面向高能物理的可计算存储系统研究与实现

高宇[1,2],程耀东[1,2,3],张敏行[1,2],程垚松[1],毕玉江[1]
(1.中国科学院高能物理研究所,北京 100049;2.中国科学院大学,北京 100049;
3.四川天府新区宇宙线研究中心,四川 成都 610213)

摘 要: 高能物理实验数据处理普遍采用存算分离的计算模式,计算过程中需要在计算节点和存储节点间传输数据。实验数据和数据分析需求的不断增长造成了数据传输瓶颈,降低了系统整体的处理效率。针对上述问题提出面向高能物理的可计算存储系统。首先,对存储软件 EOS 进行扩展,在原架构的基础上增加可计算存储插件,存储服务器解析用户命令后,在文件 I/O 的基础上执行本地计算,减少数据移动,缓解网络压力,提升数据处理效率。然后,构建基于中央处理器-现场可编程门阵列(CPU-FPGA)异构计算架构的可计算存储服务器。针对 I/O 密集型任务计算复杂度较低的特点,将适合并行计算的任务通过 PCIe 总线卸载到 FPGA 中,扩展了存储服务器的计算能力。对系统的实验评估结果表明,可计算存储系统能有效消除排队时间和网络延迟,进而缩短计算任务的整体执行时间。基于 FPGA 的硬件加速,有效弥补了存储服务器中 CPU 计算性能较弱的缺陷,提升了可计算存储设备的算法通用性。在基于 LHAASO 的解码作业的测试中,可计算存储系统实现了约 6 倍的速度提升。

关键词: 可计算存储;异构计算;现场可编程门阵列;大数据;高能物理
中图分类号: TP391 **文献标志码:** A **DOI:** 10.19678/j.issn.1000-3428.0070060

Research and Implementation of Computational Storage System for High-Energy Physics

GAO Yu[1,2], CHENG Yaodong[1,2,3], ZHANG Minxing[1,2], CHENG Yaosong[1], BI Yujiang[1]
(1.Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China;
2.University of Chinese Academy of Sciences, Beijing 100049, China;
3.TIANFU Cosmic Ray Research Center, Chengdu 610213, Sichuan, China)

【Abstract】 In high-energy physics experiments, data processing typically involves a compute-storage separated computing model. During the computation process, data must be transferred between the computing and storage nodes. The continuous growth in experimental data and data analysis demands has led to data transfer bottlenecks, reducing the overall processing efficiency of these systems. This paper proposes a computational storage system for high-energy physics. First, the storage software EOS is extended. Computational storage plugins are introduced by building on the original architecture. After parsing user commands, the storage server executes local computations based on the file I/O, thereby reducing data movement, alleviating network pressure, and enhancing data processing efficiency. Second, a computational storage server based on a Central Processing Unit-Field Programmable Gate Array (CPU-FPGA) heterogeneous computing architecture is constructed. Considering the lower computational complexity of I/O-intensive tasks, tasks suitable for parallel computing are offloaded to the FPGA via the PCIe bus, thereby extending the computational capabilities of the storage server. Experimental evaluations show that the computational storage system eliminates queuing time and network latency, thereby shortening the overall execution time of computational tasks. Moreover, leveraging FPGA-based hardware acceleration effectively compensates for the weak computing performance of CPUs in storage servers, thereby enhancing the algorithmic versatility of computational storage devices. In tests based on decoding by LHAASO, the computational storage system achieves a speedup of approximately sixfold.

【Key words】 computational storage; heterogeneous computing; Field Programmable Gate Array (FPGA); big data; high-energy physics

基金项目: 国家自然科学基金(12075268);中国科学院高能物理研究所创新基金(E15451U210)。

作者简介: 高宇,男,博士研究生,主研方向为分布式存储、可计算存储;程耀东(通信作者),研究员、博士;张敏行,博士研究生;程垚松,硕士;毕玉江,副研究员、博士。

收稿日期: 2024-07-01 **修回日期:** 2024-08-28 **E-mail:** chyd@ihep.ac.cn

0 引言

随着人工智能、物联网以及各种音视频技术等的广泛应用和快速发展，产生数据的规模和速度都与日俱增。人类社会将很快进入尧字节(YB)数据时代，数据的处理、传输和存储都将面临巨大的挑战。在高能物理领域，由于实验装置的复杂度和规模的不断提升，产生了大量待分析的原始数据。仅国家高能物理科学数据中心每年分析的数据量就达到 400 PB，对数据的处理和存储能力提出了新的要求。

高能物理实验数据处理流程因实验而异，但普遍包括数据获取、重建、物理分析、数据存储与管理以及数据共享与分发等步骤。实验各个阶段产生的有效数据都需要进行存储，尤其是原始数据，由于其特殊性，往往需要永久化保存，因此高能物理领域通常采用存算分离的模式[1]。

典型的高能物理计算环境包括登录集群、存储集群、计算集群、分级存储系统、防火墙和管理系统等。这些子系统相互独立，通过高度可靠的高速核心网络互联。其中，登录集群是用户访问计算环境的单一入口。登录成功后即可使用海量存储、高性能计算系统、数据库等资源。计算集群通过作业调度系统整合大量计算节点上的 CPU 资源，对用户提交的任务进行统一的调度安排。存储集群挂载到计算集群中。计算节点执行计算任务时，可以像访问本地文件系统一样读写存储集群中的文件。存储集群由多个存储节点组成，并部署分布式文件系统。目前，高能物理领域常用的大型分布式文件系统有 Lustre[2]、EOS[3]等。

EOS 等分布式文件系统的使用，使得系统的构建和扩充更为灵活，支持大规模的异构硬盘存储环境和冷热存储的分离。然而，存算分离架构的弊端在 I/O 密集型任务中也充分暴露出来。一方面，数据从存储节点通过网络拷贝到计算节点，计算完成后又将结果传输回存储节点。频繁的数据搬运会占用大量网络带宽，造成网络拥塞。另一方面，I/O 密集型任务通常 CPU 利用率并不高，在计算节点上执行此类任务，会因“存储墙”问题造成计算资源的浪费。

随着数据量的不断增大，网络升级的速度难以匹配数据量增加的速度。因此，需要尝试新的计算架构，以应对海量数据带来的挑战。基于“move process to data”的范式，已经有诸多尝试。例如，存内计算[4-5]直接在存储器芯片内部进行数据处理，从根本上消除了数据的搬运，实现了存储与计算的深度融合。存内计算当前受计算精度、应用生态以及成本等多方面的制约，距离大规模应用仍存在诸多挑战。

可计算存储技术将计算单元的部分任务卸载到存储单元，使数据处理迁移到更靠近数据的地方，以此减少数据的移动，提升系统整体的处理效率[6]。除了利用 CPU 来完成计算任务外，还可以通过异构计算来补充存储单元的计算能力，如 GPU、DPU、SoC 等。根据具体应用的特点，把不同类型的任务分配到适合的计算部件上，以达到计算加速的目的。

高能物理实验数据的特点是：数据量大，有冷热数据之分，原始数据需要永久化存储。因此，高能物理实验数据的存储系统需要具备以下特点：存储容量大；根据数据热度采用不同的存储介质进行分级存储；能有效应对频繁的数据搬运带来的影响。基于这些特点，在单个硬盘中实现面向高能物理数据的可计算存储，可能造成计算资源的浪费，同时大幅增加成本。因此，本文提出了面向高能物理的可计算存储系统实现方法，主要研究内容和贡献有以下 3 个方面：

1)提出面向高能物理的可计算存储整体框架，包括底层硬件设备、可计算存储基础软件以及领域内的热点算法的移植。

2)扩展高能物理领域常用的分布式存储软件 EOS，实现可计算存储功能。在 EOS 原有功能的基础上将计算节点的计算任务卸载到存储节点，计算和 I/O 操作都在存储节点本地完成。

3)采用 ARM CPU 和现场可编程门阵列(FPGA)异构计算架构组建可计算存储服务器。该架构在保证满足计算需求的前提下，能够匹配硬盘设备的读写速度，具有高性能、低功耗和低成本的特点。

1 相关工作

1.1 可计算存储

可计算存储有多种不同层次的实现方法。全球网络存储工业协会成立了可计算存储工作组，制定了可计算存储的相关技术标准[7]，定义了支持可计算存储的软硬件的推荐行为，并介绍了几种典型的可计算存储应用方式。例如：可计算存储驱动器(CSD)，可计算存储处理器(CSP)，可计算存储阵列(CSA)等。其中，CSD 用于提供可计算存储服务和持久化数据存储。CSP 用于在不提供持久化数据存储的前提下提供可计算存储服务。而 CSA 则是由 CSD、控制软件和可选的存储设备组成。

文献[8]将 SSD 和 FPGA 进行集成，实现了 SmartSSD 可计算存储驱动器，解决了 SSD 吞吐量和 CPU 计算能力失配的问题。基于 SmartSSD，文献[9]实现了近存排序系统。文献[10]将开销较大的表扫描操作从 CPU 卸载到基于 FPGA 的可计算存储驱动器中，有效降低了 PolarDB 数据库的查询延迟和数据传输量。在虚拟化环境中，应用可计算存储面临的基本挑战是如何经济高效地实现虚拟化。文献[11]提出了 FCSV-Engine FPGA 卡，通过硬件辅助的虚拟化和资源编排来实现高虚拟化性能，并构建多个虚拟可计算存储设备，从硬件层面进行调度，完成近存处理。目前，硬件层面的可计算存储主要是利用 FPGA 对 SSD 进行扩展，将计算能力扩展到硬盘内部，使计算尽可能地靠近数据。在存储器层面，扩展可计算存储能力的好处是显而易见的，但是在实际应用中容易因数据访问频率低造成计算资源的浪费。

软件层面的可计算存储主要是对已有存储系统进行扩展，使之具备可计算存储的特点。文献[12]提出了 λ-I/O，通过扩展 Linux I/O 栈并提供 API 接口，构建统一的运行时环境以及动态请求分派，实现了对主机和可计算存储设备间的计算和存储资源的高效管理。文献[13]扩展了 Ceph 分布式存储系统的数据处理和管理功能，使存储服务器能够语义化地解释对象数据，以执行一些 SQL 语句。其优势在于 I/O 和计算均具有弹性，存储系统自动在可用服务器之间重新平衡对象。文献[14]基于 Zynq MPSoC 和 SSD 实现了可计算存储设备 Catalina，为应用程序提供了文件系统级的数据访问。在 Catalina 上可以直接部署 Hadoop MapReduce 和 HPC 应用程序，通过在存储侧执行计算任务，实现了性能和功能的有效优化。在当前的可计算存储研究中，以数据库的优化加速这样的 I/O 密集且计算不复杂的应用为主。现有的方法难以适应高能物理实验数据的特点，因而不能直接应用到高能物理数据处理中。实现面向高能物理的可计算存储系统，需要兼顾高能物理实验数据的特点和领域内正在大规模应用的系统。对现有存储系统进行扩展，将计算任务卸载到存储节点本地，以提升系统的整体性能，降低网络开销。

在大数据领域，MapReduce[15] 并行计算模型通过将大规模数据集切分为多个分片，并分发到多个具有相同程序副本的节点进行并行计算，然后对计算结果进行整合，得到最终计算结果，使得大批量数据的处理更加高效，更适合计算密集型任务。与分布式计算模型相比，可计算存储系统在存储节点完成对数据的处理，减少了数据在存储节点和计算节点间的传输，有效缓解了 I/O 密集型任务对网络带宽的占用。

1.2 FPGA 计算加速

由于 FPGA 具备良好的可编程性和可并行性，且具有较低的功耗，因而被越来越多地用在计算加速场景中。文献[16]实现了对纠删码编码算法的 FPGA 加速。文献[17]实现了基于 FPGA 的浮点运算加速器。文献[18]将 FPGA 部署到数据中心的服务器中，组成了可重构、可扩展的大规模集群，实现了对网络搜索引擎的加速。文献[19]提出了基于混沌系统的 FPGA 图像加密算法，加密和解密时间远低于其他软件解决方案。

神经网络模型的推理加速也是当前研究的热点领域。文献[20]设计了基于图神经网络的带电粒子轨迹算法的 FPGA 加速器，推理速度相对于 CPU 有显著的提升。文献[21]提出了基于 FPGA 的 CNN 模型并行方法。此外，文献[22]实现了基于 FPGA 的脉冲神经网络的在线训练。可见，随着 FPGA 的不断发展，基于 FPGA 的神经网络训练也有很大前景。

2 面向高能物理的可计算存储系统架构

结合高能物理领域的数据特点，和当前正在大规模使用的计算模式和相关软件，本文提出了面向高能物理的可计算存储整体架构。

如图 1 所示，首先给出基于 ARM CPU 和 FPGA 异构计算架构的可计算存储服务器。其中 FPGA 以可计算存储处理器卡的形式，与多个硬盘组成的硬盘阵列一同挂载到 ARM CPU 的 PCIe 总线上。通过 FPGA 可并行计算的特点，对 CPU 的计算能力进行补充。

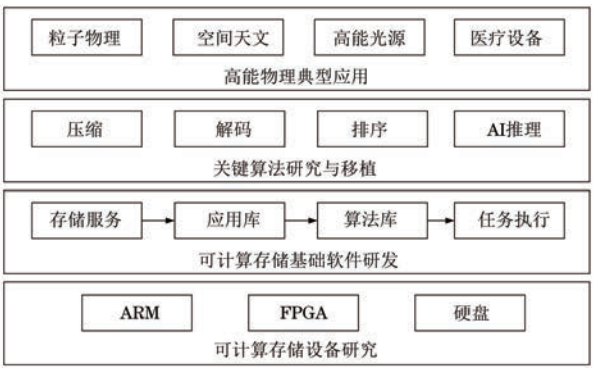

图 1 面向高能物理的可计算存储架构

Fig. 1 Computational storage architecture for high-energy physics

在可计算存储设备或现有存储设备的基础上，提出对现有分布式存储软件 EOS 进行改进，以增加可计算存储插件的方式，提供可计算存储服务。用户只需要在 Open 指令中增加可计算存储指令字段，即可调用可计算存储服务，以此实现面向高能物理的可计算存储基础软件。包括基本的存储服务、可计算存储应用库、各种算法库以及计算任务执行的具体操作。在可计算存储基础软件的支持下，对高能物理领域的关键算法进行移植或研究相应的硬件加速方法。同时，基于关键算法，实现对高能物理典型应用方向的支持。

3 可计算存储系统软件实现

针对现有软件进行可计算存储功能扩展，是可行性最强与开发成本最低的方案。

EOS 软件是欧洲核子中心（CERN）实现的分布式存储系统。高能物理领域的通用数据访问协议 XRootD[23] 为 EOS 提供了高效的文件访问和传输功能。如图 2 所示，EOS 的主要组成部分包括客户端、元数据服务器和数据服务器。在元数据服务器中，管理（MGM）模块完成认证、授权、数据调度、元数据管理和存储管理任务，消息队列（MQ）模块完成消息代理，QuarkDB（QDB）数据库提供了高可用的键值存储，用于持久化元数据。在数据服务器中，由文件存储传输（FST）模块完成文件的存储和传输。客户端对于数据文件的访问操作，需要先访问元数据服务器内存中的元数据，再访问数据所在的服务器，以此保证操作的可靠性和多副本的一致性，也保证了对文件元数据的快速响应和对文件的低延迟访问。另外，通过设置多个 I/O 节点来实现并行处理和负载均衡。元数据与数据分离的模式增强了数据的安全性，提高了海量数据的访问效率，保证了 EOS 的高性能和可扩展性。

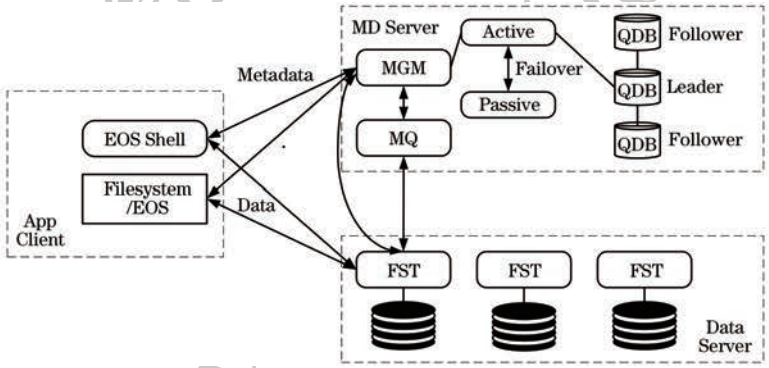

图 2 EOS 软件架构

Fig. 2 EOS software architecture

为了在此基础上实现可计算存储，将计算节点的计算任务直接卸载到存储节点，本文提出一种基于 EOS 分布式存储系统的可计算存储软件架构。通过扩展 EOS 的 FST 模块，实现对现有存储系统的可计算存储功能扩展。

如图 3 所示，在原有的 FST 模块基础上，以插件的形式增加提供可计算存储服务的文件存储和传输（CSSFST）模块。客户端在解析用户命令时，识别到可计算存储参数后，通知存储服务器中的 CSSFST 替代 FST 执行对文件的具体操作。在文件 I/O 的基础上执行计算操作，并根据具体计算任务类型将计算结果存放在存储服务器本地，或返回给调用方。

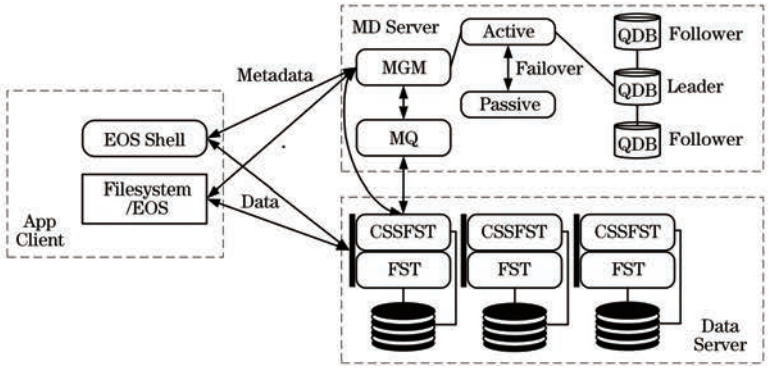

图 3 基于 EOS 的可计算存储架构

Fig. 3 Computational storage architecture based on EOS

CSSFST 的实现流程如图 4 所示。存储节点的 CSSFST 收到 Open 调用后，首先对调用参数进行解析，检测是否存在“CSS”关键字。如果不存在，则直接调用 EOS 传统的 FST 的 Open 方法。如果存在“CSS”关键字，则进一步依据配置文件进行 CSS 参数的解析。

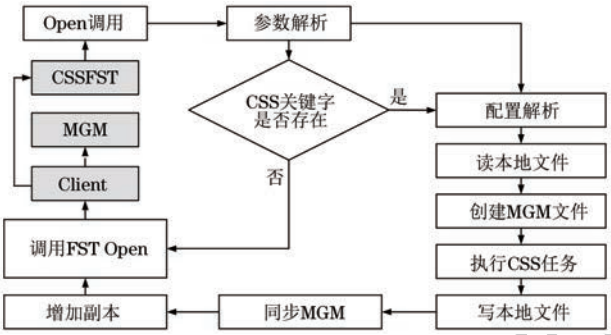

图 4 CSSFST 执行流程

Fig. 4 CSSFST execution flow

配置文件以 JSON 格式提供以下信息：

1）算法的名称。算法名称是客户端调用 CSS 服务时的依据，与执行的算法名称一一对应。

2）算法可执行文件或脚本的路径。算法可执行文件或脚本用于完成针对文件进行的计算任务，其路径由用户在客户端调用时指定，且可完全独立于 EOS 运行。

3）输出文件的尾缀。首先指出是否存在输出文件，若存在，则进一步指定输出文件的尾缀。

经过参数解析，CSSFST 已经获得了目标文件及对应的可计算存储算法。然后，读取节点本地的目标文件，并向元数据服务器注册 MGM 文件。接着，对读取的目标文件调用指定的可计算存储算法。算法的输出结果将写到节点本地的输出文件中，输出文件的命名基于 EOS 的规则。如果算法本身不产生输出文件，则可将 stdout 或 stderr 重定向到输出文件中。随后，将结果同步到元数据服务器的 MGM 模块，并增加副本备份。最后，对输出文件执行 Open 调用，并返回给客户端。

4 基于 FPGA 的可计算存储硬件加速

4.1 硬件架构

FPGA 是一种可编程硬件设备，具有可配置的内部逻辑和内部连接，用户可根据需要实现特定的数字电路。不同于 CPU、GPU 通过指令实现串行计算，FPGA 通过定制电路可实现并行计算，因此能在特定计算任务中发挥显著优势。

为了基于大容量存储设备实现可计算存储，并对特定计算任务进行硬件加速，本文提出一种基于 ARM CPU 和 FPGA 异构计算架构的可计算存储服务器。其中，FPGA 以可计算存储处理器卡的形式，作为对 CPU 计算能力的补充。磁盘阵列与 FPGA 通过 PCIe 挂载到 CPU 上，配备 10 Gb/s 以太网卡实现高速通信。通过高速以太网交换机，该系统可直接连接到数据中心网络中。除了实现可计算存储功能，也可作为普通的存储服务器提供存储服务。如图 5 所示，系统中主要分为 4 个模块，分别为 ARM CPU、FPGA、磁盘阵列和网络模块。其中，FPGA 和硬盘阵列均通过 PCIe 总线与 CPU 进行连接，硬盘阵列中最多可挂载 12 个 SATA 接口硬盘。

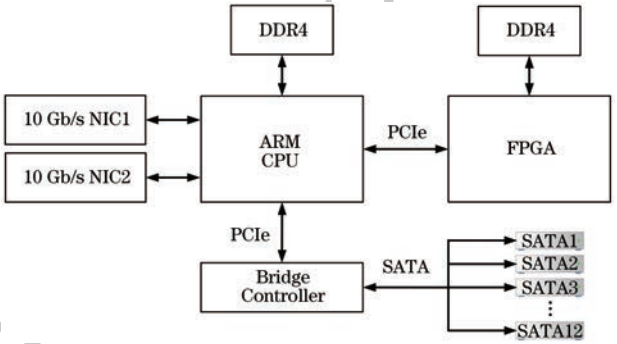

图 5 可计算存储服务器系统架构

Fig. 5 Computational storage server system architecture

基于 CPU、网络模块和磁盘阵列，系统即可完成传统存储服务器的任务。在此基础上，可从软件层面提供可计算存储服务，实现计算任务在存储节点的卸载，大幅减小跨节点数据搬运造成的计算任务总体延迟。另外，通过调用 FPGA 进行硬件加速，可实现部分算法的本地高效执行，进一步加快计算任务在存储节点本地的执行速度。

4.2 硬件加速方式

FPGA 作为硬件加速器时受内部资源限制，不能把所有算法都加载到 FPGA 内部。因此，以预先综合实现 FPGA 算法核心，调用时以动态加载的方式实现加速算法的切换。将算法模块单独划分到动态逻辑区，其他模块处于静态逻辑区。其中，静态逻辑区内的电路逻辑在运行时保持不变，动态逻辑区内的电路逻辑可在运行时根据需要实时更新。

如图 6 所示，FPGA 以 PCIe 从机的形式与硬盘阵列一同挂载到 ARM CPU 的 PCIe 总线上。在 FPGA 内部，通过 PCIe 控制器（PCIe Controller）完成 PCIe 信号和 AXI-Stream 总线信号的相互转换。其中，AXI-stream 信号用于 FPGA 的内部通信。算法模块、DDR 控制模块与 PCIe 控制模块是数据的主要提供方和接收方。三者共同连接到路由模块

上，通过路由模块做数据的转发。发送数据时在数据帧的最前方增加标识数据长度和目标模块的帧头，作为路由模块转发数据的依据。收到数据的模块首先对帧头进行解析，根据数据长度接收有效数据。

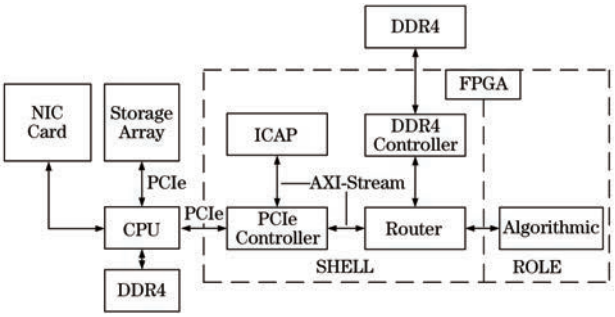

图 6 FPGA 功能模块示意图

Fig. 6 Schematic diagram of FPGA function module

执行 FPGA 算法加速任务时，首先，CPU 将算法模块的二进制程序文件动态加载至 FPGA 中。然后，CPU 向 FPGA 输入数据。根据具体应用，可以直接将数据帧发送至 FPGA 的算法模块中，实现流式处理，也可以先缓存至 FPGA 的 DDR4 内存中，再由算法模块读出，实现批处理。数据传输完成后，由 FPGA 算法模块对输入数据进行处理。最后，FPGA 将计算结果传回 CPU，由 CPU 存放至本地。

FPGA 加速器与 CPU 构成了异构计算架构，FPGA 以其灵活的可编程性和并行性，能够在特定应用中扩充 CPU 的计算能力。FPGA 加速器提供可直接调用的可执行程序或用于混合编程的程序接口，供可计算存储软件调用。例如前述的基于 EOS 实现的 CSSFST 可以直接调用可执行程序，实现经硬件加速的可计算存储服务。

5 实验评估

可计算存储系统的性能与应用密切相关，在 I/O 密集型应用或可利用硬件进行加速的应用场景下具有良好的性能。本文结合高能物理数据处理的实际需要和可计算存储的特点，进行了 3 种应用模式的实验评估：第 1 种是纯软件的可计算存储服务，直接将原先放在计算节点进行的解码程序，放到存储节点运行；第 2 种是纯硬件的可计算存储服务，利用 FPGA 实现完整的压缩算法，存储节点的 CPU 只负责将数据传输到 FPGA 中；第 3 种是软硬件混合的可计算存储服务，在计算的不同阶段使用不同的计算部件，充分发挥 CPU 和 FPGA 的性能。

5.1 可计算存储服务评估

本节对可计算存储服务的性能评估基于对高海拔宇宙线观测站(LHAASO)[24]实验数据的解码任务。LHAASO 实验通过精确测量宇宙线粒子和伽马射线在大气中产生的空气簇射，探索高能宇宙线起源。LHAASO 的电子学探测器每年产生约 10 PB 的原始二进制数据。原始数据缓存之后，需要进一步解码成规范化数据并经压缩后存储，供后续的物理分析使用。每天会从实验站点的存储服务器传输上万个数据文件到远程数据中心进行解码。

在执行解码任务时，首先读取二进制文件，按照固定的格式解析文件数据，提取出时间、能量、位置等各种信息，将解析的结果封装成事例数据，并以 ROOT[25]格式进行保存。ROOT 是由欧洲核子中心开发的数据分析框架，其定义了 ROOT 文件格式。这一过程只占用较少的 CPU 资源，但需要大量的 I/O 操作，属于典型的 I/O 密集型任务。

按照传统的存算分离模式，计算节点从一个 EOS 数据服务器读取原始数据文件，经过解码运算后将 ROOT 文件写入到某个 EOS 数据服务器中。采用可计算存储服务后，任一 XRootd 客户端均可以启动存储节点的可计算存储服务，在存储节点本地对数据进行解码处理。

图 7 所示是基于存算分离模式与基于可计算存储模式下，执行 LHAASO 解码任务时的耗时对比。测试环境采用了单台存储服务器，CPU 型号是 Intel® Xeon® CPU E5-2683 v4 @ 2.10 GHz，硬盘型号是 SAMSUNG MZ7KH960 960 GB。测试中采用的原始数据文件的大小为 953.7 MB。多任务时使用同一文件的多个副本。两种模式采用相同的解码可执行文件。在存算分离模式下，计算节点从存储服务器读取原始数据文件，并执行解码操作，最后将结果写回存储服务器。在可计算存储模式下，计算节点发出命令后，存储节点执行解码操作，并将结果保存到本地。

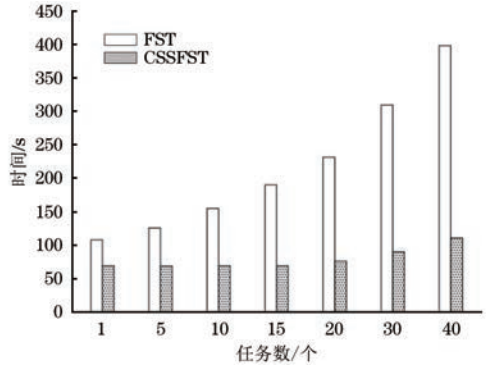

图 7 不同模式下的解码耗时对比

Fig. 7 Decode time-consuming comparison in different modes

从测试结果可以看出，存储节点的计算能力足以胜任 I/O 密集型任务的计算资源需求。由于数据的读写都发生在本地，不需要经过网络的传输，可计算存储模式下的执行时间相对于存算分离模式下的执行时间明显缩短。且随着执行任务数的增加，充分利用了本地 I/O 带宽，性能提升明显。而在存算分离模式下，大量数据的搬运严重依赖网络带宽，造成了随着任务数增多，耗时迅速增加的情况。

5.2 FPGA 加速评估

数据压缩是高能物理以及各种大数据场景下的常见操作。因为压缩算法多为流式处理流程，因此适合使用 FPGA 进行算法加速。本节基于 Xilinx 公司开源的 Zlib 和 Zstd 的 L1 级压缩算法 IP 核，与相应的 CPU 压缩算法进行了比较分析。

如图 8 所示，分别在 X86 架构服务器和本文提出的配备了 FPGA 加速卡的 ARM 服务器上，对 Zlib 和 Zstd 两种压缩算法进行了测试。图中的 3 组数据分别表示 X86 服务器纯软件压缩、ARM 服务器纯软件压缩和 ARM 服务器使用 FPGA 压缩的耗时。

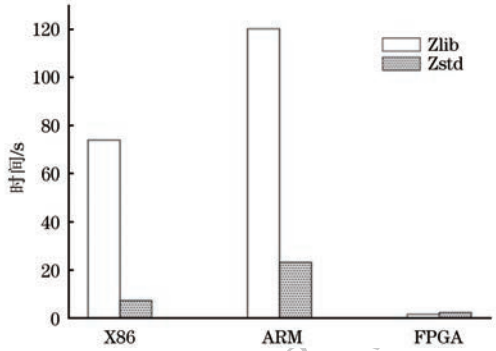

图 8 压缩算法的耗时测试

Fig. 8 Time-consuming testing of compression algorithms

X86 服务器采用的 CPU 型号为 Intel® Xeon® Silver 4215R CPU @ 3.20 GHz。ARM 服务器使用的 CPU 型号为 NXP LS1046a @1.8 GHz，装载的 FPGA 卡使用的 FPGA 型号为 XCKU5P-FFVB676-2-E。测试采用的数据为 980.8 MB 大小的 LHAASO 原始数据文件。压缩后的文件大小约为 330 MB。

在 X86 服务器和 ARM 服务器执行纯软件压缩算法时，直接使用命令行软件对目标文件进行压缩，并记录从运行命令到压缩完成的总时间。采用的命令行软件版本分别为 Gzip v1.9 和 Zstd v1.4.4。在 ARM 服务器上调用 FPGA 加速卡执行压缩算法加速时，由 CPU 打开目标文件，将文本内容读出并通过 PCIe 总线传输至 FPGA 加速卡，然后从 FPGA 接收压缩后的数据，写入到输出文件中。FPGA 压缩耗时是 FPGA 执行压缩算法和 CPU 读写和传输数据的总耗时。

由测试结果可以看出，仅使用 CPU 执行压缩任务时，X86 服务器的执行性能优于 ARM 服务器，这与所采用的 CPU 性能有很大关系。在 ARM 服务器上使用 FPGA 执行压缩任务后，其压缩速度远超 X86 服务器和 ARM 服务器上的纯 CPU 压缩速度。其中，Zlib 算法的性能提升非常明显，相对于 X86 服务器和 ARM 服务器上的 CPU 算法，分别得到了约 46 倍和 75.45 倍的速度提升。而 Zstd 算法由于本身运行速度快，提升相对较少，相对于 X86 服务器和 ARM 服务器上的 CPU 算法分别得到 2.23 倍和 9.23 倍的速度提升。

5.3 作业调度环境下的可计算存储服务评估

第 5.1 节中以手动输入命令行的方式，比较了在 EOS 分布式存储系统中采用可计算存储模式和传统模式下的性能差异。然而，在实际的计算环境中，会以提交作业的形式申请计算资源，拉取作业数据。图 9 所示为一个真实环境下的解码作业的执行过程。登录节点提交解码作业后，调度系统分配计算节点。计算节点通过网络从远程的存储节点读入约 1 GB 的输入数据文件。经过数据处理后，计算节点再将输出的约 300 MB 文件通过网络写入到某一存储节点中。

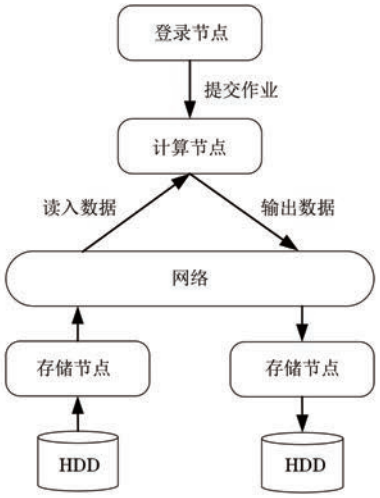

图 9 解码作业执行过程

Fig. 9 The execution process of the decode job

在部署了可计算存储系统的计算环境中，将不再需要分配计算节点，也不需要提交作业，只要在登录节点对输入文件执行含“CSS”关键字及指定算法的 Open 调用即可。如果可计算存储系统的配置文件中含该指定算法，将自动执行该计算任务，并将结果写入本地的硬盘中。在整个过程中，不经过计算

节点和网络，彻底消除了作业排队时间和网络延迟，数据传输完全在本地进行，且部分算法可以经过硬件计算部件进一步加速。

在作业调度环境下，解码作业的处理流程具体可分为作业排队、数据读入、数据解码、压缩、文件输出等几个独立的步骤。表 1 比较了在高能物理实际计算环境中大量作业排队执行的情况下，使用作业调度方式和可计算存储方式执行 LHAASO 解码作业时的平均耗时。从表中可以看出，在作业调度方式下，平均排队耗时高达约 120 s。而在可计算存储方式下，节省了排队过程，数据的读入和输出不经过网络，不再受网络拥塞影响。另外，由于在测试中使用了 FPGA 对压缩算法进行了加速，因此数据压缩部分的耗时也大大缩减。

表 1 两种计算方式下的耗时对比

Table 1 Comparison of time-consuming between the two calculation methods

单位：s

计算方式	作业调度方式	可计算存储方式
排队	～120	0
读入	～40	～10
解码	～15	～15
压缩	～33	～3
输出	～12	～3
总计	～220	～31

总体来讲，可计算存储方式相对于传统的作业调度方式，消除了排队时间和网络时延。同时，部分应用经过硬件加速，速度也得到进一步提升。在实际测试中，可计算存储方式执行解码作业的平均耗时从作业调度的约 220 s 缩减到了约 31 s，速度提升了约 6 倍。

6 结束语

高能物理实验数据量的不断增加，对现有的存算分离的计算架构造成极大的挑战。为了降低计算量小但数据读写量大的 I/O 密集型任务对网络资源的占用，同时更好地使用计算资源，本文提出了适用于高能物理数据的可计算存储系统实现方法，将部分计算任务卸载到存储节点中，从软硬件两个层面实现了可计算存储对高能物理数据处理的适配。通过对 EOS 分布式存储系统进行扩展，实现了基于 EOS 的可计算存储功能，在 LHAASO 解码任务的测试中，相对于传统的存算分离模式，执行时间明显缩短。在基于 ARM CPU 和 FPGA 的可计算存储服务器的测试中，FPGA 对压缩算法的加速效果显著。在 LHAASO 解码作业的测试中，通过消除作业调度和网络延迟，以及使用 FPGA 进行硬件加速，计算作业的整体运行速度同样有明显提升。未来的工作包括三个方面。首先，对基于 ARM CPU 和 FPGA 的可计算存储服务器进行升级，提升整体硬件性能。例如，实现 FPGA 和硬盘设备之间的“零拷贝”，进一步减少设备内部的数据搬运。其次，进一步优化基于现有分布式存储系统的可计算存储方案，包括性能、鲁棒性以及采用 FPGA 进行计算加速时的任务划分和调度等，同时考虑在其他分布式系统中扩展对可计算存储功能的支持。最后，丰富面向高能物理的可计算存储系统的关键算法的种类和硬件加速库，扩展系统整体的应用范围。

参考文献

[1] 程耀东，石京燕，陈刚. 高能物理计算环境概述[J]. 科研信息化技术与应用，2014，5(3)：3-10.
CHENG Y D，SHI J Y，CHEN G. Overview of high energy physics computing environment[J]. Research Information Technology and Application，2014，5(3)：3-10. (in Chinese)

[2] PHILIPS S. Lustre：building a file system for 1 000-node clusters[C]//Proceedings of the 2003 Linux Symposium. Washington D. C.，USA：IEEE Press，2003：380-386.

[3] PETERS A，SINDRILARU E，ADDE G. EOS as the present and future solution for data storage at CERN[J]. Journal of Physics：Conference Series，2015，664(4)：042042.

[4] 郭昕婕，王光燿，王绍迪. 存内计算芯片研究进展及应用[J]. 电子与信息学报，2023，45(5)：1888-1898.
GUO X J，WANG G Y，WANG S D. Research progress and application of memory computing chips[J]. Journal of Electronics and Information，2023，45(5)：1888-1898. (in Chinese)

[5] 方旭东，吴俊杰. 基于忆阻器的计算存储融合体系结构研究进展[J]. 计算机工程与科学，2020，42(11)：1929-1940.
FANG X D，WU J J. Advance in memristor-based computing storage fusion architecture[J]. Computer Engineering & Science，2020，42(11)：1929-1940. (in Chinese)

[6] 李迦雳，刘铎，陈咸彰，等. 基于闪存存储的近数据处理技术综述[J]. 集成技术，2022，11(3)：23-41.
LI J L，LIU D，CHEN X Z，et al. A survey of flash memory based near-data processing technology[J]. Journal of Integration Technology，2022，11(3)：23-41. (in Chinese)

[7] SNIA Standard. Computational storage architecture and programming Model[EB/OL]. [2024-06-01]. https://www.snia.org/sites/default/files/technical-work/computational/release/SNIA-Computational-Storage-Architecture-and-Programming-Model-1.0.pdf.

[8] LEE J H，ZHANG H，LAGRANGE V，et al. SmartSSD：FPGA accelerated near-storage data analytics on SSD[J]. IEEE Computer Architecture Letters，2020，19(2)：110-113.

[9] QIAO W K，OH J，GUO L C，et al. FANS：FPGA-accelerated near-storage sorting[C]//Proceedings of the 29th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines. Orlando，USA：IEEE Press，2021：106-114.

[10] CAO W，LIU Y，CHENG Z S，et al. POLARDB meets computational storage：efficiently support analytical workloads in cloud-native relational database[C]//Proceedings of the 18th USENIX Conference on File and Storage Technologies. Washington D. C.，USA：IEEE

Press, 2020: 29-41.
[11] KWON D, KIM D, BOO J, et al. A fast and flexible hardware-based virtualization mechanism for computational storage devices[C]//Proceedings of the 2021 USENIX Annual Technical Conference. Washington D. C., USA: IEEE Press, 2021: 729-743.
[12] YANG Z, LU Y Y, LIAO X J, et al. λ-I/O: a unified I/O stack for computational storage[C]//Proceedings of the 21st USENIX Conference on File and Storage Technologies. Washington D. C., USA: IEEE Press, 2023: 347-362.
[13] LeFEVRE J, MALTZAHN C. SkyhookDM: data processing in Ceph with programmable storage[J]. USENIX Magazine, 2020, 45(2): 13-18.
[14] TORABZADEHKASHI M, REZAEI S, HEYDARIGORJI A, et al. Computational storage: an efficient and scalable platform for big data and HPC applications[J]. Journal of Big Data, 2019, 6(1): 100.
[15] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.
[16] 杨思捷, 陈俊奇, 王勇, 等. 基于 FPGA 的软硬件协同纠删码编码加速方案[J]. 计算机工程, 2024, 50(2): 224-231.
YANG S J, CHEN J Q, WANG Y, et al. FPGA-based software and hardware cooperative acceleration scheme of erasure code encoding[J]. Computer Engineering, 2024, 50(2): 224-231. (in Chinese)
[17] 关明晓, 刘嘉堃, 张鸿锐, 等. 基于 FPGA 误差可控的浮点运算加速器研究[J]. 计算机工程, 2024, 50(5): 291-297.
GUAN M X, LIU J K, ZHANG H R, et al. Study of FPGA-based error-controllable floating-point operation accelerators[J]. Computer Engineering, 2024, 50(5): 291-297. (in Chinese)
[18] PUTNAM A, CAULFIELD A M, CHUNG E S, et al. A reconfigurable fabric for accelerating large-scale datacenter services[J]. ACM SIGARCH Computer Architecture News, 2014, 42(3): 13-24.
[19] MAAZOUZ M, TOUBAL A, BENGHERBIA B, et al. FPGA implementation of a chaos-based image encryption algorithm[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(10): 9926-9941.
[20] HEINTZ A, RAZAVIMALEKI V, DUARTE J, et al. Accelerated charged particle tracking with graph neural networks on FPGAs[EB/OL]. [2024-06-01]. https://arxiv.org/abs/2012.01563.
[21] WANG J N, TONG W Q, ZHI X L. Model parallelism optimization for CNN FPGA accelerator[J]. Algorithms, 2023, 16(2): 110.
[22] 刘怡俊, 曹宇, 叶武剑, 等. 基于 FPGA 并行加速的脉冲神经网络在线学习硬件结构的设计与实现[J]. 华南理工大学学报(自然科学版), 2023, 51(5): 104-113.
LIU Y J, CAO Y, YE W J, et al. Design and implementation of online learning hardware structure of pulsed neural network based on FPGA parallel acceleration[J]. Journal of South China University of Technology (Natural Science Edition), 2023, 51(5): 104-113. (in Chinese)
[23] DORIGO A, ELMER P, FURANO F, et al. XROOTD: a highly scalable architecture for data access[J]. WSEAS Transactions on Computers, 2005, 1(4): 348-353.
[24] CHENG Y D, LI H B, BI Y J, et al. Construction and application of LHAASO data processing platform[J]. Radiation Detection Technology and Methods, 2022, 6(3): 418-426.
[25] BRUN R, RADEMAKERS F. ROOT: an object oriented data analysis framework[J]. Nuclear Instruments and Methods in Physics Research, 1997, 389(1/2): 81-86.

文字编辑 索书志
栏目编辑 宋 圆