

# 高性能计算环境通用计算平台<sup>①</sup>

和 荣<sup>1,2</sup>, 王小宁<sup>1</sup>, 卢莎莎<sup>1</sup>, 赵一宁<sup>1</sup>, 肖海力<sup>1</sup>, 迟学斌<sup>1,2</sup>

<sup>1</sup>(中国科学院 计算机网络信息中心, 北京 100190)

<sup>2</sup>(中国科学院大学, 北京 100049)

通讯作者: 和 荣, E-mail: [herong@sccas.cn](mailto:herong@sccas.cn)



**摘 要:** 随着高性能计算环境的持续运行, 用户的计算需求快速增长, 对资源的需求不断上升. 环境急需扩充资源以提供更强大的计算能力, 满足不同应用领域的计算需求, 同时也对环境扩展性、易用性和可靠性提出更高要求. 高性能计算环境通用平台从底层设计到上层页面实现了全新的蜕变, 从用户角度出发设计了相关的功能模块, 为用户提供更加优质的服务. 本文围绕国家高性能计算环境通用计算平台展开叙述, 重点介绍平台的整体结构、采用的关键技术、相关模块以及实现过程和最终效果. 通过部署测试表明通用计算平台可满足现有用户的需求并将进一步扩大高性能计算环境的用户群, 推动高性能计算在更多领域的发展, 提高国家高性能计算环境影响力.

**关键词:** 高性能计算; 作业管理; 微服务; 数据安全; 磁贴

引用格式: 和荣, 王小宁, 卢莎莎, 赵一宁, 肖海力, 迟学斌. 高性能计算环境通用计算平台. 计算机系统应用, 2019, 28(12): 55-62. <http://www.c-s-a.org.cn/1003-3254/7176.html>

## Platform for High Performance Computing Environment

HE Rong<sup>1,2</sup>, WANG Xiao-Ning<sup>1</sup>, LU Sha-Sha<sup>1</sup>, ZHAO Yi-Ning<sup>1</sup>, XIAO Hai-Li<sup>1</sup>, CHI Xue-Bin<sup>1,2</sup>

<sup>1</sup>(Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)

<sup>2</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** With the continuous operation of high performance computing environment, computing requirements of accounts increased rapidly and demand for resources continually rise. The environment needs to extend resources to supply more powerful computing power and meet needs of different fields. Meanwhile, expansibility, easiness, and reliability also need to be improved. Platform for high performance computing environment is designed according to requirements of accounts to achieve the goal of supplying quality service. The study extends the following content around the platform for high performance computing environment, highlighting structure of the platform, related important technologies, main functions, and implementation process. Through deploying and testing, results show that the platform can meet requirements of existing accounts. In the future, it will extend user group of high performance computing environment and promote the development of high performance computing in more areas.

**Key words:** high performance computing environment; job management; micro service; data security; tile

高性能计算环境是现代社会科学、社会服务、经济活动中一种极为重要的战略工具. 我国高性能计算资源日益丰富, 多领域的大规模应用对计算能

力也提出了超高需求, 关键问题是如何充分、有效的组织和利用国家高性能计算环境<sup>[1]</sup>的资源满足这些需求. 国家高性能计算环境聚合国内优秀的高性能计算

① 基金项目: 国家重点研发计划课题 (2018YFB0204001)

Foundation item: National Key Research and Development Program of China (2018YFB0204001)

收稿时间: 2019-04-28; 修改时间: 2019-05-23; 采用时间: 2019-05-30; csa 在线出版时间: 2019-12-10

资源,面向用户提供便捷的高性能计算服务.从趋势来看,国家高性能计算环境不仅是计算的基础设施,也要成为一种公共服务,为广大用户群提供及时、优质和不间断的计算服务,既要支持大规模的并行计算,也要利用云计算和大数据的理念以最小的代价为用户提供服务,提供便捷的高性能计算服务.云计算的个性化定制和服务化是高性能计算未来发展的趋势.此外,大数据的发展也为高性能计算带来了新的机遇和具体应用,高性能计算已经成为大数据存储和处理的基础平台,提供大规模数据密集型计算所需的大容量、低延迟和高带宽的处理能力,为数据分析人员提供计算服务.

国家高性能计算环境目前包括底层环境关键模块、通用计算平台<sup>[2]</sup>、应用社区等.随着高性能计算环境的持续运行,用户的计算需求快速增长,对资源的需求不断上升.环境急需扩充资源以提供更强大的计算能力,满足不同应用领域的计算需求,同时也对环境扩展性、易用性和可靠性提出更高要求,环境关键模块<sup>[3]</sup>必须进行优化升级和功能扩展.

高性能计算环境通用计算平台从底层架构设计到上层页面显示实现了全新蜕变,包括对其功能进行扩展,根据用户的需求设计相关模块,为用户提供更加优质的服务.通用计算平台 Portal 2.0 以更好的满足用户计算需求,方便快捷的提交作业任务并获取计算结果为设计理念.针对用户提供个性化的高性能计算服务,以达到应用软件按需动态发布,支持用户定制计算软件和计算资源等个性化服务的目标.通用计算平台的建设将进一步扩大高性能计算环境的用户群,推动高性能计算在更多领域的发展,提高国家高性能计算环境影响力.

本文围绕国家高性能计算环境通用计算平台 Portal 2.0 展开叙述,重点介绍平台的整体结构、涉及的关键技术、主要功能以及实现过程和最终效果.

## 1 整体结构

高性能计算环境通用计算平台 Portal 2.0 主要包括底层的基础服务、服务端和客户端 3 部分组成.具体如图 1 所示.

### 1.1 基础服务

基础服务提供 Portal 2.0 的运行环境.它包括日志管理、SCEAPI<sup>[4]</sup>、Memcached、MySQL 和 Web 容器等组件.

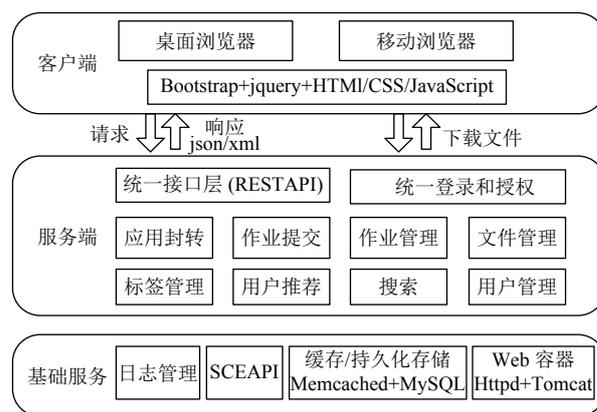


图 1 Portal 2.0 整体结构图

日志主要用来记录系统中的各种事件,方便出现错误时查找原因等.记录的日志还可以作为数据来源进行分析,从日志中提取有用的信息,如错误信息等,在以后的运维中有效避免类似错误.SCEAPI 是中国科学院科学计算环境 (Scientific Computing Environment, SCE)<sup>[5]</sup>的跨接口和语言的通用接口.SCEAPI 接口包括用户、资源、作业、数据和监控等多个方面的功能.为保证数据的安全性,有效避免直接操作数据库,所有获取环境数据的请求必须通过调用 SCEAPI 接口实现.Portal 2.0 使用 MySQL 数据库保存用户的数据等相关信息.为避免重复访问接口,减少服务端负载统计信息等采用 Memcached 机制.Web 容器主要保证系统运行,处理各个模块之间的端口转发.

### 1.2 服务端

服务端主要利用基础服务中的组件开发的功能模块.作业提交、作业管理、文件管理是通用计算平台必备的核心功能.作业提交服务完成 JSDL (Job Submission Description Language)<sup>[6]</sup>描述标签处理、JSDL 作业描述文件完成、文件上传、向 SCE 中间件提交作业等功能.作业管理服务实现向用户展示实时作业的状态信息和终止作业功能.文件管理涉及文件实时查看和无缓存数据流下载完成实时作业工作目录文件列表、文件属性和文件内容查看、大文件下载功能.

统一登录和授权为技术人员使用 SCEAPI 进行社区或网站开发提供便捷服务.开发人员无需重新开发登录模块,可直接调用统一登录模块为系统添加登录功能.授权机制为各个服务访问的安全准确提供保障.

### 1.3 客户端

开发人员面向用户提供的服务,主要是指浏览器

端的 Web 页面. 用户通过客户端发送各种请求到服务端, 服务端响应后返回数据显示在客户端. 如通过提交作业页面可提交作业并查看作业状态信息; 在作业运行过程中可实时在线浏览作业文件信息或下载文件. 在此过程中只需输入作业的基本信息、参数, 点击输入文件上传到服务端, 即可提交一个作业, 极大的简化了用户提交作业的流程, 使用起来也更加方便.

客户端页面主要采用 HTML5、CSS、JavaScript 实现. 客户端从服务端获取 XML 或 JSON 格式的数据后根据页面的设计呈现给用户.

## 2 关键技术

考虑到不同用户的使用习惯和模式, 不同科学领域计算任务的特点, 不同应用软件的特点, 以及当前网络发展的趋势, 高性能计算环境通用计算平台重点提出了以下 4 方面的关键技术: 微服务开发、应用按需发布、关联性搜索、数据安全机制.

### 2.1 微服务开发

高性能计算环境通用平台涉及到多个模块, 包括作业管理、作业提交、用户管理等 (图 2). 为了方便每个模块相互独立的开发部署, 提出了微服务<sup>[7,8]</sup>的开发方式.

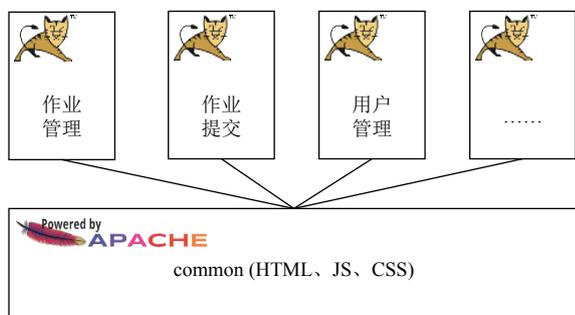


图 2 微服务开发结构

为更好地实现微服务的开发方式, 我们从以下 3 个方面来实施. 首先, 每个模块的公共文件如页面的头尾、JS、CSS 存放到公共的位置, 并单独部署在一个 Tomcat 服务器上, 功能模块在实现时只需引用相应的页面或 JS. 其次, 针对服务器端代码, 分析并提取通用的代码, 底层 API 接口的调用封装成通用接口, 并将整个代码封装成代码包. 开发人员开发相应模块时可直接调用该代码包, 复写相应的接口, 这样对开发者来说特别方便快捷, 不需要写重复代码, 简化整个开发流

程. 最后, 各个功能模块开发完成后单独部署在 Tomcat 服务器上, 通过 Apache 配置端口转发实现同一域名访问.

通过实现微服务开发, 各个模块之间可独立开发部署. 当更新或修改某个模块时, 平台中其他模块可以不受影响而继续提供服务.

### 2.2 应用按需发布

在处理高性能计算任务时, 按照计算环境指定的方式描述作业的各种参数和输入文件, 不用学科领域的应用软件, 要求的参数和输入数据各不相同. 其次, 按照作业管理系统命令格式的要求, 组织数据并将计算任务提交到计算环境. 最后, 可在计算环境中管理作业, 包括查看作业状态, 挂起和终止作业, 获取计算结果等.

为了实现应用的快速发布, 解决不用学科领域应用封装问题或作业提交页面快速生成问题, 平台提出了一个应用封装便捷工具 HPC-AIT<sup>[9]</sup>. HPC-AIT 以通用的作业描述语言 JSDL 规定的作业提交信息为出发点, 基于 JSR-168 标准的 Portlet, 支持复杂作业参数的 Web 页面自动填充.

图 3 为应用集成组件的技术路线图<sup>[9]</sup>. 首先分析多种类型的应用需求, 设计描述模板. 管理员在 Web 图形化页面填写详细的应用需求, 后台服务根据模板生成应用描述文件; 其次, 渲染引擎根据应用描述文件提供的数据, 替换 HTML 模板中的表达式, 生成作业提交页面; 作业提交页面中自带默认参数, 用户也可修改其中的作业参数, 作业参数和输入数据传输到后台服务器, 后台服务器保存这些信息同时生成 JSDL 描述文件. 最后, 提交作业描述文件和输入数据到计算环境, 完成计算任务的提交功能.

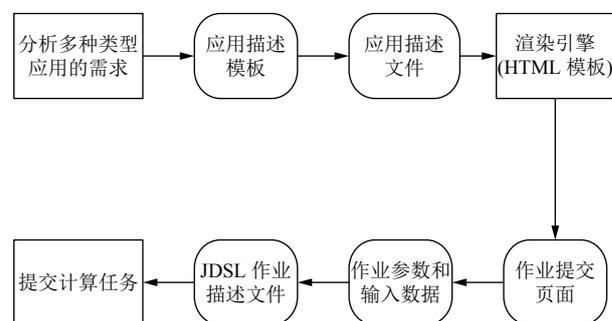


图 3 高性能计算应用集成组件技术路线图

下面以 Gaussian 应用为例来说明如何快速实现应用封装和作业提交. 运行封装工具 HPC-AIT 可获得 Web 页面上应用的封装, 其中包括页面的固定信息和

动态信息. 针对应用名称、执行程序、版本、应用描述等固定信息, 管理员只需要在页面上填写相应内容, 无需修改其他标签等数据; 输入文件等动态信息定制页面, 根据应用自身的特点选择文件输入框, 比如 Gaussian 应用只需要一个输入文件, 所有只需勾选一个输入文件框即可. 所有设置结束后可得到自动生成的作业提交页面. 为了能够在平台中使用 Gaussian 应用, 需要将自动生成的作业提交页面加载到通用计算平台中, 这样就实现了一个完整的应用发布.

通过实现应用按需发布, 管理员可有效提高应用封装的响应速度和降低操作难度, 用户能够方便的提交作业和重用复杂的作业参数, 更好的为实现用户的个性化服务提供便利.

### 2.3 关联性搜索

用户为快速找到自己想用的资源或应用经常会通过关键字进行搜索. 为了保证搜索结果更详细准确, 我们采用关联性搜索的技术.

前端服务器接受客户端请求后向后端搜索服务发起请求. 后端搜索服务通过 ES (Elastic Search)<sup>[10]</sup>实现. 核心功能是将搜索请求按照 ES 定义的访问格式, 其中需要特别定义对资源对象用户名属性的值匹配: ( $u|any$ ), 其搜索结果将包括所有公共资源和该用户提交的作业资源.

后端搜索服务在调用 ES 执行搜索前, 将搜索请求中包含的搜索字符串分解成一组搜索关键词  $W$ , 同时获取用户名  $u$ , 然后向 ES 发送搜索命令, 形式为 ( $must(should(u) | should(any)) \& must(should(W1) | should(W2) | \dots | should(Wn))$ ). ES 服务器执行搜索命令后会返回搜索结果, 访问接口将分解结果并形成一个包含资源所有分段信息的命中数组, 并返回给前端接口服务, 然后转发至客户端.

用户搜索的同时后端服务会根据相关搜索结果给出相应的推荐内容以使用户了解更多相应信息. 具体操作: 获取搜索结果中的标签, 并将其作为推荐的关键词; 由标签组成新的搜索请求并发送给 ES 服务器; 对 ES 返回的结果按照资源类型 (集群、应用等) 进行分类, 并将相关度最高的前 3 条作为推荐结果返回到前端接口.

关联性搜索的后端服务是通过 ES 实现的, 所以为实现关联性搜索必须搭建一个 ES 环境. 环境建好后导入资源数据以便搜索时使用. 目前已导入的数据主要包括用户的作业信息. 数据整合成 JSON 格式后通过

ES 命令导入到索引中. 后端服务通过发送 ES 指令便可获取搜索结果. 为了给前端更好地提供服务, 封装 ES 搜索指令为 REST 风格的接口. 用户只需发送请求即可获取结果, 使用方便快捷.

### 2.4 数据安全机制

在通用计算平台设计中为保证数据的安全, 主要从输入框数据格式、服务端检测、修改信息邮件提醒等方面进行限制检测.

在输入框数据格式上, 针对用户所输入的数据必须不包括特殊字符. 如联系电话: 手机号/ $^1[0-9]\{10\}[/math>; 电话号码/ $^0[0-9] \{2,3\}-[0-9]\{7,8\}[/math>; 项目日期: / $^-[0-9]\{6\}-[0-9]\{6\}[/math>.$$$

服务端检测主要包括验证用户输入的邮箱是否是真实的邮箱; 用户修改密码时验证原密码是否正确; 服务器的格式审查.

用户修改信息时会针对不同的修改内容通过邮件提醒用户是否是自己的行为, 避免恶意操作. 修改过程中会记录修改的 trace, 方便以后查找记录.

在平台实现过程中, 每个页面都有一个 validate.js 来验证输入数据的合法性; 服务端采用 Jersey 框架实现, 其中会对相关参数进行检查保证没有语法错误; 另外各种操作均记录相应 trace, 保证出错时可查询. 总之, 通过各种手段保证用户在使用平台时的数据安全.

## 3 模块设计

高性能计算环境通用计算平台主要是为用户提供计算服务, 方便用户通过 Web 端提交作业并查看作业结果, 所以平台设计了提交作业、作业管理、个人信息、应用、集群 5 大模块, 同时支持关键字搜索功能. 每个模块的功能介绍如表 1 所示. 下面依次介绍每一个模块, 重点突出提交作业和作业管理.

表 1 功能模块

模块名称	功能介绍
提交作业	作业提交
作业管理	查询作业信息并查看下载结果文件
个人信息	修改个人信息或密码
应用	列出平台所有应用
集群	列出平台所有集群

### 3.1 提交作业

提交作业模块如图 4 所示, 包括顶部应用菜单导航和应用提交页面.



图4 应用提交页面

应用菜单导航列出所有可通过平台提交的应用。菜单列表按照用户点击顺序设计,默认第一项为上一次提交的作业应用。用户常用的应用显示显示,其他应用隐藏显示,如需查看所有应用可点击扩展符号。点击应用菜单可切换应用提交页面。

应用提交页面为应用封装时定义发布的页面,根据每个应用使用方式的区别,提交页面参数以及输入文件不同。通过梳理作业的输入参数,将参数设置为作业参数和高级参数两类,高级参数为不常用或不经常修改的参数,默认隐藏显示。应用资源列表随着用户输入的CPU核数、作业运行时间等不同而实时变化,用户可以通过搜索框输入相应信息选择自己想用的资源。如果显示的资源没有自己的想要的资源,可点击展开更多资源查看不可用的全部资源,鼠标滑过每一行可显示资源不可选择的原因。

用户输入作业参数、上传文件时会做相应的语法以及文件类型检查,当用户输入错误或用户输入文件不正确直接提示错误并给出详细错误信息。用户输入参数和输入文件都通过校验后,通过点击提交作业,会把用户的相关参数写入JSDL文件并把信息传递给服务端后台,由服务端调用底层API接口把作业提交到环境中执行。提交作业成功后会弹框提示作业提交成功,用户可选择留在此页面或进入作业页面查看提交作业的相关信息。

### 3.2 作业管理

作业管理模块主要是方便用户查看作业信息,快速了解作业的运行状态。具体设计如图5所示,作业列

表随着查询条件的不同实时刷新,并支持分页显示,支持多维度作业条件查询,包括时间(最近一天、最近一周、最近一月)、类别(排队、运行、已完成、已删除);多种条件组合查询。通过更多选项进行高级筛选,包括应用名称、集群名称、开始时间、结束时间、作业名称和作业号。用户可输入相应信息精确的查找作业。



图5 作业查询页面

在作业列表中,根据作业的状态备注中给出相应信息和操作,具体如表2所示。作业处于完成状态,备注给出相应的运行时长,用户可点击操作中的删除按钮删除作业;作业处于提交出错状态,备注显示失败详情,鼠标移到上面显示出错原因;作业处于排队状态,备注给出排队时长,如排队时间过长,显示排队时间过长的原因,用户可点击操作中的终止按钮终止作业;作业处于删除状态,用户可点击操作中的恢复按钮恢复作业。

表2 作业状态对应的备注与操作

作业状态	备注	操作
完成	作业耗时	删除
提交出错	失败详情	-
排队	排队时长	终止
运行	运行时长	终止
完成	作业耗时	删除

点击作业号进入作业文件页面,用户可在线查看或下载作业结果文件。文件查看页面支持按行查询和分页显示,方便用户直接查找相关结果或通过关键信息判断作业是否执行正确。对于固定输出格式的文件,用户也可输入行号直接查询定位所需内容。

### 3.3 个人信息

在用户使用平台时经常遇到邮箱变化或登录密码

忘记等情况,为了在通用计算平台上我们设计了个人信息模块,用户可修改个人基本信息、项目信息以及登录密码。

用户在修改个人基本信息和项目信息的同时会记录相应的 trace 文件,同时邮件通知用户以确保是用户自己的行为,避免恶意修改。在用户修改的过程中,对各项输入项都做了严格的限制来保证数据的安全。

用户修改密码需要验证原始密码的正确性,根据输入的新密码给出密码强度提示。对太过于简单的密码直接提示用户需重新输入,修改密码之后会邮件通知用户,并要求用户重新输入新密码登录平台。

### 3.4 应用集群

应用和集群页面主要让用户了解平台中的所有应用和集群基本信息。页面支持未登录情况下查看,通过这两个页面可清楚地了解平台中是否有自己可使用的应用或关注的集群。应用和集群页面采用仿 Win8 的设计,每一个应用或集群以磁贴的形式展示,并对其颜色进行了搭配,外观清楚简单,如图 6。



图 6 应用页面

用户鼠标放在磁贴上显示编辑颜色按钮,点击编辑颜色可自行修改配色<sup>[11,12]</sup>。页面首次以系统默认配色显示,如用户修改某一项配色,显示用户修改后的配色。

高性能计算环境通用平台通过提交作业和作业管理模块达到了用户采用传统命令行方式使用高性能计算环境的效果。用户只需在 Web 页面修改应用相关参数,上传文件即可实现作业的提交与运行。对于对应用使用命令不是很熟悉的处理用户,高性能计算环境通用平台使用起来尤其方便。同时,平台提供推荐搜索功能,根据用户的需求快速的推荐资源,提供个性化服务。为了方便用户使用,平台采用的关联性搜索技术使得用户非常方便的获取到自己的资源。应用查看与提交

之间通过超链接彼此关联,使用特别灵活。高性能计算环境聚合了国内优秀的高性能计算资源,包括 6 个国家的超级计算中心,只要用户提交应用时满足各个队列的使用条件就可以调度到各大超算中心上进行计算。相对而言,计算和存储资源都具有一定的优势。平台的性能也相关比较乐观,可满足用户的需求,具体性能指标会在第 5 节中详细论述。

## 4 实现

按照前面叙述的模块功能和设计,客户端采用 HTML5+CSS+JavaScript 实现。页面采用 Backbone Layout<sup>[13]</sup>布局,内容包括 header、body 和 footer 3 部分。header 和 footer 引用公共页面, body 通过定义模板加载。服务端采用 Jersey<sup>[14]</sup>(Java 编写的轻量级框架)开发。服务端调用底层 API 接口获取数据供浏览器端调用。为区分登录和匿名登录,服务端通过两套系统实现,浏览器端通过 js 判断是否登录,匿名登录情况向匿名登录服务端发送请求;登录情况下向登录服务端发送请求,获取数据后渲染到页面。实现过程对用户来说是透明的,用户看到的页面没有区别,只有显示数据有差异而已。

通用计算平台开发过程中同时考虑了国际化的需求,采用 JQuery.i18n<sup>[15]</sup>来实现。国际化配置包含两部分内容:一部分是国际化变量配置文件,扩展名为 properties;一部分是国际化处理文件,扩展名为 js,包括 jquery.i18n.properties.js(提供实现国际化的 api)和 portal2\_i18n.js(自己编写的处理文件)。portal2\_i18n.js 主要设置语言环境、加载配置文件后实现语言切换功能。

## 5 部署测试

通用平台目前部署在测试环境的服务器中,Java 版本 1.7, Tomcat 为 7.0.68。开发过程采用了微服务的开发模式,部署时提交作业、作业查询和个人信息、应用集群和搜索分别部署在 Tomcat 容器中。平台借助 Apache 配置端口转发,实现域名访问。部署完成后对平台的功能、安全、兼容性以及性能等进行了相关测试。

根据平台提供的功能,设计相关用例对其进行功能测试并及时修复测试过程中出现的问题。测试结果表明,高性能计算环境通用计算平台可满足用户提交作业、查看作业等各种需求。为更好地体现平台的有效性,除功能测试之外还对平台的安全、兼容性以及

性能进行了测试. 根据常见的安全性测试点设计测试用例并验证了平台的安全性. 表 3 是浏览器兼容性测试, 测试结果表明平台在浏览器兼容性方面考虑了比较完善, 可支持主流的浏览器.

表 3 浏览器兼容性测试

浏览器	测试结果
IE 11	网页正确显示
360 浏览器 9.1.0.325	网页正确显示
火狐浏览器 61.0.1(32 位)	网页正确显示
谷歌浏览器 55.0.2883.87	网页正确显示
QQ 浏览器 9.6.5	网页正确显示

图 7 是针对搜索不同的关键词时的平均搜索时间, 横坐标为搜索的不同关键词, 纵坐标表示所需搜索时间, 单位: ms. 由图可以看出基本上可以控制在 1 s 完成内容的搜索并显示在页面上.

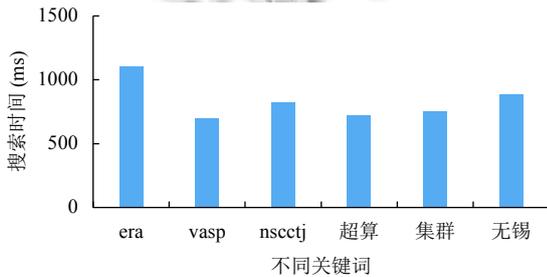


图 7 搜索不同关键词所需的平均时间

图 8 是平台中各个功能模块页面的平均响应时间, 横坐标为不同的模块页面, 纵坐标为响应时间, 单位秒. 由图可以看出首页基本上在 0.5 s 即可显示, 作业模块受作业数目的影响响应时间略长, 其余模块都在 1 s 之内.

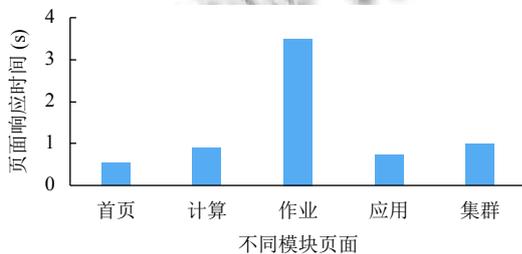


图 8 功能模块页面平均响应时间

利用 Jmeter<sup>[16]</sup>测试工具对平台的并发性进行了测试. 图 9 是不同并发数目下的请求响应时间, 其中横坐标为模拟的并发用户数, 纵坐标为响应时间, 单位: s.

由图可以看出, 随着并发用户数的增加平均响应时间也随着增长, 但最短时间基本没有变化.

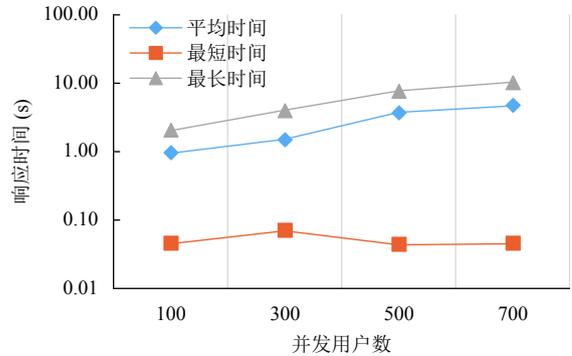


图 9 不同并发数目下的请求响应时间

图 10 表示不同并发数目下的吞吐量, 横坐标跟图 9 一样表示模拟的并发用户数, 纵坐标为每秒处理的请求数. 随着并发用户数的增加, 吞吐量基本维持在 50-70 之间. 这表明平台在并发用户数比较少时响应时间可以维持在 1 s 之内, 但过大时会变的比较长. 后期工作中会对并发性进行改善使其性能更好.

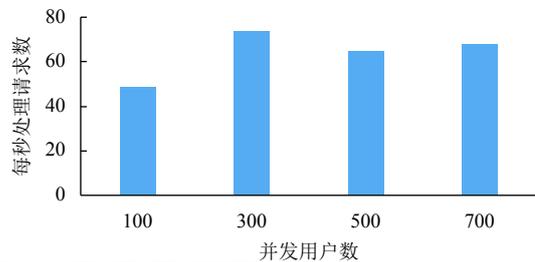


图 10 不同并发数目下的吞吐量

为了更好地调查用户的需求和满意度, 平台开放给部分用户试用, 试用结果表明平台不仅可以满足用户的需求还可以方便的为用户推荐资源.

## 6 结论与展望

高性能计算不断发展, 越来越多的应用需要通过高性能计算环境运行. 随着高性能计算环境的持续运行, 用户的计算需求快速增长, 对资源的需求不断上升, 环境急需扩充资源以提供更强大的计算能力, 满足不同应用领域的计算需求, 同时也对环境扩展性、易用性和可靠性提出更高要求. 为满足用户的需求, 本文就高性能计算环境通用计算平台新的设计理念, 如何为用户提供个人性服务展开论述, 包括平台的整体结

构、关键技术、功能模块. 重点突出了计算平台中的关键技术微服务开发模式、应用按需发布、关联性搜索以及数据的安全机制; 平台主要的功能模块提交作业、作业查询.

经过部署测试表明平台已达到满足用户需求的目标, 但同时还存在一些问题, 如封装应用不够多、应用版本不够新等, 这些问题在后续的开发过程后会逐渐解决, 使得高性能计算环境通用计算平台为用户提供更友好的服务. 除此之外, 计算平台还需考虑做好上层接口以便支持接入更多的应用社区, 达到用户登录通用计算平台即可利用国家高性能计算环境资源的目标.

### 参考文献

- 1 中国国家网络. <http://www.cngrid.org>, 2016.
- 2 曹荣强, 迟学斌, 武虹, 等. 基于 Portlet 的高性能计算 Portal. 计算机工程, 2009, 35(15): 1-3.
- 3 戴志辉. 三层架构超级计算环境优化设计与实现研究[博士学位论文]. 北京: 中国科学院研究生院, 2011.
- 4 曹荣强, 肖海力, 王小宁, 等. 基于 REST 风格的科学计算环境 Web 服务 API. 科研信息化技术与应用, 2016, 7(3): 43-48.
- 5 中国科学院超级计算环境. <http://cscgrid.cas.cn>, 2016.
- 6 Anjomshooa A, Brisard F, Drescher M, *et al.* Job submission description language (JSDL) specification, version 1.0. Open Grid Forum, GFD. 2005. 56.
- 7 张晶, 黄小锋, 李春阳. 微服务框架的设计与实现. 计算机系统应用, 2017, 26(6): 259-262. [doi: 10.15888/j.cnki.csa.005796]
- 8 李红健. 微服务架构和容器技术应用分析. 无线互联科技, 2018, 15(8): 134-135.
- 9 曹荣强, 王小宁, 卢莎莎, 等. 基于 Portlet 的高性能计算应用集成组件. 科研信息化技术与应用, 2017, 8(2): 18-30.
- 10 Elasticsearch. <https://www.elastic.co/products/elasticsearch>. [2019-03-31].
- 11 任琦, 徐迎庆, 庄越挺, 等. 面向办公应用的自动配色方案创作与应用系统. 软件学报, 2005, 16(5): 691-699.
- 12 李雪瑞, 王小平, 余隋怀, 等. 智能配色设计衍生方法研究. 计算机工程与应用, 2010, 46(11): 70-72, 134.
- 13 Backbone.js. <http://backbonejs.org/>. [2019-02-19].
- 14 Jersey. <https://jersey.github.io/>. [2019-03-12].
- 15 JQuery.i18n. <https://www.javascripting.com/view/jquery-i18n>. [2017-10-13].
- 16 JMeter. <http://jmeter.apache.org/index.html>. [2019-05-28].