



基于超算的多模式计算融合支撑系统*

卢宇彤, 陈志广

中山大学计算机学院, 广东 广州 510006

摘要: 复杂的科学与工程计算应用要求在数值模拟、大数据处理、人工智能3种计算模式之间实现融合, 而这3种计算模式具有不同的负载特征, 它们在执行、调度、数据访问方面具有显著的差异, 传统的超级计算机不能同时高效地支撑以上3种计算模式。我们重构了超级计算机的并行文件系统、并行通信系统、资源管理与作业调度系统等系统软件, 并设计了基于超算的大数据处理框架和人工智能推理框架, 支持在高性能计算应用中融合大数据和人工智能计算模式, 形成了基于超算环境的多模式计算融合支撑系统。应用表明, 所研发的融合支撑系统能够支持3种计算模式的耦合, 且在性能上表现出显著的优势, 为复杂的科学与工程计算应用提供了完善的运行环境。

关键词: 多模式融合计算; 超级计算机; 科学与工程计算

中图分类号: TP393 **文献标志码:** A **文章编号:** 2097-0137(2024)06-0150-11

The convergent computing based on supercomputer

LU Yutong, CHEN Zhiguang

School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

Abstract: Complex science and engineering computing applications require the converging of three computing modes: Numerical simulation, big data processing, and artificial intelligence. However, traditional supercomputers cannot efficiently support the above three computing modes at the same time. We reconstructed the supercomputer's parallel file system, parallel communication system, resource management and job scheduling system etc., and further designed a supercomputer-based big data processing framework and artificial intelligence inference framework, building a converged environment based on supercomputers. Experiments show that the proposed converged system can support the coupling of three computing paradigms as well as improve the performance significantly, providing a comprehensive environment for complex scientific engineering computing applications.

Key words: converged computing; supercomputer; science and engineering computing

高性能计算与理论和实验一起, 并称为科学与工程研究的三大手段, 是一个国家综合国力和科技创新力的重要标志。长期以来, 高性能计算领域主要通过数值模拟来解决科学与工程计算问题。近年来, 随着各领域应用的不断演进, 复杂业务应用不再是单一的数值模拟, 数值计算、大数据

处理、人工智能分析的融合是主流的发展趋势。以气象预测为例, 传统的预测方法以气压计、温度计、雷达等设备采集的原始数据经同化后作为初值, 通过数值运算模拟出气象状况的变化过程。随着气象部门积累的历史数据越来越多, 相关从业人员可通过近期气象状况与历史数据中某个片

* 收稿日期: 2024-10-01

录用日期: 2024-10-09

网络首发日期: 2024-10-22

基金项目: 广东省基础与应用基础研究重大项目(2019B030302002);

国家自然科学基金(62461146204)

作者简介: 卢宇彤(1969年生), 女; 研究方向: 高性能计算与超级计算机;

E-mail: luyutong@mail.sysu.edu.cn

全文阅读



ZR20240293

段之间的模式匹配来预测天气, 甚至利用历史数据训练出人工智能模型来做气象预测, 这种基于大数据的气象预测方法已成为传统方法的重要补充。在理化基础研究领域, 基于密度泛函理论(DFT, density functional theory)的第一性原理数值计算已成为物理学、材料科学、化学和生物学等科学研究不可或缺的一部分, 然而该方法计算耗时长, 很难扩展到数万个原子规模的分子模拟计算。相比之下, 基于人工智能技术的第一性原理替代模型已取得进展。总之, 人工智能、大数据技术已广泛地运用到传统的高性能计算应用中。

复杂应用要求将数值模拟、大数据处理、人工智能分析3种计算模式融合到同一套计算系统中。一般认为, 传统面向大数据的分布式集群难以高效支持数值模拟和人工智能训练, 但是面向数值模拟的超级计算机经优化后可支持大数据处理。因此, 基于超级计算机构建融合应用支撑系统是可行的解决方案。然而, 三者的融合在数据存储、作业调度、应用框架等方面仍然存在严峻的挑战, 大数据处理和人工智能应用产生的海量文件将对超级计算机的全局并行文件系统形成巨大的压力; 在融合平台上实现数值模拟、大数据处理和人工智能作业的混合调度同时面临机遇和挑战; 当前的大数据、人工智能计算环境并没有充分发挥超级计算机的性能优势。

针对以上需求和挑战, 本文首先从存储、通信和调度3个方面重构了超算系统软件, 并依此优化大数据处理、人工智能训练推理运行时系统, 实现基于超算的多模式计算融合支撑系统。

1 面向多模式计算的系统软件重构与优化

为了高效地支持数值模拟、大数据处理、人工智能等多种计算模式的融合, 需要在数据存储、通信与调度、大数据处理框架、人工智能训练与推理等多方面重构当前的系统软件。

1.1 面向融合应用的大规模并行文件系统

基于超算的计算融合系统将不同计算范式集成为一个统一的环境中, 以最大化资源利用率和计算效率。这种融合能够同时支持传统数值模拟、数据密集型计算和人工智能应用, 实现更强大的协同处理能力。然而, 在计算融合系统中, 底层并行文件系统面临着新的挑战。传统的并行文件

系统主要针对大文件数据优化, 而在大数据处理和AI应用中, 生成和操作大量小文件的场景变得尤为普遍, 而传统并行文件系统在处理大量小文件时元数据极易成为主要的性能瓶颈, 从而影响系统的整体性能。在元数据节点内, 传统并行文件系统使用本地文件进行元数据存取, 增加了额外的软件栈开销, 并且无法有效发挥新型存储设备(如Nvme SSD、NVM等)的硬件性能; 在多个元数据节点间, 传统的元数据分布方法没有针对大目录做优化, 在应对包含大量小文件的大目录场景下极易出现单点瓶颈的问题。为应对以上问题, 本文从纵向单节点内和横向多节点间两个方面实现对并行文件系统元数据管理的优化, 以提升并行文件系统的小文件处理性能。

在单个元数据节点内, 传统并行文件系统普遍使用本地文件系统(如ext4、xfs等)进行元数据存取, 但这种方法会带来额外的软件开销, 不能有效地发挥存储设备性能。为减少额外软件开销, 并且更好地管理异构存储设备, 我们设计了一个扁平化的键值系统进行高效的元数据存取。其整体架构如图1所示, 键值系统包含持久化域和内存域, 持久化域将存储设备按不同块大小分成多个块组, 每个块组内的块大小固定, 并且每个块组内部基于日志追加的方式进行元数据写入。内存域主要由内存哈希表和内存缓冲区组成, 内存哈希表基于inode号直接哈希索引到对应元数据在持久化设备上的物理地址, 实现快速的元数据直接读取。内存缓冲区主要用于写入优化, 缓冲区分为多个组并与持久化设备上的块组一一对应, 并且每个组都包含active和read-only两个缓冲区, 两种缓冲区相互转化。扁平化键值系统的设计缩短了传统元数据在存储设备上的读取路径, 有效发挥存储设备性能, 多块组和内存缓冲区设计也有效提高了元数据的并发写入性能。

在多个元数据节点间, 传统并行文件系统主要静态哈希(如Lustre)或子树划分方式(如Ceph、BeeGFS)进行多节点元数据分布, 但这些方法都没有很好地应对单个大目录多节点分布的问题。我们设计了基于虚拟目录的目录分布方法, 其底层思想是在文件系统层面对大目录引入对用户透明的虚拟目录以进行划分, 将原本存储在单个目录中的子文件/目录划分到多个虚拟目录中, 并且虚拟目录基于哈希方法在多个元数据节点间进行均匀分布, 实现大目录在多个元数据节点间的

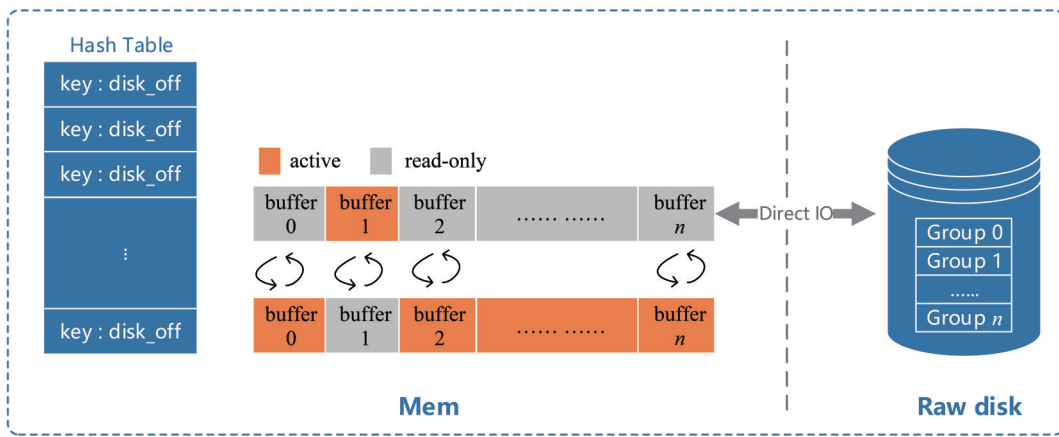


图1 扁平化的键值存储系统

Fig. 1 The flat architecture of key-value store

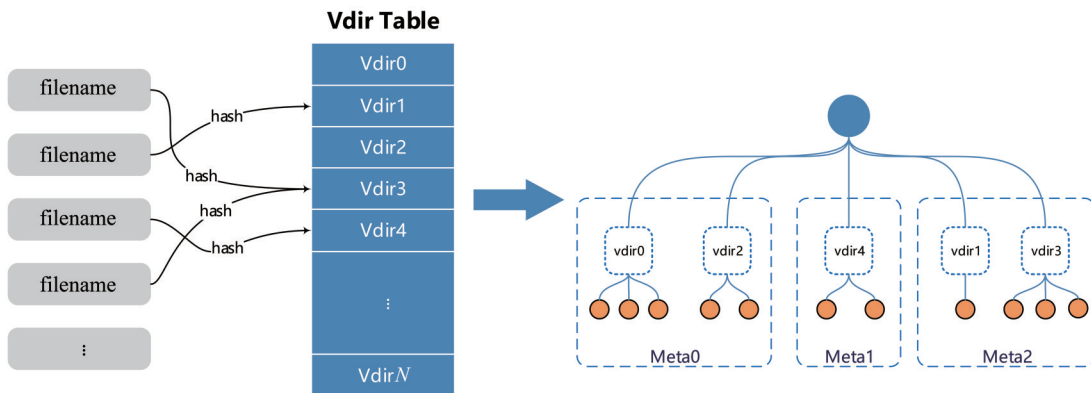


图2 基于虚拟目录的目录分布方法

Fig. 2 The directories striping based on virtual directory

高效组织与分布。其整体架构如图2所示，对于大目录，系统通过一致性哈希算法划分为多个虚拟目录，虚拟目录均匀分布在多个元数据节点上，文件名通过哈希可直接索引到对应的虚拟目录并进行相应的元数据操作。系统采用一致性哈希算法进行虚拟目录分布，可有效应对元数据节点和虚拟目录数量的伸缩问题。同时，得益于虚拟目录细粒度的划分方式，系统可实现高并发的元数据操作，以提高文件系统的元数据扩展性能。

在数据读写方面，我们提出了小文件数据内联存储与数据条带化融合的数据管理策略，以兼顾不同大小的文件数据读写请求。小文件数据内联存储策略的核心思想是将小文件数据存储于文件元数据中，配合元数据键值存储系统进行高效存取，这样可以通过单次RPC请求同时读取小文件元数据和数据，大大减少小文件访问延迟。通过设定小文件大小阈值，将阈值内的数据存储于

文件元数据中，超过小文件大小阈值的，则使用条带化方式分块分布于存储集群上，因此大文件场景下文件系统可以同时聚合多个存储节点的存储资源，实现高效并发读写。同时，为进一步提升大文件数据写效率，我们在客户端设计了多缓冲区的数据聚合写策略，将同个文件短时连续的小数据写请求聚合成一个大写请求，以减少RPC请求次数，提升文件数据写性能。基于以上多方面的技术，我们研发了面向融合应用的并行文件系统Kapok，能够有效应对多种计算模式的复杂I/O负载。

1.2 面向融合应用的通信与调度优化

在融合计算系统中网络需要同时承载多种不同应用类型的网络流量，多种不同的流量模式叠加在同一网络中会导致流量间的相互干扰，从而延长流完成时间，导致应用性能降低，其主要原因在于应用间链路竞争、应用内拓扑错配、小数据包延迟瓶颈等3个方面。

针对链路竞争, 我们提出了一种用于可重构网络的流量调度方法来解决应用间带宽竞争的瓶颈, 将一些大规模计算作业或者多个相关的作业部署在若干个网络组中, 并使之独占部分的网络组间可重构链路, 且将这部分的可重构链路配置为与其流量模式相匹配的逻辑拓扑。这样不但可以使重负载作业的网络流量之间形成隔离, 还可以增加带宽瓶颈处的带宽。针对应用内的拓扑错配瓶颈, 我们对常用的 3 种经典的聚合通信算法 Recursive Doubling、Bruck、Rabenseifner 算法进行了优化。我们为这些算法增加了额外的一步数据重排操作, 使节点之间优先与拓扑距离最近的通信对象进行大量的通信, 而与距离较远的通信对象进行小数据量的通信。通过这种方法, 可以减少网络中的负载, 从而减少流量间发生冲突的风险。为了尽可能避免小数据包的延迟瓶颈, 我们设计了免同步通信方法。在传统的远程直接内存访问操作中, 需要进行前同步与后同步操作, 其中前同步用于保证通信对象已就绪, 后同步则用于通知通信对象通信已完成, 这两种同步操作都需要使用到小数据包进行通信, 而这种小数据包通信很容易成为通信的瓶颈。针对前同步问题, 我们提出了一种将前同步隐藏在上一次通信中的方法, 这样可以免去一次同步时间开销; 针对后同步问题, 我们设计了通知式远程直接内存访问, 通过调用底层网络接口, 在通信完成时可以直接通知通信双方通信已完成, 无需额外的后同步操作, 其实现机制如图 3 所示。通过这两种优化方法, 我们可以实现完全地免同步通信, 从而避免其中不必要的小数据包通信。

数值模拟、人工智能训练等作业类型和大数

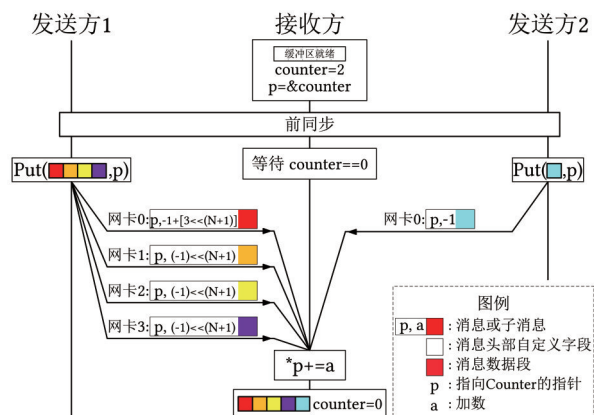


图 3 通知式远程直接内存访问-多网卡多消息聚合机制

Fig. 3 Notification-based remote direct memory access-multi-NIC multi-message aggregation mechanism

据处理作业在执行模式上存在较大的差异。数值模拟作业一般处理的数据量较少, 作业调度时对网络通信的要求不高; 属于同一作业的进程之间很可能存在同步操作, 这些进程必须一起分发到计算节点; 计算作业普遍对实时性要求不高, 能够容忍排队等待; 作业计算量较大, 一旦作业启动将运行很长时间。相比之下, 大数据处理作业可能面临很大的数据集, 在调度作业时要考虑数据的迁移; 整个作业可能由几个存在依赖关系的阶段组成, 由于无法准确估计每个阶段的计算量, 调度系统在分发任务时很难做出完美的决策; 一些处理任务具有实时性, 作业调度有优先级的需求; 每个作业运行的时间长短不一, 且差别很大。除此之外, 数值模拟计算任务通常在物理机资源上调度执行, 而多模式计算任务需要在物理机、虚拟机、容器等不同的环境下执行。

针对以上需求, 我们提出层次式的资源管理框架, 支持将大规模系统中的物理资源动态构建不同虚拟化方式的资源池, 并通过通信路径智能选择、低干扰共享访问等机制优化虚拟化环境构建以便充分发挥超算硬件性能; 突破了多类型资源融合调度技术, 通过对多模式计算负载的动态监测以及对资源需求的智能推断自动调整虚实资源池, 消解竞争冲突, 提高虚实资源统一调度效率, 增强系统资源对应用作业变化的适应性; 提出了基于主动消费和被动接收相结合的计算节点调度算法, 对高性能计算作业采用被动模式, 由调度节点分配作业给计算节点, 对大数据处理作业采用主动消费模式, 允许计算节点主动请求队列中的作业, 在提高资源利用率的同时降低调度主节点的计算开销; 提出网络拓扑感知的作业调度方法, 提升应用作业在运行过程中的通信效率。

1.3 基于超算的大数据处理框架

随着超算技术的不断进步, 高性能设备和应用类型的多样化对超算系统的大数据处理能力和融合计算能力提出了更高的要求(Mei et al., 2018; Yin et al., 2022; 梅宏等, 2023)。传统大数据处理框架如 Hadoop、Spark 等的设计初衷是面向分布式的商业集群, 在超算上高效部署和运行大数据处理框架仍存在着两个方面的不匹配。第一个方面是主流框架中松散耦合的并行任务与紧密耦合的科学计算任务不匹配, 过于简化的并行机制和编程接口导致科学计算应用开发成本高、运

行性能低。第二个方面是传统框架的资源管理和 I/O 机制与国产超算体系结构不匹配，缺乏面向尖端众核处理器、超算网络特征的针对性优化，导致系统硬件资源的利用率不够。为了解决上述不匹配问题，我们基于国产超算系统提出了自主研发的通用大数据处理框架和一系列优化支撑技术。

我们首先提出了一个面向高维有拓扑科学大数据的通用处理框架 Topo。拓扑数据处理是科学计算领域的一种新兴方法，主要是利用拓扑结构和空间关系来分析和处理高维数据。Topo 创新地

提出了拓扑感知的高效紧耦合并行机制，使得同一计算阶段内具有拓扑关系的各个工作线程能够进行高效数据交换，而不用等待全局数据同步。如图 4 所示，在同一个计算阶段内，并行任务可以根据拓扑结构信息进行消息传递，并在框架中实现自动并行的复杂科学计算应用。Topo 还提供了具有拓扑结构感知能力的简易用户编程接口，包括 3 个 InterMap 操作和 29 个通用大数据处理操作，其中 InterMap 操作实现了通过拓扑结构信息获取空间相邻工作线程的数据，并进行相应的拓扑数据处理。

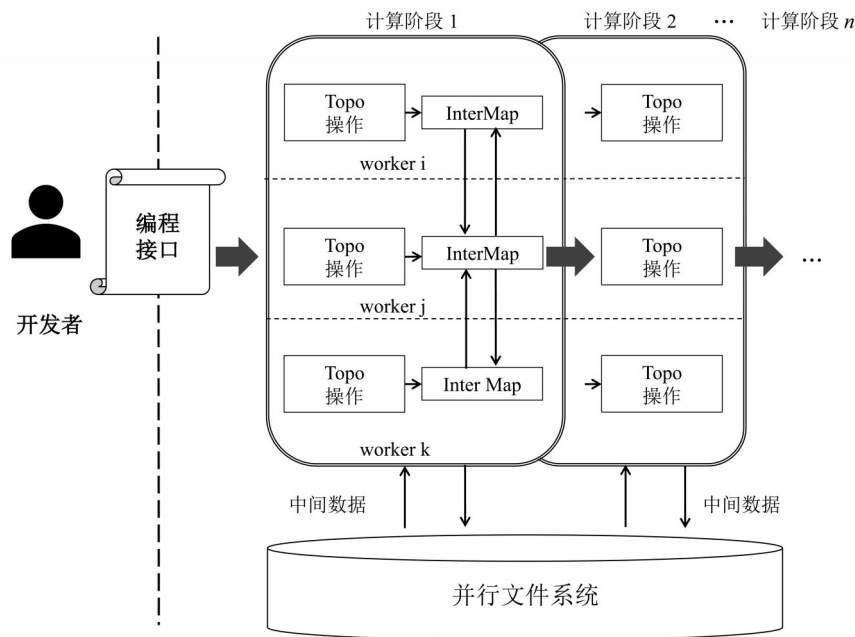


图 4 拓扑感知的大数据处理框架总体架构

Fig. 4 The overall architecture of topology-aware big data processing framework

为了进一步提升基于国产众核超算系统的大数据处理性能，我们还设计了多层次的体系结构优化支撑技术。在处理器层面，针对国产超算中的新兴众核处理器，我们设计了一个细粒度的动态任务调度系统 FinD，图 5 描绘了该系统的基本架构。FinD 通过可扩展的轻量级线程执行模型 (Seo et al., 2018) 实现了细粒度的任务调度和管理，从而显著减少众核处理器中的远程内存访问，并提出了一个双边 NUMA 感知的任务窃取机制，高效实现了众核处理器上的动态负载均衡。

在数据读写层面，我们基于超算的并行文件系统，设计了面向海量中间数据的聚合 I/O 机制，以改进大数据框架中大量零散临时文件对元数据服务造成压力的难题。如图 6 所示，我们设计的框架会

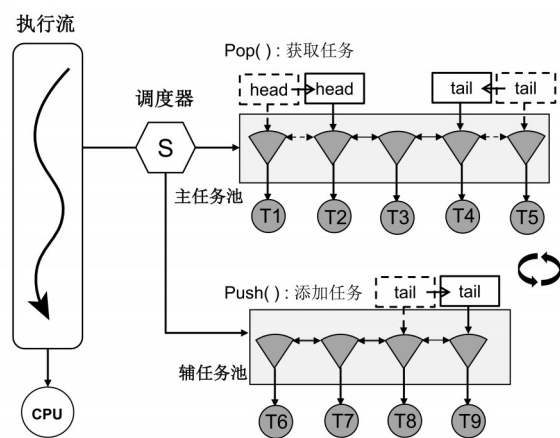


图 5 NUMA 感知的细粒度任务调度系统

Fig. 5 The NUMA-aware fine-grained task scheduling system

在并行文件系统上创建一个空的共享 MPI_FILE 文件, 用于存储计算阶段内所有工作进程的中间数据。每个 MPI 进程可以通过独立的偏移量将其中间数据并行地写入 MPI_FILE 文件的一个独立区域。通过设立缓冲区, 实现了对 MPI_FILE 文件的动态读写以及多个 MPI 进程间的并发 I/O。该技术充分利用了超级计算系统中高速网络的优势, 并大大减轻了元数据服务器的负担。

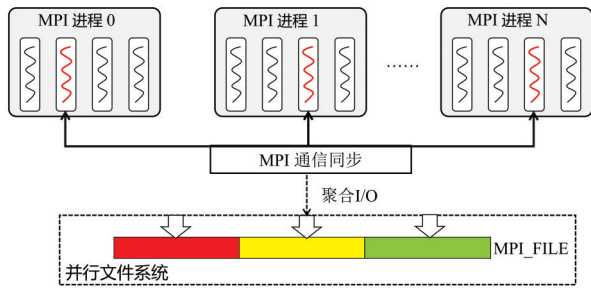


图 6 基于并行文件系统的中间数据聚合 I/O
Fig. 6 The aggregated I/O of intermediate data based on parallel file systems

1.4 人工智能推理并行优化

超智融合应用主要通过数值模拟与人工智能推理的融合来实现。人工智能推理由于庞大的计

算与内存需求, 并行推理已成为服务部署的必要手段。然而, 一方面, 张量并行(Shoebvi et al., 2019; Du et al., 2024)推理存在大量的数据同步需求, 频繁的通信引入大量开销; 另一方面, 流水线并行(Huang et al., 2019; Wang et al., 2024)推理受限于自回归推理复杂的运行流程, 存在大量空泡的问题, 这些问题导致现有并行推理系统的硬件利用率不高, 推理成本居高不下。

为了实现大模型高效并行推理, 我们提出了全新的时间维度分离式流水线并行推理架构 TD-Pipe。流水线并行推理将模型按层进行切分, 因其较小的通信需求而极具潜力, 然而大模型推理 Prefill(预填充)与 Decode(解码)阶段的负载不均衡与复杂的数据依赖使得实现紧凑的调度十分困难, 产生大量气泡。如图 7 所示, TD-Pipe 根据 Prefill 和 Decode 两步骤任务计算与依赖的不同特性, 将它们在时间维度上尽可能进行分离。Prefill 阶段扮演生产者的角色, 负责不断将请求填充入显存中, Decode 阶段扮演消费者的角色, 负责完成大模型解码工作并完成请求。通过时间维度上的分离, 这两种不同的计算模式能够避免相互干扰, 减少了 Prefill-Decode 切换产生的流水线气泡, 实现整体推理性能优化。

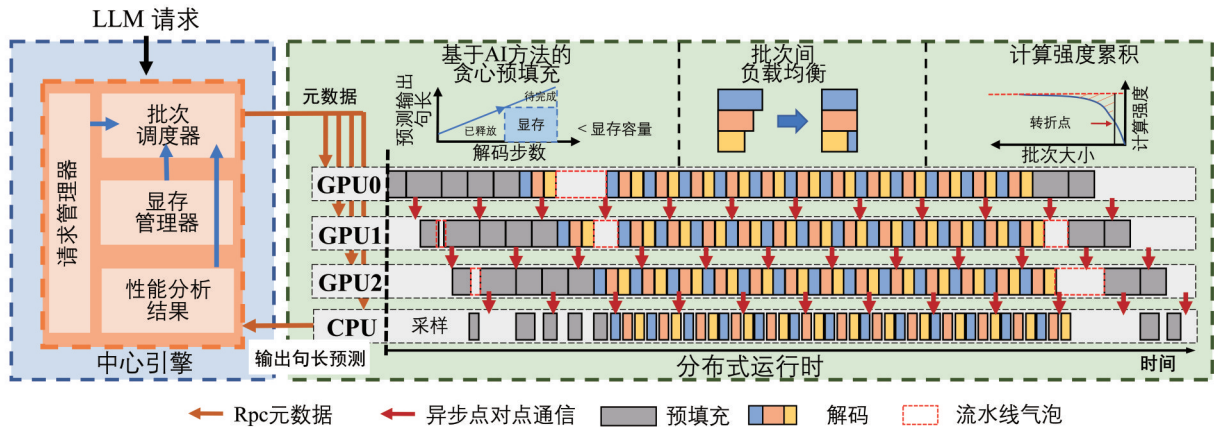


图 7 TD-Pipe 总体架构
Fig. 7 The overall architecture of TD-Pipe

为了更好的实现时间维度分离式流水线并行, TD-Pipe 还包括基于 AI 方法的贪心预填充, 解码批次间负载均衡, 以及计算强度累积切换 3 个方法。基于 AI 方法的贪心预填充方法帮助系统运行更多的 Prefill, 其通过额外的 AI 模型预测请求的输出长度范围, 在避免超过显存容量的同时更加贪心地执行 Prefill, 从而延缓由 Prefill 切换到 Decode 的过

程; 解码批次间负载均衡方法在整个解码阶段过程中持续动态的平均工作负载, 缓解由于请求在解码过程中退出的不确定性产生批次间负载不均衡, 进而产生的空泡问题, 如图 8 所示, 在这个过程中我们基于了滑动窗口思想不断在批次间对批大小进行平衡; 计算强度累积切换方法则用于决定由 Decode 切换到 Prefill 的时机, 即减少由于频



图8 解码批次间负载均衡

Fig. 8 Load balancing between decoding batches

繁切换导致的开销, 同时也要兼顾由于 Decode 阶段计算强度下降导致的硬件利用率不足, 如图9所示, 强度积累方法通过比较计算强度下降带来的性能损失与切换阶段产生的流水线气泡, 来决定是否进行阶段切换。

2 多计算模式应用性能评估

我们借助多种测试工具和应用负载评估了融

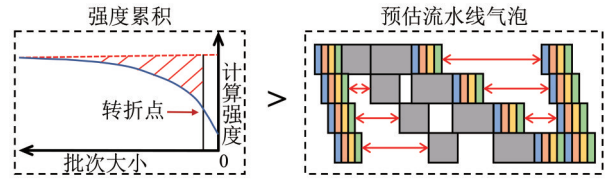


图9 计算强度累积切换方法

Fig. 9 The switching policy based on computation intensity accumulation

合支撑系统的效能, 实验表明, 所研发的多模式融合支撑平台可高效支撑数值模拟、大数据处理、人工智能推理等多种负载模式。

2.1 I/O 性能评估

为了支持多计算模式融合应用, 我们研发的并行文件系统 Kapok 能够同时高效支持数值模拟应用的共享大文件并发访问和大数据、人工智能应用的海量小文件随机访问。当前, 支撑数值模拟和人工智能应用的并行文件系统主要有 Lustre 和 BeeGFS, 支撑大数据处理应用的主要有 HDFS。我们将 Kapok 与它们进行了对比, 实验结果展示在表1中。表1比较了4种文件系统的性能, 其他3种文件系统的性能根据 Lustre 的性能做归一化处理, 结果显示 Kapok 相比其他文件系统在性能上具有显著优势, 且严格遵循 POSIX 语义。

表1 Kapok与主流文件系统的性能对比

Table 1 Comparison between Kapok and SOTA file systems

文件系统	小文件 随机访问	共享大文件 并发访问	Filebench 基准应用I/O测试	元数据 服务器性能	大目录 并发访问	POSIX 国际标准规范
Kapok	2.044	3.373	4.577	2.725	6.207	支持
Lustre	1	1	1	1	1	支持
BeeGFS	0.662	1.834	1.909	0.651	0.927	支持
HDFS	0.134	不支持	不支持 POSIX	0.644	1.597	不支持

2.2 大规模数值模拟应用及通信性能评估

为了评估融合应用支撑系统对传统数值模拟应用的支撑能力, 我们基于新一代天河系统测试了湍流直接数值模拟应用 PowerLLEL 的性能。图10展示了 PowerLLEL 在新一代天河超算上的强可扩展性测试结果。在网格规模高达1433亿的测试算例中, PowerLLEL 实现了从288节点到1728节点的线性扩展, 并行效率达到85%。该良好的可扩展性主要得益于系统强大的通信能力。

为了进一步验证通信优化对 PowerLLEL 的加速效果, 我们在3个不同的超算平台上做了不同规

模的性能评估, 实验结果如图11所示, 通信优化至少将 PowerLLEL 的性能提升30%, 显示了融合支撑系统强大的通信能力。

2.3 大数据处理性能评估

对数值模拟产生的结果实施大数据分析是高性能计算应用后处理的重要手段, 我们研发的大数据处理框架 Topo 针对科学大数据的特征做了专门优化。为了评估其性能, 我们与多种大数据处理框架进行了对比。图12展示了不同大数据框架在科学计算负载上的对比测试结果, 实验环境为新一代天河系统上256节点的计算规模。结果表

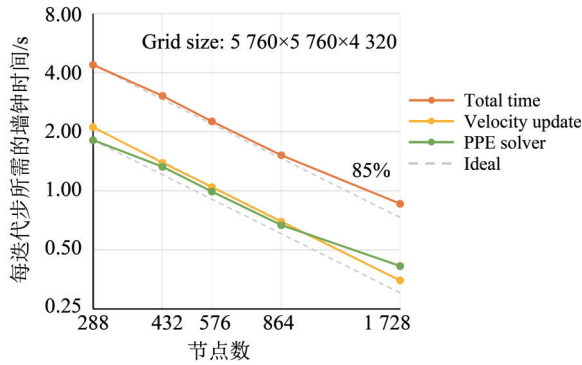


图 10 PowerLLEL 在新一代天河超算上的强可扩展性测试结果

Fig. 10 The strong scalability of PowerLLEL on the new generation Tianhe Supercomputer

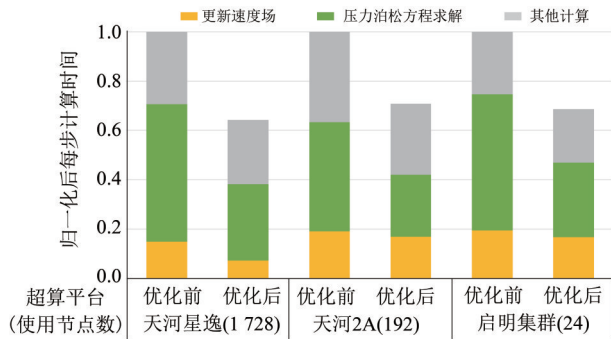


图 11 通信优化对 PowerLLEL 的加速效果

Fig. 11 The performance improvement introduced by communication optimization

明, Topo 框架在 6 个数据集上得到了 5.3 倍的平均加速比, 并且在并行 GaBP 求解器应用中(Zheng et al., 2014), Topo 框架实现了 3.8 倍的平均加速比, 性能高于所有大数据框架和并行库。这种性能提升来自于 Topo 框架内大规模的细粒度任务并行能力, 能够高效地将高维数据的处理映射到大量的一维工作线程上。其次, Topo 能够通过分析数据内的拓扑结构, 减少冗余的全局数据交换过程, 从而缩短应用处理时间。相比之下, 使用 Topo 框架编程的程序还拥有最少的代码行数来实现复杂科学计算应用编程。

图 13 显示, 在 5 项典型大数据处理应用中, 无论是 I/O 密集型应用还是计算密集型应用, Topo 均能获得比其他框架平均快 6.3 倍的性能, 充分证实了我们提出的一系列体系结构优化技术, 为大数据处理框架在超算系统上实现高效融合计算提供了支撑。以上结果表明, Topo 不仅能够支持科学大数据的处理, 在传统大数据处理上也具有性能优势。

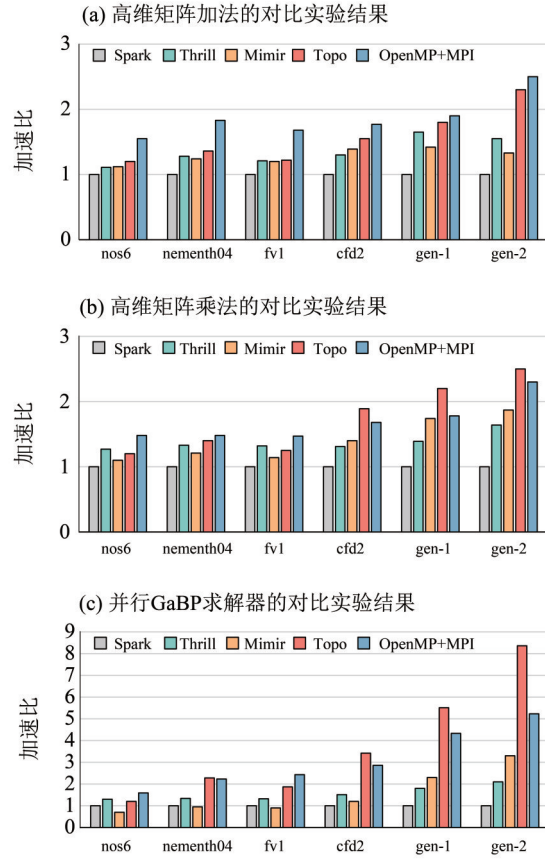


图 12 不同大数据框架在科学计算负载上的对比测试结果
Fig. 12 Comparison between different big data frameworks on scientific computing workloads

2.4 大模型推理性能评估

当前超算与智算融合的主要表现形式的数值模拟与人工智能推理的耦合。为此, 我们研发了 TD-Pipe 并行推理框架, 并评估了其性能。如图 14, 我们将 TD-Pipe 与基线方法进行比较, 并展示了它们在处理 5 000 条输入序列时的整体吞吐量结果, 考虑到内存容量与模型大小之间的比例, 我们选择了 4 种节点与模型的组合: L20 + Llama2-13B、L20 + Llama2-34B、A100 + Llama2-34B 和 A100 + Llama2-70B (Llama2-70B 表示 700 亿参数规模的 Llama2 模型)。为了评估推理的性能, 我们首先将它们的 GPU 从 1 个扩展到 4 个。对于所有节点-模型配置的 4 个 GPU 情况, TD-Pipe 的吞吐量比 vLLM 张量并行高出最多 2.53 倍; 比 vLLM 流水线并行高出 2.65 倍; 比使用了 chunked prefill (Agrawal et al., 2024) 的 vLLM 流水线并行高出 2.29 倍。TD-Pipe 展示了超线性加速, 随着 GPU 数量的增加, 其性能超过了其他流水线并行方法。当 GPU 数量从 2 个增加到 4 个时, L20 + Llama2-34B 的吞吐量增长了 3.24 倍, 显著提高大模型推理吞吐能力。

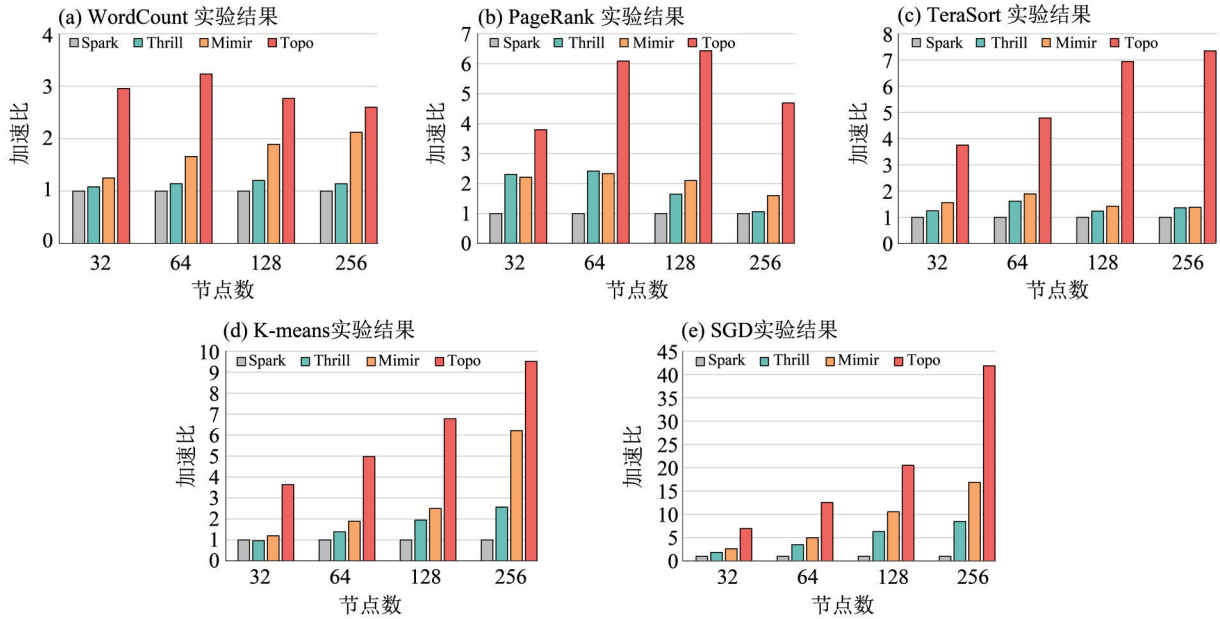


图 13 不同大数据框架在典型大数据处理应用上的对比测试结果
Fig. 13 Comparison between different big data frameworks on traditional big data applications

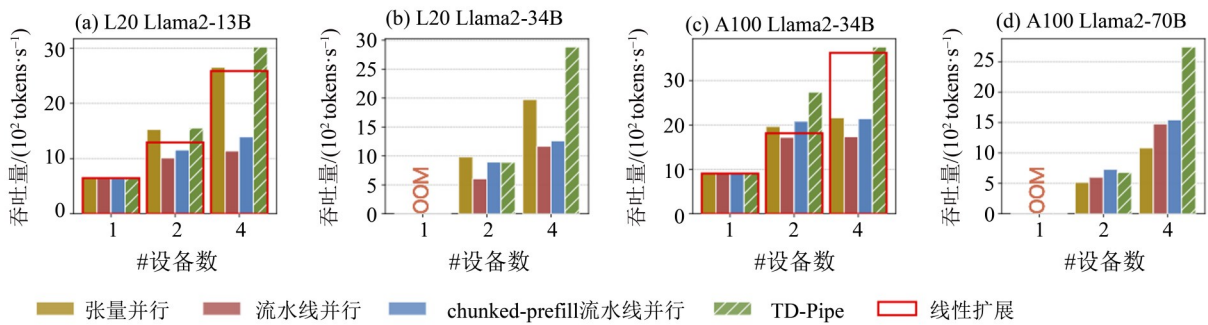


图 14 并行推理实验结果
Fig. 14 Experimental Results of Parallel Inference

进一步地，我们进行了TD-Pipe实例数量扩展实验用以验证大模型高通量推理服务能力。实验配置为A800 + Llama2-70B，模型参数规模为700亿。每个实例依然采用4张卡，一个中央调度器负责将请求分发到不同实例。由于实验环境的每个A800节点配有8张卡，在高通量实验中，一个节点可容纳两个实例，一共64个GPU被启用，测试了最高64个GPU同时参与的高通量推理服务能力。如图15所示，从4个GPU扩展到64个GPU，随着设备数量的扩展请求服务能力几乎呈现线性增长，主要由于中央调度器能力不足导致小幅下降。

2.5 融合应用性能评估

为了验证融合计算应用的性能，我们在新一代天河系统上开展了大规模高通量的密度泛函理论(DFT, density functional theory)计算实验。图16显示了强扩展性的加速比和并行效率。在使用

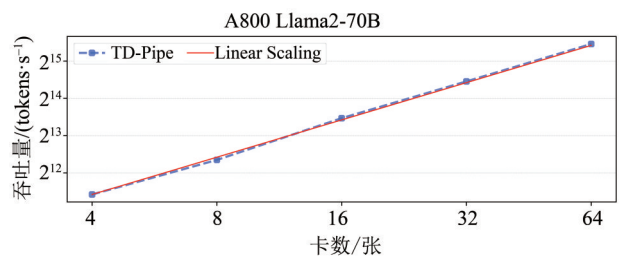


图 15 基于高通量负载的扩展性测试结果
Fig. 15 The scalability on high-throughput workloads

48 000个计算核的情况下，相比2 000核规模计算加速比达到20.93倍，并行效率保持在89%。然而DFT计算极其耗时，为了提高计算效率，我们采用人工智能算法对DFT计算数据进行拟合，从而直接预测材料性质。如图17所示，我们基于自主研发的Crysformer模型(Chen et al., 2022)，利用高性能计算得到的DFT数据进行训练。研究发现，随着DFT数据规模的增加，模型的准确性逐渐提

高, 这表明人工智能模型对数据规模有较高的依赖性。同时, Crysformer模型在准确性方面优于CGCNN(Xie et al., 2018)和MEGNet(Chen et al., 2019)模型, 但略低于Material Project(Jain et al., 2013)、AFLOW(Curtarolo et al., 2012)和OQMD(Kirklin et al., 2015)等主流数据库的计算精度。为了进一步提升预测精度, 我们采用迁移学习策略,

利用少量的湿实验数据对模型进行了调优。经调优后的Crysformer-TL模型在材料性质预测测试中取得了0.45 eV的准确性, 与高精度HSE泛函(Heyd et al., 2003)的结果相近。在本研究中, 我们结合高性能计算和人工智能方法, 成功构建了高精度的材料性质预测模型, 同时计算效率提升超过1 000倍。

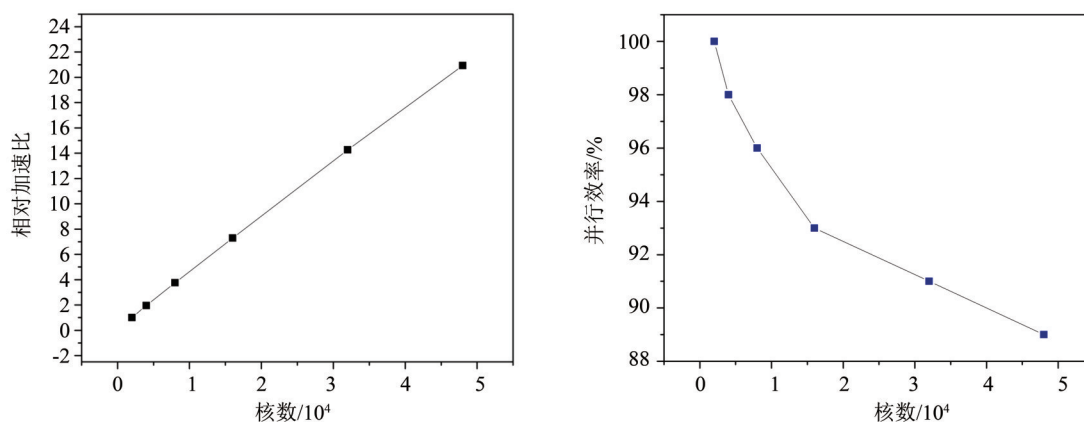


图16 密度泛函计算的强扩展加速比和并行效率

Fig. 16 The strong scalability and parallel efficiency of density functional theory computing

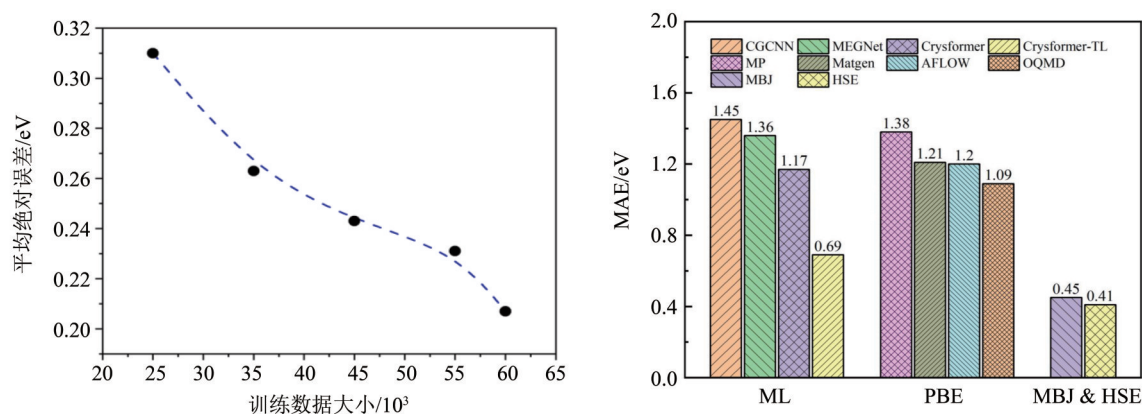


图17 基于AI4Sci的材料性质预测模型

Fig. 17 Material property prediction model based on AI4Sci technology

3 总结

随着超算应用范式的不断演进, 当前的超算系统不仅需要支持传统的数值模拟应用, 还需要同时支持大数据处理与人工智能应用。以上新的应用场景迫切需要重构超算的系统软件, 从而支撑多种计算模式的融合。为此, 我们优化了超算上的并行文件系统、通信系统、资源管理和作业

调度系统。基于上述系统软件的优化, 我们进一步重新设计了大数据处理框架和大模型推理框架, 实现了数值模拟、大数据处理、人工智能推理在超算上的融合。我们从I/O性能、通信性能、数值模拟、大数据处理、人工智能推理、融合应用等多个层面评估了系统的性能, 应用结果表明, 所研发的融合支撑系统在不同类型的应用负载下均取得了显著的加速比。

参考文献:

- 梅宏, 杜小勇, 金海, 等, 2023. 大数据技术前瞻 [J]. 大数据, 9(1):1-20.
- AGRAWAL, A, LEDIA N, PANWAR A, et al, 2024. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve [C]//18th USENIX Symposium on Operating Systems Design and Implementation. Santa Clara, CA, USA: 117-134.
- CHEN C, YE W, ZUO Y, et al, 2019. Graph networks as a universal machine learning framework for molecules and crystals[J]. Chem Mater, 31(9): 3564-3572.
- CHEN P, CHEN J, YAN H, et al, 2022. Improving material property prediction by leveraging the large-scale computational database and deep learning[J]. J Phys Chem C, 126(38): 16297-16305.
- CURTAROLO S, SETYAWAN W, WANG S, et al, 2012. AFLOWLIB.ORG: A distributed materials properties repository from high-throughput *ab initio* calculations [J]. Comput Mater Sci, 58: 227-235.
- DU J, WEI J, JIANG J, et al, 2024. Liger: interleaving intra- and inter-operator parallelism for distributed large model inference [C]//Proceedings of the 29th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming. Edinburgh United Kingdom, ACM: 42-54.
- HEYD J, SCUSERIA G E, ERNZERHOF M, 2003. Hybrid functionals based on a screened Coulomb potential[J]. J Chem Phys, 118(18): 8207-8215.
- HUANG, Y, CHENG Y, BAPNA A, et al, 2019. Gpipe: Efficient training of giant neural networks using pipeline parallelism [C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. NY, USA :103-112.
- JAIN A, ONG S P, HAUTIER G, et al, 2013. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation[J]. APL Mater, 1(1): 011002.
- KIRKLIN S, SAAL J E, MEREDIG B, et al, 2015. The Open Quantum Materials Database (OQMD): Assessing the accuracy of DFT formation energies[J]. NPJ Comput Mater, 1: 15010.
- MEI S, GUAN H, WANG Q, 2018. An overview on the convergence of high performance computing and big data processing [C]//2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). Singapore, IEEE: 1046-1051.
- SEO S, AMER A, BALAJI P, et al, 2018. Argobots: A lightweight low-level threading and tasking framework [J]. IEEE Trans Parallel Distrib Syst, 29(3): 512-526.
- SHOEYBI M, PATWARY M, PURI R, et al, 2019. Megatron-LM: Training multi-billion parameter language models using model parallelism [EB/OL]. arXiv: 1909.08053. <http://arxiv.org/abs/1909.08053>.
- WANG J, FANG J, LI A, et al, 2024. PipeFusion: Displaced patch pipeline parallelism for inference of diffusion transformer models[EB/OL]. arXiv: 2405.14430. <http://arxiv.org/abs/2405.14430>.
- XIE T, GROSSMAN J C, 2018. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties[J]. Phys Rev Lett, 120(14): 145301.
- YIN F, SHI F, 2022. A comparative survey of big data computing and HPC: From a parallel programming model to a cluster architecture[J]. Int J Parallel Program, 50(1): 27-64.
- ZHENG H, SONG A, LIU Z, et al, 2014. Research of multi-core-based parallel gapb algorithm with dynamic load-balance[J]. Int J Numer Anal Mod B, 5(1): 123-135.

(责任编辑 张 冰)