

多核处理器可重构 Cache 功耗计算方法的研究

方 娟 陈 欣

(北京工业大学计算机学院 北京 100124)

摘 要 多核动态可重构 Cache 是解决 Cache 功耗困扰的一个重要方法。现有 Cache 功耗模拟器并不能很好地支持多核动态可重构 Cache 功耗研究,通过对多核动态可重构 Cache 的功耗模型进行研究,找到了计算可重构 Cache 的方法和思路,应用 CACTI 来分别构建各个组成结构的 Cache 功耗模型,以较为准确地测算可重构 Cache 的功耗。在 Simics 模拟器下构建动态可重构 Cache,运行测试程序,对比传统的体系结构,可重构 Cache 的功耗能够得到 10.4% 的降低。同时,实验中发现功耗的降低不仅仅是动态可重构 Cache 贡献的,而是由系统综合产生的,因此在低功耗设计中,要综合考虑整体系统的功耗和性能,避免片面地考虑 Cache 结构而导致整体功耗的提高。

关键词 多核处理器,动态重构,Cache,功耗

中图法分类号 TP393 文献标识码 A

Research of Reconfigurable Cache Method for Power Calculation in CMP

FANG Juan CHEN Xin

(College of Computer Science, Beijing University of Technology, Beijing 100124, China)

Abstract Multicore dynamically reconfigurable Cache is an important way to solve the Cache-power troubled. Existing Cache-power simulator does not support multi-core dynamic reconfigurable Cache-power research, reconfigurable multi-core dynamic power consumption model of the Cache was researched to find the methods and ideas that can calculate reconstruction Cache. CACTI was used to build the structure of the various components Cache power model to more accurately estimate reconfigurable Cache-power. In Simics simulator, building dynamically reconfigurable Cache, running the test, comparing conventional architectures, reconfigurable Cache power consumption can be obtained a 10.4% decrease. Also it is found that the decrease in power consumption is more than just Dynamically Reconfigurable Cache contribution, but the contribution by the integrated system, prompting a comprehensive consideration of the overall system power consumption and performance in a low-power design, to avoid one-sided consideration of the Cache structure, leading to the improvement of the overall power consumption.

Keywords Multi-core, Reconfigurable, Cache, Power

1 引言

随着半导体技术的发展,微处理器的新结构与设计层出不穷。在计算机处理器发展过程中性能提高的主要原因得益于时钟周期的缩短,指令访问的局部性原理也使微处理器的性能得到了显著的提高。伴随高性能处理器的快速发展,体系结构功耗问题也愈加显著。时钟频率是性能提高的必要因素,而系统的功耗又主要随着系统时钟频率的增长而增长。虽然设计人员在低功耗电路设计方面做出了很多努力和贡献,但是功耗还是伴随着微处理器的发展呈几何级上涨。

在开发过程中,每一代的微处理器的翻转电压降低了,也就意味着时钟翻转的功耗降低了,但是时钟频率的提高加上系统功能的扩展导致功耗仍然翻倍增长^[1]。到目前为止,这一问题的产生已经成了制约系统体系结构设计的重要矛盾。

在微处理器中,存储处于不可或缺的地位,而存储主要由

主存和 Cache 组成。存储的设计对于微处理器来说不言而喻,它的影响不仅体现在性能上,也体现在功耗上。为了提高性能以及降低功耗,现有的主流存储分为多级 Cache 加主存的结构。Cache 作为体系结构的重要组成部分,占据芯片大部分面积,同时也是功耗的主要来源。Cache 的配置参数包括尺寸、相联度、块的大小等,但单方面改变一种参数很可能会提高一方面性能而降低另一方面性能。因而如果能根据当前所运行程序的特性来重构一个适合当前应用的 Cache 配置,将会得到一个比较好的功耗结果。

动态多核可重构 Cache 是指在一个程序运行时,根据程序对系统资源的要求,动态监控程序运行状态,自动选择功耗最低的 Cache 结构。由于每次访问数据都需要通过 Cache,其功耗也就占到了系统功耗的很大一部分,经过统计表明,其功耗占到系统功耗 60% 左右^[2]。在体系结构设计中,优化这一部分功耗是非常有必要的。在本文中,讨论 Cache 功耗的

本文受国家自然科学基金(61202076),北京市教委科技计划面上项目(KM201210005022)资助。

方娟(1973—),女,博士,副教授,主要研究领域为计算机系统结构、多核计算,E-mail:fangjuan@bjut.edu.cn;陈欣(1988—),男,硕士生,主要研究领域为计算机系统结构。

计算模型的基础,就是通过可重构的 Cache 设计来降低功耗。这种设计的主要思想是由于系统的设计满足高性能,但是当系统处于低负荷状态运行时大容量 Cache 并不会得到充分的利用。此时适当容量的 Cache 即可满足要求。基于此种想法,在系统运行时判断系统状态,只分配给满足其运行的 Cache 容量大小,关闭剩余的空闲 Cache^[3]。这样既满足了程序的运行性能,又节省了功耗。

对 Cache 的优化设计将能够很大程度地提高系统的性能或者降低功耗。对于片上的 Cache 来说,可以进行改变的参数配置包括尺寸、块数、相联度等等,然而 Cache 的整体性能跟每个参数都相关,内部参数互相之间往往是反相关的^[4]。一个数据的提高往往伴随着其他参数的下降。总的结果并不能满足性能及功耗的全部要求。同一个 Cache 在不同的运行情况下,性能和功耗也都不尽相同。设计人员现在正在巧妙利用已有的大容量 Cache 来满足现有的功能^[5]。因为大容量 Cache 只是为了满足高性能运算而设计的,如果在程序不需要大容量 Cache 的时候,关闭掉一部分 Cache,将能够节约功耗并且满足程序的各种需求。

这种设计目前正在研究甚至付诸实践。目前,现有的体系结构功耗模拟器还存在诸多的不足,传统的功耗模型并不能很好地计算出其中 Cache 的功耗变化,本文的思想就是针对这种可重构 Cache 结构,根据主流通用的系统体系结构模拟器,设计一个可以支持动态多核可重构 Cache 的功耗计算模型。

2 可重构 Cache

可重构 Cache 的思想最早来源于嵌入式处理器。因为嵌入式处理器对于功耗的降低需求更为直接和强烈。这种技术可以很好地移植到目前的主流多核处理器当中^[6,7]。可以从两个方面来进行可重构 Cache 的设计,1)通过开始对程序的了解,设计优化后的 Cache 各项参数,或者是为每个指令片段都找出对应的优化后的 Cache 各项参数,统称为静态可重构 Cache。2)在程序的运行中,时时监控程序的需求,控制系统状态,为其选择最优的 Cache 结构,称为动态可重构 Cache。其中静态可重构在设计时更为简单,效果也更好,但是它需要首先了解所有程序的运行特征以及各项指标,因此实用性不强。然而动态可重构可以根据实际计算机的使用情况进行动态改变,本文讨论的就是动态可重构 Cache 的方法。在多核系统中,Cache 分为私有和共享两种方式,在私有 Cache 中,其本身就比较小,再进行重构显然是不现实的,所以对更大的共享 Cache 进行重构来完成功耗的降低^[8]。由于模拟器中的共享 Cache 为 L2Cache,本文也即对 L2Cache 进行重构。根据程序的运行状态相应调整共享 Cache^[9],使其能够在尽量满足程序需要的同时降低功耗。

3 功耗计算模型

多核处理器中,Cache 功耗包括静态功耗与动态功耗,动态功耗主要是指翻转功耗,由电压、频率、有效电容决定,而静态功耗是指由漏电流导致的功耗。由于静态功耗主要涉及电子元器件层面,本文对静态功耗不做考虑。动态功耗由 CACTI 来测算,CACTI 是一种独立的功耗评估模型,本文设计的模拟器为 simics + gems^[10]。模型本身不具有功耗评估模型,所以将 CACTI 与程序运行中获得的数据结合起来,放

入公式中进行计算。动态功耗由以下 3 部分构成:Cache 访问功耗、Cache 未命中访问主存功耗和 Cache 未命中 CPU 等待数据产生的功耗。

总体功耗计算公式为:

$$\text{总功耗} = \text{Cache 的访问功耗}(\text{Cache-energy}) * \text{访问次数}(\text{access-times}) + \text{访问主存产生的功耗}(\text{DRAM-energy}) * \text{Cache 丢失次数}(\text{miss-times}) + \text{CPU 等待功耗}(\text{CPU-energy}) * \text{Cache 丢失次数}(\text{miss-times}) + \text{访问主存产生的功耗}(\text{DRAM-energy}) * \text{Cache 写回次数}(\text{write-back-times})$$

在 Cache 访问过程中,由于 Cache 的动态变化,Cache 的功耗并不相同,但是在 Cache 的结构体固定之后,相应的变化结构分为十几种类型,把它们分别利用模拟器产生的参数进行统计。根据模拟器的参数,得出多核动态可重构 Cache 的功耗,比较重构的体系结构和传统的体系结构的功耗差别。

3.1 Cache 功耗

在多核处理器计算功耗的工具中,CACTI 一直是公认的有效计算 Cache 功耗的评估模型,CACTI 也在不断发展,现在已经是 6.5 版本。CACTI 是在当前各类对 Cache 的功耗研究工作中被广泛地提到或者使用的工具,是集成了高速缓存和主存的访问时间、周期时间、面积、静态泄漏和动态功耗的评估模型^[11]。将这些模型整合在一起,用户可以在时间、功耗和面积之间权衡最优化的解决方案。CACTI 可以被系统设计师用来更好地了解内存系统结构,并对系统性能做出评估和综合考虑。

通过 CACTI 对 Cache 进行功耗计算,在本文中 Cache 的路变化范围为:1 路,2 路,4 路,8 路。Cache 的容量变化范围为 256k,512k,1024k,2048k。利用这些 Cache 结构,在 CACTI 中修改对应的参数,可以获得每种 Cache 结构的能量消耗。

3.1.1 主存功耗

主存的功耗计算如下:

$$\text{内存周期时间}(\text{DRAM-cycle-time}) = 1 * 2 / \text{内存频率}(\text{DRAM-frequency}) = 2 / 333\text{MHz} = 6\text{ns}$$

$$\text{内存延迟}(\text{DRAM-Latency}) = \text{内存周期时间}(\text{DRAM-cycle-time}) * (\text{CL} + 1) = 36\text{ns}$$

$$\text{内存功耗}(\text{DRAM-Energy}) = (36 + 6)\text{ns} * 1.339\text{J/s} = 56.238\text{nJ}$$

需要注意的是写回时同样需要消耗主存的功耗。

3.1.2 处理器功耗

处理器的功耗计算如下:

$$\text{CPU 停顿功耗}(\text{Energy-cpu}) = (\text{nACTRD} + 1) * 6\text{ns} * 65\text{mW} = 5.070\text{nJ}$$

nACTRD 是指从主存中读入块需要等待的时钟周期,主存读取操作需要等待数据传输和 8 个时钟周期来将指令传送给 DRAM。因此,遇到突发的读取操作需要 12 个时钟周期,再加上 Cache 控制的一个时钟周期,CPU 共需等待 13 个时钟周期。

4 实验设计及分析

4.1 实验配置

实验采用 simics 模拟系统进行仿真,多核处理器中的每个核私有一级指令和数据 Cache,共享二级 Cache。实验中共采用 4 核和 8 核多核处理器系统。Cache 初始设置为 8 路组

相联地址映射方式,一级 Cache 大小均为 64b,共享二级 Cache 为 2Mb,替换策略为 LRU。

具体配置如表 1 所列。

表 1 4-core CMP 配置

Processor	4cores,4issues,out-of-order
L1 Cache	F-cache:32KB,64byte line size,4-way LRU
Private F-cache and d-cache	D-cache:32KB,64byte line size,4-way LRU
L2 Cache—Shared	2MB,64byte line size,8 way LRU

本文设计的 Cache 结构对于数据的集中性要求较高,如果数据过于分散,会影响系统的优化效果,在程序的选用中也考虑到了这一点。选用 SpecOMP 基准测试程序中的 10 个测试程序,分别为: ammp, swim, mgrid, applu, wupwise, art, apsi, equake, frma3d, galgel, 其中不同的程序对于访存的敏感度并不相同[12]。

4.2 实验结果

本文在多核环境下对共享 L2Cache 结构进行改变,相比传统 Cache 进行动态重构,并且通过实验模拟 Cache 的工作过程,得出需要进行功耗计算的关键数据。总体硬件系统功耗增长时,每个路都有了一个新的开关和寄存器,Cache 中的每个块都有寄存器和开关,路与块的比例微乎其微,因此硬件相对于传统 L2Cache 导致的额外开销在此忽略不计。

在分别用传统 Cache 和动态可重构 Cache 执行完 Art 程序后,得到程序运行参数,包括 L1 和 L2 的访存数和命中数。总的指令数设置为 200000000,当程序运行完相应指令数后会停止,由于指令延迟,程序会相应多运行一些指令,数量比例在千分之二左右。表 2 列出 art 程序的结果。

表 2 Art 程序中的 Cache 访问命中和丢失数据

Cache	Art run with ruby original	Art run with ruby after change
Total number access of L1 Cache	200214079	200131068
Total miss of L1 Cache	1878401	2008978
Total number access of L2 Cache	1809754	2023504
Total miss of L2 Cache	1124677	726160

功耗计算采用如下公式:

总功耗 = Cache 的访问功耗 (Cache-energy) * 访问次数 (access-times) + 访问主存产生的功耗 (DRAM-energy) * Cache 丢失次数 (miss-times) + CPU 等待功耗 (CPU-energy) * Cache 丢失次数 (miss-times) + 访问主存产生的功耗 (DRAM-energy) * Cache 写回次数 (write-back-times)

通过分别计算每个测试程序的功耗,得到 4 核系统结构下,10 个测试程序的功耗计算结果,如图 1 所示。其中蓝色为测试程序在原始 Cache 下的运行功耗,红色为本实验更改优化后的 Cache 结构下的运行功耗。

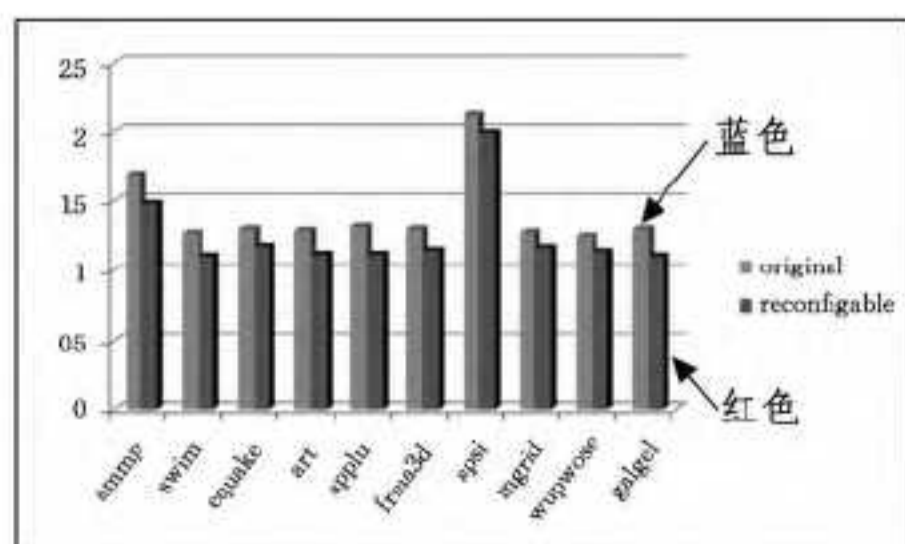


图 1 4 核功耗降低对比图

通过实验得出的数据,发现动态可重构 Cache 确实能够降低系统功耗,4 核条件下访问 L2 的系统功耗能够得到平均 10.7% 的降低,最高的为 applu,能够降低 14.9% 的功耗,最差的为 apsi,也有 5.8% 的功耗优化。

8 核处理器中,10 个测试程序访问 L2 的平均功耗能够得到 11.2% 的降低,最高的为 frma3d,能够降低 17.4% 的功耗,最低的为 mgrid,功耗也有 7.8% 的降低,可以看出 8 核处理器条件下功耗的节省稍微有所提高,但效果并不十分显著。对于测试程序的访存敏感性,同样符合实验预期,如图 2 所示。

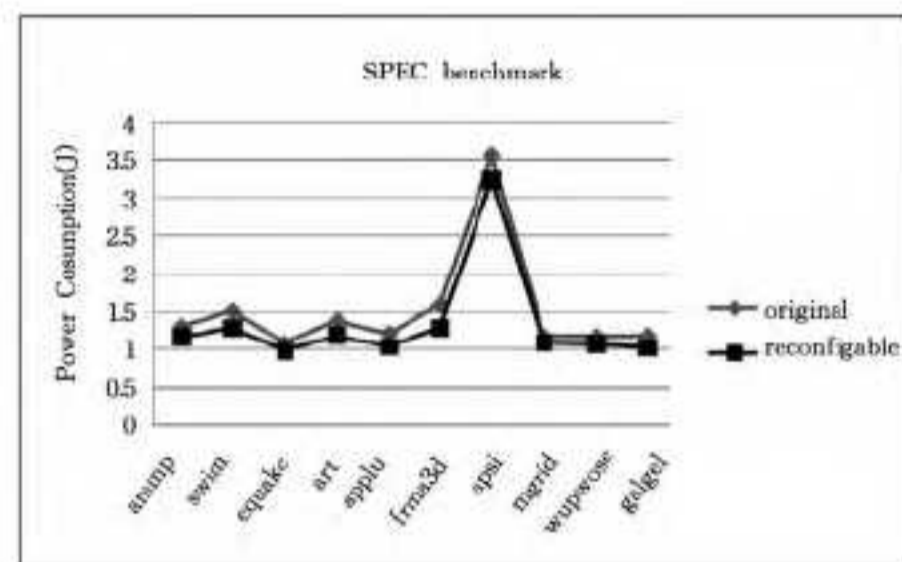


图 2 8 核功耗降低对比图

通过前文提到的 Cache 访存敏感程序表,可以看出当程序为访存敏感时,Cache 功耗可以得到较大降低,当程序为非 Cache 敏感时,Cache 结构调整可能对系统性能造成较大的影响,当 Cache 容量较小时,系统的性能会有所下降,而功耗的节省效果也会有所下降。

结束语 本文使用 Simics 多核模拟器为基础实验环境,通过实验获取功耗计算需要的数据参数,将可重构实验设计进行实现,得到实验数据。针对实验数据进行分析,验证了功耗计算模型,证明了动态可重构 Cache 可以降低功耗。同时,通过数据分析也可以发现功耗的降低不仅仅是动态可重构 Cache 贡献的,而是由系统综合产生的,因此在低功耗设计中,要综合考虑整体系统的功耗和性能,避免片面地考虑 Cache 结构而导致整体功耗的提高。

参考文献

- [1] Xu Zhi-wen, Guo Xiao-xin, Che Xiang-jiu, et al. The Batch Patching Method Using Dynamic Cache of Proxy Cache for Streaming Media[M]. 2550; 5532-5537
- [2] Wang Wei-xun, Mishra P, Ranka S. Dynamic Cache Reconfiguration and Partitioning for Energy Optimization in Real-Time Multi-Core Systems [C] // Design Automation Conference (DAC). 2011; 948-953
- [3] Alouani I, Niar S, Kurdahi F, et al. Parity-based mono-Copy Cache for low power consumption and high reliability[C] // 2012 23rd IEEE International Symposium on Rapid System Prototyping. Tampere, 2012; 44-48
- [4] Chung C-M, Kim J. Low-power L2 Cache design for multi-core processors[J]. Electronics Letters, 2010, 46(9): 618-620
- [5] Juan Fang, Jiang Pu. Fairness of shared cache dynamic partitioning in CMP[J]. International Journal of Advancements in Computing Technology, 2012, 4(1): 44-51
- [6] Zhang Chun, Lin An, Zhong Yan, et al. Performance optimization of EMU real-time information system based on cache technology[J]. International Journal of Advancements in Computing Technology, 2012, 4(12): 351-358

[7] Magnusson P S, Christensson M, Eskilson J, et al. Simics: A full system simulation platform[J]. Computer, 2002, 35(2): 50-58

[8] Dropsho S, Buyuktosunoglu A, Balasubramonian R, et al. Integrating Adaptive On-Chip Storage Structures for Reduced Dynamic Power[C]// Int. Conf. on Parallel Architectures and Compilation Techniques, 2002; 141-152

[9] Seyedi A, Armejach A, Cristal A, et al. Novel SRAM bias control circuits for a low power L1 data cache[C]// NORCHIP, 2012; 1-6

[10] Ahmed R E. Energy-Aware Cache Coherence Protocol for Chip-Multiprocessors[C] // Electrical and Computer Engineering, 2006; 82-85

[11] Martin M M K, et al. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset[C] // Comput Archit News, 2005; 92-99

[12] Merino J, Puente V, Gregorio J A. ESP-NUCA: A low-cost adaptive Non-Uniform Cache Architecture[C]// Digital Object Identifier, 2012; 1-10

(上接第 93 页)

持向量机预测模型的误差集中在 $-2\% \sim 2\%$, 只有少数几个时刻的预测误差超出了上述范围。综上所述, 可以得出, 遗传算法优化的支持向量机预测模型是一个较好的预测模型。

表 1 不同模型预测误差

时间	BP 网络	网格搜索 SVM	GA 优化 SVM
1:00	-0.04564	-0.00926	-0.00933
2:00	0.025376	-0.01128	-0.01045
3:00	-0.00481	0.068517	0.008078
4:00	0.011247	0.006784	-0.01978
5:00	0.020977	-0.01017	-0.0184
6:00	-0.10938	0.018498	0.047694
7:00	0.03598	0.016409	-0.00267
8:00	0.02058	0.063782	0.004284
9:00	0.034773	-0.01973	-0.03155
10:00	-0.02022	-0.01946	-0.01807
11:00	-0.02114	-0.02237	-0.00154
12:00	0.04302	0.014751	-0.01171
13:00	-0.03949	-0.02391	0.009531
14:00	0.041753	-0.01363	-0.00626
15:00	0.011474	0.000754	0.008032
16:00	0.0308	0.03925	0.016115
17:00	0.010898	-0.02791	0.02619
18:00	-0.01345	-0.01175	-0.00986
19:00	0.018411	-0.01082	-0.02473
20:00	-0.10802	-0.02632	-0.00862
21:00	0.017756	-0.02032	-0.00359
22:00	0.044645	-0.01133	-0.01882
23:00	-0.00808	-0.0238	-0.00015
24:00	0.058198	-0.00484	-0.02198
MAPE	79.61	49.56	33.74

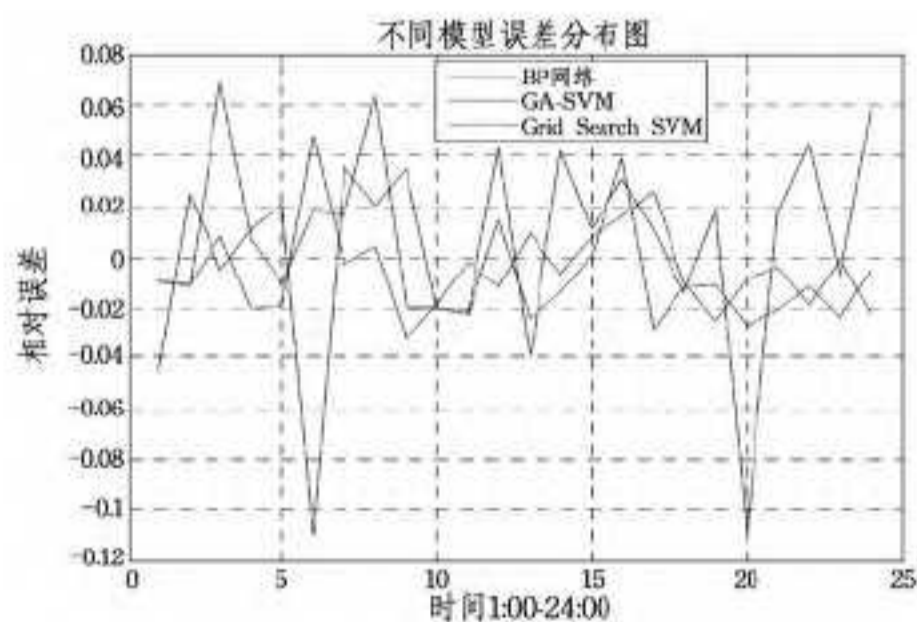


图 4 不同模型相对误差分布

结束语 由于支持向量机可以将原空间的非线性问题映射到一个高维空间而转化为一个线性问题, 因此实际上它是一个二次规划问题, 所求的解是全局最优解。它在小样本、非线性的回归问题上有着较好的优势。

采用径向基核函数的支持向量机, 其误差惩罚项和核参数直接影响着回归预测模型的计算速度和泛化能力。本文提出了利用改进遗传算法来优化支持向量机的参数, 其具有收敛速度快、得到的解是所求的最优解等特点。实例证明, 用遗传算法来优化支持向量机回归预测模型, 相对于 BP 网络和网格搜索寻优, 具有较好的预测性能和稳定性、鲁棒性。

参考文献

[1] 康重庆, 夏清, 刘楠. 电力系统负荷预测[M]. 北京: 中国电力出版社, 2007

[2] 庞清乐. 基于粗糙集理论的神经网络预测算法及其在短期负荷预测中的应用[J]. 电网技术, 2010, 34(12): 168-173

[3] Cai Y, Wang J Z, Tang Y, et al. An efficient approach for electric load forecasting using distributed ART (adaptive resonance theory) & HSARTMAP (hyper-spherical ARTMAP network) neural network[J]. Energy, 2011, 36

[4] 马文晓, 白晓民, 沐连顺. 基于人工神经网络和模糊推理的短期负荷预测方法[J]. 电网技术, 2003, 27(5): 29-32

[5] Rahman S, Bhatnagar R. An expert system based algorithm for short term load forecast[J]. IEEE Transactions on Power Systems, 1998, 3(2)

[6] Hong W C. Application of chaotic ant swarm optimization in electric load forecasting[J]. Energy Policy, 2010, 38(10)

[7] Al-Hamadi H M, Soliman S A. Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model[J]. Electric Power Systems Research, 2004, 68(1)

[8] Huang S J, Shih K R. Short-term load forecasting via ARMA model identification including nongaussian process considerations[J]. IEEE Transactions on Power Systems, 2003, 18(2): 673

[9] Keerthi S, Gibert E. Convergence of a generalized SMO algorithm for SVM classifier design[J]. Machine Learning, 2002, 46(3): 351-360

[10] 雷英杰, 张善文, 李续武, 等. MATLAB 遗传算工具箱及其应用[M]. 西安: 西安电子科技大学出版社, 2005

[11] Reis A J R, Silva A P A. Feature extraction via multi-resolution analysis for short-term load forecasting[J]. IEEE Trans on Power Systems, 2006, 21(1): 392-401

[12] 赵杰辉, 葛少云, 刘自发. 基于主成分分析的径向基函数神经网络在电力系统负荷预测中的应用[J]. 电网技术, 2004, 28(5): 35-40