分布计算系统

鞠九滨

高等教育出版社

分布计算系统

鞠九滨

高等教育出版社

(京)112号

内 容 提 要

本书介绍用计算机网络特别是标准局部网络组成的分布计算系统的结构和实现，侧重于基本概念、基本原理和基本方法。全书共分九章。在介绍分布计算系统特征和典型的网络环境后详细介绍分布计算系统中的进程通信、命名和保护、资源控制、分布式文件系统以及工作站调度等。接着介绍分布式程序设计语言的特点以及分布式共享存储器技术。各章节中包括很多实例。最后一章集中介绍国际上有代表性的五个典型的分布计算系统。

本书比较全面和系统地反映了国际上十多年来在这一领域取得的主要成果，特别是最近几年取得的成果。

本书叙述详细、由浅入深，可作为大学高年级学生和研究生教材，也可供有关科技人员参考。

分布计算系统

鲍九滨

*

高等教育出版社出版

新华书店总店科技发行所发行

高等教育出版社印刷厂印装

*

开本 787×1092 1/16 印张 16.5 字数 440 000

1994 年 6 月第 1 版 1994 年 6 月第 1 次印刷

印数 0 001－2 409

ISBN7－04－004613－X/TP・126

定价 7.25 元

前　言

作者从1983年开始为研究生讲授分布计算系统课程，已讲授过八次，每次的内容均有较大变化。本书是在这一基础上重新组织材料写成的。分布计算系统的研究仍处于非常活跃的阶段。尽管其理论体系仍处于发展时期，但很多基础部分已趋于稳定和成熟。本书介绍的内容主要是一些基本概念、基本原理和基本方法，力求反映最新研究成果。使用的参考资料尽量选用有权威性的原著，以确保本书内容的正确性、准确性和时代性。

按照国际学术界大多数人的观点，分布计算系统可以分成紧密耦合式和松散耦合式两种。在现有的分布系统中，绝大多数是用计算机网络（主要是局部网络）支持的松散耦合式。所以本书介绍这种系统的结构和实现，主要包括进程通信、命名和保护、资源控制、分布式文件系统、工作站调度、分布式程序设计语言以及分布式共享存储器等问题。在说明基本原理和方法时，列举的例子都是当前国际上已实现的有代表性的著名的系统。最后一章集中介绍了五个典型的分布式系统，以便使读者对设计分布计算系统的几个方法和分布式系统的整体结构有个清楚的了解。由于篇幅所限，本书不讨论紧密耦合分布系统、分布式数据库以及分布式系统的应用等问题。

国家教育委员会已把分布计算系统列为高等学校计算机专业基础选修课，本书是为这一课程编写的教材。学生应在学过计算机组成原理、操作系统之后学习本课程。当然，在上本课程以前如果对计算机网络有所了解则更好。通过本课程的学习，要求学生掌握分布计算系统的概念，组成分布系统的网络基础，分布系统中的进程通信、命名与保护的特点，分布式同步机构及其在互斥、并发控制、失效恢复和多副本更新中的应用，分布式文件系统的设计问题与实现方法，分布式程序设计语言的特点，分布式共享存储器概念，并对国际上几个著名的分布式系统有所了解。各章中每节平均讲授两学时。如果学生已学过计算机网络，则2.1～2.4节可略去不讲。目录中带有*标记的部分可选择讲授。

中国科学技术大学陈国良教授审阅了全稿，对本书原稿提出不少宝贵意见，作者在此表示诚挚的谢意。

衷心希望读者指正错误和提出改进意见。

鞠九滨
吉林大学计算机科学系
1993年10月

目　录

第一章 绪 论

在计算机组成与系统结构课程里我们已经知道，从计算机诞生到现在已经经历了四代。现在，我们可以从另一个角度，即人们使用计算机的方法，来划分计算机的历史。

在50年代，程序设计人员必须预约上机时间。当他们使用计算机时，使用全部计算机资源。60年代出现了批处理技术。人们提交作业并排队等候处理，计算机每次运行一个作业，用户晚些时候来取结果。70年代人们使用计算机的主要方式是分时方法，即人们可同时使用计算机，而每个用户觉得好象他们在单独使用整个计算机。80年代是所谓个人计算机时代，每个用户在自己的办公室有一台属于他自己专用的计算机。

操作系统的发展使得新技术的发展与经济方面的考虑相结合成为可能。批处理系统的出现是由于计算机存储器可以做得充分大，以致可以装下整个操作系统以及待处理的应用程序；需要批处理技术的原因是它可以有效地使用昂贵的计算机时。使用分时系统的原因是因为它可使程序设计人员获得较高的生产率，而它之所以能出现是因为计算机变得更为便宜而且功能更强大。超大规模集成电路和局部网络技术的发展使得工作站代替了分时系统，不仅在经济上是可以接受的，而且在任何时间都能保证使用所需的计算能力。

今天，处理机的性能增加了一个数量级，可以满足个人用户的大多数需求，而处理机的价格仅是原先的十分之一。分时系统已不总是使用计算机的令人满意的方式。一个例子是出现了具有图形接口的位变换显示器，它要求图形子系统的快速响应，这只能由专用的处理机提供。

90年代的工作站功能将更为强大，它将具有高分辨率的彩色显示器并拥有声音和视频输入/输出设备。网络接口支持较高的通信速率，允许若干个通道进行实时视频传输。

分时系统给用户提供一个单一的共享环境，允许很多设备如打字机、存储器以及软件和数据共享。在工作站环境中，为了给用户提供这些服务，工作站通常用网络连接起来。工作站操作系统软件允许在网络上的工作站之间复制文件和远程登录，用户必须知道本地对象和远程对象之间的差别，以及对象放在哪个远程机器上。如果系统很庞大，这个问题将变得很严重。

系统管理问题也是个很大的问题。在分时系统年代，操作人员每天晚上进行文件系统备份复制操作，系统管理员把可用的处理机分配给最需要处理机的用户，系统编程人员只需简单地安装新的或已改进的软件。但在工作站环境中，每个用户必须是一个操作员、系统管理员和系统编程人员，在拥有上百个独立工作站的建筑物中，操作员不可能四处奔走进行文件备份复制操作，系统编程人员也不再能把新的软件简单地放到文件系统中。

以上这些问题有很多解决方法，但没有一个能象使用分时系统共享环境那样令人满意。例如最常用的一种方法是增加一个网络复制命令把文件从一个工作站传送到另一个工作站。另一种稍好的方法是使用网络文件系统，它允许文件的某种真正的共享。但在所有的方法中，除极少数以外，用户总要知道本地操作和远程操作的差别。问题在于，传统的操作系统(它仍然是现代工作站软件的基础)从来都不是为具有多个处理机和很多文件系统的环境所设计的。这种环境需要分布式操作系统。90年代将是分布式系统的十年。在分布式系统中，本地操作和远程操作对用户来说都是一样的。在一个工作站上打入的命令所要运行的程序可以在其他工作站上运行。整个系统只有一个文件系统，它为所有用户共享。外部设备也可以共享。处理机可以动态分配给最需要的地方。

分布式系统比通常的集中式系统具有更好的容错性和操作并行性。

1.1　分布计算系统

1.1.1　什么是分布计算系统

分布计算系统又叫分布式计算机系统或分布式(数据)处理系统,简称为分布式系统。关于它的定义,国内外很多学者做过研究,可概括如下:

分布式系统是由多个相互连接的处理资源组成的计算系统,它们在整个系统的控制下可合作执行一个共同任务,最少依赖于集中的程序、数据或硬件。这些资源可以是物理上相邻的,也可以是在地理上分散的。

现在对这一定义进行说明:

(1) 系统是由多个处理器或计算机系统组成;

(2) 这些计算资源可以是物理上相邻的、使用机器内部总线或开关连接的处理器,通过共享主存进行通信;也可以是在地理上分开的、使用计算机通信网络(远程网或局部网)连接的计算机系统,使用报文(message)进行通信;

(3) 这些计算资源组成一个整体,对用户是透明的,即用户使用任何资源时不必知道这些资源在哪里;

(4) 一个程序可分散到各计算机上运行;

(5) 各计算机地位平等,除了受全系统的操作系统控制外,不存在主从控制和集中控制环节。

这种计算机系统属于多指令流多数据流(MIMD)结构。现有的分布式系统只有一定程度的分布特征。

1.1.2　松散耦合分布式系统

从结构上分,有两大类分布系统:紧密耦合系统和松散耦合系统。紧密耦合系统由多个处理机经内部总线或互连网络连接而成,因此又叫多处理机系统。它们使用共享的主存储器,也可以有自己专用的主存储器(图 1.1.1)。紧密耦合系统一般限于一个局部区域,各部分相距很近,可以说是在物理上分散的,但不是在地理上分散的。各处理机交换信息是通过共享主存进行的,并具有较高的通信速度。本书不讨论这种系统。

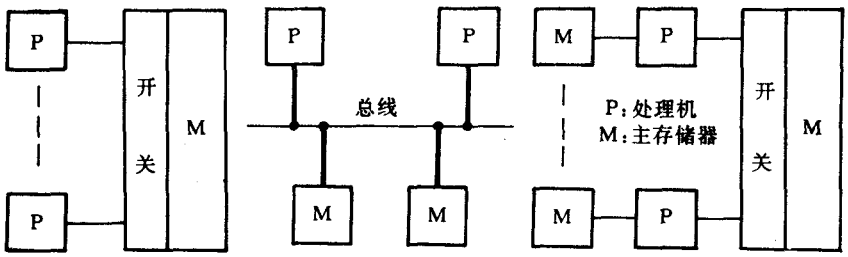

图 1.1.1　紧密耦合分布式系统

松散耦合系统由多个计算机(包含处理机和相应的主存储器)经过通信网络连接而成,是多计算机系统(图 1.1.2)。计算机之间或处理机之间使用报文交换方法通信。这种系统没有专用于各

处理机共享的主存(像紧密耦合系统那样),但各计算机的主存可以共享,构成所谓分布式共享存储器系统。通信网络可以是远程网,也可以是局部网络。目前绝大部分松散耦合分布系统都是由局部网络组成的。通常"分布式系统"一词指的是松散耦合系统,本书以后各部分的分布系统除特别说明外均指松散耦合式。

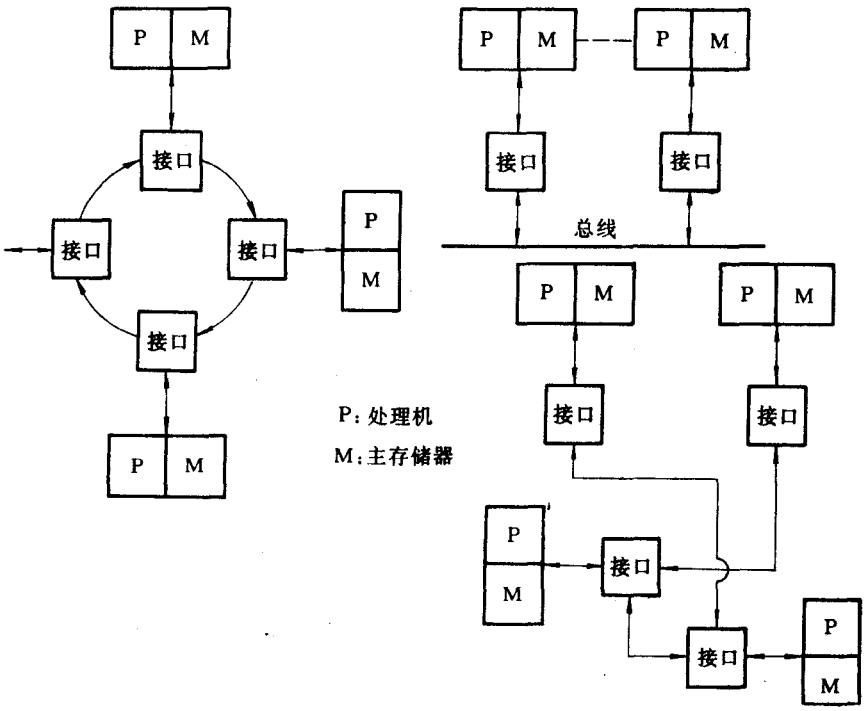

图 1.1.2 松散耦合分布式系统

根据组成松散耦合分布系统的计算机机种,可以有三种模式:小型机模式、工作站模式和处理机库模式。

在小型机模式中,系统由若干具有多用户的小型机(如 VAX)组成。每个用户在一个机器上登录,可对其他机器进行远程访问。这种模式是集中式的分时系统的简单扩充。

在工作站模式中,每个用户有自己的专用工作站,它通常配有功能很强的处理机、主存储器和图形显示终端,有时还配有硬磁盘,几乎能完成所有的工作。它所以要加入分布式系统,是因为系统中有一个全局的文件系统可供各工作站访问,另外还常常需要共享别人的处理资源及其他资源。

处理机库模式是工作站模式的进一步发展。分时系统的中央处理机(CPU)数目对登录的用户数之比通常远小于 1,工作站模式中的比数近于 1,而处理机库模式中的比数远大于 1。随着 CPU 越来越便宜,处理机库模式将越来越普及。这种模式的主要原理是:当一个用户需要较多的计算资源时,将若干个 CPU 暂时分配给此用户,工作完成时再将这些 CPU 归还给处理机库,等待下一个请求。例如要重新编译 10 个子程序(分别在不同的文件上),这时可分配 10 个处理机并行完成这项工作。上述三种模式亦可混合起来组成新的模式,例如一个系统中除了工作站之外还有处理机库。

1.1.3 异构型分布式系统

分布式系统按照其组成的计算机种类和使用的计算机网络可分成同构型分布式系统和异构

型分布式系统。异构型分布式系统由若干不同的计算机和网络的硬件或软件系统互连而成，而同构型分布式系统由相同或非常相似的计算机和网络的硬件和软件组成。异构型分布式系统的例子：由3个VAX、16个Sun组成的系统；由1个DEC-2060、1个IBM-4341和20个IBM-PC组成的系统；由12个VAX机，其中6个运行VMS而另外6个运行UNIX软件组成的系统。同构型系统的例子：只由Sun工作站(运行Sun OS)经Sun网络组成的系统。

计算机的异构性在硬件和软件方面的主要表现如下：

硬件异构性

首先是计算机指令系统不同。这意味着一种机器上的程序模块不能在另一种不兼容的机器上运行。其次是数据表示方法不同。例如不同类型的计算机虽然都是按字节编址的，但是高位字节和低位字节的规定可能恰好相反。浮点表示方法也常不一样。第三是机器系统配置不一样，即使相同的CPU也可能有不同大小的主存和外部设备配置。不同的计算机网络和同一种计算机之间的接口硬件可能不同，连接方法及通信方法也可能不同。

软件异构性

即使硬件完全相同，如果运行的操作系统不同，则由这些机器组成的系统仍为异构型。软件不兼容首先表现在操作系统所提供的功能可能大不相同。其次是操作系统提供的系统调用在语法、语义和功能方面也不相同。第三，文件系统不一样是经常遇到的现象。最简单的例子如文件名的表示：UNIX和VMS文件系统都是分层结构，但UNIX用符号/分隔路径名的各部分，而VMS文件系统中文件的表示是第一部分表示设备名，跟着一个：符号，最后的部分采取××××××·×××形式。这样，UNIX程序可以产生和使用任何VMS文件名，反过来则不行。文件系统的差异还表现在其他方面。不同计算机网络的通信协议也不同。

应该指出，使用不同种类的机器以及多种网络互连通常是不可避免的。现代科学技术使得具有不同性能的计算机型号层出不穷。一个单位对计算机性能和资源共享的要求不断变化和提高，不可能总是满足于已获得的单一型号的计算机及有限区域内的资源共享，这就使得异构型分布式系统不可避免。

1.1.4　分布式系统的优点

分布式系统有以下优点：

(1) 增强性能：增加处理部件的数目提高并行程度可以得到要求的性能；

(2) 可扩充性：随用户需求的增长，包括功能方面的要求和性能方面的要求增加新节点数，不必替换整个系统；

(3) 可靠性：由于控制、数据、软件和硬件的分散性(不存在集中环节)，资源冗余以及结构上可动态重组提高了可靠性；

(4) 资源共享：系统中的软件、硬件资源如外部设备、文件系统和数据库等可为多个用户所共享；

(5) 经济性：由于可扩充性，可以避免较大的初始投资，以及用多个微小型机代替一个大型主架机可以获得很好的性能/价格比；

(6) 适应性：与很多应用场合(如银行、铁路、商业等本来就分散而又必须互相协调的部门)很适应。

分布式系统由于有这么多优点，发展很快。用户的需求极大地刺激了分布式系统的发展，而现代半导体技术及计算机网络通信技术的进步为分布式系统的发展提供了技术保证。

1.1.5 分布式系统的新问题

由于分布式系统包含多个(可能是不同种类的)分散的、自治的处理资源,要想把它们组织成一个整体,最有效地完成一个共同的任务,比起通常集中式的单一计算机系统要困难得多,需要解决很多新问题。

首先,由于处理资源的多重性,系统可能产生的差错类型和次数都比集中式系统多。最明显的是所谓部分失效问题:系统中某一个处理资源出故障而其他计算机尚不知道,但单一计算机任何一部分出故障时将停止整个计算。另一个例子是多副本信息一致性问题。可见多重性资源使得差错处理和恢复问题变得很复杂。多重性还给系统资源管理带来新困难。

其次,系统资源在地理上的分散性带来很多新问题。在松散耦合系统中,使用报文在进程之间通信,通信将产生不可预测的、有时是巨大的延迟,特别是在远程网络所组成的分布式系统中更是这样。例如使用卫星通信产生 270 毫秒延迟。分布式的状态信息和不可预知的报文延迟使得系统的控制和同步问题变得复杂,要想及时地、完整地搜集到系统各方面的信息是很困难的,从而使处理机进行最佳调度相当困难。

在异构型分布系统中,由于各种不同资源(计算机和网络)的数据表示和编码、控制方式等均不相同,这样就产生了翻译、命名、保护和共享等新问题。

由于上述原因,分布式系统的研制,特别是软件的验证、调试、测量和维护问题变得很复杂。这正是分布式系统研制者要解决的主要问题。

1.2 分布式系统与计算机网络

分布式系统不一定由计算机网络组成,紧密耦合分布式系统就是这样。但松散耦合分布式系统是用计算机网络(远程网或局部网)组织起来的。松散耦合分布式系统与计算机网络系统都是多机系统,从硬件组成来看没有什么区别,很多作者对此不加以区别。例如世界上著名的计算机网络 ARPANET,很多学者认为它也是分布式系统。另一种观点认为,充其量它只能叫作计算机网络,不能叫分布式系统。本书采取后一种观点。这是因为,根据分布式系统的定义,尽管分布式系统由多个计算机组成,但从用户或编程人员角度看到的应该是一个单一的计算机系统整体;而计算机网络就不是这样,用户必须用命令明确指定使用一个计算机,指定一个文件从什么地方传送到另一个地方。粗略地说,如果用户能说明他在使用哪一个计算机,则他是在使用一个计算机网络而不是分布式系统。一个真正的分布式系统的用户不必知道他的程序在哪个机器上运行,他的文件在哪里存放,等等。使分布式系统具有这种性质的是它的软件:分布式操作系统。

1.2.1 网络操作系统与分布式操作系统

计算机网络在初期产生时(60 年代末)并没有操作系统,只有若干面向功能的协议如虚拟终端协议、文件传送协议、远程作业输入协议等,它们建立在进程通信协议之上。这种网络不能实现资源共享和发挥分布计算的潜力,这是因为它没有提供一个以模块方式较容易地在现有资源或服务之外创建新资源或新服务的基础,每个程序员在希望提供或使用一个新的网络共享资源时必须重新研究如数据类型转换、命令报文和回答的格式以及语法分析、命名、保护、差错控制、资源管理、同步、进程通信协议层的接口等问题,终端用户或程序员必须知道网络各主机和各种服务所要求的不同的命名和其他访问机构,同时对网络的管理也很困难。

有了操作系统，就可以提供比较一般的服务，代替象处理文件或终端等这些分散的、专用的、单一的服务。例如在操作系统中整个文件的传送只是将信息从一个文件随机复制到另一个文件的特殊情况，文件服务员将在适当的抽象级上处理数据类型及其在异构型系统中的转换。虚拟终端定义将并入操作系统的终端服务中，就象在一个单机操作系统中一样。

70年代初，为了更好地发挥资源共享的潜力，研制了网络操作系统。这种网络操作系统一般不具备分布式操作系统的全部功能。典型的例子是用局部网络把若干个人计算机和共用的打字机服务器、文件服务器连接起来，这种系统一般具有以下特点：

(1)　每个计算机都运行自己的操作系统，而不是运行共同的、全系统范围的操作系统或其一部分；

(2)　每个用户通常在自己的计算机上以“远程登录”方式或明确的指定方式使用不同的机器，而不是动态地给用户进程分配计算机，因而不能并行执行某个程序；

(3)　用户知道他们的文件都放在哪里，在机器之间移动文件时必须明确地使用“文件传送”命令；

(4)　系统根本没有或具有很少的容错能力。

网络操作系统是分布式操作系统的先驱，可以发展成分布式操作系统。最近几年，一些研究人员将一批运行UNIX操作系统的小型机用局部网络连接起来运行网络操作系统，其中有的已具备分布式系统的特征。

分布式操作系统是分布式系统的核心。和集中式系统一样，分布式操作系统是一个程序，它控制整个分布式计算机系统的资源。它给用户提供一个接口或虚拟计算机，使得用户使用起来比“裸机”方便。分布式操作系统使用户感觉不到他在使用一个多机系统，即实现系统对用户的透明性。分布式操作系统与集中的单一计算机的操作系统的差别在于，分布式操作系统是在多个处理机上运行而不是在一个机器上运行。

从实现方法上看，分布式操作系统可分成两大类：一种是在整个系统中只有一个固有的操作系统，各计算机上均运行这一系统，统一管理和分配全部资源；另一种是系统中各主机分别独立运行自己的本地操作系统，在它们的上面加上一些新成份进行通信和资源共享，这种操作系统也叫网络操作系统(NOS)。所加的新成份常常包括对本地操作系统的某些修改，但仍保持各主机在网络中的独立性。由于人们希望在不同厂商机器硬件上使用现存软件，NOS将得到很大发展。

应当注意，作为分布式操作系统一种形式的网络操作系统，与计算机网络的操作系统常常被混淆。一般计算机网络的操作系统并不具备分布式操作系统(DOS)的特性，但也可以发展成DOS。一个计算机网络仅当其操作系统具有很强的透明性时才是分布式系统。为此，NOS应该达到以下几个目标：

第一，程序(进程)、终端用户或程序员对全部分布资源应该有一个唯一的连贯的观点，不必明确地知道所需资源是在本地、远程或是分散的，主机之间的边界应尽可能隐匿。这不是说用户不能控制一个进程在何处运行或将其他资源放在何处，或不能知道资源的位置，而是说用户不必根据资源位置编写不同的程序和使用不同的终端过程，或在很大程度上甚至完全隐匿网络的操作和各局部主机的特性。当然，资源的相对位置将影响性能。这样，如果某一资源或其控制服务员由于经济或性能等原因迁移到网络中另一个系统上，仅须改地址，不必更改程序或资源访问机构。

第二，在性能方面，NOS结构的实现是有效的，可用的。本地用户进程访问本地服务时应象单机操作系统一样有效，不增加额外数目和类型的报文或系统调用。

第三，可扩充性，用户很容易在现存服务上增加新的服务而不必要求系统程序员增加新的驻存

程序。

1.2.2 计算机网络与分布式系统的区别

区别计算机网络和分布式系统最重要的一个方法是如何知道用户事实上是否在使用多个计算机。下面从三个方面来说明这个问题。

(1) 文件系统的访问

当把两个或更多的不同计算机系统连接在一起时，首先一个问题是如何把它们的文件系统合并到一起。可以有三种方案；

第一种是没有操作系统的网络使用的方法，即完全不合并方法，计算机 A 上的程序不能用系统调用访问计算机 B 上的文件，而是运行一个文件传送程序，将 B 上的文件复制到 A 上来，然后再在 A 上访问此文件。著名的例子如 USENET 网络，它使用的文件传送程序是 UNIX 的 uucp。

第二种方法是有操作系统的网络使用的方法，即把不同的文件系统连接起来，一个机器上的程序可以使用路径名打开另一个机器上的文件，例如：

```
open("/ machine1 / pathname",READ);
或  open("machine1 ! pathname",READ);
或  open("/../machine1 / pathname",READ);
```

其中最后一种是用在 Newcastle 连接以及 Netix 网络的操作系统中，将若干 UNIX 系统连到一起，在所有的 UNIX 系统的根目录上创建一个虚拟的"超级目录"。"/.." 表示从本地根目录开始上升一级到"超级目录"上。图 1.2.1 示出计算机 A、B、C 的三个根目录和它们的"超级目录"。为了访问计算机 C 上的文件 X，可以使用这样的系统调用：

```
open("/../ C / X",READ ONLY)
```

在 Newcastle 系统中，这种"命名树"实际上是很通用的，因为"machine1"事实上可以是任何目录，所以可把某机器的目录树作为"叶子"挂到"虚拟树"上的任何地方。

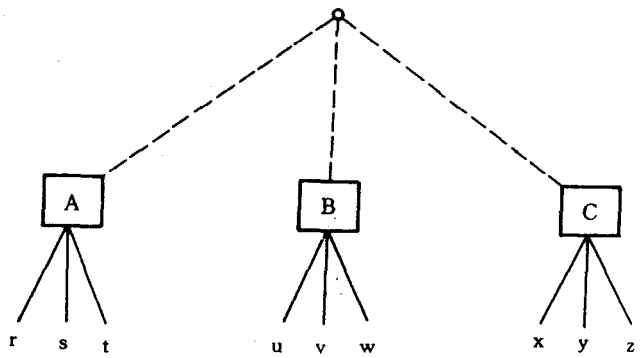

图 1.2.1 用虚拟的超级目录访问远程文件

第三种方法是分布式系统使用的方法。所有各子系统的文件系统组成一个整体文件系统，只存在一个二进制程序的"bin"目录，一个 password 文件，一个……。任何机器上的程序想要读 password 文件时都使用系统调用：

```
open("/etc/password",READ ONLY)
```

而不必指出该文件在哪里。分布式操作系统 LOCUS 使用的就是这种方法。可见，采用全局命名空间的办法是很方便的。此外，在这种系统中，操作系统可以在各机器之间自由移动文件，保持各磁盘

空间的有效利用。也可以在必要时设置任意数目的副本文件。

(2) 访问控制

UNIX 和其他许多操作系统给每个用户赋予一个唯一的内部标识符。例如,文件系统中的每个文件都有一个小表与之对应,在 UNIX 中称作 inode,它表示谁是此文件的拥有者(用户标识符),此文件放在磁盘上什么地方等等。两个机器连到一起时,便产生这样的问题:某一内部的用户标识符(UID),例如数 12,赋给每个机器的不同用户,用户 12 访问一个远程文件时,该远程文件系统无法确定这一访问是否合法,因为两个不同的用户具有相同的 UID。有三种方法解决这个问题。

一种不用操作系统的办法是要求所有要访问机器 X 上的文件的用户先使用属于 X 的用户名字在 X 上登录。这样,网络只是用作一种交换机,允许任何终端上的用户在任何机器上登录,正像电话公司交换中心允许一个用户和其他用户通话一样。这种方法对人们通常是不方便的,对程序来说也是不现实的。

较好的办法是由操作系统对不同机器上的 UID 进行变换。这样如果一个用户在其自己机器上使用 UID12,要访问的远程机器上他的 UID 是 15 时,远程机器把所有的访问都当作用户 15 访问一样处理。这种方法需要使用一个充分大的表,提供每一个用户在其自己机器上使用的 UID 和相应的其他机器上所用的 UID 的变换。

以上两种方法是计算机网络中所使用的方法。在分布式系统中,对每个用户只设一个 UID,使用它可以访问任何机器,不必经过变换。这样对远程文件的访问没有保护问题。对远程访问的处理和本地访问一样,都使用同一个 UID。由此可看出计算机网络与分布系统的差别:计算机网络中有各种机器,每一个都有其自己的用户-UID 变换,而分布式系统中只有单一的整个系统范围使用的 UID,到处都有效。

(3)执行地点

从程序的执行可以看到计算机网络中机器的边界。当用户要运行一个程序时,该程序在什么地方运行? 也有三种方法。

第一种方法是用户远程登录到任何机器上,然后在那里运行作业。这种情况下,不同机器上的进程不能通信和交换数据,但可由人工进行负载平衡。这是不具有操作系统的网络使用的方法。

第二种方法是用户在自己的终端上打入一个特殊的命令,指定一个机器运行一个程序:

```
remote vax4 who
```

这个命令是让远方计算机 vax4 运行程序 who。这时新进程的环境是远方计算机,工作目录以及要访问的文件都在那个计算机上,而不是在本地计算机上。

为了能在程序间通信和交换数据,操作系统可以提供一个系统调用"CREATE PROCESS"(创建进程)并用参数指出在何处运行这个新进程(也可使用隐含地点)。这时,运行环境一般说来也是远程机器。但在很多情况下,不同机器的过程之间的信号和通信不能正常工作。

第三种方法是用户只简单说"who"或"CREATE PROCESS",并不指出在何处运行这个程序,而由操作系统指定。操作系统可根据负载情况和要用到的文件的位置等找一个合适的 CPU 运行这个程序,也可以指定用户登录的机器运行该程序(本地执行)。使用这种方法,用户完全觉察不到机器的边界,这是分布式系统的方法。

·以上从三个方面说明计算机网络与分布式系统的区别。简单地说,用户使用计算机网络时觉察到多个计算机和网络本身的存在,而在使用一个分布式系统时觉得他是在使用一个单机系统,并不知道多个计算机和网络本身的存在,这叫作分布式系统或网络的透明性。下面一节将集中讨论透明性的一些问题。

1.3 分布系统的透明性

1.3.1 透明性的概念

先看一个在计算机网络中进行远程文件访问的例子，如图 1.3.1 所示。假定计算机网络中计算机 A 和 B 各有自己的文件系统。在 A 上访问 B 的文件 Y 应使用系统调用 open B：Y，在 B 上访问 A 上的 X 及 B 上的 X 应分别使用调用 open A：X 及 open X。如果把在 A 上写的一个程序：

```
…
open X
read
write
…
open B：Y
read
…
```

移到 B 上执行，则文件 X 已不是 A 上的而是 B 上的不同的 X 了，因此需将 B 上的 X 改名，并把 A

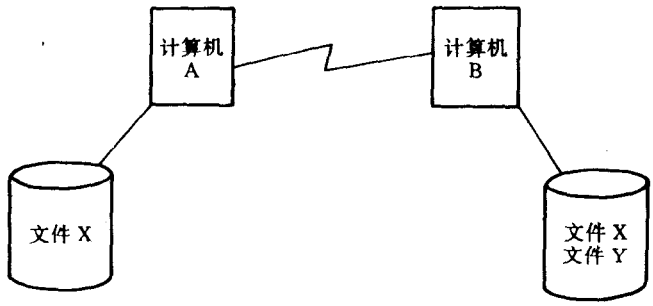

图 1.3.1 计算机网络中的远程文件访问

上的 X 移到 B 上。如果程序不动，而把 B 上的 文件 Y 移到 A 上，则此程序中的 open B：Y 将不能正确执行。可见由于文件名与所放位置有关，当文件移动和程序移动时，原来为某机写的程序都不能不加修改而正确执行。

分布式系统追求这样一个目标：打开一个文件时，不管此文件是在本地还是在远处，都应能用同一种方式，网络对用户或程序员来说是透明的，也就是说，用户或程序员看不见网络的存在，正象虚拟存储器中隐匿了外存储器一样。这样，从用户看来，网络中的全部机器表现为一个，用户看不到（隐匿了的）机器的边界和网络本身。透明性包括数据透明和进程透明，也就是说用户不必知道数据放在什么地方以及进程在何处执行。

当然，为了最佳化目的，用户仍需要某种手段来控制某种资源。但应把这种控制和用于访问资源的系统调用的语法和语义分开。也就是说，在访问资源时网络的存在不应与用户有关。

系统具有透明性时有以下一些优点：

(1) 使软件的研制变得容易，因为访问资源的方法只有一种，功能与位置无关。