基于 OpenMP 的网络管理系统性能改进

顾文杰, 李杰臣, 龚 俭
(东南大学 计算机科学和工程学院, 江苏 南京 211189)

摘 要: 为了减少网络管理系统 NBOS 中应用统计功能的响应时间, 文中采用 OpenMP 对系统进行了并行改进. 首先, 指出由于海量数据的统计, NBOS 的应用统计功能需要用并行化的方法加以改进. 其次, 仔细研究流行的两种并行设计模型(共享内存模型和消息传递模型), 结合已有的硬件平台, 选择 OpenMP 作为并行设计工具. 由于对数据进行了有效地分割和合理地分配, 保证了数据处理的高度并行性. 最后, 通过从不同时间跨度对结果进行分析, 发现系统响应时间减少了一半, 很大程度提高了系统实用性.

关键词: 并行; OpenMP; 共享内存; 性能改进
中图分类号: TP393 **文献标识码**: A **文章编号**: 1000- 7180(2008)09- 0075- 03

Performance Improvement of a Network Management System Based on OpenMP

GU Wen-jie, LI Jie-chen, GONG Jian
(School of Computer Science and Engineering, Southeast University, Nanjing 211198, China)

Abstract: To overcome the slow response of NBOS, OpenMP is used to develop a parallel program to improve it. So, firstly, it proposes that it is the mass data that results in the slow response and parallel methods should be used to improve the performance of NBOS. Secondly, it studies two concurrent design patterns, which include shared memory pattern and message passing pattern, and selects shared memory pattern to develop the program. Because of the effective data partition and the rational allocation, it makes data processing have a high concurrency. Compared with the original under conditions of different time span, the response time is halved. So, the practicability is improved to a large extent.

Key words: concurrency; OpenMP; memory-shared; performance improvement

1 引言

网络行为观测系统 NBOS(Network Behavior Observation System)是 CERNET 华东(北)地区网络中心正在开发的基于流的网络管理系统. 它针对当前 NetFlow 机制, 实现 NetFlow 流记录生成, 并对生成的流记录文件进行部分分析, 以支持网络行为学相关研究. 其中, 应用统计功能是 NBOS 的一个重要功能, 它通过统计观测区域数据包的流量来分析区域内的网络应用情况. 但是, 它存在一个海量数据的近实时处理问题, 采用传统的顺序处理方法效果不理想, 因此, 有必要用并行技术对其进行改进[1].

从应用统计的模块图(图 1)可以看出, 其分析的数据来自流文件. 流文件是流记录的集合, 由收集功能每隔 5 分钟创建的.

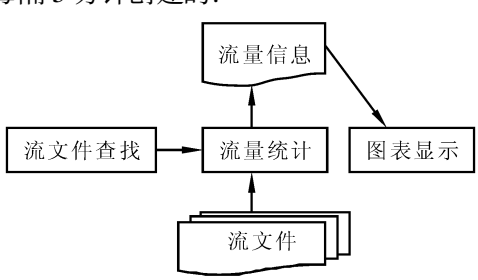

图 1 应用统计的模块

收稿日期: 2008- 03- 13
基金项目: 国家自然科学基金项目; 国家"八六三"计划项目

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

由表 1 可以发现, 每个流文件有 40 万左右条流记录, 数据量大小是 28MB 左右, 若统计一天的数据, 将会涉及到 8G 左右的数据量, 而仅仅读取这些数据就需要 2 分钟左右. 在这种条件下, 如果能够在多处理下采用并行技术, 将有效降低响应时间, 提高数据吞吐量, 增强系统性能.

表 1 流数据统计信息

类别 / 时段	流记录数 /个	流数据量 /MB	读取时间 /s
每个流文件(5 分钟)	3.88413E+05	28	0.46
一小时	4.66096E+06	335	5.6
一天	1.11863E+08	8054	133.4
一周	7.83041E+08	56379	933.9

2 并行设计

2.1 并行设计方案

目前主要的设计方案有两种:

(1) 基于消息传递多处理机模型的设计

该模型[2] 的主要特点是, 在网络环境下, 通过计算机之间的通信来实现并行. 其优点是由于每台计算机都是独立的, 拥有自己的硬盘, 有助于 I/O 并行. 其缺点是每台计算机都需要多个流文件的复本. 可以看出, 流文件的数量是巨大的, 需要很大的存储成本. 而且不同处理器上的进程之间的消息传递需要通过网络进行, 通信代价比较高.

(2) 基于共享存储多处理机模型的设计

该模型的特点是多个处理器共享内存的方式组织起来. 其优点是进程之间的通信可以通过共享内存的方式进行, 通信代价比较低. 其缺点是, 由于多个处理器共享一个硬盘, 不利于流文件的并行读取.

在进一步的分析下, 本方案采用共享存储多处理机的方法, 并在一定程度上解决了并行 I/O 的问题.

2.2 OpenMP

共享存储模型是和共享存储多处理机对应的编程模型. 共享存储模型向程序员隐藏了分布式存储的细节, 带来的好处是程序员不必自己管理节点之间的通信. 当前最重要的共享存储标准是 OpenMP 标准.

OpenMP(Open Multi Processing) API 规范[3] (简称 OpenMP), 是用户对共享内存编程模型最新要求的全面反映. OpenMP 是一个标准、跨平台、可伸缩的模型, 适合于 SMP 系统, 其特点有:

(1) 它有更小的额外开销. OpenMP 基于多线程的运行模型, 这从根本上减小了系统开销, 而且更有利于充分利用存储器.

(2) 它总结了对并行程序模型研究的一些新的成果, 更加便于手工并行化和编译器自动并行化, 同时, 也提供了较为充分的控制功能.

(3) OpenMP 主要面向循环的并行性开发, 它可以很容易的实现增量性地并行化. 它还允许表达嵌套并行性, 对某些应用, 这种方式有相当好的效果.

总之, OpenMP 是 SMP 系统上一种高性能, 而且相对比较简单的并行程序模型. 它可以作为那些中等计算规模应用的并行程序模型, 以较小的并行化代价在 SMP 集群系统的单个节点上高效的运行; 也可以作为 SMP 集群系统的节点内部的并行程序模型, 和其他模型一起完成更大规模的计算.

2.3 基于 OpenMP 的并行设计

本系统的硬件开发平台是一台拥有多 CPU 的服务器. 其具体的配置如下:

(1) 处理器: 由 4 个 XEON CPU 以共享总线的方式组成. 其中每个 CPU 的频率是 3.00G, 高速缓存(Cache) 容量为 1MB. 4 个 CPU 以共享内存的方式组成 SMP 结构.

(2) 内存: 容量为 2G.

在共享内存的多 CPU 的并行体系结构下, 由于多线程的并行处理能够有效地降低系统开销, OpenMP 被选择用来作为开发工具.

下面主要从两个方面来说明如何用 OpenMP 来进行开发.

首先, 在并行设计中最关键的部分就是数据分割. 这里主要是将不同时间段的数据交由不同的线程处理, 由于每个文件包含 5 分钟的流数据, 所以可以直接将不同的文件分配给不同的线程处理. OpenMP 的 parallel、parallel for 语句能够有效地支持这些分割数据的并行处理.

其次, 共享存储空间的互斥访问是保征并行程序正确性的关键. OpenMP 提供了诸如 critical、atomic 语句有效地实现了临界区, 保证了对共享区的互斥访问.

3 测试结果

3.1 性能测试结果

考虑到实际运行时间受多种因素的影响, 相同的测试实验每次运行时结果都会有所差别, 所以需

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

要设计多次实验. 本次实验中对不同时间范围的时间进行多次测试, 以便进行比较分析, 在实验中, 选取了时间范围分别为一小时、一天、一周, 对每种时间范围在多次不同的时间段测试, 将改进前后的运行时间以折线图的方式显示, 能够直观的比较各程序的运行时间. 结果如图 2、3、4 所示.

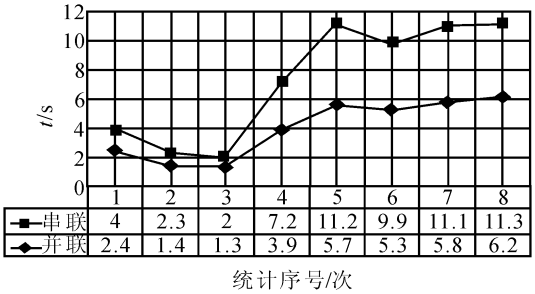

图 2　统计一小时数据量的响应时间

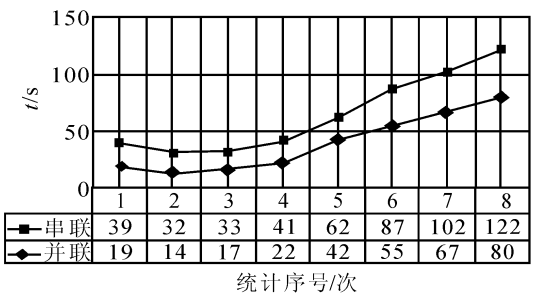

图 3　统计一天数据的响应时间

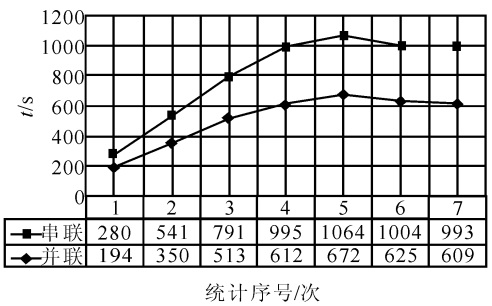

图 4　统计一周数据的响应时间

3.2　测试分析

首先, 由以上各图可见, 并行程序较串行程序运行时间减少了一半, 性能只有提高一倍, 究其原因, 主要有以下几点:

(1) 并行程序并不是完全的并行程序, OpenMP 技术实质上是在主线程运行时分出多个线程处理并行程序段, 然后再回到主程序运行, 而且由于多个线程有共享的数据, 需要进行互斥操作.

(2) 操作系统管理线程的开销, 包括线程的分配, 撤销等.

(3) 程序中开始的初始化部分并没有划进并行段中, 即程序并没有 100% 地被并行处理.

(4) 服务器并不只是运行本程序, 至少整个操作系统都会在服务器上运行, 所以服务器虽然有四个 CPU, 实际运行时不一定会有四个 CPU 处理本程序, 即并行程序的性能也受系统运行环境的限制.

3.3　响应时间分析

由以上结果可知, 改进后的程序统计一小时的数据一般需要 5s 左右的时间; 统计一天的数据一般需要 80s 左右的时间; 统计一周的数据需要 500s 左右的时间; 统计 2008 年 2 月一月的数据, 需要大约 1144s 时间, 即 19 分钟左右. 如果说, 5 分钟是人能接受的最大响应时间, 那么本功能的使用范围仅限于最多一周内数据的统计.

4　结束语

从上面的响应时间分析来看, 改进后的统计功能虽然在性能上取得了很大的进步, 但是依然具有很大的局限性, 因此, 有必要进行进一步的改进.

(1) 从修改 flow-tools 的库函数 ftio-read 这个角度考虑. 从前面的原理分析知道: ftio-read 函数被调用时, 首先将记录从硬盘复制到内存, 接着又从内存中复制一条流记录到流记录结构中. 其中, 流记录被复制了两次, 由于流记录的数量很多, 如果能够减少第二次的复制, 那么可能会提高系统性能, 降低响应时间.

(2) 从提高硬件平台的角度出发考虑. 如果能够增加 CPU 的数量和主频, 将有效地提高性能. 但是这样做的成本太高.

参考文献:

[1] Cisco System. IOS net flow feature(S) [EB/OL]. [2008-02-19]. http://www.cisco.com/warp/public/732/Tech/nmp/netflow/.

[2] Bary Wilkinson, Michael Allen. 并行程序设计[M]. 2 版. 陆鑫达, 译. 北京: 机械工业出版社, 2005.

[3] Michael J Quinn. MPI 与 OpenMP 并行程序设计: C 语言版[M]. 陈文光, 武永卫, 译. 北京: 清华大学出版社, 2004.

作者简介:

顾文杰　男, (1984-). 研究方向为网络测量和网络行为学.
李杰臣　男, (1985-). 研究方向为网络测量和网络行为学.
龚　俭　男, (1957-), 博士, 教授, 博士生导师. 研究方向为开放分布式处理理论及其应用、计算机互联网络工程、网络管理和网络安全.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net