一种实用、高效的虚拟远程超级计算环境*

谢 非, 杨广文, 鞠大鹏, 王鼎兴, 郑纬民

(清华大学 计算机科学与技术系,北京 100084)
E-mail: xiefei98@mails.tsinghua.edu.cn
http://www.tsinghua.edu.cn

摘要: 远程计算是指用户在本地计算机上通过互联网利用远程超级计算机上计算资源的技术.传统的远程计算方式是用户通过 telnet 协议登录到远程机器上完成各项任务.这种方式在高速、稳定的网络环境下效率是很高的.但是当网络条件比较差时,如在低带宽、不稳定的网络上,这种方式会严重影响用户的工作效率.提出并实现了一个远程虚拟计算环境,它所采用的计算方式可有效完成在低带宽、不稳定的网络环境下效率较低甚至无法完成的远程计算,其中使用了如检查点设置/恢复、压缩传送、目录树传送等技术以达到尽量减少网络流量的目的.实践证明,这在我国当前的网络条件下是一种高效的远程计算方法.

关 键 词: 远程计算;虚拟计算
中图法分类号: TP393 文献标识码: A

广域计算机网络已经成为当代计算环境的一个重要组成部分.广域网把性能、用途各不相同的众多计算机资源在物理上连通起来,使得用户在低档计算机上远程使用高性能计算机资源成为可能.但是,当前的一些远程计算系统如 RCS[1],Ninf[2]等人在进行远程计算时,对于带宽条件较差的网络环境基本上没有考虑.在这种网络条件下,用传统的远程计算方式[3]使用高性能计算机资源存在着明显的缺陷,主要表现在:

在传统的 telnet 方式下,用户从本地机通过网络登录到远程机上,在远程机上编程、调试、运行作业和分析结果.除了后台运行作业时可停止连接之外,其余步骤都要联网交互进行.所以,在 telnet 使用方式下需要较长时间占用网络,并且要求响应速度能够满足实时交互处理的需要.

如果使用通常的 ftp 工具传输大规模文件,就需要网络连接足够可靠.否则,当网络比较频繁地出现故障时,如果不采用特殊的方法(检查点和恢复机制等),大文件的传输几乎不可能实现.而大部分超级计算任务都具有程序量和数据量大的特点,大文件的传输也几乎是不可避免的.由于我国目前能够提供给广大用户使用的带宽十分有限,实时性和可靠性都不是很理想,而对于拨号上网的用户,网络的低带宽和不稳定性将造成很大的开销.

因此,为了提高远程高性能计算资源的利用率,需要向用户提供一个使用户能够方便、有效地对远程计算资源进行利用的软件平台.

本文设计并实现了一个适合目前国内网络现状的远程计算资源使用方案——虚拟远程超级计算环境,该环境向用户提供了一套完整的解决方案,用户可以在该环境中完成并行计算任务的编码、提交、调试、运行、分析等各项工作,较之传统的远程计算方式有效率高、使用方便的特点.

* 收稿日期: 2001-04-02; 修改日期: 2001-07-16
基金项目: 国家自然科学基金资助项目(60173007);国家高技术研究发展计划资助项目(2001AA111080)
作者简介: 谢非(1976–),男,湖北黄石人,博士生,主要研究领域为元计算,网格计算;杨广文(1963–),男,山西人,博士,副教授,主要研究领域为并行算法设计,网格计算;鞠大鹏(1967–),男,吉林长春人,讲师,主要研究领域为并行处理,分布式计算;王鼎兴(1937–),男,江苏吴江人,教授,博士生导师,主要研究领域为计算机系统结构,并行计算;郑纬民(1945–),男,浙江宁波人,教授,博士生导师,主要研究领域为计算机系统结构,并行计算.

1 系统的总体构成

虚拟远程超级计算环境在物理上由用户本地计算机和远程超级计算机协同工作,共同完成计算任务.它的一个重要概念就是合理划分用户本地机和远程计算机所需要完成的任务.用户能够在本地机上完成的工作尽量由本地机完成,这样可以尽量减少联网操作,只有最关键的计算过程才真正需要超级计算机的计算能力.Harris 等人[4]对在客户端和服务器之间如何进行任务划分进行了讨论.

系统的工作过程如下:用户在本地机上完成程序的编辑、初步调试等计算的准备工作,然后通过优化设计的运行系统将每个计算问题的程序源码、所需要的数据等提交到远程机上执行,执行完毕后用户回取计算结果,在本地进行计算结果分析等后续工作.

为了实现上述目标,虚拟远程超级计算环境由 3 个子系统组成:

- 本地高性能计算环境仿真系统:它在本地机上向用户提供与远程机上相兼容的开发平台,用户在该系统中编写的并行程序可以不用修改源码而在远程机上编译运行.
- 远程作业运行系统:它是本地机与远程机之间的桥梁.用户的并行程序所使用的源代码及数据文件等,都是通过这个系统传送到远程机上去的.同时,该系统还负责将远程机上的运行结果文件或其他用户指定的输出文件取回到用户的本地机上.
- 远程可视化程序运行行为监测系统:它对在远程机上运行的并行任务的工作过程进行跟踪记录,使用户可以在本地机上回放这一过程,从而帮助用户对程序运行行为进行分析,为改进程序提供依据.

下面论述上面提及的 3 个子系统.

2 本地高性能计算环境仿真系统

在互联网络系统中,用户在本地机上可能使用各种不同的操作平台,而这一部分的任务就是使用户可以在各种主流平台上完成对需要在远程机上运行的并行计算程序的编码、编译、链接及调试工作等.本地高性能计算环境仿真系统向用户提供一个与远程高性能计算环境相兼容的并行计算程序开发平台.它提供的兼容性保证在本地机上开发的并行计算程序在远程机上可以不经过修改而直接编译运行.

为了使用户能够在互联网上的各不相同的客户端结点上都能方便地完成远程计算,必须针对不同的客户端结点的平台提供相应的兼容性,这些兼容性主要包括:操作系统平台的兼容性、编程语言的兼容性、并行程序环境的兼容性.

在操作系统平台的兼容性上,本机环境支持了当前在超级计算机上使用最多的几种操作系统,它们是 Linux,Sun Solaris,IBM AIX.这一兼容性是在分析这些操作系统的编译器、运行库等一些主要区别的基础上,用预编译头文件的方式来提供的.在语言平台的兼容性上,支持在并行计算中最常使用的两种语言:C 和 Fortran.在并行环境的兼容性上,支持当前使用最多的两种并行环境:PVM 和 MPI,这两个环境都是使用消息传递机制来实现并行通信的通信原语库.

用户在安装本地高性能计算环境仿真系统时,可根据自己所使用的操作系统类型来对安装程序进行配置,安装程序根据这些配置对系统的源码进行编译链接,完成安装.之后用户就可以在该系统上使用系统所支持的语言和并行环境进行并行程序的开发工作.

3 远程作业运行系统

远程作业运行系统管理从用户作业提交到作业结果回送的全过程.该系统的主要目标是使用户通过本系统能够高效、可靠、方便地对远程机进行访问,完成远程计算的基本操作.它使本地调试成功的程序和原始数据能够加载到远程超级计算机上,然后由系统按照用户的要求对用户程序自动完成编译、链接工作,并为其分配处理机运行,最后用户可以取回执行结果等信息.为了减少占用网络带宽,系统只在需要的时候才与用户计算机建立网络连接.系统同时要负责对用户的权限控制.

远程作业运行系统的工作流程如下:

- 用户根据计算问题的情况和调试需求编写任务描述文件,提交给用户端进程;
- 用户端进程解析任务描述文件,并与远程超级计算机上的服务进程建立网络连接;
- 用户端进程与远程服务进程根据作业运行协议完成包括身份检查、文件发送、结果回取等操作.

3.1 用描述文件提交用户操作

与传统的 telnet 方式不同,本系统中用户在系统中传输数据的方式不是交互式的,而是使用描述文件的形式,一次性地将用户所要求进行的操作提交给系统,然后系统自动地、一步步地执行描述文件指定的操作,以批处理方式完成整项工作.在网络实时性不够好的情况下,描述文件的操作方式能够提高系统的效率,减少用户的等待时间,用户提交描述文件后,系统自动完成操作,基本不需要用户的干预,所以用户可以转向其他工作.另外,在并行程序调试期间,用户可能需要反复执行相同的操作序列,这时用交互方式是不合适的,而使用描述文件的方式将使操作简便、快捷.

在系统实现中,我们设计了一套完整的描述文件语法,使得用户可以在描述文件中方便地定义用户所要执行的操作.

在描述文件的语法设计中注意了两个问题:一是语法必须有足够的表现力,能够表达本系统向用户提供的各种操作,这是最基本的要求;二是语法应该简洁易懂,方便用户的使用.

本系统采用的描述文件语法的基本形式为“标题-条目”式:

① 对本系统提供的每类操作都定义相应的操作标题和条目语法规则,从而使语法具有足够的表现力,能够满足第 1 项要求.

② 在描述每一类操作时,都以括在方括号中的操作标题引导,其后每一行表示一条具体的请求,直到下一个标题或文件尾为止.例如,如果本地当前目录下有如下的子目录和文件:

```
dir1        (子目录)
file2       (文件)
```

那么,将“dir1”和“file2”保持名称不变发送到远程超级计算机用户起始目录下的操作,可以描述为

```
[send]
dir1
file2
```

3.2 优化的传输机制

系统为了在低带宽、大延迟、可靠性差的网络上有效地完成传输任务,采用增量更新和压缩传送以降低网络带宽的占用;采用检查点和恢复机制以提供可靠的服务;提供目录树的传送以及其他一些功能以方便用户的使用.

3.2.1 基于命令帧的网络连接操作接口

本系统的网络连接模块将网络操作隐藏于其中,向系统其他部分提供机器无关的网络操作接口,以提高系统的可移植性,本地与远程的网络操作是以命令帧为单位进行的.

命令帧格式

命令部分为一个 2 字节非负整数;属性部分有 16 位,字节 2 是命令选项部分,它的不同位模式组合被不同命令所识别,字节 3 表示某种安全加密措施,它被应用于后面的数据段应小于某一最大值,以便双方可以保留足够大的缓冲区来接收命令.

用户端进程和服务方进程之间采用基于此命令帧格式的协议通信,完成所需的操作.

3.2.2 增量更新

增量更新的目的在于仅传输那些确实是“新”的文件,避免由于重复传输相同的文件而造成带宽的浪费.

为了完成一个计算任务,工作过程中可能会出现“执行-修改-再执行-再修改”的现象,这时采用增量更新就会是一种比较有效的策略.例如,假设一个任务涉及 20 个源文件和数据文件.在第 2 次执行前,用户修改了其中一个文件,那么使用增量更新,系统将只发送这个修改过的文件.若文件长度均相同,则与全部重传相比,增量更

新将节省 95%的带宽.

3.2.3 压缩传送

压缩传送是减少网络流量的有效方法.本系统采用压缩传送的方法.在文件传输时,如果用户指明进行压缩传送,那么本系统采用 UNIX 的压缩工具 gzip 对文件进行压缩.在网络上传输压缩的文件.传送到目的方后,再对文件进行解压缩处理.一般的文件经 gzip 压缩后长度平均减少一半,而源程序文件通常可以压缩为原长的 1/3～1/4,从而可以显著减少网络流量.

但是,压缩却增加了 CPU 的开销.所以本系统将压缩作为可选的方式,用户可以在描述文件中指明对每一发送请求是否用压缩方式传送,给用户以权衡和选择的余地.

3.2.4 检查点和恢复机制

检查点技术是在各种计算中常用的容错方法,它的基本原理是在计算过程中随时记录当前的计算状态,当出现异常情况,计算中断时,检查点记录计算中断的时间,待故障排除后,可以根据检查点的记录恢复计算现场,从中断处开始继续进行计算[5].本系统为了向用户提供可靠的作业传输服务需要引入传输的检查点和恢复机制.

记录的检查点信息可以分为 4 类:

① 对每一大类操作整体需要设置检查点.例如,在所有的文件发送操作之前,记录发送开始,在作业提交之前记录提交开始.

② 对每一大类操作中每条具体的请求设置检查点.例如,在每次建立目录操作前,记录开始建立此目录,而在完成后记录成功或失败.

③ 传送一个目录时,目录中每个文件的发送必须进行记录.

④ 发送每个文件时,成功发送且接受的位置也应记录.

只有这样,才能通过检查点记录确定究竟在处理哪类操作、哪个请求时发生了故障.如果这是一个目录传送请求,那么还可以确定是在发送目录中哪一文件时出现了故障.当故障出现在文件发送过程中时,进一步可以确定文件已经成功发送了多大长度,从而准确地定位故障点,使恢复可以从故障点处进行.精确到文件内部的检查点是很有必要的.只有这样才能真正解决大文件在可靠性较差的线路上进行传输的问题.

根据本地机和远程机任务分工的思想,本系统检查点设置和恢复的绝大部分工作由客户机完成,而服务方进程仅在必要的时候起配合的作用.上面 4 类检查点中前 3 类都可以由客户端进程单独或根据服务方对请求的正常响应进行记录,而第 4 类检查点,即文件内部的检查点,则需要双方协同努力才能实现.

检查点记录

在以上 4 类检查点中,前两类由主近代模块完成记录.每开始一类操作之前在日志文件中记录操作的标题;在每类操作中处理每个请示时,处理前记录“开始处理”,处理后记录“成功”或“出错”.另两类检查点分别由传输模块和通用连接模块完成.

恢复操作

恢复有两种:立即恢复和事后恢复.立即恢复是在故障出现后,由主控模块自动重新连接所进行的恢复,适用于非严重故障;而当出现本地机系统瘫痪(如突然断电)、网络严重故障(如拨号上网时,电话线出现故障)等情况时,就需要进行事后恢复,在系统工作环境正常之后,再进行恢复.

✧ 立即恢复

主控过程调用每个请求处理过程时,在发现出现故障后,立即恢复.如果不能重新连接或者一次请求中出现了多次故障,那么说明可能故障较严重,终止整个程序的处理,等待事后恢复.

✧ 事后恢复

如果命令行参数中指明“r”选项,那么就进入事后恢复.

首先,读取日志文件,提取出故障点.日志记录信息可能不符合语法,这是由于用户方进程可能在记录时,因某种原因终止,从而使记录的数据不完整.读取日志文件时,程序能够处理这种情况,正确提取故障点.然后,对于故障点之前的操作,进行屏蔽.接着,转入故障出现的请求处理.当这是一个文件传输类请求时,就会以恢复方式

调用传输模块处理.

3.2.5 目录树传送

通常用户的源程序文件和数据文件是组织在一定的目录结构中的,而不是平坦地存放的.于是,很有必要提供目录树传送的功能.

(1) 目录传送的方法

可以通过两种方法实现目录树的传送:外挂的打包工具或由系统进行目录遍历.

- 外挂的打包工具

使用 tar 等打包工具,先将整个目录树转化为一个文件,将目录结构隐藏于文件之中,发送到目的机后再使用相应的方法解包恢复目录结构.这时,系统只需具备文件传送的功能,就可以用外挂的打包工具达到目录传送的效果.例如,在使用 ftp 时,通常采用此种方法传送一个目录(在使用 ftp 时,打包、解包需手工完成).

- 系统进行目录遍历

系统自行遍历目录树,完成目录的传送.

- 比较

相对于后一种方法,前一种方法简单得多.但是打包后的目录通常形成一个大文件,这样一来实际上就迫使系统以整个目录为单位进行增量更新.由于一般目录树中总会有些文件被修改,所以即使使用一种能够深入到打包文件内部的增量更新判断方法,即使确实发现大多数文件是相同的,由于少数文件不同,也必须传送整个打包文件.打包方式使增量更新机制基本上失去了作用.

- 采用系统自行遍历目录树的方法——充分发挥增量更新的作用

为了充分发挥增量更新的作用,本系统采用自行遍历的方式实现目录树的传送.于是,增量更新是以单个文件为单位进行的.

(2) 符号链的处理

目录树传送中必须考虑符号链处理的问题.

- 请求本身或其前缀为符号链

如果请求本身或其前缀为符号链,本系统将处理符号链指向的文件或目录,因为这通常是用户的目的所在.

例如,假设用户本地机当前目录下,有如下的子目录和符号链:

dir1			(目录)
dir1/link2	—>	../dir3	(符号链)
dir3			(目录)

而远方用户起始目录中没有这些子目录和符号链.当发送请求为“dir1/link2”时,由于请求本身为符号链,将发送“dir1/link2”指向的目录——“../dir3”.发送完成后,远方用户起始目录下有:

dir3	(目录)

- 遍历目录树时遇到的符号链

如果在遍历目录树时遇到了符号链,本系统将处理符号链本身,而不是处理符号链指向的文件或目录.这也是 tar 等打包工具的默认作法.因为:

① 用户建立符号链可能是为了使用方便.

例如,用户可能在本地建立了一个指向/usr/include 的符号链,以便于查看头文件,当然用户并不想将本地的/usr/include 发送到远方.

② 如果用户确实希望传送符号链指向的文件或目录,那么用户完全可以明确地指出这一文件或目录,以达到传送的目的;并且,明确的表示也方便了用户自己对目录的管理,减少混乱的产生.

③ 如果处理符号链的内容,那么目录树就会成为目录有向图,而遍历有向图会增加系统实现的复杂性.

仍然采用上例中关于用户本地当前目录和远方起始目录的假设,来说明这种情况下的符号链处理.当请求为“dir1”时,在遍历到“dir1/link2”时,仅在远方建立一个相同的符号链,并不发送“dir1/link2”指向的“../dir3”.发送后,远方用户起始目录下有:

dir1 (目录)
dir1/link2 —> ../dir3 (符号链)

3.3 可定制的安全机制

本系统采用下述方法保障一定的安全性:

(1) 身份检查

用户在本地必须输入正确的远方用户名和密码,才能使用本系统进行远程计算,这样基本上可保证使用者确有使用许可.

(2) 借助 UNIX 帐号进行访问权限管理

UNIX 系统基于用户帐号的访问权限管理,是一种成熟的安全措施.本系统的用户帐号采用 UNIX 系统的实际帐号,从而具备了 UNIX 的访问权限管理.

(3) 日志记录

本系统提供日志记录措施.管理员可以通过查看日志记录,发现异常操作,监督非法行为.

(4) 可扩展的安全框架

在加密系统方面,提供了一个可扩展的安全框架,为加密系统的挂接提供一个通用的接口.因为加密系统的设计和应用,需要分析和比较国内外已有的算法,验证它们的可靠性,由用户自己来进行选择或者设计自己的方法.这其中不仅有技术问题,而且涉及到加密算法的版权问题和政策问题,需要设立专门的课题,由专门的研究小组进行研究.这些都远远超出了本系统现阶段的目标.但是,为了今后可能扩展的加密系统,设计比较通用的接口,还是有必要的.

4 远程可视化程序运行行为监测系统

远程可视化程序运行行为监测系统的目标是使用户在本地机上能够掌握远程机上程序运行的情况,并根据这些信息来为将来程序的优化修改提供依据.

该监测系统分为两部分:第 1 部分运行在远程机上,根据用户的要求为用户所提交的计算任务的运行情况进行跟踪,并将跟踪结果存储在用户所指定的文件中;第 2 部分运行在本地机上,用户使用远程作业运行系统将第 1 部分的跟踪结果取回本地机后,使用播放工具对该结果进行回放和分析,以了解算法的性能和未来的优化前景.

本地机上的部分我们选择了与并行环境相对应的比较成熟的工具,即与 PVM 平台对应的 XPVM 和与 MPI 平台相对应的 XMPI.这两个可视化工具的原有功能是分别对 PVM 程序和 MPI 程序进行调入运行并对运行情况进行实时跟踪显示.为了能在远程机上也实现它们的跟踪功能,我们仔细研究了 XPVM 和 XMPI 用于跟踪监测的接口,然后将其余的部分(如实时状态显示)去掉,在远程机上单独实现了这一接口,并保持了与原始 XPVM 和 XMPI 的一致性.这样,当用户在远程机上提交并行计算任务以后,系统将对其运行行为进行详细的跟踪,跟踪监测结果可以由用户取回,然后在本地机的 XPVM 和 XMPI 上播放.从原理上说,该系统的作用是把 XPVM 和 XMPI 的跟踪工具和分析工具分离开来,分别放到远程机和用户本地机上运行.

5 结 论

本文所提到的虚拟超级计算环境已经开发了可以使用的实际系统.本系统现在正在国家高性能计算环境的结点机同方探索 108 集群计算机系统上运行,作为国家高性能计算环境的一部分,向 INTERNET 用户提供虚拟计算服务.

本系统当前仍有一些不完善的地方,如现在的用户使用界面是全字符模式的,使用起来不够方便,有必要将其操作界面图形化.而在远程机上,由于 MPI 环境的多样性也存在一些兼容性问题.对这些不足我们正在进行改进工作.

References:

[1] Sato, M., Nakada, H., Sekiguchi, S., *et al*. A network based information library for global world-wide computing infrastructure. In Hertzberger, B., Sloot, P., eds. High Performance Computing and Networking. Lecture Notes in Computer Science 1225, Springer-Verlag, 1997. 491~502.

[2] Helary, J-M., Netzer, R., Raynal, H.B. Consistency issues in distributed checkpoints. IEEE Transactions on Software Engineering, 1999,25(2):274~281.

[3] Arbenz, P., Gander, W., Oettli, M. The remote computation system, Parallel Computing, 1997,23:1421~1428.

[4] Christopher, S., Margo I, S. MiSFIT: constructing safe extensible systems. IEEE Concurrency, 1998,6(3):34~41.

[5] Harris, V. Thin-Client/Server computing. In: Computer Technology Review. West World Productions Inc., 1997. 42~45.

A Practical and Effective Virtual Remote Supercomputing Environment*

XIE Fei, YANG Guang-wen, JU Da-peng, WANG Ding-xing, ZHENG Wei-ming

(*Department of Computer Scienceand Technology, Tsinghua University, Beijing* 100084, *China*)

E-mail: xiefei98@mails.tsinghua.edu.cn

http://www.tsinghua.edu.cn

Abstract: Remote computing means that the users on local machine make use of computing resources on remote super computers through the Internet. The traditional remote computing method is logging onto the remote machines by using telnet protocol to accomplish all kinds of work. This method is efficient in a high-speed and stable networking environment. But when the network is in a low-bandwidth and unstable condition, this method will seriously influent the users' work efficiency. A computing method called remote virtual computing is presented which is suitable for low-bandwidth and unstable networks. It uses the technologies such as check-point setting/recovering, compressing directory tree transferring to furthest reduce the network flux. The practice proves that this is an efficient remote computing method under the present network condition in China.

Key words: remote computing; virtual computing

* Received April 2, 2001; accepted July 16, 2001

Supported by the National Natural Science Foundation of China under Grant No.60173007; the National High-Tech. Research and Development Plan of China under Grant No.2001AA111080