文章编号:1671-4598(2024)09-0044-07 DOI:10.16526/j.cnki.11-4762/tp.2024.09.007 中图分类号:TP311 文献标识码:A

国产通用计算机性能测试系统的设计与验证

刘 建[1,2],何冬辉[1,2],刘 维[1,2],杨攀飞[1,2]
(1.工业和信息化部 电子第五研究所,广州 511300;
2.基础软硬件性能与可靠性测评工业和信息化部 重点实验室,广州 511300)

摘要:针对目前国产CPU架构众多、操作系统技术路线分散,面向异构平台的通用计算机综合性能测试工具较少的情况,设计并实现了一套跨国产CPU平台、兼容不同服务器和桌面操作系统的通用计算机综合性能基准测试系统;测试系统选取了CPU、GPU、内存、存储、网络、操作系统、运行时及典型业务模型等8组具有代表性的基准测试程序来模拟真实的工作组合,并以插件方式集成基准测试程序,测试系统采用统一运行框架、统一打分模型;将性能测试系统与GLmark2、SPEC CPU 2017等专项测试工具进行测试比较,通过归一化、t检验等方式验证其测试结果无显著差异性,国产通用计算机性能测试系统满足性能评测需求,为国产软硬件环境下相关测试工具的设计和改进提供了借鉴和参考。

关键词:通用计算机;异构环境;综合性能;基准测试套件;评测

Design and Verification on Performance Measurement System of Domestic General Purpose Computer

LIU Jian[1,2], HE Donghui[1,2], LIU Wei[1,2], YANG Panfei[1,2]
(1. The Fifth Electronics Research Institute of the Ministry of Industry and Information Technology of P. R. CHINA, Guangzhou 511300, China;
2. The Ministry of Industry and Information Technology Key Laboratory of Performance and Reliability Testing and Evaluation for Basic Software and Hardware, Guangzhou 511300, China)

Abstract: KeywordsFor the current architecture of numerous central processing units (CPU), scattered operating system technology, limited comprehensive performance testing tools for general-purpose computer for heterogeneous platforms, a general-purpose computer comprehensive performance benchmark test system is designed and implemented, which is adopted by domestic CPU and compatible with different types of and operating systems. The test system adopts 8 sets of representative benchmark test programs such as CPU, GPU, memory, storage, network, operating system, runtime and typical service, simulating the real work combinations, and the benchmark test programs are integrated in a plug-in manner. The unified operation framework and scoring model are used to compare the performance testing system with specialized testing tools such as GLmark2 and SPEC CPU 2017 et al., the normalization and t-test methods are used to verify that there is no significant difference between the test system and other test tools. A general-purpose computer meets the feasibility and effectiveness of the performance measurement system, and it provides a reference for the design and improvement of relevant test tools in domestic software and hardware environment.

Keywords: Keywordsgeneral-purpose computer; heterogeneous environment; comprehensive performance; benchmark suites; measurement

0 引言

计算机系统性能评估是厂商发现性能瓶颈,优化系统整体性能的必要手段,也是客户了解产品的重要参考依据。随着计算机技术的进步,特别是多核体系结构的出现,对系统性能评估提出了各种要求和限制。因此,设计科学公正的测评系统是一项具有挑战的任务,测试评估计算机性能的标准及技术,还在不断发展中。

计算机系统各方面的性能可以通过指标来反映,性能评估把计算机系统分解为器件、部件、模块、分系统等部分,将用户关心的性能特性形成指标和对应的测试方法。评估内容包括通用计算性能,用于评价CPU等器件计算能力,包括定点计算、浮点计算等特性。内存性能,用于评价各类内存模组等部件性能,包括访存带宽等特性。外部存储性能,用于评价存储控制芯片、存储颗粒、硬盘、固态盘、RAID卡、HBA卡等器件和设备的性能,包括数据传输率、IO吞吐能力等特性。网络性能,用于评价网卡、

收稿日期:2024-02-05; **修回日期**:2024-04-23。
基金项目:工业和信息化部电子第五研究所专项基金(22Z12)。
作者简介:刘 建(1979-),男,硕士,高级工程师。
通讯作者:刘 维(1986-),女,硕士,高级工程师。
引用格式:刘 建,何冬辉,刘 维,等.国产通用计算机性能测试系统的设计与验证[J].计算机测量与控制,2024,32(9):44-50.

网络交换芯片等器件和模块性能，包括吞吐率、时延、丢包率等特性。面向特定应用的专用计算性能，用于评价 FPGA、加速器、GPU、显卡等器件、部件、模块或分系统。

现存许多商用或开源的计算机性能测试工具，Whetstone[1]是为了比较不同的计算机的浮点性能而设计的综合型基准测试程序，基于 Fortran 程序进行大量浮点计算，但它对于具有高度内部并行性（管道，矢量计算等）的计算机不是很适用，特别是在优化和并行编译器结合使用时[2]。Dhrystone[3]主要目的是测试 CPU 的整数运算和逻辑运算的性能。Stream[4]是强调内存系统的单独基准测试，主要测试稳定的系统内存带宽和单矢量核相应的计算速率[5]。Linpack[6-7]是使用广泛的测试高性能计算机系统浮点性能的基准测试，其中的 HPL 测试是针对现代并行计算机提出的测试方式[8]，但其测试只反映计算机性能的一个方面，准确地评价机器和操作系统需要收集更可靠和更有代表性的数据[9]。NASA 设计的并行基准测试程序 NAS Parallel Benchmark（NPB）[10]主要用来评测大规模并行机和超级计算机的并行计算能力。

处理器性能评估一直是性能基准工具研究的热点，SPEC（Standardized Performance Evaluation Corporation）是性能测试套件研究最成功的代表之一，从原始版本 SPEC CPU 89 开始，随着计算机领域的发展，处理器架构的更新，制造工艺的改进以及内存容量的不断增加[8]，SPEC 推出的 SPEC CPU 也已发展至第六代的 SPEC CPU 2017，旨在准确地评估系统的处理器、内存子系统和编译器。SPEC CPU 作为业界最流行的 CPU 性能基准测试套件之一，国内外众多学者和工程师对工具负载的特性等内容和相关测试技术展开了研究[11-12]。尽管 SPEC 在 CPU 的性能评测领域具有很高的权威，SPEC CPU 也随着计算机技术的发展而演变和完善[13]，但依然存在版本继承时间跨度大，历史包袱较重、因版本更迭缓慢而导致测试套件缺乏新兴领域的应用代表、不兼容部分类型国产硬件架构或操作系统[14,15]等问题。为此，国内研究机构和企业开发了国产版本的 CPU 性能评测基准工具 CPUbench[14]。

此外，针对计算机系统性能评测的工具还有多媒体性能测试工具 qtperf、GLmark、glxgears，存储性能测试工具 Iozone、Iometer、SPC，网络性能测试工具 Netperf、iPert 等专注计算机某方面性能指标的测试工具。综合性能测试方面则有 SysBench[16]、UnixBench[17]等工具。SysBench 主要用于评估计算机系统在不同负载条件下的性能表现，主要实现对 CPU、内存、文件 IO、线程、数据库性能的测试，UnixBench 是一款经典的用于测试类 Unix（Unix-like）系统的综合性能测试工具，它执行 Dhrystone、Whetstone 和 Graphical tests 图形测试等 10 项单项测试，可以提供系统性能的基本指标。

综上，开源的计算机性能测试工具种类繁多、配套相对健全，但部分测试工具对被测对象有一定的限制条件，不适用于对通用计算机的测试，且大部分开源工具功能相对单一，而业界现有的综合性能测试工具也面临一些问题，如不能满足或充分反映通用计算机系统的各项性能指标测试需求，当需要完整地评测通用计算机各项性能指标时，通常需要使用多款工具，综合参考其测试结果，这无疑提高了测试成本。其次，由于现今计算机系统的复杂性，综合性能测试工具的测试套件落后于当前通用计算机的实际业务发展，无法体现新兴业务对计算机系统性能的测试需求。

另一方面，随着计算机领域异构多核架构的发展，计算机硬件系统异构化、软件系统混合多元趋势愈发明显，加上国产基础软硬件碎片化多技术路线并行发展，底层 CPU 有 MIPS/LoongArch、ARM、S_W64 等指令集架构，通用操作系统也基于各种不同的根社区开发，技术路线上存在一定差异，在国产异构环境中开展测试，业界现有的测试工具不兼容部分国产软硬件，需要考虑不同架构的国产平台的指令集差异，基础测试环境的管理和测试工具的封装，以及不同编译器、不同版本开发语言运行环境等问题。

本文针对上述问题设计了一种集聚开源工具优点、支持不同国产硬件平台及兼容多种国产操作系统的通用计算机综合性能测试系统，首先，从统一运行框架、开发语言选择、多技术路线覆盖、多软件栈版本兼容等角度，解决测试工具在不同平台上的兼容性问题。其次，测试系统根据现今通用计算机的物理、软件特征进行性能基准程序设计，选用的基准程序能够充分代表当前通用计算机的实际业务特征，并尽可能覆盖各个功能模块，测试内容包含 CPU、GPU、内存、存储、网络、操作系统、运行时，以及典型业务模型等。在异构硬件快速迭代，多技术栈软件并行发展背景下，为评测异构环境下计算机的不同性能指标和综合性能指标提供了依据。

1　系统设计

通用计算机性能测试系统支持多 CPU 系统的测试，实现对通用计算机单任务处理性能、多任务处理性能以及并行处理能力的测评。可以根据测试目标选择不同的基准用例，再根据各项基准用例分值得到相应测试套件分值乃至被测机综合性能分数，实现对不同硬件平台、不同技术路线操作系统的性能测试及性能问题分析。

1.1　通用计算机性能测试系统总体框架

测试系统由运行框架和 workload 基准测试程序组成，运行框架通过用户输入配置（通过配置文件或命令行参数），编译、构建、运行各基准测试程序，最后生成测试报告。支持 C、C++、Java、Python 语言构建运行基准测试程序，以组件形式管理，各组件相互独立，实现标准的组件接口。运行框架从层次结构上可分为使用接口层、框架运行层、测例接口层及主要由 CPU、操作系统组成的运行平台模块。

使用接口层：包括参数选择、全局配置、基线选择等内容。具备图形操作界面及命令行操作两种运行方式，图形操作界面提供基准测试操作模块、测试结果显示模块、

工具基本情况等模块。

框架运行层和测例接口层：实现对评测模型的功能运行、测试套件的运行调度，及一些通用功能，如日志记录、版本升级等。

测试程序：测试系统以组件形式管理基准测试程序，具备扩展性强、轻量化、灵活配置的特点。

操作系统和硬件平台：支撑测试系统的运行。

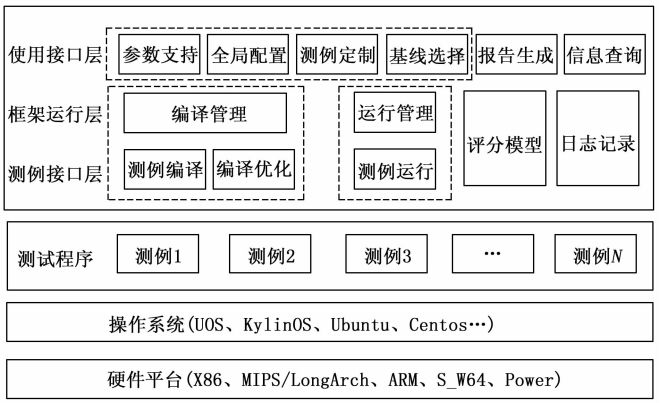

图 1 台式计算机性能基准测试套件框架示意图

为确保 workload 跨平台兼容，使用性能采集工具采集热点函数，分析函数调用栈，梳理核心业务运行代码路径，对 workload 模型未使用到的模块进行裁剪，削减线程、屏蔽相关接口等处理，确保 workload 聚焦相关能力评估，及重写相关代码等措施确保 workload 的跨平台兼容。

测试系统具有如下特征。

1）多种使用场景：该工具适用于对台式机、笔记本、瘦终端、工作站、服务器等通用计算机设备开展性能测试。

2）统一评价模型：针对带宽、延迟、时长等指标进行归一化处理，提供统一的分数评价模型。

3）配置灵活：可配测试用例权重、测试分类权重、运行参数、编译优化等。

4）测试类型各异：涵盖 CPU、网络、磁盘、图形、运行时等测试用例，为多种使用场景、不同测试目标提供支撑。

5）接口统一：提供标准的输入输出、运行命令、支持工具等，使得框架接口统一，仅需较小改动测试用例即可完成集成。

1.2 测试程序集设计

工作负载表征是性能评估中的核心问题之一，基准测试背后的原则是用一组较小的代表性项目来模拟真实的工作组合[18]。如果一组基准选择得好，那么实际作业组合中的每个程序都具有与一个或多个基准程序相同的性能特征。基准测试有 4 种不同类型[19]：实际应用程序、小型基准测试、基准测试套件和综合基准测试。每种类型都有其优缺点，基准套件是来自不同行业的不同基准的选集，它们共同代表了一台计算设备上的各种计算负载。套件在涵盖各种参数和特征方面非常有用，但需要定期更新，以便在典型工作负载发生变化时更改应用程序。

服务器性能测试系统集成了 RUNTIME、SYSTEM、NETIO、FILEIO、CPU、COMPLEX、MEMORY 等 7 个基准测试套件。桌面计算机对于服务器，硬件上的主要差异在于声卡和显卡，以进行音视频等多媒体处理，因此需要增加显卡性能测试，即桌面计算机性能测试系统增加了 GRAPHIO 测试套件。每个套件都被设计成执行特定的测试类，作为一个组运行，所有基准套件都遵循相同的方法，以产生一个总体指标。测试系统将每个单项测试的结果与基准系统（参考机器）上测得的基线进行比较，每个单项测试得到一个比值，比值越高，性能越好，这样的比值比原始值更具参考价值。测试集合里所有指标值组合起来，形成系统的总体指标。

以桌面计算机测试系统为例，当前版本集合了 152 个基准测试程序，不同套件运行不同类型的工作负载，例如处理器性能测试针对系统单核和多核，测试 CPU 的整数运算性能、浮点运算性能、计算圆周率、素数的加法运算。内存性能测试设计重点是对数组的复制（Copy）、数组的尺度变换（Scale）、数组的矢量求和（Add）、数组的复合矢量求和（Triad）4 个方面。操作系统性能测试主要关注对操作系统关键核心模块、库/包函数的验证，测试项包括函数调用、互斥体加锁/解锁耗时、线程同步和调度速率、管道传输自身耗时、高并发处理耗时等。8 组基准测试套件的测试内容概括如下，括号内为每个套件的测试用例数。

CPU 基准测试（28）：整数运算、浮点运算、缓存带宽等。

MEMORY 基准测试（8）：内存带宽、内存延迟等。

SYSTEM 基准测试（20）：系统调用、递归调用、系统接口调用等。

FILEIO 基准测试（27）：文件顺序读/写、文件随机读/写、文件基本操作、存储设备读/写等。

NETIIO 基准测试（5）：TCP 延迟、TCP 带宽、UDP 延迟、UDP 带宽等。

GRAPHIO 基准测试（24）：2D 图元及图形绘制、3D 场景绘制等。

RUNTIME 基准测试（33）：JAVA、C＋＋、LUA、PYTHON 等语言运行时。

COMPLEX 基准测试（7）：httpd 服务、压缩解压缩程序、数据库等。

1.3 参考机选择

选择参考样机的主要目的是解决原始测试结果无法横向对比的问题。运行测试负载获得的原始数值是运行时间，单个测试项可以直接对比，但多个测试项综合进行对比时，因测试项“大小”不等，测试项运行的时间就会有较大差异，无法直观地展示不同计算机的性能差异。因此，对于多维度的性能评测，需要借助数学算法将多个测试项的结果进行无量纲、加权处理。而上述过程用到的一组标定值即是来自于参考机运行各个测试项得到的运行时间。通过

这种方式即可把不能直接对比的时间值，转换成可对比的分值，并通过数学换算后对计算机的性能进行综合对比。具体而言，先采用参考机运行各个业务负载获得各个 workload 的参考性能基准，然后通过实际测试性能对比参考性能基准，标准化为统一的指标维度，最后对各个指标维度求几何平均得到总分值。

根据产品市场销量、产品成熟度、技术指标状态、软件栈（操作系统、三方库、基础服务组件）固化状态、性能值波动情况、产品可获得情况等原则和策略，参考机（基准机器）最终选择如下：

服务器：HPE DL380 Gen10 Intel（R）Xeon（R）Bronze 3106 @ 1.7 GHz（双路）；

台式机：ThinkStation 30A6 Intel（R）Xeon（R）E5-1603 v3 @2.80 GHz。

1.4　通用场景性能测试评价方法

基准测试中的一个常见场景是，为了总结结果，需要将多个不同的指标聚合为单个值。总分值的计算方式对结果有很大影响，采用不合适的方法可能导致无法反馈真实系统性能信息。总结测量样本的最常用方法是计算平均值，并将其用作表征测量属性或特征的度量，复合度量（composite metric）通常被定义为一组基本度量的平均值（算术、调和或几何）[20]。文献［18，21－22］描述了算数平均、调和平均、几何平均等计算平均值的算法的特点和在不同场景下的适用条件。几何平均值作为集中趋势的指标，它通常位于大多数测量值分布的区间中心。几何平均值具有这样的属性，即在对标准化数据进行平均时，它确保了一致的排名[20]。基准套件设计中采用几何平均方法来计算多个基准测试指标的一个典型例子是 SPEC CPU，自从它的第一个版本发布（SPEC CPU 89）以来，该基准测试套件就一直遵循相同的方法来得出综合指标，评测被测 CPU 的性能[20]。

本测试系统采用几何平均来计算每个分套件和系统综合性能的分值，每个套件的总分值用 SuiteScore 表示，EntireScore 表示被测系统整体分值，即综合性能指标。根据参数设置，每次测试的结果最多可以得到 8 个分套件的分值，测试报告显示各个套件每个单独测试用例的分值以及系统整体性能得分，这样测试人员既能看到被测计算产品系统的总体性能表现，也能获取所关注的每个测试套件以及组成每个套件的单个测试用例（测试负载）的性能表现，从而确定系统设计在哪些方面可能是性能瓶颈。

在通用场景性能测试中，各技术指标对应测试工具或测试包（基准程序），结果值之间权重均默认为 1（可根据实际需求进行设定），评价方法如图 2 所示。

1）参考基线结果值（BaseVal）：按照“参考机选择”中的方法，在选定的参考机器上获得的结果值。

2）基准程序结果分值（ResultScore）：被测机结果值（ResultVal）/参考基线结果值 * 系数。需注意一下问题：

（1）对同一部件或同一技术指标的测试，测试结果分值应做标准化处理；

（2）针对时间、网络延时指标，结果分值宜转换为频率、吞吐率或其它；

（3）基准程序多副本（多核）结果分值，应取多个副本分值之和。

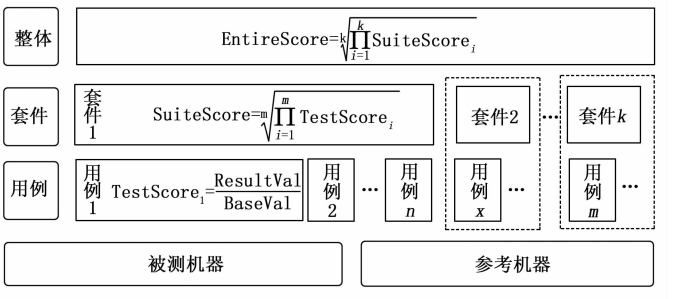

图 2　通用场景性能测试评价模型示意

3）套件得分（SuiteScore）：指选择多个技术指标进行测试，获得多个技术指标的综合测试结果值。计算方法见公式（1）：

$$\text{SuiteScore}=\sqrt[n]{\prod_{i=1}^{n}\text{TestScore}_i} \tag{1}$$

注：当式（1）中 TestScore＝0 时须确认测试程序或测试包及测试环境的正确性。

4）整体得分（EntireScore）：是指选择所有技术指标进行测试，所有基准程序的性能测试结果值。计算方法见公式（2）：

$$\text{ntireScore}=\sqrt[k]{\prod_{i=1}^{k}\text{SuiteScore}_i} \tag{2}$$

2　系统运行环境及运行说明

通用计算机性能测试系统支持对国产主流 CPU 和操作系统的测试，测试系统安装包提供针对不同指令集架构的内置依赖库，简化测试环境部署过程，提高了工具的易用性。系统运行环境如表 1 所示。

2.1　系统运行环境

表 1　测试系统运行环境

硬件环境	CPU 平台：兼容鲲鹏、飞腾、龙芯、兆芯、华为海思和申威 CPU，即兼容 ARM、X86、MIPS/LoongArch、Alpha、S_W、Power 架构。 内存：容量＞CPU 物理核数 * 2 GB 及以上； 硬盘：200 GB 及以上容量。
软件环境	操作系统和版本：兼容类 Linux/Unix 操作系统，包括中标麒麟操作系统、银河麒麟操作系统、统信 UOS 操作系统等。 依赖库或包：libreadline-dev、libxml2-dev、libpcre3-dev、libncurses5-dev、dc、rpcbind、x11perf、default-jdk、libgl1-mesa-dev、libpng-dev、python3-dev、libssl-dev。

2.2　运行说明

该测试套件基于一个开放平台框架，针对多种使用场景、不同测试目标，提供了统一运行框架和评价模型，可融合类型各异的测试用例。

添加新的测试程序，需设置如下参数配置方式：负载名、参数选项、迭代模式、日志输出、依赖等。

1）配置文件定义了环境信息、编译构建、运行、日志级别等一系列配置项，是测试系统与测试环境的重要交互方式。基于相同的软硬件环境和测试工具版本，用户使用相同的配置文件可稳定复现基本一致的测试结果。

配置文件是 ini 格式的文件，遵循业界通用的 ini 文件语法。用户可以定制配置文件，并通过以下两种方式来指定它：

通过＜code＞--config＜/code＞或＜code＞-f＜/code＞来指定配置文件的绝对路径；

将配置文件放在 config 目录中，并通过＜code＞--config＜/code＞或＜code＞-f＜/code＞来指定文件名，文件后缀可以省略。

2）测试和环境信息：

配置文件支持描述配置项，补充描述被测计算机的软硬件环境系统，方便用户理解被测计算机产品信息，比如网卡、显卡的型号，测试工具获取不到或采集的型号不准确，用户可通过配置文件纠正配置信息。

3）编译选项：

配置文件提供配置项控制 workload 的编译行为。

4）JAVA 运行参数：

配置文件提供配置项以方便用户进行 JVM 调优。

测试所有用例时直接执行 ./Run 即可，主要参数运行说明如表 2 所示。

表 2 运行参数说明

主要参数说明	命令运行示例	解释
－c：指定运行副本数，默认为 1，即单核运行；若设为 CPU 最大核数，即满核运行 －i：指定迭代运行次数，默认为 10 －t：指定运行并发线程数，默认为 1 －f：指定用例库类别，如 server. lst、desktop. lst －b：指定基线配置，该文件在 config 目录下 server：运行除了图形之外的所有基准用例	./Run	默认运行两轮用例全集，每个用例测试 3 次。第一轮是单进程测试，第二轮是多进程（CPU 核数）测试
	./Run -c 1 -i 1 server	运行单进程、单次，用例为除了图形之外的所有基准用例
	./Run -c 8 -i 3 RUNTIME	运行 8 进程、3 次，用例为 RUNTIME 用例组
	./Run -t 8 -i 5 hpcg_s	运行单任务 8 线程、5 次，用例为 hpcg_s
	./Run -c 1 -i 1 -f example_min	运行单进程、单次，用例所在文件 config/example_min

3 试验验证及结果分析

为了验证性能测试系统的可行性，分别对 CPU 测试程序、内存测试程序、典型应用程序测试程序等 8 个基准测试套件的测试程序进行了功能性验证；通过计算每个测试基准套件十次测试结果的标准差对测试稳定度进行测试；计算每个维度得出的算数平均值进行算法模型结果的正确性验证等可行性测试。此外，将性能测试系统的测试结果与业界较有影响力的测试工具的结果进行一致性对比验证。本章展现了部分实测结果。

3.1 单核/满核性能测试结果对比

测试五台分别搭载鲲鹏、飞腾、海光、龙芯、兆芯五款不同品牌国产 CPU 的台式机在 CPU 单核和满核情形下运行时的整体性能。硬件条件保持不变分别进行了两轮测试，每轮测试的被测机运行不同品牌的国产主流操作系统，分别用 OS1、OS2 表示，测试结果如图 3、图 4 所示，性能分值对应的台式机从左往右分别以 A、B、C、D、E 表示。

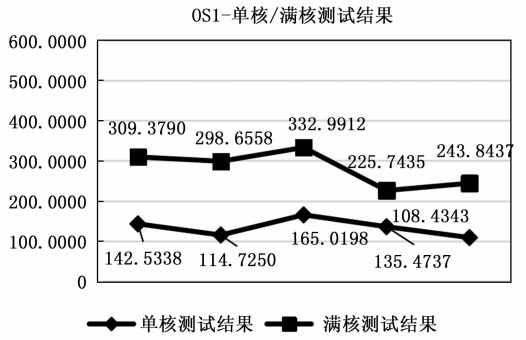

图 3 基于 OS1 的台式机单核/满核系统性能测试结果对比

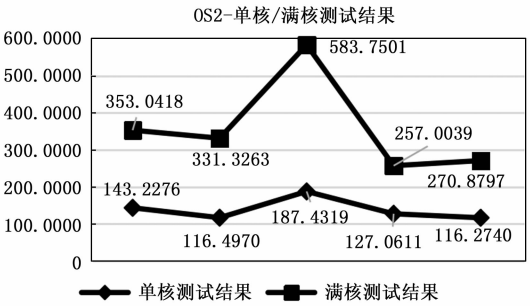

图 4 基于 OS2 的台式机单核/满核系统性能测试结果对比

图 5 对上述测试结果进行处理，黑色条柱/条纹条柱代表同一台台式机运行 OS2 时单核/满核性能分值高出运行 OS1 时的分值，结果可知，除台式机 D 外，台式机运行 OS2 时无论在单核还是满核情况下其综合性能分值均高于运行 OS1 时的分值。即在硬件环境保持不变的条件下，操作系统对被测机的总体性能得分具有一定影响，可通过测试报告的各个套件及测试用例的分值获得相关细节。可能的原因是不同品牌型号的操作系统与硬件的适配优化程度不同，编译器、操作系统内核版本等差异影响整机系统性能分值。

图 5 中 OS1-RATIO/OS2-RATIO 虚线上的点分别表示运行 OS1/OS2 时，满核相较单核提升的性能比值，即满核性能与单核性能分值之差和单核性能分值的比。运行 OS1 时，台式机 B 满核性能分值相较单核提升最大，该值为 160%；运行 OS2 时，台式机 C 性能分值提升最大，为

211%。该结果说明操作系统对被测台式机的满核性能相较单核性能提升的影响也不相同。

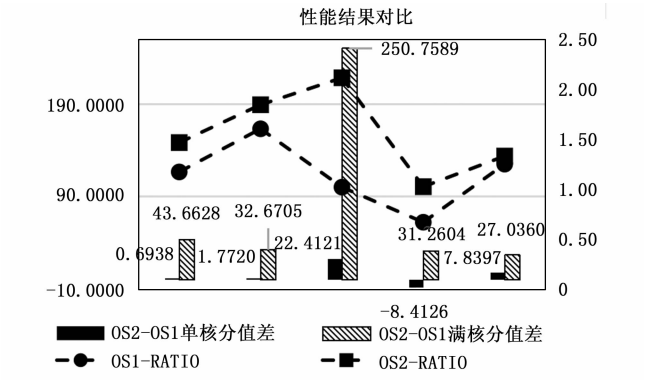

图 5　基于 OS1/OS2 的台式机单核/满核系统性能测试结果对比

3.2　图形性能测试结果对比

使用台式机综合性能测试系统（下称测试系统）与图形性能测试工具 GLmark2 分别对前述搭载不同品牌型号国产 CPU 的 5 台台式机的图形性能进行两轮测试，从对比测试结果图 6 和图 7 可知，在控制被测机的软硬件配置环境不变的情况下，两款工具因使用不同参考机型、选用的测试用例类型及数量存在差异等原因，在每轮测试中测得的分值不重合。注：该项目测试中为保持数量级一致，图 6 及图 7 中 GLmark2 测试结果数值皆除以 10。

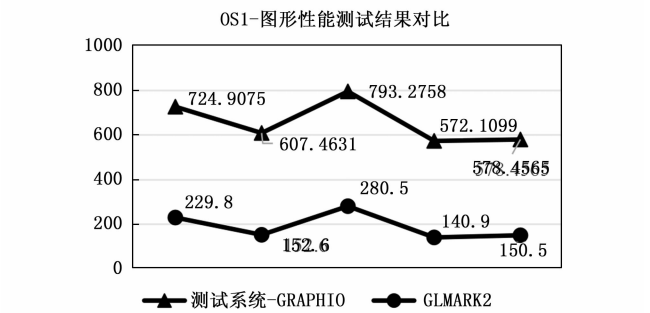

图 6　测试系统与 GLmark2 对基于 OS1 的台式机图形性能测试结果对比

图 8 中实线和虚线上的点分别表示运行 OS1 和 OS2 时测试系统和 GLmark2 测得的图形性能分值归一化后的结果。为分析运行两种操作系统时测试系统测得的分值与 GLmark2 分值的差异，采用 Shapiro-Wilk 正态性检验分别检验测试系统与 GLmark2 归一化后的分值是否符合正态分布，当满足正态分布时进行独立样本 t 检验以验证两组数据的相关性。经计算可知，运行 OS1 时两款工具归一化后的分值均符合正态分布，t 检验结果其 t 值为 0.064，P 值为 0.950，两组数据未呈现出显著差异性（P 值小于 0.05 或 0.01）[23]。以同样方式验证运行 OS2 时两款工具归一化后分值是否符合正态分布及其相关性，经计算可知 t 检验计算结果其 t 值为 0.542，P 值为 0.603，两组数据未呈现出显著差异性。

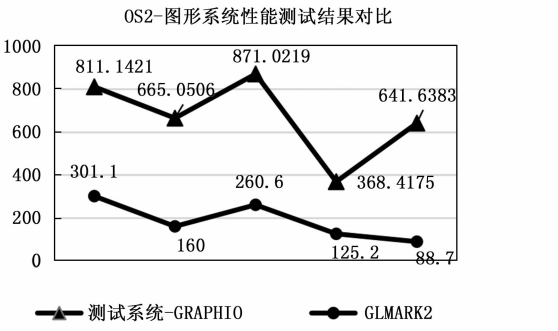

图 7　测试系统与 GLmark2 对基于 OS2 的台式机图形性能测试结果对比

上述 t 检验结果表明测试系统与 GLmark2 工具对 5 台式机图形性能的测试结果未呈现出显著差异，且台式机运行 OS1 时图形性能分值趋势更接近 GLmark2 工具测得的分值，测试系统与 Glmark2 分值的差异是两款工具对工具所取测试用例、计算方式、被测计算机软硬件适配优化程度等的体现。

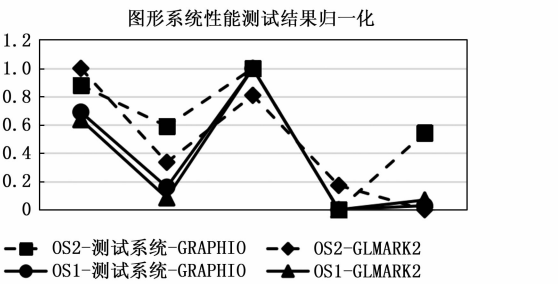

图 8　测试系统与 GLmark2 对运行 OS1/OS2 的台式机图形性能测试结果归一化图

3.3　测试系统与 SPEC CPU 2017 对 CPU 性能测试结果对比

使用台式机性能测试系统分别测试 3 台搭载飞腾、海光、龙芯 CPU 的台式机的单核及满核性能。测试系统对 CPU 性能的测试结果与 SPEC CPU 2017 的 CPU-int-base（整型测试模式）、CPU-fp-base（浮点测试模式）测得的 CPU 的性能得分进行对比，结果如图 9 和图 10 所示，次坐标代表测试系统的分值。采用前述的 t 检验方法分析测

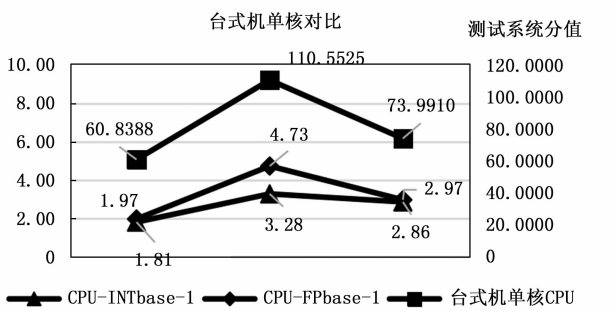

图 9　测试系统与 SPEC CPU 2017 对台式机 CPU 单核性能测试结果对比

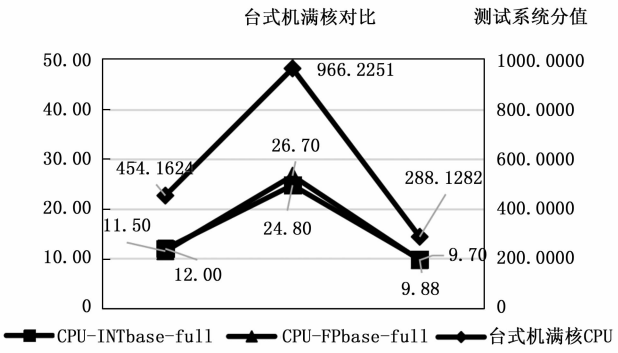

图 10 测试系统与 SPEC CPU 2017 对台式机 CPU 满核性能测试结果对比

试结果，单核和满核测试模式下可认为测试工具对台式机 CPU 性能测试结果与 SPEC CPU 2017 CPU 测的分值具有一致性。

4 结束语

通用计算机性能测试系统采用统一运行框架、统一打分模型、以插件方式集成性能测试项，具备扩展性强、轻量化、灵活配置的特点。由于本测试系统业界并无完全对标的产品，测试系统 8 组测试套件对被测机的测试结果分别与业界有较大影响力的测试工具测得的结果进行对比验证，测试分值具有一致性，考虑测试套件用例集的不同选取、测试程序本身的误差、分值计算方式等因素，其测试表现符合设计预期结果，综合考虑工具可靠性、标准符合性、可移植性、易用性等指标特征测试结果，该测试系统能作为一款通用计算机综合性性能测试工具，为计算产品综合性能评价提供评测手段，指导行业质量提升。对测试工具的优化升级、提高测试工具的稳定性和精确度，设计更科学、测试结果更符合客观事实的测试系统是我们下一步的工作重点。

参考文献：

[1] CURNOW H J, WICHMANN B A. A synthetic benchmark [J]. The Computer Journal, 1976, 19 (1): 43 - 49.

[2] KANT K. Introduction to computer system performance evaluation [M]. New York: McGraw-Hill, 1992: 14.

[3] Weicker R P. Dhrystone: A synthetic systems programming benchmark [J]. Communications of the ACM, 1984, 27 (10): 1013 - 1030.

[4] JOHN D. MCCALPIN. STREAM: Sustainable Memory Bandwidth in High Performance Computers [EB/OL]. [2024 - 01 - 10]. http://www.cs.virginia.edu/stream.

[5] 李春艳，张学杰. 基于高性能计算的开源云平台性能评估 [J]. 计算机应用，2013，33 (12)：3580 - 3585.

[6] PETITET A, WHALEY R C, DONGARRA J, et al. HPL-A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers [EB/OL]. [2018 - 12 - 02] https://netlib.org/benchmark/hpl/.

[7] DONGARRA J J. The linpack benchmark: An explanation [C] //International Conference on Supercomputing. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987: 456 - 474.

[8] 杜 琦，黄 卉，龚 盛，等. Intel Cascade Lake 架构 CPU SPEC CPU2017 评测 [J]. 计算机工程与科学，2021，43 (1)：49 - 57.

[9] 都志辉，吴 博，刘 鹏，等. LINPACK 与机群系统的 LINPACK 测试 [J]. 计算机科学，2002 (5)：8 - 10.

[10] NASA ADVANCED SUPERCOMPUTING (NAS) DIVISION. NAS Parallel Benchmarks [EB/OL]. [2023 - 07 - 26] https://www.nas.nasa.gov/software/npb.html.

[11] 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 (ISPASS). IEEE, 2018: 149 - 158.

[12] SINGH S, AWASTHI M. Memory centric characterization and analysis of spec cpu2017 suite [C] //Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering, 2019: 285 - 292.

[13] SPEC. SPEC CPU® 2017 Documentation Index [EB/OL]. [2022 - 11 - 07] https://www.spec.org/cpu2017/Docs/.

[14] 逯海涛，任 翔，钟伟军，等. CPUBench：一款开放的通用计算 CPU 性能基准工具 [J]. 微电子学与计算机，2023，40 (5)：75 - 83.

[15] SPEC. Building the SPEC CPU® 2017 Toolset [EB/OL]. [2021 - 01 - 22]. https://www.spec.org/cpu2017/Docs/tools-build.html.

[16] ALEXEY KOPYTOV. Scriptable database and system performance benchmark [EB/OL]. [2024 - 3 - 10]. https://github.com/ akopytov/sysbench.

[17] KDLUCAS KELLY LUCAS. Byte-unixbench [EB/OL]. [2024 - 1 - 10]. https://github.com/kdlucas/byte-unixbench.

[18] SMITH J E. Characterizing computer performance with a single number [J]. Communications of the ACM, 1988, 31 (10): 1202 - 1206.

[19] WYANT C M, CULLINAN C R, FRATTESI T R. Computing performance benchmarks among cpu, gpu, and fpga [J]. Computing, 2012.

[20] KOUNEV S, LANGE K D, JóAKIM VON KISTOWSKI Systems Benchmarking: For Scientists and Engineers [M]. Cham: Spring, 2021: 67.

[21] HOEFLER T, BELLI R. Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results [C] //Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2015: 1 - 12.

[22] FLEMING P J, WALLACE J J. How not to lie with statistics: the correct way to summarize benchmark results [J]. Communications of the ACM, 1986, 29 (3): 218 - 221.

[23] 孙振球. 医学统计学 [M]. 第 4 版. 北京：人民卫生出版社，2014：29 - 32.