

## 基于神经网络的多核功耗预测策略

袁景凌 缪旭阳 杨敏龙 向尧

(武汉理工大学计算机科学与技术学院 武汉 430070)

**摘要** 多/众核处理器是计算机发展的趋势。在多/众核处理器的设计过程中,如何从庞大的设计空间中找出满足条件的设计结构,成为了关键和难点。为了解决传统软件模拟技术开销大、效率低等问题,提出了基于神经网络的模型来预测多核处理器的性能和功耗,建立了 BP 与 RBF 两种神经网络预测模型,利用 SESC 模拟器进行 CPI 与 POWER 模拟,并比较分析了两种预测模型的预测精度和可靠性。模拟结果表明,采用神经网络预测模型平均误差控制在 1.6%~6.6%,较传统的软件模拟等方法,能更有效地节省时间、提高效率,其中,RBF 神经网络预测模型具有更好的预测精度。

**关键词** 多核体系结构,机器学习,SESC 模拟,预测模型

中图法分类号 TP183 文献标识码 A

### Neural Network Based Power Prediction Strategy for Multi-core Architecture

YUAN Jing-ling MIAO Xu-yang YANG Min-long XIANG Yao

(Department of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

**Abstract** Multi/many-core processor is a trend of computer development. In the design process of multi-core processor, how to find the design structure from a large design space to meet the conditions has become a critical and challenging task. In order to solve the high overhead and low efficiency problem of traditional software simulation technology, this paper proposed a model based on neural network to predict the performance and power of multi-core processors, established two kinds of BP and RBF neural network prediction models, used SESC simulator to simulate CPI and power, and compared and analyzed the prediction accuracy and reality of two kinds of prediction models. Simulation results show that the neural network prediction model average error is 1.6%~6.6% in control. Compared to general software simulation method, it can also save time and improve efficiency. More importantly, RBF neural network prediction model has better prediction accuracy.

**Keywords** Multi-core architecture, Machine learning, SESC simulation, Prediction model

### 1 简介

多/众核处理器已经成为计算机发展的趋势<sup>[1,2]</sup>。现有的研究表明,对体系结构和微体系结构设计空间和负载空间的有效分析,能够指导体系结构设计和资源优化。但随着核数的不断增加,理解和分析这些大规模分布式体系结构特征,面临着极大的挑战,并且随着硬件规模增大,能耗和散热问题日益严重。从单核扩展到多/众核分析,研究者采用各种模拟方法,提前探索设计空间的特征,寻求以最小的时间和资源开销获取更多特征信息。主要有传统的基于时钟周期级别的详细模拟<sup>[3]</sup>、采样模拟<sup>[4]</sup>、FPGA 加速模拟<sup>[5]</sup>和统计模拟<sup>[6]</sup>等。为了降低模拟参数的数量和减少模拟的开销,各种分析模型如线性回归分析模型<sup>[7]</sup>、样条回归模型<sup>[8]</sup>、神经网络模型等<sup>[9-11]</sup>也被应用到设计空间的参数预测和分析。近年来,研究者开始将机器学习方法应用在特征提取、资源调度、负载平

衡和微体系结构设计空间探索等重要问题中<sup>[8-11,14-21]</sup>,取得了一定的成果。如神经网络被应用到构造体系结构性能参数与设计参数的预测模型中<sup>[8-11,16]</sup>。特别值得关注的是,中科院与南京大学合作研究了半监督学习 COMT 方法,用于微体系结构设计参数的预测和分析,得到了更优的设计参数配置<sup>[14]</sup>。

常见的多/众核体系结构分析方法主要包括 3 类。第一类,传统基于时钟周期级别的模拟方法<sup>[3]</sup>,对于大规模众核系统的模拟非常耗时且昂贵;第二类,改进的快速模拟方法<sup>[4-6,12]</sup>,能减少单个设计参数的模拟数量,但对于整个设计空间而言,仿真规模依然庞大。上述两类方法,随着核数增加,模拟参数也越来越多,无法准确地预测复杂大规模分布式众核体系结构的 2D 空间行为。第三类,为了进一步减少模拟规模和开销的预测模型<sup>[7-11,14]</sup>,相对于传统方法有较大优势,特别是文献<sup>[14]</sup>采用的半监督学习算法,进一步减少了模

本文受国家自然科学基金(61303029),留学回国人员科研启动基金([2012]1707),中央高校基本科研业务费专项资金(2013-IV-054)资助。  
袁景凌(1975—),女,博士,教授,CCF 高级会员,主要研究领域为机器学习等智能方法及应用、计算机体系结构,E-mail: yuanjingling@126.com;  
缪旭阳(1989—),男,硕士,主要研究领域为多核体系结构、智能算法;杨敏龙(1988—),男,硕士,主要研究领域为机器学习算法、计算机体系结构;向尧(1989—),男,硕士,主要研究领域为多核体系结构、数据挖掘。

型的训练开销,但这类方法对模型本身的要求甚高,并且需要对体系结构特征进行有效的挖掘和分类,以提高预测模型本身的准确性和有效性。

因此本文主要针对多/众核处理器的参数及性能分析仅采用传统时钟细节模拟技术十分耗时等问题,研究机器学习算法,构建了两种基于神经网络的功耗特征预测模型,实现了对多核性能功耗的快速获取,提高了设计空间探索效率。

## 2 基于机器学习的预测模型

预测模型方法正是从减少模拟设计组合数量的方向来提高设计空间探索的效率。图1为预测模型方法的基本框架。预测模型通常会分为训练阶段和预测阶段。在训练阶段,我们利用随机抽样,从设计空间可能的组合中抽取部分样本进行模拟,得到模拟的性能和功耗结果。根据抽取样本的参数和对应的响应结果,通过统计分析或者机器学习建立预测模型。在预测阶段,从整个设计空间中再次随机抽取样本(与训练阶段样本不重复),不再需要耗时的详细模拟过程,只需要将其输入到预测模型中就可以直接预测其模拟的结果,从而避免了大量耗时的模拟操作。

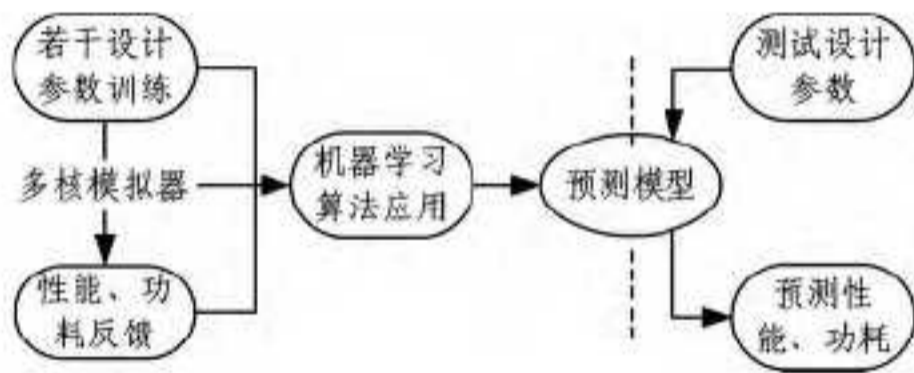


图1 预测模型方法基本框架

表1 配置参数表

配置参数	Fetch width	ROB size	IQ size	L2 Size(kB)	L2 lat	l1 size(kB)	dl1 size(kB)	dl1 lat
参数取值	2,4,8,16	64,96,128,160	32,64,96,128	256,1024,2048,4096	8,12,14,16,20	8,16,32,64	8,16,32,64	1,2,3,4

实验需要从每个配置参数的参数取值中选取一个值形成一组参数值,可能组合共有  $N=4 \times 4 \times 4 \times 4 \times 5 \times 4 \times 4 \times 4 = 81920$  种。

本实验计划采集 50 组不同的参数组合进行实际模拟,包括 40 组训练参数数据和 10 组预测参数数据,前 40 组数据用于神经网络的训练过程,在网络训练完成后,将利用后 10 组数据预测 CPI 性能值和功耗值,并与通过实际模拟获取的 CPI 性能值和功耗值进行比较。后 10 组预测参数数据可以采用随机函数选取,而训练参数数据的选择却十分重要,它可能会影响后面的神经网络训练过程。为了提高预测模型的精度,我们需要选择比较合理的参数组合。在训练数据的选取过程中,采用简单的随机函数并不能确保样本参数分散到整个数据集中或者遍布所有的参数。为了使得样本数据集容量尽量小、消耗成本尽量低却又能使整个可选数据集中的每一个数据都能够被包含,我们采用拉丁超立方体抽样(Latin Hypercube Sampling, LHS)。它是一种高效的抽样方法,在抽样效率和运行时间方面有着巨大的好处,LHS 产生的样本比较均匀地分布在整个可选样本空间内,为后面的神经网络预测提高了精度,更加全面地覆盖了所有数据并且具有更好的代表性。

在机器学习领域,预测模型将数据的学习与数学应用结合在一起,其中,数据学习可以转换为寻找一组输入数据字段与某个响应或目标变量之间的映射函数,数学应用是由其它输入和映射函数确定近似的目标变量机器学习及其应用。如果要构建一个预测模型,首先需要整理将要用于训练的输入数据集,并列出需要研究的输入参数字段,在本文中即为各种设计配置参数,例如发射宽度(fetch-width)、ROB-size、IQ-size、二级缓存大小(L2-size)等。假设现在有  $n$  组输入数据,需要分析  $t$  种配置参数对目标变量的影响,我们将这些数据设为  $x_1, x_2, \dots, x_n$ , 其中  $x_i$  是一个  $t$  维向量,每个向量都是一个配置参数,对  $x_i$  进行模拟得到  $y_i$ , 在本文中即为处理器的性能、功耗和可靠性等指标,分别对  $x_1, x_2, \dots, x_n$  进行模拟,得到对应的  $n$  个目标变量  $y_1, y_2, \dots, y_n$ , 利用  $n$  组输入数据与  $n$  个目标变量,采用机器学习的一些算法(决策树、SVM 和神经网络)构建预测模型。在文献[22]中,采用支持向量基的方法进行预测。性能预测平均误差值分布于 0.52%~10.1% 之间,功耗预测平均误差值分布于 1.08%~5.17% 之间。

## 3 神经网络功耗预测模型构建

### 3.1 数据样本选择

在多核体系结构中影响微体系结构的配置参数有很多。为了探索多核处理器设计空间,通过实验选取影响性能比较大的 8 个重要参数:发射宽度(Fetch-width)、ROB-size、IQ-size、二级缓存大小(L2-size)、二级缓存命中延迟(L2-lat)、I 缓存大小(il1-size)、d 缓存大小(dl1-size)、d 缓存命中延迟 dl1-lat,如表 1 所列。

### 3.2 SESC 模拟

在选取好合适的样本数据后,我们采用了 SESC 模拟器(即 SuperEScalar Simulator),它是一款能够仿真微处理器体系结构的模拟器,主要由伊利诺伊大学和其他大学的部分研究团队组成的 i-acoma 组织共同研发。经实验验证,SESC 在 Intel pentium4 3GHz 的处理器上的执行速度超过了每秒 150 万指令,可见模拟器的运行速度非常快。图 2 是某一个应用程序通过 SESC 模拟器模拟的具体流程。将样本数据中 40 组配置参数全部使用 SESC 进行模拟,获取对应的 CPI 性能值和功耗值。本文选取了著名的 SPLASH-2 基准测试程序中的 3 个程序:fft、radix、fmm 进行测试。

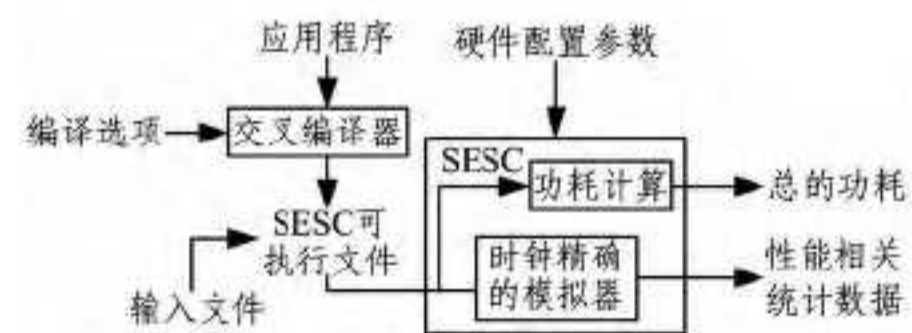


图2 SESC 模拟器框架图

在负载模拟运行后,我们获取文件报告的可视化结果,从中收集到 CPI 性能值和功耗值。表 2 列举了几组模拟运行 fft 负载后采集的性能值和功耗值。

表 2 几种模拟的性能值和功耗值结果

配置参数	Fetch width	ROB size	IQ size	L2 size	L2 lat	l1 size	dl1 size	dll lat	fft 负载性能值	fft 负载功耗值
组合 1	2	128	32	1024	14	8	32	2	0.6619	48.488
组合 2	16	128	128	2048	8	32	16	1	0.74951	52.447
组合 3	8	64	32	4096	14	8	64	2	0.60458	42.495

### 3.3 归一化处理

为了避免学习速度很慢、训练过程耗时过长、训练网络收敛过程较慢、奇异样本数据出现等问题,需要对原始数据做一定的数据预处理,即归一化处理。将原始数的取值范围从  $[\min, \max]$  映射到  $[-1, 1]$  区间。本文采用线性函数转换,变换方法为  $Y = 2 * (x - \min) / (\max - \min) - 1$ 。  $x, y$  分别对应着转换前和转换后的值,  $\min$  和  $\max$  分别对应着所有样本的最大值和最小值。表 3 列出几组原始数据变换后的结果。

表 3 几组原始数据与归一化数据的对比

配置参数	组合 1	变换 1	组合 2	变换 2	组合 3	变换 3
fetch-width	2	-1	16	1	8	-0.1429
ROB-size	128	0.3333	128	0.3333	64	-1
IQ-size	32	-1	128	1	32	-1
L2-size	1024	-0.6	2048	-0.6667	4096	1
L2-lat	14	0	8	-1	14	0
l1-size	8	-1	32	-0.1429	8	-1
dl1-size	32	-0.1429	16	-0.7143	64	-1
dll-lat	2	0.3333	1	-1	2	-0.3333
fft 负载性能值	0.6619	0.021	0.74951	0.7805	0.60458	-0.4759
fft 负载功耗值	48.488	0.1652	52.447	0.7591	42.495	-0.7339

### 3.4 BP 和 RBF 神经网络训练

我们选择了比较常见的反馈 (BP) 神经网络和径向基 (RBF) 神经网络进行实验。在实际问题中的应用,如果遇到比较复杂的输入输出映射关系,往往需要使用多隐层神经网络。根据 komogorow 定理,我们都能够用单隐层网络来模拟任意一个连续函数,其中输入层有  $m$  个节点,中间层有  $2m + 1$  个节点,输出层有  $n$  个节点,所以本实验采用单隐层网络进行训练。在 BP 网络的具体结构确定后,还需要确定训练过程中的一些参数。训练函数和学习率会影响整个网络的训练过程,它们影响着每一次训练过程中权值的具体变化。训练函数有 `traingd`、`traingdm`、`traingda`、`traingdx`、`trainlm` 等各种模式,本实验经过多次测试比较,选用 `traingdx` 作为训练函数效果最佳。

根据以上分析,本实验构建输入层节点数为 8、单隐藏层节点数为 10、输出层节点数为 1 的 BP 网络。设定学习率为 0.01,最大训练轮回数为 10000,目标均方误差设为 0.0001,训练函数为 `traingdx`。在经过 13.789 秒、1632 次训练以后达到目标预测误差 0.0001,训练停止。采用类似的分析,我们再构建输入层节点数为 8、输出层节点数为 1 的 RBF 网络。目标均方误差 `goal` 设为 0.0001,扩展速度 `spread` 设为 3,最大神经元数目与输入节点数相同,设为 40,两次显示之间添加的神经元数目为 1。在经过 5.445 秒、37 次训练以后达到目标预测误差 0.0001,训练停止。图 3 和图 4 分别为 BP 网络和 RBF 网络训练记录曲线。

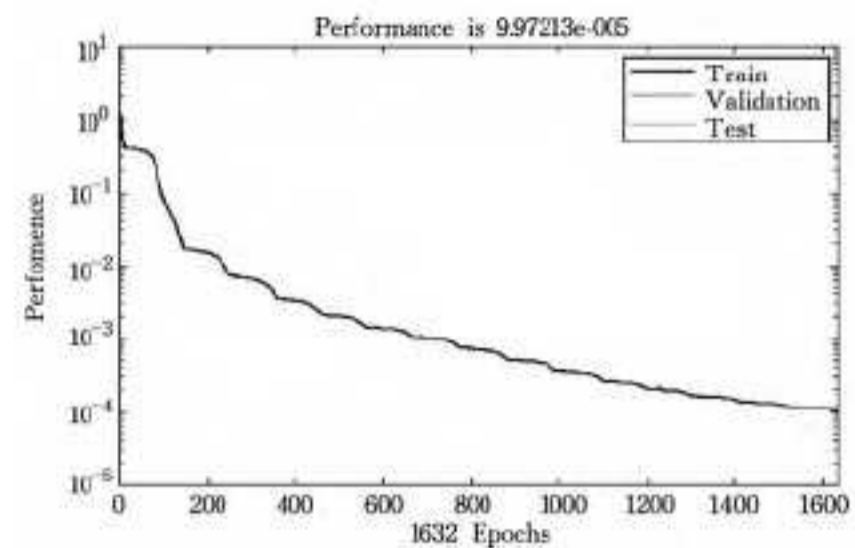


图 3 BP 网络训练记录曲线

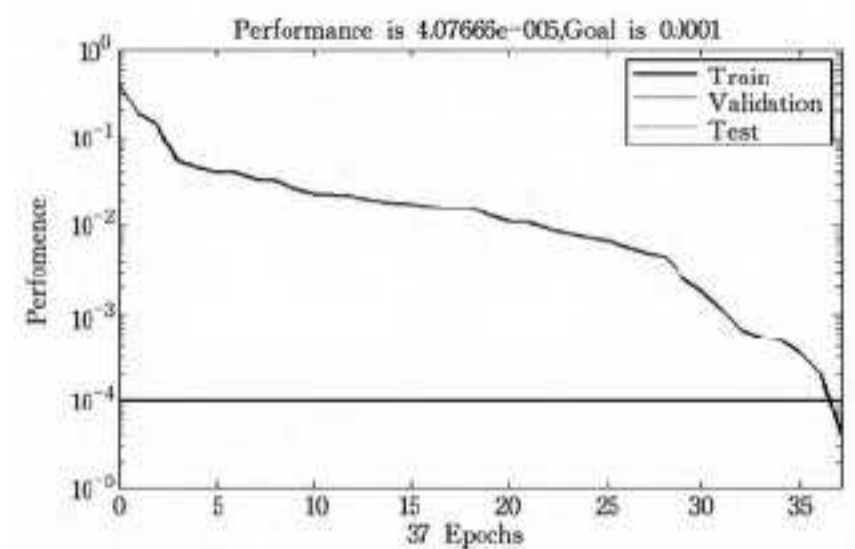


图 4 RBF 网络训练记录曲线

对于网络的训练时间,BP 预测模型的训练时间大约在 10.9s~17.7s 之间,fft 和 radix 的预测时间较长, `fmm` 的预测时间较短,而 RBF 预测模型的训练时间集中在 4.5s~5.1s 之间,对于不同的程序,RBF 网络的训练时间相差不大,训练过程中 RBF 收敛速度较快,达到了目标均方误差要求值时就停止训练,当精度提到较高的时候,不同负载的训练时间才会有差异。总体上,BP 网络的训练时间显然要比 RBF 网络的训练时间长得多,总平均耗时达到了 RBF 的 2.9 倍。

## 5 BP 和 RBF 神经网络预测及分析

经过训练后,我们使用 `sim` 函数,根据 10 组测试样本数据预测 CPI 性能值和功耗值,并将 BP 和 RBF 预测的结果与实际模拟值进行对比。`Sim` 函数的格式为:  $[Y, Pf, Af, E, perf] = \text{sim}(\text{net}, P, Pi, Ai, T)$ ,其中, `net` 是已经训练好的神经网络, `P` 是要给定某一个输入向量, `Pi` 与 `Pf` 与分别对应着训练开始和训练终止时的输入延迟, `T` 是目标矢量, `Y` 是由 `P` 所预测得到的结果, `Ai` 与 `Af` 分别对应着训练开始和训练终止时的层延迟状态, `E` 是 `Y` 与 `T` 之间的误差, `perf` 是网络性能状态。最终预测结果如图 5—图 10。

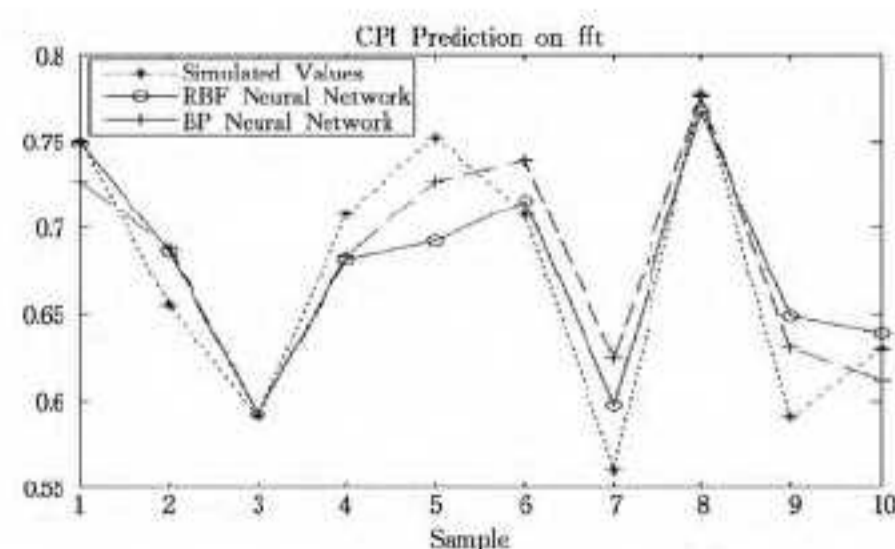


图 5 BP 和 RBF 神经网络在 fft 下的 CPI 预测

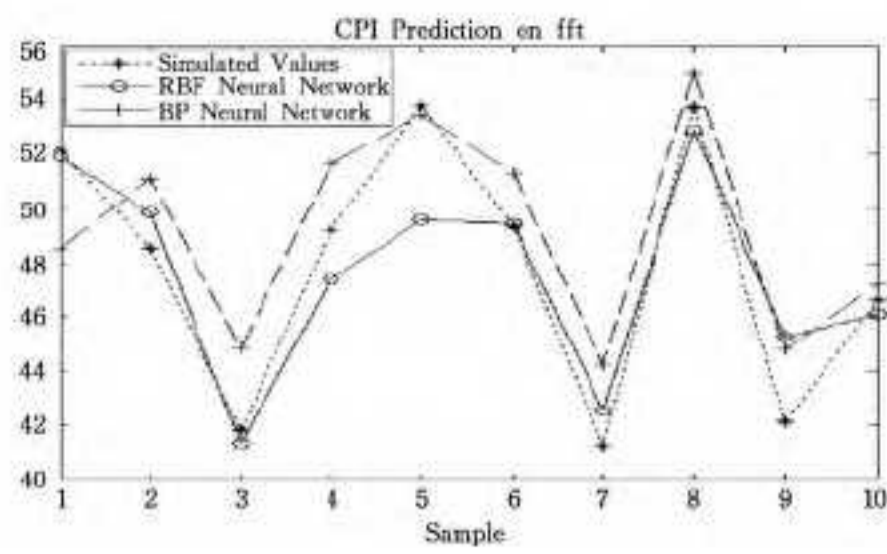


图 6 BP 和 RBF 神经网络在 fft 下的功耗预测

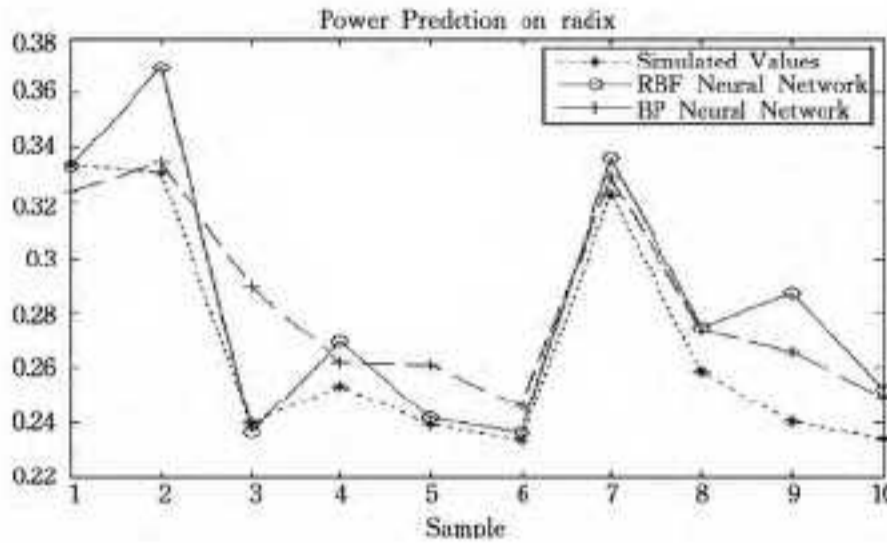


图 7 BP 和 RBF 神经网络在 radix 下的 CPI 预测

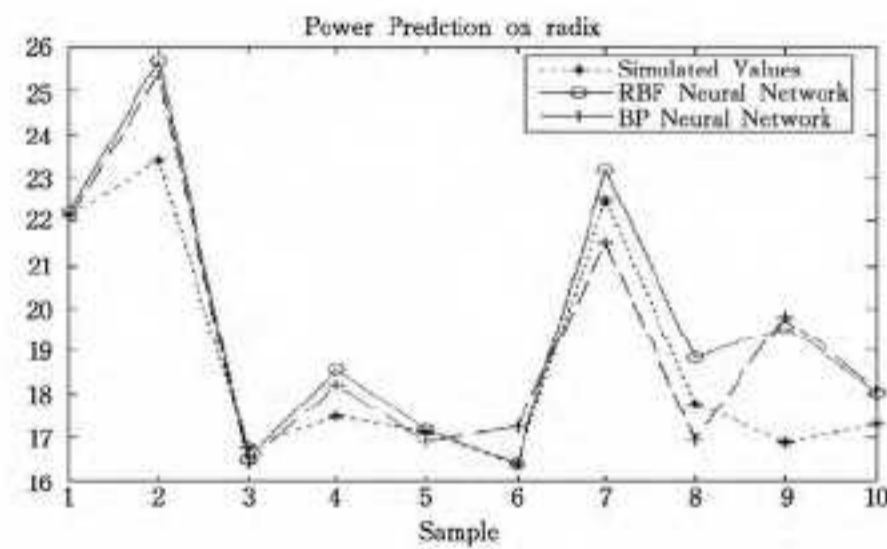


图 8 BP 和 RBF 神经网络在 radix 下的功耗预测

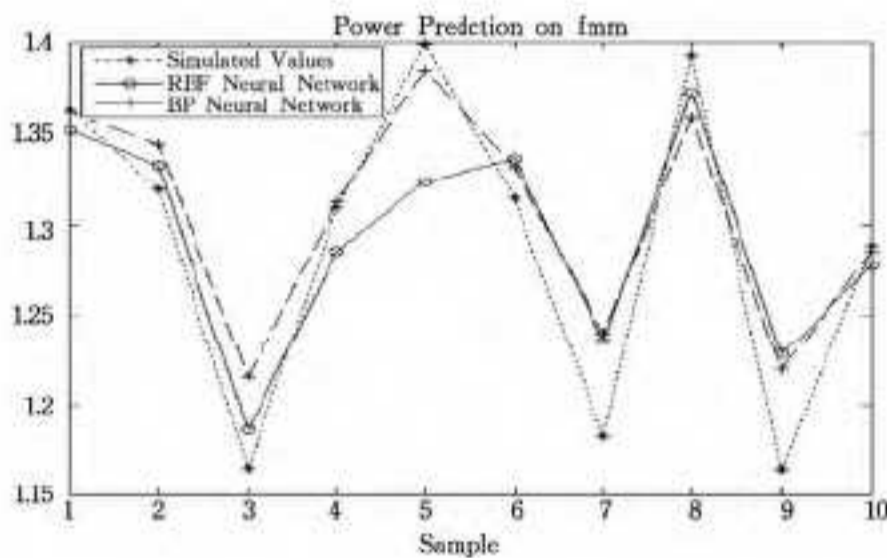


图 9 BP 和 RBF 神经网络在 fmm 下的 CPI 预测

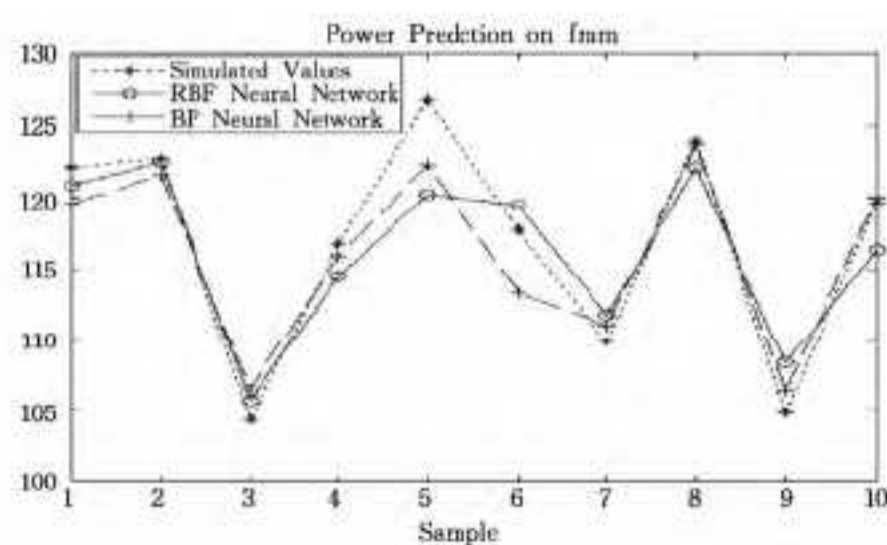


图 10 BP 和 RBF 神经网络在 fmm 下的功耗预测

图 5—图 10 分别给出了在 fft、radix、fmm 不同负载下两种神经网络预测的结果。

1) 通过分析比较,与实际模拟值对比,两种预测模型的平均误差都在 1.6%~6.6% 之间,整体上 RBF 的预测误差要比 BP 神经网络的低。

2) 从 BP 预测曲线看,虽然 BP 神经网络预测在个别样本上存在较大误差,但整体上 BP 神经网络能够较好地预测不

同负载下的性能和功耗值,整体预测趋势符合实际趋势的走向。可见 BP 神经网络具备非线性拟合能力和逼近能力,具有较高的模拟精度。

3) 从 RBF 预测曲线看,RBF 神经网络可以很精确地预测在不同负载下的性能和功耗值。但是其中 radix 程序较难预测,在该程序下使用 RBF 神经网络预测存在一定误差。同时,在第 5 个样本点上的误差较为明显,可能是该样本点的配置参数取值过于极端造成的。

4) 不管是 BP 还是 RBF 神经网络预测,与统计采样和统计模拟方法相比,都能够利用较少的开销,较好地预测不同负载下的性能和功耗值。

5) 相对于其他机器学习模型如 SVM<sup>[23]</sup>,本文性能预测平均误差值分布于 2.12%~6.63% 之间,功耗预测平均误差值分布于 1.63%~5.21% 之间。总体上,两种方法平均误差值相近,而神经网络相对于支持向量基而言,预测精度更稳定一些,误差集中在一个较小的区间内,而支持向量基的误差变化幅度较大。

结束语 本文建立了 BP 与 RBF 两种神经网络预测模型,利用 SESC 模拟器对多核处理器的 CPI 与 POWER 等进行模拟,并分析比较了两种预测模型的精度和可靠性,结果表明 BP 网络和 RBF 网络具有较强的非线性拟合能力,它们能根据多核设计参数,较好地预测不同负载的执行性能和功耗值;基于神经网络的预测模型与传统软件模拟技术等方法相比,能有效降低模拟器模拟多核配置的次数,节省开销;RBF 网络相对于 BP 网络,具有更快的收敛速度、更高的预测精度和可靠性。这些机器学习方法的开销主要集中在模型训练阶段,在预测和分析阶段开销并不大且泛化能力强。在今后研究中,我们将尝试选择应用更多的机器学习模型,找到适合多/众核体系结构的参数预测和性能分析的高效方法。

## 参考文献

- [1] 计算机体系结构国家重点实验室. 21 世纪计算机体系结构[J]. 中国计算机学会通讯, 2012, 12(8): 70-81
- [2] 王海霞, 汪东升. 多/众核处理器的关键技术[J]. 中国计算机学会通讯, 2009, 11(5): 12-19
- [3] Hu W, Wang J, Gao X, et al. Godson-3: a scalable multicore RISC processor with X86 emulation[J]. IEEE Micro, 2009, 29(2): 17-29
- [4] Van Biesbrouck M, Eeckhout L, Calder B. Considering all starting points for simultaneous multithreading simulation[C] // Proceedings of ISPASS. 2006: 143-153
- [5] Chiou D, Sunwoo D, Kim J, et al. FPGA-accelerated simulation technologies (FAST): fast, full-system, cycle-accurate simulators [C] // Proceedings of MICRO. 2007: 249-261
- [6] Genbrugge D, Eeckhout L. Chip multiprocessor design space exploration through statistical simulation[J]. IEEE Transactions on Computers, 2009, 58(12): 1668-1881
- [7] Joseph P J, Vaswani K, Thazhuthaveetil M J. Construction and use of linear regression models for processor performance analysis[C] // Proceedings of HPCA. 2006: 99-108
- [8] Joseph P J, Vaswani K, Thazhuthaveetil M J. A predictive performance model for superscalar processors[C] // Proceedings of MICRO. 2006: 161-170
- [9] Lee B, Collins J, Wang H, et al. CPR: composable performance regression for scalable multiprocessor models[C] // Proceedings of MICRO. 2008: 270-281

- [10] Hamerly G, Perelman E, Lau J, et al. Using machine learning to guide architecture simulation[J]. *Journal of Machine Learning Research*, 2006, 7(2): 343-378
- [11] Ipek E, McKee S A, Caruana R. Efficiently exploring architectural design spaces via predictive modeling[C]// *Proceedings of ASPLOS*. 2006: 195-206
- [12] Jung H, Ju M, Che H. A Theoretical Framework for Design Space Exploration of Many Core Processors[C]// *Proceedings of MASCOTS*. 2011: 117-125
- [13] 喻之斌, 金海, 邹南海. 计算机体系结构软件模拟技术[J]. *软件学报*, 2008, 19(4): 1051-1068
- [14] Guo Q, Chen T Y, Zhou Z, et al. Effective and efficient micro-processor design space exploration using unlabeled design configurations[C]// *Proceedings of IJCAI*. 2011: 1671-1677
- [15] Tesauro G. Online resource allocation using decomposition reinforcement learning[C]// *Proceedings of AAAI*. 2005: 886-891
- [16] Ganapathi A, Kuno H, Dayal U. Predicting multiple performance metrics for queries; better decisions enabled by machine learning [C]// *Proceedings of ICDE*. 2009: 592-603
- [17] Cho C-B, Poe J, Li Tao, et al. Accurate, scalable and informative design space exploration for large and sophisticated multi-core oriented architectures[C]// *Proceedings of MASCOTS*. 2009: 16-25
- [18] 李胜梅, 程步奇, 高兴誉, 等. 主成分线性回归模型分析应用程序性能[J]. *计算机研究与发展*, 2009, 46(11): 1949-1955
- [19] Huffmire T, Sherwood T. Wavelet-based phase classification [C]// *Proceedings of PACT*. 2006: 95-104
- [20] Yuan Jing-ling, Jiang Tao, Zhong Luo. Grey neural network based predictive model for multi-core architecture 2D spatial characteristics[C]// *Proceedings of ISNN*. LNCS551, 2009: 889-892
- [21] Martinez J F, Ipek E. Dynamic multi-core resource management; a machine learning approach[J]. *IEEE MICRO*, 2009, 29(5): 8-17
- [22] 余凯, 贾磊, 陈雨强, 等. 深度学习的昨天, 今天和明天[J]. *计算机研究与发展*, 2013, 50(9): 1799-1804
- [23] Pang Jiu-feng, Li Xian-feng, Xie Jin-song. Microarchitectural Design Space Exploration via Support Vector Machine[J]. *Acta Scientiarum Naturalium Universitatis Pekinensis*, 2010, 46(1): 55-63

(上接第 46 页)

直到系统计算资源被完全利用, 不仅可缩短整体作业运行时间, 还可提高系统资源利用率。

**结束语** 本文针对异构的 Hadoop 集群提出基于 DAG 的 MapReduce 调度算法。该算法把每个提交给 Hadoop 的作业当成一个 DAG, 因此需要对每个作业进行 DAG 转化。针对集群的异构性将节点进行分类, 其目的是: (1) 合理分布数据; (2) 选择适当的节点运行任务。采用改进向上排序值使其在异构集群中更准确地算出每个任务的优先级, 再根据集群节点的计算能力及当前使用状况合理地选择节点执行任务。通过实验证明, 本文提出的方法能有效提高数据本地性并减少整个作业集的完成时间。

## 参 考 文 献

- [1] Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters[J]. *Communications of the ACM*, 2008, 51(1): 107-113
- [2] Apache Hadoop. Hadoop [EB/OL]. <http://hadoop.apache.org/>, 2009-03-06
- [3] Vaquero L M, Rodero-Merino L, Caceres J, et al. A Break In the cloud: Towards a Cloud Definition[J]. *ACM SIGCOMM Computer Communication Review*, 2009, 39(1): 50-55
- [4] 陆嘉恒. Hadoop 实战(第 3 版)[M]. 北京: 机械工业出版社, 2011
- [5] Zaharia M, Borthakur D, Sarma J S, et al. Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling[C]// *Proceedings of the 5th European Conference on Computer Systems*. 2010: 265-278
- [6] Xie J, Yin S, Ruan X J, et al. Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters[C]// *IEEE International Symposium on Parallel & Distributed Processing, Workshops and PhdForum*. 2010: 1-9
- [7] Zhang X H, Zhong Z Y, Feng S Z, et al. Improving Data Locality of MapReduce by Scheduling in Homogeneous Computing Environments[C]// *IEEE 9th International Symposium on Parallel and Distributed Processing with Applications*. 2011: 120-126
- [8] Guo Lei-tao, Sun Hong-wei, et al. A data distribution aware task scheduling strategy for mapreduce system[C]// *First International Conference on Cloud Computing*. 2009
- [9] Verma A, Cherkasova L, Campbell R. Resource Provisioning Framework for MapReduce Jobs with Performance Goals[J]. *Lecture Notes in Computer Science*, 2011, 7049: 165-186
- [10] Polo J, Carrera D, et al. Performance-driven task co-scheduling for mapreduce environments[C]// *Proc of IEEE/IFIP Network Operations and Management Symposium*. 2010
- [11] Kc K, Anyanwu K. Scheduling Hadoop Jobs to Meet Deadlines [C]// *IEEE Second International Conference on Cloud Computing Technology and Science*. 2010: 388-392
- [12] Polo J, Carrera D, Becerra Y, et al. Performance-Driven Task Co-Scheduling for MapReduce Environments[C]// *IEEE proceedings of Network Operations and Management Symposium*. 2010: 373-380
- [13] Tang Zhuo, Zhou Jun-qing, Li Ker-li, et al. MTSD: A task scheduling algorithm for MapReduce base on deadline constraints[C]// *IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*. 2012
- [14] Zaliva V, Orlov V. Hamake: A Data Flow Approach to Data Processing in Hadoop[C]// *CLOSER*. 2012: 457-461
- [15] Furst S. Challenges in the design of automotive software[C]// *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2010: 256-258
- [16] Arabnejad H, Barbosa J. Fairness resource sharing for dynamic workflow scheduling on Heterogeneous Systems[C]// *Parallel and Distributed Processing with Applications (ISPA)*, 2012 *IEEE 10th International Symposium on*. IEEE, 2012: 633-639
- [17] Klobedanz K, Koenig A, Mueller W. A reconfiguration approach for fault-tolerant flexray networks[C]// *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011. IEEE, 2011: 1-6