

跨平台的一站式算力服务管理系统

陈家慧, 刘 杰

(上海超级计算中心, 上海 201203)

摘 要: 为支撑前端在线算力服务, 满足资源接入、授权使用和操作合规方面的资源使用需求以及中心管理需求, 设计并开发了跨平台的一站式算力服务管理系统。系统基于Spring架构和Hibernate框架构建, 采用Java语言及模块化方法进行系统开发, 并利用大数据技术分析处理系统日志。通过构建基于状态感知的全流程自动化闭环接入模型, 将集群资源接入从传统的手动或半自动化脚本操作转变为以服务状态为驱动的智能闭环流程。系统设计基于用户组的账号与资源映射模型, 简化多集群环境下的用户权限与资源管理, 帮助管理员实现集约化管理。测试结果表明, 算力服务管理系统较好地实现了集群管理、软件管理、账号资源管理、操作审计等功能, 可有效支撑面向用户的高性能计算服务。

关键词: 算力服务; 管理系统; 一站式; 跨平台; 高性能计算

DOI: 10.11907/rjdk.251574

中图分类号: TP317

文献标识码: A

文章编号: 1672-7800(2026)006-0122-06

扫描二维码阅读全文:



Cross Platform One-Stop Computing Power Service Management System

CHEN Jiahui, LIU Jie

(Shanghai Supercomputer Center, Shanghai 201203, China)

Abstract: A cross platform one-stop management system has been designed and developed to support front-end online computing power services, meet the usage needs and management demands for resource access, authorized use, and operational compliance. It is built on Spring architecture and Hibernate framework, developed using Java language and a modular approach, and analyzed and processed system logs using big data technology. The system constructs a fully automated closed-loop access model based on state perception, transforming cluster resource access from traditional manual or semi-automated script operations to intelligent closed-loop processes driven by service status. An account and resource mapping model based on the user groups is designed to simplify user permissions and resource management in multi-cluster environment and assist administrators in achieving intensive management. The test results indicate that the computing power service management system effectively implements cluster management, software management, account resource management, operation auditing, and other functions, which can effectively support high-performance computing services for users.

Key Words: computing power service; management system; one-stop; cross platform; high-performance computing

0 引言

2019年, 习近平总书记指示: “要以我为主, 掌握核心技术, 国家重大科技领域都需要超级计算机, 超级计算机非常重要, 要高质量研制好新一代超级计算机, 发挥引领作用。”^[1] 作为科学的第三种手段, 高性能计算(High Performance Computing, HPC)利用超级计算机实现并行计

算的理论、方法及应用, 可帮助人们探索科学、工程及商业领域的一些世界级重大难题, 服务国家重大需求^[2-5]。

2024年, 中国HPC TOP100排行榜前10名的计算机中有半数部署在超算中心^[6]。为了让计算普惠化, 使计算资源服务于更多科学家, 各超算中心开展了一系列工作, 包括提供基于网页的高性能计算服务系统, 例如中科院中国科技云、无锡超算高性能仿真平台、广州超算星光超算应用平台、天津超算天河HPC云平台等^[7-12]。这些系统的共

收稿日期: 2025-11-27

基金项目: 上海市城市数字化转型专项资金项目(202501003)

作者简介: 陈家慧(1972-), 女, 硕士, 上海超级计算中心高级工程师, 研究方向为高性能计算及云计算技术应用、业务平台测试与质量评价; 刘杰(1984-), 男, 硕士, 上海超级计算中心高级工程师, 研究方向为高性能计算及云计算技术应用、业务平台研发。
本文通讯作者: 陈家慧。

同特点在于以应用为核心,设计了具有任务提交、管理和查看等功能的操作界面,使用户能够通过互联网远程、高效地使用计算资源。然而,各个系统因各超算中心的实际使用和管理需求不同而具有差异性,一个超算中心的系统并不一定适用于其他超算中心。因此,上海超级计算中心推出了符合自身定位及用户需求的在线算力服务平台,以提供专业、可靠的高性能计算服务。用户根据工作需要,可以通过网页上传输入文件、设置作业参数、提交作业,完成高性能计算任务。在算力服务平台上完成这一系列动作依赖于底层的计算、存储、软件等资源是否可用,以及用户的操作是否合规,因此有必要开发一站式跨平台的算力服务管理系统。该系统具备异构资源快速自动化接入、集群资源状态实时监控、算力资源授权使用和运维操作全链路审计能力,可为前端算力服务提供支撑,从而为用户提供顺滑的使用体验。

1 系统需求及设计

1.1 系统需求

算力服务管理系统作为在线算力服务的支撑系统,是用户“看不见”的部分^[13]。其为前台用户“看得见”的操作提供支持和服务,所以应具备稳定性、可移植性、可扩展

性、兼容性和及时性。

(1)稳定性。管理系统应持续稳定运行,并且能够承受各种恶意攻击。

(2)可移植性。管理系统应能在不同的操作系统和硬件平台上部署运行,当系统运行所依赖的环境发生变更时,相关人员无需修改代码即可轻松地实现系统的跨平台迁移。

(3)可扩展性。管理系统应可以根据新需求轻松实现功能扩展,因此要求现有功能之间逻辑清晰、松耦合,新功能的加入不会引起代码大幅变动。

(4)兼容性。管理系统的使用者不是单人,而是多人,每个人使用的浏览器可能是不同的,因此系统应实现多浏览器兼容。

(5)及时性。管理系统应具备及时响应能力,可以快速处理用户提交的申请并做好用户响应,提升用户体验。

1.2 总体设计

算力服务管理系统基于 MySQL 关系数据库^[14],采用 Spring 架构^[15]和 Hibernate 框架^[16]构建。系统通过 SSH 协议^[17]连接底层资源,通过跨平台、跨语言的 Web 服务接口^[18]调用第三方短信服务,实现对集群队列、应用软件、主机账号等的管理,提供支撑在线算力服务必备的一站式后台管理服务。算力服务管理系统架构如图 1 所示。

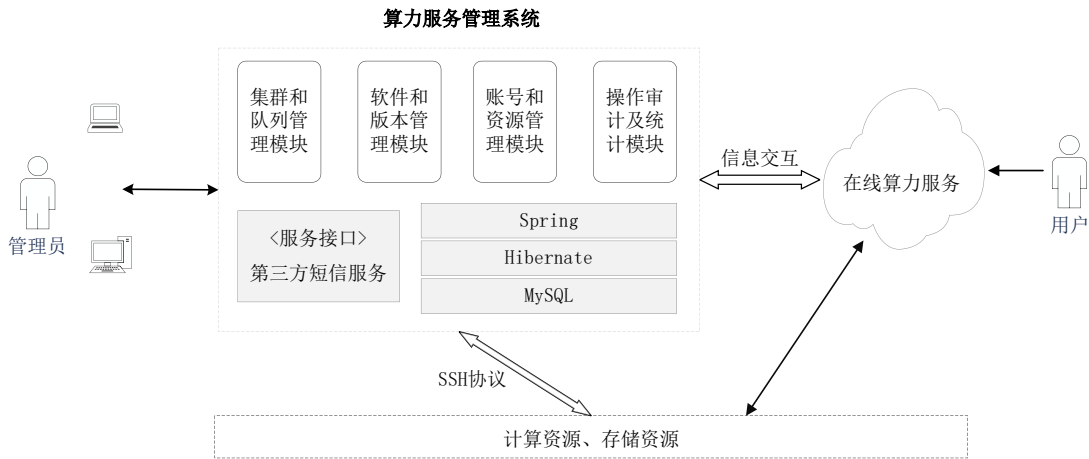


Fig. 1 Computing power service management system diagram

图1 算力服务管理系统示意图

算力服务管理系统根据需求进行设计:在稳定性方面,管理系统通过错误处理机制,及时捕获异常并作出响应,以避免系统故障,此外,通过操作审计功能记录各类操作行为,提供事后追溯能力;在可移植性方面,管理系统选择 Java 语言,支持在 Windows、MacOS 和 Linux 等操作系统与硬件平台上部署运行,实现系统的跨平台迁移;在可扩展性方面,管理系统采用模块化方法,将系统功能划分为集群和队列管理、软件和版本管理、账号资源管理、操作审计及统计 4 个模块,各模块功能相对独立,模块间松耦合,以降低系统的复杂度,使得软件易于控制和扩展;在兼容性方面,管理系统为 B/S 架构,采用 JavaScript、

HTML 和 CSS 编写代码,能够支持 Firefox、Chrome 和 Edge 等主流浏览器;在及时性方面,当用户提交新的资源申请时,系统除在网页展示申请信息外,还会以短信通知的方式告知管理员,使管理员可以及时获知用户请求并进行处理,减少用户等待资源审批的时长,切实提升用户的使用体验。

2 系统实现

算力服务管理系统根据在线算力服务对硬件、软件、用户三要素的使用及管理需求,实现集群与队列管理、软

件与版本管理、账号与资源管理、操作审计及统计等功能,为在线算力服务提供后台支撑。

2.1 集群与队列管理

集群与队列管理模块主要实现集群的增删改查、集群与队列监控功能。集群是提供在线算力服务的主体,由大量独立的计算机组成,用于大规模数据处理、模拟仿真等高性能计算任务^[19]。队列是集群上的分区,一个队列通常适用于某个特定的使用场景。用户提交一个计算任务,该任务会在计算集群的某个队列上执行。如何降低集群接入的复杂度,在线快速接入资源,为用户提供一个可用的计算环境,是集群和队列管理的难点。

传统的资源接入采用手动或半自动化脚本操作方式,依赖于操作人员的技术能力和经验,操作复杂度高且易出现人为操作错误,资源接入时间通常以小时计。设传统资源接入时间为 T_1 ,则 $T_1 > 1 \text{ hour}$ 。为了降低资源接入的复

杂度,提升资源接入效率,算力服务管理系统构建了基于状态感知的全流程自动化闭环接入模型。模型包括部署阶段(自动化执行流水线)、验证阶段(状态感知与智能决策)、运行阶段(闭环反馈及资源监控),形成“部署—验证—运行”的完整闭环。在部署阶段,系统通过SSH协议连接集群接入节点、自动推送集群服务安装包并部署服务、自动启动集群服务,以自动化流水线取代传统的手工操作,提升了资源接入效率。在验证阶段,系统通过主动检测服务启动状态来智能判断接入流程成功与否,并根据结果作出不同决策,以系统具备的状态感知能力实现了从自动化到智能化的跨越。在运行阶段,模型基于成功路径与失败路径形成闭环反馈,并实现对资源的监控,极大地提升了系统的可靠性。

集群资源接入模型图展示了从启动到形成闭环的完整流程,如图2所示。

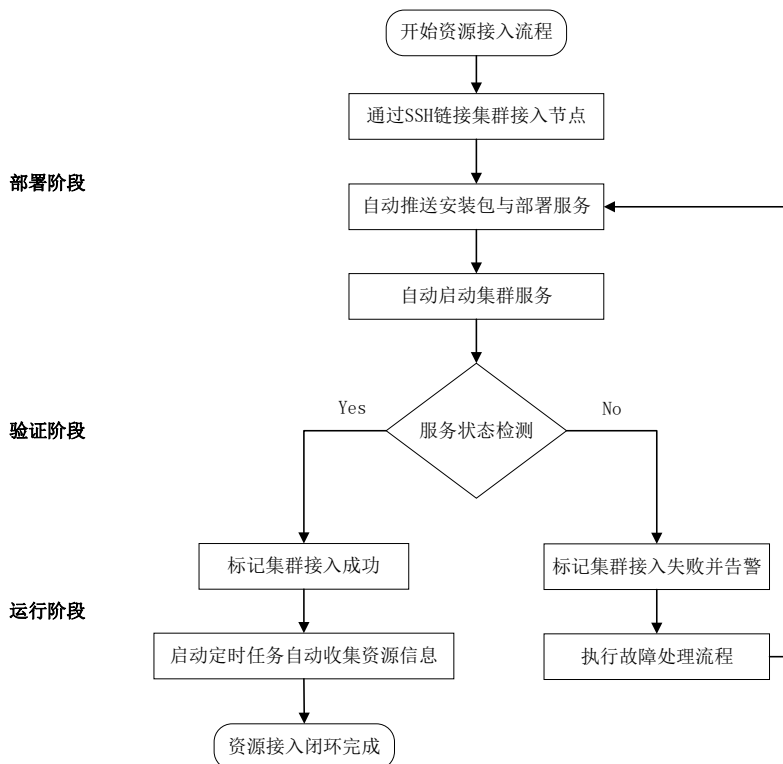


Fig. 2 Resource automated access model

图2 资源自动化接入模型

算力服务管理系统通过资源自动化接入模型,采用基于密码的安全验证方式,实现集群登录、集群服务部署、服务状态检测及资源信息(如队列等)采集的全自动化,将资源接入时间缩短至半小时。假设自动化资源接入时间为 T_2 ,则 $T_2 = 0.5 \text{ hour}$ 。由 $T_1 > 1 \text{ hour}$,得出 $\frac{T_2}{T_1} < \frac{1}{2}$,自动化资源接入时间比手动或半自动化脚本接入时间少一半以上,执行速度更快。

2.2 软件与版本管理

软件与版本管理模块支持添加软件、修改软件信息(包括软件类型和状态)和版本信息(包括运行队列),还支

持下线、克隆软件版本及修改运行脚本。用户通过集群上的应用软件执行高性能计算任务,应用软件通过运行脚本实现与集群的交互。因此,如何快速、便捷地修改软件运行脚本及添加软件版本,为用户提供良好的使用体验,是软件与版本管理的关注点。

算力服务管理系统通过网页形式的编辑窗口提供运行脚本修改功能,帮助管理员以shell语言在线编排计算任务^[20]。软件运行脚本定义集群名、队列、作业名、核数、单节点核数、作业参数等变量和计算任务执行逻辑,网页形式的在线脚本编辑操作相比命令行窗口操作更为便捷、直观,可减少编辑错误的发生,提升编辑效率。图3为在线

修改脚本示例。

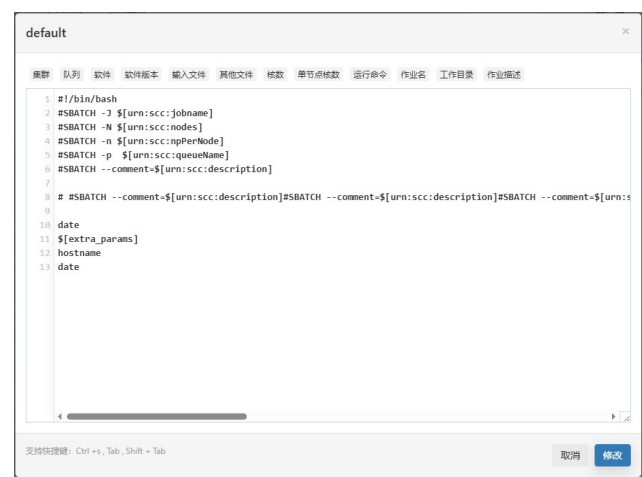


Fig. 3 Example of script editing

图3 修改脚本示例

软件运行脚本通常在反复编辑及测试后,才可以支持用户应用该软件执行高性能计算任务。当软件与版本存在1:N的对应关系时,如何减少软件运行脚本编辑的工作量,快速实现新软件版本的添加,是版本和软件管理的另一个关注点。

一个软件不同版本的运行脚本会存在差异,但差异并不是很大。算力服务管理系统可提供克隆软件版本功能,在管理员已经编辑过软件某版本运行脚本的前提下,可以

通过克隆功能将已有的版本信息复制成新版本。新版本名字为原版本名后加_copy,新版本状态、描述、脚本与原版本一致。管理员根据实际情况修改新版本名字、脚本等信息,即可快速完成新的软件版本添加,减少了管理员工作量。

2.3 账号与资源管理

平台账号是用户在算力服务平台上的身份许可,主机账号是用户在集群上的身份许可。用户以平台账号身份登录在线算力服务平台提交作业,实际上真正在集群上执行计算任务的是用户的主机账号身份。为保障用户正常使用算力服务,管理员需要为用户配置集群主机账号及资源使用权限。如果基于“用户—主机账号”两层映射关系进行账号管理,管理员需要为每个用户在每个集群上单独配置账号和权限,在多集群环境、多用户条件下,工作量大且易造成混乱。在实际应用过程中经常发现有一些用户属于同一组织结构,如企业某部门或科技机构某一项目组,该组织结构通常在一个集群上开设一个主机账号,供组织内用户共同使用。为适配用户组织结构,降低账号管理的复杂度,减少管理员工作量,算力服务管理系统构建基于用户组的集中式、层级化账号与资源映射模型,通过逻辑抽象简化多集群环境下的用户权限与资源管理,实现账号资源的集约化管理。账号与资源映射模型如图 4 所示。

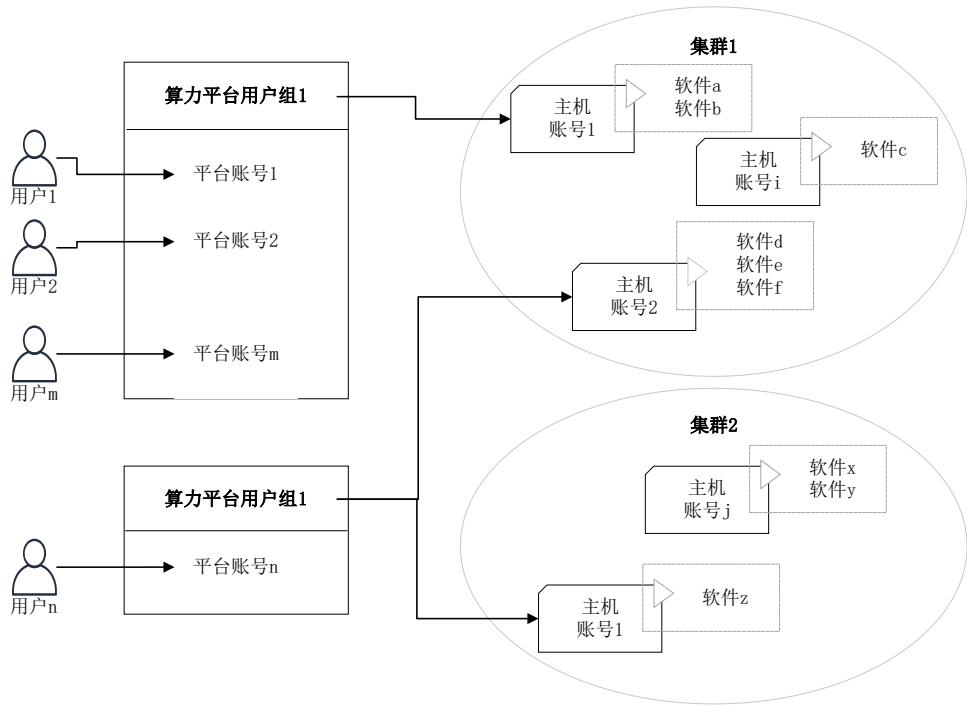


Fig. 4 Account and resource mapping model based on user groups

图4 基于用户组的账号与资源映射模型

账号与资源映射模型引入用户组这一逻辑层,作为连接平台用户和不同计算集群上实体主机账号的桥梁。一个用户只需加入一个用户组,即可自动获得该组在所有已授权集群上的资源使用权。每个用户在逻辑上隶属于一个

用户组,用户组可对应现实中的科研小组、项目组、部门等组织结构。在每个集群内部,用户组与主机账号为1:1关系。每个用户组可跨多个集群,一个用户组可以映射到多个集群上的主机账号,每个主机账号被授权使用所属集

群上的某些应用软件。账号与资源映射模型将传统的“用户—主机账号”映射关系转变为基于用户组的映射模型,更为适配算力使用单位的组织结构,同时易于实现账号的集约化管理。例如M个用户同属于一个用户组,在N个集群有主机账号,在“用户—主机账号”映射关系下,管理员需要进行 $M \times N$ 次操作才能完成配置,操作次数 $O_u = M \times N$ 。而在基于用户组的映射模型下,管理员只需针对用户组进行N次操作,而无需关注组内用户(组内用户由用户组自行在前台进行管理),操作次数为 $O_g = N$ 。 $O_g = O_u / M$,M越大, O_g 越小,大大减少了管理员工作量。

管理员通过账号资源管理模块在线设置用户组与主机账号的映射关系,并分配用户组应用软件权限。基于集中式、层级化的账号与资源映射模型,管理员只需在系统层面操作用户组与集群账号的映射关系,即可批量完成权限分配,管理效率显著提升。此外,管理系统会对主机账号下的资源进行监控,包括文件数、磁盘空间、核数、队列权限等。在资源告急的情况下,管理员可及时通知用户。

2.4 操作审计及统计

操作审计主要关注用户在系统上的操作行为,并进行记录、分析和监控^[21]。其核心价值在于保障数据安全、满足合规性要求,并提供事后追溯能力,这也是中心安全管理要求的内容之一。因此,在算力服务管理系统中可实现操作审计及统计功能。

操作审计利用大数据技术,以记录系统活动行为的日志为数据源,经过数据清洗、解析等过程,提取操作人、操作行为(如新增软件、分配主机账号、上线/下线软件版本、添加运行队列等)、操作IP和操作时间等重要信息,并在页面上以记录的形式作出展示^[22-23]。基于大数据技术的运维操作全链路审计,清晰记录了谁(操作人)在何时(操作时间)从何处(操作IP)做了什么(操作行为),为精准定位问题提供完整证据链。

统计功能提供机时统计等信息,并以折线图方式绘制作业、活跃用户、新增主机账号等趋势图,同时可以按时间粒度(天、周、月、年)筛选统计信息。可视化的趋势图可以提供直观的统计信息展示。这些均有助于管理员做好运维工作,降低运维风险。

3 测试与应用

3.1 系统测试

测试环境:系统部署在Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz、32 GB内存、200 GB硬盘的Linux服务器上,客户端为Intel i5-4590 @ 3.3GHZ、4 GB内存的Windows台式机,服务器与客户端的网络连接带宽为百兆。

功能测试采用黑盒测试方法,以检验算力服务管理系统所提供功能的可用性以及与需求的匹配度^[24]。经测试,算力服务管理系统提供的集群管理、软件管理、账号资源

管理、操作审计等功能完善,能够为前台的在线算力服务提供良好支持。

系统性能测试如下:①系统时延测试:测试客户端从发起请求至接收到请求结果的时延,客户端向算力服务管理系统发送“按关键词查询审计记录”请求,100次请求的平均时延为325.1 ms,测试结果表明算力服务管理系统响应能力良好;②负载测试:通过jmeter对系统逐步加压或者瞬间加压,在系统出错率超出5%时停止加压,记录系统的吞吐量和负载性能。测试结果如表1所示。由表1可知,系统在350~360个并发数时开始出现错误,表明在请求成功率不低于99.9%的指标下,系统负载能力为350个并发数,每秒可处理13个业务请求,满足使用需求。

Table 1 System load testing

表1 系统负载测试

逐步加压			瞬时加压		
并发数/个	吞吐量/个/s	出错率/%	并发数/个	吞吐量/个/s	出错率/%
50	14.2	0.00	50	12.7	0.00
100	14.6	0.00	100	12.8	0.00
200	14.7	0.00	200	13.7	0.00
300	15.3	0.00	300	13.7	0.00
350	13.7	0.00	350	13.7	0.00
360	12.5	1.62	360	13.6	3.61
375	14.1	3.73	362	13.6	3.87
376	14.9	5.59	363	14.3	6.34

3.2 应用示例

算力服务管理系统为工业生产、科学实验等领域的使用者提供算力支撑服务。目前,算力服务管理系统共接入4个异地异构资源,提供26.46 PFlops计算能力,服务超800位用户。资源信息如表2所示。

Table 2 Resources information

表2 资源信息

资源名称	峰值速度/ PFlops	计算节点/台	CPU	GPU
魔方III	3.4	1 120	Intel Xeon 2.6GHz	Nvidia Tesla V100
华东一区	7.68	1 600	Intel Xeon 2.7GHz	Nvidia A100
华东二区	7.68	1 600	AMD EPYC 2.0GHz	-----
西北一区	7.7	1 500	HYGON 7285	海光 DCU 2.5GHz

某钢铁研究院利用上述算力资源完成了工业产品加工及加工工艺方面的研究工作,例如箱罐封头冲压模拟、汽车天窗加强框轻量化设计、线材水冷温度均匀性研究、矫直机辊系结构优化分析等,服务了很多工程技术和研究人员,有效节省了大量仿真计算时间,使相关钢铁产品设计、工艺改进和设备优化等工作更有成效。

某电子科技有限公司通过中心提供的算力资源和专业技术服务,完成了平板探测器在承压和不同高度跌落工况下的结构仿真及设计优化工作,取得了良好效果,使生产研发环节更加紧凑,降低了企业的生产研发成本,提高了企业的竞争力。

4 结语

本文从用户使用需求和上海超算自身的管理需求出发,针对资源接入、资源使用和操作合规方面的问题,开发跨平台的算力服务管理系统,提供支撑在线算力服务的一站式后台管理服务。管理系统支持异构集群接入的全流程自动化与智能闭环,通过逻辑抽象简化多集群环境下的账号与资源管理,同时提供基于网页的计算任务在线编排,并对系统上的操作行为进行记录,提供事后追溯能力。测试结果表明,算力服务管理系统具备集群管理、软件管理、账号资源管理、操作审计等功能,系统质量良好。后续会根据技术与需求的变化,持续完善后台管理系统,以期具有类似需求的高性能计算支撑能力建设提供参考。

参考文献:

- [1] National Supercomputer Center in Tianjin. Putting the general secretary's independent innovation mandate into practice: Tianhe's three-year path of breakthroughs [EB/OL]. <https://www.nscctj.cn/xwxq.php?id=555>. 国家超级计算天津中心. 践行总书记对自主创新的重要指示,回顾天河的三年答卷之路[EB/OL]. <https://www.nscctj.cn/xwxq.php?id=555>.
- [2] SUN N H. High performance computing: the third tool of scientific research[J]. Communications of the CCF, 2006, 2(1): 20-21. 孙凝晖. 高性能计算:科学的第三种手段[J]. 中国计算机学会通讯, 2006, 2(1):20-21.
- [3] China Computer Federation. High performance computing [EB/OL]. https://test2.ccf.org.cn/Chapters/TC/TC_Listing/TCHPC. 中国计算机学会. 高性能计算[EB/OL]. https://test2.ccf.org.cn/Chapters/TC/TC_Listing/TCHPC.
- [4] Oracle. What is high-performance computing (HPC) [EB/OL]. <https://www.oracle.com/cn/cloud/hpc/what-is-hpc/?bcid=6260067197001>. 甲骨文. 什么是高性能计算(HPC)[EB/OL]. <https://www.oracle.com/cn/cloud/hpc/what-is-hpc/?bcid=6260067197001>.
- [5] SUN N H, TAN G M. Development and policy of high performance computer [J]. Bulletin of Chinese Academy of Sciences, 2019, 34(6): 609-616. 孙凝晖,谭光明. 高性能计算机发展与政策[J]. 中国科学院院刊, 2019, 34(6):609-616.
- [6] ZHANG Y Q, DENG L, YUAN L, et al. State of the art analysis of China HPC 2024 [J]. Computer Engineering & Science, 2024, 46(12): 2091-2098. 张云泉,邓力,袁良,等. 2024年中国高性能计算机发展现状分析[J]. 计算机工程与科学, 2024, 46(12):2091-2098.
- [7] QIAN D P. Building the new generation of computing infrastructure supporting technology innovation [J]. Frontiers of Data & Computing, 2020, 2(1): 1-17. 钱德沛. 构建支撑科技创新的新一代计算基础设施[J]. 数据与计算发展前沿, 2020, 2(1):1-17.
- [8] LIANG W J, ZHANG H H, ZHANG L L, et al. Design and implementation of CSTCloud portal management system based on MVVM [J]. Frontiers of Data & Computing, 2022, 4(2): 99-108. 梁文婧,张宏海,张蕾蕾,等. 基于MVVM模式的中国科技云门户管理系统的设计与实现[J]. 数据与计算发展前沿, 2022, 4(2):99-108.
- [9] National Supercomputing Center in Wuxi. High-performance simulation platform [EB/OL]. <https://www.nscctx.cn/Businessinfo/67/22.html>. 国家超级计算无锡中心. 高性能仿真平台[EB/OL]. <https://www.nscctx.cn/Businessinfo/67/22.html>.
- [10] National Supercomputing Center in Guangzhou. Starlight platform [EB/OL]. <https://starlight.nscg-gz.cn/>. 国家超级计算广州中心. 星光超算应用平台[EB/OL]. <https://starlight.nscg-gz.cn/>.
- [11] National Supercomputer Center in Tianjin. Tianhe HPC cloud [EB/OL]. <https://hpc.nscctj.cn>. 国家超级计算天津中心. 天河HPC云[EB/OL]. <https://hpc.nscctj.cn>.
- [12] Beijing Super Cloud Computing Center. Super computing cloud [EB/OL]. <https://www.blsc.cn>. 北京超级云计算中心. 超算云[EB/OL]. <https://www.blsc.cn>.
- [13] XU X X. Serverside development technology and practices [M]. Beijing: China Machine Press, 2016. 徐晓鑫. 后台开发:核心技术与应用实践[M]. 北京:机械工业出版社, 2016.
- [14] BOTROS S, TINLEY J. High performance MySQL [M]. 4th Edition. Beijing: Publishing House of Electronics Industry, 2022. BOTROSS, TINLEYJ. 高性能MySQL[M]. 第4版. 北京:电子工业出版社, 2022.
- [15] Spring. Spring home [EB/OL]. <https://spring.io/>.
- [16] Hibernate. Everything data [EB/OL]. <https://hibernate.org/>.
- [17] LONVICK C, YLONEN T. RFC4252: the secure shell (SSH) authentication protocol [EB/OL]. <https://www.rfc-editor.org/rfc/rfc4252>.
- [18] ZHOU X Y, CHEN W, WU G J, et al. REST API design analysis and empirical study [J]. Journal of Software, 2022, 33(9): 3271-3296. 周芯宇,陈伟,吴国金,等. REST API设计分析及实证研究[J]. 软件学报, 2022, 33(9):3271-3296.
- [19] YANG B, ZHUANG Y. Fault detection method of control cluster based on AOA-MSVM [J]. Computer and Modernization, 2023 (12): 112-117. 杨博,庄毅. 基于AOA-MSVM的控制集群故障检测方法[J]. 计算机与现代化, 2023(12):112-117.
- [20] JIANG S B, NI Z W. A brief talk about bottom-up Shell script programming and efficiency optimization [J]. Computer and Modernization, 2011 (2): 73-80. 江松波,倪子伟. 浅谈自底向上的Shell脚本编程及效率优化[J]. 计算机与现代化, 2011(2):73-80.
- [21] BaiduDeveloper. Practical guide to CentOS system operation and log audit [EB/OL]. <https://developer.baidu.com/article/details/3332675>. 百度开发者. CentOS系统操作与日志审计实战指南[EB/OL]. <https://developer.baidu.com/article/details/3332675>.
- [22] SHI S F. Big data analysis and dig [M]. Beijing: Posts & Telecom Press, 2018. 石胜飞. 大数据分析与挖掘[M]. 北京:人民邮电出版社, 2018.
- [23] HAN J R, JI Z X, LI Y M, et al. Trusted log storage and verification system based on blockchain [J]. Computer Engineering, 2019, 45 (5): 13-17. 韩菊茹,纪兆轩,李一鸣,等. 基于区块链的可信日志存储与验证系统[J]. 计算机工程, 2019, 45(5):13-17.
- [24] PAN W W, WANG X Y, SONG M L, et al. Survey on generating adversarial examples [J]. Journal of Software, 2020, 31(1): 67-81. 潘文雯,王新宇,宋明黎,等. 对抗样本生成技术综述[J]. 软件学报, 2020, 31(1):67-81.