

## 专刊：东数西算：开启算力经济时代的世纪工程（上）

Special Issue: East-to-West Computing Requirement Transfer: A Century Project to Start the Era of Computility Economy

ISSN 2096-742X  
CN 10-1649/TP文献CSTR:  
32002.14.jfdc.  
CN10-1649/  
TP.2022.05.004文献DOI:  
10.11871/jfdc.issn.  
2096-742X.2022.  
05.004

页码: 33-41

获取全文



## 基于高性能计算环境的HPC算力编程模式

王小宁<sup>1\*</sup>, 卢莎莎<sup>1</sup>, 吴璨<sup>1</sup>, 和荣<sup>1</sup>, 闫晓婷<sup>1,2</sup>, 肖海力<sup>1</sup>, 迟学斌<sup>1,2</sup>1. 中国科学院计算机网络信息中心, 北京 100083  
2. 中国科学院大学, 北京 100190

**摘要:** 【目的】随着云计算、大数据、人工智能等技术的兴起和广泛应用，促进了基于多元算力的融合计算发展。在国家“东数西算”战略的指引下，充分发挥HPC算力优势，提供新型HPC算力编程模式，是新一代计算基础设施可编程能力的重要变革。【方法】分析了高性能计算环境服务模式发展和现有计算环境下不同的编程模式，提出了基于高性能计算环境的HPC算力编程模式HPC as a function，定义了HPC算力和任务模式的基本抽象，以及HPC算力编程模式的参考体系结构。【结果】HPC算力编程模式可支持科研业务中融合计算对HPC算力的基本需求，可将适用于HPC的计算任务分发到合适的计算资源执行并有效管理，相比传统的工作流系统提供了更大的灵活性和可编程性。【结论】HPC算力编程模式可望有效提高“东数西算”计算基础设施的可编程性。

关键词: 国家高性能计算环境CNGrid; CNgrid系统软件; 融合计算; 算力网络; 编程模式

## HPC Computing Power Programming Paradigm Based on CNGrid

WANG Xiaoning<sup>1\*</sup>, LU Shasha<sup>1</sup>, WU Can<sup>1</sup>, HE Rong<sup>1</sup>, YAN Xiaoting<sup>1,2</sup>, XIAO Haili<sup>1</sup>, CHI Xuebin<sup>1,2</sup>1. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100083, China  
2. University of Chinese Academy of Sciences, Beijing 100190, China

**Abstract:** [Objective] With the rise of new computing technologies including cloud computing, big data, and artificial intelligence, hybrid computing based on multiple computing power has been promoted. Under the guidance of the national strategy of “East-West Computing Requirement Transfer”, it is important to develop the programming paradigm for the new generation of computing infrastructure to take full advantages of the HPC computing power and provide a new HPC computing power programming model. [Methods] Firstly, the current computing service mode of the high-performance computing environment and the existing HPC programming paradigm are introduced. Secondly, an HPC computing power programming paradigm (HPC as a function) based on the high-performance computing environment is proposed according to the HPC computing power and task abstraction. Then design of the reference architecture of the HPC computing power programming paradigm is given. [Results] The HPC computing power programming paradigm can support the basic needs of scientific research for HPC computing power, and distribute tasks that are suitable for HPC to appropriate computing

基金项目: 中国科学院战略性先导科技专项项目（A类）（XDA19020101）

\*通信作者: 王小宁 (E-mail:wxn@sccas.cn)

resources. Compared to a traditional workflow system, the new paradigm provides greater flexibility and programmability. **[Conclusions]** The HPC computing power programming paradigm is expected to effectively improve the programmability of the new computing infrastructure.

**Keywords:** CNGrid; CNGrid system software; hybrid computing; computing power network; programming paradigm

## 引言

随着计算基础设施能力的不断提升和计算技术的发展, 计算已经成为科学的研究和工程制造的必不可少的工具与手段。

传统高性能计算依赖大规模、高速网络互联的计算资源形成强大的计算力, 通过数值计算技术满足了包括气象预报、灾害预警、材料科学、飞行器制造、宇宙演化等诸多领域中大规模模拟和仿真需求, 成为最早也是最强的计算力代表。传统高性能计算系统追求计算速度快、精度高; 这不仅需要国家投入巨大的建设和运维成本, 还需要发展大规模科学计算软件和工具, 尤其面向国产的异构计算系统仍需投入较高的时间和人力成本。

相比昂贵的传统高性能计算资源, 云计算<sup>[1]</sup>通过虚拟化技术将相对低廉的计算资源转化为看似无限可用的计算资源并按需提供给用户, 通过规模化效应显著降低了成本。早期的云计算平台主要提供基本的云主机和云存储服务, 用于搭建各类信息系统和服务, 随后以容器、微服务、DevOps 等技术为基础的各类云原生技术相继而出; 云超算采用云计算技术支持快速构建并行计算环境, 满足科学的研究和工程模拟的一般计算需求。随着云计算平台带给用户越来越多的代码部署和资源管理负担, 当前云计算正发展至下一个阶段, 即无服务器计算<sup>[2]</sup>。无服务器计算主要面向编程人员, 允许用户专注于业务逻辑编程, 屏蔽繁琐的部署、监控、容错、日志、扩展性等问题; 其中, 编程模式成为其发展的重要挑战。

大数据<sup>[3]</sup>和人工智能<sup>[4]</sup>是近几年新兴的技术。大数据包括数据采集、数据预处理、数据存储及管理、数据分析及挖掘、数据展现和应用几个阶段, 其中数

据分析及挖掘在海量数据的负载下需要较大的算力资源支撑。以深度神经网络为代表的人工智能技术近年处于蓬勃发展时期, 大幅跨越了科学与应用之间的技术鸿沟, 迎来爆发式增长的新高潮, 其中从模型训练到推理无不以大量算力资源为基础, 甚至出现了以单浮点精度为度量的人工智能芯片和集群。

今天, 无论科学问题求解还是工业仿真, 都不再拘泥于单一算力资源或计算技术, 推动了融合计算的出现和发展。本文的融合计算是指应用编程根据业务需要, 结合不同算力的特征, 采用多种计算技术, 分布式运行在多元化算力资源上。

表 1 算力资源比较

Table 1 Comparation of computing system

	高性能计算系统	云计算平台
计算资源	物理计算节点	虚拟化云主机
网络资源	高带宽、低延迟专用网络	普通网络
目标应用	计算密集型 /通信密集型	数据密集型
建设成本	高	一般
管理模式	批处理任务管理/ 相对封闭	自主管理虚拟资源/ 开放

表 2 计算技术比较

Table 2 Comparation of computing technology

	数值计算	数据处理	人工智能
计算特点	计算密集型 (双精度浮点计算) /通	数据密集型	计算密集型 (单精度浮点计算) / 数据密集型
资源平台	高性能计算系统	云计算平台	云计算平台/高性能计算系统
应用场景	数学计算	海量数据	智能决策

在国家“东数西算”战略<sup>[5]</sup>的指引下, 通过网络建设可将各有优势的算力基础设施资源互联互通, 通过融合计算可实现多种计算技术的协同工作, 充分利用不同算力和计算方法的优势。本文主要研究并提出基于高性能计算环境的新型 HPC 算力编程模式 (HPC as a function), 旨在融合计算中充分发挥 HPC 算力优势, 是新一代计算基础设施可编程能力的重要变革。

## 1 高性能计算环境服务模式发展

高性能计算环境<sup>[6]</sup>聚合了跨地域、跨组织的高性能计算资源, 通过网格技术和云服务技术等为用户提供统一的高性能计算服务环境。中国科学院计算机网络信息中心自主研发了高性能计算环境系统软件 SCE<sup>[7-9]</sup>, 采用微服务架构, 通过驱动机制有效屏蔽资源层在管理和组织方面的差异, 支持集成不同的资源调度算法, 为用户按需匹配计算资源, 并提供命令行、WEB、接口等多种使用方式。

目前 SCE 系统软件已应用于国家高性能计算环境 (暨中国国家网格)<sup>[10]</sup>的建设, 建成了包括天津、无锡、广州、深圳、郑州、昆山等多家国家超级计算中心在内的大型跨域分布式计算环境。同时, SCE 软件也应用于中国科学院超级计算环境的建设, 目前由中国科技云<sup>[11]</sup>统一管理和运行。

高性能计算环境从服务模式发展来看大致以下几个阶段:

第一阶段, 应用服务阶段。

相比单一集群提供的 SSH 命令行使用模式, 高性能计算环境为了降低传统高性能计算系统的使用门槛, 发展了以应用为中心的服务模式。通过系统软件屏蔽了不同高性能计算系统中作业管理系统的差异, 应用软件部署差异、环境配置差异等, 允许用户仅提交计算数据, 指定计算规模即可开展计算。服务形式为命令行和 WEB 界面两种方式。

第二阶段, 接口服务阶段。

十二五至十三五期间, 为支撑发展应用领域社

区和业务平台的建设, 高性能计算环境通过多源认证与授权、接口网关、服务治理等技术提供了环境计算服务接口<sup>[12]</sup>。环境计算服务接口是 REST 风格的网络编程接口, 支持跨平台、跨语言的编程方式, 可为应用领域社区和业务平台的搭建提供较大的灵活性。目前, 计算服务接口已支持建设了二十多个应用领域社区和业务平台, 包括高能物理软件系统, HPC 教育实践平台等。

第三阶段, 计算函数阶段。

随着融合计算的出现, 高性能计算仍然是应用业务的重要数据处理手段。高性能计算环境即将进入新的计算服务阶段, 即计算函数阶段。该阶段提供新型 HPC 算力编程模式 (HPC as a function), 旨在应用编程层面对 HPC 算力资源和任务进行抽象, 并形成易用的编程函数库。计算函数立足于应用业务编程人员, 使得用户不再受限于应用领域或业务平台固有的业务工作流, 不受限于 HPC 算力资源, 支持融合计算业务编程, 提升“东数西算”多元计算基础设施的可编程性。

## 2 相关工作

传统 HPC 算力编程模式是基于单一高性能计算系统的并行编程模式。早期代表性的编程模式是 OpenMP<sup>[13]</sup> 和 MPI<sup>[14]</sup>, 分别支持多核并行编程和跨节点并行编程; 随着异构计算系统成为主流, 多种异构编程模式出现, OpenCL<sup>[15]</sup> 是第一个面向异构系统通用目的并行编程的开放式、免费标准; CUDA<sup>[16]</sup> 是建立在 NVIDIA GPU 上的一个通用并行计算平台和编程模型; 可移植异构计算接口 HIP<sup>[17]</sup> 是一种 C++ 运行时 API 和内核语言, 适用于 AMD 和 NVIDIA GPU; SYCL<sup>[18]</sup> 是 OpenCL 的高级编程模型, 用于提高编程效率; OneAPI<sup>[19]</sup> 是 Intel 推出的统一的、简化的编程模型, 旨在简化跨多架构的开发过程 (如 CPU、GPU、FPGA、加速器)。

与传统 HPC 算力编程模式不同, 本文立足多个高性能计算系统, 并假设单一高性能计算系统内部

应用计算已形成计算程序, 重点研究其在融合计算应用中的集成。

在美国能源部 E 级计算计划 (Exascale Computing Project, 简称 ECP)<sup>[20]</sup> 中, 洛斯·阿拉莫斯国家实验室联合多个大学实验室构建了一个 HPC 应用统一运行时框架 BEE<sup>[21]</sup>, 分别针对 Charliecloud 容器平台、虚拟机平台、AWS 云平台、OpenStack 云平台提供了 HPC 应用运行时的后端实现, 并提供了易于使用的统一用户界面、执行环境和优化性能。BEE 可支持 HPC 应用在高性能计算系统和云计算平台无差异化运行, 并支持上层工作流构建。

美国东北大学和麻省理工学院林肯实验室联合针对科学计算工作流提出了一种混合执行策略 Mashup<sup>[22]</sup>, 用来自平衡计算任务在常规云计算平台 (EC2) 或在新型无服务器计算环境 (Lambda) 的执行。实验表明 Mashup 在执行时间和成本降低方面均优于最新工作流执行引擎。

加州大学伯克利分校和纽约大学针对无服务器计算提出了一种新型编程框架 Kappa<sup>[23]</sup>。Kappa 可以支持程序员在新型无服务器计算范式下表达一组短期任务, 使用检查点机制处理函数超时, 并提供并发机制支持并行计算和协同。

上述工作中 Mashup 和 Kappa 均为面向云计算基础设施的编程支持, 其中计算任务的表达和编排等技术对本文工作有借鉴意义。BEE 支持 HPC 算力, 但仅提供了工作流表达, 应用可编程性存在一定局限性。本文重点研究 HPC 算力在融合计算应用中的新型编程模式, 相比跨域工作流系统更侧重于 HPC 算力在应用编程层面的抽象与表达, 可与其他类型算力接口协同提供更强的可编程性。

### 3 HPC 算力编程模式

#### 3.1 HPC 算力抽象

本文定义了 HPC 算力的如下基本抽象:

队列: 表示计算资源, 具备的属性包括所属集

群名称、总核数、当前运行核数、当前排队核数。队列为计算资源的基本单位。

应用: 表示应用资源, 具备的属性包括所属集群、版本、可用队列、执行环境设置信息。一个计算任务可执行的基本条件是计算资源上部署有应用资源。

作业: 表示计算任务, 具备的属性包括应用、计算核数、输入参数和数据、输出数据、状态。一个作业的状态包含排队、运行、完成三个值。

数据: 为计算任务的输入和输出, 具备的属性为作业、存储位置。数据在融合计算编程中将是 HPC 算力与业务其他流程连接的重要枢纽。

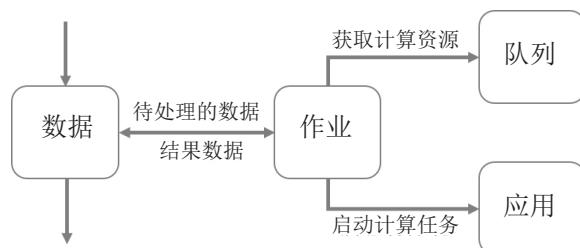


图 1 HPC 算力抽象

Fig.1 Abstraction in HPCaaF

HPC 算力抽象的关系如图 1 所示, 数据是 HPC 算力服务的输入和输出, 数据计算通过作业执行, 作业向队列申请并获取计算资源, 必要时进入排队状态, 当资源满足时调用应用启动计算任务, 待计算完毕通过数据进入应用业务的下一个流程。对于整体应用业务编程, 此过程为异步执行。

#### 3.2 计算函数库

基于高性能计算环境的 HPC 算力编程模式以计算函数库为具体实现形式, 采用 Python 编程语言。

Python 已经成为大数据处理和人工智能应用的主流编程语言, 提供基于 Python 编程语言的计算函数库更有利融合计算中 HPC 算力资源表达; 同时, Python 程序可以通过 Jupyter<sup>[24-25]</sup> 笔记形式进行共享和重现, 便于应用业务的协同。

计算函数库的主要函数定义如表 3。

表 3 主要计算函数定义

Table 3 Function definition

函数名	GetQueue
基本功能	获取环境队列资源列表
可选参数	集群名、应用名、应用版本、并行规模、预估时长
返回值	符合条件的队列结构列表, 字段包含队列名、所在集群、计算运行限制、可用状态、运行和排队作业数
函数名	SubmitJob
基本功能	提交计算任务请求
必填参数	应用名
可选参数	应用版本、计算参数、目标集群、目标队列、并行规模、数据文件
返回值	作业号
函数名	GetJobInfo
基本功能	查询作业基本信息
可选参数	作业号、集群名、队列名、应用名、查询时间区间、状态
返回值	符合条件的作业列表, 字段包括作业号、提交时间、状态更新时间、当前状态、提交队列、应用名等
函数名	Wait
基本功能	等待作业执行结束
可选参数	作业号列表
返回值	无
函数名	GetJobFile
基本功能	从高性能计算环境下载作业文件
必填参数	作业号、文件名列表、存储路径
返回值	无

### 3.3 关键技术问题

在 HPC 算力编程模式实现中, 待解决的关键技术问题如下:

#### (1) 用户认证与授权

在高性能计算环境中, 用户需要通过环境认证和授权方可使用环境中高性能计算资源, 以作业日志作为资源核算的重要依据。

高性能计算环境采用多源用户认证与授权技术<sup>[26]</sup>,

不仅支持环境用户通过命令行、WEB 接口、领域社区和业务平台认证访问环境计算资源, 而且基于 OAuth 和 Token 技术支持第三方领域社区和业务平台的用户访问环境计算资源。

不同于平台服务的多用户运行模式, 计算程序往往是单用户运行模式; 而多源用户认证与授权技术需要采用回调机制确保认证安全, 因此融合计算程序难以凭借用户名和密码在程序中完成高性能计算环境的认证过程, 需要新的技术方案在兼容已有认证体系同时支撑计算函数库安全访问 HPC 算力资源。

当前可借鉴的技术方案为 GITHUB 提供的基于 Token 的登录方式<sup>[27]</sup>。用户可通过 GITHUB 网站根据需求设置参数并生成 Token, 然后在命令行等运行环境中通过 Token 访问 GITHUB 的各项接口功能。

#### (2) HPC 算力评估

HPC 算力评估是融合计算应用中判定一个计算任务是否需要 HPC 算力支撑的重要决策依据, 是实现融合计算应用高效率、低成本目标的关键。

国家高性能计算环境建设团队在十三五项目执行期间提出高性能计算环境资源评价标准<sup>[28]</sup>, 立足于单一计算集群量化评价集群计算服务水平并形成规范, 但缺乏针对单个算例的分析评测方法。论文相关工作 Mashup 面向云计算环境提出了计算任务执行的量化定义, 作为选取最优目标计算资源的重要依据, 可供本文后续进一步工作借鉴。

当前实际应用业务中, HPC 算力评估尚主要依靠应用业务人员的经验, 一般从资源和成本两个层面考虑。资源层面, 当计算所需的硬件资源(如加速卡)或应用软件资源仅在 HPC 算力中具备时, 优先考虑使用 HPC 算力资源; 成本层面, 主要包括时间成本和花销成本, 其中时间成本主要考虑计算加速比和排队时间两个重要因素。

#### (3) 计算任务编排

应用业务流程中往往包含多个 HPC 计算任务, 他们或可独立运行, 或存在数据依赖, 通过计算任

务编排可优化融合计算应用业务中的多个 HPC 计算任务执行。

文献 [29] 提出的 Brane 框架使用直观的领域特定语言 DSL 表达任务编排逻辑, 帮助具有有限编程经验的终端用户自行编写应用程序, 实现跨计算设施的任务编排和应用构建。文献 [30] 提出的 BEE 编排系统通过调度组件和与不同计算资源的交互组件实现云和 HPC 组件任意组合的通信, 从而支持跨资源的复杂工作流实现。

HPC 算力编程模型下的计算任务编排可借鉴已有科学工作流中的任务编排技术, 并重点解决一个核心问题, 即以应用为单位, 考虑 HPC 集群排队的实际情况, 如何编排其中多个 HPC 计算任务, 使得整体计算完成时间最小。

## 4 HPC 算力编程模式参考体系结构

HPC 算力编程模式参考体系结构如图 2 所示。

参考体系结构自底向上分为四层:

物理层: 物理层是由支撑 HPC 算力的高性能计算资源和网络构成。高性能计算资源包括通用计算系统和国产异构计算系统。目前高性能计算资源之间通过互联网相连, “东数西算”高速专网的建成有望大大提升整体 HPC 算力效率。

连接层: 在物理网络之上建立虚拟叠加网络, 通过智能路由在现有网络条件下优化数据传输, 并提供 SSL 数据加密传输等安全举措。

服务层: 采用微服务架构, 通过资源驱动、分布式消息总线等技术聚合跨地域、跨组织、不同体系结构的高性能计算资源, 形成统一的资源信息存储; 并通过授权管理、数据传输、智能调度、服务监控等技术实现计算任务和数据的智能分发和协同工作, 提供统一的高性能计算服务。

表达层: 通过 REST 风格编程接口、接口网关服务面向应用提供网络服务接口, 提供用户认证与授权、接口安全检测机制等保障网络服务接口的安全。基于应用编程接口提供计算函数库, 支持应用业务编程与高性能计算环境交互完成 HPC 计算任务和数据的管理。

## 5 原型系统应用与测试

本文依托国家高性能计算环境初步实现了原型系统, 并应用于风暴潮实时预测工作。风暴潮是沿海近岸的一种巨大的海洋危害, 风暴潮实时预报系统<sup>[31]</sup>能够对风暴潮进行快速准确的实时预报, 可以在一定程度上降低风暴潮带来的经济损失和人员伤亡。其预报流程如图 3。首先抓取特定区域的实时观

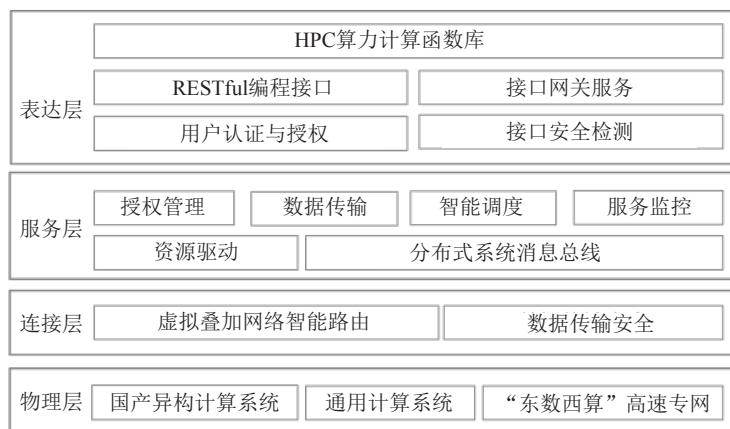


图 2 HPC 算力编程模式参考体系结构

Fig.2 HPC as a function reference architecture

测数据; 然后以实时观测数据为驱动, 计算区域内风场、浪场、水位高度等变化情况; 将计算数据进行格式转化, 并通过多种方式对实时预报数据进行可视化表达。

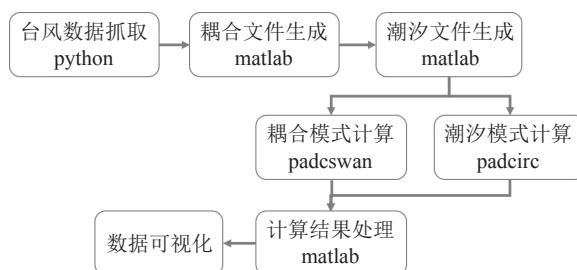


图 3 风暴潮预测基本流程  
Fig.3 Storm surge forecasting workflow

其中, 台风数据抓取和数据可视化由于联网需求运行于云计算平台, 而数据生成和处理的计算过程可根据算力评估确定。

本文测试基于中国科学院先导专项“地球大数据科学工程”建设的专用计算基础设施<sup>[32]</sup>提供的云计算平台资源, 以及国家高性能计算环境提供的 HPC 算力资源。同时, 该专用计算基础设施提供的高性能计算资源已集成于国家高性能计算环境, 可通过共享存储与云计算平台共享应用数据。

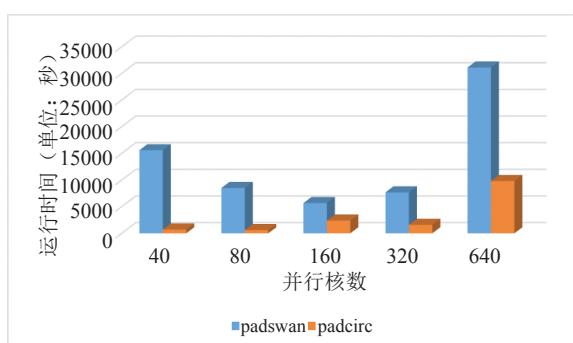


图 4 核心模式计算并行测试  
Fig.4 Kernel modules parallel testing

由图 4 可见, 潮汐模式计算 40 核并行最优; 耦合模式计算 160 核并行最优, 因此采用 HPC 算力; 耦合文件生成、潮汐文件生成和计算结果处理时间

因商业软件 MATLAB 的需求在本测试中考虑使用 HPC 算力。

应用核心计算运行时间 6,300 秒 (1 小时 45 分钟), 各阶段运行时间如图 5 所示。假设云主机具备 HPC 相同的计算能力, 按单节点 40 核并行估算, 预计计算在云主机中完成需要 16,211 秒 (4 小时 30 分 11 秒)。相比估算的云主机运行时间, 混合算力可缩短计算时间约 61%。

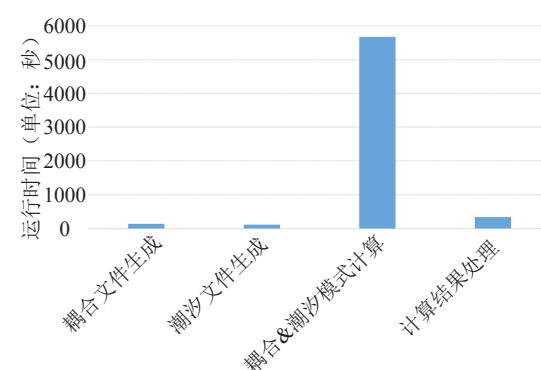


图 5 风暴潮预测各阶段运行时间  
Fig.5 Storm surge forecasting testing

## 6 结论与展望

在国家“东数西算”战略背景下, 论文提出了一种新型 HPC 算力编程模式, 旨在充分发挥 HPC 算力优势, 提高“东数西算”算力基础设施的可编程性。论文实现了原型系统, 并初步应用于风暴潮实时预测工作。

未来工作将基于 HPC 算力编程模式原型系统, 进一步解决用户认证与授权、HPC 算力评估、计算任务编排等关键问题, 并通过实际应用于科研业务流程开展优化工作。

## 利益冲突声明

所有作者声明不存在利益冲突关系。

## 参考文献

- [1] ARMBRUST, MICHAEL, FOX, et al. Above the Clouds: A Berkeley View of Cloud Computing[J/OL]. Science, 2009[2022-07-21]. [http://www.researchgate.net/publication/277295384\\_Above\\_the\\_Clouds\\_A\\_Berkeley\\_View\\_of\\_Cloud\\_Computing](http://www.researchgate.net/publication/277295384_Above_the_Clouds_A_Berkeley_View_of_Cloud_Computing). DOI:doi: 10.1145/ 1721654. 1721672.
- [2] JONAS E, SCHLEIER-SMITH J, SREEKANTI V, et al. Cloud Programming Simplified: A Berkeley View on Serverless Computing[J/OL]. arXiv, 2019[2022-07-21]. <http://arxiv.org/abs/1902.03383>.
- [3] 程学旗, 靳小龙, 王元卓, 等. 大数据系统和分析技术综述[J]. 软件学报, 2014, 25(9): 1889-1908.
- [4] 钟义信. 人工智能:概念·方法·机遇[J]. 科学通报, 2017, 62(22): 2473-2479.
- [5] 王建冬, 于施洋, 窦悦. 东数西算: 我国数据跨域流通的总体框架和实施路径研究[J]. 电子政务, 2020(03): 13-21.
- [6] XU Z, CHI X, XIAO N. High-performance computing environment: a review of twenty years of experiments in China[J]. National Science Review, 2016, 3(1): 36-48.
- [7] XIAO H, WU H, CHI X. SCE: Grid Environment for Scientific Computing[M/OL]//VICAT-BLANC PRIMET P, KUDOH T, MAMBRETTI J. Networks for Grid Applications: volume 2. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 35-42.[2022-07-21].[http://link.springer.com/10.1007/978-3-642-02080-3\\_4](http://link.springer.com/10.1007/978-3-642-02080-3_4).
- [8] 王小宁, 肖海力, 曹荣强. 面向高性能计算环境的作业优化调度模型的设计与实现[J]. 计算机工程与科学, 2017, 39(04): 619-626.
- [9] 吴璨, 王小宁, 肖海力, 等. 基于消息总线的高性能计算环境系统软件优化设计与实现[J]. 高技术通讯, 2020, 30(03): 248-258.
- [10] 中国国家网格[EB/OL]. <http://www.cngrid.org>.
- [11] 中国科技云[EB/OL]. <http://www.cstcloud.cn>.
- [12] CAO R, XIAO H, SHASHA L, 等. SCEAPI: A unified Restful Web API for High-Performance Computing[J/OL]. Journal of Physics: Conference Series, 2017, 898: 092022. DOI:10.1088/1742-6596/898/9/092022.
- [13] OpenMP[EB/OL]. <https://www.openmp.org/>.
- [14] MPI - The Complete Reference: Volume 2, The MPI Extensions | Books Gateway | MIT Press[EB/OL]. <https://direct.mit.edu/books/book/2091/MPI-The-Complete-ReferenceVolume-2-The-MPI>.
- [15] OpenCL[EB/OL]. <https://www.khronos.org/opencl/>.
- [16] CUDA[EB/OL].<https://www.nvidia.cn/geforce/tech-nologies/cuda/>.
- [17] HIP[EB/OL].<https://github.com/ROCM-Developer-Tools/HIP>.
- [18] SYCL[EB/OL].<https://www.khronos.org/sycl/>.
- [19] OneAPI[EB/OL].<https://www.oneapi.io/>.
- [20] ECP项目[EB/OL].<https://www.exascaleproject.org>.
- [21] CHEN J, GUAN Q, LIANG X, 等. Build and Execution Environment (BEE): an Encapsulated Environment Enabling HPC Applications Running Everywhere [C]// 2018 IEEE International Conference on Big Data (Big Data), 2018: 1737-1746.
- [22] ROY R B, PATEL T, GADEPALLY V, et al. Mashup: making serverless computing useful for HPC workflows via hybrid execution[C/OL]//Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. Seoul Republic of Korea: ACM, 2022: 46-60.[2022-07-21]. <https://dl.acm.org/doi/10.1145/3503221.3508407>.
- [23] ZHANG W, FANG V, PANDA A, et al. Kappa: a programming framework for serverless computing[C/OL]// Proceedings of the 11th ACM Symposium on Cloud Computing. Virtual Event USA: ACM, 2020: 328-343.[2022-07-21]. <https://dl.acm.org/doi/10.1145/3419111.3421277>.
- [24] KLUYVER T, RAGAN-KELLEY B, PÉREZ F, et al.

- Jupyter Notebooks - a publishing format for reproducible computational workflows[C/OL]//ELPUB, 2016. DOI: 10.3233/978-1-61499-649-1-87.
- [25] BEG M, TAKA J, KLUYVER T, et al. Using Jupyter for Reproducible Scientific Workflows[J]. Computing in Science & Engineering, 2021, 23(2): 36-46.
- [26] 和荣, 肖海力, 王小宁, 等. 高性能计算环境多源用户认证方法研究与实现[J/OL]. 计算机科学与探索, 2022: 1. DOI:10.3778/j.issn.1673-9418.2202037.
- [27] GITHUB 文档[EB/OL]. <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>.
- [28] 白杨, 杜云飞, 王多强, 等. 国家高性能计算环境资源评价规范研究[J]. 科研信息化技术与应用, 2018, 9(02): 79-89.
- [29] VALKERING O, CUSHING R, BELLOUM A. Brane: A Framework for Programmable Orchestration of Multi-Site Applications[C]//2021 IEEE 17th International Conference on eScience (eScience), 2021: 277-282.
- [30] TRONGE J, GRUBEL P, RANDLES T, et al. BEE Orchestrator: Running Complex Scientific Workflows on Multiple Systems[C]//2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC), 2021: 376-381.
- [31] FENG X, LI M, LI Y, et al. Typhoon storm surge in the southeast Chinese mainland modulated by ENSO[J]. Scientific Reports, 2021, 11(1): 10137.
- [32] 地球大数据云服务门户[EB/OL] <http://portal.casearth.cn>.

收稿日期: 2022 年 7 月 29 日

**王小宁**: 中国科学院计算机网络信息中心, 副研究员, 主要研究方向为网格计算、高性能计算、分布式计算与云服务。



本文中主要承担工作为模型提炼和论文撰写。

WANG Xiaoning is an associate professor at the Computer Network Information Center, Chinese Academy of Sciences. Her research interests include grid computing, high-performance computing, and cloud service.

In this paper, she is responsible for HPC programming modeling and paper writing.

E-mail: wxn@sccas.cn

王小宁,卢莎莎,吴璨,和荣,闫晓婷,肖海力,迟学斌. 基于高性能计算环境的HPC算力编程模式[J]. 数据与计算发展前沿, 2022, 4(5):33-41. CSTR: 32002.14.jfdc.CN10-1649/TP.2022.05.004.DOI:10.11871/jfdc.issn.2096-742X.2022.05.004.

WANG Xiaoning, LU Shasha, WU Can, HE Rong, YAN Xiaoting, XIAO Haili, CHI Xuebin. HPC Computing Power Programming Paradigm Based on CNGrid [J]. *Frontiers of Data & Computing*, 2022, 4(5):33-41. CSTR: 32002.14.jfdc.CN10-1649/TP.2022.05.004.DOI:10.11871/jfdc.issn.2096-742X.2022.05.004.