

基于神经网络的独立程序在单机上运行功耗的预测

谭一鸣 曾国荪

(同济大学计算机科学与技术系 上海 200092)

(国家高性能计算机工程技术中心同济分中心 上海 200092)

摘要 程序运行能耗分析是目前绿色计算、高效能计算的研究热点。由于应用程序自身的复杂性、硬件平台的异构性、环境因素对功耗影响的不确定性,导致很难直接给出程序运行功耗的预测公式,因此提出一种基于神经网络的程序运行功耗预测方法。通过构造一个 BP 神经网络,以影响程序运行功耗的软件、硬件和环境因素为神经网络的输入,以程序运行功耗和执行时间为神经网络的输出,并选取多个应用领域的典型实用程序来提取相应特征参数,且将其作为训练样本来训练神经网络,从而得到程序功耗预测模型。重复实验表明,提出的功耗预测方法具有合理性和可行性。

关键词 绿色计算,功耗分析,神经网络,可执行程序

中图分类号 TP393 文献标识码 A

Power Analysis for Executable Program on Single Computer Based on Artificial Neural Network

TAN Yi-ming ZENG Guo-sun

(Department of Computer Science and Technology, Tongji University, Shanghai 200092, China)

(Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai 200092, China)

Abstract Power management of application program is a hot research in the area of green computing and high productivity computing. Because of the complexity of application programs, heterogeneity of processors and uncertainty of the running environment, it is difficult to propose an accurate method to predict energy consumption for application program directly. So we presented a power analysis paradigm for application program based on artificial neural network. First, we built a power analysis model based on back propagation neural network(BPNN). The three factors of software, hardware and environment were taken as the inputs of BPNN, and energy consumption and finish time as the outputs of BPNN. Next, we chose lots of classic application programs from different fields as training samples. After learning and training, an expected BPNN was obtained which can be used to predict energy consumption for other new programs. Repeated experiments show that this power analysis paradigm is rational and feasible.

Keywords Green computing, Power analysis, Artificial neural network, Executable programs

1 引言

随着人们对科学、工程计算需求的增长,大规模计算环境的高功耗给人们带来了巨大经济负担。因此,计算机的高功耗成了目前亟待解决的问题,低功耗设计也就成了研究热点^[1]。低功耗设计和优化策略早期主要侧重在硬件角度^[2],根据硬件系统设计阶段的不同,可以分为工艺级、版图级、门级、RTL级和系统结构级。目前,基于软件角度的低功耗设计、运行、管理成为新的研究热点。近几年,国内外的学者对程序运行功耗的分析、预测技术进行了大量研究,主要是以指令、函数、算法、任务等为对象来开展功耗研究。

1994年,普利斯顿大学的 Tiwari^[3]首次从软件角度提出

了指令级功耗模型,该模型把程序运行功耗的来源分成3部分:单指令功耗、指令间功耗和资源限制引起的功耗。以测量电流的方法确定每一部分的功耗值并建立功耗数据库,根据此功耗数据库,可预测应用程序的运行功耗。该方法的缺点是需要测量指令集中的每一条指令和每一种指令组合的功耗,如果指令集中有 N 条指令,则至少需要测量 $N+N(N-1)$ 次,并且每次测量都需要编写专门的测试程序,因此工作量巨大。2000年,加州大学洛杉矶分校的 Qu^[4]提出了函数级功耗模型,该模型把程序运行功耗的来源分成4部分:库函数、自定义函数、main函数和其它代码。通过测量方法确定所有常用库函数的功耗值,其它3部分的功耗值通过文献[3]提出的指令级功耗模型的方法得到,并建立功耗数据库。根

到稿日期:2011-06-17 返修日期:2011-11-09 本文受 863 项目(2009AA012201),973 计划课题(2007CB316502),国家自然科学基金项目(90718015),NSFC-微软亚洲研究院联合资助项目(60970155),教育部博士点基金项目(20090072110035),上海市优秀学科带头人计划项目(10XD1404400),高效能服务器和存储技术国家重点实验室开放基金项目(2009HSSA06)资助。

谭一鸣(1982-),男,博士生,主要研究方向为异构计算、绿色计算,E-mail:tan1ming@126.com;曾国荪(1964-),男,博士,教授,博士生导师,主要研究方向为异构计算、信息安全。

据此功耗数据库,可预测应用程序的运行功耗。该方法估计库函数功耗的准确性依赖于对程序运行过程的分析。同时,由于用户自定义函数结构的不确定性,因此对自定义函数功耗测量的可操作性较差。2007年,Lien^[5]提出了服务级功耗模型,通过对流媒体服务器日志文件的分析,建立了功耗与CPU利用率之间的函数关系。通过该功耗模型,可以实时估计流媒体服务器的功耗。但是,该功耗模型只考虑了CPU利用率与功耗间的关系,没有考虑内存、外存和通信部件等因素。另外,该方法并未考虑影响功耗的软件因素,因此该功耗模型的可移植性较差。2009年,东京大学的Ren^[6]提出了算法级功耗模型,该模型以多核处理器为硬件平台,以Bridging Model为计算模型,分析了矩阵相乘算法的计算和通信复杂度与程序运行功耗之间的关系。该方法只分析了程序中的计算操作与运行功耗之间的关系,对其它类型的操作,如输入输出类操作、控制类等操作没有考虑,因此该方法对程序运行功耗预测的准确性较差。2010年,Manuel^[7]开发了一个程序运行功耗的预测工具,该工具把程序运行功耗的来源分成4部分:指令功耗、数据功耗、cache功耗和外设功耗。该工具可自动地为指令集中每一条指令生成一个功耗测量程序,以大大降低功耗测量实验的工作量,并使用线性回归的方法分析指令中操作数对功耗的影响。还提出了时钟驱动和功耗驱动的处理器的电流测量技术,以大大提高功耗预测的准确性。

在国内,2002年,中国科学院计算技术研究所的赵荣彩^[8]针对多线程微体系结构,提出了一种基于频率动态调整的结合细粒度多线程划分的低功耗优化模型,并设计了功耗优化的线程划分算法,其在不影响开发应用程序并行性的同时有效地减少了处理器的运行功耗。2004年,中科院的李佳^[9]提出了一种体系结构级的功耗分析方法,即通过对可配置的功能单元分别进行功耗建模,基于准确的性能模拟可实现对系统级芯片的功耗模拟。该方法可应用于体系结构、指令集、编译器以至软件的低功耗优化中。2005年,国防科技大学的易会战提出了一种基于语法树的实时动态电压调节低功耗算法^[10],该算法基于程序的语法树,根据语句权重阈值的调节点选择策略,自动插入调节点。在实时条件下,根据静态时间分析方法进行实时电压调节,解决了循环和函数调用引入的复杂问题。2006年,中国科学技术大学的雷霆^[11]将性能无损的低能耗电压调度问题形式化为一个混合整数规划模型(MILP),提出了基于剖析结果的PGS算法和基于分析结果的ADS算法,该算法可有效降低软件运行中的能耗。2009年,四川大学的罗刚^[12]在硬件微结构级和电路级分析了程序运行功耗的来源和构成,探讨了影响程序运行功耗的软件特征参数,并从源程序级和算法级两个角度提出了降低程序运行功耗的优化策略。

上述研究有些是以源程序为对象,但如果源程序不开放,则此类方法不可行。另外,大多数研究没有考虑应用程序自身的异构性对功耗的影响。例如,计算密集型应用程序和存储密集型应用程序在同一单机上运行功耗一般不同。也有些功耗研究只是针对计算机系统内的计算部件,如CPU、DSP的功耗进行分析,没有考虑其它硬件部件,例如内存、硬盘、通信设备和其它外设所产生的功耗。同时,也没有考虑处理部件的异构性对功耗的影响,例如不同的处理器类型或频率、不

同的内存大小对功耗有着重要影响。本文针对任意可执行程序在任意计算机节点上运行的情况,提出一种基于BP神经网络的独立程序在单机上运行功耗的预测方法。

2 影响程序运行功耗的因素

程序运行功耗是指应用程序在计算机上运行过程中,指令和数据驱动计算机硬件运转所产生的功耗。实验表明,不同应用程序在同一单机上的运行功耗不同;同一应用程序在不同配置单机上的运行功耗也不同;同一应用程序在同一单机、不同自然环境下的运行功耗也不同。可见,程序运行功耗与应用程序自身的特征、单机的硬件配置和自然环境3方面因素有关。

2.1 程序运行环境假定

通常可从硬件、指令系统、函数、过程、程序段、可执行程序、并行任务等角度和粒度来开展计算机的功耗分析。对于移动或嵌入式电子设备,特别关注低功耗的器件设计。对于现在十分普及的工作站集群、计算中心、数据中心、云计算环境等,因为运行能耗巨大,成本越来越高,迫使人们不得不研究计算任务在并行、异构、云计算环境中的优化调度,从而降低功耗^[13]。本文正是为了这一目的而开展基础研究,为计算任务在云环境中进行能耗感知的优化调度提供理论依据。本文研究和讨论的软硬件对象前提假设如下:

(1)软件实体为单个独立的可执行程序。例如Windows下后缀名为exe的文件,或Unix下文件属性有x的文件,或一个java语言编写的可执行程序等。可执行程序是软件最普遍的存在形式之一,代表了一个独立的计算任务。另外,大部分应用软件的源程序除开发商之外根本无法获得,例如WinZip。因此,选择可执行程序进行功耗分析,具有合理性和可行性。事实上,用户提交计算任务给计算中心,并请求服务,一般是可执行程序,很少是计算任务的源程序。

(2)硬件平台为单个计算机节点。例如可以是IBM PC兼容机、SUN Ultra工作站、笔记本电脑、掌上电脑等。单个计算机节点(简称单机)是使用最广泛的计算机系统形式之一。2010年,单机的世界保有量已接近每3人一台。另外,单机是组成工作站集群、并行多处理机系统、云计算平台的基本组成部分。因此,选择单机为硬件平台进行功耗分析,具有研究和实用价值。

2.2 影响功耗的软件因素

实验表明,不同应用程序在相同单机、自然环境下的运行功耗不同,可见程序运行功耗受程序自身各种特征的影响。根据对功耗的影响程度,程序运行功耗的主要软件因素如下:

(1)可执行程序的大小。其一定程度上反映了应用程序的规模,对功耗有一定影响,用 P_{size} 来表示。

(2)汇编代码行数。其是指对可执行程序进行反汇编后得到的汇编代码行数。代码行数从另一角度反映了程序的运行规模,对功耗有一定影响,用 L_{code} 来表示。

(3)输入数据量。其是指应用程序要进行处理的数据大小。输入数据量反映了计算任务的规模,例如用WinZip对1GB文件进行压缩,则WinZip的输入数据量为1GB。输入数据量对程序运行功耗具有重要的影响,用 D_{size} 来表示。

(4)指令操作次数。其是指应用程序在运行过程中各种

指令实际执行的次数。按照指令的功能、操作所涉及的硬件部件和对功耗的影响程度,将指令分成4类:传送类指令、运算类指令、I/O类指令和控制类指令,各自的操作次数分别用 E_{MOV} 、 E_{OPER} 、 $E_{I/O}$ 和 E_{INT} 表示,则应用程序中各种指令的总操作次数 $E_{total} = E_{MOV} + E_{OPER} + E_{I/O} + E_{INT}$ 。

2.3 影响功耗的硬件因素

实验表明,同一应用程序在不同单机、相同环境下的运行功耗不同,可见程序运行功耗受计算机的各种硬件特征影响。根据计算机体系结构中各种硬件单元对功耗的影响程度,程序运行功耗的主要硬件因素如下:

(1)处理器核心个数。在一定面积和工艺尺寸条件下,处理核心的个数反映了处理器线程级并行(TLP)的能力,而TLP的大小直接影响了处理器的功耗;处理核心的个数还直接影响了片上互连网络的复杂度,而片上互连网络也是影响处理器功耗的重要因素。另外,片上缓存大小一定的条件下,处理核心越多,片上缓存的访问缺失率越大,从而导致片外存储访问的次数增加,而片外存储的功耗是片上存储的35倍^[14]。可见,处理核心的个数是影响功耗的重要因素。处理核心的个数用 N_{core} 来表示。

(2)CPU频率。根据CMOS电路的基本功耗模型^[15] $P = 0.5C_L V_{dd}^2 AF$ 可知,功耗 P 与CPU频率 F 成正比,即CPU频率直接影响电路功耗。CPU频率用 F_{CPU} 来表示。

(3)内存大小。随着存储层次级别的降低,存取时间也随之显著增加,同时存取功耗也大幅提高。内存的大小决定了内存命中率或是缺失率的大小,当内存越小时,内存缺失率就越高,访问外部存储的几率也越高,功耗就会越大。内存大小用 M_{size} 来表示。

(4)硬盘转速。其是指硬盘主轴电机带动磁盘盘片单位时间内的转数。硬盘转速越大,硬盘主轴电机单位时间做功越多,消耗的电量也越多。硬盘转速用 D_{spd} 来表示。

(5)光驱数据传输率。其是指光驱的激光头读取光盘数据的快慢程度。数据传输率越大,则光驱主轴电机的转速越快,单位时间做功越多,消耗的电量也越多。光驱数据传输率用 R_{spd} 来表示。

(6)网络带宽。其是指在单位时间内网络传输数据量的多少。网络带宽越大,则单位时间内通信设备处理的数据量越大。网络带宽用 B 来表示。

2.4 影响功耗的环境因素

物理上,温度是影响半导体电阻大小的重要因素。当计算机的工作电压一定时,半导体电阻变化会直接导致工作电流的变化。电流大小的变化又会导致单机的功率变化。温度用 T_{emp} 表示。空气湿度是影响半导体电阻的另一重要因素。相对湿度用 H_{umi} 表示。

2.5 程序运行功耗预测问题

由上述分析可知,程序运行功耗受软件、硬件和环境的影响。因此,程序运行功耗可看成是软件、硬件和环境的函数,即 $Energy = f(\text{Software}, \text{Hardware}, \text{Environment})$ 。要预测程序运行功耗,需建立该函数关系。但是由于应用程序自身的复杂性、硬件平台自身的异构性、环境因素对功耗影响的不确定性,使得很难直接给出程序运行功耗与软件、硬件、环境之间的数学公式,因此应使用人工神经网络的方法来建立程

序运行功耗与软件、硬件和环境之间的非线性函数关系。

3 计算任务功耗的神经网络模型

人工神经网络是模拟人脑生物过程的人工智能技术。由大量的神经元互连形成复杂的非线性系统,主要功能有非线性映射、联想记忆、分类和识别、优化计算和知识处理等。其中,BP神经网络具有良好的输入输出映像特性,在多变量函数逼近方面具有很强的优势,能够充分逼近任意复杂的非线性函数关系,非常适用于函数逼近问题的研究^[16]。因此,对于程序运行功耗的预测问题,在很难直接建立功耗数学公式的情况下,本文使用BP神经网络的方法分析和预测应用程序运行的功耗。

3.1 功耗预测BP神经网络

误差反向传播(BP)神经网络是目前应用最广泛的神经网络之一,由输入层、隐含层和输出层组成,各层之间实现全连接,如图1所示。

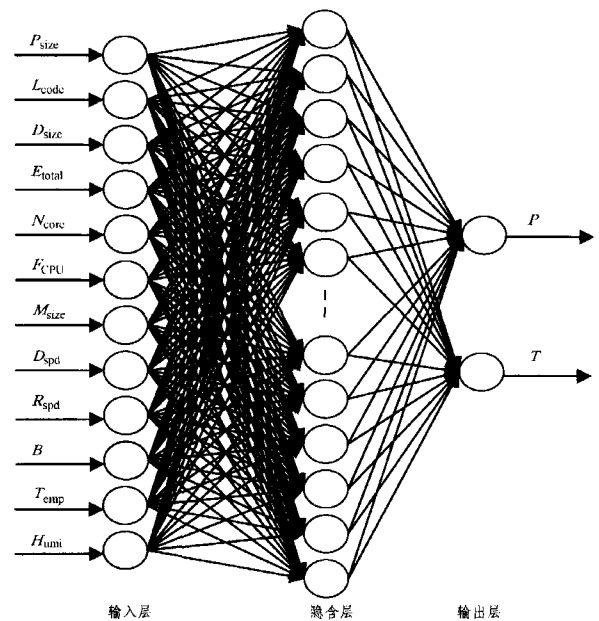


图1 功耗预测BP神经网络模型

其中输入层神经元的个数由影响功耗的因素个数决定,输入向量 $X = (P_{size}, L_{code}, D_{size}, E_{total}, N_{core}, F_{CPU}, M_{size}, D_{spd}, R_{spd}, B, T_{emp}, H_{umi})$,其中 P_{size} 为可执行程序的大小, L_{code} 为汇编代码行数, D_{size} 为输入数据量, E_{total} 为指令操作次数, N_{core} 为处理核心个数, F_{CPU} 为CPU频率, M_{size} 为内存大小, D_{spd} 为硬盘转速, R_{spd} 为光驱数据传输率, B 为网络带宽。隐含层神经元的个数是影响神经网络性能的重要因素,根据Kolmogorov定理^[17],隐含层神经元的个数设定为 $2n+1$,其中 n 表示输入层神经元的个数。因此,本文隐含层神经元的个数设计为25。输出层神经元的个数由要得到的输出个数决定,输出向量 $Y = (P, T)$,其中 P 为程序运行功耗, T 为程序执行时间。网络的输出如式(1)所示:

$$\begin{cases} u_j = f(\sum_{i=1}^n \omega_{ij} x_i - \theta_j), j = 1, 2, \dots, h \\ y_k = g(\sum_{j=1}^h \omega_{jk} u_j - \theta_k), k = 1, 2, \dots, m \end{cases} \quad (1)$$

式中, x_i 表示第 i 个输入层神经元的输入, $1 \leq i \leq n$; u_j 表示第 j 个隐含层神经元的输出, $1 \leq j \leq h$; y_k 表示第 k 个输出层神

经元的输出, $1 \leq k \leq m$; ω_{kj} , θ_k 和 f 分别表示输入层到隐含层的权值、阈值和激活函数; ω_{jk} , θ_k 和 g 分别表示隐含层到输出层的权值、阈值和激活函数。

4 训练样本

神经网络的训练方法采用有指导训练, 训练样本可表示为训练对, 即 (X_i, Y_i) , $i=1, 2, \dots, N$, 其中 X_i 为训练输入向量, Y_i 为训练输出向量。训练的输入向量从选择的应用程序、硬件平台和环境参数中获取, 训练输出向量则通过测量应用程序在单机上的运行功耗和执行时间得到。训练过程包括正向传播和反向传播。正向传播是输入信号 X_i 向前传递, 直到输出端产生预测输出 \hat{Y}_i , 如式(1)所示。反向传播是误差信号 $f(Y_i - \hat{Y}_i)$ 从输出端反向传递, 对权值进行修正。

4.1 软件样本的选取

4.1.1 广泛的应用程序样本

为了使建立的功耗预测神经网络模型有较好的泛化能力, 并且考虑到超级计算中心常常处理各行各业的计算任务, 如航空航天、天文物理、气候气象与环境科学、新材料和新能源等领域的计算任务, 本文选择的应用程序样本如表 1 所列。

表 1 广泛的应用程序样本

序号	应用类型	应用程序领域描述	文件名
1	数据压缩	数据传输, 图像、声音、视频处理	WinZip
2	播放器	游戏娱乐、新闻、广告、视频电话、会议	Smpayer
3	病毒扫描	软件特征码分析, 软件执行特征分析	AntiSpyware
4	网络下载	网络文件传输、本地资源共享	Flashget
5	图像分析	数字图像处理, 人脸、指纹识别	ImageComparer
6	编译器	计算机 C 语言程序设计、编译、调试	Gcc
7	加密解密	数据保护, 网络传输的数据安全等	Cffiles
8	N 体模拟	天体运行轨迹建模, 火焰模拟, 宇宙探索	Nbody
9	气候模拟	天气预报, 人类活动对气候影响估计等	Cs
10	分子模拟	微孔、表面、纳米等微结构新材料的研究	Chem
...

4.1.2 应用程序特征的提取

利用样本数据训练神经网络, 首先需要分别从 4.1.1 节选取的多个应用程序样本中, 提取出软件参数 (P_{size} , L_{code} , D_{size} , E_{total})。提取各种参数的方法如下:

(1) 可执行程序大小的提取。只要在操作系统下查看文件的属性即可获得。例如, WinZip V14.5 在 Windows XP 32 位操作系统下的大小是 7713608 字节。

(2) 汇编代码行数的提取。可执行程序的汇编代码可通过反汇编软件 W32DASM 来得到。反汇编得到的全部代码中, 除了汇编代码还包含有对软件的解释性代码。因此, 统计汇编代码执行行数时, 需删除所有解释性代码和空行。例如, 对 WinZip V14.5 反汇编得到的总代码行数为 871166 行, 其中汇编代码为 629562 行。

(3) 输入数据量的获取。对数据进行处理的应用程序, 如数据压缩程序 WinZip 的输入数据量可通过查看要进行处理的数据的属性来获得。对于应用程序 Nbody, 用粒子的数量来表示输入数据量, 例如 100000 个粒子。对于编译器 GCC, 输入的数据量可用要进行编译的源程序代码的执行行数来表示。

(4) 指令操作次数的提取。指令的执行次数与程序结构密切相关, 例如循环程序段中的指令操作次数受循环次数的影响, 但是某些程序段的循环次数很难预测; 同样, 分支程序

段的执行路径也很难事先确定, 导致对汇编程序中各种指令操作次数很难准确统计。因此, 本文采用度量程序复杂度的 Halstead^[18]方法来近似预测指令的操作次数。计算公式为: $H = n_1 \log_2 n_1 + n_2 \log_2 n_2$, 其中 n_1 为程序中不同运算符的个数, n_2 为不同操作数的个数, H 为预测的程序复杂度, 并且认为 $H \approx E_{total}$ 。

4.2 单个计算机硬件特征的获取

训练输入样本中的硬件参数 (N_{core} , F_{CPU} , M_{size} , D_{spd} , R_{spd} , B) 同样需要提取, 如处理器中处理核心的个数 N_{core} 、CPU 主频 F_{CPU} 和主存大小 M_{size} 、硬盘转速 D_{spd} 和光驱数据传输率 R_{spd} 可从计算机的说明书中得到, 或者可以通过计算机硬件参数检测软件 EVEREST Ultimate Edition 获得。网络带宽 B 可通过检测软件 Bandwidth Monitor 获得。

4.3 样本功耗数据实验采集

训练输出样本参数 (P , T) 可以通过实验方法获得。通过将 4.1.1 节选取的所有应用程序样本在不同的单机上运行, 来测量其运行功耗和执行时间。实验环境如下。

4.3.1 实验环境

为了便于研究, 所有测量实验在相同自然环境下进行, 室温 and 相对湿度通过空调分别固定在 26℃ 和 37%。单机选择 DELL OPTIPLEX 330 (简称 DELL330) 和 DELL OPTIPLEX 320 (简称 DELL320) 系列的两台 PC 作为实验平台。为了彰显两台 PC 的异构性, 增加 DELL320 的内存为 2GB, 并把显卡更换为 ATI Radeon HD 5800。其中, DELL330 的闲置功耗为 70.16W, DELL320 的闲置功率为 138.17W。两个单机的操作系统都为 Windows XP。功耗测量仪器采用北电微型电力监测仪、PowerBay A 款系列的专业版智能插座表, 该款电表能够记录电器的累计电量和运行时间, 并能实时显示有效功率、功率因子。功率范围为 5~2200W, 功率精度误差值在 $\pm 0.01 \sim 0.1W$ 范围内, 因此适用于测量单机功耗。测量实验过程: 首先将电力监测仪插在电源上, 再将 PC 的主机插在电力监测仪上, 并启动 PC。然后等 PC 运行稳定, 即电表显示的功率波动范围在 $\pm 1W$ 时, 将电表的累计电量和累计时间清零。其次运行一个应用程序, 对事先准备的数据集进行处理。程序运行结束后, 记录累计电量和累计时间。按上述步骤, 反复对该应用程序进行测量。多次测量的电量和时间的平均值即为该应用程序的运行功耗和执行时间值。测量实验的示意图如图 2 所示。

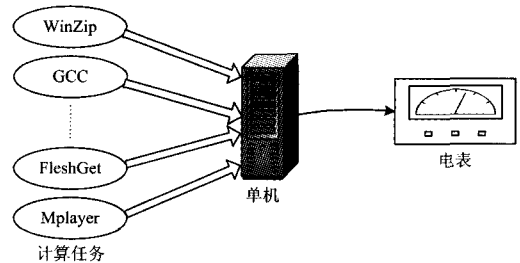


图 2 程序运行功耗的测量示意图

4.3.2 训练样本数据

由 4.1 节可得到所有训练输入样本中的软件参数值, 由 4.2 节可得到所有训练输入样本中的硬件参数, 训练输入样本中的环境参数设定为固定值, 由 4.3.1 节的方法可得到所有训练输出样本参数值。所有训练样本数据如表 2 所列。

表2 全体训练样本数据

训练样本	训练输入										训练输出			
	应用程序参数				计算机硬件参数				环境参数		功耗	时间		
	Psize (B)	Lcode (行)	Dsize (MB)	Etotal (次)	Ncore (个)	FCPU (GHZ)	Msize (GB)	Dspd (转/分)	Rspd (B/S)	B (MB/S)			Temp (℃)	Humi (%)
WinZip	7713608	629562	169264.33	19143581.54	2	1.8	1	7200	21600	100			2074590.76	22258
Smplayer	3222528	634705	3997.71	23351777.1	2	1.8	1	7200	21600	100			3124391.24	36724
AntiSpyware	2424560	434042	44892.57	13631022.72	2	1.8	1	7200	21600	100			2553947.05	28845
Flashget	2467376	691897	529.39	21602864.14	2	1.8	1	7200	21600	100			1653757.52	14456
Ajc	909824	207733	94518.43	5700936.02	2	1.8	1	7200	21600	100	26	37	2615610.89	32437
Gcc	1295374	73444	5364	2292025.03	1	3	2	7200	21600	100			3257902.22	21682
Cffiles	884224	108725	76541.25	2659971.57	1	3	2	7200	21600	100			4519736.86	25274
Nbody	654216	547315	100000	2356781.45	1	3	2	7200	21600	100			6601417.34	36795
Cs	5647842	604765	54236	16478324.27	1	3	2	7200	21600	100			4747903.50	28878
Chem.	2064384	563314	24561	54066474.99	1	3	2	7200	21600	100			5403374.97	29451

5 神经网络的训练和功耗预测

5.1 神经网络的训练

在BP神经网络中,由于训练输出,即测量输出Y与由式(1)计算的预测输出 \hat{Y} 存在误差,因此需要通过信号正向传播与误差逆向传播修改权值,并且修改过程是反复进行的。对权值的修改过程就是对网络的训练,此过程一直进行到网络输出的误差逐渐减少到可接受的程度或达到设定的学习次数为止。功耗预测神经网络的训练过程通过MATLAB提供的神经网络工具箱进行模拟。将网络各层的权值初始化为(-1,1)之间的随机数,训练精度 $\epsilon=0.01$,最大训练次数 $t=50000$,学习率 $\eta=0.05$,训练方式采用批量训练方式,隐含层

和输出层的激活函数分别采用单极sigmoid函数和线性函数。为了降低神经网络对于误差曲面局部细节的敏感性,抑制网络限于局部极小,在调整权值时加入动量项以改善收敛性^[19],公式为 $\omega(t+1)=\omega(t)-(1-\alpha)\eta\frac{\partial E}{\partial \omega(t)}+\alpha\Delta\omega(t)$,其中 α 为动量系数,且 $\alpha=0.95$ 。

5.2 功耗预测和验证

为了验证本文提出的方法,以及构建的功耗预测神经网络模型的泛化能力,选取音乐播放软件Winamp和文件压缩软件WinRAR作为应用程序验证测试样本。选取的单机分别为DELL330、DELL320、ThinkPad SL410(简称SL410)笔记本,进行6组不同的实验。所有测试输入样本数据如表3所列。

表3 验证测试输入样本数据

测试样本	测试输入												
	应用程序参数				计算机硬件参数				环境参数				
	Psize (B)	Lcode (行)	Dsize (MB)	Etotal (次)	Ncore (个)	FCPU (GHZ)	Msize (GB)	Dspd (转/分)	Rspd (B/S)	B (MB/S)	Temp (℃)	Humi (%)	
Winamp	1596032	97544	668.64	2052457.30	2	1.8	1	7200	21600	100	26	37	
					1	3	2	7200	21600	100			
Winrar	5754880	197458	677057.31	3838228.53	2	2.3	1	5400	21600	100	26	37	
					2	1.8	1	7200	21600	100			
					1	3	2	7200	21600	100			
					2	2.3	1	5400	21600	100			

通过将表3的测试数据输入训练好的功耗预测神经网络,可得到Winamp和WinRAR预测的运行功耗和执行时间,测试多次并取平均值作为其最后的预测值。另外,将应用程序Winamp和WinRAR分别在DELL330、DELL320和SL410上运行,记录各自实际运行功耗和执行时间,同样测量多次取平均值作为最后的测量值。运行功耗和执行时间的预测值与实测值的比较如图3和图4所示。

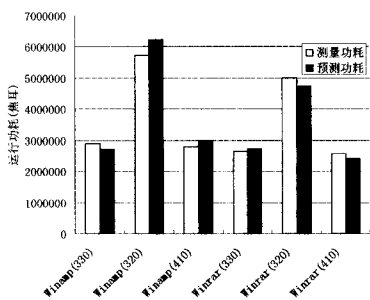


图3 预测运行功耗与实测运行功耗的比较

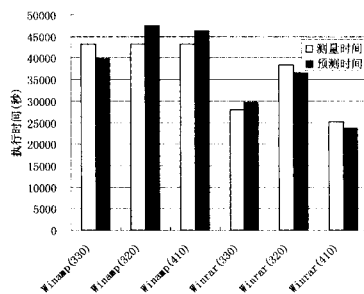


图4 预测执行时间与实测执行时间的比较

从上述6组数据可知,预测与实测运行功耗的最大绝对误差为484636.35(J),平均绝对误差为303216.19(J),最大相对误差为8.46%,平均相对误差为6.54%。预测与测量执行时间的最大绝对误差为3960(s),平均绝对误差为2810(s),最大相对误差为9.69%,平均相对误差为7.78%。从预测结果和误差来看,本文建立的功耗预测神经网络具有较好

(下转第294页)

[5] 闻立杰,王建民,孙家广.用着色 Petri 网建模 workflow 模式[J]. 计算机科学,2006,33(6):135-139

[6] 柏晓莉,姜军,罗雪山,等.基于 UML&-OPN 的 C4ISR 系统体系结构仿真验证研究[J]. 系统工程与电子技术,2008,30(4):671-676

[7] 杨璐,柳溪,王林章,等.面向基于场景规约的 Web 服务消息流分析与验证[J]. 计算机学报,2009,32(9):1759-1772

[8] Peterson J L. Petri Nets Theory and the Modeling of Systems [M]. Eaglewood Cliffs,Prentice Hall,1981

[9] 罗雪山.基于对象 Petri 网的离散事件系统建模仿真环境(OPMSE)[J]. 计算机仿真,2000,17(5):42-44

(上接第 286 页)

的泛化能力。其中,对于应用程序 Winamp,其预测值的绝对误差和相对误差都要略大于应用程序 WinRAR,其原因是 WinRAR 与训练样本中的应用程序 WinZip 属于相同类型的软件,说明了功耗预测神经网络具有良好的联想记忆能力。而对于 Winamp,在训练样本中则无应用程序与其类似,但是预测值的绝对和相对误差依然在较低的水平。实验发现 WinRAR 分别在 DELL330、DELL320 和 SL410 上对相同数据进行压缩处理时,SL410 不仅有最好的性能,功耗也最低,说明 WinRAR 适合在 SL410 上运行。其原因是:虽然 DELL320 的内存和显卡性能优于 SL410,但是 SL410 的 CPU 性能优于 DELL320。并且,WinRAR 的执行特征为计算密集和 I/O 密集型,计算部分完全由 CPU 完成,并不涉及显卡的计算。因此,DELL320 的高性能显卡不仅不能改善 WinRAR 的执行性能,反而大大增加了运行功耗。由此可见,在异构集群环境中,对于不同类型的计算任务需要调度到相匹配的单机上执行,才能有较好的性能和较低的能耗。

结束语 本文以可执行程序为软件实体,以单机为硬件平台,分析了影响程序运行功耗的软件、硬件和环境因素,给出了提取相关参数值的方法。并且,基于 BP 神经网络构建了一个程序运行功耗预测模型,该模型以软件、硬件和环境参数为训练输入,以测量的程序运行功耗和执行时间为训练输出,对功耗预测神经网络进行训练。为使神经网络的训练样本具有代表性,选取了不同应用领域中的典型应用程序。通过学习训练得到的神经网络可以作为程序在计算机上的运行功耗的预测模型。两个验证实验表明了本文提出的功耗预测方法的合理性和可行性。下一步的研究工作是以本文提出的功耗预测方法为基础,深入开展集群、云计算环境中的功耗优化管理研究。

参 考 文 献

[1] 郭兵,沈艳,邵子立.绿色计算的重定义与若干探讨[J]. 计算机学报,2009,32(12):2311-2319

[2] Macii E, Pedram M, Somenzi F. High-level power modeling estimation and optimization[J]. IEEE Transactions on Computer-aided Design of Integrated Circuits and System, 1998, 17(11): 1061-1079

[3] Tiwari V, Malik S, Wolfe A. Power analysis of embedded software: a first step towards software power minimization[J]. IEEE Transactions on VLSI Systems, 1994, 2(4): 437-445

[4] Qu G, Kawabe N, Usami K, et al. Function level power estimation methodology for microprocessors[C]// Proceedings of the Design Automation Conference. 2000: 810-813

[5] Lien C H, Bai Y W, Lin M B. Estimation by software for the energy consumption of streaming-media servers[J]. IEEE Transactions on Instrumentation and Measurement, 2007, 56(5): 1859-1870

[6] Ren D Q, Suda R. Modeling and estimation for the energy consumption of matrix computation on multi-core platform[C]// Proceedings of the International Joint Conference on Computational Sciences and Optimization. 2009: 42-46

[7] Wendt M, Grumer M, Steger C, et al. Tool for automated instruction set characterization for software power estimation[J]. IEEE Transactions on Instrumentation and Measurement, 2010, 59(1): 84-91

[8] 赵荣彩,唐志敏,张兆庆,等.低功耗多线程编译优化技术[J]. 软件学报,2002,13(6):1123-1129

[9] 李佳,徐勇军,李晓维,等.体系结构级功耗分析方法[J]. 系统仿真学报,2004,16(12):2821-2827

[10] 易会战,陈娟,杨学军,等.基于语法树的实时动态电压调节低功耗算法[J]. 软件学报,2005,16(10):1726-1734

[11] 雷霆,李曦,周学海.低功耗软件设计中的性能无损电压调度技术研究[J]. 计算机研究与发展,2006,43(6):1090-1096

[12] 罗刚,郭兵,沈艳,等.源程序级和算法级嵌入式软件功耗特性的分析与优化方法研究[J]. 计算机学报,2009,32(9):1869-1875

[13] Zikos S, Karatza H D. Performance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times [J]. Simulation Modelling Practice and Theory, 2011, 19(1): 239-250

[14] Moreshet T, Bahar R I, Herlihy M. Energy-aware microprocessor synchronization: transactional memory vs. locks[C]// Proceedings of the International Symposium on High-Performance Computer Architecture. 2006: 47-54

[15] Najm F N. Transition density: a new measure of activity in digital circuits[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and System, 1993, 12(2): 310-323

[16] Song J M, Chew W C. Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering [J]. Microwave and Optical Technology Letters, 1999, 10(1): 14-19

[17] Trenn S. Multilayer perceptrons: approximation order and necessary number of hidden units[J]. IEEE Transactions on Neural Networks, 2008, 19(5): 836-844

[18] Howard M. Halstead: Elements of Software Science [M]. Elsevier, 1977

[19] Rumelhart D E, Hinton G E, Williams R J. Learning representations back-propagating errors[J]. Nature, 1986, 323(9): 533-536