

8-7-2020

Advances in Research and Application of In-situ Visualization

Wang Fang

1. School of Computer, National University of Defense Technology, Changsha 410073, China;;2. Computational Aerodynamics Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China;

Sikun Li

1. School of Computer, National University of Defense Technology, Changsha 410073, China;;

Zhao Dan

2. Computational Aerodynamics Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China;

Zeng Liang

1. School of Computer, National University of Defense Technology, Changsha 410073, China;;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Advances in Research and Application of In-situ Visualization

Abstract

Abstract: Scientific-Visualization is the most effective means of analyzing numerical simulation data. With the continuous improvement of the high performance computing technology, numerical simulation scale is growing rapidly. It is very difficult to analyze large scale parallel numerical simulation data efficiently due to the limits of IO bandwidth, storage system capacity and post processing pattern of visualization. In-situ visualization utilized the same computer system that ran the simulation tasks to operate visual computing and rendering, so it could overcome the limits of IO and storage system. *The advances of in-situ visualization in recent years were reviewed, including data management, large-scale parallel visualization algorithm, interaction, application framework, visualization software and some use cases. The challenges and trends of in-situ visualization were discussed.*

Keywords

in-situ visualization, extreme-scale parallel processing, framework, visualization software

Recommended Citation

Wang Fang, Li Sikun, Zhao Dan, Zeng Liang. Advances in Research and Application of In-situ Visualization[J]. Journal of System Simulation, 2015, 27(10): 2589-2599.

原位可视化研究与应用进展

王昉^{1,2}, 李思昆¹, 赵丹², 曾亮¹

(1.国防科学技术大学计算机学院, 长沙 410073;
2.中国空气动力研究与发展中心计算空气动力学研究所, 绵阳 621000)

摘要: 科学可视化是分析数值模拟数据的最有效手段。随着高性能计算机运算能力的不断提升, 数值模拟问题的规模迅速增长, 但 IO 带宽、存储系统容量和后处理的可视化模式很难满足超大规模并行数值模拟数据的可视化分析需求。原位可视化在运行数值模拟任务的高性能计算机系统上进行可视化计算和绘制, 克服 IO 和存储容量的限制, 实现了在线数据可视化。从数据组织管理、并行可视化算法、交互、集成框架与软件、应用案例等方面介绍了原位可视化近年研究和应用进展, 对原位可视化面临的技术挑战以及未来发展趋势进行了分析。

关键词: 原位可视化; 大规模并行处理; 软件框架; 可视化软件

中图分类号: TP391

文献标识码: A

文章编号: 1004-731X (2015) 10-2589-11

Advances in Research and Application of In-situ Visualization

Wang Fang^{1,2}, Li Sikun¹, Zhao Dan², Zeng Liang¹

(1. School of Computer, National University of Defense Technology, Changsha 410073, China;
2. Computational Aerodynamics Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China)

Abstract: Scientific-Visualization is the most effective means of analyzing numerical simulation data. With the continuous improvement of the high performance computing technology, numerical simulation scale is growing rapidly. It is very difficult to analyze large scale parallel numerical simulation data efficiently due to the limits of IO bandwidth, storage system capacity and post processing pattern of visualization. In-situ visualization utilized the same computer system that ran the simulation tasks to operate visual computing and rendering, so it could overcome the limits of IO and storage system. *The advances of in-situ visualization in recent years were reviewed, including data management, large-scale parallel visualization algorithm, interaction, application framework, visualization software and some use cases. The challenges and trends of in-situ visualization were discussed.*

Keywords: in-situ visualization; extreme-scale parallel processing; framework; visualization software

引言

科学可视化^[1]是分析处理数值模拟数据的重要手段, 其实质是利用计算机图形学技术, 将数据

以人类可感知的视觉信号(几何形状、色彩等)重新表达, 并借助人机交互手段辅助研究人员对数据进行进一步的探知和分析。传统的科学可视化普遍采用后处理模式, 即将数值模拟结果数据保存在磁盘中, 当进行可视化处理时再从磁盘读取数据。后处理可视化模式存在两个突出问题: 1) 数值模拟数据必须在时空两个维度上进行采样后才能与有限的存储能力匹配, 在此过程中大量有效信息被丢弃, 用于可视化的科学数据完整性无法保证; 2) 磁



收稿日期: 2015-06-14 修回日期: 2015-08-31;
作者简介: 王昉(1976-), 男, 陕西宝鸡, 博士生, 研究方向为科学可视化、并行计算; 李思昆(1941-), 男, 山东青岛, 教授, 博导, 研究方向为虚拟现实与可视化、SoC 设计方法学; 赵丹(1983-), 男, 四川绵阳, 博士, 研究方向为科学可视化。

<http://www.china-simulation.com>

• 2589 •

盘 I/O 性能有限, 超大规模数据在计算节点、磁盘阵列和可视化节点之间的传输、管理和分区预处理等操作会耗费大量的系统资源和处理时间^[2], 极大影响了可视化效率, 无法满足及时、实时数据分析的要求。

原位可视化(in-situ visualization)指在同一套高性能计算机系统上, 数值模拟与可视化处理紧密结合、协同工作, 对计算结果数据在原位直接进行可视化处理, 从而有效减少数据传输和 I/O 操作开销, 满足大规模数值模拟数据及时或实时可视分析的需求^[3]。

原位可视化技术的概念原型早在 1987 年就被提出^[1], 由于当时的后处理可视化能力能够满足绝大多数数值模拟数据规模的要求, 研究工作出现短暂停滞。2006 年后, 随着高性能计算进入 P 级时代, 磁盘 I/O 性能与计算能力之间的差异进一步扩大, 原位可视化研究重新被业内重视。在解决数据生产速度、复杂度和精度与存储能力和后处理可视化能力不匹配问题的众多手段中, 原位可视化是一种最有效的技术途径。近年来, 国内外对原位可视化技术的研究工作越来越深入, 应用成果不断增多^[4]。

2009 年 Ma^[5]总结了原位可视化技术发展面临的挑战, 主要包括: 原位可视化具有哪些特有需求, 在可视化计算与数值模拟间如何实现计算与存储资源高效共享, 如何确定两类计算时间比例, 哪一种可视化方式最适合于原位数据处理, 原始数据如何缩减和存储, 如何构建通用原位可视化平台, 现有的商业或开源可视化软件能否直接支持 E 级的

原位可视化等。2013 年, 单桂华等^[6]结合千万亿次计算机的原位可视化, 对原位可视化的数据压缩、特征提取与跟踪和绘制技术进行了研究综述。

本文简要介绍了原位可视化的基本模式, 重点对近年来的原位数据组织管理、并行可视化算法、交互、集成框架与软件、典型应用等进行研究综述, 并对存在的问题和发展趋势做出分析。

1 原位可视化基本模式

原位可视化有 3 种模式: 紧耦合、松耦合(图 1)和混合模式(图 2)^[7]。紧耦合模式指数值模拟和可视化在相同的计算节点上运行, 通过内存共享数据。其优势在于能最大限度减少数据传输和 IO 开销, 主要缺点: 1) 同一节点上资源有限, 两类应用存在竞争资源的可能; 2) 在进行可视化计算时, 数值模拟进程必须等待可视化处理完毕才能继续运行; 3) 计算节点上一般不配置图形处理部件, 绘制效率低。

松耦合模式指数值模拟和可视化处理在不同节点集合上运行, 通过推送(push-driven)或抽取(pull-driven)方式进行数据通信。其优势在于: 1) 可视化对数值模拟进程干扰小, 两类应用可以以异步方式运行, 不会出现争夺处理器核心和节点内存的情况; 2) 可视化节点可配置专用图形卡, 绘制效率高, 缺点在于: 1) 数据传输量大; 2) 数值模拟和可视化节点数量不同, 数据需要重新组织和管理, 操作复杂。

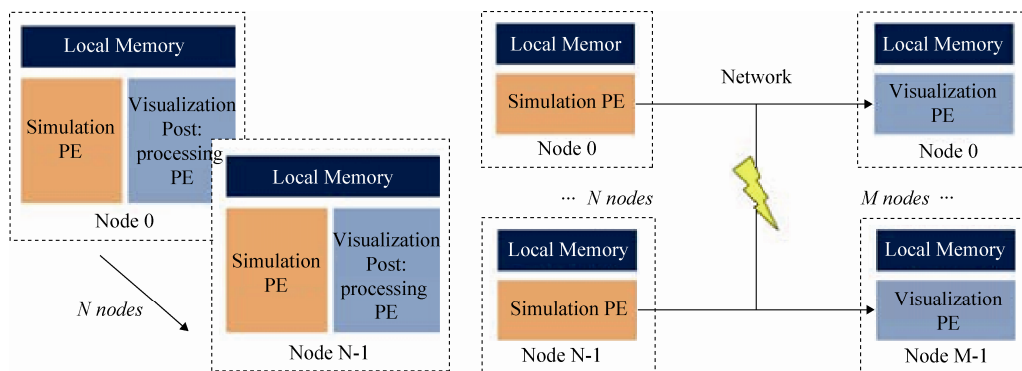


图 1 原位可视化的紧耦合(左)与松耦合(右)模式^[7]

混合模式是在松耦合模式的基础上, 将一部分计算量不大的数据处理操作放在计算节点上进行, 处理后的数据传至可视化节点上继续可视化计算和绘制。这种模式是否有效的关键在于计算节点上的数据处理操作是否有效, 判断标准主要有两点:

1) 对数值模拟进程的影响小, 即占用的处理器核心和内存资源少, 运行速度快; 2) 能有效缩减传输数据规模。混合模式是否适用, 需考察在数据分析时使用的可视化手段, 以及由此提出的数据缩减、数据重组织要求。

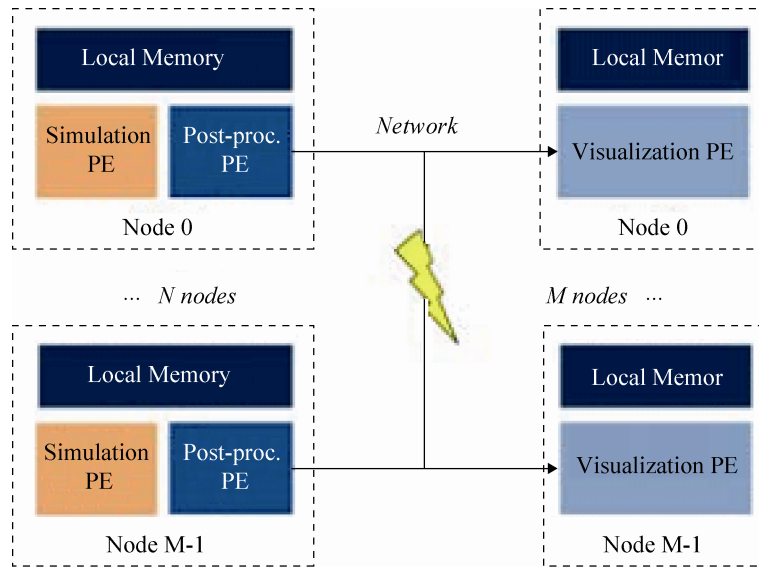


图 2 原位可视化的混合模式^[7]

2 原位数据组织管理

在原位可视化中, 数值模拟和可视化数据均存储于分布式内存中, 两类应用/进程之间的数据共享和交换通过内存和节点间高速互连网络完成, 合适的数据组织管理策略对于减少总处理时间、优化资源利用率有重要意义。

紧耦合模式下, 数值模拟和可视化进程通过同一节点上的内存共享数据, 采用相同的数据结构可减少数据转换产生开销。Tu 等以在分布式并行环境下实现了 PDE 数值计算的网格生成、解算、结果数据体绘制和图像合成^[8], 所有类型的计算均建立在同一套并行八叉树数据结构上, 不需在内存内拷贝、转换数据。Duque 等在实现与 CFD 数值模拟紧耦合的原位可视化时, 在同一进程内使用不同线程实现 CFD 解算、数据采样和可视化, 不同的线程共享内存数组^[9]。VisIt^[10]使用 libsim^[11]、

ParaView^[12]使用 Catalyst^[13-14]插件实现紧耦合原位可视化时, 数值模拟应用通过 API 接口申明自身数据结构的描述信息, 可视化时通过适配器将数据转换成 vtk^[15]结构, 灵活性较好, 但内存开销较大。

在松耦合或混合原位可视化模式下, 由于数值模拟和可视化节点数量存在差异, 需要数据重分布等操作完成内存到内存的数据移位和管理。针对此类应用场景, 常见的思路是建立中间件^[16-17]或分层应用框架^[18-20], 通过接口方式向上层应用提供相应服务, 在实现高效传输管理的同时屏蔽底层细节。Soumagne, Biddiscombe 等在 ICARUS 框架中建立了基于 HDF5 格式的 DSM(Distributed Shared Memory)虚拟文件系统, 用于支持原位可视化的数据交换和组织管理^[21-25]。DSM 是在可视化节点集合共享内存区域上建立的一个内存文件系统, 数据传输时, 数值模拟程序调用 HDF5 接口传送结果数据到 DSM, 传输完毕后通过推送方式通知可视化

进程获取数据并启动计算和绘制；在数据重分布时，可采取掩码、块循环或随机方式在各个可视化节点的 DSM 区域中均衡分布数据块。

数值模拟产生的数据量很大，从计算节点向可视化节点迁移会出现较大的延迟，在计算节点上对数据进行压缩可以减少传输数据量，提升整体性能。原位处理不能占用太多资源，故需在压缩率和压缩解压速度方面取得平衡。Lakshminarasimhan 等提出了一种基于数据排序和 B 样条曲线拟合的误差控制有损原位压缩方法^[26]，对于有着时空维度的数据场，在相同的压缩误差精度内，压缩比远高于离散小波变换等有损压缩算法，压缩和解压开销较小，容易实现可扩展的并行算法。Schendel 等提出原位数据压缩方法 ISOBAR^[27]，以 N 个长度为 ω 字节的数据(如双精度数 ω 为 8)为一组划分数据集，每组数据可表示为 $\omega \times N$ 的矩阵，矩阵的元素就是相应字节的值，以列为单位利用分布函数考察可压缩性，形成可压缩和不可压缩数据子集，随后，根据用户选择的策略对可压缩子集进行压缩处理。

3 原位并行可视化算法

为了利用不断提升的高性能计算机能力，当前数值模拟已经普遍采用并行处理技术和方法，与之对应，原位并行可视化技术也成为研究热点。

原位并行可视化主要有三种基本并行处理模式，分别是任务并行、流水线并行以及数据并行。任务并行中，每一个进程独立执行一个任务，优点是相互独立的任务能够并行执行，缺点是当任务之间存在相关性时，并行性受到限制，负载均衡很难做好。流水线并行是将任务处理过程按照逻辑关系划分为多个阶段，通过使用多个线程并行执行不同阶段来加速处理过程。流水线并行方式能充分利用计算系统硬件资源，缺点是流水段之间的负载难于均衡，流水线的设计与具体应用密切相关。数据并行是将数据划分为数据子块并指派给多个线程，并整合各个线程任务将所产生的子结果。数据并行在

数据相关性较弱情况下能达到很高并行度，但可扩展性受限于整合过程中的进程间通信复杂度。

在设计原位并行可视化算法时，除考察具体的数值模拟应用特点、拟采用的可视化手段外，还必须考虑并行计算机体系结构的特点和发展趋势。即将出现的 E 级高性能计算机可能采用基于众核处理器、具有更深层次存储结构的异构并行体系结构^[28]，在这样的计算机上实现高效并行算法，须采用两种甚至三种基本并行处理模式混合的方式，才能取得较好的效果。此外，原位并行可视化还需在数据划分和组织管理、可扩展性方面与数值模拟良好匹配或保持一致，故一些类型的后处理并行可视化算法难以移植到原位模式下。

在原位体绘制算法方面，后处理模式的并行体绘制算法容易用于原位可视化，原因在于这些算法可以很好地适应数值模拟的数据区域划分方式，计算主要在数据块内执行，不同节点之间的通信量能够预测和控制。Ma 等、Yu 等分别将并行体绘制算法应用于大规模并行地震数值模拟^[3]、燃烧数值模拟中^[29]，由于可视化数据分区与数值模拟一致，故算法不需太多改动，图像合成也可沿用原有方式(2-3 交换等)，很好地控制了数据通信量。

在原位几何可视化算法(流线/面、迹线/面等)方面，后处理中常用的数据分区预处理^[30-31]、核外(out of core)方式^[32-33]等不适用于原位可视化，而分布式多核、众核计算平台上的并行几何可视化算法则具备改造为原位可视化算法的潜力。Camp 等提出了基于 MPI 和 OpenMP 的并行流线计算方法(图 3)^[34]。在采用种子点并行方式时，将 CPU 核心划分为两个集合，分别用于流线计算和 IO 操作，同时，建立活跃、不活跃和积分完成种子点队列，分别存储可在当前数据块上继续积分、不能继续积分和已达到积分结束条件的种子点集合。在 Cache 中有空间时，IO 线程载入不活跃种子点计算所需的数据块，将不活跃种子点中可以在新载入数据上进行积分的种子点放入活跃种子点集合中继续积分计算，直到所有种子点都移入积分完成种子点集

合中。采用数据并行方式时, 每个 CPU 中的一个核心负责数据通信, 剩余核心运行积分计算线程, 同样维持三个队列, 在积分无法在分配给本节点的数据块上进行时, 将种子点积分的最终位置通过 MPI 传送给相应的节点, 并将其放入不活跃种子点

集合。所有节点上均存储全局的积分结束种子点集合, 在一定的时间间隔后进行同步操作, 直到所有种子点完成积分。作者随后在 GPU 集群上实现了此算法^[35]。

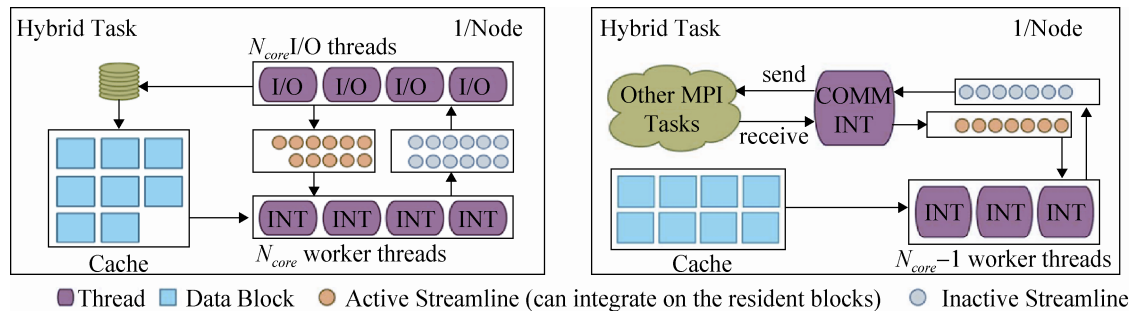


图 3 基于种子点(左)和数据(右)划分的 MPI/OpenMP 混合流线并行计算^[34]

在异构并行可视化方面, Hagan 等在多 GPU 集群上进行模拟和原位可视化计算^[36], 其思路是将 GPU 按任务划分为两个集合, 一部分用于模拟, 另一部分用于可视化和绘制。由于模拟和可视化计算速度不匹配, GPU 负载容易出现不平衡的问题, 在多体问题模拟的案例研究中, 作者根据实验结果研究了 GPU 负载平衡方法。Bethel 等对直接体绘制、流/迹线的 MPI 和 MPI/OpenMP 混合算法的强扩展性和弱扩展性进行了对比^[37], 认为在内存消耗、通信量、运行速度和可扩展性上, 混合并行算法在很大程度上优于 MPI 并行算法, 作者还考察了分段 IO 系统对可视化性能的影响。

4 原位可视化交互

原位可视化交互分为两个方面: 可视化人机交互、可视化进程/应用与数值模拟进程/应用之间的交互。可视化交互内容与后处理模式中的人机交互内容类似, 而进程/应用之间的交互主要用于数据和驾驭命令传输和同步。

在人机交互方面, 困难在于用户交互行为模式不确定, 很难固定操作流程, 且每次交互的时间长短不一, 难以与高性能计算机数值模拟作业的批处理运行方式匹配, 多用户协同和并发交互困难。

Kageyama 等提出了一种实现批处理并行计算结果交互式原位可视化的方法^[38], 在可视化空间区域不同位置布置足够数量的摄像机, 在原位生成可视化图像序列并压缩成视频文件, 通过特定的视频播放软件对这些视频进行交互的查看(包括转动角度、变换距离等)。Ahrens 等^[39]利用 ParaView 的脚本录制功能, 将用户使用图形界面设定的可视化流水线、摄像机参数等生成用于原位处理的 Python 脚本, 结合 Catalyst 组件, 生成可视化图像并存入数据库, 在需要交互时, 对库存图像进行查询、显示、合成。这类基于图像/视频的交互对数值模拟影响很小, 易于实现多用户并发, 但交互内容有限, 灵活性较差, 调整可视化算法后需要重新运行数值模拟和可视化应用, 只适合于预先可以确定可视化方式的问题。

另一类原位人机交互方法是基于数据的方法, 数值模拟产生数据后交由可视化流水线处理, 数值模拟按照交互要求等待或继续运行。Duque 等在实际与 CFD 数值模拟紧耦合原位可视化时^[9], 基于 FieldView 的图形化用户界面和 Python 脚本实现交互功能, 能获取数值模拟当前计算时间步数据进行交互可视化, 也能够按照用户设定的计算时间步间隔自动对相应的数据进行可视化处理和交互。Rivi

等使用支持原位可视化的开源软件 ParaView 和 Visit, 实现了原位可视化交互^[7]。在紧耦合模式下, 为了提高效率, 可视化流水线会在原位处理之前定义好^[40], 限制了交互内容, 且由于数值模拟和可视化处理速度不匹配, 需建立数据同步机制。

在多用户和远程人机交互方面, 孟创斌针对多用户远程交互的特点, 基于 ParaView 和 Web 服务器构建了一个高性能并行可视化服务器^[41], 支持基于共享会话方式的多用户协作机制, 通过会话管理对可视化会话进行统一管理和约束, 使用服务端的推送技术将可视化应用在多个用户之间进行实时同步, 通过浏览器在线调节可视化参数, 共享可视化结果。

在数值模拟和可视化进程/应用交互方面, ICARUS^[22]通过在内存文件的特殊位置建立存储驱动命令和数据的交互组(Interaction Group), 采用锁机制控制并发访问, 数值模拟进程通过 HDF5 接口获取用户的控制参数后即可按命令调整计算过程。使用 VisIt 实现原位可视化时, 通过注册回

调函数、在数值模拟代码中加入命令响应函数的方式实现简单的进程间交互(如停止、继续、步进等操作)^[7]。总的说来, 原位可视化中进程间交互大多限于对数值模拟的简单操控, 改变计算参数、网格或分区等复杂交互实现比较少见。

5 原位可视化集成框架与软件

原位可视化集成框架是数值模拟和可视化代码/软件的粘合剂, 由其提供数据通信、应用/进程同步机制, 完成数据转换管理, 并通过 API 屏蔽底层实现细节和分布式环境差异, 减小数值模拟和可视化应用代码的改动, 让用户能以尽可能小的代价获取灵活的原位可视化能力。

Cloud 等设计了一个松耦合分布式计算驾驭框架 EPSN (图 4)^[42], 每个计算和可视化节点上均运行一个 EPSN 端口线程, 响应请求和命令, 在模拟和可视化端, 各有一个代理进程, 负责向节点发布来自客户端的命令, 控制计算和可视化进程。计算和可视化节点之间的通信通过 CORBA 完成。

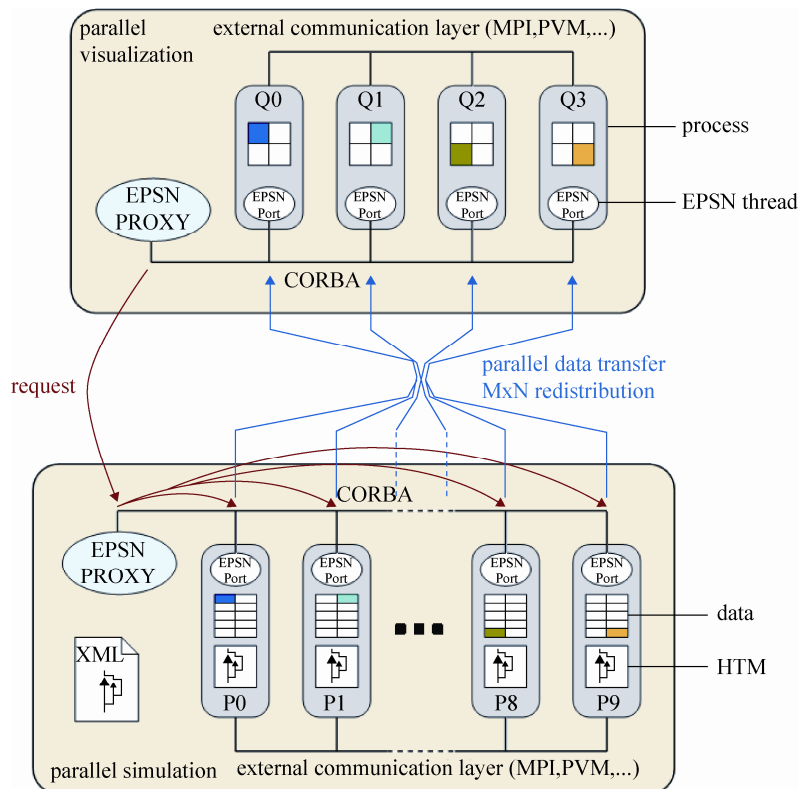


图 4 EPSN 框架^[42]

Zhang 等建立了一个分布式数据共享环境下的松耦合原位科学计算 workflow 框架(图 5)^[43], 通信层可根据数据位置的不同采用远程直接内存访问(RDMA)或共享内存方式完成数据访问操作; workflow 管理服务器根据不同功能软件之间的数据流关系建立应用间的通信图, 通过分析将需要通信的任务分配在同一个或网络距离接近的计算节点上, 减小总通信量和通信延迟; 相比紧耦合原位可视化方式, 此框架提供了更高的灵活性; 相比分级存储的数据共享和通信方式(如 DataSpace^[18]), 此框架不需在应用和分级存储节点之间复制传输数据, 通信开销小, 但在执行前需人工确定 workflow、数据分区和对应关系, 分析数据流向和指定处理节点。

Dorier 等提出一种紧耦合原位可视化框架^[44],

实现了数值模拟和原位可视化进程的内存共享, 支持分核模式(数值模拟和原位可视化计算使用不同 CPU 核心)和分时模式(数值模拟和可视化线程在相同的 CPU 核心上顺序执行)的原位可视化方式, 并在两个案例中分析了框架对模拟和可视化总体性能的影响以及可扩展性。

Biddiscombe 等建立的 ICARUS 框架^[22]核心是基于 HDF5 格式的 DSM 虚拟文件系统, 采用并行 MPI-IO 以及 RMA 方式实现了节点内、节点间双向通信, DSM 使用线性编址, 有掩码、块循环和随机三种数据重分布方式可以选择, 使用互斥锁控制共享资源的并发访问以保证数据一致性, 提供了 wait 和 free 两种模式(图 6)以满足不同应用场景的计算控制需求:

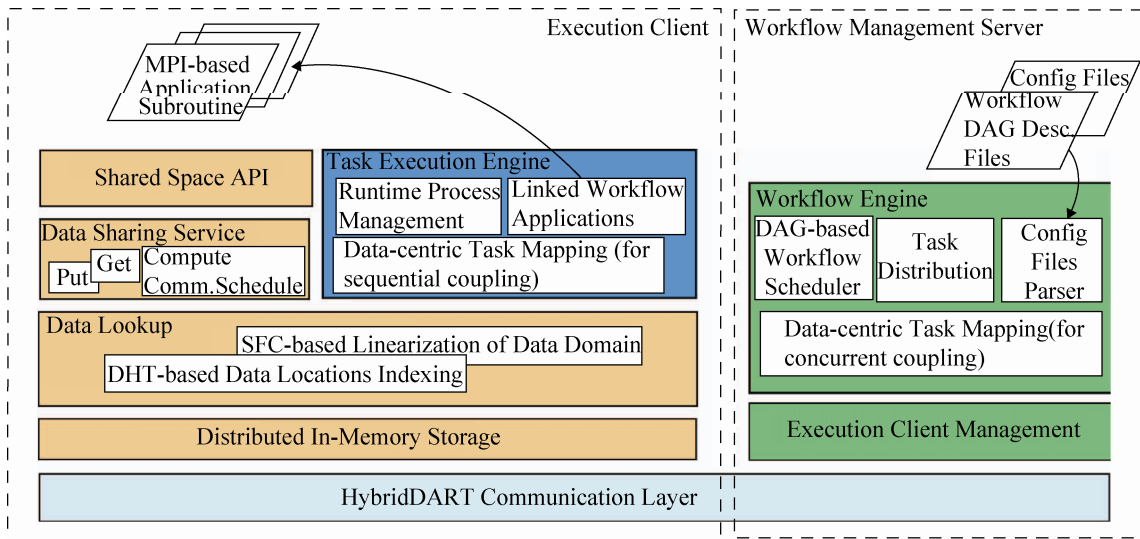


图 5 原位科学计算 workflow 框架^[43]

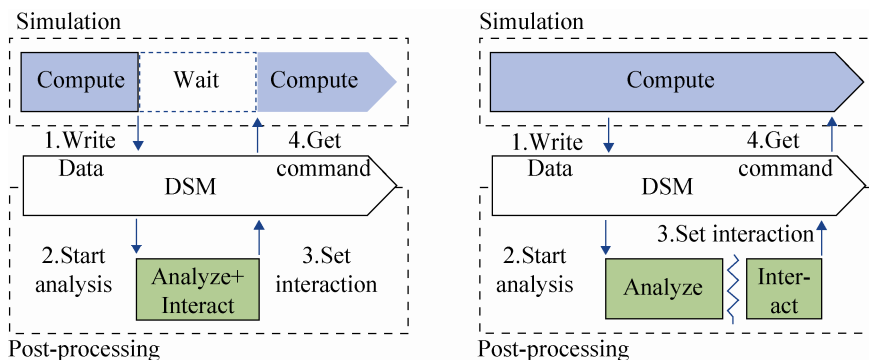


图 6 ICARUS 控制的 wait(左)和 free(右)模式^[25]

路石等^[45]根据原位可视化功能特点, 提出原位并行可视化服务器的概念, 并行可视化服务器由调度和管理结点、计算结点、绘制结点、数据结点和 I/O 结点构成, 可实现复杂的原位可视化用户服务功能。

支持原位可视化的软件有两类, 一类为针对某个具体数值模拟软件和特定的可视化应用开发的专用软件^[3,9], 另一类为通用可视化软件, 如 Visit、ParaView 等。第一类软件与应用场景耦合紧密, 优化时需要考虑的因素有限, 可以获得很好的性能, 但开发软件需要较多的应用领域和可视化知识和技能, 且通用性和扩展性不好; 第二类软件功能丰富, 通过插件和接口对软件进行扩展后可支持更多类型的数值模拟应用, 灵活性好, 易于在不同的硬件平台上实现原位可视化, 但性能调优不易。

Visit 是在美国能源部的资助下, 由劳伦斯利

物莫尔国家实验室开发的开源通用可视化软件, Visit 通过 libsim 实现紧耦合原位可视化^[11], 数值模拟应用需要在代码中适当位置调用 libsim 的 API 以实现原位可视化功能 (图 7)。数值模拟应用首先完成 libsim 初始化, 而后在主循环中监听来自 VisIt 的连接请求, 当收到连接请求并连接成功后, 数值模拟应用可接受来自 VisIt 的控制指令, 与 VisIt 共享内存数据则是以指针形式通过回调函数实现。

ParaView 常见的原位可视化插件有两个, 分别为 ICARUS^[22]和 Catalyst^[13-14]。Catalyst 向数值模拟应用提供 API, 通过这些 API 函数完成初始化, 数据信息传递、定义可视化流水线和同步等行为 (图 8), 与使用 VisIt libsim 类似, 数值模拟代码需要做相应的改动。

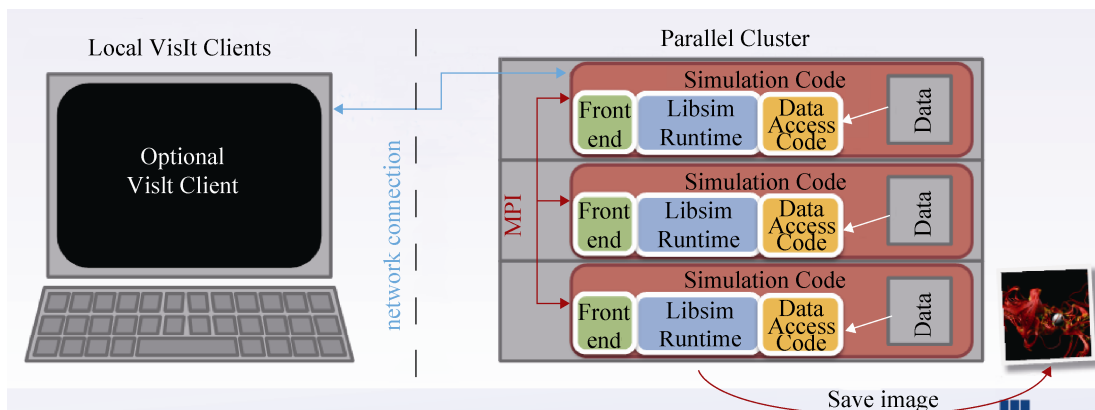


图 7 VisIt 原位可视化模式架构^[11]

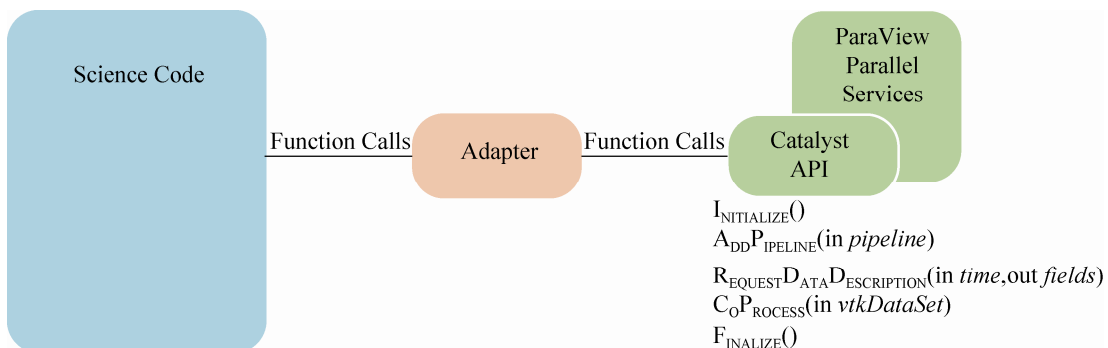


图 8 数值模拟与 Catalyst 通过适配器连接^[14]

6 原位可视化应用

通过研究人员的不断努力, 原位可视化技术发展迅速, 框架和软件可用性不断增强, 在诸多领域得到应用。

Rivi 等使用 ParaView、ICARUS 与天体物理学模拟软件 PLUTO 对年轻星体超音速轴对称磁流喷射计算数据进行了可视化计算和绘制^[7], 并实现了图形化的计算参数调整的计算驾驭功能; Whitlock 等使用 VisIt 和宇宙模拟软件 GADGET-2 在 216 个节点上, 实现了规模为 10 亿粒子数的数值模拟原位可视化和计算过程驾驭^[11]。

在 CFD 数值模拟原位可视化应用上, Yu 等建立了基于开源 CFD 直接数值模拟软件 S3D 的原位可视化系统^[29], 在 Cray XT5 计算机上使用最多 15360 个核心开展大规模燃烧计算和可视化工作; Fabian 等使用 ParaView、Catalyst 插件和 CFD 模拟软件 Phasta、CTH、S3D 实现了原位可视化^[46], 系统运行于 IBM BG/P 计算机上, 最多使用了 32 768 个计算核心, 采用了等值面、切面等手段表现模拟数据; Biddiscombe, Soumagne 等使用 ParaView 和插件 ICARUS^[24], 在 Cray XE6 的 1 536 个计算核心上实现了与 CFD SPH(Smoothed Particle Hydrodynamics)数值模拟应用的整合和原位可视化; Lorendeau 等^[40]使用 ParaView 和 Catalyst 插件、开源 CFD 解算器软件 SALOME, 在 Ivanoe 计算机的 16 584 个核心上实现了 2 亿量级网格的流场计算和原位可视化。

7 结论

当前原位可视化面临的挑战和问题体现在以下几方面:

一、大规模异构并行原位可视化算法研究方面。原位可视化并行算法必须与数值模拟规模相适应, 在并行数值模拟数据分区的基础上处理好扩展性和负载均衡问题, 且不能占用过多的计算和内存资源。主流可视化算法中, 等值线/面、体绘制等

易于实现原位并行可视化, 而数据相关性很强的几何可视化算法还未见理想的原位并行实现方式。

二、大规模分布式数据管理方面。在分布式异构的并行环境下, 数据存储层次更多, 节点内 CPU 与协处理器、CPU 与 CPU 之间, 以及节点间数据交换更加频繁, 设计能够支持多层次并行计算的数据结构, 建立合理的在线数据管理和访问机制, 在很大程度上影响着原位可视化的效率。

三、原位可视化交互方法研究方面。原位可视化时, 数值模拟任务一般采用批处理方式运行于高性能计算机上, 数据动态产生、变化(时变问题), 很难在保持交互灵活性的同时保持数值模拟的高效运行; 在对结果数据没有先验知识支持的研究类数值模拟中, 很难预定可视化算法和流水线以实现实时交互; 在多用户针对同一数据集的协同交互可视化, 以及高性能计算机上的多用户多数据集并发可视化交互上, 现有的框架和可视化软件还不能很好支持。

四、原位可视化集成框架和软件研究方面。原位可视化框架研究多集中在数据通信和管理方面, 对交互和多学科计算 workflow 支持不是特别充分, 框架在极大规模并行环境下的扩展性和效率有待验证; 主流可视化软件对原位可视化的支持程度不一, 需要适配器将数值模拟数据转换为可视化数据格式, 消耗了额外的计算和内存资源。

在未来的原位可视化研究应用中, 可能有如下一些趋势:

一、修改或重新设计数据相关性强的并行可视化算法。尽可能消除数据依赖, 减少节点间通信和同步等待开销, 充分利用众核 CPU 和协处理器多线程的并行计算能力。

二、构造一个更全面的应用框架。传统数值模拟的前处理-解算-可视化后处理顺序执行模式已逐渐改变, 各个环节之间已无特别清晰的界限, 各类软件之间的交互增多, 控制也不再以解算为中心, 未来的框架应全面考虑数值模拟各组成部分之间的联系、交互和控制, 在分布式环境下建立统一

的多级存储的数据管理和通信机制、建立可配置的计算 workflow 引擎, 建立丰富的交互手段以实现全过程驾驭; 与计算机资源调度管理软件相结合, 实现动态调整和分配计算及可视化资源; 支持批处理和实时交互两种运行方式; 支持松耦合、紧耦合和混合的原位可视化模式; 提供应用管理机制, 能够集成多种不同类型的前置、解算和可视化分析软件或代码; 支持十万核以上的并行应用。

三、在人机交互和可视化软件研究上与领域专家建立紧密合作关系。原位可视化与应用的具体领域和问题有明显相关性, 人机交互和可视化软件是直接面向最终领域用户的, 必须加强联系与合作。将领域专家的先验知识固化在预处理和特征提取等算法中, 对不同问题的数据可视化和分析流程分类, 建立可视化流水线模板, 丰富软件功能, 建立支持协同和多用户的人机交互机制和系统。

原位可视化技术多学科交叉特性明显, 需在研究中关注计算机和应用领域诸多专业的发展趋势。在现有计算机体系结构不发生重大变化情况下, 固有的 IO 瓶颈决定了原位可视化是最可行的大规模并行数值模拟数据可视化分析技术, 现实的需求将推动原位可视化技术加速进步, 应用的广度和深度也会随之不断拓展。

参考文献:

- [1] McCormick B H, Defanti T A, Brown M D. Visualization in scientific computing [J]. *Acm Sigbio Newsletter* (S0163-5697), 1988, 10(1): 15-21.
- [2] Wong P C, Shen H-W, Johnson C R, *et al.* The Top 10 Challenges in Extreme-Scale Visual Analytics [J]. *IEEE Computer Graphics and Applications* (S0272-1716), 2012, 32(4): 63-67.
- [3] Ma K-L, Wang C, Yu H, *et al.* In-situ processing and visualization for ultrascale simulations [J]. *Journal of Physics: Conference Series* (S1742-6588), 2007, 78: 012043
- [4] 李思昆, 蔡勋, 王文珂, 等. 大规模流场科学计算可视化 [M]. 北京: 国防工业出版社, 2013.
- [5] Ma K L. In Situ Visualization at Extreme Scale: Challenges and Opportunities [J]. *IEEE Computer Graphics and Applications* (S0272-1716), 2009, 29(6): 14-19.
- [6] 单桂华, 田东, 谢茂金, 等. 千万亿次科学计算的原位可视化 [J]. *计算机辅助设计与图形学学报*, 2013, 25(3): 286-293.
- [7] Rivi M, Calori L, Muscianisi G, *et al.* In-situ Visualization: State-of-the-art and Some Use Cases [J]. PRACE White Paper, 2012
- [8] Tu T, Yu H, Ramirez-Guzman L, *et al.* From mesh generation to scientific visualization — an end-to-end approach to parallel supercomputing [C]// SC2006. Tampa, FL, USA: IEEE, 2006: 12.
- [9] Duque E P, Hiepler D, Stone C P, *et al.* IFDT-Intelligent In-Situ Feature Detection, Extraction, Tracking and Visualization for Turbulent Flow Simulations [C]// ICCFD7. Hawaii, USA: ICCFD, 2012: 7.
- [10] VisIt <https://wci.llnl.gov/simulation/computer-codes/visit/>
- [11] Whitlock B, Favre J M, Meredith J S. Parallel In Situ Coupling of Simulation with a Fully Featured Visualization System [C]// Eurographics Symposium on Parallel Graphics and Visualization, 2011. Llandudno, UK: Eurographics Association, 2011: 101-109
- [12] Paraview. <http://www.paraview.org/>
- [13] Fabian N, Moreland K, Thompson D, *et al.* The ParaView coprocessing library: A scalable, general purpose in situ visualization library [C]// Large Data Analysis and Visualization, 2011 IEEE Symposium on. USA: IEEE, 2011: 89-96.
- [14] Rogers D, Moreland K D, Oldfield R A, *et al.* Data co-processing for extreme scale analysis level II ASC milestone (4745) [R]. Albuquerque: Sandia National Laboratories, 2013
- [15] Vtk <http://www.kitware.com/opensource/vtk.html>.
- [16] Zheng F, Klasky S, Podhorszki N, *et al.* FlexIO: I/O Middleware for Location-Flexible Scientific Data Analytics [C]// Parallel and Distributed Processing, 2013 IEEE 27th International Symposium on. USA: IEEE, 2013: 320-331.
- [17] Lofstead J F, Klasky S, Schwan K, *et al.* Flexible IO and integration for scientific codes through the adaptable IO system (ADIOS) [C]// Challenges of Large Applications in Distributed Environments, 2008. Boston, MA, USA: ACM, 2008: 15-24
- [18] Docan C, Parashar M, Klasky S. DataSpaces: an interaction and coordination framework for coupled simulation workflows [J]. *Cluster Computing* (S1386-7857), 2012, 15(2): 163-181.
- [19] Docan C, Parashar M, Cummings J, *et al.* Moving the Code to the Data - Dynamic Code Deployment Using ActiveSpaces [C]// Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International. USA: IEEE, 2011: 758-769.
- [20] Abbasi H, Wolf M, Eisenhauer G, *et al.* DataStager: scalable data staging services for petascale applications [J]. *Cluster Computing* (S1386-7857), 2010, 13(3): 277-290.
- [21] Jerome Soumagne, John Biddiscombe, Clarke J. An HDF5 MPI Virtual File Driver for Parallel In-situ

- Post-processing [J]. Lecture Notes in Computer Science (S0302-9743), 2010, 6305: 62-71.
- [22] Biddiscombe J, Soumagne J, Oger G, *et al.* Parallel Computational Steering and Analysis for HPC Applications using a ParaView Interface and the HDF5 DSM Virtual File Driver [C]. Eurographics Symposium on Parallel Graphics and Visualization, 2011. Llandudno: Eurographics Association, 2011: 91-100
- [23] Soumagne J, Biddiscombe J, Esnard A. Data redistribution using one-sided transfers to in-memory HDF5 files [J]. Lecture Notes in Computer Science (S0302-9743), 2011, 6960: 198-207.
- [24] Biddiscombe J, Soumagne J, Oger G, *et al.* Parallel Computational Steering for HPC Applications Using HDF5 Files in Distributed Shared Memory [J]. IEEE Transactions on Visualization and Computer Graphics (S1077-2626), 2012, 18(6): 852-864.
- [25] Soumagne J. An In-situ Visualization Approach for Parallel Coupling and Steering of Simulations through Distributed Shared Memory Files [D]. France: Université Sciences et Technologies - Bordeaux I, 2012.
- [26] Lakshminarasimhan S, Shah N, Ethier S, *et al.* ISABELA for effective in situ compression of scientific data [J]. Concurrency and Computation: Practice and Experience (S1532-0634), 2013, 25(4): 524-540.
- [27] Schendel E R, Pendse S V, Jenkins J, *et al.* ISOBAR hybrid compression-I/O interleaving for large-scale parallel I/O optimization [C]// Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing. USA: ACM, 2012.
- [28] Kogge P M. ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems [M]. Darpa Information Processing Techniques Office, 2008.
- [29] H Y, C W, R W G, *et al.* In Situ Visualization for Large-Scale Combustion Simulations [J]. IEEE Computer Graphics and Applications (S0272-1716), 2010, 30(3): 45-57.
- [30] Guo H, Zhang J, Liu R, *et al.* Advection-Based Sparse Data Management for Visualizing Unsteady Flow [J]. IEEE Transactions on Visualization and Computer Graphics (S1077-2626), 2014, (1): 2555-2564.
- [31] Nouanesengsy B, Lee T-Y, Shen H-W. Load-balanced parallel streamline generation on large scale vector fields [J]. IEEE Transactions on Visualization and Computer Graphics, (S1077-2626) 2011, 17(12): 1785-1794.
- [32] Chen C-M, Nouanesengsy B, Lee T-Y, *et al.* Flow-guided file layout for out-of-core pathline computation [C]// Large Data Analysis and Visualization, 2012 IEEE Symposium on. USA: IEEE, 2012: 109-112.
- [33] Chen C-M, Xu L, Lee T, *et al.* A flow-guided file layout for out-of-core streamline computation [C]// Pacific Visualization Symposium, 2012 IEEE. USA: IEEE, 2012: 145-152.
- [34] Camp D, Garth C, Childs H, *et al.* Streamline integration using MPI-hybrid parallelism on a large multicore architecture [J]. IEEE Transactions on Visualization and Computer Graphics (S1077-2626), 2011, 17(11): 1702-1713.
- [35] Camp D. GPU Acceleration of Particle Advection Workloads in a Parallel, Distributed Memory Setting [C]// Proceedings of the 13th Eurographics Symposium on Parallel Graphics and Visualization, 2013. Girona: Eurographics Association, 2013: 1-8.
- [36] Hagan R, Cao Y. Multi-gpu load balancing for in-situ visualization [C]// The 2011 International Conference on Parallel and Distributed Processing Techniques and Applications, 2011. San Diego, USA: UCMSS, 2011.
- [37] Bethel E. Towards Exascale: High Performance Visualization and Analytics-Project Status Report [R]. Portland: DOE 2012 Exascale Research Conference and Workshop, 2013.
- [38] Kageyama A, Yamada T. An approach to exascale visualization: Interactive viewing of in-situ visualization [J]. Computer Physics Communications (S0010-4655), 2014, 185(1): 79-85.
- [39] Ahrens J, Jourdain S, O'leary P, *et al.* An image-based approach to extreme scale in situ visualization and analysis [C]// High Performance Computing, Networking, Storage and Analysis, SC14: International Conference for. IEEE, 2014. USA: IEEE, 2014.
- [40] Lorendeau B, Fournier Y, Ribes A. In-Situ visualization in fluid mechanics using Catalyst: A case study for Code Saturne [C]// 2013 IEEE Symposium on Large-Scale Data Analysis and Visualization. USA: IEEE Computer Society, 2013: 53-57.
- [41] 孟创斌. 高性能计算数据的远程交互可视化技术研究与实现 [D]. 长沙:国防科技大学, 2014.
- [42] Coulaud O, Richart N, Esnard A. A Steering Environment for Online Parallel Visualization of Legacy Parallel Simulations [C]// Distributed Simulation and Real Time Applications, IEEE/ACM International Symposium on. USA: IEEE Computer Society, 2006: 7-14.
- [43] Fan Z, Docan C, Parashar M, *et al.* Enabling In-situ Execution of Coupled Scientific Workflow on Multi-core Platform [C]// Parallel and Distributed Processing Symposium, International. USA: IEEE, 2012: 1352-1363.
- [44] Dorier M, Sisneros R, Peterka T, *et al.* Damaris/Viz: A nonintrusive, adaptable and user-friendly in situ visualization framework [C]// Large-Scale Data Analysis and Visualization, 2013 IEEE Symposium on. USA: IEEE, 2013: 67-75.
- [45] 路石, 孟创斌, 李思昆, 等. 高性能并行可视化服务器的资源管理技术研究 [J]. 软件工程与应用, 2014, 3(5): 131-143.
- [46] Fabian N, Moreland K, Thompson D, *et al.* The ParaView Coprocessing Library: A scalable, general purpose in situ visualization library [C]// Large Data Analysis and Visualization, 2011 IEEE Symposium on. USA: IEEE, 2011: 89-96.