基于 SMP 机群的层次化并行编程技术的研究

祝永志,张丹丹,曹宝香,禹继国
(曲阜师范大学计算机科学学院,山东日照 276826)

摘 要: 针对多核 SMP 机群的体系结构特点,讨论了 MPI + OpenMP 混合并行程序设计技术.提出了一种多层次化混合设计新方法.设计了 N-body 问题的多层次化并行算法,并在曙光 5000A 机群上与传统的混合算法作了性能方面的比较.结果表明,该层次化混合并行算法具有更好的扩展性和加速比.

关键词: SMP 机群; 层次化; 混合编程; 性能分析
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2012) 11-2206-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.11.011

Research of Parallel Programming Techniques of Hierarchical Model Based on SMP Clusters

ZHU Yong-zhi, ZHANG Dan-dan, CAO Bao-xiang, YU Ji-guo
(*School of Computer Science, Qufu Normal University, Rizhao, Shandong* 276826, *China*)

Abstract: For multi-core SMP cluster systems, this paper discusses hybrid parallel programming techniques based on MPI and OpenMP. We propose a new hybrid parallel programming methods that are aware of architecture hierarchy on SMP cluster systems. We design a hierarchically parallel algorithm on the N-body problem, and compared its performance with traditional hybrid parallel algorithms on the Dawning 5000A cluster. The results indicate that our hierarchically hybrid parallel algorithm has better scalability and speedup than others.

Key words: SMP clusters; hierarchical; hybrid programming; performance analysis

1 引言

国内外目前主流的并行计算机体系结构主要有 MPP、COW、SMP 及 SMP 机群.MPP、COW 的特点是基于分布式存储,可扩展性好,但处理器之间的通讯开销较大,编程困难,其编程模型主要是消息传递,实现标准有 MPI、PVM 等.SMP 是基于共享存储的,处理器之间通讯开销较小,易于编程,但可扩展性差,主要编程模型是共享变量,实现标准有 OpenMP 和 Pthreads 等[1].

从内存结构来看,SMP 机群(Symmetrical Multi-Processing Cluster)是节点内共享内存结构和节点间分布式内存结构的混合,它综合了 MPP、COW 及 SMP 的优点,多级体系结构和混合编程模式使之成为当今高性能计算机领域的主流.

2 OpenMP 模型和 MPI 模型

并行计算编程模型主要有三种:数据并行模式、共享地址空间模式和消息传递模式.其中,MPI 和 OpenMP 是目前最流行的编程模型标准.在共享内存系统中,基于编译制导的并行计算编程的标准是 OpenMP,在分布内存系统中,基于消息传递的并行标准是 MPI、PVM 等.

2.1 OpenMP 编程模型

OpenMP 是基于共享内存并行系统的多线程程序设计的编程标准.程序员可向串行程序中插入专用的 pragma 并行化语句,再由编译器自动将程序进行并行化,并在必要之处加入同步互斥以及通信.忽略这些 pragma 时不影响源程序运行[2].OpenMP 实现的是线程级的并行,并行化设计是通过向 C,C++ 或 Fortran 源程序插入编译制导语句实现的[3].

多核处理器的发展,使得 OpenMP 的应用越发广泛.它利用共享变量进行线程间通讯,避免了进程间消息传递带来的通讯开销,并提供循环级细粒度和粗粒度的并行机制.OpenMP 的不足之处是可扩展性差,共享内存使得可计算单元受限,当问题规模增大,很难扩充资源提高加速比[4].

收稿日期:2011-10-28;修回日期:2012-05-16
基金项目:山东省研究生教育创新计划(No.SDYY12060);山东省科技攻关项目(No.2009GG10001014);山东省教育厅科研项目(No.J10LG09)

2.2 MPI 编程模型

在机群计算系统中,节点间可以采用消息传递技术,如 MPI(message passing interface)和 PVM(Parallel Virtual Machine)来进行并行程序设计.机群计算系统的优点包括高可扩展性,高可用性和高性价比.机群可用于执行计算密集的环境,如 N-body 问题,DNA 序列的实验模拟,气象预报,核模拟,高能物理等等[5].

MPI 是一个基于消息传递的并行编程模型标准,是目前分布式存储计算系统上的主流编程模型.并行语言 MPI 库可以被 FORTRAN77、C、Fortran90、C++ 调用,MPI 允许静态任务调度,显示并行提供了良好的性能和移植性,用 MPI 编写的程序可直接在 SMP 机群上运行.

MPI 显示的域分解要求整个程序由串行到并行的一次性转换,开发、调试相对困难.而共享存储模式允许在源串行代码中插入并行说明从而实现逐步转换.另外 MPI 的显示通讯开销大,为减少延迟通常采用较大代码粒度.

3 SMP 机群层次化并行模型

3.1 SMP 机群体系结构

通过提高 CPU 主频和带宽来提高计算机性能是受限的.于是,人们又期望通过增加 CPU 数目和内存容量来提高性能.如向量机、对称多处理机(SMP)等,但当 CPU 数目达到某个阈值,像 SMP 这类的多处理机系统的可扩展性(Scalability)变得很差,因为 CPU 访存的带宽并不随着 CPU 的数目的增加而有效增长.而机群系统有着很好的性能价格比和可扩展性,并且用户编程方便[6].借助采用 NVIDIA Tesla 机群技术的由国防科技大学自主研制的天河一号 A(Tianhe-1A),以 2507 万亿次的 linpack,在 2010 年 6 月发布的世界 TOP500 高性能计算机排行榜上位居榜首[7].

很多高性能计算平台(HPC)采用具有层次结构的 SMP 机群系统[8].SMP 机群节点间采用消息传递的分布式存储结构,机群节点内部采用共享存储模型.如图 1 所示.

基于 SMP 机群多层次体系结构特点,可以实现

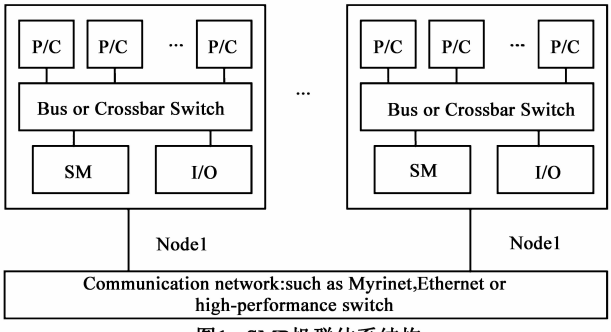

图1 SMP机群体系结构

OpenMP 与 MPI 模式的混合编程.节点内利用共享模式的 OpenMP 编程,节点间采用基于消息传递的 MPI 模型.节点间的 MPI 并行位于上层,节点内的 OpenMP 并行位于下层.首先对任务进行 MPI 分解,每个任务被分配到一个 SMP 节点,进程间 MPI 通讯;然后在每个节点进程中,使用 OpenMP 制导指令创建一组线程,并分配到 CMP 节点的不同 CPU 上并行执行.如图 2 所示.

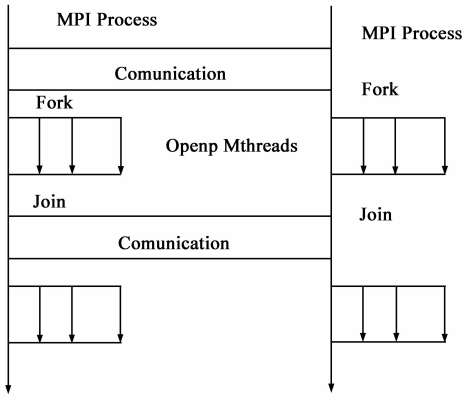

图2 MPI与OpenMP混合模型

3.2 SMP 机群 MPI 与 OpenMP 混合编程的优点

(1)改善并行代码的可扩展性

因负载均衡的原因 MPI 代码不易进行扩展,它的一些不规则的应用都存在负载不均衡问题[8].采用混合编程模型,MPI 只负责节点间的通信,实现粗粒度的并行.OpenMP 负责节点内的并行,因不存在负载均衡的问题,性能得以提高.

(2)在纯的 MPI 应用中,节点间的进程需要消息传递,而混合模型中,节点内部采用 SM 访问,减少进程数从而降低了通信开销.

(3)更充分的利用 CPU

在某些应用中,纯 MPI 模式的效率并不随 CPU 的数量增多而提高,有一个最优值.此时,采用混合编程,用 OpenMP 线程替代从而减少进程数,可充分发挥各处理器的作用.

此外,在使用 MPI 与 OpenMP 混合编程模型中,要注意不同的 MPI 进程内的线程之间进行点对点通讯时的正确发送与接收;要注意用轻量级的 OpenMP 线程代替重量级的 MPI 进程的并行的可行性,同时 OpenMP 也要产生系统开销,如线程 fork/join.

4 CMP 节点的 SMP 机群层次化并行模型

多核处理器(CMP-Chip Multi-Processor)与对称处理器(SMP)相比,内核间通讯的开销远远小于 CPU 之间的通讯开销.随着多核处理器技术的发展,很多 SMP 机群节点都采用多核处理器.如图 3 所示为一个 SMP 机群

的 CMP 节点.节点由含有若干多核处理器组成.SMP 机群节点通过高速互联网络进行通讯.

在图 3 所示结构中,为充分利用多核机群结构特点,可先将划分的任务进程映射到每个计算节点上,再将该进程所生成的每个线程映射到相应计算节点 CPU 的多核上.

在图 3 所示结构中,消息传递分为三类:片内通讯(Intra-CMP),片间通讯(Inter-CMP)和节点间通讯(Inter-Node).该结构下执行算法也有三种方式[8]:

(1)为每个处理器创建一个 MPI 进程.

(2)为每个处理器创建一个 MPI 进程,每个进程内创建线程.

(3)为每个节点创建一个 MPI 进程,并创建占用处理器的线程.

混合并行模式可分为 MPI + 细粒度 OpenMP 以及 MPI + 粗粒度 OpenMP 两种模式.

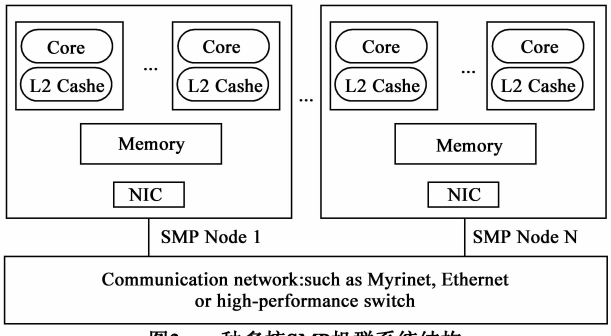

图3 一种多核SMP机群系统结构

4.1 MPI + 细粒度 OpenMP 混合并行

细粒度 OpenMP 并行指的是只并行求解循环部分的计算,亦称循环级并行.因此 MPI + 细粒度 OpenMP 混合并行较为简单,即在 MPI 进程内的循环使用 OpenMP 多线程共享内存并行[5].一个细粒度混合程序的伪代码[9]:

```
# include < mpi. h >
# include < omp. h >
…
# define NUM_ THREADS 8
/ * Some Computation and MPI Communication * /
omp_ set_ num_ threads(NUM_ THREADS);
# pragma omp parallel
# pragma omp do private( i) share( n)
do i = 1, n
/ * Computation * /
end do
/ * Some Computation and MPI Communication * /
MPI_ Finalize();
```

4.2 MPI + 粗粒度 OpenMP 混合并行

在粗粒度 OpenMP 混合并行设计中,OpenMP 采用 SPMD(single program multiple data)编程方式与 MPI 结合.首先,OpenMP 在主程序中生成多个线程,每个线程(类似 SPMD 格式中的一个进程)均执行相同的代码段,对应不同的数据域.每个线程基于并行域内的线程数和本身的 id 号,决定对进程内的哪部分进行计算.一个粗粒度混合程序的伪代码[10]:

```
…
omp_ set_ num_ threads(8);
# pragma omp parallel private(…) shared(n)
{
thread_ id = omp_ get_ thread_ num();
num_ threads = omp_ get_ num_ threads();
begin = n * thread_ id/num_ threads;
if (thread_ id = num_ threads-1)
        end = n - 1;
else
        end = n * (threads-num + 1)/num_ threads;
for( i = begin; i < = end; i + + )
        {
            a( i) = b( i) * c( i)
        }
}
…
```

在实践中发现,在 SMP 机群 MPI + OpenMP 混合编程中,选择细粒度 OpenMP,可以取得性能和易用性的兼顾.一方面大多的计算可在循环中实现,细粒度并行化只需在循环计算处使用 OpenMP 编译制导指令并行化;另一方面虽然通常粗粒度并行化的性能优于细粒度并行,但 MPI 并行计算代码几乎要全部重写,而细粒度并行化的工作量小,优化后的性能亦可以提升.

5 实验与结果分析

5.1 测试环境

实验硬件环境:曙光 TC5000 机群,以 6 个 CB65 刀片为计算节点,以一台 A620r-H 作为登陆管理节点.每个 CB65 刀片节点拥有两颗 AMD CPU,共计 8 个核,3.0Gfloads.

实验软件环境:操作系统为 SUSE Linux Enterprise Server10 SP2,OpenMP 编译器采用 OMPi-1.4 版,gcc 版本为 4.2.4,网络协议 IPv4.OpenMP 使用的是支持 OpenMP 制导语句的 GCC4.2.4,MPI 使用的是 MPICH2,在编译 MPI 与 OpenMP 程序时使用的是 GCC4.2.4 编译器,编译时需加参数-lmpich 和-fopenmp,在 MPICH2 环境中运行.

5.2 测试程序

在本文中,有关 MPI 与 OpenMP 的测试是在针对 N-body 的 OpenMP 细粒度混合算法中进行的.

N-body 问题:在一定的物理空间中,分布有一定数量的粒子,每对粒子间都存在相互作用(如万有引力,

库仑力等).给定它们的初始速度和位置,在作用力的作用下,每隔一定的时间步,要求计算出它们新的速度和新位移.这样在每个时间步的计量是 $O(N2)$.当要求时间步足够小时,计算量是巨大的,并行计算是 N-body 计算问题的必然选择[11].

N-body 算法很多,典型的树形分级算法基于如下理论:在考虑远程一组粒子对单个粒子的作用时,可将组中所有粒子作为一个整体来考虑.在这种算法里,整个物理空间被递归的分成不同层次的单元格,在三维空间里就是八叉树,二维空间中是四叉树.

SPMD 伪代码如下[12]:

```
/* mian loop */
while(time < end){
/* build the octree */
O_tree(p,t,N,my_first_node,my_last_node);
All_Gatherv (my_subtree,tree);
/* compute the forces */
forces(p,t,N,my_first_particle,my_last_particle);
/* compute the minimal time step */
delt = tstep(p,N,my_first_particle,my_last_particle);
All_Reduce(delt,new_delt);
/* update the positions and velocities of the particles */
nextv(p,N,delt,my_first_particle,my_last_particle);
All_Reduce(my_particles,allparticles);
/* update simulation time */
time = time + delt;
}
```

5.3 实验结果

一般情况下,单个处理器上可以通过启动更多的线程以提高机群计算性能.但在 MPI/OpenMP 混合程设中,SMP 节点内采用的线程数不宜过多[13].图 4 显示在曙光 TC5000 刀片机群 6 节点(双 CPU)上做的 N-body 问题在 MPI/OpenMP 混合程设中执行时间与线程数的关系.其中天体数 $N = 14000$,迭代 40 次.从图 4 可以看出,当线程数等于机群中物理 CPU 数($2 \times 6 = 12$)时,加速比最好.当节点内的线程数小于机群中物理 CPU 数目($2 \times 6 = 12$)时,SMP 的加速潜力没有发挥到最大.而

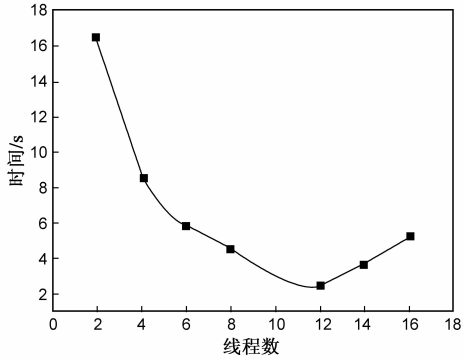

图4　MPI/OpenMP混合编程线程数与执行时间之间的关系

当线程数大于机群中物理 CPU 数($2 \times 6 = 12$)时,系统因为线程竞争总线带宽等因素而导致性能下降.

需要注意的是,在混合程设中,线程数不宜过多,因为过多线程的启动和终止的开销得不偿失[14].另外线程间彼此对 Cache 的争夺会降低性能.通常,节点上的线程数不宜超过硬件线程个数.

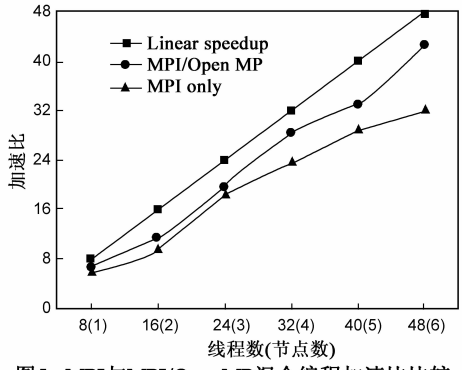

图5　MPI与MPI/OpenMP混合编程加速比比较

图 5 是在图 4 实验环境中,天体数目 $N = 1600$,迭代 20 次时,纯 MPI 与 MPI/OpenMP 程序加速比的比较图.从图中可以看出在以不同的节点数和线程数进行的同规模并行计算中的计算加速比是不同的,混合编程都要高于纯 MPI 模式.由于 MPI 模式可能的负载不均衡和细粒度问题难以扩展等原因,采用 MPI/OpenMP 混合编程模型可能大幅度提升计算性能.

6 结论

本文从 SMP 机群体系结构入手,提出了一种新的层次化并行程序设计模型.在此模型下重点研究了 MPI 与 OpenMP 混合程序设计的实现机制、并行化粒度的选择方法、进程数及线程数控制策略以及性能优化措施.并通过 N-body 的纯 MPI 和 MPI/OpenMP 混合求解的多类型实验进行了数据化对比分析.在曙光 TC5000 多核机群中的实验数据表明,在本文提出的层次化编程模型下的 MPI 与 OpenMP 混合程序设计相比于纯 MPI 编程有较大的优势.

目前,SMP 机群在并行计算机中占得的比重逐渐增大,探求多核机群的并行计算的算法、计算模型及其应用等将是未来高性能机群计算应用研究的热点.

参考文献

[1] 陈国良,吴俊敏,张锋,章隆兵.并行计算机体系结构[M].北京:高等教育出版社,2002.
CHEN Guo-liang, WU Jun-min, ZHANG Feng, ZHANG Long-bing. Parallel Computer Architecture[M]. Beijing: Higher Education Press, 2002. (in Chinese)

[2] 富弘毅,丁滟,宋伟,杨学军.一种基于扩展数据流分析的

OpenMP 程序应用级检查点机制[J].计算机学报,2010,33(10):1809 – 1822.
FU Hong-yi, DING Yan, SONG Wei, YANG Xue-jun. An application level checkpointing based on extended data flow analysis for openMP programs[J]. Chinese Journal of Computers, 2010,33(10):1809 – 1822.(in Chinese)

[3] BARBARA CHAPMAN, GABRIELE JOST, RUUD VAN DER PAS. Using OpenMP[M]. The MIT Press, 2007.

[4] Bull, J M, Enright, J P, Ameer, N. A microb-enchmark suite for mixed-mode OpenMP/MPI[A]. In: Müller, M S, de Supinski, B R, Chapman, B M(eds.) IWOMP 2009[C]. LNCS, vol. 5568, Springer, Heidelberg, 2009. 118 – 131.

[5] YANG Chao-tung, LAI Kuan-chou. A directive based MPI code generator for Linux PC clusters[J]. The Journal of Supercomputing, 2009, 50(2): 177 – 207.

[6] 刘志强,宋君强,卢凤顺,赵娟.基于线程的 MPI 通信加速器技术研究[J].计算机学报,2011,34(1):154 – 163.
LIU Zhi-qiang, SONG Jun-qiang, LU Feng-shun, ZHAO Juan. A study of thread-based MPI communication accelerator[J]. Chinese Journal of Computers, 2011, 34(1): 154 – 163.(in Chinese)

[7] 刘江."天河一号 A"摘取高性能计算机世界冠军[EB/OL]. http://cloud.csdn.net/a/20101029/281074.html, 2010-10-29.

[8] 魏伟.基于 SMP 集群的性能优化方法的研究[D].兰州:兰州大学,2006.
WEI Wei. Research on performance tuning methodology based on SMP cluster[D]. LanZhou: LanZhou University, 2006.(in Chinese)

[9] ZHANG Yuan-yuan, Hidetoshi Iwashita, Kuninori Ishii, Masanori Kaneko and et al. Hybrid parallel programming on SMP clusters using XPFortran and OpenMP[A]. M. Sato et al.(Eds.) IWOMP 2010[C]. LNCS, vol. 6132, Springer, Heidelberg, 2010. 133 – 148.

[10] 龙柏,孙广中,熊焰,陈国良.一种基于多核机群架构的混合索引结构[J].电子学报,2011,39(2):275 – 279.
LONG Bai, SUN Guang-zhong, XIONG Yan, CHEN Guo-liang. A hybrid index structure based on muti-core cluster[J]. Acta Electronica Sinica, 2011, 39(2): 275 – 279.(in Chinese)

[11] 王小伟,郭力,杨章远.N-body 算法及其并行化[J].计算机与应用化学,2003,20(2):195 – 200.
WANG Xiao-wei, GUO Li, YANG Zhang-yuan. N-body algorithms and parallezation of them[J]. Computers and Applied Chemisty, 2003, 20(2): 195 – 200.(in Chinese)

[12] Rocco Aversa, Beniamino Di Martino, Nicola Mazzocca, et al. Performance analysis of hybrid openMP/MPI N-body application[A]. B. M. Chapman (Ed): WOMPAT 2004[C]. LNCS 3349, 2005. 12 – 18.

[13] 杨际祥,谭国真,王荣生.多核软件的几个关键问题及其研究进展[J].电子学报,2010,38(9):2140 – 2145.
YANG Ji-xiang, TAN Guo-zhen, WANG Rong-sheng. Some key issues and their research progress in multicore software [J]. Acta Electronica Sinica, 2010, 30(9): 2140 – 21 45.(in Chinese)

[14] Rabenseifner, R. Hybrid Parallel Programming on HPC Platforms[A]. In: Proceedings of the Fifth European Workshop on OpenMP[C]. EWOMP 2003, Aachen, Germany, September 22 – 26, 2003. 185 – 194.

作者简介

祝永志 男,1964 年 6 月出生于吉林省榆树市.现为曲阜师范大学计算机科学学院教授,硕士生导师,中国计算机学会高级会员.主要研究方向为网络与并行计算.
E-mail: rizhaozyz@126.com

张丹丹 女,1988 年 11 月出生于山东省嘉祥县.现为曲阜师范大学计算机科学学院硕士研究生.主要研究方向为网络与并行计算.
E-mail: zhdandan2006@126.com