面向航天器设计的高性能计算平台研究与应用

安 洲 袁 义 宋文龙 潘慧芳 苗 奇
（北京空间飞行器总体设计部，北京 100094）

摘要：针对航天器规模越来越大、数量越来越多，需有效地整合计算资源，利用高性能计算技术，为航天器设计提供高效地仿真计算服务，围绕航天器设计过程中的仿真计算需求，对高性能仿真计算云平台架构、关键技术进行深入研究，梳理典型航天器设计中开展仿真计算的关键流程，建设易用、可靠性高、扩展性强的设计、计算一体化云平台，有效提升了航天器设计过程中的仿真计算效率。

关键词：航天器设计；仿真计算；高性能计算；云平台
中图分类号：TP399 **文献标识码**：A

Research and Application of High Performance Computing Platform for Spacecraft Design

An Zhou Yuan Yi Song Wenlong Pan Huifang Miao Qi
(Institute of Spacecraft System Engineering, Beijing 100094)

Abstract: With the increasing scale and quantity of spacecraft, how to effectively integrate computing resources and use high-performance technology to provide efficient simulation and computing services for spacecraft design is an urgent engineering problem to be solved. Focusing on the simulation and computing requirements in the spacecraft design process, this paper deeply analyzes the architecture and key technologies of the high-performance simulation and computing cloud platform, combs the key processes of simulation and computing in typical spacecraft design, builds an easy-to-use, reliable, scalable design and computing integrated cloud platform, and it effectively improves the efficiency of simulation calculation in spacecraft design.

Key words: spacecraft design; simulation calculation; high performance computing; cloud platform

1 引言

航天器仿真技术主要包括航天工程仿真和航天器系统仿真[1]，利用仿真技术模拟航天器在特定环境下的运行规律和技术状态，验证设计方案、模型的可行性，及早发现设计缺陷、暴露潜在的风险，能够有效提高航天器设计质量，降低物理试验验证成本。通过将航天器各分系统模型进行集成仿真，可有效提高航天器总体仿真水平[2]。

航天器设计涉及的专业多，如结构、流体、电磁、热等，各个专业之间耦合性强，且随着航天器规模越来越大、数量越来越多，整星级仿真验证愈加复杂、频繁。如在结构力学分析领域，需开展大量的整星频响分析、模态分析，太阳翼展开动力学分析、着陆缓冲装置力学分析等单机产品的仿真分析；热分析领域，在极端温度场下对热设计方案进行仿真验证，确保航天器温度范围满足在轨各项指标要求。在航天器设计过程中对仿真结果及时性要求高，需要针对多种计算工况快速响应、迭代验证，通过修正设计参数，持续进行仿真计算。

为设计、求解更大规模的模型，缩短模型求解时间，亟需构建面向航天器设计的高性能仿真计算云平

作者简介：安洲（1988），硕士，计算机软件与理论专业；研究方向：高性能计算技术。
收稿日期：2022-07-15

台，对计算资源进行统一管理和调度，按需向航天器型号设计人员提供仿真设计、计算云服务。

2 平台架构设计

面向航天器设计的高性能仿真计算云平台架构如图 1 所示，主要包括平台功能层、仿真应用层、平台接口层、文件系统层、基础设施层。平台采用 B/S 部署模式，面向设计师提供一体化的仿真设计、计算服务，集成通用的结构、流体、电磁、热分析等专业应用，同时预留二次开发接口供定制化仿真分析应用的集成对接。

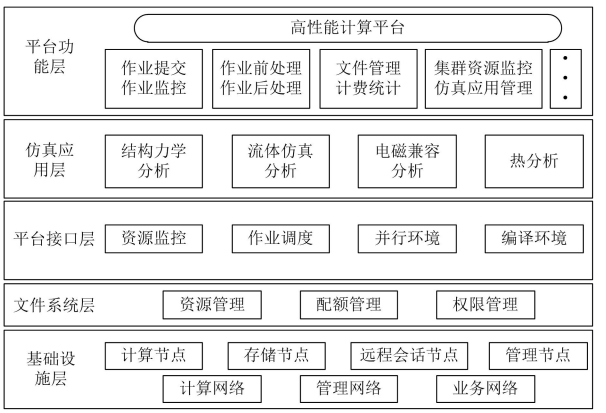

图 1 平台整体架构设计

2.1 基础设施层

平台采用混合体系架构模式，支持异构计算，由计算节点、存储节点、远程会话节点、管理节点等 4 类主要节点和管理网络、计算网络、存储网络等 3 类核心网络组成。

其中，计算节点由分布式集群计算节点（Cluster）、共享内存计算节点（SMP）、异构计算节点（GPU+CPU）组成，面向不同业务场景提供有针对性的计算服务，与高性能计算平台的架构发展同步，分布式集群计算节点（Cluster）作为当前高性能计算机的绝对主流[3]，提供了平台的大部分算力。基于 ParaStor 并行分布式文件系统的存储节点，提供高可靠、支持横向扩展的数据存储服务。远程会话节点提供三维设计服务，用于仿真计算的前、后处理环节。管理节点作为平台资源调度核心，通过集群模式部署。

管理网络通过以太网连接管理节点和其他各类节点，用于传输调度、管理指令。计算网络采用低延迟互联设备 InfiniBand 交换机连接管理节点、计算节点，可有效提升并行计算效率。存储节点之间采用 InfiniBand 网络互连，获得更高的网络带宽和更低的通信延迟。

2.2 文件系统层

面对多用户、多进程的并发数据读写操作，平台需要配置并行文件系统对数据的访问进行管理。天河系列超级计算机在开源分布式并行文件系统 Lustre 基础上，构建高性能混合层次式并行文件系统 H2FS（Hybrid Hierarchy File System）对数据、存储进行管理[4]；红衫采用 IBM 公司推出的并行分布式通用并行集群文件系统 GPFS[5]，基于扩展哈希技术支撑大量文件和目录的管理。

为满足平台对存储的扩展性、可靠性、并发访问的要求，采用曙光 ParaStor300S 并行分布式云存储系统，其弹性扩充的 Scale-out 架构、数据容错重构技术和故障自动恢复机制、多级性能加速特点，为上层应用及平台用户提供透明的数据存储服务，支撑管理节点、计算节点的高并发访问。

2.3 平台接口层

平台接口层包括作业调度系统、并行程序接口、资源监控接口、编译运行环境等，其中作业调度系统是平台资源高效利用的关键，负责统一管理和调度平台的软硬件资源，按照一定的策略为用户分配资源、提供服务。

当前应用广泛的作业调度系统[6]包括 LSF（Load Sharing Facility）、PBS（Portable Batch System）、SGE（SUN Grid Engine）、Slurm（Simple Linux Utility For Resource Management）等，其中 Slurm 是一个开源的、具有较好的容错机制和高度可扩展的集群管理和作业调度系统，适用于大型和小型 Linux 集群管理[7]，支持定制集群管理和作业调度功能。

2.4 仿真应用层

应用层面向平台用户，针对航天的工程仿真计算需求，提供结构、流体、电磁、热等专业应用，解决 4 大物理场的仿真计算任务。航天型号总体系统仿真是针对某一特定任务需求，涉及多个专业、多个分系统的联合仿真。从航天器型号研制过程看，仿真分析贯穿航天器可行性论证、方案论证、工程设计、飞行试验的各个阶段，已经成为航天器设计与评价的主要手段，发挥着越来越重要的作用[1]。

通过对仿真分析应用进行封装，集成到应用 Portal 供设计师使用。调用底层平台接口层提供的接口，支持自定义参数方式提交个性化的仿真计算作业。利用仿真技术模拟航天器在特定环境下的运行规律和技术

状态，为确定系统边界提供定性、定量的依据。

2.5 平台功能层

平台功能层面向用户提供统一的 Web 操作界面，包括作业管理、文件管理、远程会话管理、仿真应用管理、集群资源监控等。

面向平台使用人员，通过与仿真应用层进行集成，用户可直接通过作业提交模块，选择仿真应用，配置作业参数，选择计算文件、计算资源进行作业的提交。针对运行中的作业，可实时查看作业相关进程的运行情况、资源占用情况、计算输出文件等信息，并可对作业状态进行灵活控制。

面向平台管理人员，实时监控计算资源的使用情况，计算作业和设计任务的运行情况，仿真分析软件的使用分布，快速掌握平台全局概况，包括节点启用率、资源使用率、许可使用率等信息。

3 平台关键技术

3.1 资源管理与调度

资源的管理与调度是高性能计算平台的核心组件，平台中资源是相对固定的，当多用户并发访问时，如何对资源进行动态管理和分配，使平台利用率最大化，充分发挥高性能计算的优势才是解决问题的关键。平台利用 Slurm 作为资源管理与调度工具，基于插件机制的基础架构对系统的扩展、定制、维护带来极大的便利。Slurm 体系结构如图 2 所示，其核心进程是 slurmctld 控制进程，负责监控资源和作业。slurmctld 在主管理节点上运行，同时为提高系统可靠性，在备管理节点上对 slurmctld 控制进程进行备份，当主管理节点进程出现故障时，备管理节点上运行 slurmctld 控制进程。

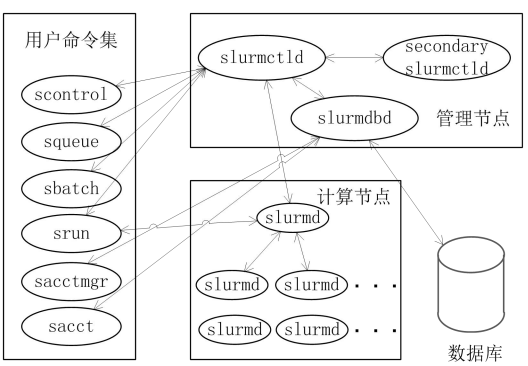

图 2 Slurm 体系结构

对于平台中的每个计算节点运行一个 slurmd 守护进程，用于接收 slurmctld 控制进程分配的任务，监控任务的运行状况、定时收集计算节点资源使用情况并反馈给控制进程。

Slurm 作为集群管理和作业调度系统，核心功能是调度，按照一定的策略将多个作业合理地分配到计算节点上执行，为合适的作业分配合适的资源。Slurm 提供了插件机制的作业调度策略，支持多种常用策略选取，还提供了丰富的 API 接口，允许用户定制调度策略。常用的调度策略有：FIFO（First In First Out，先进先出）、Reservation（预约策略）、Backfill（回填策略）。

在对资源进行充分利用的前提下，设计调度策略对计算节点进行休眠、唤醒操作，降低资源的空置率。以运行队列为维度，当队列中的计算节点一段时间内无负载时，调用节点 IPMI 接口，空闲节点休眠；在队列的排队作业一段时间内超过阈值时，唤醒队列中已休眠的计算节点，供排队作业使用。为避免出现节点频繁的休眠、唤醒操作，需定期对作业运算时长、排队时长等信息进行统计分析，动态调整阈值。

3.2 平台前端框架

为了提升用户体验、提高开发效率、易于扩展迭代，平台采用了前后端分离的架构，通过对比当前主流前端框架 Bootstrap、Angular、React、Vue，选择了渐进式 JavaScript 框架 Vue。

Vue 是一套用于构建用户界面的渐进式 JavaScript 框架，基于 MVVM（Model-View-ViewModel）设计模式开发，简化了用户界面的事件驱动编程方式。此外，Vue 的轻量化、高运行效率、双向数据绑定、开发的生态环境等特点，使前端开发更加便捷、高效。通过 JSON 格式字符串进行数据交互，实现了系统前后端完全分离，提高了系统的灵活性和可扩展性[8]。

3.3 请求负载均衡

对于平台的最终用户，直接面对的是平台 WEB 访问页面，对应平台架构中的功能层。为了保证用户的访问请求得到快速、稳定的响应，平台功能层采用集群模式部署，利用负载均衡[9]技术将用户请求按照一定的规则分配到多台服务器。

平台采用 Nginx 的反向代理功能，实现将用户请求分发到后端集群，处理请求并通过反向代理服务响应用户。在配置负载均衡策略时，采用了加权轮训的算法，根据集群中节点实际性能设置不同的权重，实现用户访问请求的转发。在实际应用过程中，可以对集群节点的有效权重进行动态调整，比如发现某节点出现异常，可以降低该节点的有效权重，减少转发至

该节点的用户请求。此外，在集群节点负载激增时，可以通过增加集群节点，调整负载均衡策略，将用户请求分流至新节点。对于负载均衡策略的变更，Nginx 支持热部署模式，在不影响用户当前访问的前提下，对配置进行升级，提高了用户的使用体验。

3.4 高可用数据库

MySQL 是一个开源的关系型数据库管理系统，使用标准化的 SQL 访问数据库，基于其轻量、速度快、运行成本低的特点，平台选用 MySQL 作为后台数据库，用于存储管理用户、作业、运行记录等信息。

为提高平台整体的可靠性，作为底层数据库的 MySQL 稳定运行的重要性不言而喻。对于数据库的可用性、可维护性对平台的影响进行了分析，为避免出现数据库死锁，提高平台的响应速度，选择主备模式的异步复制方案[10]。主节点数据库的更新、删除等变更操作会记录到二进制日志文件，同步更新日志内容至备节点，备节点执行日志中包含的更新操作，实现主备节点数据的一致性。

3.5 并行计算环境

对于一个支持并行计算的作业，通过对作业进行分解，将分解后的子任务分配给平台 Cluster 中多个节点，各个节点之间相互协同，并行开展计算，加快作业的求解速度，提高求解精度，支持更大规模的求解应用。

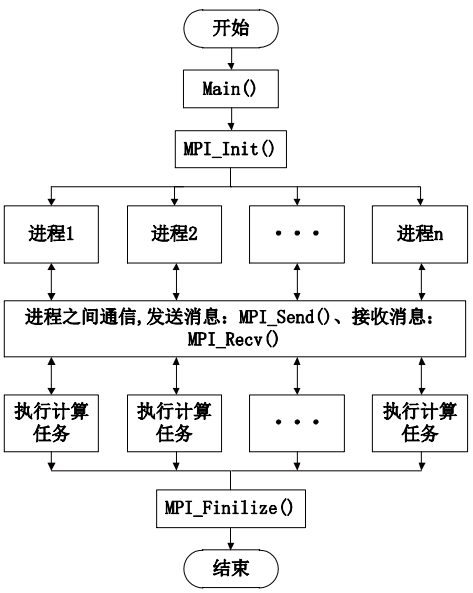

图 3 并行计算流程

消息传递接口（Message Passing Interface，MPI）是高性能计算领域中主要的并行计算技术，提供了并行计算的标准编程接口，不依赖于底层的编程语言。通过使用 MPI 编写进行数据交换的并行程序，在进程间发送和接收封装了数据的消息，并行计算流程如图 3 所示。MPI 已成为分布式计算集群上、进程间进行通信的并行计算的标准，主要的 MPI 实现有 MPICH[11]、LAM-MPI[12]、OpenMPI[13]，其中 OpenMPI 是在 LAM-MPI、LA-MPI 和 FT-MPI 基础上整合的一种基于构件的 MPI 实现，被应用在 TOP500 中非常多的超级计算机上[14]。

4 平台实现及应用

4.1 平台部署

平台计算节点由曙光 CX50-G35 刀片服务器、I980-G30 服务器、X745-G30 服务器构成，管理节点由曙光 I840-G30 服务构成，存储网络、计算网络由迈络思 CS7510 的 IB 交换机连接，管理网络由华为 S5731 交换机连接，平台拓扑结构如图 4 所示。

存储节点由曙光 ParaStor300S 节点构成，部署 ParaStor 分布式文件系统，在创建节点池、存储池、文件系统的基础之上，将文件系统挂至管理节点、计算节点，提供文件并行访问服务。

管理节点、计算节点分别部署 Slurm 的对应的管理节点、计算节点组件，实现资源的管理与调度。在分布式文件系统中，部署 OpenMPI 以及所需的编译运行环境，支持程序编译及并行计算，部署仿真应用软件，支持开展工程仿真计算。管理节点部署 MySQL 数据库、Nginx、Web 前端、后端服务，初始化数据库。

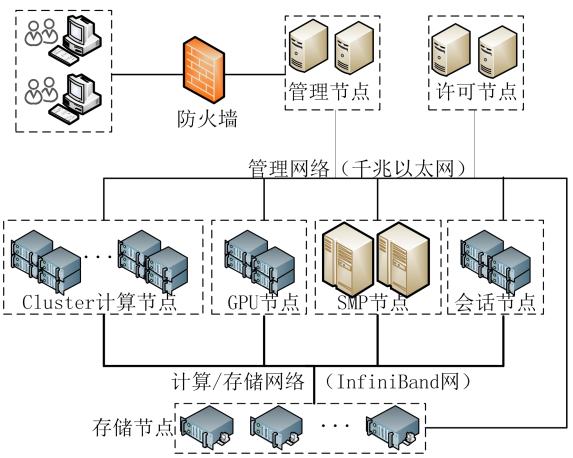

图 4 平台拓扑结构

4.2 应用流程

以某型号航天器设计中基于 MSC.Patran/Nastran 开展结构力学分析计算[15]为例，对平台应用流程进行

梳理。在完成相关模块部署，初始化配置后，通过浏览器访问平台并登录，利用平台设计、计算一体化服务，通过建模、求解计算、后处理完成仿真任务。

4.2.1 建模

在远程会话模块，利用平台图形处理能力进行仿真建模。选择仿真应用 Patran 建模，创建几何模型，划分有限元网格，确定边界条件，最后提交分析模型，生成 bdf 文件。

4.2.2 提交作业求解

通过作业管理模块中提交作业，利用平台的计算能力进行仿真计算。在作业提交页面，选择 Nastran 应用，选择建模阶段生成的 bdf 数据文件、运行队列、计算节点数量、计算资源等参数，提交作业，进行求解计算。

用户在提交作业页面完成参数配置后，平台首先会根据配置信息自动生成相应的可执行脚本文件，然后通过作业调度系统 Slurm 运行脚本文件，作业进入队列排队，匹配调度策略，分配计算资源，执行计算任务等一系列操作。

4.2.3 作业监控

在作业运行过程中，可实时查看作业相关进程的运行情况、资源占用情况、计算输出文件等信息，并可对作业状态进行灵活控制。

4.2.4 后处理

作业求解完成后，在远程会话模块，通过 Patran 处理计算结果 xdb、op2，显示变形图、应力云图。

4.3 应用效果

高性能计算平台建成后，使用平台远程会话模块开展仿真前、后处理，在线提交作业开展仿真计算，实现了仿真设计、计算一体化，数据在云端存储无缝流转。通过设置不同的运行队列，提供有针对性的计算服务，满足航天器仿真对多 CPU、大内存、GPU 等不同计算资源的需要，如使用有限元分析软件开展结构力学仿真，适合基于共享内存的 SMP 队列；在支持多节点并行计算热分析中，选取多节点并行 Cluster 队列。平台提供友好的作业提示信息、预填充的运行参数、参数校验机制，提升了平台的用户体验，降低了用户应用的学习成本。对于平台核心功能资源管理与调度、前端服务、数据库等设计冗余架构，避免出现单点故障，提升平台整体的可靠性。在资源管理与调度中，提出了绿色节能的设计思路，动态调整阈值，通过算法自动休眠、唤醒计算节点，提高了能源的利用率。

5 结束语

面对航天器设计涉及专业多、模型复杂的特点，仿真计算需求日益增长的现状，梳理典型仿真计算流程，建设高性能仿真设计、计算一体化云平台。平台实现资源集中管理与调度，面向设计师提供易用、可靠性高、扩展性强的仿真设计/计算服务，有效提升了航天器设计过程中的仿真计算效率。

参考文献

1 包为民. 对航天器仿真技术发展趋势的思考[J]. 航天控制，2013(31)：4～8

2 邢涛，周晖，魏传锋. 航天器分布式系统仿真验证平台设计与实现[J]. 航天器环境工程，2015(32)：496～499

3 袁国兴，张云泉，袁良. 2020 年中国高性能计算机发展现状分析[J]. 计算机工程与科学，2020，42(12)：2103～2108

4 董勇，周恩强，卢宇彤，等. 基于天河 2 高速互连网络实现混合层次文件系统 H2FS 高速通信[J]. 计算机学报，2017(40)：1961～1979

5 张新诺，王彬. GPFS 文件系统的安装配置与维护[J]. 计算机技术与发展，2018(28)：174～178

6 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(4)：8

7 Sched M D. Slurm workload manager [OL]. http://slurm.schedmd.com,2022

8 尤雨溪. Vue.js[OL]. https://cn.vuejs.org,2022

9 张宇星，马明栋，王得玉. 基于 Nginx 负载均衡的动态改进算法[J]. 计算机技术与发展，2020，30(3)：5

10 张伟丽，江春华，魏劲超. MySQL 复制技术的研究及应用[J]. 计算机科学，2012(39)：168～170

11 Gropp W, Lusk E. User's Guide for mpich, a Portable Implementation of MPI[J]. Office of Scientific & Technical Information Technical Information Technical Reports, 2017(17): 2096～2097

12 Margaris A I. Local Area Multicomputer (LAM-MPI)[J]. Computer and Information Science, 2013, 6(2): 1～8

13 Perks O, Beckingsale D A, Dawes A S, et al. Analyzing the influence of Infini Band choice on Open MPI memory consumption[C]. 2013 International Conference on High Performance Computing & Simulation (HPCS). IEEE, 2013

14 郭羽成. MPI 高性能云计算平台关键技术研究[D]. 武汉：武汉理工大学，2013

15 袁家军，陈珅艳，黄海. 基于 Patran/Nastran 的结构优化系统的工程应用[J]. 北京航空航天大学学报，2006(32)：125～129