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

应用感知的算力优化调度方法

寇大治[1*]，韦建文[2]，唐小勇[3]

1.上海超级计算中心，上海 201203
2.上海交通大学，高性能计算中心，上海 200240
3. 长沙理工大学，计算机与通信工程学院，湖南 长沙 410114

ISSN 2096-742X
CN 10-1649/TP

文献CSTR:
32002.14.jfdc.
CN10-1649/
TP.2022.05.001

文献DOI:
10.11871/jfdc.issn.
2096-742X.2022.
05.001

页码：3-10

获取全文

摘　要：【目的】在“东数西算”工程的大背景下，为了更好地实现对分布在不同地域超级计算机资源的调度管理，针对计算资源忙闲不均等问题，提出通过研究典型应用作业的运行特征，开发多中心任务的调度系统，以解决国家高性能计算环境统一调度的关键技术问题。【方法】首先收集了若干超级计算中心的应用运行历史情况，建立了应用运行历史数据库；其次将用户应用对资源的需求和典型应用的资源使用特征分析相结合，通过机器学习的方法，建立了一种可精确描述应用特征的框架；然后实现了跨集群高性能计算应用的容器方式迁移；最后研究了基于多中心应用特征的任务调度方法，开发了基于应用感知的全局资源优化调度系统。【结果】该系统为国家高性能计算环境服务化运营和稳定运行提供了有力的技术支撑。【结论】基于应用感知的算力优化调度方法可望有效提高“东数西算”的可靠性、可用性和可维护性。

关键词：高性能计算系统；历史数据库；应用特征；算力调度方法

Application-Aware Method for Optimized Computing Power Scheduling

KOU Dazhi[1*], WEI Jianwen[2], TANG Xiaoyong[3]

1. 1.Shanghai Supercomputer Center, Shanghai 201203, China
2.Center for High Performance Computing, Shanghai Jiao Tong University, Shanghai 200240, China
School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, Hunan 410114, China

Abstract: [Objective] Under the background of the project of “East-West Computing Requirement Transfer”, the super-computing resources distributed in different regions will be scheduled and managed. In order to avoid the problem of busy and unevenly distribution of computing resources, it is necessary to develop a multi-center task scheduling system by investigating the runtime characteristics of typical applications to achieve unified management of the national high-performance computing environment. **[Methods]** Firstly, the log data about application execution at several national supercomputing centers are collected and the database for the application log data is established. Secondly, by taking the user resources demand and the resource usage characteristics of typical applications into consideration, a machine learning framework is established to accurately depict the application execution features. Then migration of HPC applications across clusters using containers is implemented. Finally, a task

基金项目：国家重点研发计划“基于应用的优化调度方法与实现”（2018YFB0204004）
*通信作者：寇大治（E-mail: dzkou@ssc.net.cn）

scheduling system based on application-aware resource scheduling optimization is developed. **[Results]** This system provides powerful technical support for services and efficient operation of the national high-performance computing environment. **[Conclusions]** The application-aware method for computing power scheduling optimization is expected to effectively improve the reliability, availability, and maintainability of the "East-West Computing Requirement Transfer" project.

Keywords: High Performance Computing system; historical database; application feature; computing power scheduling method

引言

近年来，随着我国高性能计算应用的发展，应用领域和计算需求逐步增加，由应用领域的计算需求产生的无效调度问题日益显著，如计算故障、计算资源排队、计算过程的波动、计算资源的预留与回填不畅等一系列问题，已经影响到了用户作业的正常运行。随着用户对高性能计算应用服务水平的要求越来越高，从应用的角度出发，基于应用的调度方法还有优化提升的空间。例如：准确可靠的作业运行时间预测技术不仅能够为作业调度技术提供保障，为调度模型的优化提供基础，还将为用户提供更可靠的作业提交信息。这些优化都能很大程度上提高国家高性能计算环境的稳定性，从而有效增加用户的满意度以及国家高性能计算环境对用户的吸引力，缓解高性能计算需求的压力[1-5]。

高性能计算应用往往需要大规模和长时间的计算才能完成，总体来看应用软件总是基于特定的力场、泛函或理论，以及这些力场、泛函或理论的排列组合再经过前后处理开发完成的。计算的体系各有不同，计算的规模有大有小，当这些计算体系和规模确定之后，通过特定力场、泛函或理论的计算及过程就有可能可控并可预测。为了对国家高性能计算环境的建设起到支撑，本文将基于对应用的数据及参数分析，探索建立作业运行时间与参数及数据相关的优化调度模型。模型的建立将围绕应用的实际运行情况进行分析，提取应用作业的特征或参数并辅以并行计算所需的系统指标或参数，综合各方面的信息建立优化调度模型。同时本文还将选取典型应用例进行研究，依照本文建立起来的优化调度模型，提取应用计算体系特征并做评估和测算。在应用的基础上建立基于异构的超大规模计算资源协同调度原型系统，该原型系统将针对多中心的高性能超级计算机系统、中小型服务器设备、集群以及普通计算节点的计算能力、存储和网络通信能力进行建模，针对应用特征，研发和实现基于大规模异构计算资源的任务调度算法。对接入的异构计算资源进行统一调度和管理，实现面向计算任务需求的资源按需分配。相关工作以期对全局资源优化调度方法产生指导作用。

1　超算历史作业信息数据库

本文建立了超算系统应用作业历史数据库，收集了上海超级计算中心、国家超算无锡中心、甘肃省计算中心、上海交通大学高性能计算中心和中国科学技术大学超级计算中心的应用作业历史数据。根据"并行作业负载资料库（Parallel Workloads Archive, PWA）"整理为 SWF 格式，形成的历史数据库将其命名为"中国超算任务负载资料库"（Chinese Supercomputers Workloads Archive, CSWA），经过数据脱敏之后公开发布在 https://git. lug.ustc.edu.cn/yshen/CSWA 上。

其中 PWA 所使用的 SWF 为开放的任务记录格式，其中将可能涉及用户隐私的敏感信息用数字代替，保护了用户的信息安全。该资料库积累了从 1993 年以来 38 个系统的数据，为业界广泛使用，近年来相关文献每年有 2,000 余篇，在超级计算任务调度等研究工作中起到了重要的作用。但该资料库中数据较老，最新数据为 2015 年捷克 MetaCentrum 系统的数据，且其中没有来自中国的超算系统的数据，故以本文工作整理的数据为基础，建立了我国自己的超算任务负载资料库，这项工作将会为超算研究提供更新更全的基础数据。

目前超算运行数据资料库收集有 667 万条数据，其中上海超级计算中心为魔方 II 超级计算系统上 2017 到 2019 年度的 149.2 万条历史运行数据；国家超算无锡中心为神威太湖之光高效能计算系统和商用辅助计算系统上 2017 和 2020 年度的 58.1 万条历史运行数据；甘肃省计算中心为曙光 5000 高性能计算集群系统上 2019 到 2020 年度的 2.3 万条历史运行数据；上海交通大学高性能计算中心为高性能计算集群型 π 异构高性能计算系统上 2017 到 2019 年度的 190.5 万条历史运行数据；中国科学技术大学超级计算中心为曙光 TC4600 百万亿次超级计算系统上 2014 到 2021 年度的 267 万条历史运行数据。

2　典型应用的运行时间预测

针对高斯是化学计算软件的特点，读取输入文件，提取特征，建立输入参数与运行时间之间的模型，在预测中引入新的特征描述分子体系——库伦矩阵，提供了对分子体系坐标原点无关、编号起始原子无关、旋转无关的描述方法，能够一定程度提升预测准确度 [6-7]。库伦矩阵是一个 n 阶对称方阵 (n 为分子体系中的原子个数)，使用原子两两之间的关系来刻画整个分子结构，具有与原点位置无关、旋转不变等优良特性。考虑到库伦矩阵的维度不一致，我们将其 f 范数抽象出来作为一个特征。矩阵元素定义如下：

$$M_{ij}^{Coulomb}=\begin{cases}0.5Z_i^{2.4} & ,i=j\\ \frac{Z_iZ_j}{|R_i-R_j|} & ,i\neq j\end{cases} \quad (1)$$

由于高斯作业的运行时间差异，模型预测结果与真实值之间的绝对误差均值不能很好地反映模型预测的准确性，我们使用平均相对误差率 P_r 来评价模型。设 T_{true} 为作业真实运行时间，T_{pred} 为模型预测时间，取一个很小的值 ϵ，P_r 定义如下：

$$P_r=1-\frac{|T_{true}-T_{pred}|}{\max(T_{true},T_{pred})+\epsilon} \quad (2)$$

采用深度人工神经网络（DNN）和梯度提升（XGBoot）两种机器学习算法，对模型进行训练，进行运行时间的预测。在运行时间预测方面，我们的模型对短作业（一小时以内）的预测精度有着非常好的效果。图 1 给出了 DNN 模型在测试集上的预测结果与真实值比对，蓝色部分为真实值，黄色部分为预测值，整体契合度较高，误差集中在小部分样本上。图 2 给出了 XGBOOST 模型预测得到的结果与真实值结果对比，与 DNN 得到的结果相似，整体比较契合。图 3 给出了采用库伦矩阵的作用。

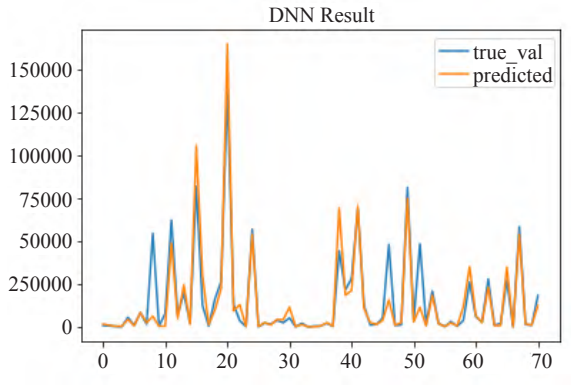

图 1 DNN 模型在测试集上的预测结果与真实值比对

Fig.1 DNN predicted values compared with true values

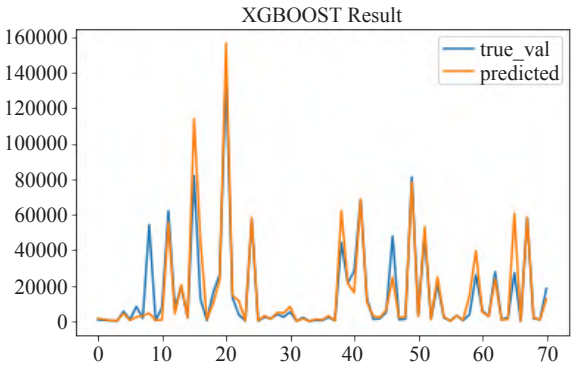

图 2 XGBOOST 模型预测得到的结果与真实值结果对比

Fig.2 XGBOOST predicted values compared with true values

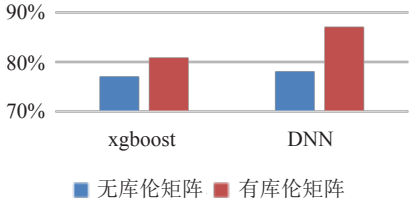

图 3 库伦矩阵在预测中的作用

Fig.3 Coulomb matrix in prediction

通过机器学习的方法，我们对 Gaussian 作业运行时间的预测有比较准确的效果，其中 DNN 的准确率达到了 85%，XGBOOST 的准确率达到了 83%。通过迁移模型的方法，我们的模型在上海超级计算中心平台上同样取得了较好的效果。在输入文件提取方面，我们实现了用户层面无感知的读取工作和用户层面无感知的预测流程，即用户只需提供 Gaussian 作业的输入文件便可以得到预估的作业完成时间。该功能已经集成到国家高性能计算环境中实际运行使用，图 4 是该功能在实际应用中的界面展示，作业提交到 cngrid 上的界面和提交后的界面，在提交之后作业列表中可以看到预测时间。

图 4 国家高性能计算环境中的预测功能界面

Fig.4 Prediction in national high performance computing environment

3 多中心间任务迁移机制的研究

Dockers 可运行于 Linux 和 Windows 环境，用于创建、管理和编排容器。从 Dockers 1.12.0 版本开始，Docker swarm 已经包含在 Docker Engine 中，并且内置了相关服务工具。在实验环境的两台节点上，选择其中一台节点作为 Leader，另外一台作为 worker。Docker+swarm 可以简单实现服务的容器部署。要实现 HPC 应用的跨节点运行，支持多节点的 MPI 环境，需要使用 Docker 的覆盖网络 (overlay network)，同时配置 NFS 服务器，启动服务时将 NFS 目录挂载到所有节点上，这样才能保证 MPI 的运行目录一致。拉取官方已有的 openmpi 镜像，根据实验环境修改模板文件，包括了覆盖网络的名称、NFS 服务器地址和目录，需要启动的节点数目等，根据实际情况进行配置，确定服务启动。运行 MPI 程序可以登录任意一个节点运行。虽然可以使用 Docker 来实现 HPC 应用的运行，但由于集群中多用调度管理器来进行资源的分配和管理，采用的是 cgroups 针对每个作业配置资源，而 Docker 镜像的启动是由 Docker daemon 去执行的，这样资源的限制也就失效。而且 Docker daemon 为 root 用户启动，不安全。

Singularity 是一个轻量的容器系统，它可以无缝地和现有的环境结合，为应用提供一个“运行时环境”，相比 Dockers，它没有 Daemon 进程和网络虚拟化技术。相比 Docker 具有独特的优势：（1）更加轻松的环境打包迁移：借助于 singularity 沙盒方式构建镜像，可以做到类似在虚拟机上安装部署应用一样的使用场景。在应用构建时所产生的文件或所依赖的环境都以镜像文件方式存储，不需要单独打包或导入，直接拷贝走镜像即可。（2）可以和现有系统无缝整合：系统用户权限、网络等直接继承宿主机 (host) 配置，并且无需进入到某个镜像后再执行命令，可以直接在外部调用镜像内的执行，类似于本地安装的指令。将 singularity 和集群调度系统 slurm 结合，可以实现通过 slurm 来调度管理容器资源，并实现与全机集群系统的无缝整合，如交大集群中就采用了这种方式，用户可以根据自身的情况选择采用物理机上运行应用，也可以通过 singularity

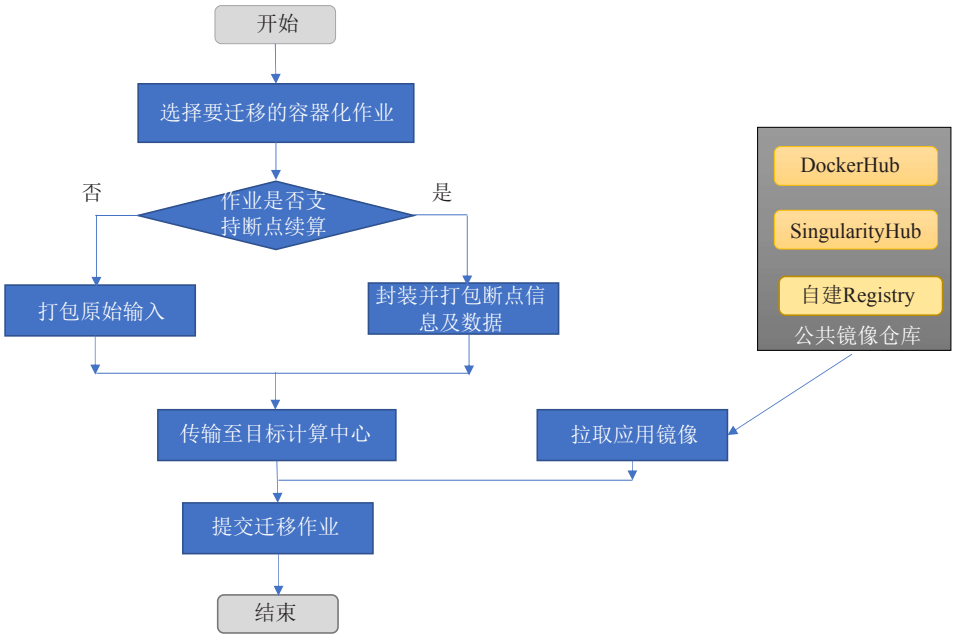

图 5 容器化作业迁移流程

Fig.5 The process of containerized job migration

实现应用的容器化运行，全机整体进行调度。在两台节点上部署 slurm 和 singularity，采用一个 MPI 程序进行测试。自定义 singularity definition file，构建 sif 镜像，可以单机也可以通过 srun 提交脚本来运行。

通过对容器技术开展的研究、测试和部署，实现了跨集群或平台的高性能计算应用的容器方式迁移。容器化作业迁移流程如图 5 所示，利用国家高性能计算环境中中国科学院计算机网络信息中心和上海交通大学已封装完成的镜像，实现在不同超算的容器平台间的应用迁移。其中中国科学院计算机网络信息中心提供了已封装镜像仓库 https://sin.cngrid.org，上面已封装完成包括基础镜像在内的 20 多个镜像，在进行迁移测试时，选择其中的三个镜像 lammps、gromacs 和 namd 分别在上海交通大学 Pi 集群和上海超级计算中心的实验集群两个 singularity 平台进行。

4　应用感知的调度方法

基于国家高性能计算服务环境周期查询计算节点队列资源信息，并将计算资源信息转化为作业调度信息，获取满足作业运行所需资源队列，实现作业到计算节点队列的直接调度与运行，系统周期查询节点资源，只有资源满足作业运行要求时 [8-13]，系统才把作业调度到相应网格节点上，其流程如图 6 所示。

目前，利用超级计算中心的计算资源进行高性能计算研究已经在国内得到了极大的普及。然而，目前大部分超级计算中心针对任务的调度策略都存在一些不可忽略的问题：第一，由于任务调度的不充分性，导致作业的排队时间过长，造成调度效率低下；第二，由于各地对使用超级计算中心的定价不一样，导致需要大规模处理器进行计算的作业需要花费更高的价格完成计算，从而大大增加了成本；第三，由于该调度策略未使用有效的负载均衡策略，任务在可以提供计算的多个队列中不能被高效地调度到空闲队列上进行计算，导致负载较轻的队列处于空闲状态，负载较重的队列处于满负荷状态，从而造成严重的负载不均衡状况，进而形成了较为严

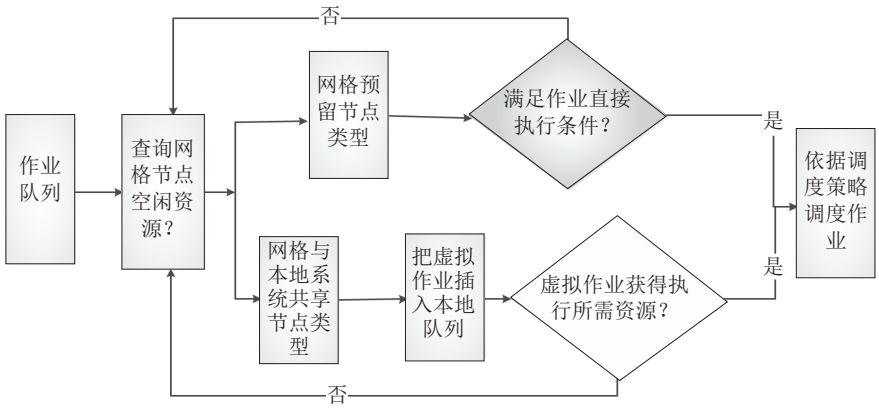

图 6 应用感知调度方法流程

Fig.6 The process of application-aware based scheduling method

重的调度性能瓶颈。针对以上缺陷或改进需求，我们研发了一种用于超级计算中心的并行任务调度方法和系统，其目的在于解决现有超级计算中心所使用的调度策略由于任务调度的不充分性，导致的作业的排队时间过长、调度效率低下的技术问题，以及由于各地对使用超级计算中心的定价不一样，导致需要大规模处理器进行计算的作业需要花费更高的价格完成计算，从而大大增加了成本的技术问题，以及由于未使用有效的负载均衡策略，造成严重的负载不均衡状况，进而形成了较为严重的调度性能瓶颈的技术问题。研究的基本思路在于，采用基于超级计算中心处理器使用价格最低的任务优先进行调度的计算方法做出最终的任务到处理器的映射决策，该计算方法将解析的所有数据分别进行存储，对每一个待调度的作业计算作业到队列上的使用价格，得到的若干个二元组数据进行排序，获取使用价格最小的数据执行优先调度，将它调度到对应的队列上之后周期性地更新资源信息，以确保剩下的作业调度时每一个队列的资源数据的确定性。通过了上述方案的执行，实现了更高的性能和更好的负载均衡效果，降低了开销。

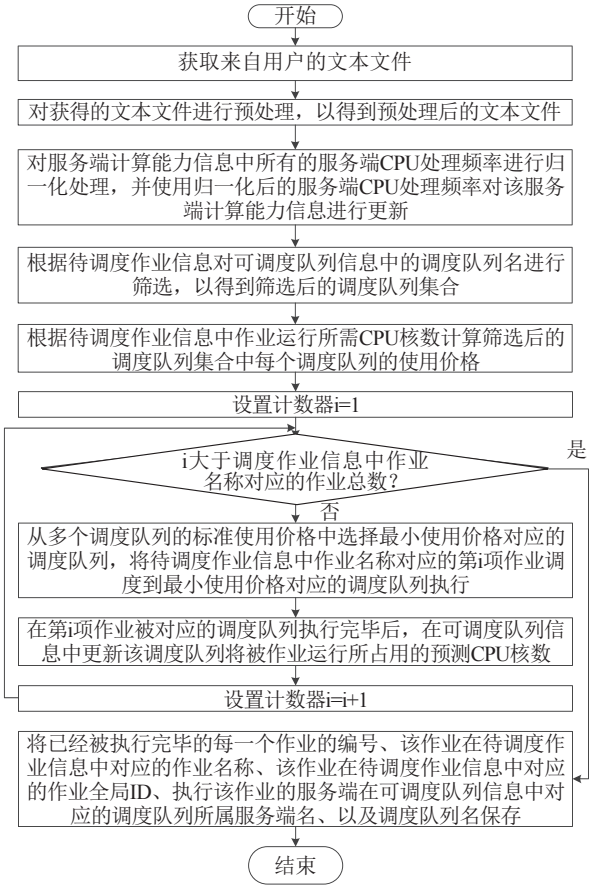

图 7 应用感知调度方法流程

Fig.7 The process of application-aware based scheduling method

研究实现了基于应用的作业调度算法。第一、主要依据环境特征对异构计算资源实现标准化，依据高性能应用的多样性提出作业调度属性特性和作业

可调度约束条件，以实现时间优先的全局资源优化（Application-aware Time First Strategy, ATFS)。第二、针对服务环境计算节点队列负载的非均衡性和时间二维序列性，提出基于长短期记忆神经网络 LSTM 的节点负载预测方法，实现资源利用率优先的网格作业调度策略（Application-aware Workload First Strategy, AWFS)。第三、实现应用感知的计算成本优先作业调度算法（Application-aware Cost First Strategy, ACFS）。建立环境感知的作业直接调度体系，对网格环境中的节点，调度系统周期性查询节点状态，应用调度策略，当节点资源满足作业运行时即直接把作业调度到该节点上。

经过测试，分别在时间优先、资源优先和成本优先三种条件下，分别使用三种调度算法以及与原有的调度算法进行对照，图 8 给出了不同策略调度后的结果与现有调度策略的比较图，可以看到在时间优先考虑时 ATFS 优于原有调度策略 11.03%，优于 AWFS65.87%，优于 ACFS0.38%。

从实验结果看到本文开发的三种调度算法无论在何种外在约束条件下均优于原有的调度算法，且在相对应的约束条件下都表现出了相应的优势。

5 结论与展望

本文收集了国家高性能计算环境中部分超级计算机系统的历史任务数据，研究了典型应用的特征，基于对应用的数据及参数的分析，建立了作业运行时间与参数及数据相关的预测模型，研究了多中心作业调度迁移方法。作为国家高性能计算环境中核心软件层的计算资源调度模块，研究了为国家高性能计算环境提供基于作业类型 - 算法映射优化的自适应调度方法，包括对应用感知负载均衡调度算法的研究。基于本文研究的系统已经开发完成并整合到国家高性能计算环境中，实际测试运行情况良好。

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] LEFF A, RAYFIELD J T, DIAS D M. Service-level agreements and commercial grids [J]. IEEE Internet Computing, 2003, 7(4): 44-50.

[2] 王娟. Backfilling 算法介绍及其在集群中的应用分析

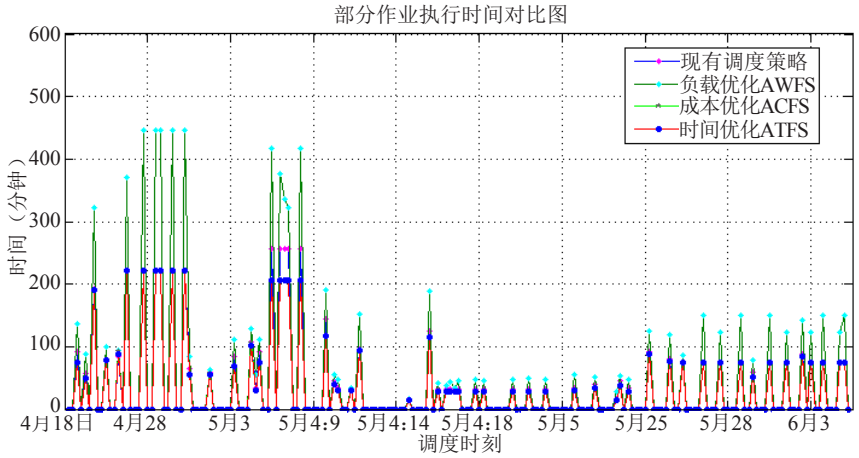

图 8 三种调度策略与现有调度策略的比较图

Fig.8 Comparison of three scheduling strategies and the existing scheduling strategy

[D]. 北京: 北京邮电大学, 2008.

[3] Talby D, Feitelson D G. Supporting priorities and improving utilization of the IBM SP scheduler using slack-based backfilling[C]//Proceedings 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing, IPPS/SPDP 1999. IEEE, 1999: 513-517.

[4] Srinivasan S, Kettimuthu R, Subramani V, et al. Selective reservation strategies for backfill job scheduling[C]// Workshop on Job Scheduling Strategies for Parallel Processing. Springer, Berlin, Heidelberg, 2002: 55-71.

[5] Ward W A, Mahood C L, West J E. Scheduling jobs on parallel systems using a relaxed backfill strategy[C]// Workshop on Job Scheduling Strategies for Parallel Processing, Springer, Berlin, Heidelberg, 2002: 88-102.

[6] LeCun, Y., Bengio, Y., Hinton, G. Deep learning[J]. Nature, 2015, 521(7553):436-444.

[7] Himanen L, Jäger MO, Morooka EV, Canova FF, Ranawat YS, Gao DZ, Rinke P, Foster AS. DScribe: Library of descriptors for machine learning in materials science[J]. Computer Physics Communications, 2020, 247:106949.

[8] Liu C, Li K, Li K. A game approach to multi-servers load balancing with load-dependent server availability consideration[J]. IEEE Transactions on Cloud Computing, 2021, 9 (1) : 1-13.

[9] Liu C, Li K, Tang Z, et al. Bargaining game-based scheduling for performance guarantees in cloud computing [J]. ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS), 2018, 3(1): 1-25.

[10] Bellavista P, Cinque M, Corradi A, et al. GAMESH: A grid architecture for scalable monitoring and enhanced dependable job scheduling[J]. Future Generation Computer Systems, 2017, 71: 192-201.

[11] Hu M, Veeravalli B. Requirement-aware scheduling of bag-of-tasks applications on grids with dynamic resilience[J]. IEEE Transactions on Computers, 2013, 62(10): 2108-2114.

[12] Wyatt M, Herbein S, Ahn D, et al. Unstructured data analytics for next-generation HPC scheduler: capturing jobs' needs from unstructured job scripts[R]. Lawrence Livermore National Laboratory(LLNL), Livermore, CA(United States), 2017.

[13] Cunha R L F, Rodrigues E R, Tizzei L P, et al. Job placement advisor based on turnaround predictions for HPC hybird clouds[J]. Future Generation Computer Systems, 2017, 67:35-46.

收稿日期：2022 年 7 月 11 日

寇大治，上海超级计算中心，高级工程师，主要研究领域为高性能计算集群系统、高性能计算的应用。

本文中负责制定论文框架，撰写第 1 节超算历史作业信息数据库，第 3 节多中心间任务迁移机制的研究，第 5 节结论与展望。

KOU Dazhi, is a senior engineer at the Shanghai Super-computer Center. His research interests include HPC cluster systems and HPC applications.

In this paper, he is responsible for drawing up the paper framework and writing: 1. Application running history database, 3. Container migration of HPC applications, 5. Conc-lusion and prospect.

E-mail: dzkou@ssc.net.cn

寇大治,韦建文,唐小勇. 应用感知的算力优化调度方法[J].数据与计算发展前沿,2022,4(5):3-10. CSTR: 32002.14.jfdc.CN10-1649/TP.2022.05.001.DOI: 10.11871/jfdc.issn.2096-742X.2022.05.001.

KOU Dazhi, WEI Jianwen, TANG Xiaoyong. Application-Aware Method for Optimized Computing Power Scheduling [J]. *Frontiers of Data & Computing*, 2022,4(5):3-10. CSTR: 32002.14.jfdc.CN10-1649/TP.2022.05.001.DOI:10.11871/jfdc.issn.2096-742X.2022.05.001.