数很大，依赖于系统上的作业的通过量.

CMS 是在机器现存的操作系统之上(核外)工作，安装时不需要修改内核，就像安装其它软件包一样. 可以把这种软件叫做机群操作系统.

分布式操作系统

分布式操作系统是这样的系统，它支持对各类系统对象的一致访问，支持程序的并行执行. 它提供系统调用、命令和文件系统访问的方式使得该系统表现为运行在单一平台上的单一系统. 程序员只需按照传统模型编程. 这类系统支持分布式共享存储器模型，即任何参加方的物理内存均可被任何其它参加方系统通过操作系统访问. 这样，几个线程可在由各物理处理机提供的虚拟地址空间上操作. 操作系统调用将自动处理存储器的一致性.

分布式操作系统与机群管理软件不同，在内核中实现机群管理软件的功能.

机群计算环境

这一类机群并行计算模型是把工作站当作分布式并行机中的处理部件. 这一类模型的机群软件典型地用做应用环境，帮助用户在此机群中的不同系统上同时执行一个应用程序的不同部分. 属于这类的软件有报文传递系统和并行编程环境. 就是这一类机群模型可以代替传统的向量机或并行巨型机.

报文传递系统提供并行执行设备的方法是在远程执行作业的不同部分. 其并行性是通过独立执行若干进程达到的，这些进程通过传递报文协调它们的处理. 报文是相当小的数据段或同步信息. 作业由服务员管理. 服务员与软件库一起还提供报文传递原语和协调原语. 报文传递系统可用标准操作系统和连网设备实现小组合作和报文原语.

很多从事全分布操作系统研究的学者指出，大量修改内核的实现是非常费时的，对供应商说来是没有吸引力的. 除了简单的报文传递外，另一个是写一个比较综合的并行编程环境. 这一工作的主要部分是扩充程序设计技术，特别是面向对象技术. 这里只考虑关于能对其进行操作的对象的程序模块及其对这些对象的作用. 这就扩充了构造和组织传统系统的程序设计方法论. 只有函数的外部说明的性质及其作用才有意义. 并行编程环境的功能介于分布式操作系统功能和报文传递系统功能之间，不提供透明性(如操作系统调用在本地执行还是在远程执行)，但提供比纯报文系统更为丰富的设备和更容易使用. 该环境由运行在所有参加方机器上的服务员和一个基层软件库支持，用户程序可通过这个库请求本地和远程服务.

这对程序是透明的，因为该库跟服务员联系发动远程服务，跟内核或本地服务员联系发动本地服务．该软件库可支持一组非常丰富的调用，并能使用网络提供与一个部分共享虚拟地址空间的近似．

CCE 中有的也工作在机器现存操作系统之上，安装在核外；有些则需要用具有某些服务的核替换现存操作系统的内核，因为某些功能如虚拟共享存储器不能在核外实现．PVM 的免费版本安装在现存操作系统之上，而其商用版本则并入核内以使性能最佳化．

应该指出，机群管理软件的某些功能可在并行编程环境中实现．本书作者们经常采用这种方法．

1.1.3 机群并行计算的问题

在引进机群并行计算工作中，用户可能接受它很慢．他们提出的基本问题是用机群环境编程是如何困难、机群并行计算是否能提供高性能计算结果以及其它一些担心的问题．

并行程序设计

虽然报文传递系统相对容易安装并且用户接口掌握起来也不困难，但用户缺少在设计或实现并行算法方面的支持．用户要划分数据空间，确定对该数据划分进行并发处理的任务，确定和实现任务之间的通信，对任务进行调度以保持合理的负载平衡和任务同步．所有这些都缺乏自动支持．报文传递并行计算的最新方法类似于过去的汇编语言程序设计．此外并行程序由于各种执行次序通常是不确定的，同步问题(如竞争条件)可能是短暂的，以及并行程序的调试也十分困难．所以开发并行应用程序对缺乏经验的人是很成问题的．

尽管还没有并行程序设计风范或环境的整体标准，当前趋势是各并行系统的厂商都支持共享存储器程序设计模型，包括物理上分布的存储器系统．这对于通常可能具有顺序程序设计背景的程序设计人员是一个肯定的优点，对于从事的主要活动不是程序设计的科学家和工程师来说更是如此．不幸的是，报文传递程序是基于分布式存储器模型的，一般需要完全重新设计的算法．

这还意味着通常不能直截了当地把现成的应用程序从串行环境移植到机群上．一般对并行计算机均如此，与早期向量计算系统的用户的经历相同．向量计算机基本上是单指令多数据(SIMD)并行机，大多数科学和工程的程序代码有大量 SIMD 内容，通常表示为 Fortran do 循环．这对于即使较早的编译程序来说，抽取 SIMD 内容也是相当容易的．很多用户能注意到在此系统上执行现成的程序时立即得到性能改进．但从程序抽取 MIMD 内容通常要求很大的创造性和用户的经验．一般说来，将程序从串行环境移到并行环境需要重新设计基本算法并重新编写程序．

并行粒度

限制报文传递性能的一个因素是在任务之间交换报文所需时间，或报文传递延迟．这在具有高速互连的紧密耦合系统中不是个问题，但对使用 10MBPS 以太网的工作站这是一个明显的瓶颈．对于大粒度应用这也不成问题，因为它的任务是以计算为主，很少通信．在细粒度应用中，任务之间必须频繁地交换数据，由于大通信延迟，效率极低．特别是它们在有大批处理机可用的环境中可扩充性不好．很多应用程序在基于以太网的机群上执行得不好．但对于某些类型的计算问题，这种方法比起专用的高性能计算机不失为一个好方法．

对主人的影响

一般说来工作站为个人、小组或单位所有，并为所有者专用．这种所有关系常常妨碍组

成机群. 典型的，有三类所有者. 一类所有者使用其工作站收发电子邮件或写文章. 另一类是软件开发者，使用工作站编辑、编译、调试和测试软件. 最后是需要很强计算能力的工作站运行大批如模拟程序等计算量很大的人. 只有最后一种需要额外的计算资源，利用前两种所有者的空闲机时满足其需要. 这说起来容易，做起来难，需要融洽的协商才能成为现实. 人们乐于使用别人的机器，但不愿意让别人使用自己的机器. 机器的主人担心外来的任务会干扰自己使用其机器. 解决办法是可以将报文传递系统和负载平衡软件一起使用，当机器被交互使用时降低外来任务的优先权. 但是，如果外来任务需要很多内存时，不管其优先权多低，仍然要降低工作站交互作用性能. 用户总是认为外来任务会降低其工作站的响应时间(即使不会).

安全

另一个最担心的问题是安全问题，特别是在跨越部门的机群环境中更为突出. 允许远程用户通过报文传递系统访问本地机器时，未被授权的用户也可能通过同样的途径访问. 可以使用防火墙阻止，因为报文传递不能跨越防火墙. 但随着这一措施的广泛使用，机群的规模越来越可能局限在本地网络.

1.1.4 机群并行计算的前景

上述并行计算的困难将逐步克服. 机群并行计算代表高性能计算的未来，并且现在随着多处理器计算机和基于线程的操作系统的普及，它甚至进入主流计算. 最后，标准技术和风范将出现，支持研制者的软件也将可用. 现在，程序员参加工作站群并行程序设计正处于利用新技术的好时期. 基于 10MBPS 以太网的机群的长延迟问题不大，因为较快的网络(如 ATM 和高速以太网)将来会很普遍. 与工作站共享有关的社会问题和安全问题可能意味着机群只能广泛地用于单位内部或防火墙后，但在很多情况下它仍能提供巨大的计算资源. 随着负载共享软件不断成熟，机群的任务对交互作用用户的影响也不成问题. 此外，某些校园工作站机群(如跟大学生实验室相联系的机群)有利于用于通用计算，让报文传递系统的用户访问这些机群不会再产生更严重的安全问题.

机群并行计算没有按我们预期的速度被接受，最可能的原因是它提供一种不同的计算方法，需要重新学习. 这一技术对于肯花费时间熟悉它的人，特别是没有其它高性能计算系统可用的单位，是个非常有用的财富.

使用工作站群提高用户应用程序的吞吐量在美国和欧洲已日益普遍. 现在已产生一大批机群管理软件包，几乎都来自研究项目，其中很多已为供应商采用. 其重要性不仅从这一点看出，而且可从世界上大多数主要计算设备广泛地安装这些软件看出.

PC 机和工作站的使用无疑会大大增加. PC 机和工作站在性能上的差别将消失. 使用这些系统进行并行计算具有重要意义. 当前使用队列平衡负载的批处理系统，如果具有最小管理开销时，可能开发和利用当前空闲机时的一半. 另一半可能需要更大的代价. 人们对开发用于并行计算而具有最低要求的通用系统有巨大兴趣. 如果采用现有系统和标准，未来的硬件开发者就可能将他们的系统用于大量并行计算方面，并行使用并不限于大量科学处理方面.

机群管理软件是大量研制的和需要的，并行程序设计环境的研制也很广泛. 随着基于微核的操作系统的出现并被用做商业操作系统，使得在核外使用标准工具实现的操作系统级合作的研究成为可能. 它可能为超大规模和灵活的并行系统提供一个基础.

1.2 机群管理

机群既是批处理的另一种便宜方法，又便于进行并行计算. 高性能机群能从饱和的向量巨型机上卸载，用较小的开销提供与之相当的周转时间. 如果机群中的工作站高速互连，可以作为一个价格十分便宜的并行计算机使用. 然而并不是仅有一组工作站或 PC 机在网络上连接就构成机群，还必须由机群管理软件(机群操作系统)来管理这些汇集在一起的计算能力，高效地在众多用户间分布，通过使用机群管理软件来作为单一的计算资源运行.

目前尚没有一个真正的机群管理软件标准. 本节介绍机群管理软件的功能和几个最流行的机群管理软件，对现有的主要系统进行比较.

1.2.1 机群管理软件的功能

从用户和管理员的角度看，机群管理软件应该具有下面若干功能. 但每个机群管理软件不一定具备所有这些功能.

支持的计算环境

如果一个机群系统在某个地方不支持流行的机器，那么它的使用就会受到限制. 有两类机群环境：同构型和异构型. 同构型的计算环境由具有相同结构、运行相同操作系统的计算机组成. 异构型计算环境由不同结构或运行不同操作系统的计算机组成. 许多场合由于各种资源(图形，计算，I/O 操作等) 和性能考虑需要使用大量不同计算机. 有些 CMS 仅能在单一结构计算机和操作系统上运行，大多数可在多种结构计算机和操作系统上运行.

支持的应用程序类型

批作业　支持作业的提交. 机群的一种广泛用处是从饱和的巨型机上卸载. 对于在内存和 CPU 方面要求不高的小型批处理作业，机群经常比巨型机提供更短的周转时间.

交互作业　机群应为用户提供可以执行交互作业的功能. 输入、输出和错误信息都应能返回到用户交互使用的机器上.

并行作业　配合并行软件包(如 PVM、MPI 或 HPF)可运行并行程序. PC 机和工作站价格便宜，加上可购买新的独立部件代替旧的，便于升级，因此机群可作为可扩充的并行机使用.

队列机制

当作业被提交时，运行在用户机器上的顾客程序试图寻找一个能立即运行该作业的系统. 若找不到合适的系统，作业将排队等待. 队列是作业等待调度的场所. 通过建立一个调度脚本、定义给定队列的属性以及它如何调度作业，调度程序可以访问主机上的资源可用信息并根据它分配作业. 主调度程序编制一个满足用户资源要求的队列表. 若作业已请求了某指定队列，则该作业被分派，否则作业直到能立即启动才被分派. 调度程序定期查看排队的作业是否能运行，并根据最佳算法来分配它们. 一般根据队列顺序号来调度作业. 也可以赋给作业一个优先数，使它在待处理的作业表中移到前面或后面. 调度脚本可写成实现优先级机制，并为它确定地点. 优先级可以是简单的队列内和队列间优先级，但并不限于这些.

每台机器上的守护程序向主调度程序发送负载和作业状态. 提交作业的机器根据用户要求向主调度程序查询最佳可用队列来运行该作业. 队列中的调度算法随场合不同可变，可选项包括 CPU 空闲时间、内存大小、空闲内存大小、CPU 数、空闲的暂存磁盘空间大小、队列运

行长度、交换速率、当前运行进程数以及磁盘 I/O 量.

可创建不同的队列服务于不同的调度和资源分配. 支持多重可配置的队列对于管理大的多厂商机群是必要的，因为这时系统中的作业类型很多，从短的交互作用的对话到大计算量的并行应用程序. 然而，也可以在一个机器上只建立一个队列.

作业调度和分配策略

一个作业进入系统后，知道何时何地以及如何被调度是很重要的. 如果调度不当，作业就不能被尽可能高效地处理. 大多数机群管理软件中的调度都具有如下功能：

作业分配原则 为了达到最佳的负载平衡，分配作业时必须考虑系统的许多属性. 这些属性可能包括系统负载、目标机计算能力(CPU 类型、计算负载、内存、磁盘空间)、物理相邻性、需求的资源、主人控制的程度和单位的位置. 此外系统还应考虑下列因素：

对用户的影响 机群软件系统应能识别机群中的机器何时被使用，为该机调度作业时应尽量减少对当前用户的影响，包括控制台和远程用户. 这可能引起作业的挂起，甚至将作业迁移到其它可用机器上.

磁盘空间 批处理作业执行时经常要写很大的数据文件. 如果系统支持检查点机制，执行作业要消耗很大的磁盘空间. 因此如果目标机没有足够的磁盘空间就不应把作业分配到该机运行.

系统负载 机群管理软件的目标是平衡所有机器的负载，在机群中分布计算负载以使每台机器做等量的工作. 在网络中可能会有一些机器空闲而另一些却忙于处理其工作负载. 在轻负载的 CPU 上执行一个作业能使它尽快完成.

对换空间 对换空间即磁盘空间中实现虚拟存贮器的部分. 作业从巨型机移到机群可能要求大量内存. 因此不应把作业分配到没有足够的对换空间的机器.

调度和资源分配的优先级 除了能使作业高效地执行，实现调度和资源分配的优先级也是十分必要的，它允许机群调度程序考虑对时间要求严格和其他有特殊要求的程序.

对工作站主人的影响 与购买新机器相比，使用现存的机器来建立或扩充机群是一种经济的办法. 利用分散的属于个人、小组和单位的机器是一个有争议的问题. 为使机群不断发展，必须保证机器主人在交互式工作时其机器不承担很重的负载. CMS 应将对主人的影响减少到最小. 当然，管理无主工作站群的 CMS 软件没有必要知道主人是谁，只要把资源分配给作业并且保证有效地达到吞吐率.

检查点 在作业执行时定期保存其状态，使得系统失效时余下的计算可在其最后一个检查点文件所记录的点开始. 这个设施能保证长期运行的作业在发生系统失效时也能最后完成. 它和进程迁移设施一起提供动态负载平衡和容错功能. 做检查点需要额外资源支持，运行时间不长的作业不宜使用.

进程迁移 指不必再从头启动程序就可把正在执行的作业从一个工作站迁移到另一个，通常当主人收回其工作站时使用. 为达到负载平衡也使用，将重负载工作站上的进程迁移到轻负载工作站上运行. 进程迁移要在网络上移动大批数据，产生很大开销，会严重影响网络用户，必需权衡得失.

由于在异构型环境中实现检查点和进程迁移比在单一结构中困难，一般提供这些设施的机群管理软件有一些限制. 例如只支持单进程作业，不支持信号和信号处理程序，不支持进程通信，所有文件操作必须是只读或只写. 某些机群管理软件还要求用户把其应用程序跟指定软件库重新连接. 这些限制使检查点和进程迁移设施对某些应用程序(包括必须与其它进

程通信的并行或分布式作业)不适用.

作业监控和重新调度 CMS 应能监视作业的运行，当失效时重新调度运行.

作业暂停和恢复 能暂停然后恢复一个作业的执行对于减少作业对工作站主人的影响是特别有用的. 当机器主人在控制台登录时，大多数系统都能挂起当前所有正在该机上运行的作业. 该机空闲时挂起的作业又被恢复执行. 若主人继续使用时，作业就被迁移到另一台机器上；如果没有机器可用，就在分配队列中挂起.

开销 CMS 的运行对工作站的影响应该最小. 作业运行时有一定影响. 作业暂停和被做检查点或迁移到其它工作站时也有不希望的影响. 例如进程迁移要求作业保存其状态，然后移动到另一个工作站. 这将影响工作站 CPU/主存和磁盘空间以及网络带宽.

管理员和用户对资源的控制

资源管理 管理员应能控制可用资源，允许用户对系统资源有不同的访问权限，包括 CPU 时间、磁盘空间、主存、对换空间、执行特定作业的能力和向某些机器提交作业的能力. 当用户计划运行某个特殊作业时可能要求撤回自己的机器，也可能为了运行一次某作业而增加机器. 机群不应该经常重新设置. 机器主人应该有能力不必访问根和系统管理员的帮助就能撤回其机器.

作业运行时间限制 能限制 CPU 对作业的运行时间，以便确保小作业不会在一个运行很长时间的作业后面等待，因而被拖延很久，实现在用户之间公平分配资源.

进程管理 为给定作业配置共享的或独占的资源. 资源的有效使用需要严格控制工作站上运行的进程数目，甚至允许某个作业独占工作站. 独占工作站即一个工作站上只运行一个应该程序，这可通过等待所有正在运行的作业完成或挂起当前正在运行的作业来实现. 具有特殊要求的作业可能需要独占 CPU，比如基准程序、实时环境或需要最小周转时间的高优先级应用程序等. 此外还可以控制作业在工作站上运行的优先权以利负载平衡和使作业对主人的影响最少.

作业调度控制 用户和管理员应能调度作业在何时运行. 在一天工作结束和周末时，用户们离开实验室，大多数资源空闲可用于计算. 把大计算量作业延迟执行可使小作业获得更短的周转时间，提交大量作业的用户可以把其工作推迟到有更多资源可用时完成.

图形用户接口(GUI) 一个设计良好的图形用户接口可以帮助指导用户使用系统. 一般说来，基于 Motif 的 GUI 是标准的，但现在 HTTP 协议和 WWW 的使用非常普遍，所以基于这一技术的 GUI 将成为未来的共同标准.

用户账号 很多单位的网络中都有大量不同的计算机，可从几十台到上千台. 在一个有几百台或更多机器的环境中，要求在所有机器上都建立账号就增加了复杂性. 尽量减少可能的逻辑上的和结构上的障碍将会使机群有更多的机器加入.

用户分配资源 为使一个作业在机器上执行达到最佳性能，有时需要用户指定若干资源. 这些被要求的资源可能在作业提交文件中或命令行接口中指定. 如果用户指定的资源过多，作业就要被延迟，直到所有的那些资源可用时为止. 这些资源包括机器的特定结构、内存、对换空间、操作系统及磁盘空间. 假如用户打算提交同一作业若干次，这就十分有用，并且对新用户也很有帮助.

用户作业状态查询 用户一旦提交了一个作业就再不能直接控制它. 为使用户能修改作业状态，必须有某种监控作业状态的方法，用户可以使用这种方法查询其作业状态，例如是否在运行或暂停，以及多长时间能运行完毕.