通用CPU性能基准测试研究综述

史惠康[1],王泽胜[2],张士宗[2],高 翔[3],赵有健[1]
(1. 清华大学计算机科学与技术系,北京 100084; 2. 中国电子技术标准化研究院,北京 100007;
3. 龙芯中科技术股份有限公司,北京 100095)

摘 要: CPU性能基准测试旨在给出可对比、定量的指标数据,为产品选型提供依据,它已成为引领计算产业发展的风向标之一. CPU技术发展迅速,性能基准测试也在不断演进. 本文对包含SPEC CPU在内的主流基准测试进行了研究,从测试目标、测试方法等角度,综述主流CPU基准测试的演进过程、最新研究成果,以及通用CPU性能指标和基准测试需求,分析了通用CPU性能基准测试所面临的挑战,并对今后可能的研究趋势进行了展望.

关键词: 通用CPU;测试基准;性能测试;评价指标;基准测试程序集
中图分类号: TP306 **文献标识码:** A **文章编号:** 0372-2112(2023)01-0246-11
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220169

Performance Evaluation Benchmark of General-Purpose CPU:A Survey

SHI Hui-kang[1],WANG Ze-sheng[2],ZHANG Shi-zong[2],GAO Xiang[3],ZHAO You-jian[1]
(1. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*;
2. *China Electronic Standardization Institute, Beijing* 100007, *China*;
3. *Loongson Technology Corporation Limited, Beijing* 100095, *China*)

Abstract: CPU performance evaluation benchmark aims to provide comparative and quantitative index data for product selection. It is one of the vane leading the development of computing industry, and as CPU technology evolves rapidly, performance benchmarks are evolving. This paper systematically reviews the mainstream benchmarks including the SPEC CPU. From the perspectives of evaluation objectives and methods, the evolution, recent research results of the mainstream CPU benchmarks, and the performance metrics and benchmark requirements of general-purpose CPU are reviewed. Finally, this paper analyzes the challenges of general-purpose CPU performance evaluation benchmarks and prospects for possible future research trends.

Key words: general-purpose CPU; benchmark; performance evaluation; evaluation indicator; benchmark suites

1 引言

在计算产品测试领域,基准测试常用于评估软硬件的性能[1]. 通过运行一组或多组可重复的标准测试程序,基准测试能够评估被测对象性能的优劣,给出可对比、可衡量的指标数据,为产品选型、提升质量、指导研发提供依据. 当前,计算产品性能基准测试凭借其基础性、普适性等显著优势,已成为引领CPU和计算机整机等计算产品性能发展的风向标,被广泛应用在计算产业的各领域. 如文献[2]利用性能基准测试提升机器学习算法效率;文献[3]基于性能基准测试结果,指导系统架构完善;文献[4]通过给出硬件基础性能、功耗,以及面积和准确度等综合指标,优化智能芯片的设计;文献[5]通过基准测试为数据中心与计算集群的节能和均衡优化调度提供依据等.

CPU是信息技术产业的核心基础元器件,对其性能进行测试至关重要[6]. 通过设置不同使用场景和关键性能指标,CPU性能基准不仅可以帮助芯片厂商发现问题瓶颈、提升产品能力,而且可以帮助行业建立公开透明的评价准则、促进优胜劣汰,进而带动CPU产业整体水平和竞争力提升,加速技术创新. “斯诺登事件”以来,信息基础设施自主可控逐渐受到各国的重视,研发推广X86架构之外的通用CPU,例如ARM,RISC-V,Alpha,MIPS等指令集架构,已成为推动信息产业变革发展的主要路径之一. 近年来,采用X86,ARM等不同指令集架构,常用于服务器和桌面计算的异构通用CPU加速迭代升级,不同架构CPU在功耗、适用场景等

收稿日期:2022-02-15;修回日期:2022-10-30;责任编辑:宋甲英

方面各有优势[7,8]. 但主流CPU性能测试工具与方法主要围绕X86架构芯片设计,如何对异构通用CPU的质量水平进行科学评价,成为引导技术突破、支撑重大信息化工程建设和促进行业发展的关键. 围绕异构通用CPU开展性能基准测试研究,也逐渐成为学术界和产业界关注的重点[9,10].

近年来,面向通用CPU的性能基准测试发展迅速,有学者针对特定的基准测评工具涉及的相关参数、测试场景等进行了归纳. 如文献[11]对SPEC CPU 2006基准测试程序组件集进行了研究,并分析了评价指标及使用方法. 文献[12]讨论了在嵌入式系统中开展基准测试的基本方法,分析了对比测试原理、测试环境的构建以及主要的测试过程. 许多学者针对不同程序合成方法、测试算法等开展了前沿研究. 如文献[13]围绕深度神经网络学习和加速优化的基准方法,梳理了当前存在的技术挑战和未来发展趋势. 文献[14]基于多核系统性能优化,通过线程级测试方法构建了一种面向多核系统的测试基准. 文献[15]对各类深度学习加速器进行梳理,并在此基础上提出了一种适合多场景的鲁棒测试基准和测试方法.

不同于以上文献,本文从测试目标、测试方法的角度综述了CPU基准测试的发展现状和趋势,并对测试工具的演进和最新成果进行对比分析,旨在为研究者提供一个覆盖SPEC、TPC等多类性能基准测试工具和多线程、跨平台等各类场景,以及包含速度和速率性能指标分类、基于预置模型的测试结果修正等创新方法的说明,增加相关人员对通用CPU性能基准测试研究的理解,并使其得到启发.

2　性能基准测试及其演进

性能基准测试的目标是提供一种定义并计算产生一系列量化指标数值的基础通用方法,手段是通过运行基准测试程序获得相关指标的评分,以此来比较不同CPU、应用程序乃至不同体系结构软硬件产品的性能,实现CPU及计算机整机之间的直观性能比较.

自20世纪60年代以来,性能基准测试程序就被视为CPU和计算机整机性能的一种重要测试对比工具[16]. 最初的性能基准测试程序仅以简单的加、乘等指令作为指标. 20世纪80年代,可以衡量整型及浮点型计算能力的小型性能基准测试受到各界的广泛关注[17]. 然而,这类性能基准测试程序由于定义宽泛而逐渐被边缘化,标准性能评估机构(Standard Performance Evaluation Corporation, SPEC)、事务处理性能委员会(Transaction Processing Performance Council, TPC)等专业化性能基准测试组织成立后,CPU相关产品的性能基准测试才逐步确定,并发展成为学术界和产业界公认的事实准则[18]. 目前,性能基准测试程序已基本形成了技术指标体系化、被测产品多样化的格局,可满足用户对不同维度性能测试的需求,包括计算能力、二维和三维图形处理能力、多媒体处理能力、大数据处理能力、多线程能力等.

2.1　通用CPU性能指标及其基准测试需求

CPU性能表现受诸多因素影响,包括结构参数、接口参数、物理参数以及多核参数等[19],如图1所示. 在各类参数中,核心数量、生产工艺、主频、缓存大小等是决定CPU计算能力的直接因素,而多线程能力、指令调度能力乃至指令集类型同样对CPU的整体性能产生影响. 仅通过简单的参数对比来决定性能的方法存在局限性. 比如,由于CPU的内部结构不同,不能完全通过主频来对比CPU的性能,主要原因是在并行计算需求不断增长的趋势下,多核计算也成为影响CPU整体性能的重要因素. 考虑到对CPU进行孤立测试以获取其性能表现的方式仅适用于生产环境,且实际参考价值有限,通常基准测试程序均采用面向部分应用场景,综合多类基准测试指标的集成测试的方式对计算系统进行整体评估,进而有针对性地反映CPU的性能[7].

纵观CPU指令集架构发展史,工程驱动的软硬件生态建设、用户群体规模等决定了CPU应用的广度和深度. 目前,多指令集共存并行发展已成为常态[20],这给工程应用带来了一定的困难. 此外,为推动CPU加速创新发展,兼具前瞻性的应用场景也成为CPU性能基准测试的关键考虑因素,这主要体现在5G、人工智能、图像计算、自动驾驶、物联网,以及CPU与GPU/FPGA集成应用的异构计算等层出不穷的新技术,对CPU性能提出了全新的需求. 为适应新的变化,CPU性能基准测试从最初的仅关注裸性能发展为关注裸性能、系统性能等各维度,各类测试工具也应运而生. 本文有关通用CPU性能基准测试综述的整体结构如图2所示.

2.2　常用的性能基准测试

2.2.1　SPEC

SPEC性能基准测试于1988年由标准性能评估机构SPEC提出,目前已发展成为包含CPU性能、服务器能效、文件系统性能、高性能计算、Web应用性能等在内的基准测试簇[21]. 其中SPEC CPU系列基准是公认的、具有事实性影响力的CPU性能基准测试标准,通过测试程序在被测系统和基准系统中执行时间的比值来考察系统CPU运算性能[22],原理如图3所示.

多年来,SPEC CPU性能基准测试与CPU的发展相互促进,基准测试程序不断演进升级,CPU技术和产业应用也加速创新发展. 20世纪90年代初,整型运算和浮点型运算的分化应用,使得传统的MIPS度量(单字长定点指令平均执行速度)的指导性大幅降低,难以形

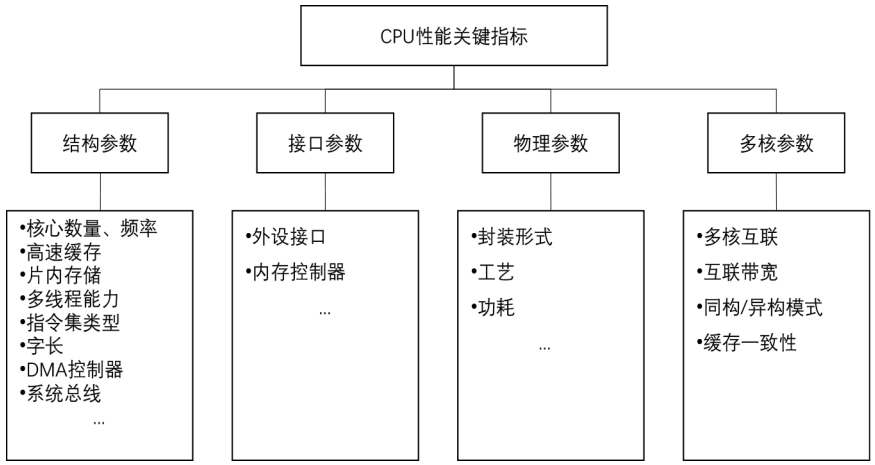

图1 CPU性能关键指标

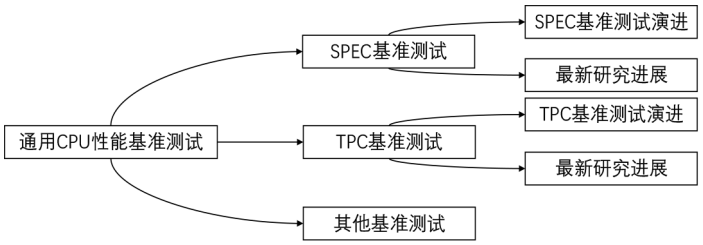

图2 基准测试综述结构图

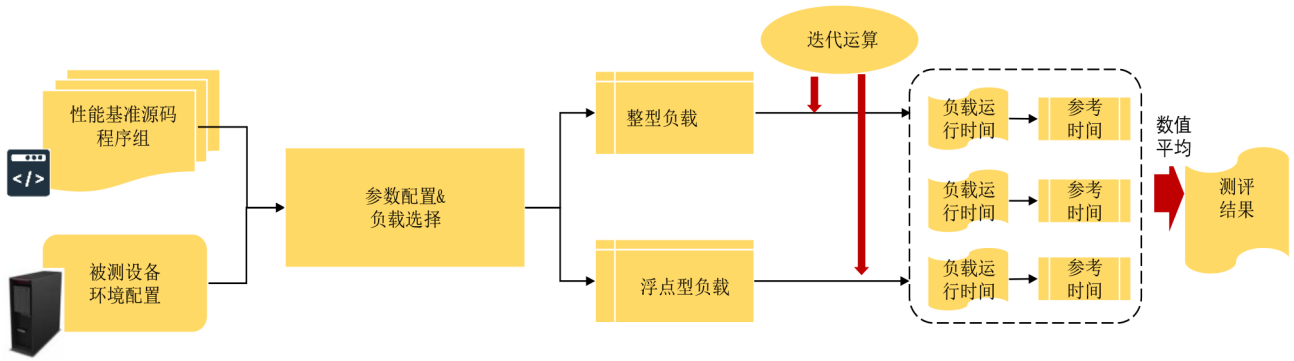

图3 SPEC CPU原理流程

成对CPU技术和产品的有效规范. SPEC CPU 92通过调整测试基准来应对行业需求,迅速取代MIPS度量,成为产业界公认的事实标准[23],也驱动各界在提升整型运算和浮点型运算的同时,积极寻求新的CPU技术突破方向. 随着技术的创新迭代,CPU时钟频率加速攀升,高速缓存容量持续增大、性能不断提高,也使得SPEC CPU 92的性能测试受到较大的影响[24]. 为进一步提升测试准确性,业界推出了SPEC CPU 95,在应对新的容量、性能测试需求时,提供了更全面的场景来丰富CPU性能测试能力,引导技术创新由局限于关注裸性能向注重实际场景应用效果转变. SPEC CPU 2000延续了SPEC CPU 95由两套基准测试程序分别测试CPU整型运算性能和浮点运算性能模式. 同时,为更好地应对不断普及的多核处理器计算系统测试需求,保证测试结果的科学性和合理性,SPEC CPU 2000分别设置了不同的度量方法针对单核处理器和多核处理器计算系统进行测试. 其中,单核处理系统主要测量系统的运算速度指标,即单位工作量需要多少时间来完成;而多核处理器系统则主要测量系统的吞吐量,即系统在给定时间内能完成多少工作量. 为保证测试结果的公正性,SPEC CPU 2000还引入了MD5数字签名,在编译和运行程序时,支持产生并验证可执行测试程序文件和测试结果的校验,从而确保测试结果来自基准程序测试输出,而未经过第三方篡改,这大大提高了测试结论的可信度[25]. C语言、C++语言等编程语言的不断丰富,催生了对应的编译器的多样化发展,进一步促使CPU的配置系统环境也逐渐呈现多样性特征. 同时,计算密集型应用和跨硬件使用的需求也不断增加[26]. 为了满足

以上场景的测试要求,SPEC再次对测试基准进行完善,推出了SPEC CPU 2006. SPEC CPU 2006新增的测试套件涵盖到CPU、存储系统、编译器等,特别是编译器在延续了SPEC CPU 2000对Fortan和C语言覆盖的同时,更好地支持了C++语言. 近年来,CPU内存、缓存和虚拟内存尺寸的急剧增大,以及乱序执行和序列化等技术的不断成熟,又向CPU性能基准测试提出了新的挑战[27]. 经过10年的沉淀,在SPEC CPU 2006的基础上推出的SPEC CPU 2017,进一步丰富应用场景,具体包含43个基准,分为两类四个套件. 其中,SPEC speed Integer和SPEC speed Floating Point套件用于比较计算机完成单个任务的时间,SPEC rate Integer和SPEC rate Floating Point套件则可以测量每单位时间内的吞吐量或工作量. 这也是第一次对速率(rate)和速度(speed)进行区分,以有效满足复杂场景下对性能基准测试的稳定性提出更高要求,进而指导CPU发展.

SPEC CPU 2017虽然极大地丰富了基准测试场景和算法,但这也相应提升了基准测试本身操作的难度,对测试结果的准确性提出了挑战. 为此,学术界和产业界围绕性能基准表征、内存性能表征等基准测试涉及的关键环节开展了大量的研究. Song等人[28]对SPEC CPU 2017基准之间的相似性、冗余性,以及测试覆盖范围的平衡性等进行了研究,明确了SPEC CPU 2017的工作负载表现出明显的内存密集型特征,对内存提出了更高的要求,测试的有效性更强. Singh等人[29]首次给出了SPEC CPU 2017套件运行时的内存行为全面表征分析,通过使用动态二进制检测、硬件性能计数器和基于操作系统的统计工具等,对工作指令集大小、各种工作负载的内存容量消耗和内存带宽利用率进行了统计,实验结果显示相较于SPEC CPU 2006,SPEC CPU 2017在提高内存要求的同时,工作负载对内存带宽的消耗也有了明显的提升. Bucek等人[30]分析了SPEC CPU 2017基准在功耗数据收集、系统数据收集等方面的改进,并对其在测试指标计算方式和测试结果披露形式等方面的调整进行了系统梳理,明确SPEC CPU 2017虽然测试指标更丰富和复杂,但是通过进一步梳理测试指标的分类展示形式,提升了测试结果的可读性.

2.2.2　TPC

不同于SPEC从最初关注裸性能不断拓展到系统性能,TPC性能基准测试在设计之初就将系统级应用的综合性能测评作为关注的重点,测试实现方式如图4所示. 20世纪80年代,事务处理模式出现. 与20世纪70年代占统治地位的批量计算模式不同,事务处理模式采用相对单一的方式,直接通过在线数据库系统进行简单的事务处理[31]. 同时期,用于度量系统对该类事务处理性能的主要测试基准包括TP1(Transaction Process 1)和DebitCredit[32]. 该类测试基准由于缺少对测试执行过程和综合测评结果的有效监督,易出现测试过程不规范,甚至给出误导性测试结果的情况. 20世纪80年代末,第一个TPC基准TPC-A[33]发布,对事务处理时限、测试系统终端数量等提出明确要求,澄清了当时混乱的市场,为推动CPU的系统级性能测试提供了重要依据.

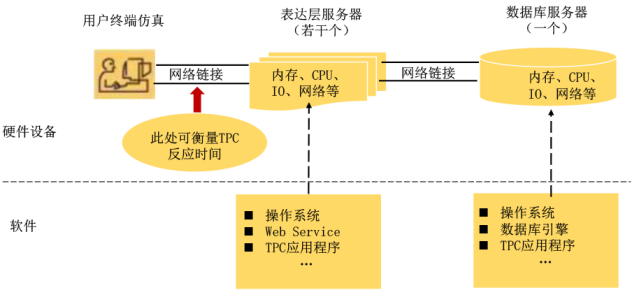

图4　TPC测试实现示意图

经过持续迭代升级,TPC已发展成为能够满足多种应用场景性能测试需求的基准测试簇,根据测试场景和测试事务的不同,可将TPC性能基准测试分为三类:联机在线事务处理系统(OLTP)测试,包括TPC-C,TPC-E;决策支持和大数据(DS)测试,包括TPC-H,TPC-DS;服务器虚拟化(VMS)测试,包括TPC-VMS. 其中,TPC-C性能基准测试通过模拟较复杂且具有代表意义的OLTP应用环境,来衡量联机事务处理系统性能与可伸缩性[34]. TPC-E则在TPC-C的基础上,对传统的C/S架构模拟环境进行了完善,从而实现对当时日益盛行的B/S架构系统的高效评价,为引导产业提升大规模并发处理能力提供了重要依据[35]. TPC-H提供了一套决策支持系统的性能基准测试依据,强调服务器在数据挖掘、分析处理方面的能力[36]. TPC-DS则补充了单用户响应时间、多用户吞吐量等测试,对测试基准的数据模型、业务模型和执行模式进行了完善[37]. TPC-VMS的目标是模拟服务器虚拟化环境,并实现对TPC-C,TPC-E,TPC-H,TPC-DS的综合测试[38],从而推动CPU围绕应用日益广泛的云计算模式不断提升性能.

为进一步改进TPC性能基准测试,许多学者围绕优化测试框架、简化工作流程、改善测试策略等进行了探索. 刘建鹏等人[39]为了提升TPC-DS的性能,将其与HiBench测试框架合并,实现了对系统性能的高效测试,为进一步改进决策支持类测试基准提供了很好的思路. 文献[40]针对TPC在采用SSD阵列的大型计算机中工作负载的复杂性问题,提出了一种基于Small-File表空间的方法,并通过构建位置感知的终端映射策略,有效提升了TPC-C在大规模评估系统中的适用性. 冯志丹[41]提出了一种压力自动摸顶机制,根据被测系

统执行事务的响应时延快速定位出最大吞吐量,在保证TPC-C测试准确性的同时,有效简化了测试流程,提升了测试基准的可操作性.

2.3 其他性能基准测试

在CPU性能基准测试的不同发展时期,学术界和产业界也提出了一系列有关的性能基准测试方法,针对跨平台、内存共享、多线程等多种场景进行测试.

Geekbench是近年来受关注较多的一种跨平台CPU性能基准测试,其主要方式是通过构建多维评分系统,将单核、多核性能与模拟真实场景的工作负载分隔开. 该性能基准测试适用于Windows、Linux、macOS等多种操作系统下的测试[42,43]. Kozhirbayev等人[44]在利用Geekbench对单核和多核下的整形计算性能、浮点计算性能和存储性能的基准测试中,获得了很好的对比参考效果. Polvinen等人[45]在实验过程中,采用Geekbench快速、准确地度量了处理器在向量点乘、矩阵LU分解等场景中的性能.

为优化Geekbench性能基准测试,许多学者也围绕测试体系、复杂模型的适应性进行了研究. Morabito等人[46]在虚拟机环境下,验证了Geekbench系统索引体系的完整性,以及对CPU存储性能测试敏感性方面的优越性. Barker等人[47]将Geekbench的单核和多核CPU性能基准测试的数据作为重要参考,对测试模型进行了创新和优化. Wang等人[48]通过详细分析CPU性能基准测试的内容、精度、误差等,研发了基于Geekbench的CPU性能基准测试新方法,对其使用场景进行了有效拓展.

此外,Splash,PARSEC Benchmark,LINPACK Benchmark,MiBench,NAS Parallel Benchmark,CPU-Z等也常用于计算实验或实际应用中的性能基准测试,来展示和对比CPU的各项性能. Singh等人[49]提出了基于共享内存的CPU性能基准测试方法——Splash,在真实多处理器场景和并行体系结构模拟器上验证了方法的有效性. Woo等人[50]定量分析了Splash-2的特性,分析了处理器数量和工作集之间的相互作用,并明确了测试程序和参数的主要调试策略,为促进集中式和分布式共享地址空间多处理器研究提供重要依据. Sakalis等人[51]提出了Splash-3,结合Splash-2在与不同版本编译器和硬件配合使用中暴露出的内存非一致性缺陷,给出了一组针对并行性能的基准测试套件,通过建立测试结果修正方法,有效提升了测试有效性. Bienia等人[52]对PARSEC进行了特性和结构分析,实现了对大规模多线程商业程序的识别、挖掘、合成和模拟,为多核CPU高性能测试做了有效的补充. Bienia等人[53]在多处理器上对PARSEC Benchmark和Splash-2两个多线程基准测试中的程序进行定量比较,为面向不同核数CPU开展测试时选择合适的测试程序提供重要参考. Barrow-Williams等人[54]结合PARSEC Benchmark和Splash-2测试的处理器通信时间和空间特性,提出了一种提升CPU在通信领域性能测试精度的方法. Bienia等人[55]深入研究了PARSEC基准输入的保真度和导入范围,提出了一种基于预置模拟输入集修正性能测试结果的方法,较好地获取了处理器的原始特性. Chasapis等人[56]面向PARSEC基准测试,通过对计算任务并行性的影响因素进行分析,提出了一种提升CPU并行处理性能测试精度的方法. Cebrian等人[57]研究了一种面向PARSEC基准测试的向量化分析方法ParVec,通过矢量化方法提升CPU任务执行效率和能耗之间的平衡性,为CPU性能的综合评价提供了新的思路. Huynh等人[58]基于PARSEC,给出了5种并行编程模型,在保证负载均衡的同时,有效提升了复杂任务并行性能基准测试效果. LINPACK Benchmark,MiBench,CPU-Z等常针对具体型号的CPU进行测试分析或作为工程环节进行研究,为工程实践中生产环境下CPU性能测试提供了重要参考[59,60].

3 性能基准测试的对比分析

为直观地展示本文所述通用测试基准的测试重点及主要特性,各类性能基准测试及相关工具的支持语言、编译程序、适配系统、支持的CPU架构及测试侧重点信息,如表1所示.

表1中各类工具支持语言、编译程序、适配系统、支持的CPU架构均具有一定的差异性. 特别是基于不同的测试目标,各类工具的重点测试内容各异,不仅涉及运算能力、内存性能、内存带宽等重要指标,而且与联机业务处理、数据挖掘、并行计算等各类综合应用场景相关. 随着CPU性能测试维度的增加,CPU裸性能以及单一场景下的基准测试,难以全面反映CPU的综合性能,因此,多种基准测试工具的配合测试,已成为通用CPU性能基准测试行业共识.

在稳定性方面,选取常用的性能基准测试工具——SPEC CPU和UnixBench,通过对实际应用情况的分析和梳理,为通用CPU性能基准的深入研究提供参考.

测试工具SPEC CPU的重要参数包含缓存缺失率(cache-misses)、分支指令预测错误率(branch-misses)和地址块表缓存缺失率(dTLB-load-misses)等,相关指标越大,表明CPU的测试强度越高. SPEC CPU测试工具的三个版本分别发布于2000年、2006年和2017年. 从2000年至2017年的17年间,商业通用CPU的性能至少增长了10倍. SPEC CPU2000/2006/2017三个版本演进过程中,工具开发人员希望通过加大数据集等手段,提高对通用CPU的测试压力,获取更准确的性能评测结

表1　性能基准测试对比分析

基准/工具	支持语言	编译程序	适配系统	支持的CPU架构	测试重点
SPEC-CPU	C, C++, Fortran	GCC, G++, Gfortran	Linux, Windows, MacOS X, AIX…	ARM, X86, Power ISA…	主要考察CPU的运算能力,以及内存性能和编译器能力
Dhrystone	C, Fortran, Pascal…	GCC, GFortran	Linux	ARM, X86	处理器的整数运算、逻辑运算和内存缓存等性能
NBench	C	GCC	Linux, Windows	ARM, X86, Alpha	处理器的整数运算、浮点运算和内存运算
stream	C, Fortran	GCC, GFortran	Linux	ARM, X86, Alpha…	处理器的内存带宽
coremark	C	GCC	Linux	ARM, X86, Alpha…	衡量嵌入式系统中使用的中央处理器的性能
UNIXBENCH	C	GCC	Linux	ARM, X86, Alpha…	系统整机性能集成CPU、调用、读写、进程、图形化以及编译器等
Stress	C, C++	GCC	Linux, Mac OS	ARM, X86	系统、CPU、内存以及磁盘IO压力测试
TPC-C	C	GCC	Linux	ARM, X86, Alpha…	主要通过模拟MIS和ERP系统,来测试计算机整机的联机事务处理数据库的查询、更新、mini-batch事务和吞吐量等性能
Netperf	C	GCC	Linux	ARM, X86, Alpha…	文件系统中数据传输的网络性能
IOZone	C	GCC	Linux, Windows, MAC OS X, AIX…	ARM, X86, Alpha…	操作系统中文件系统的读写性能
FIO	C	GCC	Linux	ARM, X86…	磁盘文件系统读写性能
DPDK	C	GCC	Linux	ARM, X86…	网络包的处理性能

果. 但从实际测试情况来看,对通用CPU访存、分支预测等关键操作的测试压力,并未因为数据集的增大而发生明显变化. 部分测试指标甚至在数据集增大后反而降低.

本文分别选取SPEC CPU 2000、SPEC CPU 2006和SPEC CPU 2017三款工具共同内嵌的MCF程序作为测试项,通过对某国产CPU的测试,直观反映了随着SPEC CPU的更新和版本迭代,相关指标的变化情况(图5).

SPEC CPU2000/2006/2017三款测试工具内嵌MCF

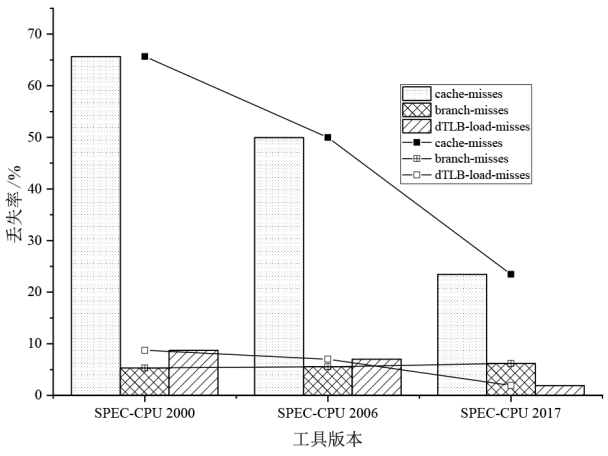

图5　不同版本SPEC-CPU指标对比

程序的数据集空间占用分别为200 MB、1.7 GB和600 MB. 尽管2000版工具的数据集占用最低,但是该版工具的Cache缺失率和数据TLB的缺失率显著高于后两代. 另外,各个版本工具的分支指令预测错误率基本持平. 可知,SPEC CPU版本迭代更新,虽然丰富了测试场景,完善了校验算法,满足越来越多的基准测试需求,但关键部件测试压力仍需进一步提高和优化,以便提升基准测试的准确率和整体性能.

关于UnixBench测试,随机选取X86、ARM等主流架构及14 nm和16 nm制程的4款CPU芯片进行测试. 4款芯片的基本参数如表2和表3所示.

表2　CPU基本参数说明

CPU分类	架构类型	主频/GHz	制程/nm	核心数
CPU A	X86	2.8	14	8
CPU B	ARM	2.3	16	8
CPU C	X86	2.7	16	16
CPU D	ARM	2.1	16	64

选择UnixBench的浮点数运算性能测试项Double-Precision Whestone,4款CPU耗时测试结果如图6所示. 随着运算量的增加,4款CPU的计算耗时整体呈上升趋势. 运算量增大,推升CPU中运算单元与缓存及内存之间数据和指令读写频率,增加数据冒险和控制冒险的发生概率,导致缓存缺失率、分支指令预测错误率升

表3 X86和ARM指令集简介

架构类型	架构特征	架构优势	主要应用领域
X86	指令系统庞大,功能复杂,寻址方式多,且长度可变,有多种格式; 各种指令均可访问内存数据; 部分指令需多个机器周期完成; 复杂指令采用微程序实现; 系统兼容能力较强	兼容性强; 配套软件及开发工具相对成熟; X86架构功能强大,高效使用主存储器; 处理复杂指令和商业计算的运用方面有较大优势	服务器、工作站和个人计算机等
ARM	指令长度固定,易于译码执行; 大部分指令可以条件式地执行,降低在分支时产生的开销,弥补分支预测器的不足; 算数指令只会在要求时更改条件编码	ARM结构功耗低; 体积小,聚焦移动端市场,适用于消费类电子产品	智能手机、平板电脑、工业控制、网络应用、消费电子产品等

高,处理器闲置概率增大,运算延时变大. 参考图6中的测试结果,CPU B,CPU C,CPU D这3款CPU耗时较为接近,由于CPU B与CPU D同属于ARM架构,CPU C为X86架构,表明架构的差异性对CPU耗时影响较小. CPU A的耗时明显小于CPU B,CPU C,CPU D,由于CPU A与CPU C主频接近,CPU C的核心数量是CPU A的2倍,但是CPU A与CPU C的耗时差距最大,可知X86架构CPU的频率成为CPU耗时性能的决定性因素,而非核心数量. 针对相同ARM架构、频率相近、核心数量差异悬殊的CPU B和CPU D,耗时性能在运算量逐步增加的压力测试下,表现出相似性能,表明增加核心数量不能显著改善ARM架构CPU的耗时性能,核心之间数据及指令同步、调度等操作反而增加时间开销. 总体而言,UnixBench测试工具针对X86架构、主频较高的CPU测试精度较高,而针对精简指令集ARM架构CPU测试结果并不理想.

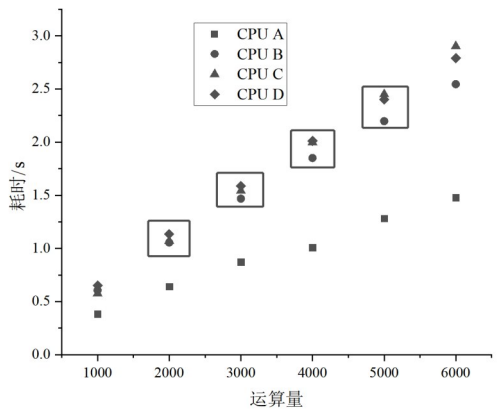

图6 不同CPU浮点数运算测试结果对比

进一步分析可知,Double-Precision Whestone算法的耗时呈现指数型增长,而非线性,如图7所示. 这导致CPU主频越高,双精度浮点性能的优势反而越不明显. 受测试基准算法设置等因素的影响,随着运算量的增加,不同芯片微架构的流水线级数、指令执行策略(结构冒险、数据冒险、控制冒险、相关处理等)、分支预测算法等的差异性,导致CPU性能急剧恶化. 因此,测试基准算法设置的合理性也是影响测试结果准确性的重要因素之一,在未来的性能测试基准优化和完善过程中,需要对算法设置的科学性和合理性进行重点考量.

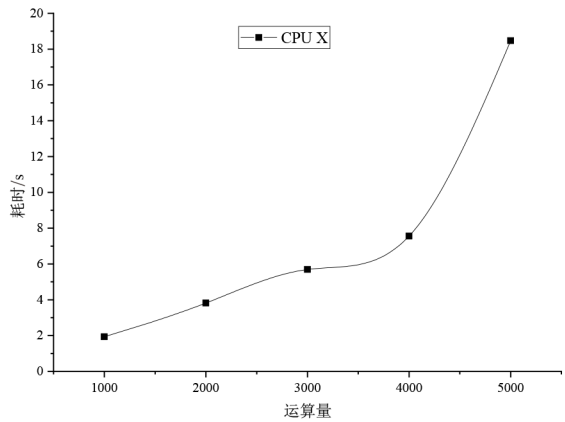

图7 双精度浮点数运算测试结果

系统总线直接关系到CPU与磁盘、网络等IO类应用的交互性能. 以最新发布的某国产CPU为例,如表4所示,当CPU芯片不变,系统总线的频率由1.6 GHz提升至3.2 GHz时,通过FIO软件测试磁盘文件读写性能,数据指标提升了14%~40%,表明总线频率加倍不能使CPU性能加倍,CPU主频相对总线频率成为CPU带宽提升的主要限制因素. 通过DPDK软件进行包转发测试,如表5所示,采用40 GHz网卡1 500字节包测试条件. 在1.6 GHz系统总线配置下,DPDK的单口最大发送和接受速度只能达到限速的61.6%和48.7%,而在3.2 GHz系统总线配置下,DPDK的单口最大发送和接受速度可以达到限速,此种场景下总线频率成为CPU带宽性能提升的主要限制因素. IO类测试结果共同表明,系统总线配置对发挥CPU运算处理性能有着重要影响. 而当前常用的CPU性能基准测试工具尚未集成相关的测试项并将其纳入CPU综合评价的结果中,对CPU架构的发展引导的全面性依然不足.

表4　FIO测试对比

测试项	1.6 GHz总线频率/(MB/S)	3.2 GHz总线频率/(MB/S)	性能提升/%
顺序读	1 251	1 581	26.4
顺序写	1 161	1 622	39.7
随机读	840	965	14.9
随机写	1 158	1 626	40.4

表5　DPDK测试对比

单口最大速率	1.6 GHz总线频率		3.2 GHz总线频率	
	包数/s	线速比/%	包数/s	线速比/%
发送速率	2 025 047	61.60	3 280 816	99.70
接收速率	1 602 238	48.70	3 258 598	99.10

4　面临的挑战

总体来看,通用CPU性能基准测试依然面临诸多挑战,具体包括:

(1)适用性问题. 随着云计算、人工智能技术的不断发展和工程应用的不断深化,跨硬件、跨平台、跨系统CPU应用场景不断丰富,操作系统的多样性和异构CPU共同发展成为行业发展的必然趋势. 但目前的主流CPU性能基准测试工具主要基于传统的X86架构体系发展而来,对应用越来越广泛的异构CPU的适用性受到一定的限制,特别是对于ARM等新兴架构的CPU,受发展周期的影响,其与传统X86架构CPU迭代周期具有一定的差异,各类基准测试程序与异构CPU之间适配环境的全面性仍显不足,缺少历史数据积累导致研发周期较长. 此外,数据密集型及计算密集型应用场景,对CPU的晶体管密度、集成度、电路规模、功耗等相关的制造工艺提出更高的要求,测试指标对部分真实场景下的业务的代表性不足,甚至存在AI类的负载、网络IO等版块缺失的情况.

(2)性能优化问题. 从测试准确性上看,当前针对核心频率、片内存储等基础指标的CPU性能基准测试能达到一定效果,但针对多节点调度、多核调度、处理器内/间通信能力等指标的测试程序构建和调优仍需加强;部分工具对CPU拓扑识别能力弱,同一种CPU在不同操作系统中的调度策略差异较大. 从测试效率上看,CPU的综合性能测试普遍超过48小时.

(3)综合性问题. CPU的发展需要的技术先进性高,但由于受知识产权保护等诸多因素的影响,部分基准测试工具仅对特定的主体开放,使得异构CPU的创新发展面临一定的困难. 受早期测试基准的限制,在静态测试、自动化测试、领域基准测试、仿真验证等方面,目前主要依赖现有的基准测试工具,而部分工具支持测试开发所使用的语言单一,拓展能力有限,不利于维护升级,甚至出现编译受阻等兼容性问题,给开发和应用造成了诸多不便,这也在一定程度上影响了基准测试的效果.

5　总结与展望

本文首先对当前国际主流的CPU基准测试现状进行了综述,并对各类测试基准和工具的功能、算法、应用场景等方面的迭代升级和最新成果进行了分析;然后对当前通用CPU性能指标和基准测试需求进行了梳理,并分析了通用CPU性能基准测试所面临的挑战. 目前,CPU性能基准测试工具和算法较早期已取得了较大的进步,满足的测试场景不断丰富,测试精度也有了明显提升,但在适用性、性能优化等方面依然有较大的提升空间.

未来CPU性能基准测试的新研究趋势包括:支持开放架构,提升对windows和Linux等多类型的操作系统及其衍生系统的适配能力,面向X86和ARM等不同架构的CPU研制具有针对性的测试基准,保障异构CPU的兼容性;提升面向新兴ARM架构CPU性能基准测试的体系化发展水平,建立基准簇,并培育可持续运营的基准组织机构,完善统一的综合性测试平台并打造自主"事实"标准,带动芯片制造、融合平台、行业应用的等全链条的一体化发展;丰富工程应用场景,在不断完善单项指标测试能力的同时,针对人工智能运算、大数据分析、大规模分布式等日益丰富的工程应用场景,研发有代表性的综合性测试基准、方法;提升算法效率和准确性,持续优化基准测试算法,构建丰富的测试模型,并综合考虑抗干扰、任务分解、耐久度、计算效能、实用性等较难测试的参数指标,实现CPU基准测试效率和准确性的全面提升.

参考文献

[1] TICHY W F, LUKOWICZ P, PRECHELT L, et al. Experimental evaluation in computer science: A quantitative study[J]. Journal of Systems and Software, 1995, 28(1): 9-18.

[2] TAO J H, DU Z D, GUO Q, et al. BenchIP: Benchmarking intelligence processors[J]. Journal of Computer Science and Technology, 2018, 33(1): 1-23.

[3] THAKKAR P, NATHAN S, VISWANATHAN B. Performance benchmarking and optimizing hyperledger fabric blockchain platform[C]//2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. Milwaukee: IEEE, 2018: 264-276.

[4] 王文凯. 基于DNN的智能芯片性能评估及优化[D]. 武汉: 华中科技大学, 2018.
WANG W K. Intelligent Chip Performance Evaluation and Optimization Based on DNN[D]. Wuhan: Huazhong Uni-

versity of Science and Technology, 2018. (in Chinese)
[5] HSIEH S Y, LIU C S, BUYYA R, et al. Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers[J]. Journal of Parallel and Distributed Computing, 2020, 139: 99-109.
[6] NIDER J, FEDOROVA A S. The last CPU[C]//Proceedings of the Workshop on Hot Topics in Operating Systems. New York: ACM, 2021: 1-8.
[7] PAWANEKAR S, UDGIRKAR G. Performance of Reinforcement Learning Simulation: X86 vs ARM[M]//Communications in Computer and Information Science. Cham: Springer International Publishing, 2021: 420-430.
[8] MATHÁ R, KIMOVSKI D, ZABROVSKIY A, et al. Where to encode: A performance analysis of x86 and arm-based Amazon EC2 instances[C]//2021 IEEE 17th International Conference on eScience. Innsbruck: IEEE, 2021: 118-127.
[9] LIMAYE A, ADEGBIJA T. A workload characterization of the SPEC CPU2017 benchmark suite[C]//2018 IEEE International Symposium on Performance Analysis of Systems and Software. Belfast: IEEE, 2018: 149-158.
[10] BACH M, KRETZ M, LINDENSTRUTH V, et al. Optimized HPL for AMD GPU and multi-core CPU usage[J]. Computer Science - Research and Development, 2011, 26 (3): 153-164.
[11] KALYANASUNDARAM K. SPEC HPG benchmarks [C]//Proceedings of the 2006 ACM/IEEE Conference on Supercomputing. Tampa: ACM, 2006: 17-es.
[12] WEISS A R. The standardization of embedded benchmarking: Pitfalls and opportunities[C]//Proceedings 1999 IEEE International Conference on Computer Design: VLSI in Computers and Processors(Cat. No.99CB37040). Austin: IEEE, 1999: 492-508.
[13] CAPRA M, BUSSOLINO B, MARCHISIO A, et al. Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead[J]. IEEE Access, 8: 225134-225180.
[14] SEN A, DENIZ E. Thread-level synthetic benchmarks for multicore systems[J]. Microprocessors and Microsystems, 2015, 39(7): 471-479.
[15] KARKI A, KESHAVA C P, SHIVAKUMAR S M, et al. Tango: A deep neural network benchmark suite for various accelerators[EB/OL]. (2019)[2022]. https://arxiv.org/abs/1901.04987.
[16] LEWIS B C, CREWS A E. The evolution of benchmarking as a computer performance evaluation technique[J]. MIS Quarterly, 1985, 9(1): 7-16.
[17] WEICKER R P. An overview of common benchmarks[J]. Computer, 1990, 23(12): 65-75.
[18] JOHN L K, EECKHOUT L. Performance Evaluation and Benchmarking[M]. Boca Raton: CRC Press, 2006.
[19] MATHÁ R, KIMOVSKI D, ZABROVSKIY A, et al. Where to encode: A performance analysis of x86 and arm-based Amazon EC2 instances[C]//2021 IEEE 17th International Conference on eScience. Innsbruck: IEEE, 2021: 118-127.
[20] CHONG N, SORENSEN T, WICKERSON J. The semantics of transactions and weak memory in x86, power, ARM, and C++[C]//PLDI 2018: Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation. Philadelphia: ACM Press, 2018: 211-225.
[21] Standard Performance Evaluation Corporation. SPEC CPU 2017[EB/OL]. [2022]. http://www.spec.org/cpu2017/.
[22] AMARAL J N, BORIN E, ASHLEY D R, et al. The Alberta workloads for the SPEC CPU 2017 benchmark suite [C]//2018 IEEE International Symposium on Performance Analysis of Systems and Software. Belfast: IEEE, 2017: 159-168.
[23] DIXIT K M. New CPU benchmark suites from SPEC[C]// Digest of Papers COMPCON Spring. San Francisco: IEEE, 1992: 305-310.
[24] SIMON J, VIETH M, WEICKER R. Workload Analysis of Computation Intensive Tasks: Case Study on SPEC CPU95 Benchmarks[M]//Euro-Par'97 Parallel Processing. Berlin, Heidelberg: Springer, 1997: 971-984.
[25] 廖秋林, 莫玮, 陈大为. SPEC CPU2000性能测试程序分析及其应用[J]. 国外电子测量技术, 2006, 25(6): 65-68.
LIAO Q L, MO W, CHEN D W. Analysis and application of SPEC CPU2000 performance test program[J]. Foreign Electronic Measurement Technology, 2006, 25(6): 65-68. (in Chinese)
[26] NAIR A A, JOHN L K. Simulation points for SPEC CPU 2006[C]//2008 IEEE International Conference on Computer Design. Lake Tahoe: IEEE, 2008: 397-403.
[27] PANDA R, SONG S, DEAN J, et al. Wait of a decade: Did SPEC CPU 2017 broaden the performance horizon? [C]//2018 IEEE International Symposium on High Performance Computer Architecture. Vienna: IEEE, 2018: 271-282.
[28] SONG S, WU Q Z, FLOLID S, et al. Experiments with SPEC CPU 2017: Similarity, balance, phase behavior and simPoints[EB/OL]. (2018)[2022]. http://lca.ece.utexas.edu/pubs/UT_LCA_TR-180515-01.pdf.
[29] SINGH S, AWASTHI M. Memory centric characteriza-

tion and analysis of SPEC CPU2017 suite[C]//Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering. New York: ACM, 2019: 285-292.

[30] BUCEK J, LANGE K D, KISTOWSKI J V. SPEC CPU2017: Next-generation compute benchmark[C]// ICPE'18: Companion of the 2018 ACM/SPEC International Conference on Performance Engineering. Berlin: ACM, 2018: 41-42.

[31] WALLACE R. Performance Benchmark and assessment of a mixed batch and real-time transaction processing system[C]//Proceedings of Computer Measurement Group Conference. Orlando: DBLP Computer Science Bibliography, 1997: 863-872.

[32] GRAY J. A view of database system performance measures[J]. ACM SIGMETRICS Performance Evaluation Review, 1987, 15(1): 3-4.

[33] LEVINE C. Why TPC-A and TPC-B are obsolete[C]//Digest of Papers. Compcon Spring. San Francisco: IEEE, 1993: 215-221.

[34] LEUTENEGGER S T, DIAS D. Modeling study of the TPC-C benchmark[J]. ACM SIGMOD Record, 1993, 22 (2): 22-31.

[35] CHEN S M, AILAMAKI A, ATHANASSOULIS M, et al. TPC-E vs TPC-C: Characterizing the new TPC-E benchmark via an I/O comparison study[J]. SIGMOD Record, 2010, 39(3): 5-10.

[36] KANDASWAMY M A, KNIGHTEN R L. I/O phase characterization of TPC-H query operations[C]//Proceedings IEEE International Computer Performance and Dependability Symposium. IPDS. Chicago: IEEE, 2000: 81-90.

[37] TRIVEDI M, CHEN Z Q. Lessons Learned from the Industry's First TPC Benchmark DS(TPC-DS)[M]//Performance Evaluation and Benchmarking for the Era of Artificial Intelligence. Cham: Springer International Publishing, 2019: 140-154.

[38] DEEHR E, FANG W Q, REZA TAHERI H, et al. Performance Analysis of Database Virtualization with the TPC-VMS Benchmark[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015: 156-172.

[39] 刘建鹏, 刘尧. 将TPC-DS工具合入HiBench测试框架的方法[J]. 数字技术与应用, 2019, 37(10): 64-65.
LIU J P, LIU Y. Method of incorporating TPC-DS tools into the HiBench test framework[J]. Digital Technology & Application, 2019, 37(10): 64-65. (in Chinese)

[40] ZHAI J D, ZHANG F, LI Q W, et al. Characterizing and optimizing TPC-C workloads on large-scale systems using SSD arrays[J]. Science China Information Sciences, 2016, 59(9): 92104.

[41] 冯志丹. 基于SCF中间件的TPC-C测试系统的设计和开发[D]. 北京: 北京邮电大学, 2016.
FENG Z D. The Design and Development of TPC-C Test System Based on SCF Middleware[D]. Beijing: Beijing University of Posts and Telecommunications, 2016. (in Chinese)

[42] Labs Primate. Geekbench 5.1.1[EB/OL]. [2022]. https://www.geekbench.com/blog/2020/04/geekbench-511.

[43] CORNERO M, ANYURU A. Multiprocessing in Mobile Platforms: The Marketing and the Reality[R]. Genève: ST-ERICSSON, 2013.

[44] KOZHIRBAYEV Z, SINNOTT R O. A performance comparison of container-based technologies for the Cloud [J]. Future Generation Computer Systems, 2017, 68: 175-182.

[45] POLVINEN T, YLIKÄNNÖ T, MÄKELÄINEN A, et al. Building a virtualized environment for programming courses[C]//WorldCIST 2020: Trends and Innovations in Information Systems and Technologies(AISC, volume 1160). Cham: Springer, 2020: 45-55.

[46] MORABITO R, KJÄLLMAN J, KOMU M. Hypervisors vs. lightweight virtualization: A performance comparison [C]//2015 IEEE International Conference on Cloud Engineering. Tempe: IEEE, 2015: 386-393.

[47] BARKER A, VARGHESE B, THAI L. Cloud services brokerage: A survey and research roadmap[C]//2015 IEEE 8th International Conference on Cloud Computing. New York: IEEE, 2015: 1029-1032.

[48] WANG Y, LEE V, WEI G Y, et al. Predicting new workload or CPU performance by analyzing public datasets[J]. ACM Transactions on Architecture and Code Optimization, 2019, 15(4): 1-21.

[49] SINGH J P, WEBER W D, GUPTA A. SPLASH: Stanford parallel applications for shared-memory[J]. SIGARCH Computer Architecture News, 1992, 20(1): 5-44.

[50] WOO S C, OHARA M, TORRIE E, et al.The SPLASH-2 programs: characterization and methodological considerations[C]//Proceedings of the 22nd Annual International Symposium on Computer Architecture. Santa Margherita Ligure: IEEE, 1995: 24-36.

[51] SAKALIS C, LEONARDSSON C, KAXIRAS S, et al. Splash-3: A properly synchronized benchmark suite for contemporary research[C]//2016 IEEE International Symposium on Performance Analysis of Systems and Software. Uppsala: IEEE, 2016: 101-111.

[52] BIENIA C, KUMAR S, SINGH J P, et al. The PARSEC

benchmark suite: Characterization and architectural implications[C]//PACT'08: Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques. Toronto: IEEE, 2008: 72-81.
[53] BIENIA C, KUMAR S, LI K. PARSEC vs. SPLASH-2: A quantitative comparison of two multithreaded benchmark suites on Chip-Multiprocessors[C]//2008 IEEE International Symposium on Workload Characterization. Seattle: IEEE, 2008: 47-56.
[54] BARROW-WILLIAMS N, FENSCH C, MOORE S. A communication characterisation of splash-2 and parsec [C]//2009 IEEE International Symposium on Workload Characterization. Austin: IEEE, 2009: 86-97.
[55] BIENIA C, LI K. Fidelity and scaling of the PARSEC benchmark inputs[C]//IEEE International Symposium on Workload Characterization. Atlanta: IEEE, 2010: 1-10.
[56] CHASAPIS D, CASAS M, MORETÓ M, et al. PARSEC-Ss: Evaluating the impact of task parallelism in the PARSEC benchmark suite[J]. ACM Transactions on Architecture and Code Optimization, 2016, 12(4): 41.
[57] CEBRIAN J M, JAHRE M, NATVIG L. ParVec: Vectorizing the PARSEC benchmark suite[J]. Computing, 2015, 97(11): 1077-1100.
[58] HUYNH A, HELM C, IWASAKI S, et al. TP-PARSEC: A task parallel PARSEC benchmark suite[J]. Journal of Information Processing, 2019, 27: 211-220.
[59] HEINECKE A, VAIDYANATHAN K, SMELYANSKIY M, et al. Design and implementation of the linpack benchmark for single and multi-node systems based on intel xeon phi coprocessor[C]//2013 IEEE 27th International Symposium on Parallel and Distributed Processing. Cambridge: IEEE, 2013: 126-137.
[60] BLIN A, COURTAUD C, SOPENA J, et al. Understanding the memory consumption of the MiBench embedded benchmark[C]//NETYS 2016: Networked Systems (LNCCN,volume 9944). Cham: Springer, 2016: 71-86.

作者简介

史惠康 男，1975 年 2 月出生，山西岢岚人. 1999 年在中国科学院计算技术研究所获工学硕士学位，现为清华大学计算机科学与技术系博士研究生. 主要从事电子与信息方面的研究工作.
E-mail: shk@pku.org.cn

王泽胜 男，1987 年 1 月出生，河北保定人. 2018 年在北京交通大学获工学博士学位，现为中国电子技术标准化研究院软件应用与服务研究中心工程师. 主要从事云计算、信息技术服务等方面的研究工作.
E-mail: 815345591@163.com

张士宗 男，1989 年 4 月出生，山东东营人. 2018 年在北京邮电大学获工学博士学位，现为中国电子技术标准化研究院信息技术研究中心工程师. 主要从事计算性能基准测试、计算机网络应用等方面的研究工作.
E-mail: zhangsz@cesi.cn

高 翔 男，1982 年出生，湖北荆州人. 教授级高级工程师. 2007 年在中国科学技术大学获工学博士学位. 现为龙芯中科技术股份有限公司副总经理. 主要从事高性能计算机体系结构、并行处理和操作系统等方面的研究工作.
E-mail: gaoxiang@loongson.cn

赵有健（通讯作者） 男，1969 年出生，甘肃会宁人. 1999 年在东北大学获工学博士学位，现为清华大学计算机科学与技术系教授、博士生导师. 主要从事高速互联网体系结构、交换与路由和高速网络设备等方面的研究工作.
E-mail: zhaoyoujian@tsinghua.edu.cn

勘 误

本刊 2021 年第 49 卷第 12 期《基于迹变换和旋转增量调制特征的模糊人脸识别》（作者：汪宇玲，陈立，黎明，钟国韵，何月顺，常玉祥，宋伟宁）一文中，基金项目应为“国家自然科学基金（No.62066003，No.61866025，No.41872243）；国家重点研发计划（No.2018YFB1702702）；江西省核地学数据科学与系统工程技术研究中心开放基金（No.JETRCNGDSS202006）”. 特此更正.

《电子学报》编辑部