

# 并行编程模型的研究与发展

董仁举, 祝永志

(曲阜师范大学 计算机科学学院, 山东 日照 276826)

**摘要:**并行编程模型在分布式计算中发挥着很重要的作用,随着人们对高性能计算需求的不断扩大和各种新技术的出现,并行编程模型也处于不断的发展和完善之中。对两种主要的编程模型进行了详细的分析和研究,针对前两种模型的优缺点分析并研究了两级并行模型的使用范围和优势等,最后针对硬件的新发展提出了新的编程模型的发展 TBB+MPI。并在基于 CMP 的集群系统中实现了矩阵相乘的算法。实验结果显示 TBB+MPI 在多核集群编程方面有明显的优势,因此模型 TBB+MPI 更适合于多核集群。

**关键词:**并行编程;消息传递;共享变量;两级并行模型;TBB

**中图分类号:**TP311

**文献标识码:**A

**文章编号:**1673-629X(2011)01-0092-03

## Research and Development on Parallel-Programming Model

DONG Ren-ju, ZHU Yong-zhi

(Department of Computer, Qufu Normal University, Rizhao 276826, China)

**Abstract:** Parallel-programming model takes a very important part in distributed computing. With the increasing need for high performance computing and the appearances of many new technologies, parallel-programming model is also in need of exploration and improvement. Two major parallel-programming models are compared in details in many aspects at first. Against the advantages and disadvantages of the first two models, the usage and advantages of the two-level parallel is studied. According to the development of hardware, given the future trend of parallel-programming model TBB+MPI. And has realized the matrix multiplication algorithm based on the CMP cluster system. The experiment result showed that the new model got the performance improved.

**Key words:** parallel-programming; message-passing; shared memory; two-level parallel model; TBB

## 0 引言

并行处理是现代计算的一种关键技术。近年来,无论是在高性能计算还是在普通计算方面,它都被广泛的接受和应用。有两个主要的发展促进了并行处理的发展:大规模并行处理机(massively parallel processor, MPP)和分布式计算(distributed computing, DC)<sup>[1]</sup>。两个方向的发展都需要并行编程模型的支持。

对于用户而言,所设计的任何并行算法最终总要通过并行编程在具体并行机上执行实现,不同的并行机体系结构模型其编程方式是不相同的。因而并行模型的发展对于并行计算来说显得非常重要<sup>[2]</sup>。随着多核产品的出现,并行编程模型也要随之而改变。在性能上,应用传统的模型编写的程序并不能使得性能提高很多。为了解决这个问题,Intel 针对多核平台,提出

了 Intel Building Blocks 这个针对 C++ 模板库的编程模型。文中针对硬件的发展,提出了结合 TBB 的新的编程方式 TBB+MPI,实验显示更加适合多核集群编程。

## 1 并行程序设计模型

### 1.1 共享存储模型

共享存储模型是一般的集中式多处理机的抽象,例如 SMP 结构并行机。其底层为一系列处理器,各个处理器可以对共享存储器中的数据进行存取,数据对于每个处理器来说都是可访问到的,不需要在处理器间进行数据传递,由于所有处理器可以访问内存中的同一位置,因而它们可以通过共享变量进行交互和同步。其初衷是为了共享存储多处理机<sup>[3]</sup>。

OpenMP 是用于共享存储的一个标准,已经被广泛应用到 Unix、WindowsNT 等多种平台上。它是基于线程的并行编程模型,通过编译制导语句来显示的指导并行化,为编程人员提供了对并行化的完整控制。主要包括三部分:(1)运行时库函数;(2)编译制导语句;(3)环境变量。如 OMP\_NUM\_THREADS 等。OpenMP 编写的程序只能在共享内存的并行机上执

收稿日期:2010-05-19;修回日期:2010-08-22

基金项目:山东省高等学校实验研究项目基金(2005-400);曲阜师范大学校级科研项目(XJ0734)

作者简介:董仁举(1985-),男,硕士研究生,研究方向为分布式计算;祝永志,教授,硕士生导师,研究方向为网络与分布式系统。

行,且效率较高<sup>[4]</sup>。目前 OpenMP 分别提供了支持 Fortran, C/C++ 的 API。在 OpenMP C/C++ 中,制导格式为“#pragma omp 制导选项列表”。

OpenMP 采用了共享存储系统中标准的并行模式 Fork-Join (Fork, 创建新线程或者唤醒已有的线程; Join, 多线程的聚合)。图 1 所示执行过程介绍如下:主线程在执行过程中,在遇到编译制导语句时,就根据环境变量派生出 4 个线程(Fork)<sup>[5]</sup>。此时各个线程与主线程同时执行。在执行过程中,第四个派生线程遇到制导语句,派生出另外一组线程。这些线程同时完成一项任务,对于原有线程来说,新线程组的工作类似于串行程序,对原有的线程不产生影响。最后线程组通过同步障(barrier),汇合(Join)成主线程。

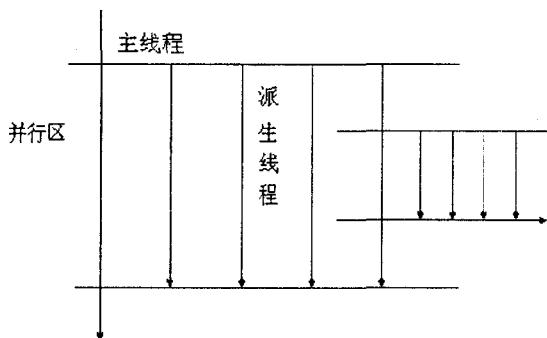


图 1 OpenMP 应用程序运行时的 Fork-Join 模型

## 1.2 消息传递模型

消息传递即用户显式地通过发送和接收消息来实现处理器之间的数据交换。在这个模型中,每个进程都有自己独立的地址空间,一个进程不能够直接访问其他进程中的应用数据,数据访问必须通过消息传递来实现。它主要用来开发大规模和粗粒度的并行性。MPI 是通过扩展串行编程语言来实现并行化的,使得程序员可以操作并行处理器的底层函数,因而为程序开发提供了更大的灵活性<sup>[6]</sup>。

它具有如下特点:

- (1) MPI 程序既可以在分布式系统又可以在共享内存系统中运行;
- (2) MPI 程序可以移植到多种系统中去,具有易用性;
- (3) 系统特别适合粗粒度并行,传递开销小;
- (4) 有很多优化的 MPI 库,如 MPICH、Intel MPI 等;
- (5) 每个进程都有自己的局部内存;
- (6) 以消息的方式,通过显式的收发函数调用完成数据在各自局部内存间的复制。

然而, MPI 存在着一些不足。由于进程的独立性和显式消息传递的特点, MPI 标准更加繁琐,基于其开

发并行程序也相当复杂。通信可能会造成很大的开销,通常使用大的代码粒度来最小化延迟<sup>[7]</sup>。

## 1.3 两级并行编程模型

针对消息传递模型和共享存储模型各自的优点而形成的两级并行模型具有更好的性能<sup>[8]</sup>。这种模型是针对 SMP 机群而提出的。模型的执行方式为在各个节点间使用消息传递的方式进行数据共享,而在各个节点内部使用共享存储的方式来共享数据,由此可以更好地利用消息传递和共享存储模型的处理数据的优势,减少了开销并且提升了性能。由 Amdahl 加速比定律可知能得到更好的加速比。节点内通信也可以采用其他的编程模型,比如 pthread, windows 线程等<sup>[9]</sup>。两级并行模型的编程过程如图 2 所示。

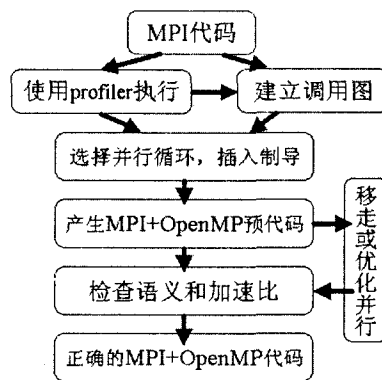


图 2 MPI+OpenMP 编程模型实现过程

这种编程模型相比于单纯的消息传递编程模型更能充分利用多处理器计算机集群的体系结构特点,在某些情况下可以更加有效地改善集群的性能,并为多处理器构成的计算机集群提供了一种良好的并行策略。

## 2 新的实现机制 TBB+MPI

随着 CMP(片上多核处理器)的出现,并行计算在计算机领域地位更加重要,使用多核 PC 构成集群成为发展方向。多核集群缺乏很好的编程模型,单纯地将 MPI+OpenMP 编程模式应用在集群上,其兼容性不是很高。因此需要新的编程模型来提高多核集群的性能。

TBB(Threading Building Blocks)的出现,简化了多核编程。OpenMP 适合那些可以通过静态分析就能做的并行,例如有确定次数的 FOR 循环并且可以被拆分。而 TBB 可以进行动态的分配,例如自动分解任务,并动态分配到各个线程上去。TBB 更强大,但应用也必然要稍微复杂<sup>[10]</sup>。文中提出将 TBB 与 MPI 的消息传递机制结合起来,不仅可以发挥出 TBB 在单台主机上多核计算的优势,还能有效地将大型计算任务分配到各个节点上分别处理<sup>[11]</sup>。

TBB+ MPI 编程模型如图 3 所示。

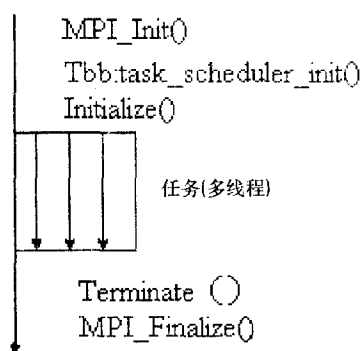


图 3 TBB+MPI 编程模型实现过程

### 3 性能测试

矩阵乘法在军事、气象预报等重要领域应用相当广泛,其在科学研究方面也发挥着举足轻重的作用。在做气象预报时,在经度、纬度和大气层方向上至少要取  $200 \times 100 \times 20 = 40$  万个网格点,当计算能力不足时,只能降低分辨率,简化方案,而影响预报的准确性。通过并行计算可以解决这些重大的计算问题。

文中通过对矩阵相乘的算法来进行试验性能分析。

实验环境配置如下<sup>[12]</sup>:

硬件:多核计算机,路由器;

软件:Microsoft visual studio 2003. net 能够支持 OpenMP 和 MPI 扩展,与 TBB 结合。

下面通过实现矩阵乘法来测试各种情况下编程模型的性能。将其分别通过 MPI、openMP+MPI 和 TBB+MPI 来实现并在多核计算机集群下来实现,测试性能数据如表 1 和表 2 所示。

表 1 矩阵规模为 500 时加速比

| 矩阵规模 500 * 500 |       |              |         |
|----------------|-------|--------------|---------|
| 节点             | MPICH | MPICH+OpenMP | TBB+MPI |
| 2              | 1.34  | 1.45         | 1.56    |
| 4              | 2.21  | 2.48         | 2.57    |
| 6              | 3.15  | 4.23         | 4.69    |
| 8              | 4.43  | 5.93         | 6.21    |

表 2 机器规模为 8 时加速比

| 机器规模 8 |       |              |         |
|--------|-------|--------------|---------|
| 矩阵     | MPICH | MPICH+OpenMP | TBB+MPI |
| 100    | 2.09  | 2.27         | 2.72    |
| 200    | 3.45  | 3.96         | 4.21    |
| 300    | 4.09  | 5.21         | 5.35    |
| 400    | 4.12  | 5.48         | 5.83    |
| 500    | 4.43  | 5.93         | 6.21    |

实验结果分析如图 4 和图 5 所示。

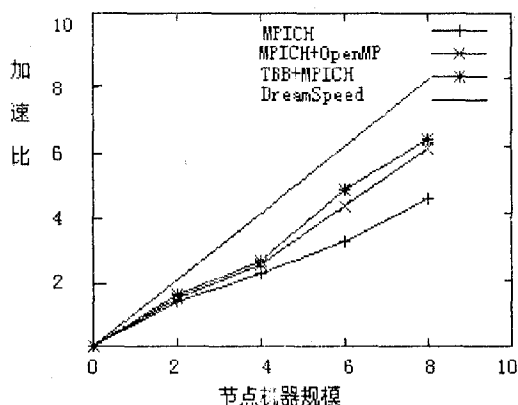


图 4 矩阵规模为 500 \* 500 时,性能比较

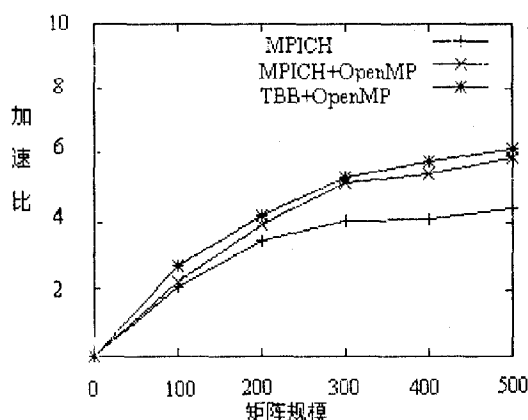


图 5 节点数为 8 时,性能比较

(1)通过固定矩阵的规模增加节点数来分析加速比,当矩阵规模为  $500 \times 500$  时,结果如图 4 所示,显示 TBB+MPI 的效果最好。开始时,节点较少并存在大量的数据交换,加速比差距较小,随着节点的增多,两级并行模型的性能提高,一方面是因为这种编程模型充分利用了多核的优势同时多线程,另一方面是多粒度的程序设计方式提高了加速比。

(2)通过固定节点个数增加矩阵规模来分析加速比,当节点数为 8 时,结果如图 5 所示,显示出两级并行模型具有较好的加速比。开始时由于矩阵规模较小本身运行时间较短,几种编程模型的性能相差不大,随着矩阵的规模增大,性能差距明显。

由此可见,新型的编程模型对于多核集群的计算性能提高有很大的帮助。TBB 将有利于并行多核编程中各方面问题的解决,同时大大降低了编程难度。对多核处理器构成的计算机集群提供了不错的编程策略。

### 4 结束语

从 OpenMP 和 MPI 的发展、改进与升级,可以看出集群环境下的并行编程模型在朝着多核多线程的发

(下转第 99 页)

### 3 结束语

文中比较全面、系统地阐述了如何在 VFP 环境下操作 ACCESS 数据库(表)问题,从而为扩大 VFP 在数据库领域的应用与安全提供了一种有益的尝试。在探索过程中,要处理好以下问题:

1) 访问数据库的方式很多,在对 ACCESS 数据库(表)处理的不同阶段,各有利弊,可根据操作情况灵活选择。

2) 两者数据表字段的类型在表述上的差异,特别要注意处理好 OLEObject 字段类型与 General 字段类型的转化问题,否则,有可能导致转化后的数据无法正确读出。处理时建议使用 VFP 提供的 STRCONV(<表达式>,<转换类型>)函数,并转换<表达式>中的单字节字符到 base64 二进制形式。

3) VFP 从 7.0 开始 SQL 语句有较大变化,需注意操作上的差别。文中的案例在 VFP 8/9 下测试通过。

#### 参考文献:

- [1] 王永国. 基于 VB 的 ACCESS 无纸化考试系统[J]. 微计算机应用, 2007, 28(1): 108-112.
- [2] VFP 完全控制大、中、小型数据库编程接口[基本介绍][DB/OL]. [2010-06-25]. <http://www.duote.com/soft/9836.html>.
- [3] 汪学祥. 在 Visual FoxPro 中访问非 DBF 格式数据文件[J]. 山西煤炭管理干部学院学报, 2002(4): 74-75.
- [4] 周松, 秦嘉杭, 周晓飞. 在 VFP 中用远程视图实现数据

的安全性[J]. 计算机与现代化, 2002(10): 52-54.

- [5] Basoz C. Linking a DB from Access 2003 to a ForPro application [DB/OL]. 2006-07-19. <http://social.msdn.microsoft.com/Forums/en-AU/visualfoxprogeneral/thread/bebecf90-90be-4df4-aacd-2a6da5e0af53>.
- [6] apple\_8180. 通过 VFP, 向 SQL Server 或 Mdb 等中保存、还原图片[DB/OL]. [2010-07-09]. <http://faq.csdn.net/read/217758>. HTML.
- [7] Hank 动态创建、压缩 Access 数据库(\*.MDB)[DB/OL]. 2000-12-29[2010-02-10]. <http://dev.csdn.net/htmls/2/2773.html>
- [8] my\_love\_xuevfp. 中怎么创建 access 数据库文件[DB/OL]. [2008-11-19]. <http://wenda.tianya.cn/wenda/thread?tid=6e23da0fe01eae2d&clk=wtptcs&exp=1>.
- [9] 李秉璋. 基于 ADOX 接口动态创建数据库方法探索[J]. 计算机应用与软件, 2006, 23(5): 41-42.
- [10] 王苏. Microsoft Visual FoxPro 9.0 中文手册[DB/OL]. 2007-11-30[2009-03-20]. [http://download.csdn.net/source/8405\\*5](http://download.csdn.net/source/8405*5).
- [11] Strahl R. Using Unicode in Visual FoxPro Web and Desktop Applications [DB/OL]. [2005-01-20]. <http://www.programmersheaven.com/2/using-unicode-in-visual-foxpro>.
- [12] wuno2001. VFP6 如何读取及写入资料到 Assess MDB 资料库[DB/OL]. [2008-11]. <http://vfp.sunyear.com.tw/viewtopic.php?t=4290>. [13] Article ID: 260405. How to import table structures and data from Microsoft Access to Visual FoxPro[DB/OL]. [2007-02-12]. <http://support.microsoft.com/kb/260405/en-us/>.

(上接第 94 页)

展,多线程不再是并发执行,而是通过多核的同时运行而达到真正的并行。因而多线程模型在并行模型中的作用越来越大。文中给出了 CMP 集群的新的编程方法 TBB+MPI 并实验验证了性能优势,充分利用了多核计算节点的计算能力,在实际编程过程中,一个主要难点就是 TBB 线程任务分配和 MPI 数据同步,这也是以后要解决的主要问题。

#### 参考文献:

- [1] Grama A, Gupta A. 并行计算机导论[M]. 北京:机械工业出版社, 2005.
- [2] 胡晨霞, 王晓蔚. 基于多核集群系统的并行编程模型的研究[J]. 计算机技术与发展, 2008, 18(4): 70-73.
- [3] Quinn M J. MPI 与 OpenMP 并行程序设计(C语言版)[M]. 陈文光, 武永未, 译. 北京:清华大学出版社, 2001: 212-245.
- [4] Utrera C, Corbalan J, Labarta J. Implementing malleability on MPI jobs[C]//Proc. Parallel Architectures and Compila-

tion Techniques. [s.l.]: [s.n.], 2004: 215 - 224.

- [5] 蔡佳佳, 李名世, 郑锋. 多核微机基于 OpenMP 的并行计算[J]. 计算机技术与发展, 2007, 17(10): 87-91.
- [6] Malyshekin V. Parallel computing technologies[C]// 8th International Conference on PaCT. [s.l.]: [s.n.], 2005.
- [7] 郭静, 祝永志, 王延玲. 基于 MPI 的动态负载均衡算法的研究[J]. 计算机技术与发展, 2009, 19(5): 150-153.
- [8] 单莹, 吴建平, 王正华. 基于 SMP 集群的多层次并行编程模型与并行优化技术[J]. 计算机应用研究, 2006(10): 254-260.
- [9] Foster I. Designing and building parallel programs[M]. 北京:机械工业出版社, 2002.
- [10] Reinders J. Intel Threading Building Blocks 编程指南[M]. 北京:机械工业出版社, 2009.
- [11] Akhter S, Robert J. 多核程序设计技术——通过软件多线程提升性能[M]. 北京:电子工业出版社, 2007.
- [12] 刘维峰, 卢伟, 许海燕. 基于局域网和 MPI 的 PC 集群计算环境[J]. 计算机工程与设计, 2005, 26(5): 1327-1329.