

8-10 A-NET 模拟程序中进程间通信接口的实现^{*}

The Implementation of the Communication Interface between the Processes of the A-NET Simulator

吕英华

TP316

(东北师范大学计算机科学系 长春 130024)

Abstract This paper describes how to utilize the functions of UNIX process management and process communication to form the communication channel between the processes of a A-NET simulator, the semaphore for holding communication channel, and the reading request signal. It also discusses the communicating realizable method

Keywords Pipe, Communication channel, Semaphore, Reading request signal

1 引言

A-NET 计算机是专用于执行面向并行对象语言 A-NETL 程序的,是具有数个结点并且这些结点由互连网络结合在一起的并行处理机。A-NET 计算机正在研制中,该机的各个结点均由 PE 和 router 两部分组成^[1-6]。PE 用于执行加载到本结点内的对象中的 method, router 用于与其它结点间的通信^[5]。

为了对 A-NETL 所描述的程序进行逻辑跟踪,获得设计 A-NET 机所需要的硬件数据,我们利用 UNIX 的进程管理和进程通信功能开发了 A-NET 机的模拟程序。

在 A-NET 机中,host 与各结点间、各结点相互之

间、节点内 PE 与 router 之间均具有并行性,因此考虑让 A-NET 机中具有并行性的各个部分分别和具有该部分功能的进程相对应,利用 UNIX 的系统调用 fork(), exec() 来实现构成模拟程序的若干个进程的产生。

当对 n 个结点进行模拟时,router 进程被 host 进程启动后,参照选定的互连网络拓扑信息,生成 n 个 router 进程。各 router 进程分别具有各结点的 router 的功能。生成后的 router 进程完成自身的初始化之后,分别生成与其一一对应的 PE 进程。因此,host 与 router 进程各为 1 个,而 router 进程和 PE 进程各为 n 个。

host 进程与各 router 进程、各 router 进程和本结

^{*} 本文工作获日本学术振兴会资助。

理器性能时微处理器设计策略将采用动态转移预测技术。

结论 理想的转移预测机制应该:(1)性能较高。因为在现代高性能微处理器中,即使 5% 的误预测率也会带来较大的性能损失,(2)不应因程序的不同表现出太大的差异。(3)占用硬件资源尽可能小。

静态转移预测性能较动态转移预测低,且不同的程序,预测准确率差异较大;动态转移预测要获得较高的性能,硬件开销大。所以单纯的转移预测方法不能很好地提高微处理器的整体性能。如果能够尽量减少微处理器的转移执行,多条件码和条件执行技术可以消除一些条件转移指令^[2]。对于必须执行的转移操作,再辅以转移预测机制,能够提高转移预测的准确率。因此

理想的方法是在硬件上支持多条件码、条件执行技术,尽量减少条件转移行为,对条件转移提前形成条件码,并采用转移预测机制来提高微处理器的整体性能。

参考文献

- 1 郑纬民,汤志忠. 计算机系统结构. 北京:清华大学出版社,1998
- 2 胡良校,方滨兴,胡铭曾. 转移预测的局限性及可能的解决方案. 计算机工程,1998,9:45~48
- 3 郑飞. 新一代 RISC 微处理器的结构特征. 微处理机,1995,(4):1~6
- 4 郑飞. 超标量与超流水线混合结构微处理器 Pentium. 微处理机,1994,(4):1~5

点的 PE 进程、物理上相邻结点的 router 进程通过各自的通信接口相连接,各 router 进程之间的同步由 router 管理进程的同步机构来实现^[2]。

进程之间的通信接口主要由以下三部分构成^[3]:

(1)进程间的通信通道。(2)为占有通信通道的信号量(semaphore)。(3)读出请求信号。

2 通信通道

2.1 host 进程与 router 进程间的通信通道

host 进程与 router 进程间的关系属于父子关系,所以其通信通道由 pipe 来实现,pipe 生成由 host 进程来完成。对于 host 进程来说,由于文件描述符的限制,不能生成与 router 进程等数量的 pipe,因此该 pipe 被 host 进程与所有 router 进程所共用。

写入到通信通道中的数据按数据到达的先后顺序排列。通信通道中的数据满了时,写入函数 sendpipe 将被锁住,直到读出用的 rcvpipe 将旧数据充分除去为止。

数据从通信通道读出同写入时一样,也是按数据到达的先后顺序进行。一旦被读出,该数据不再可能再次从数据通道中读出,一般情况下也不能送回通道原处。通信通道为空时,rcvpipe 可以选择两种工作方式。一是被锁住,直至至少有一个字节的数据来到为止;一是返回 0。

2.2 router 进程与 PE 进程间的通信通道

由于 router 进程和 PE 进程之间信息传送频繁,而且两者之间又是父子关系,因此让处于同一结点内的 router 与 PE 间具有一个 pipe 来作为通信通道。

2.3 router 进程间的通信通道

由于 pipe 只能用于具有父子关系的进程间的通信,所以 router 进程间的通信通道靠的是 FIFO。某 router 若和 n 个结点物理相邻的话,则该 router 应有 n 个输入通信通道,作为对 n 个输入通信通道的模拟。仅此一个 router 就应该生成 n 个 FIFO。这样 FIFO 与输入通信通道是一一对应的关系,既接近原来的 router 的实际结构又可使模拟简单化,但存储空间容量的限制不能大量生成 FIFO,因此,每个 router 进程只生成一个 FIFO,即一个 FIFO 和一个 router 的 n 个输入通道相对应。

3 信号量

上述的通信通道为众多并行执行的进程所共有,所以必须使用信号量以实现排斥控制,即向通信通道读写数据之前必须参考有关信号量的值。

router 间的信号量可利用系统调用 creat 来实现^[4],router 进程与 PE 进程间、host 进程与 router 进

程间属父子关系,因此可利用 pipe 的特性做成信号量,使用函数如下。

1 **mksem(s)**—信号量的生成 利用系统调用 pipe 生成 pipe,然后把 1 个字节写入 pipe。

2 **getsem(s)**—信号量的占有 从 pipe 描述符 s[0]中读一个字节的数据,如 pipe 为空,则返回 0,表示占有信号量失败。

3 **clrsem(s)**—信号量的释放 向 pipe 描述符 s[1]中写入一个字节的数据。

4 **delsem(s)**—信号量的消除 消除 pipe 描述符 s[0]、s[1],

4 读出请求信号

某进程向其它进程传送数据时,仅仅把数据送至通信通道,进程间的通信也不能顺利进行。这是因为受信一方并不知道以自己为目标地址的数据已送至通信通道。某进程在获得信号量后,从通信通道中读出数据,可以根据数据头部上的目标地址信息辨别本结点是否是目标地址。即使如此,进程间的数据通信也不能顺利进行,因为一旦被读出,如果本结点不是目标地址,一般情况下不能将数据送回通道原处。另外,存于通道内的数据也不能得到及时处理。为此,作为读出请求信号,UNIX 的 SIGUSR 被使用。

5 通信接口的通信协议

两个进程间进行通信时按以下协议进行:

1 送信方

(1)占有相应的信号量。

(2)利用 SIGUSR 向对方进程发出读出请求信号。

(3)向通信通道写入数据。

2 接收方 当捕获到 SIGUSR 或进程执行时遇到 SIGUSR 标志位,则中断现运行,做如下工作:

(1)读出通信通道的数据命令并判断其种类。

(2)处理此命令时,如果获得有关公共资源,则进行该处理。暂时不能占有时则设 SIGUSR 标志位,在不释放信号量的前提下返至被中断处继续执行。

(3)处理一旦结束则释放信号量。

6 通信接口中命令的传送

为实现 A-NET 机各部分的功能,在 host 进程和 router 进程之间、router 进程和 PE 进程之间定义了若干命令,各个命令以数据的形式通过其间的接口进行传送。下面以 host 进程和 router 进程之间的命令为例进行说明。

6.1 命令的定义和功能

根据 host 进程和 router 进程之间通信的需要,定义了如下 12 个命令。

代码	命令	功能
0	MES	信息传送
1	DMWR	以字为单位读取存贮器
2	DMWW	以字为单位写入存贮器
		加载 OS 时, host 用该命令设定存贮器空间分配
		也用于对 DMWR 的回答
3	DMBR	以块为单位读取存贮器
4	DMBW	以块为单位写入存贮器
		host 装载静态用户 object 时, 用该命令传送 object 数据
		加载 OS 时, host 用该命令传送 OS 本体
		也用于对 DMBR 的回答
5	HALT	结束模拟时由此命令来终止 router 进程与 PE 进程的执行
6	PSO	处理器状态的获取
7	PSR	用于对 PSO 的回答
8	EIO	模拟结果数据的获取
9	EIR	用于对 EIO 的回答
10	RESET	PE 进程初始化
11	GETPIO	host 进程向 router 进程传送信号时, 必须指定目标 router 进程的 PID。因此, router 管理进程在生成 router 进程之后, 用该命令将生成的 router 进程的 PID 通知 host 进程。
12	INITOS	加载 OS 后对 OS 进行初始化

6.2 命令传送过程

首先作如下设定:

#CMD($i, \dots, j$) 表示 i, j 等命令代码所对应的命令具有相同的传送过程

getsem(sem i) 表示信号量的获得($i=1, 2$ 以下同此)

clrsem(sem i) 表示信号量的释放

signal(i , host) 表示向 host 发 SIGNAL i 信号

signal(i , router) 表示向 router 发 SIGNAL i 信号

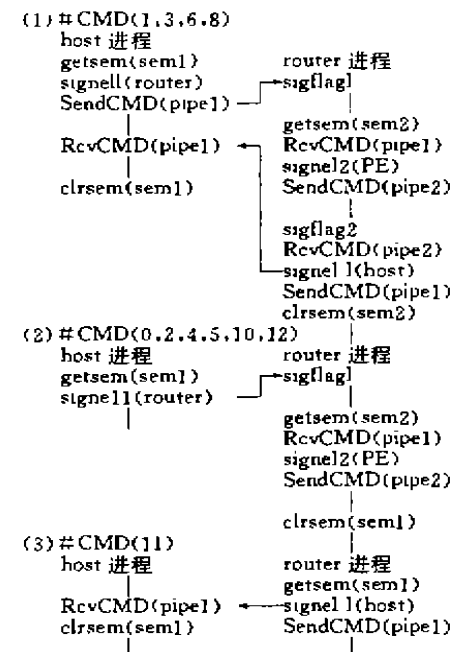
signal(i , PE) 表示向 PE 发 SIGNAL i 信号

sigflag i 表示捕获到 SIGNAL i 时设置 SIGNAL i 标志位

SendCMD(pipe i) 表示向 pipe i 写入命令数据

RcvCMD(pipe i) 表示从 pipe i 读出命令数据

基于以上的设定, 各命令传送过程表示如下:



参考文献

- 1 吕英华, 等. UNIX 进程在 PE 与 router 间通信过程模拟中的应用. 东北师大学报, 1996(3): 21~25
- 2 吕英华. UNIX システム コール による A-NET ルータ間通信模拟的实现. In: 研究纪要(日本数学教育学会春季年会发表论文). 1998. 19~21
- 3 吕英华. 并列计算机シミュレータにおける Host と Router 間のインタフェース. In: 研究纪要(日本数学教育学会秋季年会发表论文). 1998. 61~63
- 4 吕英华. UNIX のパイプ による セマフォ の作成. In: 日本第 42 回数学の文化史研究会发表论文. 1998. 11~12
- 5 A-NETL 中对象的动态生成和在 A-NET 机中的实现. 计算机科学, 1998, 25(4): 111~112
- 6 A-NETL 的消息传送和在 A-NET 机 ROUTER 中的实现. 计算机科学, 1998, 25(4): 113~114