

in操作中加快查询操作。

43. 缓存区的复制很花费时间。写一个C程序找出你访问的系统中这种复制花费了多少时间。可使用clock或times函数用以确定在复制一个大数组时所花费的时间。请测试不同大小的数组,以便把复制时间和系统开销时间分开。
44. 编写可作为客户机和服务器代码片段的C函数,使用RPC来调用标准printf函数,并编写一个主程序来测试这些函数。客户机和服务器通过一个可在网络上传输的数据结构实现通信。读者可以对客户机所能接收的格式化字符串长度以及数字、类型和变量的大小等方面设置限制。
45. 写两个程序用以模拟一台多计算机上的负载平衡。第一个程序应该按照一个初始化文件把 $m$ 个进程分布到 $n$ 个机器上。每个进程应该有一个通过 Gaussian分布随机挑选的运行时间,即该分布的平均值和标准偏差是模拟的参数。在每次运行的结尾,进程创建一些新的进程,按照Poisson分布选择这些新进程。当一个进程退出时,CPU必须确定是放弃进程或是寻找新的进程。如果在机器上有总数超过 $k$ 个进程的话,第一个程序应该使用发送者驱动算法放弃工作。第二个程序在必要时应该使用接收者驱动算法获得工作。请给出所需要的合理假设,但要写出清楚的说明。
46. 写一个程序,实现8.2节中描述的发送方驱动和接收方驱动的负载平衡算法。这个算法必须把新创建的作业列表作为输入,作业的描述为(creating\_processor, start\_time, required\_CPU\_time),其中creating\_processor表示创建作业的CPU序号, start\_time表示创建作业的时间, required\_CPU\_time表示完成作业所需要的时间(以秒为单位)。当节点在执行一个作业的同时有第二个作业被创建,则认为该

节点超负荷。在重负载和轻负载的情况下分别打印算法发出的探测消息的数目。同时,也要打印任意主机发送和接收的最大和最小的探针数。为了模拟负载,要写两个负载产生器。第一个产生器模拟重的负载,产生的负载为平均每隔AJL秒 $N$ 个作业,其中AJL是作业的平均长度, $N$ 是处理器个数。作业长度可能有长有短,但是平均作业长度必须是AJL。作业必须随机地创建(放置)在所有处理器上。第二个产生器模拟轻的负载,每AJL秒随机地产生 $(N/3)$ 个作业。为这两个负载产生器调节其他的参数设置,看看是如何影响探测消息的数目。

47. 实现发布/订阅系统的最简单的方式是通过一个集中的代理,这个代理接收发布的文章,然后向合适的订阅者分发这些文章。写一个多线程的应用程序来模拟一个基于代理的发布/订阅系统。发布者和订阅者线程可以通过(共享)内存与代理进行通信。每个消息以消息长度域开头,后面紧跟着其他字符。发布者给代理发布的消息中,第一行是用“.”隔开的层次化主题,后面一行或多行是发布的文章正文。订阅者给代理发布的消息,只包含着一行用“.”隔开的层次化的兴趣行(interest line),表示他们所感兴趣的文章。兴趣行可能包含“\*”等通配符,代理必须返回匹配订阅者兴趣的所有(过去的)文章,消息中的多篇文章通过“BEGIN NEW ARTICLE”来分隔。订阅者必须打印他接收到的每条消息(如他的兴趣行)。订阅者必须连续接收任何匹配的新发布的文章。发布者和订阅者线程可通过终端输入“P”或“S”的方式自由创建(分别对应发布者和订阅者),后面紧跟的是层次化的主题或兴趣行。然后发布者需要输入文章,在某一行中键入“.”表示文章结束。(这个作业也可以通过基于TCP的进程间通信来实现)。