

面向云计算数据中心的能耗建模方法*

罗亮¹, 吴文峻², 张飞²

¹(电子科技大学 计算机科学与工程学院, 四川 成都 611731)

²(北京航空航天大学 计算机学院, 北京 100191)

通讯作者: 罗亮, E-mail: luoliang@nlsde.buaa.edu.cn

摘要: 云计算对计算能力的需求,促进了大规模数据中心的飞速发展.与此同时,云计算数据中心产生了巨大的能耗.由于云计算的弹性服务和可扩展性等特性,云计算数据中心的硬件规模近年来极度膨胀,这使得过去分散的能耗问题变成了集中的能耗问题.因此,深入研究云计算数据中心的节能问题具有重要意义.为此,针对云计算数据中心的能耗问题,提出了一种精确度高的能耗模型来预测云计算数据中心单台服务器的能耗状况.精确的能量模型是很多能耗感知资源调度方法的研究基础,在大多数现有的云计算能耗研究中,多采用线性模型来描述能耗和资源利用率之间的关系.然而随着云计算数据中心服务器体系结构的变化,能耗和资源使用率的关系已经难以用简单的线性函数来描述.因此,从处理器性能计数器和系统使用情况入手,结合多元线性回归和非线性回归的数学方法,分析总结了不同参数和方法对服务器能耗建模的影响,提出了适合云计算数据中心基础架构的服务器能耗模型.实验结果表明,该能耗模型在只监控系统使用率的情况下,在系统稳定后,能耗预测精度可达到95%以上.

关键词: 云计算;数据中心;能耗建模;资源调度算法

中图法分类号: TP316

中文引用格式: 罗亮,吴文峻,张飞.面向云计算数据中心的能耗建模方法.软件学报,2014,25(7):1371-1387. <http://www.jos.org.cn/1000-9825/4604.htm>

英文引用格式: Luo L, Wu WJ, Zhang F. Energy modeling based on cloud data center. Ruan Jian Xue Bao/Journal of Software, 2014, 25(7): 1371-1387 (in Chinese). <http://www.jos.org.cn/1000-9825/4604.htm>

Energy Modeling Based on Cloud Data Center

LUO Liang¹, WU Wen-Jun², ZHANG Fei²

¹(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

²(School of Computer Science and Technology, BeiHang University, Beijing 100191, China)

Corresponding author: LUO Liang, E-mail: luoliang@nlsde.buaa.edu.cn

Abstract: Energy efficiency of cloud data centers has received significant attention recently as data centers often consume significant resources in operation. Most of the existing energy-saving algorithms focus on resource consolidation for energy efficiency. Accurate energy consumption model is the basis for these algorithms. This paper proposes an accurate energy model to predict energy consumption of single machine. In most of the existing cloud computing energy studies, linear models are used to describe the relationship between energy consumption and resource utilizations. However, with the changes in computer architecture, the relationship between energy and resource utilizations may not be linear. In fact, this paper explored a variety of regression analysis methods to estimate the energy consumption accurately while using low computational overhead. Initially, multiple linear regression models are used, but they often do not produce good enough results. Afterwards, this paper chooses three non-linear models and finally settled with the polynomial regression with Lasso as it produces the best estimation. Experimental results show that in adoption of energy model presented in this paper, the prediction accuracy can reach more than 95%.

* 基金项目: 国家高技术研究发展计划(863)(2013AA01A210)

收稿时间: 2013-12-30; 修改时间: 2014-03-17; 定稿时间: 2014-05-06

Key words: cloud computing; data center; energy modeling; resource scheduling algorithm

能耗模型(energy model)是云数据中心重要的组成部分.在不间断运转的云数据中心,用户和管理者都需要了解他们的行为是如何影响计算机能耗的,从而采取相应的调节措施,达到优化能效的目的.随着硬件厂商对能耗的支持,很多组件已经支持能耗的实时测量,但是单纯的物理测量手段一方面不能预测未来的能耗需求;另一方面,它们也不能为资源使用情况和能耗之间提供有效的联系,而这两点恰恰是很多能效优化算法的切入点.如今,随着云数据中心规模的不断扩张,数据中心的能耗特点也变得复杂多变,很多研究人员也提出了切合他们实际的模型.然而,针对 SaaS, PaaS 和 IaaS (software/platform/infrastructure as a service) 的能效优化工作差别巨大,不同层次的能耗模型缺乏可移植性.

因此,本文试图从云基础设施层面为云数据中心的服务器建立合理、可靠的能耗模型,并比较不同采样方式和数学方法对能耗模型的影响.首先,着重描述我们所构建的非线性能耗模型的特性;接下来,则在算法与实验上应用本文提出的能耗模型的研究成果,以验证本文所陈述的能耗模型的有效性,并且该能耗模型也有希望应用到其他能耗研究工作中.

1 云数据中心的能耗模型

1.1 能耗模型建立的原则

通常来说,一个准确、高效的能耗模型对于云数据中心的能耗优化工作是非常重要的,虽然现在的硬件中集成了很多能耗模块或与能耗相关的组件,但这只能测量该模块或该节点的实时能耗.这些模块或组件不能预测未来的能耗,也不能把能耗与系统使用率进行有效且必要的联系.因此,我们需要一个准确而高效的能耗模块来为后续的资源调度算法提供有效支撑,能耗模型应该遵循以下原则:

- (1) 全系统.建立的模型应该是系统使用信息与全系统的能耗关联起来,而非仅单独针对某个独立的组件建模;
- (2) 精确.模型针对系统能效优化算法应该有足够的精确度,例如,该模型的平均相对偏差不超过 10%.此外,先期建立的模型越精确,对后期优化工作越有利;
- (3) 快速.能耗模型应尽可能地快速,甚至实时地预测能耗;
- (4) 通用性.建模框架应该可以在不同的系统上通用,例如面对不同的处理器、不同的内存调控技术;
- (5) 弹性.模型应该能够根据云计算环境的规模轻松地加以扩展或收缩;
- (6) 简单.模型应尽可能地简单,在不影响预测准确性和支持节能优化方法的同时,应尽量减少输入参数的个数并降低模型的复杂度.

1.2 能耗建模的基本流程

能耗建模的基本流程如图 1 所示,它包括以下 3 个基本步骤:采集能耗及其相关的资源信息;根据回归分析的方法建立能耗模型;用该能耗模型监测系统能耗.具体流程如下:

(1) 采样

首先,在应用程序运行时采集系统能耗以及服务器能耗的影响因素.根据已有的研究^[1-2],本文最后确定了采样的两个基本方向:处理器的性能计数器(performance counter)和系统使用率(system utilization).

图 2 是收集能耗和特定资源信息的平台示意图.我们把需要进行能耗建模的服务器 B 通过电表连接到电源上,该电表能够实时收集自身的电流和电压信息,并及时把信息传输给收集电能信息的服务器 A.确保通过该电表的电器只有服务器 B,这样,服务器 A 收集到的能耗信息即为服务器 B 的能耗数据.如果电流电压数据分开显示,则我们就可以得到服务器 B 实时的能耗情况;而服务器 B 可以通过采样工具收集自身的资源信息.结合这两部分信息,就可以较为精确地获得建模所需的数据.为了更加有效地提取资源使用信息,我们通常需要在服务器 B 上运行标准检查程序 Benchmark.

(2) 建立能耗模型

一般情况下,服务器能耗变化情况是复杂的,多种因素混杂在一起.因此,我们必须把复杂的研究对象变为简单和理想化的研究对象,并符合我们之前提出的能耗建模的原则.然而,通过采样得到的资源信息因子对能耗贡献有强有弱,本文利用回归分析的方法,通过数学手段,合理地筛选能耗因子,从而减少被引入的因子数目,在不影响精确度的基础上达到降低模型复杂度的目的.

(3) 检验能耗模型的精确度,并利用模型监测系统能耗

根据能耗模型,我们就可以有选择地采集对能耗影响大的因素或组件,用于实时的系统能耗监测.

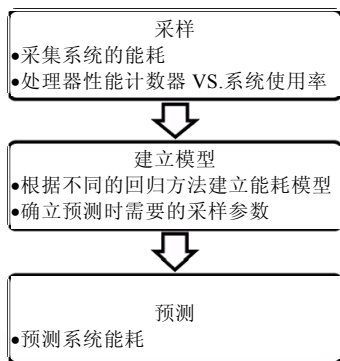


Fig.1 Basic process of energy modeling

图 1 能耗建模的基本流程

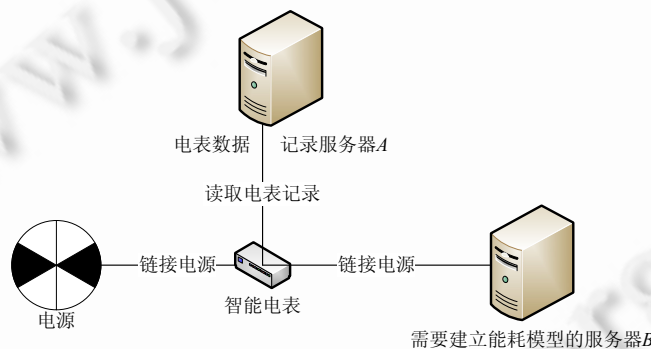


Fig.2 Diagram of energy sampling

图 2 能耗采样的示意图

本文不讨论单独将节点服务器散热当作能耗模型的参数,虽然散热所消耗的能量巨大^[3,4],但是采集热量参数需要额外的传感器和硬件,并且其建模方法和本文所陈述的方法有较大不同.因此,本文暂时不能把服务器散热当作参数来建立能耗模型,而是根据系统资源使用率,建立系统层的能耗模型.

1.3 标准检查程序Benchmark

为了获知节点服务器的性能,研究人员可以在该服务器上运行一组程序或操作,该程序即被称为标准检查程序(benchmark).Benchmark 通常与计算机硬件的某一部分性能相联系,例如处理器的浮点计算、硬盘的读写访问速度等.Benchmark 通常用在不同体系结构的计算机间的不同子系统的比较上.常见的 Benchmark 包括:计算型,例如 SPEC2006;硬盘 I/O 型,例如 I/Ozone;网络型,例如 Netperf(见表 1).

Table 1 Common Benchmark and its types**表 1** 常见的 Benchmark 及其类型

Benchmark 类型	名称	计算类型
SPEC2006	CINT (gobmk); CFP	处理器、内存密集型
NPB	cg, mg, ep, lu 等	处理器、网络密集型
Disk	postmark, iozone, bonnie	处理器、I/O 密集型
Netperf	Netperf 客户端	网络密集型

我们使用 SPEC2006(Standard Performance Evaluation Cooperation)作为能耗建模过程中的 Benchmark, SPEC 是一个松散的开源机构,它是作为 NCGA(National Computer Graphics Association)的一个小组于 20 世纪 80 年代创立的,这个小组的创始者来源于 HP,DEC,MIPS,SUN,他们拥有一组标准检查程序以评测新机器的性能.第 1 组标准检查程序叫作 SPEC89,包含 10 个程序;随后,SPEC92 扩大到 20 个程序,包括 6 个整程序序和 14 个浮点程序,分别称为 SPECint92,SPECfp92.随后,SPEC 又发布了一些新的标准检查程序(如 SPEC95, SPECchpc96,SPECweb96,SPEC2000 等).SPEC2006 是一个评估计算机在真实应用中计算性能的有效工具,这些真实场景包括机器围棋、天气预报、线性规划算法、视频编码、基因序列搜索、编译器、离散事件模拟、量子计算、分子动力学模拟、物理模拟中的光线跟踪、流体力学计算、结构力学中的有限元方法、语音识别等等.

根据相关研究^[4,5]和我们的实验总结,在云计算基础设施中,节点的 CPU 和内存是最为耗能并且状态可调节的部件.所以,我们选取这两个部件的使用率作为能耗模型的基本参数,并将其带入多种回归模型中.

2 相关工作

为了有效控制数据中心庞大的能耗,研究人员首先应当理解数据中心为何会消耗如此多的能源,所以,建立能耗模型是当务之急.在真实的云计算系统中,能耗管理技术的一个重要方面是能耗使用情况的可视性.基于这些可视信息,可以进行自动的或者人工的能耗管理决策.能耗建模可以分成两大类:基于系统使用率的能耗评估模型和基于性能计数器的能耗预测模型.

2.1 基于系统使用率的建模

基于系统使用率的建模是指利用服务器各个组件的使用情况,建立能耗模型.Fan 等人^[6]引入了基于处理器使用率的一元线性回归的全系统能耗模型,并在实验中证明了其准确性.但是 Chung-Hsinget 等人^[7]在深入研究了 SPECpower_ssj2008 从 2007 年~2010 年间公布的 177 个样本后指出,服务器的能耗行为已经发生了巨大的改变.从 2008 年起,服务器的硬件提供商和操作系统软件提供商都提出了不同的能耗管理框架,因此,能耗模型也不再是简单的一元回归模型.Chung-Hsinget 最后提出了一个一元的指数模型,但是该模型只针对于计算密集型的应用,构建的模型参数也来自经验值,缺乏通用性.Beloglazov 等人^[8]使用服务器最大功率、CPU 使用率、服务器空闲和最大功率的能耗比来组建能耗模型.

也有不少研究从服务器的各个组件展开,通过把服务器分为处理器、硬盘、主存等,构建为多元线性回归模型, $P_{服务器}=(P_{处理器}, P_{硬盘}, P_{内存}, P_{主板}, \dots)$.Economou 等人^[9]采用了非侵入式的方法来模拟整个系统的能耗,通过一次性校准阶段,生成能耗模型.他们在实验上也采用了有着完全不同的能耗与工作负载特性的服务器进行了对比研究.Lewi^[10]在硬件组件参数的基础上加入了系统的环境温度、总线活动作为影响因子,预测了长时间工作的服务器的能耗.Kliazovich 等人^[11]将 CPU 以外的其他服务器组件当作常数,与 CPU 功率和频率一起来表示整个服务器的能耗.此外,他们还描述了除计算节点外网络交换机的能耗模型.在其开发的 DENS 调度算法中^[12],描述了整个数据中心的能耗模型,其中,计算节点能耗模型采用 CPU 使用率的线性模型,交换机模型则是各耗能组件的多元一次函数.Lee 等人^[13]则使用了服务器的最大功率、最小功率和 CPU 的当前使用率来表述能耗模型.宋杰等人^[14]给出了 CPU 使用率和频率相关的能耗模型,并比较了不同类型负载的能耗特点和优化办法.

2.2 基于 Performance counter 的建模

为了提高能耗模型的精度,随着处理器的进一步改进,人们提出了基于性能计数器的能耗建模.通过读取处理器的性能计数器的数据,结合不同的组件特性建立能耗模型.例如,现有的内存能耗模型研究发现,影响内存能耗的主要因素是内存读写的吞吐量^[7].可以使用轻量级的内存吞吐量的评估方法,即记录最后一层 Cache(last level cache,简称 LLC)的缺失次数,该参数在大多数处理器上比较容易获得,也可以通过插桩技术更准确地来捕获内存吞吐量.

Contreras^[15],Singh^[16]等人利用 performance event 的数据构建了能耗模型,他们深入研究特定的处理器,在采样之间就确定了需要采样的事件,但是这种方案缺乏通用性.Isci^[17]则对 22 个事件全部采样,通过确定各个事件带来的最大能耗,建立了全回归模型.但是这种方法要求采样事件较多,并且模型较为复杂,不利于扩展到云计算数据中心的规模.

2.3 针对特定计算环境的建模

此外,还有一些是针对特定条件下的能耗建模研究.Li 等人^[18]研究了在 MapReduce 数据中心的 3 个处理过程的工作负载数据,例如 Reduce 文件系统的读操作,使用线性回归模型建立了能耗预测模型.他们的数据来自于两种典型的 MapReduce 应用:一种是计算密集型的应用;另一种是 I/O 密集型的应用.Wu 等人^[19,20]偏向于大规模多核数据中心中的 MPI/OpenMP 混合特性,他们分别利用处理器和内存组件建立了以应用为中心的系统能耗模型.其研究更关注 OpenMP 和 MPI 之间的能耗特性差异,而非模型本身.

3 基于云数据中心计算节点的能耗建模过程

3.1 基于性能计数器的能耗建模

3.1.1 性能计数器的监控事件

Intel 处理器从奔腾(Pentium)系列起,就通过一组特定模型的性能监控计数器(MSR)引入了性能监控系统.这些计数器允许选择处理器的性能参数进行监测和测量,这些性能信息可以用于调整系统和编译器的性能.

在 Intel P6 系列处理器中,性能监控机制得到了增强,允许更广泛地选择要监视的事件,并允许监视权限更大的控制事件.随后,Pentium4 和 Xeon 处理器又引入了新的监控机制和一组新的性能事件.在 Pentium、P6 系列、Pentium4 和 Xeon 中性能监控机制是非结构化的,它们全部依赖特定的模型(处理器系列中互不兼容).从 Intel Core Solo 和 Intel Core Duo 开始,处理器支持一组结构化的性能事件和一组非结构化的性能事件.通常来说,处理器中可以监控的事件包括指令、缓存和页表缓存等.

- 指令(instruction)

通过比较能耗和性能监视事件可以很明显的看出,指令周期对能耗总是有着突出的贡献.通常,当执行一个指令时,越多的计算单元参与工作,会产生越高的功率消耗.

- 缓存(cache)

以 Xeon E5620 为例,这是一款基于微结构的处理器.其 1 阶高速缓存(L1 cache)被分为两部分:一部分专注于缓存指令(预编译的指令),另一部分为数据缓存.2 阶高速缓存(L2 cache)是一个统一的数据和指令缓存.每个处理器内核都有它自己的 L1 cache 和 L2 cache.3 阶高速缓存(L3 cache)是由所有处理器内核共享的,是一个统一兼容的数据和指令缓存.缓存用于减少内存访问的延迟和提高处理器的数据处理速度.一般来说,指令、数据缓存的缺失意味着计算单元需要停滞,直到出现可用的资源.L1/LLC 缓存的低缺失或高命中,有益于指令的速度执行,从而影响了能耗.

- 页表缓存(TLB)

TLB 存储了最近经常访问的页目录和页表项,用于改进虚拟地址到物理地址的转译速度.这些有助于加快内存访问速度,当需要访问存储在系统内存中的页表时,TLB 可以有效减少访问内存的次数.TLB 被分为数据 TLB 和指令 TLB.如果发生 TLB 未命中,处理器停下来直到主存完成这些工作,从而间接影响了系统能耗.

因此,这些性能事件都有可能是能耗影响因子.但是不同的处理器,这些影响力有强有弱,通过数学方式,可以有效地筛选出重要的因子,建立模型.基于性能计数器的线性能耗模型形式如下所示:

$$P = C_0 + \sum_{i=1}^n (C_i \times p_i) \quad (1)$$

其中, C_0 为常数, p_i 表示收集到的性能计数器事件, C_i 表示第*i*个事件对能耗的影响因子. C_0 和 C_i 可以通过简单的线性回归得到.

3.1.2 性能计数器数据的采集

目前,有很多工具可以采集性能计数器的信息,常见的有 OProfile,Perf 等.通过这些工具,应用程序可以利用 PMU(power management unit),tracepoint 和内核中的特殊计数器来进行性能统计.它不但可以分析指定应用程序的性能问题,也可以用来分析内核的性能问题,当然也可以同时分析应用代码和内核,从而全面理解应用程序中的性能瓶颈.

Perf 作为 Linux 内核自带的系统性能优化工具,它在系统结合度、开发活跃度方面有着众多优势.因此,本文采用 Perf 作为性能计数器建模时的采样工具.

3.2 基于系统使用率的能耗建模

如前所述,性能计数器是由 Intel Pentium 系列处理器引入的,并不是所有现存的处理器都支持这种监控体系结构,而且不同类型的处理器之间具有一定的差异性.另一方面,因为某个特定的应用程序或虚拟机在未运行前很难得知 cache miss 等性能事件的数据,所以很难通过性能计数器预知该应用程序或虚拟机导致能耗的增加范围.然而,这些应用初步的能耗预估对具有能耗优化的资源调度算法又是极其重要的,因此,基于系统使用率的能耗模型研究是非常必要的.

早先有关资源使用率和能耗关系的研究通常采用线性模型,因为它简单、直观并且容易计算.当建模的因子相互独立时,线性回归模型是适合的,而系统使用率(如处理器、内存、硬盘使用率)通常会有一定程度的交叉依赖.此外,随着处理器的进一步优化,能耗和系统使用率也并非呈现简单的线性关系.这意味着使用传统的线性模型,其预测效果可能会很差.

采样过程收集了不同的系统负载信息,主要包括 4 个方面,即:处理器、内存、网络和硬盘 I/O.因为采样的目的是建立能耗模型,因此我们只采集了与能耗关系最紧密的负载部分,而把系统其他部分视作一个常量.根据相关研究及我们的实验结果,我们发现处理器和内存占用的能耗比较多.同时,使用率的波动会给系统能耗带来巨大变化,而网络和硬盘等较为稳定.

3.2.1 基于 CPU 使用率的能耗模型

CPU 是计算机系统中主要的耗能部件,并且它具有便捷的调节方法.因此,通过调节 CPU 状态来调节整个数据中心系统的能耗状态是节能的主要方法.在我们学习过的能耗模型中,绝大多数都涉及到 CPU 状态的调节.我们首先选取其中具有代表性的只使用 CPU 使用率来预测能耗的模型^[21-23].

第 1 个模型如公式(2)所示,能耗模型预测的能耗为 CPU 使用率的线性函数, C_0 为常数, C_1 表示 CPU 使用率对能耗的影响因子, C_0 和 C_1 通过简单的线性回归得到:

$$P = C_0 + C_1 \times U_{\text{CPU}} \quad (2)$$

第 2 个模型在第 1 个模型的基础上增加经验参数.根据第 1 个模型,能耗与 CPU 使用率应该呈线性相关,但在很多相关研究^[24-26]和我们的实验中(见第 4.2 节),能耗与资源使用率并非单纯的线性关系,因此,模型 2 通过增加经验参数 C_2 和 t 来修正两者之间的关系,模型如下所示:

$$P = C_0 + C_1 \times U_{\text{CPU}} + C_2 \times U_{\text{CPU}}^t \quad (3)$$

3.2.2 内存相关的能耗模型

现有研究表明:内存是一个重要的耗能部件,其能源消耗有可能占到总能耗的 60%^[27].但据我们观察,内存本身并不是一个能耗消费很高的部件,我们在实验环境中所使用的频率为 1 333 的 4G 内存每 Gbit 功率仅为 3W,但是由于 CPU 采用多核心技术,而每个核心需要耗费主频至少 2 倍以上的内存才能使云数据中心节点中的虚

拟机任务顺利运行,所以整体耗电才有所提高.现有内存节能方式主要是,当内存空闲时,将其待机、休眠或关掉,每一个状态的能耗有所减少(如待机状态能耗为平常的 60%,休眠状态为 10%),但响应时间有所增加.根据我们的实验观察到,内存能耗与 CPU 能耗存在正向关联的关系,即:当 CPU 利用率提高时,内存能耗也同时提高.因此,通过节约 CPU 能耗来节约内存能耗,或者同时节约两者能耗,也是一种可行的节能方法.我们将内存相关的线性能耗模型记为

$$P=C_0+C_1\times U_{\text{CPU}}+C_2\times U_{\text{memory}} \quad (4)$$

3.2.3 存储相关的能耗模型

存储能耗也是整机能耗的重要部分.由于服务器的可扩展结构,1 台服务器可同时连接多台硬盘或多个磁盘阵列,所以硬盘的能耗根据不同的结构变化很大.有研究表明,存储能耗能达到整体能耗的 65%^[27].在服务器或网络集群的系统中,由于有许多并发的进程或实例,为达到较高的并行能力或吞吐量,多个磁盘或阵列往往采用同轴共转的方式,这也增加了磁盘的能耗.目前,磁盘节能的方法主要是采用拥有不同调节功能的磁盘,如功能性调节(工作、待机、休眠)或速度调节(采用多转速硬盘)等,采用闪存技术的存储设备也可能是未来发展趋势.本文中,我们所接触到的能耗模型仅陈述磁盘对于整体功耗的影响.我们将内存相关的线性能耗模型记为

$$P=C_0+C_1\times U_{\text{CPU}}+C_2\times U_{\text{disk}} \quad (5)$$

3.2.4 网络相关的能耗模型

我们同样观察了网络相关的能耗模型,节点中网络的能耗主要包括 NIC(network interface card)的能耗以及处理传输数据所耗费的其他组件的能耗(CPU、内存或存储),但是其他组件的能耗在如上所述的相对应的能耗模型中均可以计算出来,而 NIC 本身的能耗并不大,因此,网络相关的能耗模型主要针对交换设备,其能耗模型和节能方法与计算节点有较大不同,这些内容不在本文讨论的范围内.

3.3 能耗数据的回归分析方法

回归分析是处理变量间相关关系的一种有效工具,其目的在于根据已知预报变量的变化来估计或预测响应变量的变化情况,或者根据响应变量来对预报变量作一定的控制.具体来说,对能耗数据的回归分析研究随机变量 y (能耗)与解释变量 x (资源使用率)间的相关关系,其中,对于每个确定的 x , y 是一个具有确定分布的随机变量.寻求 y 的均值与 x 之间的函数关系的表达式 $Ey=\mu(x)$,便是回归分析的基本问题.这里的 $\mu(x)$ 称为 y 对 x 的回归函数,或简称为 y 对 x 的回归.

3.3.1 多元线性回归模型

多元线性回归通常用于讨论因变量和两个或者两个以上的自变量的线性依赖关系.考虑含 m 个因变量的回归模型:

$$\begin{cases} y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_mx_m + \varepsilon \\ E(\varepsilon) = 0, D(\varepsilon) = \sigma^2 < +\infty \end{cases} \quad (6)$$

其中, y 是观测到的实时能耗; $\beta_0, \beta_1, \beta_2, \dots, \beta_m$ 是回归系数,是需要被确定的量; ε 是不可观测的随机误差; x_1, x_2, \dots, x_m 是系统各部件使用率的观察值,称为回归因子,简称因子; $\beta_i(i=1, \dots, m)$ 实际上反映了因子 x_i 对因变量 y 的贡献大小.

假设有 n 组能耗观测值($y_i, x_{i1}, \dots, x_{im}$), $i=1, \dots, n$,则有:

$$\begin{cases} y_i = \beta_0 + \beta_1x_{i1} + \beta_2x_{i2} + \dots + \beta_mx_{im} + \varepsilon_i, i = 1, \dots, n \\ E(\varepsilon_i) = 0, \text{Var}(\varepsilon_i) = \sigma^2, \text{且} \varepsilon_i \text{相互独立} \end{cases} \quad (7)$$

为了方便,我们采用矩阵表示,记:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{pmatrix}, \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix},$$

则公式(6)可表示成矩阵形式:

$$\begin{cases} y = X\beta + \beta_0 + \varepsilon \\ E(\varepsilon) = 0, D(\varepsilon) = \sigma^2 I \end{cases} \quad (8)$$

(1) 最小二乘法+逐步回归

首先,将能耗相关的系统使用率数据带入多元线性回归模型中,采用最小二乘法估计回归系数 $\beta_0, \beta_1, \beta_2, \dots, \beta_m$,目标是让:

$$Q(\beta_0, \beta_1, \beta_2, \dots, \beta_m) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \dots - \beta_m x_{im})^2 = (y - \beta_0 - X\beta)^T (y - \beta_0 - X\beta) \quad (9)$$

达到极小,则可得到 m 元回归方程为

$$y' = \beta'_0 + \beta'_1 x_1 + \beta'_2 x_2 + \dots + \beta'_m x_m \quad (10)$$

通常来说,寻求最小二乘法的最优解是很困难的,因此我们可以选择逐步回归的方法.逐步回归法的基本思想是:将变量逐个引入,引入的条件是该变量的偏 F 检验是显著的.同时,每引入一个变量后,对已有变量进行检验,将不显著的变量从模型中删除,得到最终的回归模型.

在目前的很多研究中,能耗模型采用这种方法建立,其特点是简单、直观,缺点是准确性不高,因为没有任何直观的理论或证据证明,能耗与系统使用率线性相关,所以使用这种方法建立的能耗模型准确性有限,我们的实验结果也印证了这一观点.

(2) 套索(lasso)回归

我们同样使用套索回归的方法来处理系统使用率数据,由于逐步回归对参数估计的正则化是不连续的,这种不连续性导致了模型选择的不稳定性,虽然这种方法得到的估计也是无偏的.但是某些自变量数据的微小变化都可能导致选择的最优模型的较大变化.因此,为弥补逐步回归方法的不足,我们采用了套索回归的方法.Lasso 方法用模型系数的绝对值函数作为惩罚来压缩模型系数,使绝对值较小的系数自动压缩为 0,从而同时实现显著性变量的选择和对应参数的估计.与传统的模型选择方法相比,Lasso 方法很好地克服了传统方法在选择模型上的不足.当我们使用数据训练时,需要在过度拟合与拟合不足之间达成一个平衡.防止过度拟合的一种方法就是对模型的复杂度进行约束.模型中用到解释变量的个数是模型复杂度的一种体现.岭回归(ridge regression)和套索回归(least absolute shrinkage and selection operator,简称 lasso)将解释变量的系数加入到惩罚函数中,并寻求目标函数的最小值,本质上是对过多的参数实施了惩罚.这两种方法的区别在于惩罚函数不同,但这种微小的区别却使 Lasso 有很多优良的特质,例如:

- 无偏性:当未知参数的真实值比较大时,为了使模型不出现不必要的偏差,导出的估计应该接近无偏;
- 稀疏性:为了减小模型的复杂性,导出的估计应该是一个门限规则,这种规则能自动地将相对较小的模型系数设置为 0;
- 连续性:为了避免预测模型的不稳定性,导出的估计应该对数据是连续的.

性质 b)对我们尤其重要,因为我们需要从各种系统使用率数据中排除那些能耗无关的变量.而 Lasso 的稀疏性刚好可以将这些变量的系数置为 0,很好地解决了模型的变量选择问题.下面的公式展示了线性模型中两种方法所对应的目标函数:

$$\min_{(\beta_0, \beta) \in \mathbb{R}^{m+1}} \left[\frac{1}{2n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \dots - \beta_m x_{im})^2 + \lambda P_\alpha(\beta) \right] \quad (11)$$

其中,

$$P_\alpha(\beta) = (1-\alpha) \frac{1}{2} \|\beta\|_{\ell_2}^2 + \alpha \|\beta\|_{\ell_1} = \sum_{j=1}^m \left[\frac{1}{2} (1-\alpha) \beta_j^2 + \alpha |\beta_j| \right] \quad (12)$$

当 $\alpha=0$ 时,公式(11)是岭回归;当 $\alpha=1$ 时,公式(11)是套索回归.

其中, λ 是设置参数,它控制了惩罚的严厉程度.如果设置得过大,那么最后的模型参数均将趋于 0,形成拟合不足;如果设置得过小,又会形成拟合过度.所以, λ 的取值一般需要通过交叉检验来确定.

3.3.2 非线性的模型

在线性模型定义的基础上,我们引入了非线性的模型,通过函数转换,添加模型中的因子,非线性模型可以用下式表示:

$$f(x)=w \cdot \phi(x)+b \tag{13}$$

(1) 多项式套索模型(polynomial with lasso)

多项式套索模型的基础函数由因变量的多项式形式表示: $\phi(\cdot) = \{x_i^a : 1 \leq a \leq d\}$, 其中, $d=3$. 为了防止公式中引入的参数过多,我们采用 Lasso 回归选择参数.

(2) 多项式+指数模型(polynomial+exponential with lasso)

在多形式模型的基础上, $\phi(\cdot)$ 还增加了函数 e^{x_i} . 与上面的方案一样,同样使用 Lasso 回归选择参数.

(3) 支持向量机回归(SVR)

支持向量机(support vector machine,简称 SVM)是在高维特征空间使用线性函数假设空间的学习系统,它由一个来自最优化理论的学习算法训练,该算法实现了一个由统计学习理论导出的学习偏置^[28]. 现今,支持向量机理论不仅作为数据分类的常见技术,在数据回归方面也有很好的应用.

给定一组训练样本集 $(x_i, y_i), i=1, 2, \dots, n$, 其中 $x_i \in R^m$ 为因变量, $y_i \in R$ 是自变量, n 是样本数目^[29].

当训练集为线性时,估计函数 $f(x)$ 为如下形式:

$$f(x)=w \cdot x+b \tag{14}$$

最优化问题为

$$\begin{cases} \min_{w,b} \frac{1}{2} \|w\|^2 \\ \text{s.t. } y_i - w \cdot x_i - b \leq \varepsilon, w \cdot x_i + b - y_i \leq \varepsilon, i=1, 2, \dots, n \end{cases} \tag{15}$$

在此基础上,引入松弛变量 ξ_i, ξ_i^* , 这样,公式(14)改写为

$$\begin{cases} \min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ \text{s.t. } y_i - w \cdot x_i - b \leq \varepsilon + \xi_i, w \cdot x_i + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i \geq 0, \xi_i^* \geq 0, i=1, 2, \dots, n \end{cases} \tag{16}$$

对于训练集为非线性的情况,通过某一非线性函数 $\phi(\cdot)$ 将训练集数据映射到高维线性特征空间. 在此情况下,估计函数 $f(x)$ 为如下形式:

$$f(x)=w \cdot \phi(x)+b \tag{17}$$

同样地,引入松弛变量后,最优化问题为

$$\begin{cases} \min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ \text{s.t. } y_i - w \cdot \phi(x_i) - b \leq \varepsilon + \xi_i, w \cdot \phi(x_i) + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i \geq 0, \xi_i^* \geq 0, i=1, 2, \dots, n \end{cases} \tag{18}$$

3.3.3 结果判定标准

本文中,判定一个能耗模型是否精确的标准是相对偏差 $Power_{error}$, 其中, $Power_{预测}$ 表示我们能耗模型预测得到的结果,而 $Power_{真实}$ 表示用电表监测到的结果. 通过对比二者的相对偏差,就可以知道模型的准确性. 公式(19)为 $Power_{error}$ 的计算公式,具体实验见下节.

$$Power_{error} = \frac{Power_{预测} - Power_{真实}}{Power_{真实}} \tag{19}$$

4 能耗建模实验与结果分析

4.1 基于性能计数器的能耗建模结果

4.1.1 实验平台

在本组实验中,实验环境节点采用 Intel Xeon E5620 处理器,主内存 48G, Linux 内核版本为 2.6.32-220.el6.x86_64. 监控性能计数器的工具是 perf-2.6.32. 编译环境为 gcc-4.4.6.

4.1.2 训练集合和测试集合

为了有效采样,我们把 SPEC2006 作为我们的标准检查程序. SPEC2006 分为浮点计算和整数计算两种类型,为充分利用处理器的信息,我们把这两类计算分为两部分:一部分用于采集 performance counter 信息,建立模型;另一部分在模型检测时使用,分析不同数学方法对基于 performance counter 建模的影响. 具体分类见表 2.

Table 2 Training set and testing set of energy modeling based on performance counter

表 2 基于性能计数器能耗建模的训练集合和测试集合

	训练集合	测试集合
整数型 Benchmark	bzip2, gcc, sjeng, h264ref, astar, xalancbmk, gobmk	hammer, libquantum, mcf, omnetpp
浮点型 Benchmark	bwaves, gamess, zeusmp, gromac, cactusADM, calculix, GemsFDTD, tonto, wrf, sphinx3, dealII, soplex, povray	lbm, leslie3d, milc, namd

4.1.3 采集和建模

我们在服务器上将对训练集合中的每个 Benchmark 运行了 10 遍,采集表 3 中提到的 Xeon 处理器的性能监控事件.

Table 3 Performance monitoring events of Intel Xeon E5620

表 3 Intel Xeon E5620 的性能监控事件

cpu-cycles	L1-dcache-loads	LLC-loads
stalled-cycles-frontend	L1-dcache-load-misses	LLC-load-misses
stalled-cycles-backend	L1-dcache-stores	LLC-stores
instructions	L1-dcache-store-misses	LLC-store-misses
stalled cycles per insn	L1-dcache-prefetches	LLC-prefetches
cache-references	L1-dcache-prefetch-misses	LLC-prefetch-misses
cache-misses	L1-icache-loads	context-switches
branch-instructions	L1-icache-load-misses	cpu-migrations
branch-misses	dTLB-loads	alignment-faults
bus-cycles	dTLB-load-misses	emulation-faults
task-clock	dTLB-stores	branch-loads
page-faults	dTLB-store-misses	branch-load-misses
minor-faults	iTLB-loads	
major-faults	iTLB-load-misses	

我们通过以上程序获得多组数据以后,使用 3 种数学模型进行了回归分析,分别是:线性回归下的最小二乘法逐步回归、套索回归和非线性的 SVR. 我们采用 SPSS 软件进行多元线性最小二乘法回归;使用 glmnet 包实现套索回归,该包可以高效地为线性回归做 Lasso 回归或者岭回归,以及两者的混合回归;针对 SVR,我们采用了 libsvm 工具,通过数据归一化、训练样本等步骤获得能耗模型.

4.1.4 结果分析

本节中,我们将展示 3 种模型(线性回归下的最小二乘法、套索回归和非线性的 SVR)对能耗的预测情况.

我们把样例分为训练集合和预测集合,因此,训练集合的相对偏差大小说明了该模型对数据集合的拟合情况,而预测集合的相对偏差大小说明了该模型对实际的预测能力. 相对偏差越小,说明模型的效果越好.

如图 3 所示,给出训练样本的相对偏差情况,可以看出:这 3 种回归方式在训练集合中的相对偏差都非常小,基本上都在(-2%,+2%)之间. 同时,支持向量机的回归方法在训练集的拟合方面有更大的优势.

图 4 是 3 种模型在训练集合中的拟合值与实际值的比较,可以看出,这些值都非常接近.

图 5 给出测试集合中 3 种模型的相对偏差情况,可以看到,相对偏差的范围基本上比较集中,落在了(-4%,

+4%)之间.

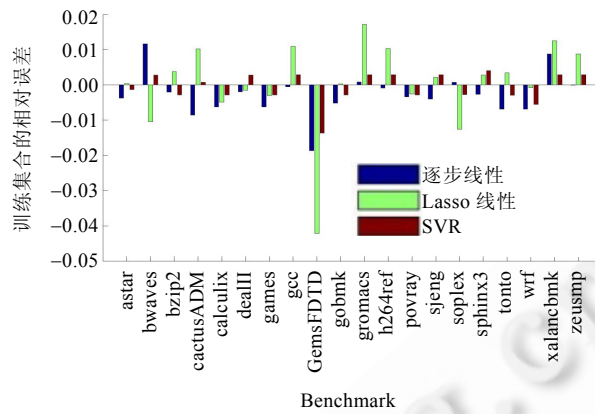


Fig.3 Comparison of relative deviation of performance counter's training set

图 3 性能计数器建模的训练集合的相对偏差比较

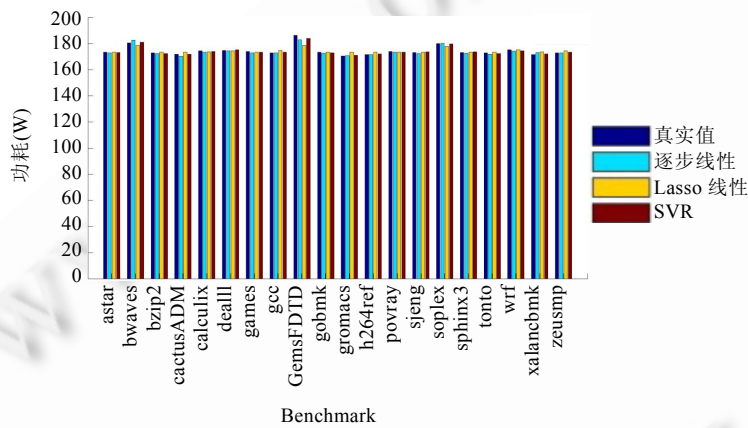


Fig.4 Comparison of fitted value and the true value of performance counter's training set

图 4 性能计数器建模的训练集合的拟合值与真实值比较

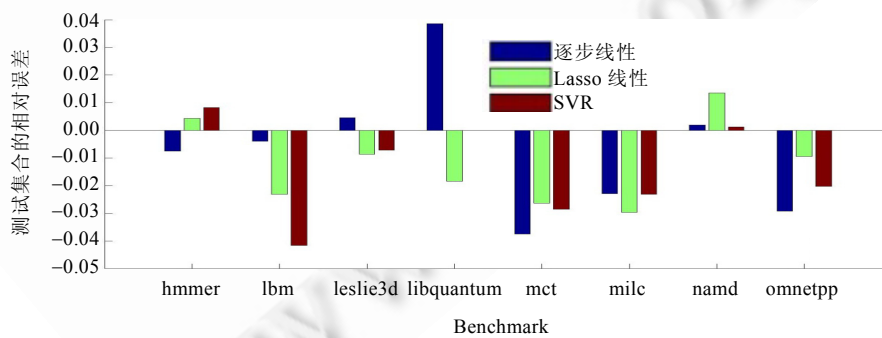


Fig.5 Comparison of relative deviation of performance counter's testing set

图 5 性能计数器建模的测试集合的相对偏差比较

图 6 是测试集合中 3 种模型的预测值与真实值的对比图,这些数据都非常接近.

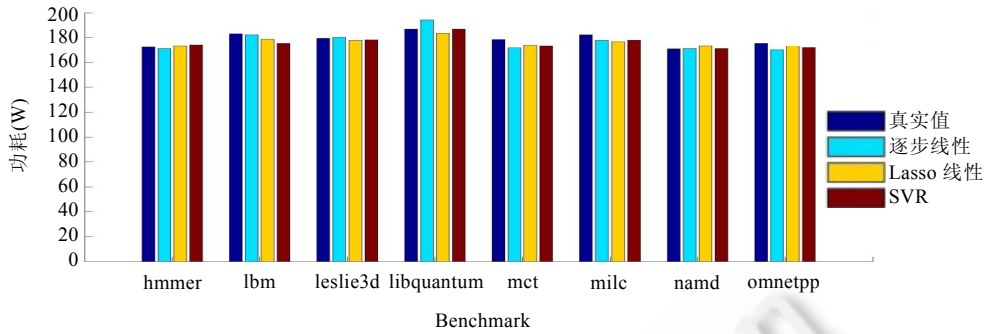


Fig.6 Comparison of prediction value and the true value of performance counter's testing set

图 6 性能计数器建模的测试集合的预测值与真实值比较

从上述分析中可以看出,这 3 种回归方案的偏差范围的区别并不明显.支持向量机回归的方法在测试集合并没有特别的优势,考虑到它在实际应用中比线性回归方案更为复杂,因此在基于 performance counter 的系统能耗预测中,直接使用线性回归方案是可行的.

4.2 基于系统使用率的能耗建模结果

4.2.1 实验平台

在本次实验中,我们采用 Intel Xeon E5620 处理器,主内存 48G, Linux 内核版本为 2.6.32-220.el6.x86_64.编译环境为 gcc-4.4.6.

4.2.2 采样和建模

在 Linux/Unix 下,处理器使用率分为用户状态、系统状态和空闲状态,分别表示处理器处于用户态执行的时间、系统内核执行的时间和空闲系统进程执行的时间.平时所说的处理器使用率是指:CPU 非系统空闲进程的执行时间/CPU 总的执行时间.在 Linux 的内核中,有一个全局变量 jiffies 代表时钟节拍,它的单位随硬件平台发生变化,每个时钟节拍 jiffies 都要加 1.处理器的使用率就是(执行用户态+系统态的 jiffies)/总的 jiffies.

在本文中,服务器的操作系统是 CentOS6.2,所以可以通过 /proc/stat 文件计算处理器的使用率.在我们的实验中,/proc/stat 包括系统处理器的负载信息,它可以表示为

```
cpu 245679 759 89549 345763563 48744 29 2316 0 0
cpu0 196816 395 4962 172780284 45243 0 44 0 0
cpu1 48862 364 84587 172983278 3501 29 2271 0 0
```

其中,在每一行中,每个数据意义如下:

- *user*:从系统启动开始累计到当前时刻,用户态的 CPU 时间(单位:jiffies),不包含 *nice* 值为负进程;
- *nice*:从系统启动开始累计到当前时刻,*nice* 值为负的进程所占用的 CPU 时间(单位:jiffies);
- *system*:从系统启动开始累计到当前时刻,核心时间(单位:jiffies);
- *idle*:从系统启动开始累计到当前时刻,除硬盘 I/O 等待时间以外其他等待时间(单位:jiffies);
- *iowait*:从系统启动开始累计到当前时刻,硬盘 I/O 等待时间(单位:jiffies);
- *irq*:从系统启动开始累计到当前时刻,硬中断时间(单位:jiffies);
- *softirq*:从系统启动开始累计到当前时刻,软中断时间(单位:jiffies).

$$CPU_Time = user + system + nice + idle + iowait + irq + softirq.$$

$$t_{stat} = t_{stat_now} - t_{stat_last} \quad (20)$$

$$Utilization_{cpu} = \frac{t_{user_mode} + t_{system_mode}}{t_{user_mode} + t_{system_mode} + t_{idle}} \quad (21)$$

基于上述公式,我们就可以计算在一个特定的时间片内的处理器使用率.通过采样得到处理器的使用信息后,采用相似的方法在/proc/meminfo 文件也可以得到内存的使用率.完成系统负载和能耗数据的数据采集后,就可以进一步研究它们之间的关系.

我们在实验中收集了 2 000 多组数据,每组数据都由一个个三元组组成($Utilization_{CPU}+Utilization_{MEM}+Power$),这些数据的范围从 0%~100%,如图 7 所示.

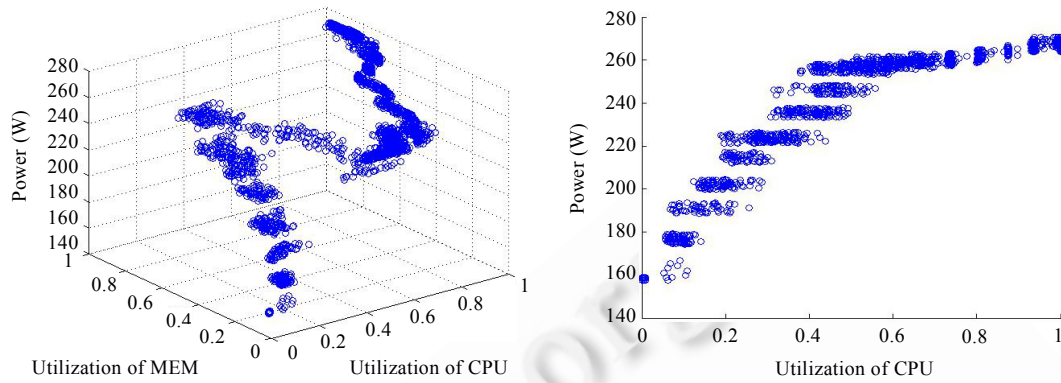


Fig.7 Relationship between system utilization and full-system power

图 7 服务器的处理器、内存使用率和能耗图

根据这些数据,再使用第 3.3 节中提到的不同的数学方法,我们进行了回归比较.这些数学方法分别是线性回归下的 Lasso 回归、非线性回归下的多项式回归(polynomial with lasso)、多项式+指数项回归(polynomial+exponential with lasso)和支持向量机回归(SVR).我们对收集到的使用率信息,结合模型特点,进行了数学处理,增加了模型中的因子(见表 4).

Table 4 Factors of non-linear regression model

表 4 非线性回归模型包含的因子

数学回归方法	模型因子
Lasso	$Utilization_{CPU}, Utilization_{MEM}$
Polynomial with lasso	$Utilization_{CPU}, Utilization_{MEM},$ $Utilization_{CPU}^2, Utilization_{MEM}^2,$ $Utilization_{CPU}^3, Utilization_{MEM}^3$
Polynomial+exponential with lasso	$Utilization_{CPU}, Utilization_{MEM},$ $Utilization_{CPU}^2, Utilization_{MEM}^2,$ $Utilization_{CPU}^3, Utilization_{MEM}^3,$ $e^{Utilization_{CPU}}, e^{Utilization_{MEM}}$

之后,我们利用 Lasso 回归的方法,删选能耗模型最终需要的因子,建立公式.该方法得到的云数据中心计算节点的能耗模型为

$$f(U) = \begin{pmatrix} 375.8550 & -401.0088 & 164.4327 \end{pmatrix} \begin{pmatrix} Utilization_{CPU} \\ Utilization_{CPU}^2 \\ Utilization_{CPU}^3 \end{pmatrix} + \begin{pmatrix} -30.6192 & 41.8946 & -19.8122 \end{pmatrix} \begin{pmatrix} Utilization_{MEM} \\ Utilization_{MEM}^2 \\ Utilization_{MEM}^3 \end{pmatrix} + 155.0537.$$

4.2.3 结果比较

我们比较了线性回归下的 Lasso 回归和非线性回归下的多项式回归(polynomial with lasso)、多项式+指数项回归(polynomial+exponential with lasso)和支持向量机回归(SVR).本节介绍了 4 种模型的相对偏差和预测值与真实值的比较.

从图 8 中我们可以看出,线性回归的效果并不理想:其偏差范围明显高于其他 3 种,其中最大正向误差接近 30%,并不符合我们在第 3.1 节中提出的能耗建模原则,而且偏差没有达到正态分布的要求.

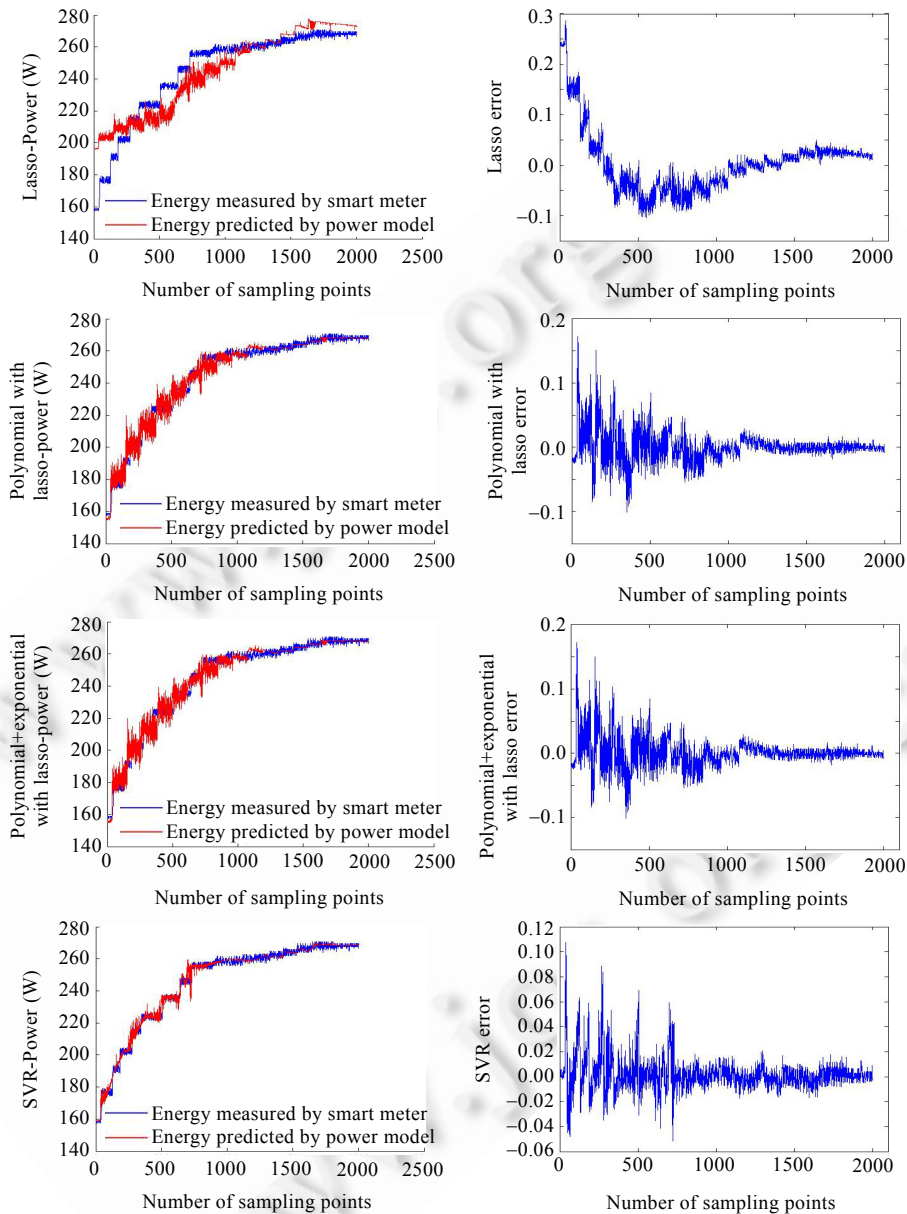


Fig.8 Comparison of actual and predicted power by different regression models

图 8 System Utilization 建模的预测值与真实值比较

图 8 也验证了我们之前的猜测,线性回归的方法并不适合基于系统使用率的能耗建模.而支持向量机回归

的偏差最小,在(-6%,+10%)之间,基本上是线性回归的偏差的一半.因此可以得出:在新型的硬件和软件影响下,计算机能耗建模很难通过单纯线性回归的方式而得到比较准确的能耗模型.

此外,我们还观察到,polynomial with lasso 和 polynomial+exponential with lasso 方法的相对偏差范围非常接近.它们大部分的点都落在了(-10%,+10%)之间,这个范围也是我们能接受的.同时也说明,模型中加入处理器和内存使用率的指数形式,并没有给模型带来更好的预测效果.

图 8 是这 4 种方法的真实值和预测值的比较,同时也给出了预测值的误差范围.图 8 中我们用不同的线条来表示通过公式估计出来的数值以及这些测试样例的真实值.从图中可以直观地看出:在线性回归的 Lasso 方案下,分别代表估计出来的数值以及真实值的数值线有着明显的分离.而支持向量机的回归图中,两者非常接近.同时,polynomial with lasso 和 polynomial+exponential with lasso 也有着很好的预测表现.

在实际运用中,尽管支持向量机回归有着非常好的预测能力,但是它更为复杂,而 polynomial with lasso 的误差范围仍在我们的可接受范围内.同时,它的全局平均误差非常小.所以在面向能耗优化的资源调度算法的研究中,我们采用了 polynomial with lasso 回归方式建立的基于系统使用率的能耗模型.

5 结束语

本文从处理器的性能计数器(performance counter)和系统使用率(system utilization)角度入手,结合多元线性回归的逐步回归和 Lasso 回归、非线性回归下的多项式、多项式+指数形式和支持向量机回归,分析总结了不同参数和方法对服务器能耗建模的影响.总的说来,基于性能计数器的建模采用一般的多元线性回归即可,而基于系统使用率的能耗模型,则建议采用多项式的非线性模型.另一方面,性能计数器的建模效果要优于使用系统使用率的模型,但是使用系统使用率的能耗模型可以有效地预估计单个应用对服务器的影响,这对于众多基于能耗感知的资源调度方法来说是非常关键的.并且在只采样少数能耗相关组件的系统使用率的情况下,在系统稳定后,也能将能耗模型的偏差控制在 5%以内,所以我们在后续研究中使用了该方法.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是北京航空航天大学计算机学院李未院士、吴文峻教授和罗杰讲师等老师、博士讨论班和网络管理组的同学表示感谢.

References:

- [1] McCullough JC, Agarwal Y, Chandrashekar J, Kuppaswamy S, Snoeren AC, Gupta RK. Evaluating the effectiveness of model-based power characterization. In: Proc. of the USENIX Annual Technical Conf. USENIX Association Berkeley, 2011. 12. https://www.usenix.org/legacy/events/atc11/tech/final_files/McCullough.pdf
- [2] Pakbaznia E, Pedram M. Minimizing data center cooling and server power costs. In: Proc. of the 14th ACM/IEEE Int'l Symp. on Low Power Electronics and Design. New York: ACM Press, 2009. 145-150. [doi: 10.1145/1594233.1594268]
- [3] Bash C, Forman G. Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. In: Proc. of the 14th USENIX Annual Technical Conf. USENIX Association Berkeley, 2007. 138-140. <http://dl.acm.org/citation.cfm?id=1364414>
- [4] Moreno-Vozmediano R, Montero RS, Llorente IM. Key challenges in cloud computing: Enabling the future Internet of services. Internet Computing, IEEE, 2013,17(4):18-25. [doi: 10.1109/MIC.2012.69]
- [5] Barbulescu M, Grigoriu RO, Neculoiu G, Halcu I, Sandulescu VC, Niculescu-Faida O, Marinescu M, Marinescu V. Energy efficiency in cloud computing and distributed systems. In: Proc. of the 2013 14th RoEduNet Int'l Conf. on Networking in Education and Research. IEEE, 2013. 1-5. [doi: 10.1109/RoEduNet.2013.6714197]
- [6] Fan X, Weber WD, Barroso LA. Power provisioning for a warehouse-sized computer. ACM SIGARCH Computer Architecture News, 2007,35(2):13-23. [doi: 10.1145/1250662.1250665]
- [7] Hsu CH, Poole SW. Power signature analysis of the SPECpower_ssj2008 Benchmark. In: Proc. of the 2011 14th IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS). IEEE, 2011. 227-236. [doi: 10.1109/ISPASS.2011.5762739]

- [8] Beloglazov A, Abawajy J, Buyya R. Energy-Aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 2012,28(5):755–768. [doi: 10.1016/j.future.2011.04.017]
- [9] Economou D, Rivoire S, Kozyrakis C, Ranganathan P. Full-System power analysis and modeling for server environments. In: *Proc. of the 14th Int'l Symp. on Computer Architecture*. IEEE, 2006. 70–77. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.84.1332>
- [10] Lewis AW, Ghosh S, Tzeng NF. Run-Time energy consumption estimation based on workload in server systems. *HotPower*, 2008, 8:17–21. https://www.usenix.org/legacy/events/hotpower08/tech/full_papers/lewis/lewis_html/
- [11] Kliazovich D, Bouvry P, Khan SU. GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, 2012,62(3):1263–1283. [doi: 10.1007/s11227-010-0504-1]
- [12] Kliazovich D, Bouvry P, Khan SU. DENS: Data center energy-efficient network-aware scheduling. *Cluster Computing*, 2013,16(1): 65–75. [doi: 10.1007/s10586-011-0177-4]
- [13] Lee YC, Zomaya AY. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, 2012, 60(2):268–280. [doi: 10.1007/s11227-010-0421-3]
- [14] Song J, Li TT, Yan ZX, Na J, Zhu ZL. Energy-Efficiency model and measuring approach for cloud computing. *Ruan Jian Xue Bao/ Journal of Software*, 2012,23(2):200–214 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4144.htm> [doi: 10.3724/SP.J.1001.2012.04144]
- [15] Contreras G, Martonosi M. Power prediction for Intel XScale® processors using performance monitoring unit events. In: *Proc. of the 2005 14th Int'l Symp. on Low Power Electronics and Design (ISLPED 2005)*. IEEE, 2005. 221–226. [doi: 10.1109/LPE.2005.195518]
- [16] Singh K, Bhadauria M, McKee SA. Real time power estimation and thread scheduling via performance counters. *ACM SIGARCH Computer Architecture News*, 2009,37(2):46–55. [doi: 10.1145/1577129.1577137]
- [17] Isci C, Martonosi M. Runtime power monitoring in high-end processors: Methodology and empirical data. In: *Proc. of the 36th Annual IEEE/ACM Int'l Symp. on Microarchitecture*. IEEE Computer Society, 2003. 93. [doi: 10.1109/MICRO.2003.1253186]
- [18] Li W, Yang H, Luan Z, Qian D. Energy prediction for MapReduce workloads. In: *Proc. of the IEEE 2011 9th Int'l Conf. on Dependable, Autonomic and Secure Computing (DASC)*. IEEE, 2011. 443–448. [doi: 10.1109/DASC.2011.88]
- [19] Lively C, Wu X, Taylor V, Moore S, Chang H, Su C, Cameron K. Power-Aware predictive models of hybrid (MPI/OpenMP) scientific applications on multicore systems. *Computer Science-Research and Development*, 2012,27(4):245–253. [doi: 10.1007/s00450-011-0190-0]
- [20] Lively C, Wu X, Taylor V, Moore S, Chang H, Su C, Cameron K. Energy and performance characteristics of different parallel implementations of scientific applications on multicore systems. *Int'l Journal of High Performance Computing Applications*, 2011, 25(3):342–350. [doi: 10.1177/1094342011414749]
- [21] Garg SK, Versteeg S, Buyya R. A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 2013, 29(4):1012–1023. [doi: 10.1016/j.future.2012.06.006]
- [22] Nathuji R, Schwan K. VirtualPower: Coordinated power management in virtualized enterprise systems. *ACM SIGOPS Operating Systems Review*, 2007,41(6):265–278. [doi: 10.1145/1294261.1294287]
- [23] Nathuji R, Schwan K, Somani A, Joshi Y. VPM tokens: Virtual machine-aware power budgeting in datacenters. *Cluster Computing*, 2009,12(2):189–203. [doi: 10.1007/s10586-009-0077-z]
- [24] Buyya R, Beloglazov A, Abawajy J. Energy-Efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. *arXiv preprint arXiv:1006.0308*. 2010.
- [25] Beloglazov A, Buyya R. Energy efficient resource management in virtualized cloud data centers. In: *Proc. of the 2010 10th IEEE/ACM Int'l Conf. on Cluster, Cloud and Grid Computing*. IEEE Computer Society, 2010. 826–831. [doi: 10.1109/CCGRID.2010.46]
- [26] Buyya R, Ranjan R, Calheiros RN. Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In: *Proc. of the 14th 2009 Int'l Conf. on High Performance Computing & Simulation (HPCS 2009)*. IEEE, 2009. 1–11. [doi: 10.1109/HPCSIM.2009.5192685]

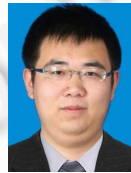
- [27] Feller E, Morin C, Leprince D. State of the art of power saving in clusters and results from the EDF case study. Institut National de Recherche en Informatique et en Automatique (INRIA), 2010. <http://hal.inria.fr/docs/00/54/38/10/PDF/RR-7473.pdf>
- [28] Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge: Cambridge University Press, 2000.
- [29] Du SX, Wu TJ. Support vector machines for regression. Journal of System Simulation, 2003,15(11):1580–1585 (in Chinese with English abstract) [doi: 10.3969/j.issn.1004-731X.2003.11.023]

附中文参考文献:

- [14] 宋杰,李甜甜,闫振兴,那俊,朱志良.一种云计算环境下的能效模型和度量方法.软件学报,2012,23(2):200–214. <http://www.jos.org.cn/1000-9825/4144.htm> [doi: 10.3724/SP.J.1001.2012.04144]
- [29] 杜树新,吴铁军.用于回归估计的支持向量机方法.系统仿真学报,2003,15(11):1580–1585. [doi: 10.3969/j.issn.1004-731X.2003.11.023]



罗亮(1981—),男,陕西汉中,博士,讲师,主要研究领域为云计算的能耗优化,能耗建模,资源调度算法,组合优化.
E-mail: luoliang@nlsde.buaa.edu.cn



张飞(1990—),男,硕士生,主要研究领域为云计算,云存储,数学建模.
E-mail: zhangfei@nlsde.buaa.edu.cn



吴文峻(1973—),男,博士,教授,CCF 会员,主要研究领域为数据密集型计算,云计算,多媒体协作,群体软件工程.
E-mail: wwj@nlsde.buaa.edu.cn