

Design and Implementation of Job Migration System (JMS) in Multiple Computing Centers

Mingxia Feng, Weiguo Wu

The School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an Shaanxi
Email: a13709262734@stu.xjtu.edu.cn, wgwu@xjtu.edu.cn

Received: Mar. 9th, 2018; accepted: Mar. 23rd, 2018; published: Mar. 30th, 2018

Abstract

There is a lack of unified job management strategy and job migration mechanism between high performance computing centers. When there is a routine maintenance in a computing center or the user adopts the checkpoint technology to make fault tolerance, jobs need migrating to other computing centers for execution. Aiming at the problem, this paper proposed a JMS system which adopts the design mode of microservice architecture, supports job migration in multiple computing centers and provides the function of multi-center job management and terminal access for users by web portal. Theoretical analysis and test show that JMS is user-friendly and has good scalability.

Keywords

High-Performance Computing, Job Migration, Fault-Tolerant, Micro-Services Architecture

多计算中心作业迁移系统JMS的设计与实现

冯鸣夏, 伍卫国

西安交通大学电子与信息工程学院, 陕西 西安
Email: a13709262734@stu.xjtu.edu.cn, wgwu@xjtu.edu.cn

收稿日期: 2018年3月9日; 录用日期: 2018年3月23日; 发布日期: 2018年3月30日

摘要

各个高性能计算中心之间缺少统一的作业管理策略和作业迁移机制。当遇到计算中心例行维护或者用户采用检查点技术进行异地作业灾备容错时, 需要将作业迁移到其他计算中心执行。针对该问题, 设计实现了多中心作业迁移调度系统JMS。该系统采用了微服务架构的设计模式, 支持多计算中心间的作业迁

移, 为用户提供了多中心作业管理和终端访问的web portal。实际测试表明, JMS具有操作简单, 可扩展性强等特点。

关键词

高性能计算, 作业迁移, 容错, 微服务架构

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

高性能计算是战略性、前沿性的高新技术, 是世界各国争夺的战略制高点, 是国家创新体系的重要组成部分。目前我国已经拥有了由 19 个高性能计算中心组成的中国国家网格[1], 资源能力世界前列。在国家高性能计算环境中, 由于网络带宽、地域和用户选择习惯等因素可能会造成不同计算中心资源忙闲不均。在缺少对多个计算中心统一资源调配统筹策略的情况下, 多中心的负载均衡问题就显得尤为突出。在集群计算中, 负载均衡对于充分地利用集群计算能力和最大化地提升并行作业执行性能起到至关重要的作用。目前高性能计算中心之间由于缺少协同完成作业的能力, 无法实现多中心间的作业迁移容错, 对于那些长时计算的应用定时做作业迁移灾备可以有效减少由于某一个计算中心出现故障带来不必要的计算损失从而可以实现高效、可靠的计算环境。

不同的高性能计算中心采用独立的集群调度系统, 用户通过账号和密码远程连接到登录节点提交作业。集群调度系统采用不同的作业调度策略对用户提交的作业进行调度。实际情况中, 用户可能拥有多个计算中心的使用权限, 但由于计算中心之间没有互联互通的机制, 当用户在不同计算中心中有运行作业时, 需要在不同的终端来回切换, 查看作业非常不方便。对某一计算中心上长时间运行的作业, 用户可能需要对作业做检查点并迁移到其他计算中心上做灾备。当计算中心时常会遇到例行维护等工作时, 用户可能需要将作业迁移到其他计算中心完成计算。

本文的剩余部分首先介绍相关工作, 包括作业迁移调度的相关工作; 第 3 部分讨论 JMS 的系统架构; 第 4 部分讨论 JMS 的设计与实现; 第 5 部分介绍 JMS 的应用实例; 最后是总结和下一步工作展望。

2. 相关工作

由于作业迁移容错具有重要的研究价值, 一部分工作集中在为作业调度系统添加作业容错机制, 提供检查点机制。文献[2]采用了 BLCR 检查点库在 SLURM 作业调度系统中实现了给予检查点/恢复的作业自动容错机制; 文献[3]介绍了在三层架构超级计算环境中通过作业管理系统的 checkpoint-recovery 机制实现作业容错。由于不同的计算中心可能采用不同的作业调度系统, 针对多计算中心的作业迁移容错没有办法依赖现有的作业调度系统完成。另一部分工作集中在基于虚拟化技术进行作业迁移, 文献[4]利用 Docker 私有仓库实现了在 Docker Swarm 集群中的容器迁移; 文献[5]基于 docker 技术设计并初步实现了 GUI 应用在线迁移的系统; 文献[6]对多种虚拟机迁移模型进行综合分析, 研发一个虚拟机在线迁移管理软件。文献[7]提出一个用于多数据中心云计算环境的虚拟机快速动态迁移架构和相应的迁移方法。在高性能计算中心, 由于需要充分利用计算资源的硬件能力而很少采用虚拟化技术进行作业计算, 因此基于虚拟化技术的作业迁移具有局限性。针对以上问题, 本文设计并实现了多计算中心作业迁移调度系统 JMS。

该系统实现了多计算中心的互联互通, 用户可以在 JMS 系统上操作不同的计算中心, 管理不同计算中心上的作业, 提高了用户对作业的监控能力。同时用户可以在不同的计算中心迁移作业, 提高了计算中心的资源利用率和容灾能力。JMS 提供了用户体验良好的 web 终端界面, 简化了用户对高性能计算中心的操作, 提高使用效率。

3. JMS 的系统架构

多计算中心作业迁移调度系统 JMS 由两部分构成: JMS_provider 和 JMS_agent。JMS_provider 单独部署在一个 web 服务器集群, 向用户提供作业迁移调度系统的 web portal。JMS_agent 部署在各个计算中心外部访问的登录集群, 对外提供 webservice 服务和文件传输服务。JMS 的系统网络拓扑结构如图 1 所示。

3.1. JMS_Provider

JMS_provider 为用户提供 web 访问 portal, 用户可以通过登录 JMS_provider 查看不同计算中心上运行的作业状态, 通过 JMS_provider 提供的远程终端功能, 用户可以方便切换到任意计算中心进行远程操作。用户可以根据计算中心的负载情况, 将作业从一个中心迁移到其他中心实现远程灾备。JMS_provider 的体系结构如图 2 所示。

JMS_provider 包含 5 个功能模块: Job Manager、Terminal Controller、Transfer Service、Migration Schedule、Web Portal。

Cluster Manager: Cluster Manager 收集用户在不同计算中作业的运行状态, 并返回给用户。

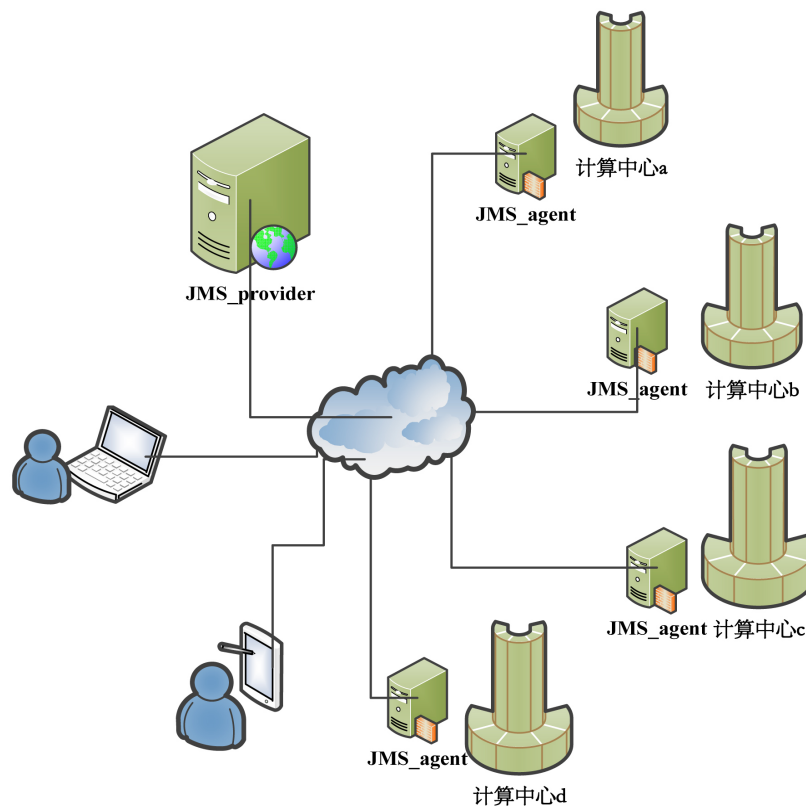


Figure 1. The networked topology structure of JMS
图 1. JMS 系统网络拓扑结构

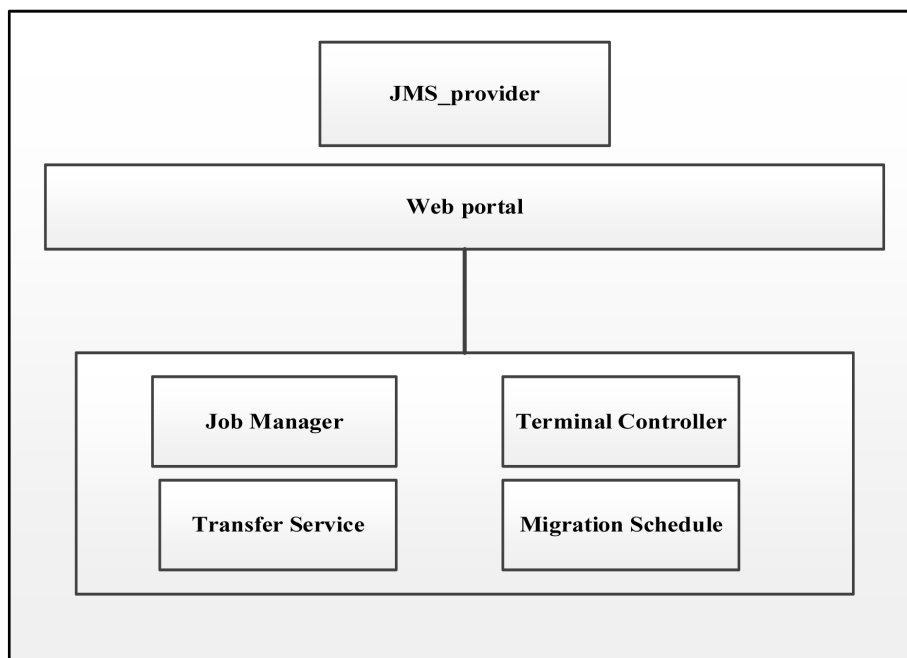


Figure 2. The architecture of JMS_Provider
图 2. JMS_Provider 体系结构

Terminal Controller: Terminal Controller 集成了多个计算中的远程终端接口, 用户可以方便的切换到不同的计算中心执行远程操作。

Transfer Service: Transfer Service 提供作业脚本提交功能, 用户可以上传提交到任意一个计算中心。

Resources Monitor: 监控各个计算中心的负载情况, 为用户提交作业和作业迁移做决策支持。

Portal: JMS_provider 对外提供的 Web 接口。用户可以通过登录 Portal 接受 JMS_provider 其他组件提供的服务。

3.2. JMS_Agent

JMS_agent 作为后台程序部署在各个计算中心外部访问的登录集群上, JMS_agent 收集本计算中心的负载情况为迁移调度提供决策支持, 同时 JMS_agent 对外提供了 Restful API 服务[8]和文件传输服务。JMS_agent 的体系结构如图 3 所示。

JMS_agent 包含 5 个功能模块: Restful API Service、Load Monitor、Communication Controller、Job Manager、Migration Service。

Restful API Service: Restful API Service 提供一组 web 服务的 API, 包括作业管理的 API (作业的查询, 提交, 检查点, 暂停, 删除)、计算中心资源查询的 API (计算中心的负载信息)、命令控制的 API (数据传输, 作业迁移)。

Load Monitor: Load Monitor 收集计算中心节点的负载信息, 包括各个节点的 CPU 负载, 内存的利用率, 温度等指标信息。

Communication Controller: 各个计算中心通过 Communication Controller 进行命令控制和数据传输。

Job Manager: Job Manager 和计算中心的作业调度系统交互, 向作业调度提交作业、查询作业运行状态。

Migration Service: Migration Service 根据用户选择的需要迁移的作业和迁移的目标计算中心, 实现

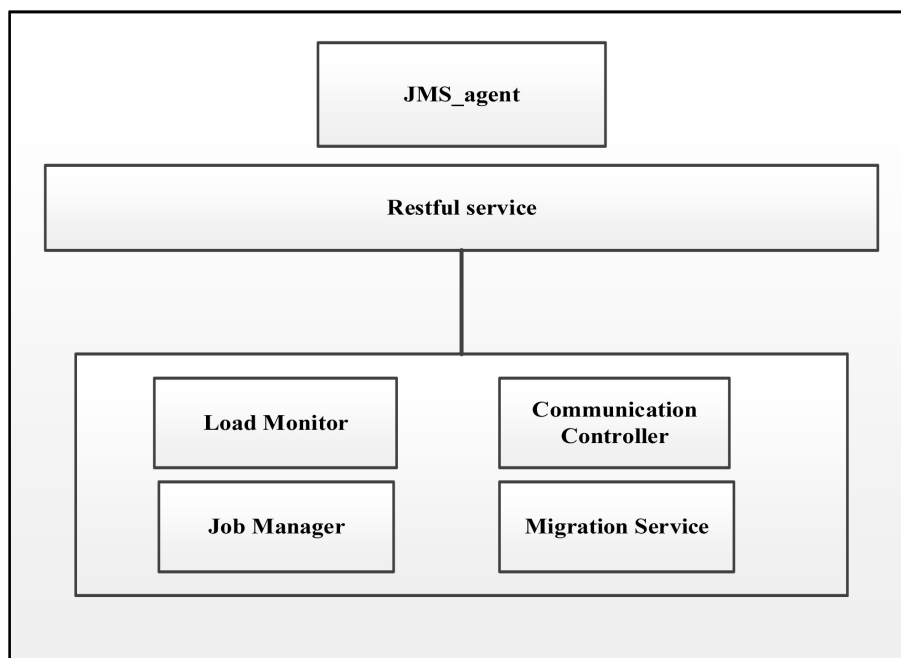


Figure 3. The architecture of JMS_agent
图 3. JMS_agent 结体系构

作业的迁移调度。

3.3. 通信协议

JMS_provider 和 JMS_agent 以及 JMS_agent 之间的通信数据主要分为控制信息和作业数据信息。

1) 命令控制协议

JMS_provider 的 Cluster Manager 模块和 Resources Monitor 模块通过 JMS_agent 提供的 Restful API 对计算中心进行访问。JMS_agent 将服务作为一种资源对外提供访问接口。

如查询作业的运行状态：

http://IP:/JMS_agent/jobstatus/jobID

如对作业做检查命令操作：

http://IP:/JMS_agent/checkpointment/jobID

2) 数据传输协议

当涉及作业提交或者作业迁移时，JMS_provider 的 Transfer Service 模块和 JMS_agent 的 Communication Controller 模块通过 SCP 协议传输大规模的作业文件。这两个模块通过对 scp 命令进行脚本包装，对外提供数据传输服务。

4. JMS 的设计与实现

JMS 系统主要的设计思想是通过整合多个计算中心的资源，方便用户实现在一个平台完成对多个计算中心作业的管理。在 JMS 平台上，用户可以同时管理多个计算中心上的作业，实现作业的动态迁移；可以切换不同的计算中心访问接口，远程控制命令行终端；可以选择任意一个计算中心提交作业，异地并发执行作业。JMS 采用 spring boot 和 docker 搭建微服务架构[9]。每一个功能模块都是 JMS 的一个组件。

4.1. 作业迁移管理模块

用户在 portal 选择的需要的迁移作业和目标计算中心。JMS_provider 的 Cluster_manage 模块将迁移的作业 ID 通过 JMS_agent 提供的 restful API 返回到 JMS_agent 的 Migration Service 模块。Migration Service 模块选择合适的作业迁移策略[10]完成迁移任务。作业迁移流程在系统中的处理流程如图 4。

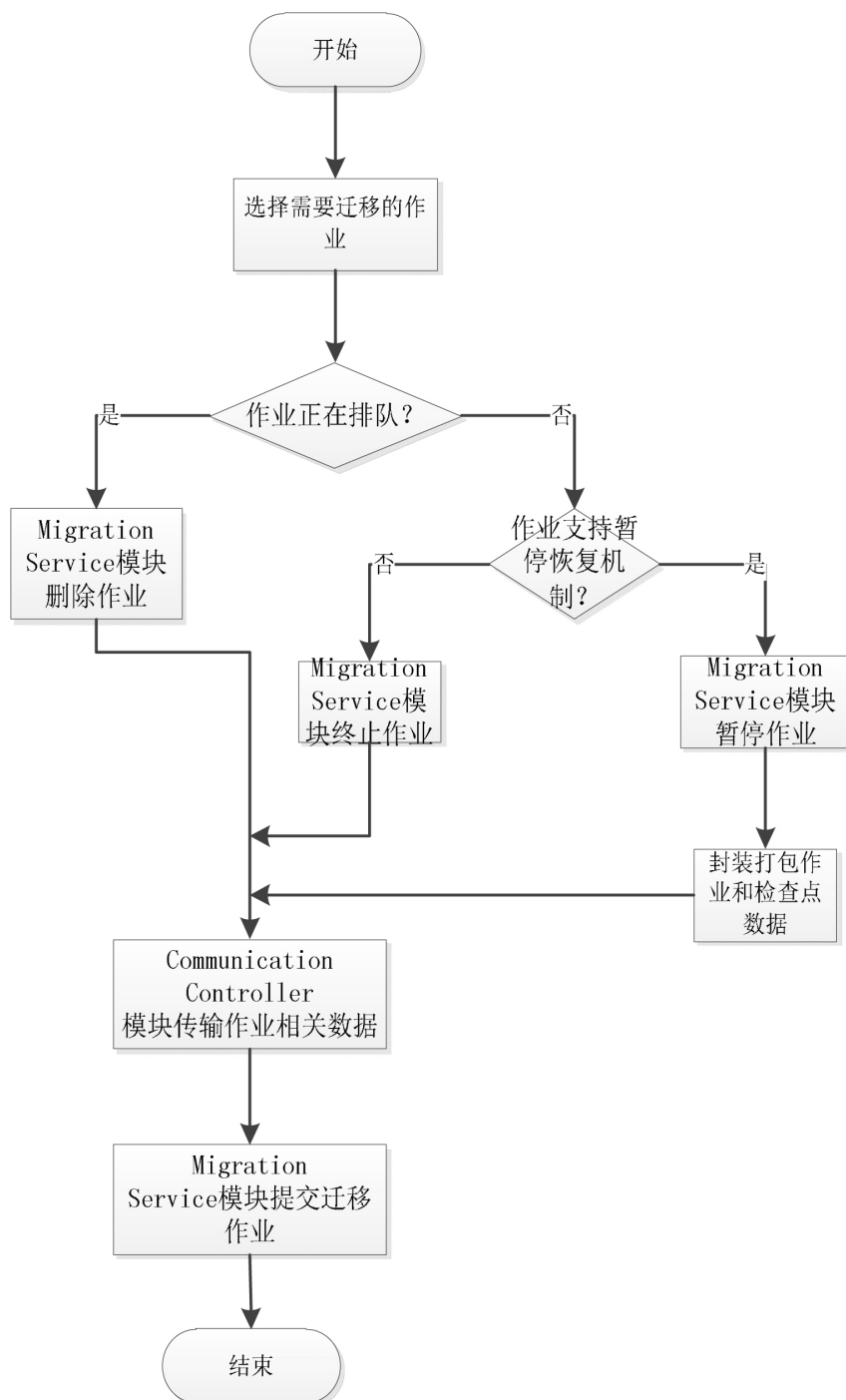


Figure 4. The process of the job migration
图 4. 作业迁移流程

对于某一个预迁移的作业, Migration Service 模块首先判断它的运行状态, 根据不同的作业的运行状态分别采用不同的迁移策略。1) 如果作业处在调度队列之中, 还没有开始运行, 则源计算中心 JMS_agent 的 Migration Service 模块将作业从源计算中心任务队列中删除, 接着通过 JMS_agent 的 Communication Controller 模块将作业传输到目标计算中心, 再由目标计算中心的 JMS_agent 的 Migration Service 模块提交作业到作业调度系统。2) 如果作业已经开始运行, 并且作业所对应的应用类型不支持暂停恢复机制, 则源计算中心 JMS_agent 的 Migration Service 模块将作业终止, 接着通过 JMS_agent 的 Communication Controller 模块将作业传输到目标计算中心, 再由目标计算中心的 JMS_agent 的 Migration Service 模块提交作业到作业调度系统。3) 如果作业已经开始运行, 并且作业所对应的应用类型支持暂停恢复机制, 则源计算中心 JMS_agent 的 Migration Service 模块将作业暂停, 保存当前任务状态信息, 将作业数据和检查点数据封装打包, 通过 JMS_agent 的 Communication Controller 模块将作业传输到目标计算中心, 再由目的计算中心 JMS_agent 的 Migration Service 模块对迁移作业进行恢复执行。

用户登录 web portal 之后, 通过菜单导航到多中心作业管理界面, 用户可以查看多个中心上作业的运行情况, 并且对作业进行管理, 用户可以选择作业进行检查点操作并进行作业迁移, 如图 5 所示。

当用户需要进行手动作业迁移时, 选择合适的迁移作业数据和目标计算中心后提交作业, 如图 6 所示。



Figure 5. The interface of multi-center job management

图 5. 多中心作业管理界面

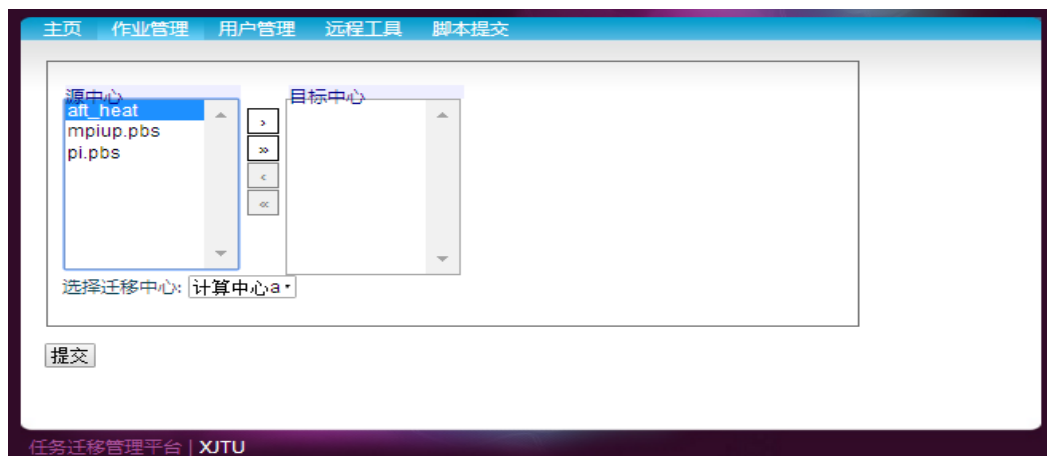


Figure 6. The interface of the job migration

图 6. 作业迁移界面

4.2. 终端控制管理模块

用户在 portal 选择的需要远程控制的目标计算中心。JMS_provider 的 Terminal Controller 模块根据用户选择的目标计算中心的 ID, 通过 JMS_agent 提供的 restful API 认证连接到了 JMS_agent 的 Communication Controller 模块。JMS_provider 的 Terminal Controller 模块和 JMS_agent 的 Communication Controller 模块建立了可靠、安全的专用通信通道和相应的通信协议。两个模块之间的通信基于 TCP/IP 协议, 为了减少 TCP 连接带来的开销, 在每次会话之后将中断连接。远程控制的通信时序图如图 7 所示。

在图 7 中, JMS_provider 的 Terminal Controller 模块和一个计算中心 JMS_agent 的 Communication Controller 模块建立了通信连接时, 首先 Terminal Controller 模块向 Communication Controller 模块发起连接请求, Communication Controller 模块产生应答并建立连接; 然后 Terminal Controller 模块向 Communication Controller 模块发送命令信息, Communication Controller 模块接收命令信息后调用本地的 shell 命令产生命令执行结果, 最后 Communication Controller 模块向 Terminal Controller 模块返回命令执行的结果, 断开连接。

TCP 是面向字节流的协议, 即没有消息边界, 应用层协议需要定义格式来决定消息边界, 通常的做法有两种: 一是使用固定分隔符的方式, 消息以固定的字符序列作为结尾, 消息内部的不同字段以固定的分隔符分割; 二是使用基于消息头消息体的方式, 消息头的长度和格式固定, 且在消息头中给出消息体的长度。本课题的多中心间的通信协议采用第一种方式, 所有协议消息以#<消息类型>开头, 以|end 作为结尾, 其他字段以#分隔。消息类型如表 1 所示。

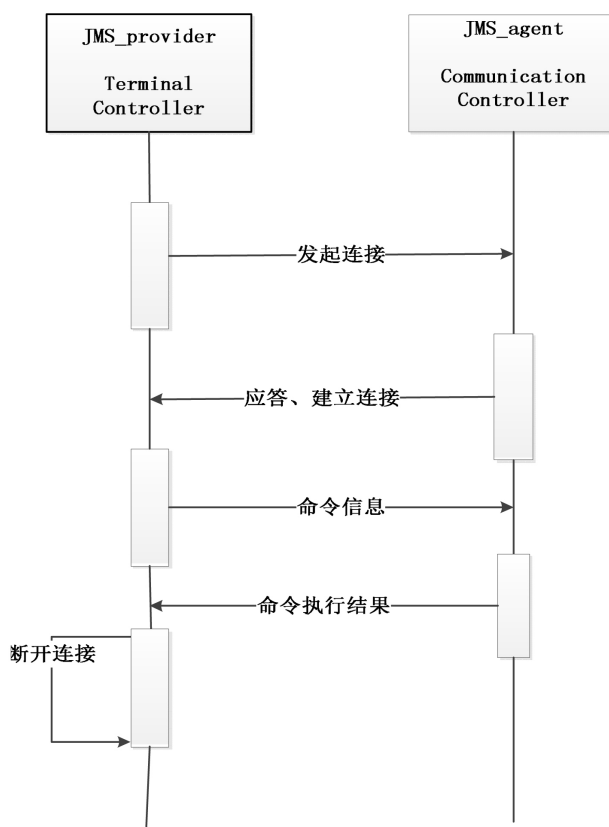


Figure 7. The communication sequence diagram of the remote control

图 7. 远程控制的通信时序图

Table 1. The table of message type**表 1.** 消息类型表格

消息类型	功能
#Request end	发起连接
#Response end	建立连接
#Command#content# end	命令信息
#Result#content# end	命令返回结果
#Close end	断开连接
#Datsent#content# end	数据发送

通信协议中分为两类消息:

- 1) 控制消息: 主要包括发起连接请求命令、建立连接命令、任务类型、任务运行状态以及任务进度。
- 2) 数据消息: 主要包括迁移作业的数据长度、数据内容以及运行脚本。

通过 web portal 用户可以跳转到多中心终端控制页面。通过该页面, 可以切换到任意一个计算中心进行远程控制, 如图 8 所示。

4.3. 作业提交管理模块

用户向 JMS 提交任务请求时, 首先根据应用类型选择作业基本属性模板, 并在 web portal 的脚本提交界面输入作业的基本属性信息, 如图 9 所示。一个作业基本的属性包括作业名称、队列名称、计算资源、预估运行时间和输入文件。这些信息被转化成 JSDL 作业描述信息并以 JSON 格式进行保存。

```
{Jobname:jobname, Queuename:queuename, Resource:resource,Walltime:walltime, Inputfile:inputfile}
```

除了作业基本属性信息, JMS 通过扩展的作业描述语言 E-JSDL 来表示一个作业的属性信息, 针对长时应用, 作业描述信息还包括用户设定的作业迁移周期和计算灾备中心。

JMS_provider 的 Transfer Service 模块通过 JMS_agent 提供的 restful API 将作业描述信息返回到 JMS_agent 的 Job Manager 模块。Job Manager 模块将用户作业提交到计算中心的作业调度系统, 根据作业的描述信息, 定期将作业结果迁移到灾备中心。

5. 应用实例

JMS 系统已经在国家计算网格(西安中心)和西安交通大学网络计算中心分别进行了部署和测试。JMS_provider 部署在 Web 服务器集群上, JMS_agent 分别部署在两个计算中心。LAMMPS 由美国 Sandia 国家实验室开发, 主要用于分子动力学相关的一些计算和模拟工作, LAMMPS 应用支持检查点机制, 可以在指定迭代步数之后进行检查点操作。下文以一个材料化学 LAMMPS 应用金属铜的剪切变形算力说明如何在 JMS 系统中进行作业迁移。

1) 通过 JMS_provider 提供的脚本提交界面, 用户输入金属铜的剪切变形算力的作业属性, 并提交。JMS_provider 根据作业描述信息, 将作业提交到国家计算网格(西安中心)。

2) 通过多中心作业管理界面, 用户点击金属铜的剪切变形算力作业的迁移按钮进入作业迁移界面。用户选择作业的检查点文件 restart_50,000 和迁移目标中心—西安交通大学网络计算中心。源中心的 JMS_agent 将检查点文件和作业迁移到目标中心。

在目标中心, JMS_agent 根据检查点文件信息重新提交作业, 恢复金属铜的剪切变形算力作业, 金属铜的剪切变形算力从第 50,000 次迭代恢复计算。金属铜的剪切应力与计算步数的关系如图 10 所示。

黑色曲线为国家计算网格(西安中心)上一次性计算完成的剪切应力与计算步数的关系曲线。红色曲线



Figure 8. The interface of the remote control
图 8. 远程控制界面

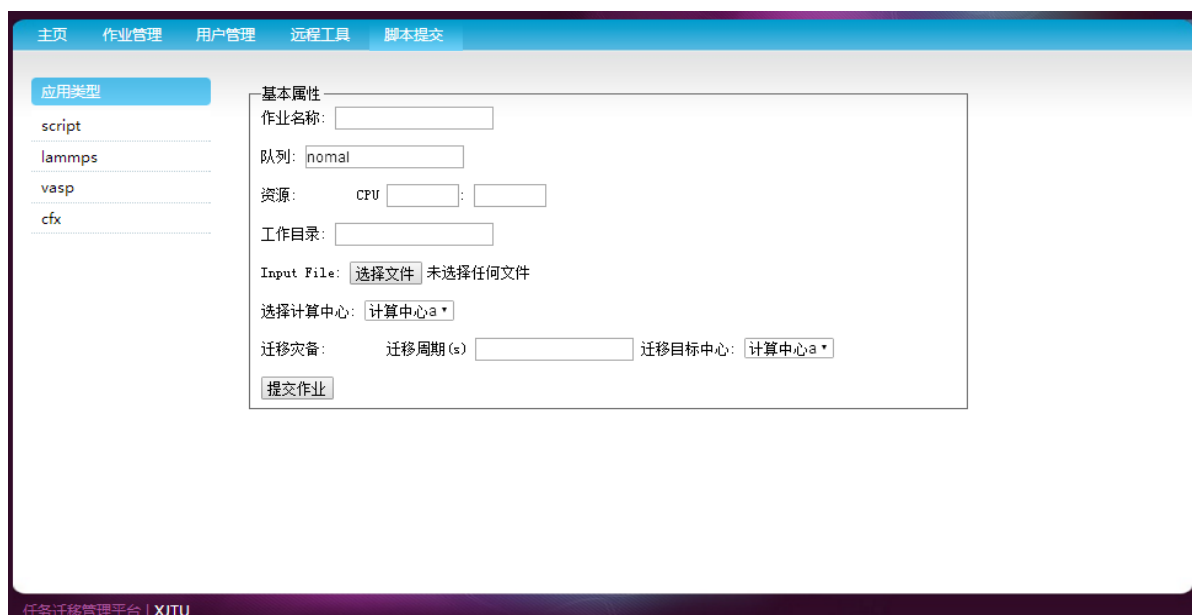


Figure 9. The interface of the script
图 9. 脚本提交界面

为在西安交通大学网络计算中心上从 50,000 步恢复执行之后的结果。可以看到, 两条曲线的变化趋势完全一致, 也在同一应力值处剪切屈服。

需要指出的是, 金属铜的剪切变形算力的作业迁移之后的曲线与原始曲线并非完全重合, 这主要是由于分子动力学计算中, 原子运动的随机性以及多 CPU (Cores) 并行计算时分配计算原子的不一致造成。

6. 结论

在现有计算中心调度系统的基础上, JMS 系统为多计算中心的作业迁移调度提供了解决方案。在 JMS 系统中, 用户可以同时管理多个计算中心的作业任务, 通过终端控制器操作不同的计算中心。用户提交作业时, 可以针对长时计算的应用做定时迁移灾备以减少由于某一个计算中心出现故障带来不必要的计算损失从而提高现有计算中心的可靠性。对于正在运行的作业, 用户可以根据需要将作业迁移到其他计

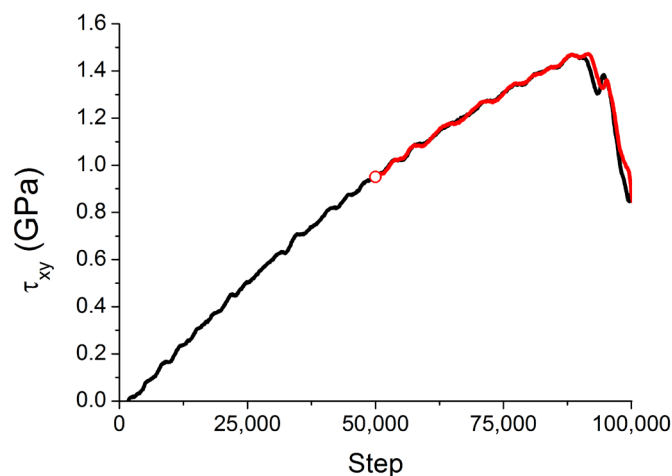


Figure 10. Shear stress-step curve
图 10. 金属铜的剪切应力与计算步数的关系曲线

算中心完成计算从而实现多计算中心协同工作。提高计算中心访问的安全性, 提出更加安全的认证方式是下一步的研究方向。

基金项目

国家重点研发计划(2016YFB0201402); 海外及港澳学者合作研究基金(61628210); 国家自然科学基金资助项目(91630206)。

参考文献

- [1] Xiao, H., Wu, H. and Chi, X. (2008) SCE: Grid Environment for Scientific Computing. *2nd International Conference on Networks for Grid Applications (GridNets 2008)*, 8-10 October 2008, Beijing, 35-42.
- [2] 曹宏嘉, 卢宇彤, 谢旻. 资源管理系统中基于作业检查点的自动容错[J]. *计算机工程与科学*, 2009, 31(11): 66-68.
- [3] 戴志辉, 肖海力, 曹荣强, 等. 三层架构超级计算环境容错框架[J]. *计算机应用研究*, 2011, 28(7): 2576-2579.
- [4] 毛祺, 卢胜林. 基于 DockerSwarm 集群的容器迁移策略的实现[J]. *信息技术*, 2016(9): 156-160.
- [5] 房锦章. 基于 Docker 技术的 GUI 应用的在线迁移研究[D]: [硕士或博士学位论文]. 北京: 中国科学院大学, 2016.
- [6] 张汉林, 王泽鹏, 陈宁江. 基于能耗感知的虚拟机迁移管理软件[J]. *广西科学院学报*, 2017, 33(1): 48-52.
- [7] 刘圣卓, 姜进磊, 杨广文. 基于副本的跨数据中心虚拟机快速迁移算法[J]. *清华大学学报(自然科学版)*, 2015(5): 579-584.
- [8] Richardson, L. and Ruby, S. (2007) *Restful Web Services*. O'reilly Media Inc., Sebastopol, 199-204.
- [9] Namiot, D. and Sneps-Sneppé, M. (2014) On Micro-Services Architecture. *International Journal of Open Information Technologies*, **2**, 24-27.
- [10] Rathore, N.K. and Chana, I. (2016) Job Migration Policies for Grid Environment. *Wireless Personal Communications*, **89**, 241-269. <https://doi.org/10.1007/s11277-016-3264-2>

知网检索的两种方式：

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：csa@hanspub.org