

基于 Celery 的国产服务器测试平台设计与实现

陈峰, 斯佳佳

(中国电子科技集团公司 第五十二研究所, 杭州 311100)

摘要: 为了提高国产服务器性能测试和可靠性测试的测试效率, 设计并实现了一种国产服务器测试平台; 平台由测试管理层, API 服务层, 任务执行层, Shell 脚本层四层架构组成, 实现了测试任务异步的并行执行和指定顺序并行执行, 执行参数的个性化配置和校验, 测试任务的监控、终止和结果展示等功能; 平台构建了不同 CPU 架构的 Docker 镜像, 能够快速组建测试环境, 对批量国产服务器进行测试; 结合实际应用场景, 与现有的测试方法进行对比, 验证了平台的可行性和应用价值。

关键词: 国产服务器; 测试平台; 异步并行; Django; Celery

Design and Implementation of Domestic Server Test Platform Based on Celery

CHEN Feng, SI Jiajia

(The 52nd Research Institute of China Electronic Technology Group Corporation, Hangzhou 311100, China)

Abstract: In order to enhance the efficiency of performance and reliability testing for domestic servers, a domestic server testing platform is designed and implemented. The platform consists of four layers: the test management layer, API service layer, task execution layer, and shell script layer, it achieves test functions such as the asynchronous and parallel execution of test tasks, sequential parallel execution based on specified requirements, personalized configuration and validation of execution parameters, as well as monitoring, termination and result presentation for test tasks. The platform incorporates Docker images with different CPU architectures, rapidly building the testing environments and extensive examination of batch production of domestic servers. Compared to existing testing methods, its feasibility and practical worth are verified by integrating practical application scenarios.

Keywords: domestic server; testing platform; asynchronous parallel; Django; Celery

0 引言

国产服务器作为装备信息系统的核心和自控可控的重点, 其技术状态将直接影响装备信息系统效能。在不同架构不同形态的服务器设计研发中, 在响应招标比测或进行产品交付时, 需要对批量的国产服务器进行测试, 同时设备出货量也逐年提升, 对测试效率的要求也越来越高。

服务器以处理器和操作系统为核心, 国产处理器中龙腾、申威、鲲鹏等, 操作系统中麒麟、统信等占据较大的市场。为满足用户业务对计算、存储、网络和 IO 能力的要求, 国产服务器的测试主要在功能、性能、可靠性、接口、安全性和适配兼容性等方面展开^[1-2]。其中性能和可靠性的测试工作量大, 测试时间长, 详细测试项(使用的测试工具)主要有: 硬盘读写性能 (IOZONE, FIO, VD-BENCH), 系统处理性能 (LMBENCH), 网络性能 (NETPERF, IPERF, WEIGHTTP), CPU 计算性能 (SPECCPU), JAVA 性能 (SPECJBB), JAVA 虚拟机性能 (SPECJVM), 内存访问性能 (STREAM), 操作系统综合性能 (UNIXBENCH), 操作系统稳定性 (LTP), 内存稳定性 (MEMTESTER), CPU 稳定性 (STRESS), 整机

重启, 拷机测试。

传统的服务器测试技术, 都是基于单机的手工测试或者单机的自动化测试, 无法在短时间内完成批量设备的测试^[3-5]。李一风等^[6-9]基于 Celery 设计的自动化测试平台, 可以对软件业务系统的自动化测试用例进行异步并行的执行, 提高了测试用例的执行效率, 或将执行时间较长、消耗性能较多、逻辑较复杂的任务设置为 Celery 定时任务, 达到了减少工作量, 提高测试效率的目的。

为了提升批量国产服务器的测试效率, 本文提出一种国产服务器测试平台, 平台设计了四层架构以支持可扩展性、模块化, 易于后期维护, 并能适应未来可能的业务变化和技术升级。平台采用 Django+Bootstrap+Celery 架构, 实现了国产服务器性能测试和可靠性测试任务异步的并行执行或指定顺序并行执行, 测试任务执行参数的个性化配置和校验, 测试任务的监控、终止和结果展示。使用 Docker 容器进行镜像部署, 实现环境标准化、快速部署和资源隔离, 可以在龙腾、申威、X86 的 CPU 架构, 安装麒麟、统信操作系统的国产服务器上快速部署使用, 降低了测试工具的学习和使用成本, 提高了国产服务器的测试效率。

收稿日期: 2024-02-29; 修回日期: 2024-04-11。

作者简介: 陈峰(1989-), 男, 硕士, 高级工程师。

引用格式: 陈峰, 斯佳佳. 基于 Celery 的国产服务器测试平台设计与实现[J]. 计算机测量与控制, 2024, 32(9): 80-85.

1 平台设计

Django 是一款高效的开源 Python 网络开发框架, 提供丰富的内置功能、遵循 MTV 架构, 实现快速、安全、可扩展的 Web 应用开发。基于 Django 实现的 Web 应用程序^[11-15]质量高, 易维护。SQLite3 是一款轻量级、嵌入式、无服务器、跨平台的关系型数据库引擎, 以单个文件形式存储数据, 提供 SQL 接口且无需配置即可在应用程序中直接使用。考虑到本平台的实际使用场景需要快速部署在局域网中, 为了简化部署和配置, 减少对在线数据库的依赖, 本系统使用 Django 内置的 SQLite3 数据库。

Celery 是一个基于 Python 语言的开源分布式队列, 具有使用简单易于扩展的优点。gevent 是一款基于 Python 的协程网络库, 利用 libev (或 libuv) 实现高效的事件循环, 结合轻量级执行单元 Greenlet, 支持异步 I/O 和并发编程, 提供透明的非阻塞操作、任务调度和协作式多任务处理能力, 适用于提升网络应用的性能和可扩展性。Redis 是一款高性能、内存键值存储数据库, 支持多种数据结构, 具备数据持久化、主从复制、集群等特性, 常用于缓存、消息队列、排行榜等应用场景。在本平台中, 通过使用 Celery 的异步协程 gevent 执行池结合 Redis 消息队列, 对测试任务进行调度执行, 达到异步并行执行测试任务的效果。

Bootstrap 是功能强大的前端开发框架, 用于构建响应式、移动设备优先的网站。Bootstrap 框架提供了功能强大的组件库, 支持菜单导航、按钮等各种组件, 软件设计人员可以选取适合的组件直接使用, 还可以根据用户的实际需求进行定制。陶瑜等^[16-18]介绍了 Bootstrap 和 Web 应用结合使用的方法。在本平台中, 使用了 Bootstrap 提供的各种组件搭建前端页面, 还使用了 Bootstrap-table 库实现表格数据的获取、展示和编辑等功能, 使用了 BootstrapValidator 库, 通过自定义校验规则, 实现对不同表单的校验。

Docker 技术是一种基于 Linux 操作系统内核的虚拟化技术, 简化了环境配置, 让开发者将应用程序及其依赖的运行环境和配置文件一起打包, 对于用户使用而言十分方便, 达到建立镜像, 在任何系统上都可以运行的效果。吴逸文等^[19-21]将 Docker 技术应用在了软件开发和应用部署方面。本平台构建了不同架构的镜像, 使用容器化部署, 用于快速搭建测试环境。

Shell^[10]是操作系统中连接用户与内核的关键命令行接口与脚本编程工具。其核心功能包括命令解析执行、脚本编写以自动化任务。Shell 让用户能够通过输入命令操作系统资源, 编写脚本高效处理复杂任务, 并受益于跨平台兼容性和各实现版本的增强功能。

1.1 分层设计

平台框架分为四层, 分别为测试管理层, API 服务层, 任务执行层, Shell 脚本层。如图 1 所示。

每层的具体功能如下。

1) 测试管理层:

测试管理层使用 Django 建立了数据模型, 数据保存在

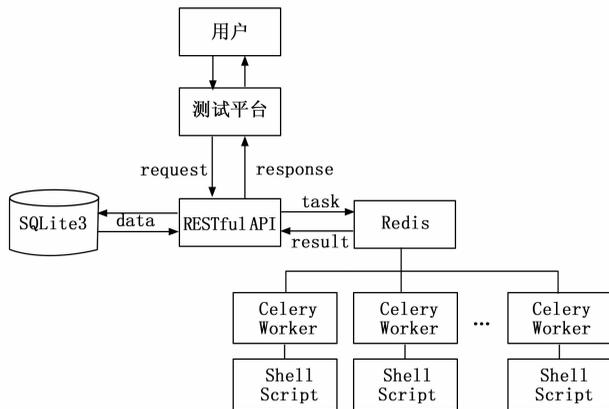


图 1 平台架构

本地的 SQLite3 数据文件中, 结合 Bootstrap 前端框架, 实现工具、设备、设备组、配置、测试集、测试链的可视化管理, 测试结果的记录和查询。

2) API 服务层:

API 服务层实现了 RESTful 接口, 对测试管理层封装了工具、设备、设备组、配置、测试集、测试链的管理接口, 向测试用户提供了增删改查等操作。

3) 任务执行层:

任务执行层是平台的核心功能, 使用基于协程的 gevent 执行池和 Redis 消息队列, 实现测试任务的异步并行执行。Celery 中的任务调度模块发起测试任务, 将任务消息发送到 Redis 消息队列。在平台部署的服务器上启动 Celery worker 执行任务, 当所有测试任务执行完成后, 记录测试任务执行结果。

4) Shell 脚本层:

Shell 脚本是用户和内核间的接口程序, 语法简单易学, 适用于麒麟、统信国产操作系统。Shell 脚本层可以完成所有性能和可靠性测试项的测试, 主要操作包括接收配置参数, 安装对应工具, 执行测试命令。

1.2 流程设计

平台的工作流程如图 2 所示。

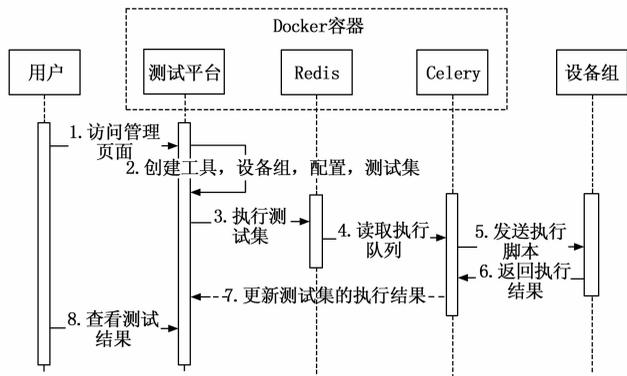


图 2 工作流程

- 1) 用户通过浏览器访问测试平台的管理页面。
- 2) 在平台管理上传测试工具或选择已有的测试工具。管理接入局域网的国产服务器，或按照模板批量导入国产服务器，创建待测设备组。创建配置信息，选择对应的测试项目，配置执行参数。创建测试集，选择对应的工具、设备组和配置。
- 3) 执行测试集，平台将待执行的测试集信息推送到 Redis。
- 4) Celery 读取 Redis 消息队列中待执行的任务队列，解析测试集中对应的参数配置。
- 5) Celery 启动 worker 进程，与每台国产服务器建立 ssh 连接，将对应的工具和 Shell 脚本上传的国产服务器，按照参数配置执行 Shell 脚本。Shell 脚本检测命令有效性，安装测试工具，执行测试命令，监测脚本执行情况。
- 6) Celery 收集国产服务器的测试任务执行结果。执行过程中，用户可以查看测试任务的执行进度。
- 7) Celery 收集完成所有设备的执行成功或失败的结果，将数据写入平台的 SQLite3 数据文件中，更新测试集对应的状态信息。
- 8) 用户查看测试结果。测试平台对执行结果进行关键字段的提取展示，用户可以筛选指定设备或指定测试项目的测试结果，记录关键字段取值，或查看完整的测试日志。

1.3 平台功能设计

图 3 为平台功能的具体功能用例图。每个功能在 Django 项目中为独立的应用，具体包括设备管理、设备组管理、工具管理、配置管理、测试集管理、测试链管理，他们的作用分别是：

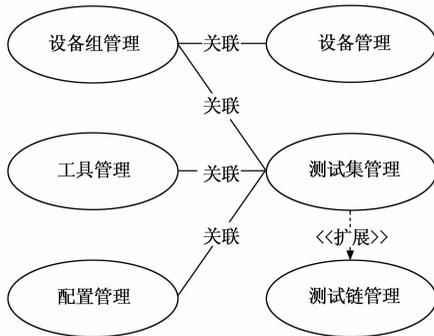


图 3 平台功能

- 1) 设备管理：测试人员可以在设备管理模块中，手工新增国产服务器，也可以管理使用 DHCP 网络分配 IP 自动接入的国产服务器。实时监控设备的测试状态以及编辑删除设备信息。
- 2) 设备组管理：多台国产服务器构成设备组，使用设备的 IP 进行区分。测试任务开始前可以批量检查设备的开机状态信息；测试任务执行时根据不同的测试任务编排不同的设备组；测试任务完成后，清理设备中的工具，脚本，日志，临时文件等。
- 3) 工具管理：管理性能测试和可靠性测试使用的各类

测试工具。通过工具支持的平台（如飞腾，申威等），工具的不同版本，实现各类测试工具的管理。

- 4) 配置管理：管理性能测试和可靠性测试不同测试项目的执行参数配置。
- 5) 测试集管理：针对某个测试任务对批量设备进行异步并行执行。使用 AJAX 方式加载已创建的设备组，通过测试项目筛选对应的测试工具和参数配置。测试人员可以对测试集进行执行、终止、结果查看、进度查看等操作。
- 6) 测试链管理：可以理解为按照指定顺序并行执行的多个测试集。测试人员可以指定不同顺序的测试任务进行测试。

2 平台实现

2.1 设备管理

设备信息包括设备的 IP，root 密码，整机 SN (Serial Number) 等信息。创建设备时校验 IP 合法性，MAC 地址合法性，配置了 root 密码，整机 SN 校验规则。设备管理对设备进行模糊查询，新增，编辑，删除，批量删除，批量导入功能。接入 DHCP 服务器的设备会自动识别设备信息并新增设备记录，也可以手工新增设备信息，新增时检查设备是否可以连接并更新连接状态信息。设备删除时判断是否使用，并给出哪些设备组正在使用。针对需要两台设备直连测试的场景，接通网线后，平台自动设置本端和远端的静态 IP，也可手工配置。

2.2 设备组管理

多台待测设备构成设备组，使用设备的 IP 进行区分，通过全选或勾选对应 IP 创建设备组。提供模糊查询，新增，编辑，删除，在线检测，环境清理等功能。

在线检测：在测试任务开始执行前，由于之前测试任务的执行或人为原因可能导致设备死机等情况的出现，可以使用在线检测功能检查待测设备是否开机正常工作，等待 15 秒左右页面给出设备在线、离线的列表。

环境清理：测试任务执行完成后，使用环境清理功能对测试中上传的工具，脚本，日志，临时文件等进行清理，将设备恢复到装机环境。

2.3 工具管理

工具信息包括类型，支持平台，工具版本，工具文件等信息。提供模糊查询，新增，编辑，删除功能。选择导航栏中的工具管理页面下的“新增”按钮，弹出工具新增页面，以 IOZONE 为例。选择工具类型，勾选支持平台，输入工具描述、工具版本，上传工具的附件文件并保存，保存时校验上传的附件文件名称是否包含工具类型和版本。工具删除时判断是否在使用，并给出提示哪些测试集正在使用。

2.4 配置管理

配置信息包括配置名称，描述，类型，参数等。提供配置模糊查询，新增，编辑，删除等功能。实现了对硬盘读写性能，系统处理性能，网络性能，CPU 计算性能，JAVA 性能，JAVA 虚拟机性能，内存访问性能，操作系统综合性能，操作系统稳定性，内存稳定性，CPU 稳定性，

整机重启, 拷机测试测试项的参数配置。

2.5 测试集管理

测试集信息包括测试集名称, 描述, 设备组, 测试类型, 工具, 配置, 测试次数。新增测试集时, 页面使用 AJAX 方式加载已创建的设备组, 根据测试类型的不同, 加载对应类型已存在的测试工具和配置。在增删改查功能之外, 提供执行, 结果查看, 任务终止, 进度查看功能。并提供所有历史执行记录的配置参数和结果查看。同时, 根据设备的整机 SN 对测试结果进行整理归档, 提供运维页面对该台设备的所有历史执行信息和日志进行管理。

2.6 测试链管理

测试链由按照指定顺序并行执行的多个测试集组成。页面使用 select2 前端框架实现下拉指定顺序多选功能。创建多个测试集, 组建为一个测试链。执行时, 按照配置的测试集顺序, 顺序解析对应的执行参数, 调用测试集的 celery task 发起异步并行执行。测试结果展示已经执行完成的测试集, 查看时可筛选设备和测试类型, 其他实现功能与测试集基本一致。

2.7 镜像构建

Dockerfile 是用于定义 Docker 镜像构建过程、包含所需软件栈与配置的文本文件, 旨在实现镜像构建的标准化、可重复与自动化。在 Docker 官网根据 python 版本号下载对应架构的 python 镜像, 编写项目工程包对应的 Dockerfile, 构建平台的 Dockerfile, 主要内容包括: 初始化对应架构的 python 镜像, 解压并安装 Redis 工具, 使用内部源安装 python 依赖包, 配置 Redis, 最后启动 Celery 和 Django 工程。

部署时可使用编译好的项目 tar 包, 使用 docker load <XX.tar 将项目导入为 docker 镜像, 使用 docker run -p port: 8000 image_id 启动项目 (8000 为 Django 项目默认启动端口, port 为服务端配置的空闲映射端口), 浏览器打开 IP: port 即可访问平台。平台构建了 X86, ARM, SW-64 架构的 Docker 镜像, 可以快递部署在不同架构的国产服务器平台上。

3 平台优势

3.1 测试任务异步的并行执行和指定顺序并行执行

测试人员完成一个测试任务的步骤包括: 确认测试工具版本, 通过存储介质或网络将测试工具上传至待测国产服务器, 解压缩测试工具并安装, 安装完成后确认工具执行参数, 执行测试任务, 执行过程中检查执行情况, 测试任务执行完成后收集记录测试结果和执行日志, 清理执行过程产生的临时文件并卸载工具。测试人员需要经过长时间的学习和操作才能熟练完成测试任务, 针对批量国产服务器的测试, 存在大量的重复性工作, 测试效率有很大的提升空间。

测试任务异步的并行执行是本平台的核心功能。平台实现了一个测试任务在批量国产服务器上的并行执行, 也可以对多个测试任务, 指定执行顺序, 在批量国产服务器上并行执行, 平台支持的测试任务类型覆盖了常用的性能

测试和可靠性测试项目, 降低了测试人员对工具的学习成本, 提高了批量设备的测试效率。

在平台的部署服务器上执行 Celery worker 命令: `celery worker -A project -l info -P gevent`, 启动一个 gevent 类型异步协程执行池的父进程, 父进程启动 N 个子进程执行测试任务, N 默认为 CPU 的核数, 理论上不受 CPU 核数限制, 可以通过参数 `-c` 设置 worker 子进程数量。

IOZONE 是一款文件系统性能测试工具, 用于评估读写速度、响应时间等关键指标, 支持多种工作负载模式 (如顺序、随机、混合读写等), 适用于多种操作系统环境。平台将每个测试项目封装为单独的测试任务, 以硬盘读写性能测试 IOZONE 工具为例, 使用 Celery 装饰器 `@shared_task` 注册测试任务 `iozone_group`, 设置任务失败的超时参数 `soft_time_limit`。

当用户在平台上点击按钮执行测试集时, 平台接收到请求后, 根据测试集 id 解析对应测试集中配置的设备组、工具和执行参数配置, 使用 `celery.group` 方法并行调度任务 `iozone_task`, 将任务消息发送到 Redis, Celery worker 读取消息队列, 异步并行的执行测试任务。 `celery.result.allow_join_result` 方法等待每个异步任务执行完成或任务超时后, 返回所有任务的执行结果。

paramiko 库是由 Python 库实现的 SSHv2 协议, 用于安全远程命令执行、文件传输和 SSH 代理, 封装 SSH/SFTP 客户端, 支持用户密钥认证, 适用于自动化运维。 `iozone_task` 任务执行时, 通过 paramiko 库与待测设备建立 ssh 连接, 上传测试集配置的测试工具, 按照配置的参数执行测试脚本 `iozone.sh`, 脚本兼容麒麟、统信操作系统, 脚本执行完成后返回测试结果。

测试任务脚本 `iozone.sh` 中, 读取 IOZONE 参数配置, 判断参数的合法性, 检查目标测试硬盘是否存在, 检查 gcc 版本是否满足环境要求, 解压缩 IOZONE 工具压缩包并安装 IOZONE 工具, 根据参数配置执行命令。

测试任务异步的顺序执行以测试集执行为基础。页面使用 Select2 前端框架实现下拉指定顺序多选功能。Select2 是一款适用于 Bootstrap 框架的高级选择器插件, 用于增强 `<select>` 元素的功能和外观。在测试中经常遇到晚上或假期等长时间无法继续测试任务的情况, 针对此种场景, 测试人员可以在测试平台创建多个测试集, 指定执行顺序组建为一个测试链, 实现测试任务异步的顺序执行。

3.2 测试任务执行参数的个性化配置和校验

性能测试和可靠性测试工具众多, 执行参数通常涵盖了多个方面, 例如包括脚本路径、并发数、超时时间、异常处理等, 实现对测试工具的全方位控制。通过调整参数设置, 测试人员可以对不同场景指标进行测试。同时也导致测试人员需要花费大量的时间熟悉每个参数的意义、使用场景和相互关系, 还要根据项目特点合理的选择和配置参数, 避免参数设置不当导致结果失准、效率低下等问题。

平台实现了性能测试和可靠性测试项执行参数的个性化

配置，预置了不同测试工具不同测试场景使用的执行参数模板，可以直接调用使用，也支持对每个测试项需要的参数进行配置，同时提供了默认参数和配置提示，并对每个参数的合法性进行检查，降低了测试人员的学习和使用成本。

参数的个性化配置采用 Bootstrap 模态框 modal，Bootstrap Modal 是 Bootstrap 框架提供的一个 JavaScript 插件，用于创建可覆盖在网页主要内容之上、独立展示特定内容的模态对话框或弹出窗口。平台为每个测试任务的配置类型的 modal 设置 id 参数，结合 html 的 onchange 属性监控切换动作，使用 JavaScript 内置的 html 方法加载不同 id 对应的 modal，实现选择不同的配置类型时，动态切换展示该类型对应的执行参数。

参数的校验使用 Bootstrap Validator 库对执行参数配置表单进行校验，Bootstrap Validator 是一个为 Bootstrap 表单添加客户端验证功能的插件，提供实时、多样化的验证规则以确保用户输入数据的准确性和合规性。平台实现了表单参数的自定义检查，如 IOZONE 执行参数块大小 block_size，检查参数不能为空，允许输入数字加字母的容量格式，当用户提交表单时，Bootstrap Validator 会自动触发验证，根据设置的规则检查输入数据的有效性。若验证通过，则表单提交；若验证失败，将阻止提交并显示对应的错误提示。

3.3 测试任务的监控、终止和结果展示

国产服务器上的性能和可靠性测试通常涉及大量数据处理、高并发访问、长时间运行等场景，对服务器的资源要求高，极易出现设备异常导致测试任务失败，因此需要及时发现并预警，及时终止测试任务的执行，排除异常情况或恢复测试环境后，重新进行测试。测试任务执行完成后，收集测试结果。每个测试工具输出的测试结果存在较大差异，需要手工筛选出关键指标，合并整理测试数据，增加了工作负担，降低了工作效率。

平台实现了性能测试和可靠性测试项运行情况的指定时间的定时监控，也可以终止执行中的测试项，方便测试人员实时查看测试任务的执行情况，出现异常或不符合预期时及时退出执行，排查原因或重新配置。梳理每个测试项需要记录的关键数据，根据需要进行加工后在结果查看页面进行展示，也可以查看完整的执行日志。

在 IOZONE 测试任务执行时，会同时调起状态监测脚本，根据配置的检测检查周期，定时监测 IOZONE 执行状态，收集检查数据，包括执行状态、测试项目、对应的测试集 id 等信息，发送到测试平台。测试平台实现了一个 csrf_exempt 的监测数据接收接口，收到数据后，更新测试集的状态并记录相关数据。如整机重启则计算每次重启的间隔时间，拷机测试计算每次拷机的完成间隔时间。

在测试任务的结果查看页面，执行中的测试任务，显示每台设备的任务状态，监测的详细日志；执行完成的测试任务，展示截取后需要记录的关键测试数据，可以直接导出所有设备的执行结果。对性能测试：IOZONE、FIO、LM-

BENCH、NETPERF、SPECPCPU、SPECJBB、SPECJVM、STREAM、UNIXBENCH、WEIGHTTP 构建了结果提取模型，并对测试设备所有的执行日志进行归档管理。对稳定性测试：整机重启、拷机测试、LTP、MEMTESTER、STRESS 实时监测测试设备运行情况。以 IOZONE 为例，测试结果可直接在平台页面查看，可以筛选指定设备，并对需要记录的关键测试数据进行了截取展示，也可以查看全部执行日志。如配置参数对应的执行命令为：iozone -a -s 10g -r 16m -i 0 -i 1 -i 2，即设置文件大小 16g，每次读取或写入块大小 16m，进行磁盘读写性能的顺序读写、重复读写、随机读写的测试。测试时需要测试三次，取平均值。平台对 IOZONE 的三次测试结果进行了整理，读取原始测试结果数据，匹配到对应的数据所在的行，正则截取需要记录的顺序读写、重复读写、随机读写的数据，根据测试结果，计算平均值，如图 4 所示。页面展示时，将设备的 IP 写入 html 页面的 class 属性中，当在结果展示页面选择对应的设备时，将对应 IP 的 display 属性设置为 true，其他设备设置为 false，达到筛选不同设备的效果。执行失败显示原始日志。

结果详情

设备:	全部				
设备: 5.5.5.11- 0224829- iozone	Writer	552665	542043	549997	平均值: 548235.0
	Re-writer	569087	595972	570778	平均值: 566479.0
	Reader	535354	533879	538164	平均值: 535799.0
	Re-Reader	530070	534377	535720	平均值: 533389.0
	Random write	495567	497154	492732	平均值: 495484.33
	Random read	381062	381910	383673	平均值: 382215.0

返回详细

图 4 测试结果

任务终止包括 Celery 任务的终止和批量国产服务器上测试进程的终止。Celery 任务的终止，结合 Celery 任务的开源管理工具 Flower，Celery Flower 是一个 Web 应用，用于监控 Celery 任务队列及工作节点状态，查看任务详情，图形化展示任务统计信息，并支持远程控制任务执行，应用运行后，会在 5 555 端口提供 web 服务。使用 Flower 提供的任务查询接口和任务终止接口实现 Celery 任务的终止。调用任务查询接口：http://IP: 5555/api/tasks，筛选出执行中的 Celery 任务，解析 Celery 任务执行参数，得到与测试集 suite_id 和任务执行时间戳匹配的 Celery 任务的 task_id，调用任务终止接口：http://IP: 5 555/api/task/revoked/task_id 终止对应的 Celery 任务。批量国产服务器上测试进程的终止，单独实现了一个进程终止的 Celery 异步并发任务，该任务与国产服务器建立 ssh 连接，使用 ps 命令查询进程 ID，结合 killall 命令杀死进程，返回进程终止的结果。

4 实验结果与分析

在测试项覆盖方面，覆盖了常用的 17 个国产服务器性能和可靠性测试项，对每个测试项预置了不同版本的测试

工具及工具常用的执行参数模板, 降低了工具管理和使用成本。

在测试稳定性方面, 平台构建了不同架构的 Docker 镜像包, 可以快速部署在不同架构的国产服务器上。测试人员对每个测试项重复执行了 10 次以上, 对不符合预期的测试项记录了问题, 在修改解决, 交叉验证后所有测试项的执行结果均可正确执行。

在易用性测试方面, 提供了所有功能的用户操作手册, 测试人员经过简单学习和培训后, 能够快速地进行测试任务的执行。对在使用过程中发现的操作便捷、界面友好、故障提示等问题进行了同步优化。

在测试执行效率方面, 准备了 90 台双路国产服务器, 操作系统: 麒麟操作系统 V10, CPU: 飞腾 S2500 * 2, 64 核, 2.0 GHz。使用 `docker load < XX_ARM.tar` 将项目的 ARM 镜像部署在其中 1 台飞腾双路国产服务器上, 使用 `docker run` 启动项目。

在 90 台国产服务器进行了性能测试项中相对用时较短的 IOZONE, STREAM, NETPERF 测试, 对应测试工具包已提前上传到服务器上, 手工测试的主要步骤包括工具安装、测试执行、结果收集, 测试项的执行参数如下:

```
iozone -a -s 10G -r 4k -i 0 -i 1 -i 2 -f /mnt -Rb iozone.xls
stream -O2 -fopenmp -DNTIMES = 10 -DSTREAM_ AR-
RAY_SIZE=10000000
```

```
netperf -H IP -t TCP_STREAM -l 60
```

测试结果如表 1 所示, 对比发现, 人工测试的时, 随着测试设备台套数的增加, 每台设备的测试任务执行时间相差不大, 但需要对每台测试服务器进行工具安装、测试执行、结果收集, 总的测试用时呈线性增长趋势。使用平台测试时, 虽然设备台套数由 40 台增加为 90 台, 但双路 CPU 的总核数为 128, 默认配置下可以并行处理 128 个测试任务, 多台设备的进行工具安装、测试执行、结果收集用时与单台设备基本相同, 总的测试用时基本不变, 使用平台的测试效率有大幅度的提升, 随着测试台套数的增加, 使用平台的优势愈发明显。

表 1 测试结果 min

性能测试项	人工 40 台	平台 40 台	人工 90 台	平台 90 台
IOZONE	28	15	62	16
STREAM	17	<1	38	<1
NETPERF	23	2	55	2

5 结束语

本文设计并实现了一种国产服务器的测试平台, 并通过实际测试验证了其有效性。该测试平台通过 Celery 实现了性能测试和稳定性测试项目的异步并行执行和指定顺序并行执行, 通过 Django 实现了测试过程的可视化管理, 使用 Docker 快速灵活的搭建测试环境, 该平台的投用大大提高了国产服务器的测试效率, 保障了国产服务器产品的研发和交付。

参考文献:

- [1] 邵红, 李亚伟. 国产服务器操作系统测试策略及测试方法研究 [J]. 电子测试, 2021, 000 (16): 87-89.
- [2] 邓蕊, 王龙涛, 李亚伟. 国产服务器操作系统测试方法研究 [J]. 电子技术与软件工程, 2023 (7): 26-29.
- [3] 李行善. 自动测试系统集成技术 [M]. 北京: 电子工业出版社, 2004.
- [4] 孙立哲. 轻量级接口自动化测试框架设计与实践 [J]. 计算机应用与软件, 2020, 37 (1): 27-30.
- [5] 张蔚瑶, 张磊, 毛建筑, 许智君, 张玉军. 未知协议的逆向分析与自动化测试 [J]. 计算机学报, 2020, 43 (4): 653-667.
- [6] 李航, 董安明, 禹继国, 等. 基于前后端分离架构的智慧农业物联网系统设计 [J]. 现代电子技术, 2022 (14): 45-49.
- [7] 杨幸, 龚琦, 范文晶, 等. 某发控设备一体化自动化测试系统研究 [J]. 计算机测量与控制, 2023, 31 (6): 53-58.
- [8] 李一风. 基于 Celery 和 Django 的分布式自动化测试系统设计 [J]. 信息技术, 2019, 43 (5): 4-8.
- [9] 吴斌. 基于 Django 和 Celery 框架的进度管理平台的设计与实现 [D]. 武汉: 华中科技大学, 2018.
- [10] 丁明一. Linux Shell 核心编程指南 [M]. 北京: 电子工业出版社, 2019.
- [11] 董锐. 基于 Django 的 IT 运维管理平台的设计与实现 [D]. 武汉: 华中科技大学, 2020.
- [12] STIGLER S, BURDACK M. A practical approach of different programming techniques to implement a real-time application using Django [J]. Athens J. Sci, 2020, 7: 43-66.
- [13] 孙睿阳, 方信昀. 基于 VISA 和 Django 的自动化测试系统的设计与实现 [J]. 计量科学与技术, 2020 (10): 47-50.
- [14] 桑雷. 基于 Django 的手机自动化测试平台 [D]. 西安: 西安电子科技大学, 2021.
- [15] 孙光民, 路浩南. 基于 Django 的金属在线硬度检测系统的设计与实现 [J]. 计算机测量与控制, 2019, 27 (9): 23-27.
- [16] 陶瑜, GU XIANG PING. 基于 Bootstrap4 的网站前端响应式设计 [J]. 中国新通信, 2020, 22 (17): 53-54.
- [17] 陈甫. Bootstrap3 在 Java Web 项目中的应用 [J]. 电脑编程技巧与维护, 2014 (17): 27-28.
- [18] JAMAL, YAMANI, MUNDZIR, et al. Design and Implementation of Web Application for Attendance List of Lecturers Using Codeigniter and Bootstrap Framework [J]. Journal of Physics: Conference Series, 2021, 1807 (1): 12-30.
- [19] 吴逸文, 张洋, 王涛, 等. 从 Docker 容器看容器技术的发展: 一种系统文献综述的视角 [J]. 软件学报, 2023, 34 (12): 5527-5551.
- [20] YAN JIANG, WEI LIU, XUANHUA SHI, et, al. Optimizing the Copy-on-Write Mechanism of Docker by Dynamic Prefetching [J]. Tsinghua Science and Technology, 2021, 26 (3): 266-274.
- [21] 刘哲源, 吕晓丹, 蒋朝惠. 基于模拟退火算法的粒子群优化算法在容器调度中的应用 [J]. 计算机测量与控制, 2021, 29 (12): 177-183.