文章编号: 1001- 2486(2010) 05- 0062- 07

基于并行模拟的多核集群系统性能预测和分析*

徐传福, 车永刚, 王正华
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要: 针对多核集群系统所表现出的新的性能特征, 提出了面向多核集群系统消息传递应用程序的并行模拟模型并设计、实现了一个并行模拟器 MCPSim(Multi-core Cluster Parallel Simulator), MCPSim 在功能模型和性能模型上体现了片内核间、结点内片间以及结点间等三个层次上消息通信的特点, 同时支持对应用的消息数量、通信量等的百分比分布的 profiling 功能, 采用 PRIME、Jacobi3D、NPB IS 以及 HPL 等 Benchmark 程序对 MCPSim 进行了测试, 结果表明 MCPSim 性能预测的精度优于 BigSim, 同时能够广泛应用于针对多核集群系统消息传递应用程序的性能分析中。

关键词: 多核集群; 消息传递; 并行模拟; 性能预测
中图分类号: TP391 文献标识码: A

Performance Prediction and Analysis of Multi-core Cluster Systems By Parallel Simulation

XU Chuan-fu, CHE Yong-gang, WANG Zheng-hua
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A parallel simulation model for message passing applications on multi-core cluster systems was presented, then a parallel simulator MCPSim (Multi-core Cluster Parallel Simulator) was designed and implemented. MCPSim adopted a three-layer (i.e., Intra-CMP, Inter-CMP, and Inter-Node) message passing model in its functional and timing model. Furthermore, MCPSim implemented a profiling module to obtain message distribution percentage in the three layers. The current research selected several benchmark applications including PRIME, Jacobi3D, NPB IS and HPL to validate MCPSim. Results show that MCPSim is more accurate than BigSim in performance prediction and can be used in the performance analysis of message passing applications on multi-core cluster systems.

Key words: multi-core cluster; message passing; parallel simulation; performance prediction

随着多核处理器技术的发展, 当前很多集群系统采用多核处理器构建集群结点, 即所谓的多核集群。图 1 给出了一个常见的多核集群系统的构成, 结点通常包含若干具有多个计算核心的片上多处理器(Chip Multi-Processor, CMP), 结点之间通过高性能互联网络进行通信。目前, 常用于构建多核集群结点的 CMP 有 Intel 的 Xeon 和 AMD 的 Opteron 等, 常见的互联网络包括 Infiniband、Myrinet 以及高速以太网等。多核集群使得人们能够以相对更高的性价比拥有更多的计算核心, 但与此同时, 应用在多核集群系统上也表现出了与传统单核处理器集群系统不同的性能特征, 给性能分析与优化带来新的问题。例如: 文献[1]对一个典型多核集群系统 Intel Bensley 进行了测试和 profiling, 发现 HPL 等 benchmark 中平均超过 50% 的消息传输位于结点内部(Intra-Node), 从而表明针对该系统优化结点内通信与优化结点间通信对应用性能的提升同样重要; 文献[2]也通过测试和 profiling 定量分析了网络资源共享、cache 竞争以及应用进程到计算核心的不同映射等对多核集群系统上并行应用性能的影响; 文献[3]对两种典型多核体系结构 Intel Clovertown 以及 AMD Opteron 进行了评估, 根据评估结果优化指导 MPI 集合操作算法设计, 可以获得 30% 的性能提升。上述研究均以典型系统为背景, 采用了 benchmark 测试的方法。性能模拟是另外一种常用的性能评价方法, 相对于 benchmark 测试和分析模型, 模拟技术在性能评价的代价、时

* 收稿日期: 2010- 03- 04
基金项目: 国家"863"计划资助项目(2007AA01Z116); 国家自然科学基金资助项目(60603055)
作者简介: 徐传福(1980—), 男, 助理研究员, 博士生。

间以及灵活性之间有很好的平衡, 对不同应用在目标体系结构上的性能提供了一种较为通用的评估方法, 而其中并行模拟将模拟任务并行化, 利用并行宿主机平台提高模拟速度和容量, 更适合于当前大规模并行目标计算机系统的性能评估[4]。已有的并行模拟器主要包括 LAPSE[5]、BigSim[6]、MPI-SIM[7] 以及 WWT(Wisconsin Wind Tunne) 系列[8-9] 等。WWT 系列由威斯康星大学发布, 1993 年推出的 WWT Ⅰ 主要针对共享存储多处理器系统的 cache 一致性研究, 2000 年推出的 WWT Ⅱ在 WWT Ⅰ 基础上增强了可移植性。LAPSE(Large Application Parallel Simulation Environment) 是 NASA 上世纪 90 年代资助的项目, 目标是实现 Intel Paragon 平台上基于 Intel 的消息传递库 nx 开发的并行应用代码的可扩展性分析和性能分析。MPI-SIM 是加州大学 COMPASS (COMponent-based Parallel System Simulator) 项目的一部分, 通过对 MPI 通信库的模拟实现对 MPI 应用的性能预测。BigSim 来源于伊利诺伊大学 Urbana-Champaign 分校为 BlueGene/C 开发的并行模拟器, 能够预测基于 AMPI 和 Charm++ 的并行应用在 BlueGene 类型机器上的性能。上述并行模拟器主要缺点是仅能模拟早期特定的目标并行体系结构和应用, 设计和实现上难以扩展, 例如: WWT 系列和 LAPSE 均不支持标准 MPI 应用的模拟, MPI-SIM 和 BigSim 虽然能从一定程度上支持基于单核处理器结点的集群系统的性能模拟, 但无法实现对新型多核集群的性能预测和分析。有必要从性能模型和功能模型上对当前的并行性能模拟器进行扩展, 以适应对新型多核集群系统的性能预测和分析。

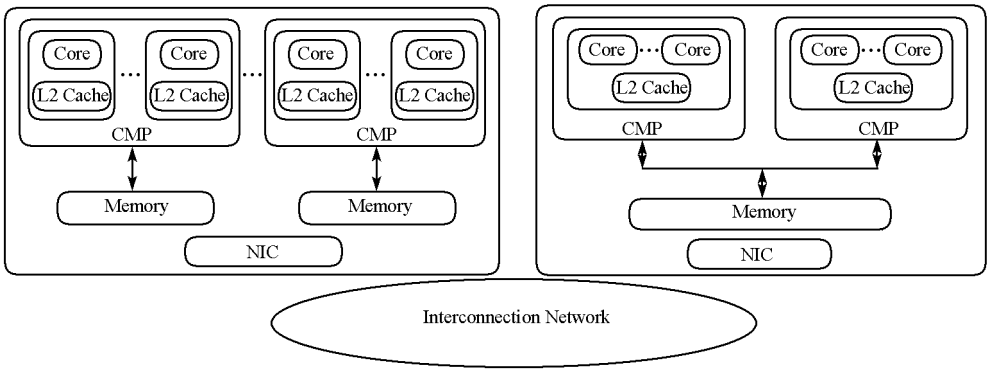

图 1 典型的多核集群系统构成

Fig. 1 The components of a typical multi-core cluster system

1 MCPSim 模拟模型

1.1 多核集群系统上消息传递应用的并行模拟模型

用户可将 MCPSim 看作一个虚拟的多核集群系统并行平台, 对任意给定参数和配置的 MPI 应用和多核集群系统组合, MCPSim 可给出其功能执行结果和预测性能。对于本地执行代码块(Local Execution Code Block, LECB), MCPSim 以直接执行方式在宿主机上实现其功能模拟, 通过相应的性能模型预测其在目标机上的执行时间; 对于 MPI 消息通信代码, MCPSim 将其替换为对应的模拟器本身提供的 MPI 接口函数, 接口函数不仅需要模拟应用进程之间的 MPI 消息通信功能, 同时需要给出消息在目标通信网络中传递时的性能预测。MCPSim 为每个目标进程维护一个本地虚拟时钟, 其时间戳随模拟过程推进, 各目标进程间采用同步策略, 保证消息通信顺序正确性和时间戳更新的精度(详细的时间戳更新过程见 1.4 节)。由于目标系统规模通常远大于宿主并行机, 通常一个模拟进程需要执行多个目标进程, MCPSim 以块分配的方式将目标进程映射到模拟进程, 并保证同一目标多核结点内的目标进程映射到同一模拟进程以提高效率。整个并行模拟模型如图 2 所示。

1.2 功能模型

由于 LECB 采用直接执行方式模拟, 这里的功能模型主要考虑消息在多核集群系统结点内部和结点间互联网络的传递。MCPSim 的多核结点功能模型如图 3 所示, 其中每个结点内包含若干 CMP, 每个 CMP 有若干计算核心, 每个计算核心有一个私有消息队列用于存放待处理的消息, 每个结点包括一个

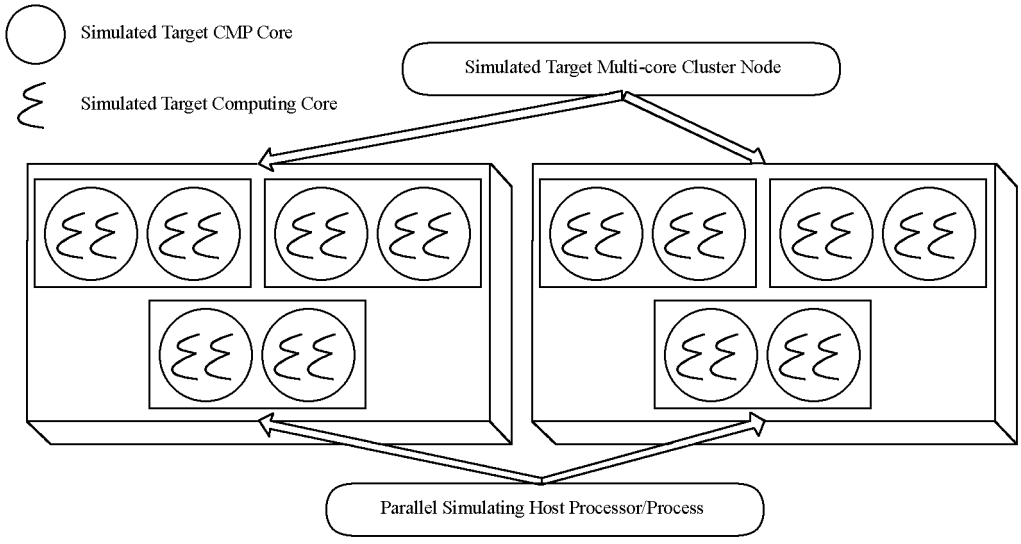

图 2 MCPSim 并行模拟模型

Fig. 2 Overview of parallel simulation model for MCPSim

全局消息队列用于存放结点内全部计算核心可处理的消息,另外,消息缓冲区 Inbuffer 和 Outbuffer 用于结点内部与互连网络的消息交互。MCPSim 结点功能模型将多核集群系统中的消息通信分为三个层次:(1) 片内消息(Intra-CMP):即位于同一 CMP 之内不同计算核心之上的 MPI 进程之间的通信消息;(2) 片间消息(Inter-CMP):即位于同一结点内的不同 CMP 之内的计算核心之上的 MPI 进程之间的通信消息;(3) 结点间消息(Inter-Node):即位于不同结点内的计算核心之上的 MPI 进程之间的通信消息。

MCPSim 利用宿主机网络环境实现结点间互联网络通信功能的模拟。整个目标多核集群系统可以看作由若干功能结点通过消息网络连接而成,后面用三元组(N, C, T) 表示其构成,其中 N 为结点的数量,C 为每个结点内的 CMP 数,T 为每个 CMP 包含的硬件线程或计算核心数,用户可通过三元组(N, C, T) 对 MCPSim 进行配置。

1.3 性能模型

性能模型决定了 MCPSim 如何估计目标程序在目标系统上的执行性能,根据 1.1 节的并行模拟模型,MCPSim 需要考虑 LECB 执行时间和 MPI 消息传输时间的预测。

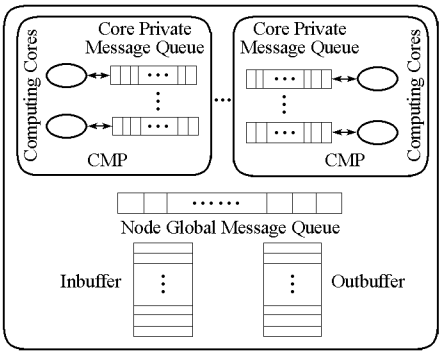

图 3 MCPSim 结点功能模型

Fig. 3 Functional model for MCPSim Node

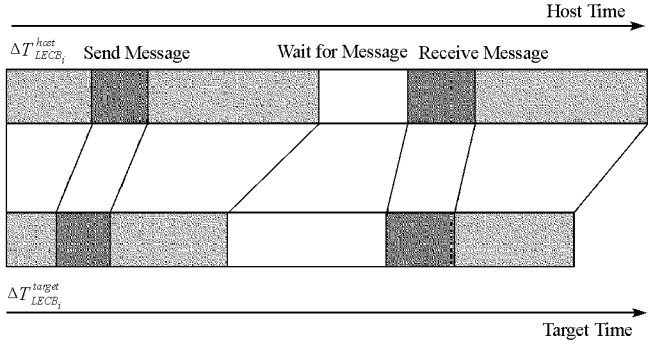

图 4 LECB 执行时间预测

Fig. 4 Prediction of LECB execution time

(1) LECB 执行时间预测

与其他直接执行驱动模拟器类似,MCPSim 以 LECB 的宿主机墙上时间乘以扩展因子(scalefactor) 的方式作为预测该 LECB 在目标系统上的执行时间。设对第 i 个 $LECB_i$,其宿主机墙上时间为 $\Delta T^{host}_{LECB_i} = T^{end}_{LECB_i} - T^{begin}_{LECB_i}$,扩展因子为 scalefactor,则其在目标系统上的执行时间为

$$\Delta T^{target}_{LECB_i} = \Delta T^{host}_{LECB_i} \times scalefactor \quad (1)$$

其中 scalefactor 反应了宿主机与目标系统本地处理部件(如 CPU、Cache 等) 之间的"能力"之比, 实际应用中通常根据两者的 CPU 主频、浮点指令流出速率等体系结构指标确定。图 4 给出了如何采用上述方法将目标程序 LECB 在宿主机上的墙钟时间映射为目标机器上的性能。

(2) 消息传输时间预测

在性能分析模型或并行模拟中, 通常采用延迟/带宽(Latency/Bandwidth, L/B) 模型对 MPI 消息的传递时间进行估计[8], 例如, BigSim 针对以太网互联采用了如式(2) 所示的 L/B 模型:

$$t = l + s/b \quad (2)$$

其中 l 指网络延迟, b 为网络带宽, s 是消息大小。多数并行模拟器只支持 L/B 模型, 尽管 L/B 模型忽略了网络竞争, 但仍然取得了较好的性能预测效果, 目前已知仅 BigSim 提供独立运行的基于竞争模型的详细网络性能模拟器。文献[2] 和[3] 的测试结果表明, 多核集群系统中的不同大小的消息通信性能会由于通信层次、共享资源竞争等差别较大, 具体表现在消息通信带宽、延迟等方面的区别, 这些均会直接影响消息传递程序(尤其是通信密集型程序) 的整体性能。根据上述特点, MCPSim 对 L/B 模型进行了改进, 提出了基于区间(interval)、层次(hierarchy) 和竞争(contention) 的 L/B 模型:

$$t = l(hierarchy, contention, interval) + s/b(hierarchy, contention, interval) \quad (3)$$

改进后的 L/B 模型的延迟、带宽参数是通信层次、消息大小区间、共享资源竞争等的函数。(3) 式可以有多种具体实现, MCPSim 采用了类似文献[11] 的方式, 如图 5 所示, 模拟器代码根据消息大小、目的地等目标进程运行时信息自适应确定延迟、带宽, 而相关的目标系统网络参数可以由用户通过配置文件的方式提供给 MCPSim。

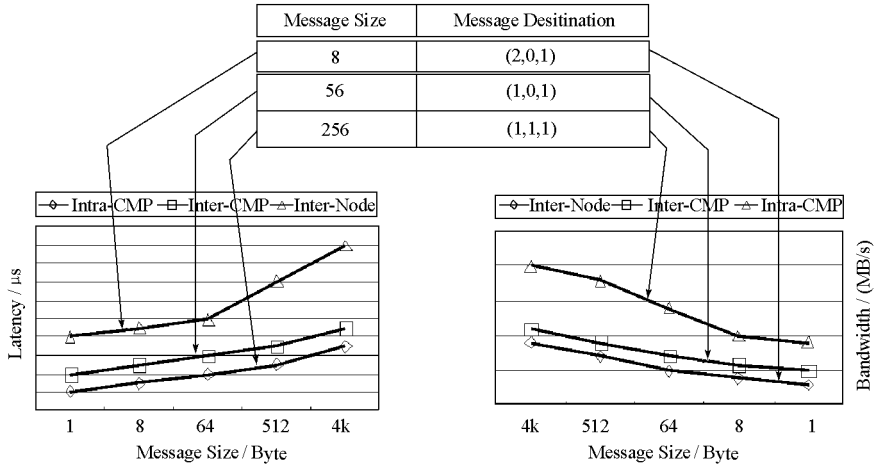

图 5 MCPSim 中改进的 L/B 模型

Fig. 5 Improved L/B model in MCPSim

1.4 时间戳更新过程

基于上述性能模型, 对任一目标进程 A, 设当前时间戳为 $T^A_{current}$, 则并行模拟过程中时间戳更新过程如图 6。

2 MCPSim 实现

基于 BigSim 框架实现了 MCPSim 原型系统。BigSim 的软件结构如图 7 所示, 其中 BlueGene 机器功能模型、同步引擎和性能模型构成了整个并行模拟器的核心层。核心层之下通过 Converse 通信库结合宿主机通信网络实现消息通信功能。核心层之上通过 AMPI(Adaptive MPI) 和 Charm++ 并行模拟接口实现对两种并行应用程序的模拟, 其中 Charm++ 是一种消息驱动的面向对象并行编程语言, AMPI 是对标准 MPI 的扩展, 基于 Charm++ 采用可迁移的用户级线程实现, 以更好地支持动态自适应负载均衡。

```
T^A_current = 0  // 首先将当前时间戳初始化为0
While (true)  // 一直处理目标进程A中的代码区域(Code Region, CR)直到结束
{  If CR is a LECB// 若为LECB
        {  T_host = GetWalltime( )  // 记录当前宿主机墙上时间
        Execute CR on host machine  // 在宿主机上直接执行LECB
        T^A_current = scalefactor× (GetWalltime( ) - T_host)  // 根据(1)式更新当前时间戳}
  If CR is a MPI _send  // 若为MPI发送消息语句
{  T^msg_predictal = T^A_current + Δt  // 根据(3)式预测消息msg所需的传输时间Δt
    Call MCPSim version of MPI _send  // 调用MCPSim版本的MPI _send并将T^msg_predicted附在消息头部}
  If CR is a MPI _recv  // 若为MPI接收消息语句
{  Shedule and process matched message msg′  // 根据同步策略调度与接收语句匹配的消息
    T^A_current = max{T^msg′_predictal, T^A_current}  // 将当前时间戳设置为msg′预测接收时间和当前时间戳两者的最大值}}
```

图6 MCPSim中目标进程的时间戳更新

Fig. 6 Timestamp updating for target processes in MCPSim

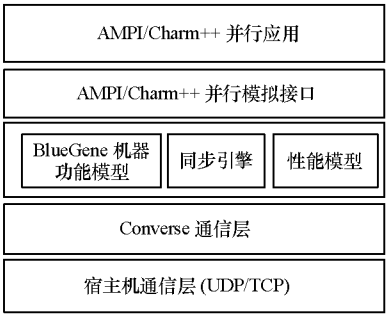

图7 BigSim软件结构

Fig. 7 The software architecture of BigSim

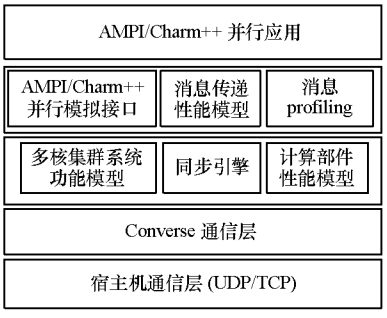

图8 MCPSim软件结构

Fig. 8 The software architecture of MCPSim

BigSim所模拟的BlueGene结点从功能模型和性能模型上均无CMP的概念,无法直接支持多核集群结点;此外,BigSim的消息传输时间等性能模型在核心层实现,在这一层不仅MPI消息被重新打包为BlueGene消息,增加了消息大小,同时有很多支持Charm++运行时环境的消息也被计入最终性能预测结果,因而导致BigSim最终给出的是针对AMPI和Charm++应用在BlueGene上的性能预测。为实现预测标准MPI应用在多核集群上的性能,MCPSim需要在AMPI并行模拟接口层实现(3)式定义的消息通信性能模型,同时将BlueGene机器功能层修改为图3所示的多核集群结点功能结构,最终实现的MCPSim软件层次如图8。在图8中,MCPSim还增加了一个profiling模块用于统计特定大小区间内消息数量和总数据量在多核集群系统三个通信层次上百分比分布,这一信息可有助于分析应用对多核集群各层次通信信道的利用情况[2]。

3 测试

3.1 测试环境

以一个千兆以太网多核集群系统作为宿主机和目标系统对MCPSim进行了测试,该系统每个结点包括2个4核的Intel Xeon,主频2.33GHz。测试采用的benchmark程序为标准C语言MPI程序:PRIME、Jacobi 3D、NPB IS和HPL。PRIME是一个求主元的程序,整个任务在各目标进程间均匀分布。Jacobi 3D程序是一个采用三维任务划分的7点格式计算程序,每个进程需要与其6个邻居进程进行通信以交换邻接面数据,每次Jacobi松弛计算后,通过组归约操作计算最大残差。NPB IS是NPB(NAS Parallel Benchmark)2.0中基于桶排序的并行程序,其中包括很多全体互换通信(MPI _ALLTOALL)。HPL是Linpack的标准并行版本,通过稠密线性代数方程组求解评价高性能计算机系统的浮点性能,目前广泛

用于Top 500 排行榜等测试中。

整个测试包括三部分: 首先通过比较PRIME 和 Jacobi 3D 在真实系统上测得的运行时间与MCPSim给出的预测时间来验证MCPSim 的精度, 这里也将与 BigSim 给出的预测时间进行比较, 但由于 BigSim 仅支持两层结构, 因此只能采用配置(N, T') 近似(N, C, T), 其中 $T' = C \times T$ 表示一个 BigSim 结点内计算核心的数量; 随后演示了一个基于MCPSim 进行性能分析的例子, 研究不同(N, C, T) 配置下NPB IS 程序的性能变化; 最后, 基于 MCPSim 的 profiling 模块给出了 HPL 程序三层消息统计信息。

3.2 结果分析

图9 和图 10 分别给出了固定问题规模(迭代次数) 时各种配置(N, C, T) 下MCPSim、BigSim 给出的PRIME 和 Jacobi 3D 的预测性能和实际执行时间的比较。总体而言, MCPSim 的预测精度均略优于BigSim, 尤其是对于通信相对密集型的 Jacobi 3D 程序, MCPSim 将 BigSim 的预测误差由最多的 10.5% 减少到 6%, 而对于PRIME 程序, MCPSim 和 BigSim 的预测精度相差不大, 主要是因为 PRIME 程序本身是一个相对计算密集型的程序, 只是在各进程计算结束时才通过 MPI _ ALLREDUCE 操作对结果进行归约。可见改进后的 L/B 模型更能精确反映通信密集型应用在多核集群系统上的不同层次上的消息通信性能, 进而从整体上提高并行性能模拟器的预测精度。

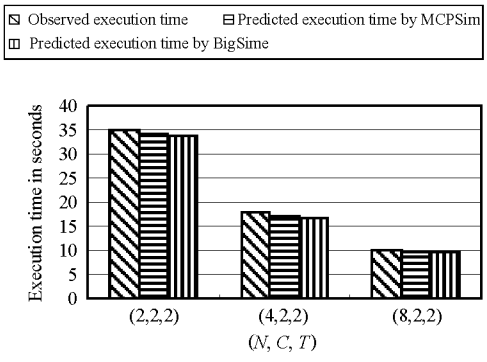

图 9 PRIME 程序预测精度验证
Fig. 9 Validation for PRIME

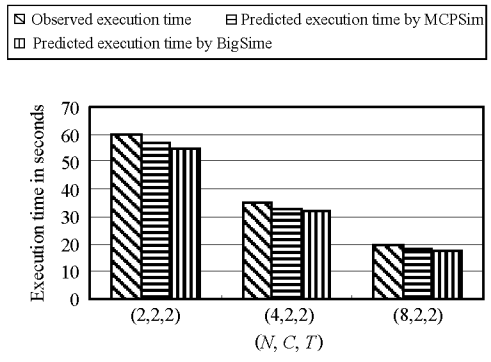

图 10 Jacobi3D 程序预测精度
Fig. 10 Validation for Jacobi3D

基于MCPSim 用户可以方便地对应用在不同系统配置下的性能进行分析研究。例如, 图 11 给出了基于MCPSim 预测NPB IS 程序在相同计算核心(16) 但不同(N, C, T) 配置时的性能变化情况, 可见尽管计算核心相同, 但NPB IS 的性能在(N, C, T) = (4, 2, 2) 或(2, 4, 2) 时相对较高。MCPSim 支持的系统配置不仅包括结点、处理器、计算核心的构成和数量等, 用户同时可以通过调整互联网络的带宽、延迟等参数研究并行应用的可扩展性, 分析其性能瓶颈。

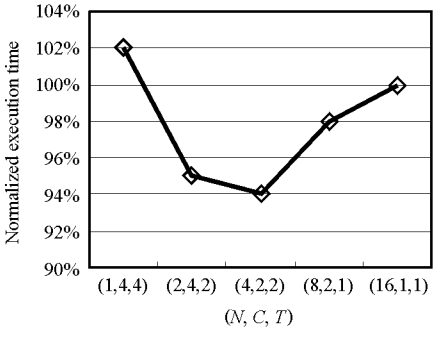

图 11 不同配置下 NPB IS 的预测性能
Fig. 11 Performance of NPB IS under different configuration

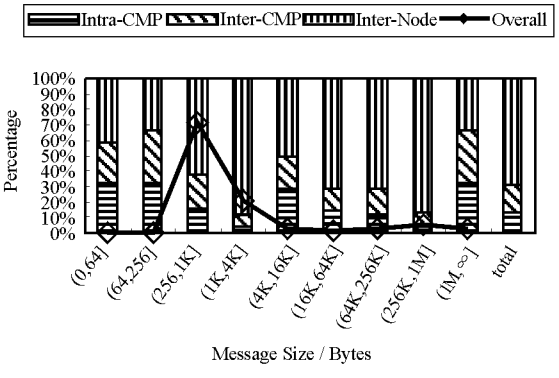

图 12 HPL 消息大小百分比分布
Fig. 12 Message number distribution for HPL

图12 和图 13 给出了基于MCPSim 获得的HPL 程序的消息通信 profiling 结果, 其中目标机器配置为(N, C, T) = (4, 2, 2), HPL 程序设置为维数 4096, 块划分为 64。这里按照大小区间(m, n] 的形式给出了

MPI 消息在三个通信层次上的百分比, 折线给出了各种区间的消息占总消息数的百分比, total 给出了三种消息各占总数的百分比。其中图 12 给出的是消息数量, 图 13 给出的是相应的消息数据量。由图 12 可见, 此时 HPL 中的多数消息为 256 字节到 4K 字节之间的小消息, 但就消息通信量而言, 16K 字节到 256K 字节之间的大消息相对较多。结合 MCPSim 的性能预测, 消息 profiling 功能可进一步帮助研究人员评估针对特定系统的消息传递库和应用的优化可能取得的效果。

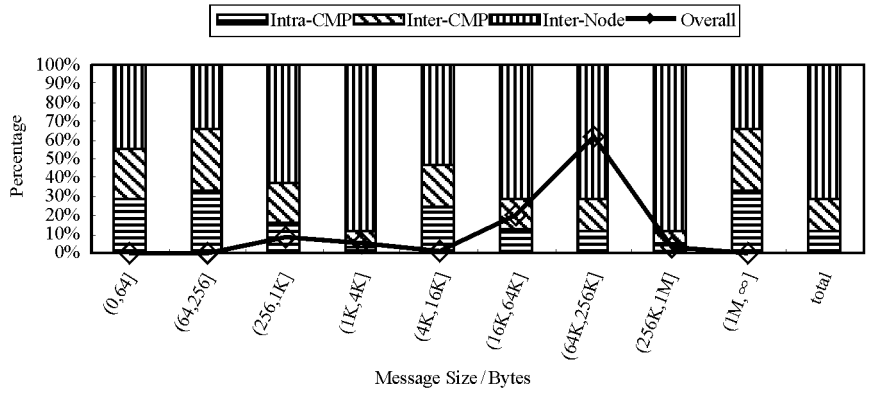

图 13 HPL 消息量百分比分布
Fig. 13 Message volume distribution for HPL

4 结束语

随着基于多核结点构建的集群系统的逐渐流行, 有必要对其表现出的一些新的性能特征进行深入分析。本文设计和实现了一个面向多核集群系统上消息传递应用程序的并行性能模拟器 MCPSim, 测试结果表明 MCPSim 性能预测精度高、灵活, 可配置、支持多核集群特有的三层消息通信的 profiling, 可广泛应用于并行应用的性能预测和分析中。

参 考 文 献:

[1] Chai L, Gao Q, Panda D K. Understanding the Impact of Multi-core Architecture in Cluster Computing: A Case Study with Intel Dual-core System [C]// Cluster Computing and the Grid, 2007: 471- 478.

[2] Narayanaswamy G, Balaji P, Feng W. Impact Of Network Sharing In Multi-core Architectures [R]. Technical Report TR- 08- 06.

[3] Mamidala A R, Debraj De R K, Panda D K. MPI Collectives on Modern Multicore Clusters: Performance Optimizations and Communication Characteristics [C]// CCGRID 2008.

[4] Hlavacs H, Ueberhuber C W. Performance Evaluation by Simulation [R]. AURORA TR2001- 15.

[5] Dickens P M, Heidelberge P, Nicol D M. A Distributed Memory Lapse: Parallel Simulation of Message-passing Programs [A]. SIGSIM Simul. Dig. [J], 1994, 24(1): 32- 38.

[6] Zheng G B, Kakulapati G, Kal' e L V. BigSim: A Parallel Simulator for Performance Prediction of Extremely Large Parallel Machines [C]// 18th International Parallel and Distributed Processing Symposium (IPDPS), Santa Fe, New Mexico, April 2004.

[7] Prakash S, Bagrodia R L. Mpi-sim: Using Parallel Simulation to Evaluate Mpi Programs[C]// Proceedings of IEEE Winter Simulation Conference, 1998.

[8] Mukherjee S S, Reinhardt S K, Falsafi B, et al. Wisconsin Wind Tunnel Ⅱ: A Fast, Portable Parallel Architecture Simulator [C]// IEEE Concurrency 2000: 12- 20.

[9] Reinhardt S K, Hill M D, et al. The Wisconsin Wind Tunnel: Virtual Prototyping of Parallel Computers [C]// Proceedings of the 1993 ACM SIGMETRICS Conference.

[10] Simon J, Wierum J M. Accurate Performance Prediction for Massively Parallel Systems and Its Applications [C]// Proceedings of European Conference on Parallel Processing EURO-PAR 1996: 675- 88.

[11] Snavely A, Carrington L, et al. A Framework for Performance Modeling and Prediction[C]// Proc of ACM/IEEE SC' 02. 2002.