一种基于统计分析的存储系统性能调优方法

陆承涛 冯 丹 王 芳 葛雄资
(华中科技大学计算机科学与技术学院 武汉 430074)
(华中科技大学武汉光电国家重点实验室 武汉 430074)

摘 要 计算机系统参数的合理配置能有效提升应用程序的性能。以NFS网络存储系统为例,提出了一种基于统计分析的存储系统性能调优方法,该方法分为关键系统参数识别和关键参数性能优化两个子阶段。阶段一采用方差分析(ANOVA)来建模系统参数的性能灵敏度,识别出对应用性能有显著影响的关键系统参数;然后在此基础上,阶段二采用响应面分析(RSM)来考察各关键参数对性能响应的影响,并综合前两个子阶段给出了性能调优算法,通过该算法找出系统的最优配置,从而最终达到性能调优的目的。最后,用实验评价了文中方法在Web,E-mail,Fileserver,Linux实用程序以及微基准测试等多种重要应用场景下的性能调优结果,实验结果证实了该调优方法的有效性和实用性。

关键词 性能调优,方差分析,响应面分析,存储系统,性能评价
中图法分类号 TP302.7 **文献标识码** A

Statistical Analysis-based Approach for Storage System Performance Tuning

LU Cheng-tao FENG Dan WANG Fang GE Xiong-zi
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)
(Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract The reasonable configuration of computer systems can dramatically improve performance of applications. Taking an NFS storage system as a case study, this paper proposed a statistical analysis-based performance tuning approach for storage systems and it proceeds in two phases. In the first phase, we leveraged the analysis-of-variance method(ANOVA) to model the performance sensitivity and thus identified the critical system parameters that have a significant effect on performance of applications; and furthermore, based on the previous phase, we employed the response surface methodology (RSM) to fulfill the tuning analysis of the critical parameters in the second phase. Combining the foregoing two steps, we then presented the performance tuning algorithm, which is eventually used to figure out the optimal combination of critical system parameters. Finally, the experimental results demonstrated the effectiveness and feasibility of our performance tuning approach through an extensive evaluation under various representative scenarios including Web, E-mail, Fileserver, Linux utilities, and micro-benchmarks.

Keywords Performance tuning, Analysis of variance, Response surface methodology, Storage system, Performance evaluation

1 引言

计算机系统技术发展到今天,随着对系统功能与性能需求的不断提高,使得计算机系统变得日趋庞大而复杂,进而导致复杂度危机的形成,复杂度危机使得系统管理日益成为计算机系统领域亟待解决的关键问题[9]。由于系统性能始终是计算机系统设计者和管理者最为关心的主要目标之一,因而性能管理是计算机系统管理的重要子课题。系统的性能由应用程序、系统环境、硬件配置等诸多因素决定,因为一种特定的系统配置不可能满足所有特性各异的上层应用的性能需求,使应用性能达到最优,所以当系统运行于缺省配置或可选的非最优配置时,应用性能存在较大的提升空间。作为性能管理的核心任务,性能调优是指当某种应用运行于某种系统环境之上时,通过改变系统的软硬件配置,使得应用性能达到最优。系统性能调优不仅能有效改善应用的性能,还能显著提高计算机系统资源的利用效率。因此,性能调优问题的研究具有极为重要的理论意义和实用价值,所以一直是计算机系统领域的研究热点之一。

到稿日期:2009-12-09 返修日期:2010-03-04 本文受"863"中国高技术研究发展计划项目(2009AA01A401,2009AA01A402),教育部创新团队项目(IRT0725)资助。

陆承涛(1976-),男,博士生,主要研究方向为文件系统与存储系统、性能评价、负载特征化,E-mail:taylorarch@163.com;冯 丹(1970-),女,教授,博士生导师,主要研究方向为文件系统与存储系统、计算机系统结构;王 芳(1972-),女,教授,博士生导师,主要研究方向为文件系统与存储系统;葛雄资(1982-),男,博士生,主要研究方向为文件系统与存储系统、能耗、固态硬盘。

为此，不同领域的研究人员已开展了大量研究，提出了多种技术方法来优化系统的性能。如在数据库系统领域，主要的性能调优方式包括数据缓存大小、页大小、多道程序等级等参数的调整以及物理设计调优等，这些方法大都通过调节单个系统组件内的参数来达到调优的目的。但其主要关注调优过程中的某个子问题或局限于系统内单个组件参数的调整，缺少对系统全局的考虑。又如在 Web 服务领域，为最大化系统的吞吐量，通常采用的是性能监视、性能分析、具体行动三步走的经验主义方法，该类方法主要依赖于管理员对系统的主观认知程度，因而缺少泛化意义。另外，考虑到 Web 服务系统的多层架构，常见的性能优化手段还包括 Web 服务组件放置策略的研究，但因系统特性的不同，这类方法不能直接适用于存储系统。关于性能调优问题的更多相关研究将在第 2 节加以详细评述。

因此，本文的研究动机在于提出一种用于存储系统的性能调优方法。首先，以往关于性能调优问题的研究工作主要集中于数据库和 Web 服务领域，而针对存储系统的相关工作则较为少见。其次，从性能调优技术的层面，已有方法主要关注解决调优问题的各个子问题，而较少从全局的角度来考虑所研究系统的性能调优问题，且已有方法大多依赖于经验规则(rule of thumb)，而这些规则对于存储系统的性能调优缺乏理论指导意义，不易移植并应用于存储系统。因此，与以往的研究工作相比，本文的主要贡献在于：(1)从系统整体的角度，以 NFS 存储系统(Network File System)为例，提出了一种基于统计分析思想的存储系统性能调优方法，给出了一种存储系统性能调优的理论依据。该方法分为两个步骤，首先利用方差分析(Analysis of Variance，ANOVA)来建模系统的性能灵敏度，对各系统参数执行性能灵敏度分析，识别出对应用性能有显著影响的关键参数；然后，在此基础上，通过响应面分析法(Response Surface Methodology，RSM)来考察各关键参数的性能响应，并给出性能调优算法，通过该算法即可找出使应用性能达到最佳的系统参数组合，从而最终达到性能调优的目的；(2)用实验评价和验证了文中调优方法在 Web，E-mail，Fileserver，Linux 实用程序、微基准测试等多种典型应用场景下的有效性和实用性。

本文第 2 节分析了相关工作，评述了本文的研究与以往工作的联系；第 3 节首先概述了 NFS 存储系统的诸多参数，然后在此基础上详细阐述了基于统计分析的性能调优方法，并给出了系统性能调优算法；第 4 节为实验验证部分，先描述了系统实验环境，然后比较了 Web，E-mail，Fileserver，Linux 实用程序和微基准测试等多种典型应用场景下系统调优前后的性能结果，并就实验结果进行了讨论；最后总结了全文的工作，并指出了下一步的研究方向。

2 相关工作

计算机系统性能调优问题因其重要的应用价值而一直为学术界和工业界所关注，目前已取得大量的研究成果，这些工作主要分布于数据库、Web 服务和存储系统 3 大领域，下面分别加以评述。

数据库系统因其规模庞大、结构复杂，从而对性能调优的研究难以求全责备，现有的调优方法主要着重于对某个子问题的研究。如文献[2]设计了一个索引分析实用工具，该程序用来执行系统性能的 what-if 分析；Debnath 等人通过 PLACKETT&BURMAN(P&B)统计方法解决了不同参数对性能影响程度的排序问题[4]；文献[6]设计了一个数据库配置参数的自动性能调优工具，其主要设计目标是调优实验的高效率、调优过程的低开销以及跨平台的可移植性；Oracle 10g 数据库系统主要采用瓶颈消除和主动监视两种方法来调优性能[10]。前者的重点在于消除资源瓶颈，而后者则采用比较方法，周期性地监视系统的运行时数据，并与历史行为数据进行比较，目的在于找出性能问题的根源，从而优化系统的性能；文献[12]采用影响图模型，依据模型对 Berkeley DB 系统的 4 个参数进行调整，但随着系统参数个数的增加，该方法的效率将变得较为低下。本文研究与上述调优方法关注的重点不同，首先要解决的问题是对系统性能有显著影响的关键参数进行识别，这是对现有工作的补充；同时，本文方法为一种端对端方法，从系统全局的角度考虑了 NFS 存储系统三方的性能优化；当系统参数个数较多时本文方法同样适用，且不需要对系统的运行进行长期监测。

其次，Web 服务领域的研究人员也提出了很多调优方法。近年来的代表性工作包括：利用目前主流 Web 服务系统的多层多组件结构，文献[8]研究了 Web 应用组件的放置策略，其被形式化为一个组合优化问题，通过对该问题的求解，以实现系统吞吐量的最大化。因系统特性各异，这些调优技术不能直接适用于存储系统且不易移植到存储系统当中。文献[3]则给出了一种用于 Web 集群系统的调优算法，该算法实现性能调优的具体方式为系统硬件的重新配置，而本文的研究则是在确定的存储系统硬件配置的前提下进行的。

再次，存储系统的性能调优目前主要依据启发式经验规则。文献[1]提出了一种磁盘块重组方法，通过观察 I/O 负载，并总结其特征，依据这些特征来指导磁盘数据布局，从而达到优化性能的目的。与之不同的是，本文方法为一种端对端方法，综合考虑了 NFS 存储系统的三方(服务器、系统互联和客户端)；文献[14]通过优化存储系统的资源分配来实现应用的性能保证，该方法采用机器学习的方法建立设备和负载模型知识库，按照观察－分析－行动的三步循环规则来优化性能，但其主要针对共享存储系统的粗粒度性能优化，而本文的方法是一种细粒度的优化技术；Ellard 等人提出了一种离线分析负载 trace 特征规律的方法[7]，利用这些经验规律达到优化性能的目的，但其需要上层应用相对固定，且需要长时间的 trace 捕获，而本文方法则不受此限制。文献[17]提出了一种块级数据重要性的评价模型，该模型用于指导数据迁移来实现性能优化，而本文方法考虑的是系统参数配置，通过调整关键系统参数来实现性能调优。除上述 3 个领域的工作之外，近年来性能调优问题的研究还包括调优自动化以及虚拟机性能调优等，有兴趣的读者可参阅相关文献[11,16]。

综上所述，与以往工作不同的是，本文研究主要解决的是存储系统关键参数的识别问题以及关键参数配置的优化问题；同时，本文的调优方法主要基于统计分析的思想，利用方差分析和响应面分析来建模系统参数的性能灵敏度和系统性能响应。

3 存储系统的性能调优方法

本节首先描述 NFS 存储系统的系统环境以及部分系统

参数，然后详细阐述了基于统计分析的性能调优方法，最后给出了性能调优算法。

3.1 NFS存储系统概述

NFS存储系统为附网存储系统的两种主要系统结构之一，主要由存储服务器、系统互联、应用客户端3个组件构成。存储服务器为多个客户端所共享，而各客户端上则对外提供某种信息或数据服务，如Web，E-mail，OLTP(在线事务处理)等。一个典型的基于NFS协议的附网存储系统(network-attached storage，NAS)如图1所示。

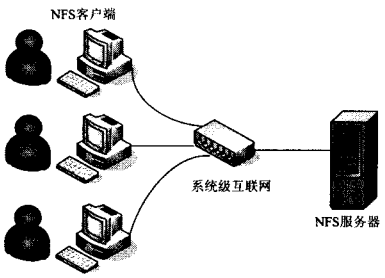

图1 典型NFS存储系统架构图

NFS存储服务器的部分系统参数如表1所列。以NFSD参数为例，该参数是指服务器启动的NFS服务进程数，当系统并发等级不同时，该参数需作相应的调整，过于保守的设置会导致应用的平均响应时间过长，而过于激进的参数值则可能导致系统资源的滥用以及服务等级目标违规等问题。又如，NFS服务器的后备存储设备通常为磁盘阵列即RAID。RAID的级别、磁盘数、条带单元大小等参数的不同配置则会导致不同的服务器供应性能。NFS客户端的参数以及系统中的其它参数均对系统性能有不同程度的影响，因篇幅所限，具体参数将在下面的性能调优方法部分加以阐述。

表1 NFS服务器部分参数描述

参数	类型	参数描述
NFSD	定量参数	NFS服务进程数
N	定量参数	NFS服务器端的磁盘个数
wdelay	分类参数	延迟同步写入
RQL	定量参数	请求队列长度
async	分类参数	异步数据写入
	…	

3.2 性能调优方法及算法

本文的性能调优方法由两个子阶段组成：关键系统参数的识别和关键参数的性能优化。

调优问题的形式化定义　在某种应用场景下，任何一个计算机系统的性能均可表示为系统参数的映射，即 $f:K\rightarrow P$。其中 P 指存储系统的性能，如吞吐量、响应时间，即 $P=\{Tp, Rt\}$，其中 Tp 代表系统吞吐量(throughput)，Rt 代表系统平均响应时间(average response time)(下文简称“响应时间”)；参数集 $K=\{K_1,K_2,\cdots,K_i,\cdots,K_n\}$，其中 K_i 代表系统的某个配置参数，如NFS服务器端的NFSD进程数。存储系统的性能调优问题就归结为在满足指定约束条件的前提下，寻找一组或多组系统配置参数，以使应用的性能达到最优值。

阶段一(基于方差分析(ANOVA)的关键系统参数识别)

在NFS存储系统中，当客户端作为网络服务平台运行某种重要应用时，因NFS存储系统参数众多，为识别出对系统性能有显著影响的系统参数，首先必须对系统各参数进行灵敏度分析。根据NFS存储系统配置参数多样的特性，且根据测量数据无法直观地判断某一参数是否会对性能结果产生显著性差异，本文选用方差分析(ANOVA)来建模系统参数的性能灵敏度。为满足ANOVA分析法的等方差前提假设，本文的实验测量均在相同的测量条件下独立进行。

假设NFS存储系统的某系统参数 K_i 的取值有 m 个水平，然后针对每一取值进行 n 次独立实验，获得系统性能的观测值矩阵：

$$P_{m\times n}=\begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mn} \end{pmatrix} \tag{1}$$

然后分别计算以下统计量：$T_i=\sum_j p_{ij}$ 为行向量元素之和，$i=1,\cdots,m, j=1,\cdots,n$；总实验次数 $N=m\times n$；每个行向量的元素之和的平均值为 $A_i=T_i/n$；性能矩阵所有元素的总和 $T=\sum_i T_i$；性能的总平均值为 $\bar{p}=\sum A_i/m$；总变差 $S_T=\sum_i\sum_j(p_{ij}-\bar{p})^2$；效应平方和 $S_A=\sum_i\sum_j(A_i-\bar{p})^2$；误差平方和 $S_E=\sum_i\sum_j(p_{ij}-A_i)^2$；$S_T=S_E+S_A$；自由度分别为 $f_T=N-1, f_A=m-1, f_E=f_T-f_A$；均方 $\bar{S}_A=S_A/f_A$ 和 $\bar{S}_E=S_E/f_E$。然后计算指定显著性水平 α 和置信度 $1-\alpha$ 下的 F 值，并与 F 分布查表值进行比较，从而判断该系统参数对系统性能的影响是否显著。

在本步骤中，由于不同存储系统的规模和复杂程度也不尽相同，所涉及系统参数的个数可能相差很大，因此为缩短实验时间，提高实验效率，可借助其它相关工具使系统参数灵敏度分析过程自动进行[15]。

阶段二(基于响应面分析(RSM)的关键参数性能优化)

在阶段一识别出对系统性能有显著影响的关键参数的基础上，余下的工作为进行系统参数的性能调优分析。在存储系统中，系统的定量参数多为离散型，在此可利用响应面分析法(RSM)来考察各系统参数对应用性能的影响，从而找到使应用性能达到最优的系统参数组合。

RSM法通常用于分析多个分类自变量和(或)多个定量自变量所产生的响应因变量，目的在于找出可获得最佳响应值的自变量参数组合。根据实验所得的响应面分析图能直观明了地反映各个因子对实验结果(响应)的影响，在存储系统中，则可用于找出最佳系统配置，以使应用性能达到最优。根据前述性能调优问题的形式化描述，当某种应用运行于NFS存储系统之上时，系统性能可表示为系统参数的函数，即 $P=F(K_1,\cdots,K_i,\cdots,K_l)$，其中 K_1 到 K_l 为灵敏度分析阶段所识别出的 l 个关键性能参数。其中每个参数的水平值个数分别为 $m_1,\cdots,m_i,\cdots,m_l$，即参数因子组合状态总数为 $M=m_1\times m_2\times\cdots\times m_l$，然后根据性能参数组合样本进行实验，实际测量出应用的性能。在RSM方法中，参数样本点和性能响应样本点之间的关系可用含有线性项、平方项和交叉项的二次响应面函数来近似，响应面函数的一般形式如方程(2)所示，其中 ε 为误差项。根据回归分析，可求得响应面函数中的常数因子 $\beta_0,\beta_i,\beta_{ij}$ $(i=1,\cdots,l; j=1,\cdots,l)$ 的最小二乘估计值，以后对系统的性能分析即可用响应面函数来代替真实的实验，以降低实验成本和缩短实验周期。

$$P=\beta_0+\sum_{i=1}^{l}\beta_i K_i+\sum_{i=1}^{l}\sum_{j=i}^{l}\beta_{ij}K_iK_j+\varepsilon \tag{2}$$

因NFS存储系统的关键参数组合状态空间还在可控的范围内，所以可针对每个参数的每个水平值进行实验，而当该

方法应用于组合状态很大的系统时，为提高实验效率，可先执行实验规划。图 2 为 NFS 系统在 Fileserver 负载下设定其它系统参数时的一个性能响应曲面实例。其中 x 轴、y 轴和 z 轴分别表示数据块读写大小 C(单位：2^C 字节)、NFSD 进程数和吞吐量(单位：MB/s)，该实例显示了数据块读写大小以及 NFSD 服务进程数对系统吞吐量的影响。

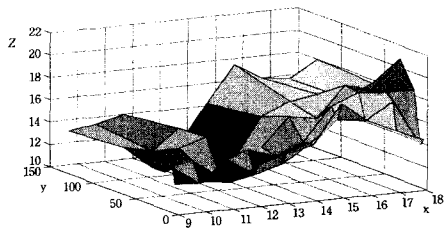

图 2　系统性能响应曲面图

3.3　调优算法

综合以上两个子阶段，在此给出 NFS 系统性能调优算法，步骤如算法 1 所示，其中的实验执行部分采用 bash 脚本语言实现，通过该算法即可找出性能最佳的系统参数组合。

算法 1　存储系统性能调优算法

Phase 1　关键系统参数识别

初始化：设定显著性水平 α 和 F 分布表。

步骤 1　取某系统参数 K_i 的 m 个水平值，就每一水平值进行 n 次独立实验，总共进行 $m\times n$ 次实验；

步骤 2　得到性能观测值矩阵 $P_{m\times n}$；

步骤 3　计算统计量 T_i 和 T；

步骤 4　计算总变差 S_T、效应平方和 S_A 以及误差平方和 S_E；

步骤 5　计算自由度 f_T，f_A 和 f_E；

步骤 6　计算给定置信度 $1-\alpha$ 下的 F 值，查 F 分布表的表值 F_α，若 $F>F_\alpha$，则可知 K_i 为影响系统性能的关键参数，否则为非关键参数；

步骤 7　返回步骤 1 检查下一参数 K_{i+1} 的系统性能灵敏度。

Phase 2　关键参数性能优化

步骤 1　取关键参数组合 $K_i=\{K_{1*},\cdots,K_{i*},K_{l*}\}$ 进行实验，测量得到对应的性能响应值 P_i；

步骤 2　根据回归分析求解响应面函数中的常数项 β_0，β_i，β_{ij} 的最小二乘估计值；

步骤 3　计算响应面函数值 $P=\beta_0+\sum_{i=1}^{l}\beta_iK_i+\sum_{i=1}^{l}\sum_{j=i}^{l}\beta_{ij}K_iK_j$；

步骤 4　绘制响应面，找出与性能峰值对应的系统参数组合。

4　实验验证

本节首先描述了实验环境及其配置，然后针对 Web，E-mail，Fileserver，Linux 实用程序以及微基准测试等多种重要应用场景，通过实验评价并比较了调优前后的性能结果，从而证实了本文调优方法的有效性和实用性。

4.1　实验环境及其配置

本文实验中所采用的 NFS 服务器和客户端的软硬件配置如表 2 所列。

表 2　实验环境配置

	NFS 服务器	NFS 客户端
CPU	Intel Pentium E2160 1.8GHz	Intel Pentium4 2.8GHz
内存	1GB DDR2 SDRAM	512M DDR SDRAM
磁盘	WD1600YS-01SHB1	ST3200827AS
网络接口	RTL8168/8111C(P)千兆以太网卡	主板集成千兆以太网卡
操作系统	Fedora 7 (内核版本：2.6.21)	Red Hat Linux 9 (内核版本：2.4.20)
基准测试程序	Filebench 以及各种 Linux 实用程序	

其中，NFS 服务器和 NFS 客户端之间采用千兆以太网互联；磁盘 WD1600YS-01SHB1 为 Western Digital 的 7200RPM/16MB 缓存/160GB 容量的 SATA-II 接口磁盘；ST3200827AS 为 Seagate 的 7200RPM/8MB 缓存/200GB 容量的 SATA-II 接口磁盘。

4.2　典型应用场景的性能调优结果

下面给出客户端运行多种典型应用时，NFS 存储系统调优前后的性能结果，并对实验结果进行了相应的分析和讨论。其中 Web，E-mail，Fileserver 以及 multistreamread 和 multistreamwrite 微基准测试采用学术界认可的基准测试程序 Filebench[13] 进行模拟，该测试套件由 Sun 公司开发，用于测试存储系统在各种典型服务器应用和各种微基准测试下的性能。另外，各实验独立地重复执行 3 次，然后取测量结果的平均值。

4.2.1　Web，E-mail，Fileserver 应用

Web 服务器应用的环境设置为操作同一个目录树下的多个文件，操作种类包括常规文件的 open/read/close 操作和 Web 日志文件的 append 操作，实际的文件操作为这些操作的混合体，访问线程数为 100 个，每 10 个线程产生一个 16kB 大小的 Web 日志追加操作；在本实验中采用 varmail 来模拟多线程 E-mail 负载，varmail 负载操作单个目录下的文件，文件操作包括 3 类：open/read/close，open/append/close 和 delete；而代表文件服务的 Fileserver 负载则与早期的 SPECsfs 基准测试程序相似，在一个多层次的目录结构下执行一系列 create，delete，append，read，write 操作以及文件系统的元数据操作。系统调优前后的性能测试结果如图 3 所示，(a)和(b)分别表示吞吐量和响应时间。

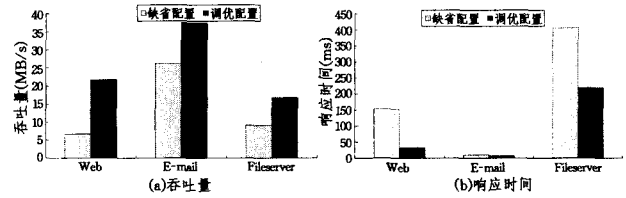

图 3　3 种服务器应用调优前后的性能

采用文中调优方法筛选出的 Web 服务性能参数组合为〈nfsd=256，rsize=wsize=16kB，async，wdelay，acregmin=300，acregmax=300，acdirmin=300，acdirmax=300，noatime=off〉。而识别出的 E-mail 服务的性能参数组合为〈nfsd=256，rsize=wsize=8kB，async，wdelay，acregmin=20，acregmax=20，acdirmin=20，acdirmax=20，noatime=on〉。调优后 Fileserver 服务的最佳性能参数组合则为〈nfsd=32，rsize=wsize=256kB，async，wdelay，acregmin=30，acregmax=30，acdirmin=30，acdirmax=30〉。从图 3 可以看出，Web 服务的吞吐量由调优前的 6.73MB/s 提高到调优后的 21.75MB/s，提高了 2.23 倍，而响应时间由调优前的 153.2ms 减少到调优后的 32.9ms；E-mail 服务的的吞吐量由 26.3MB/s 提高到 37.6MB/s，提高了 43%，而响应时间则由 9.3ms 减少到 6.6ms；文件服务 Fileserver 的吞吐量则由调优前的 9.1MB/s 提高到 16.9MB/s，同时响应时间由调优前的 405.9ms 减少到

219.4ms。从实验结果可知调优后3种典型服务器应用的吞吐量均得到了大幅提高，而响应时间则显著减少。由此可见，调优后系统吞吐量得到提高的同时，响应时间反而降低了。

4.2.2 Linux实用程序和微基准测试的性能

为保证性能测试的广泛性，除前述采用典型服务器应用来比较调优前后的性能之外，在此还采用Linux实用程序tar和find以及multistreamread和multistreamwrite两个微基准测试程序来评价调优前后的性能差异。实验结果如表3所列（其中m表示分钟，s表示秒）。实用程序tar用来解压Linux的内核源码包linux-2.6.31.6.tar.bz2，统计其总执行时间。该源码包大小为58.5MB，解压后大小为525MB，总文件个数为29032。性能调优后的系统参数组合为〈nfsd=16，rsize=wsize=16kB，async，wdelay，noatime=on，udp，acregmin=60，acredmax=60，acdirmin=60，acdirmax=60〉；find命令用来查找Linux源码文件中的某个字符串，性能调优后的系统参数组合为〈nfsd=4，rsize=wsize=16kB，async，wdelay，noatime=on，udp，acregmin=10，acredmax=10，acdirmin=10，acdirmax=10〉；multistreamread微基准测试程序用来测试多个并发I/O流的读性能，性能调优后的系统参数组合为〈nfsd=256，rsize=wsize=256kB，async，wdelay，noatime=on，acregmin=110，acredmax=110，acdirmin=110，acdirmax=110〉；multistreamwrite微基准测试程序用来测试多个并发I/O流的写性能，性能调优后的系统参数组合为〈nfsd=256，rsize=wsize=256kB，async，wdelay，noatime=on，acregmin=0，acredmax=0，acdirmin=0，acdirmax=0〉。由表3可知，tar程序的总执行时间缩短了54.3%；find命令的查找时间减少了19.8%；multistreamread微基准程序的吞吐量提高了31.7%，而响应时间则减少了23.4%；尽管multistreamwrite微基准程序的响应时间只减少了15.3%，但其吞吐量依然提高了49.2%。由此可见，NFS存储系统调优后的性能比缺省条件下的性能均有显著提高。

表3 Linux实用程序和微基准测试的性能结果

测试程序	缺省配置下的性能	调优配置下的性能
tar	20m9s	9m12s
find	4m58s	3m59s
multistreamread	375.5MB/s；105.3ms	494.7MB/s；80.7ms
multistreamwrite	31.7MB/s；38.6ms	47.3MB/s；32.7ms

综合以上各种应用场景下的实验结果可知，采用本文方法调优后，NFS存储系统的性能均得到了大幅提升，调优后的系统配置甚至能以较低的响应时间获得较高的吞吐量；同时实验结果也证实了本文调优方法的有效性和实用性。

结束语 存储系统性能调优可大幅提升应用程序的性能，因影响系统性能的参数众多，且各参数对性能贡献各异，使得研究存储系统的性能调优问题具有非常重要的理论意义和应用价值。本文以NFS网络存储系统为例，提出了一种基于统计分析的存储系统性能调优方法，该方法首先通过方差分析建模系统参数的性能灵敏度，识别出对系统性能有显著影响的关键参数；然后在此基础上，通过响应面分析法来考察不同关键参数配置时的性能响应，并给出了系统性能调优算法，通过该算法找出最优的系统配置，从而达到性能调优的目的。最后，本文通过实验验证了文中方法在Web，E-mail，Fileserver，Linux实用程序、微基准测试等多种典型应用场景下的性能调优结果，实验结果证实了本文调优方法的有效性和实用性。

在关键系统参数的识别和优化过程中，由于本文方法在系统复杂度较高时需执行大量实验，因此在后续工作中将研究如何提高最佳参数组合的识别以及性能分析效率。此外，在当前基于虚拟化技术的云计算模式下，当多个应用同时运行于公用计算平台时的性能调优问题也值得进一步研究。

参考文献

[1] Bhadkamkar M，Guerra J，Useche L，et al. BORG：Block-reORGanization for Self-optimizing Storage Systems[C]// San Francisco，CA，USA. Proceedings of the 7th USENIX Conference on File and Storage Technologies(FAST'09). Berkeley，CA，USA：USENIX Association，2009：183-196

[2] Chaudhuri S，Narasayya V. AutoAdmin "What-if" Index Analysis Utility[C]// Seattle，WA，USA. Proceedings of the International Conference on Management of Data (SIGMOD'98). New York，NY，USA：ACM，1998：367-378

[3] Chung I-H，Hollingsworth J K. Automated Cluster-Based Web Service Performance Tuning[C]// Honolulu，HI，USA. Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC'04). Washington DC，USA：IEEE，2004：36-44

[4] Debnath B K，Skarie J，Lilja D J，et al. SARD：A Statistical Approach for Ranking Database Tuning Parameters[C]// Cancun，Mexico. Proceedings of the 3rd International Workshop on Self-Managing Database Systems (SMDB'08). Washington，DC，USA：IEEE，2008：11-18

[5] Dias K，Ramacher M，Shaft U，et al. Automatic Performance Diagnosis and Tuning in Oracle[C]// Asilomar，CA，USA. Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR'05). New York，NY，USA：ACM，2005：84-94

[6] Duan S，Thummala V，Babu S. Tuning Database Configuration Parameters with iTuned[C]// Lyon，France. Proceedings of the 35th International Conference on Very Large DataBases (VLDB'09). New York，NY，USA：ACM，2009：1246-1257

[7] Ellard D J. Trace-Based Analyses and Optimizations for Network Storage Servers[D]. Cambridge，MA，USA：Harvard University，2004

[8] Karve A，Kimbrel T，Pacifici G，et al. Dynamic Placement for Clustered Web Applications[C]// Edinburgh，Scotland. Proceedings of the 15th International Conference on World Wide Web (WWW'06). New York，NY，USA：ACM，2006：595-604

[9] Kephart J O，Chess D M. The Vision of Autonomic Computing [J]. Computer，2003，36(1)：41-50

[10] Oracle Corporation. Oracle Database Performance Tuning Guide 10g. 2008

[11] Soror A A，Minhas U F，Aboulnaga A，et al. Automatic Virtual Machine Configuration for Database Workloads[C]// Vancouver，Canada. Proceedings of the International Conference on Management of data (SIGMOD'08). New York，NY，USA：ACM，2008：953-966

[12] Sullivan D G，Seltzer M I，Pfeffer A. Using Probabilistic Reasoning to Automate Software Tuning[C]// New York，NY，USA. Proceedings of the Joint International Conference on Measure-

ment and Modeling of Computer Systems (SIGMETRICS'04). New York, NY, USA: ACM, 2004: 404-405

[13] Sun Microsystems. Filebench. http://hub.opensolaris.org/bin/view/Community+Group+performance/filebench. 2009

[14] Uttamchandani S, Yin L, Alvarez G A, et al. CHAMELEON: A Self-Evolving, Fully-Adaptive Resource Arbitrator for Storage Systems[C]// Anaheim, CA, USA. Proceedings of the USENIX Annual Technical Conference (USENIX'05). Berkeley, CA, USA: USENIX Association, 2005: 75-88

[15] Wright C P, Joukov N, Kulkarni D, et al. Auto-pilot: A Platform for System Software Benchmarking[C]// Anaheim, CA, USA. Proceedings of the Annual USENIX Technical Conference, FREENIX Track. Berkeley, CA, USA: USENIX Association, 2005: 175-187

[16] Zheng W, Bianchini R, Nguyen T D. Automatic Configuration of Internet Services[C]// Lisbon, Portugal. Proceedings of the EuroSys Conference (EuroSys'07). New York, NY, USA: ACM, 2007: 219-229

[17] 赵晓南，曾雷杰，李战怀. 一种基于块级的存储性能优化方法[J]. 计算机科学，2009，36(6)：129-137

(上接第 288 页)

各模块的功能如图 5 所示。

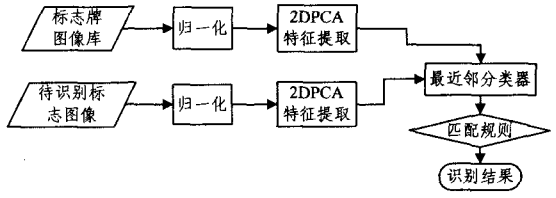

图 5 基于 2DPCA 的交通标志牌识别框图

我们进行了两个标志牌图库的识别。首先选取第一个交通标志图库(即我们选取的禁令和限速的总共 20 种交通标志，经过仿真变换后形成 1240 个样本集)，该图像库包含 20 个不同的交通标志，每个交通标志包括 62 个交通标志样本，样本的分辨率为 60 * 60。选取每个交通标志的前 31 幅图像作为训练样本，后 31 幅作为测试样本，那么训练和测试的样本都为 620 幅。经过预处理后，采用二维主成分分析进行交通标志的特征提取。这里对于主分量个数 d 的选取将直接影响到识别率和运行时间的大小。为了选择较好的主成分个数 d，我们进行了不同主成分个数下的交通标志牌识别率的实验，其结果如图 6 所示。

从图中不难看出，当主成分个数较少时，正确识别率的波动较大；当 $d=10$ 时，正确识别率不再变化。所以我们选取的主成分个数为 10，当 $d\geqslant 10$ 时，对于我们的第一个交通标志数据库，其识别率都维持在 100%。

第二个交通标志数据库(即我们实地拍摄的交通标志，基本类型如图 3 所示，每种交通标志有 40 个不同的样本，总共 400 幅图像)包含 10 个不同的交通标志，每个交通标志包括 40 个交通标志样本，样本的分辨率为 60 * 60 的灰度图。选取每个交通标志的前 20 幅图像作为训练样本，后 20 幅作为测试样本，那么训练和测试的样本都为 200 幅。经过预处理后，采用二维主成分分析进行交通标志的特征提取。采用同样的处理方式，选择主成分个数 d 从 1 到 15，得到的识别率的变化结果如图 7 所示。

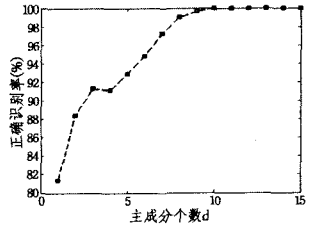

图 6 第一个数据库不同主成分个数的交通标志识别率曲线

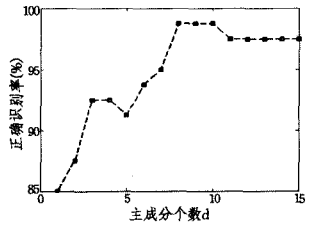

图 7 第二个数据库不同主成分个数的交通标志识别率曲线

结束语 二维主成分分析方法直接在图像矩阵上进行计算，更加方便和简化了图像的特征提取，具有较高的训练速度和较高的识别率。本文将二维主成分分析方法应用到交通标志牌识别中进行特征提取，并采用最近邻分类器作为交通标志牌识别的分类器。通过实验可以发现，对于同一幅图片进行仿真变换后的数据库，其识别率最高可以达到 100%。然而，交通标志检测作为智能车辆项目的辅助手段，让我们识别的不可能是针对同一交通标志进行变换的情况，而是车载条件下实际拍摄的交通标志。对于我们按照不同环境、不同状况拍摄的实景图，其最高识别率能达到 98.75%。目前实际采集的图像库只是针对晴天的不同场景位置状况进行的拍摄，而对于一些实际状况不是很好的情况，如阴天、遮挡等，识别率就会降低。所以下一步要尽量寻找一种对于恶劣条件下保持较好稳定性的特征，按照不同角度、不同实际条件尽可能多地拍摄交通标志图像，以增强识别的成功率。

参 考 文 献

[1] Paclik P. Road Sign Recognition Survey[EB/OL]. Available: http://euler.fd.cvut.cz/research/rs2/files/skoda-rs-survey.html

[2] Turkm A, Pentland A D. Face recognition using eigenfaces[C]// Proc. of Computer Vision and Pattern Recognition. 1991: 586-591

[3] Pentland A, Moghaddam B, Starner T, et al. View-based and modular eigenspaces for face recognition[C]// Proc. IEEE Computer Soc. Conf. on Computer Vision and Patt Recog. 1994: 84-91

[4] Yang J, Zhang D. Two-dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition [J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 2004, 26(1): 131-137

[5] 陆晓峰，朱双东. 基于 BP 网络分类器的交通标志牌识别[J]. 宁波大学学报：理工版，2007，20(3)：281-284

[6] Lafuente-Arroyo S, Gil-Jiménez P, Gómez-Moreno H. Road-sign Detection and Recognition Based on Support Vector Machines [J]. IEEE Transactions On Intelligent Transportation Systems, 2007, 8(2): 264-278

[7] 桑海峰，苑伟琦. 基于二维主成分分析的掌纹识别研究[J]. 仪器仪表学报，2008，29(9)：1929-1933