多核机群下基于神经网络的MPI运行时参数优化

王 洁[1,2,4] 曾 宇[3] 张建林[1]
(首都师范大学信息工程学院 北京 100048)[1] (中国科学院计算技术研究所 北京 100090)[2]
(北京计算中心 北京 100005)[3]
(奥地利因斯布鲁克大学分布式与并行计算研究实验室 因斯布鲁克 6020)[4]

摘 要 多核处理器的新特性给MPI应用带来了新的优化空间,其中调优MPI运行时参数被证明是优化MPI应用的有效方法。然而最优的运行时参数不仅与多核机群的体系结构有关,也决定于MPI应用的程序特征。提出并分析了一种在给定多核机群下基于人工神经网络的优化模型,用于自动为未知的MPI程序预测接近最优的运行时参数。两个不同基准的实验证明了本方法的有效性。实验证明,基于本方法得到的运行时参数所产生的加速比平均达到了实际最大加速比的95%以上。

关键词 多核机群,MPI,运行时参数优化,神经网络
中图法分类号 TP393.09 **文献标识码** A

MPI Runtime Parameters Tuning Based on Neural Network on Multi-core Clusters

WANG Jie[1,2,4] ZENG Yu[3] ZHANG Jian-lin[1]
(College of Information Engineering, Capital Normal University, Beijing 100048, China)[1]
(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100090, China)[2]
(Beijing Computing Center, Beijing 100005, China)[3]
(University of Innsbruck Distributed and Parallel Systems Group, Innsbruck 6020, Austria)[4]

Abstract The new features of multi-core add the optimization space for MPI applications, and besides tuning MPI runtime parameters is a common practice perceived to optimize the MPI application performance. However, the best configuration of the runtime parameters not only depends on the underlying architecture of a specific multi-core cluster but also on the features of MPI application. We constructed and analyzed an effective tuning model bases on artificial neural network to automatically predict the near-optimal configuration of runtime parameters for any unseen input programs under the current multi-core cluster. Experimental results from two different benchmarks were presented to show effectiveness of our approach. We observed that the speedup gained by the predicted runtime parameters can averagely achieve 95% of the speedup gained by the best parameters configuration.

Keywords Multi-core clusters, MPI, Runtime parameters tuning, Neural network

1 引言

多核技术指将两个或多个处理内核集成到一个处理器芯片当中,并通过将负载分配到多核上来加速应用的处理性能。随着多核技术以及现代网络技术的发展,越来越多的机群采用多核处理器作为核心部件,基于多核技术的机群已经成为高性能计算领域的主流平台[1]。截至2009年6月,排名世界Top500的超级计算机中,约87%采用了Intel和AMD的多核芯片,并且约82%的超级计算机采用了机群结构[2]。消息传递接口(MPI, Message Passing Interface)是机群下最常用的并行编程模型,广泛应用于分布式以及共享内存系统。随着多核技术更加广泛地应用于机群,多核机群下MPI应用的性能优化成为了研究的热点。

目前主流的MPI库实现(Open MPI, MPICH等)提供了可调的运行时参数机制,允许用户根据特定的应用需求、硬件以及操作系统来调优运行时参数,以提升MPI应用的性能。例如,可以根据通讯消息的大小来修改点到点通讯采用的协议,即修改MPI库中由立即通讯协议(Eager)转为集中通讯协议(Rendezvous)的阈值参数。可调的运行时参数对多核机群下MPI应用的性能有着重要的影响,但最优的运行时参数极大程度上依赖于多核机群的存储层次(包括节点内二级或三级缓存的共享方式等)、机群的网络互联方式(包括Infiniband网络、千兆以太网和Myrinet网络等)、机群的通讯性能(包括内存和网络的通讯延迟与带宽)、机群内MPI应用的通讯层次(包括Chip内、Chip间以及节点内通讯)等因素。

图1显示了在多核机群下5个运行时参数的不同配置组

到稿日期:2009-07-14 返修日期:2009-10-23 本文受奥地利蒂罗尔州未来基金会基金(P7030-015-024)资助。
王 洁(1977—),女,博士生,讲师,主要研究方向为并行计算、机器学习、数据挖掘等,E-mail: wangjie@ncic.ac.cn;曾 宇(1973—),男,博士,高工,主要研究方向为高性能计算机、体系结构等;张建林(1966—),男,硕士,副教授,主要研究方向为数据挖掘、信息管理等。

合对 NAS 并行基准套件中 IS 基准(Class B)的性能影响。在 Infiniband 互联的 AMD 双核 10 节点的机群下,最佳的运行时参数配置与 Open MPI 库默认设置相比可以带来最多约 15%的性能提升,而错误的配置与默认配置相比可造成约 30%的性能损失。

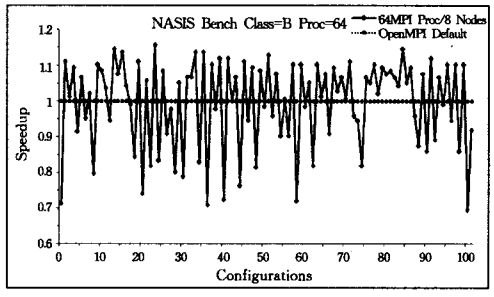

图 1　运行时参数对 IS 基准(Class B)的性能影响

如图 1 所示,可调的运行时参数可以对 MPI 应用带来可观的性能提升,但同时运行时参数的配置集合以及相应的优化空间相当庞大,难以手工实现。以主流的基于模块组件框架结构的 Open MPI 为例,假设从字节传输 btl 框架与收集操作 coll 框架中各取一个可调的数值型参数及一个标志型参数,每个数值型参数测试 20 种取值,每个标志型参数有 2 种取值,则使用自动迭代技术需要测试 4 个参数所构成的 1600 种运行时参数的组合配置。以每种配置下 MPI 程序平均执行时间为 5min 计算,共需要 5.5 天时间来找到最佳的运行时参数序列。因此迫切需要一种快速自动的参数优化方法来提升多核机群下 MPI 应用的性能。

本文设计实现了一种通用的多核机群下 MPI 运行时参数优化模型,能自动为给定软、硬件结构的多核机群下的 MPI 程序预测接近最优的运行时参数组合。本文提出的预测模型基于机器学习中的人工神经网络方法,通过对预测模型的离线训练和在线学习,能自动为未知的 MPI 应用预测接近最优的运行时参数。要预测的 MPI 程序由对源码运行一次得到的动态特征和通讯器大小等静态特征来共同描述。

本文提出的基于人工神经网络的 MPI 运行时参数优化方法在基于 InfiniBand 的多核 SMP 机群上进行验证,并运用 Open MPI 这一主流的 MPI 库作为调优 MPI 运行时参数的环境。通过 NAS 并行基准套件 2.4 中的 IS 和 LU 基准的实验证明,与 Open MPI 默认配置相比,基于人工神经网络预测模型得到的优化运行时参数组合能为多核机群下的 MPI 应用带来最多约 20%的性能提升。

2　相关研究

MPI 应用的参数调优以及运用机器学习方法来预测和优化并行应用近年来被学术界广泛研究。文献[3]中设计并开发了工具 OTPO 来优化 Open MPI 的运行时参数。OTPO 对 Open MPI 通信模式以及性能度量的可调参数的大量组合进行系统测试,以选择给定平台下指定基准的最优参数。OTPO 执行一次完整的测试只服务于一个基准程序,不像本文提出的方法可以服务于任意未知的 MPI 程序。文献[4]中建立了基于机器学习的运行时参数优化模型,但所做工作限于单个的多核节点,未对方法的可扩展性进行讨论。本文将对此模型进行扩展,研究在多核机群下运行时参数的调优方法。文献[5,6]中,使用机器学习的方法来预测并行应用的性能,但未给出优化的建议。文献[7]使用决策树来为 MPI 应用选择接近最优的收集算法。本文所提出的方法可以运用到兼有点到点通讯以及收集通讯的 MPI 应用的性能优化中。

3　基于神经网络的 MPI 运行时参数优化

本文的目的是基于人工神经网络建立参数优化模型。模型可以预测给定多核机群平台下任意未知的 MPI 输入程序的最佳运行时参数。具体方式是:首先在目标多核机群上使用不同的运行时参数配置运行训练基准,产生训练数据,用产生的训练数据对构造的优化模型进行训练,然后抽取给定的 MPI 程序的程序特性作为优化模型输入,最后模型输出接近最优的运行时参数预测值,以获得接近最大的性能加速比。优化模型的公式形式可表示为:设 f 是训练后的优化模型,$X_i=\langle x_i^1, x_i^2, \cdots, x_i^j\rangle$ 代表抽取输入的 MPI 程序的程序特征,则 $P_i=f(X_i)$ 所得向量 $P=\langle p_i^1, p_i^2, \cdots, p_i^k\rangle$ 是此程序的最佳运行时参数组合。

本文方法包括 3 个阶段:要调优的运行时参数选择、模型训练与运用已训练模型进行参数优化。图 2 描述了主要的模型训练与参数优化步骤。标准的机器学习技术——人工神经网络被用来构建优化模型,NAS 并行基准套件 2.4 的 IS 基准和 LU 基准被用来评估构造模型优化 MPI 运行时参数的准确度。

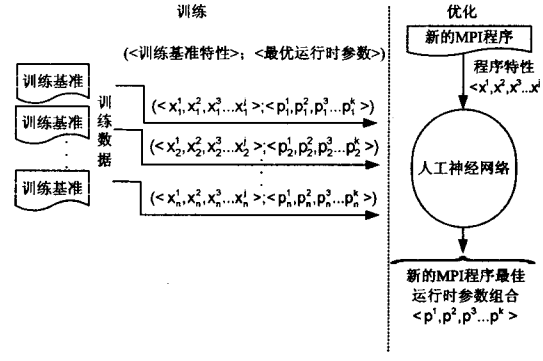

图 2　模型训练与参数优化

3.1　运行时参数选择

本文方法是在目前主流的 MPI 实现库 Open MPI 上进行验证的。Open MPI 先进的模块组件结构(MCA,Modular Component Architecture)由一系列框架、组件和模块在运行时动态组合。每个框架为专一任务特定设计,一个框架接口可以由多个模块来实现,例如 btl(Byte Transfer Layer)框架就包括对 TCP,InfiniBand 和共享内存等通讯方式的多个模块的支持。由于每个组件都定义了一系列可调的运行时参数,因此共有上百个运行时参数可能会对 MPI 应用的性能有影响。

对所有运行时参数进行调优将带来巨大的时间和资源开销,因此调优工作的第一个步骤就是对参数进行评估,选择对性能有重要影响的参数来进行分析和建模。我们使用皮尔森相关系数(PMCC,Pearson Product Moment Correlation Coefficient)来度量各个运行时参数与 MPI 应用执行时间之间的相关度。皮尔森相关系数是常用的分析变量 X 和 Y 之间线性依赖关系的方法[8],典型的皮尔森相关系数定义为

$$\rho_{X,Y}=\frac{\mathrm{cov}(X,Y)}{\sigma_X\sigma_Y}=\frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X\sigma_Y}$$

式中，E 代表数学期望，cov 代表协方差，其中 $\mu_X=E(X)$，$\sigma_X^2=E[(X-E(X))^2]=E(X^2)-E^2(X)$ 且 $\mu_Y=E(Y)$，$\sigma_Y^2=E[(Y-E(Y))^2]=E(Y^2)-E^2(Y)$。

本文实验平台为 InfiniBand 互联的多核机群，因此具体实现中首先选取了 9 个与节点内共享内存通讯以及节点间 InfiniBand 互联方式通讯性能有关的运行时参数在训练基准上进行实验，然后计算各个参数与执行时间之间的皮尔森相关系数 PMCC。相关系数绝对值大于指定阈值（本文定为 0.1）的参数用来进行实验和建模。各个运行时参数与执行时间之间的皮尔森相关系数 PMCC 计算结果如图 3 所示。

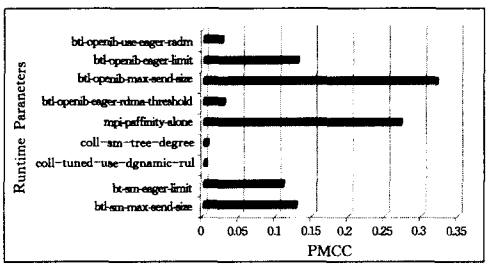

图 3　运行时参数与执行时间之间的皮尔森相关系数 PMCC

表 1 列出了相关系数大于阈值的 5 个参数及其含义。

表 1　PMCC 选择的运行时参数及其含义

参数名称	参数含义
btl_sm_eager_limit	共享内存方式下，指定 MPI 库通讯（发送和接收操作）由立即通讯协议改为集中通讯协议的阈值（以字节表示）
btl_sm_max_send_size	共享内存方式下使用流协议时，以字节表示的长消息分段的最大长度（必须大于 1）
btl_openib_eager_limit	节点间通过 InfiniBand 通讯时指定 MPI 库通讯（发送和接收）立即通讯协议改为集中通讯协议的阈值（以字节表示）
btl_openib_max_send_size	节点间通过 InfiniBand 通讯并使用流协议时，以字节表示的长消息分段的最大长度（必须大于 1）
mpi_paffinity_alone	指定是否将 MPI 进程绑定到特定的处理器（1 为绑定，0 为不绑定）

3.2　模型训练

3.2.1　MPI 程序特征抽取

本文所设计的方法是用离线训练的优化模型对未知的 MPI 应用预测最佳运行时参数，因此要从未知的 MPI 应用中抽取适合的程序特征作为优化模型输入，用来得到准确的预测结果。由于运行时参数主要影响 MPI 进程间的通讯性能，在本文中 MPI 应用的程序特性主要考虑通讯模式、通讯交换的数据量以及通讯器的大小。表 2 说明了 MPI 程序的特性，这些必要的程序特性可以通过对要预测的 MPI 程序的一次运行得到。

表 2　MPI 程序特性及描述

特性名称	特性描述
点到点通讯的时间比例	点到点通讯在程序所有通讯操作中所占的时间比例
点到点通讯的数据量	所有进程间点到点通讯所交换的平均数据量
收集通讯的时间比例	收集通讯在程序所有通讯操作中所占的时间比例
收集通讯的数据量	所有进程间收集通讯所交换的平均数据量
通讯器的大小	MPI 应用的通讯器大小

3.2.2　训练基准构造

为了产生训练预测模型的数据，本文设计了训练基准程序，在目标体系结构的多核机群上对训练基准使用可调运行时参数的多种组合来产生训练数据。同时，训练基准可接受多个输入参数来控制训练基准中点到点、收集操作传输的数据量以及通讯器大小。

根据表 2 定义的 MPI 程序特征，本文设计了训练基准，基准主要包括以下两种 MPI 通讯方式：同步的 MPI 点到点通讯和 MPI 收集操作。训练基准接收 5 个参数，可以分别用来控制训练基准中点到点通讯的比例、收集通讯的比例、两个 MPI 进程同步点到点通讯的消息的大小、收集操作中交换的消息大小以及通讯器的大小。

3.2.3　训练数据产生

通过变换训练基准的 5 个输入参数，控制点到点通讯和收集通讯的比例分别为 100%的点到点通讯、100%的收集通讯、50%的点到点通讯及 50%的收集通讯。在 3 种不同通讯比例下，分别变换点到点和收集通讯中消息大小以及 MPI 通讯器的大小，并变换运行时参数的配置组合，共产生训练数据 3000 条，用来训练神经网络优化模型。

3.3　构造预测模型进行预测

人工神经网络（ANN，Artificial Neural Network）是一类机器学习模型，可以映射一组输入参数到一组目标输出。本文采用 ANN 是由于它能很好地应用于线性和非线性回归问题，并有很好的抗噪性。

一个三层的前向型误差反传神经网络被用来构建优化模型。实验验证，对本文优化问题性能最佳的 ANN 设计为：隐藏层的传输函数为正切（Sigmoid）函数：$f(n)=\frac{2}{(1+e^{-2n})}-1$，输出层的传输函数为对数正切函数（Logarithmic sigmoid）：$f(n)=\frac{1}{1+e^{-n}}$，同时隐藏层有 10 个神经元，并且隐藏层的训练函数采用麦夸特（Levenberg-Marquardt）算法，因为它很好地结合了牛顿算法的速度与梯度下降算法的稳定性[9]。

选择训练基准中产生最高加速比的数据来训练基于 ANN 的参数优化模型，即 ANN 模型的训练数据为 $\{X_i, P_{i\_best}\}$，其中 X_i 为训练基准的程序特性，P_{i_best} 为当前程序特征下的最佳运行时参数组合。对应前文所描述的公式表示形式，设 f_{ann} 是训练后的 ANN 模型，则 $P_{best}=f_{ann}(X)$，其中 X 代表输入的 MPI 程序的程序特征向量，P_{best} 是神经网络为输入的 MPI 程序预测输出的最佳运行时参数组合向量。

4　实验与分析

实验平台为一个 10 节点的同构 Infiniband 机群，每个节点有 4 颗 AMD 2.4GHz Opteron 880 双核处理器（共 40 个处理器，80 个核）。同一个 Socket 内的双核各自有独立的 1M 二级缓存和 2G RAM，节点间通过 5Gb/s 5μs 延迟的 Infiniband 交换机互联。操作系统为 64 位 CentOS（内核为 2.6.18），MPI 库为版本 1.2.6 的 OpenMPI。Matlab 的神经网络工具箱[10]被用来训练基于人工神经网络的优化模型。本文设定当通讯器大小一定时，尽可能多地使用单一节点上的核来增加节点内通信并减少节点间通信。

4.1　IS（Integer Sort）整数排序基准

IS 基准是 NAS 并行基准套件中 5 个核心基准之一，用来实现整数排序。IS 基准主要包括 MPI_Alltoallv，MPI_Allreduce 和 MPI_Alltoall 3 个收集操作，主要的数据交换和通讯开销由 MPI_Alltoallv 产生。对 IS 基准的不同问题规模 class A，Class B，Class C 和 Class W 分别运行一次后可以抽

取 IS 基准的程序特征向量作为训练后的优化模型输入。IS 基准的程序特性可以归纳为:收集通讯占用时间 100%,收集通讯交换的数据量依赖于所选的问题规模和通讯器的大小。

图 4 对比了 IS 基准在不同问题规模下通过调整运行时参数可以获得的实际最大加速比与用本文模型预测得到的优化的运行时参数带来的加速比。对于 IS 基准,通过调优运行时参数可以获得的最大加速比为 20%(Class A)。图 4 同时显示了用基于 ANN(人工神经网络)方法预测得到的优化运行时参数为 IS 基准带来的加速比平均达到了实际最大加速比的 95%。对于取值只有 0 和 1 两种可能的 mpi_paffinity_alone 参数,神经网络预测的准确度较高,约为 75%。对于其他数值型参数,神经网络预测的准确度相对较低。例如对于问题规模较大的 Class C,实际最佳运行时参数组合为{1024,32768,8192,262144,1},使用 ANN 预测结果为{ 512,32768,1024,262144,1}。因此后续工作中将通过增大训练数据集,以及改进 ANN 训练数据前处理与后处理的方式来进一步提高神经网络的预测准确度。

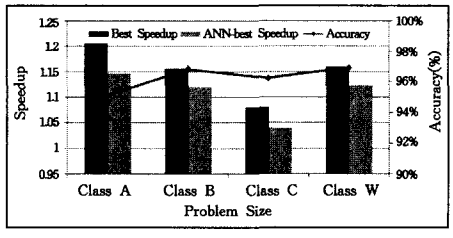

图 4 ANN 为 IS 基准预测参数产生的加速比与预测产生加速比的准确度

4.2 LU 基准

LU 基准是 NAS 并行基准套件中 3 个应用模拟基准之一,主要包括点到点通讯和 MPI_Allreduce,MPI_Barrier 和 MPI_Bcast 3 个收集操作。LU 基准中主要的数据交换和通讯开销由点到点通讯(发送和接收数据)产生。对 LU 基准的不同问题规模 Class A,Class B 和 Class W 分别运行一次后可以抽取 LU 基准的程序特性向量作为训练后的优化模型输入。LU 基准的收集通讯数据量较小,可以忽略不计,因此程序特性可以归纳为:点到点通讯占用时间 100%,点到点通讯交换的数据量依赖于所选的问题规模和通讯器的大小。

图 5 对比了不同问题规模下 LU 基准通过调整运行时参数可以获得的实际最大加速比与使用 ANN 模型预测得到的优化的运行时参数带来的加速比。同时显示了用基于 ANN(人工神经网络)方法预测得到的优化运行时参数为 IS 基准带来的加速比平均达到了实际最大加速比的 95%以上。

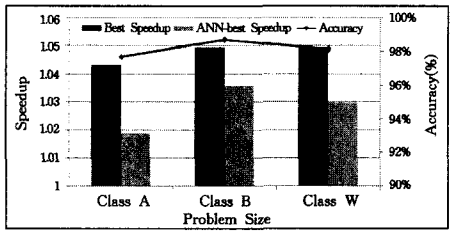

图 5 ANN 为 LU 基准预测参数产生的加速比与预测产生加速比的准确度

结束语 本文提出了一种在多核机群环境下优化 MPI 应用的新方法:运用人工神经网络对多核机群下 MPI 应用的最优运行时参数进行预测。本文设计了具有不同点到点通讯与收集通讯数据比例的训练基准在特定的多核机群下产生训练数据,同时采用能产生多个输出并具有较好抗噪性的神经网络来构建运行时参数优化模型。通过训练基准产生的训练数据对优化模型进行训练,训练后的模型用来预测未知的输入 MPI 程序的最优运行时参数。实验证明,基于 ANN 的预测模型得到的优化运行时参数产生的加速比平均达到了实际最大加速比的 95%以上。

未来工作将进一步抽取更丰富的 MPI 程序特征,以提高预测参数的准确率。由于在计算密集的应用中,将进程与核绑定可能会降低性能,因此未来将考虑抽取程序中计算与通讯比例特征。同时可以考虑节点内带宽与节点间带宽以及 MPI 应用的通讯特征,优化多核机群下 MPI 进程摆放或进程分区策略,以进一步优化多核机群下 MPI 应用的性能。

参 考 文 献

[1] Chai L, Lai P, Jin H W, et al. Designing an efficient kernel-level and user-level hybrid approach for MPI intra-node communication on multi-core systems[C]// ICPP '08: Proceedings of the 2008 37th International Conference on Parallel Processing. Washington, DC, USA, IEEE Computer Society, 2008

[2] TOP 500 Team. TOP500 Report for June 2009 [EB/OL]. http://www.top500.org

[3] Chaarawi M, Squyres J M, Gabriel E, et al. A tool for optimizing runtime parameters of Open MPI[C]// Proceedings of the 15th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface. Berlin, Heidelberg, Springer-Verlag, 2008: 210-217

[4] Pellegrini S, Wang Jie, Fahringer T, et al. Optimizing MPI Runtime Parameter Settings by Using Machine Learning[C]// Proceedings of the 16th Euro PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface. Espoo, Finland, 2009

[5] Ipek E, Supinski B R, Schulz M, et al. An Approach to Performance Prediction for Parallel Applications[C]// Euro-Par Parallel Processing. Monte de Caparica, Portugal, August 2005

[6] Lee B C, Brooks D M, de Supinski B, et al. Methods of inference and learning for performance modeling of parallel applications [C]// PPoPP: 12th Symposium on Principles and Practice of Parallel Programming. San Jose, CA, March 2007

[7] Pjesivac-Grbovic J, Fagg G E, Bosilca G, et al. Decision Trees and MPI Collective Algorithm Selection Problem[C]// Euro-PAR 2007. LNCS 4641, 2007

[8] Duan Rubing, Nadeem F, Wang Jie, et al. A hybrid intelligent approach for performance modeling and prediction of workflow activities in Grids[C]// 9th International Symposium on Cluster Computing and the Grid. IEEE Computer Society, Shanghai, China, 2009

[9] Battiti R. First-and second-order methods for learning between steepest descent and Newton's method[J]. Neural Computation, 1992, 4(2): 141-166

[10] Matlab: Neural Network Toolbox[EB/OL]. http://www.mathworks.com/products/neuralnet/