 DOI: https://doi.org/10.12349/iser.v6i10.7896

High-performance computing checkpoint technology development and application

Chanjuan Liu Wei Jiang Huan Wang

Guangzhou College of Commerce, Guangzhou, Guangdong, 511363, China

Abstract

With the expanding application scale today, the complexity of high-performance computing is also continuously improving. In order to ensure that its fault-t performance meets the actual needs, the application of checkpointing technology should be checked. Based on this, the article takes high-performance computing checkpointing technology as the starting point, briefly the development of high-performance computing checkpointing technology, and analyzes the application of high-performance computing checkpointing technology, mainly including parallel computing fault-tolerance, deep learning-tolerance, HPC scheduling and migration, and FPGA debugging, aiming at providing reference for the future development and application of high-performance computing checkpointing technology.

Keywords

checkpointing technology; HPC system; GPU cluster; FPGA

高性能计算检查点技术发展与应用

刘婵娟 姜微 王欢

广州商学院，中国 · 广东 广州 511363

摘 要

在应用规模不断扩大的今天，高性能计算的复杂程度也在不断提升，为了确保其容错性能满足实际需要，应进行检查点技术的运用。基于此，文章以高性能计算检查点技术作为切入点，简要论述高性能计算检查点技术的发展，并对高性能计算检查点技术的应用进行分析，主要包括并行计算容错、深度学习容错、HPC的调度和迁移以及FPGA的调试，旨在为高性能计算检查点技术的未来发展与应用提供参考。

关键词

检查点技术；HPC系统；GPU集群；FPGA

1 引言

检查点技术作为高性能计算工作中的重要技术，不仅可以对进程状态加以保存，还能在系统出现故障时进行恢复，有效提高应用程序的容错率，确保高性能计算工作的稳定。因此，应做好高性能计算检查点技术的应用研究工作，以便确保程序运行的可靠。

2 高性能计算检查点技术

2.1 系统层检查点工具

系统层检查点工具能够对应用进程状态和上下文环境进行保存，其常见使用的工具较多。其一，BLCR。BLCR 是由伯克利实验室开发的检查点工具，具有用户透明的特点，该工具能够为用户提供 liber 库与 kernel module，进而在其内核开展检查点工作与重启工作，并访问全部内核资源。BLCR 检查点工具还能利用汇编代码对底层硬件信息进行访问，明确其运行状态。其二，DMTCP。DMTCP 是由东北大学研发的检查点工具，该工具不仅可以实现用户透明性的集群计算功能，还能提供跨节点计算的任务迁移支持。在运用该工具执行检查点时，可以对计算组件中的全部进程加以协调。在对进程进行启动时，该工具会优先启动协调器进程，之后利用动态库注入的方式对进程程序进行加载，确保每个进程中形成一个检查点管理器线程，并运用 libdmtep.so 与其他插件对应用程序内的库调用进行截取，进而打造一个与进程信息相关的影子数据库，并对进程状态信息加以收集。除此之外，系统层检查点工具还包括 CRIU、Libckpt、Kekpt 等，可为高性能计算提供支持 [1]。

2.2 应用层检查点工具

应用层检查点工具可以对应用程序状态和上下文信息加以保存，其工具具有多种类型。其一，面向 MPI 计算的检查点工具。MPI 作为 HPC 系统中的消息传递库，可支持并行程序的开发。MPI 程序主要由多个进程加以组成，可

【作者简介】刘婵娟（1987–），女，中国湖南湘潭人，博士，讲师，从事高性能计算，新能源计算模拟研究。

在单片机或机群系统中运行。通过 MPI 应用层检查点工具能够对进程状态加以保存，进而提高 MPI 的容错率。一般而言，技术人员可运用协调协议来进行本地创建，确保检查点具有一致性。其二，面向异构计算的检查点工具。CUDA 与 OpenCL 均为并行编程模型，其中，前者为通用的并行计算架构，一般可应用于 GPU 异构计算工作中。而后者为并行编程语言，能够运行于不同类型的处理器中，为高性能计算提供较高的容错率，确保负载均衡。其三，面向异构存储的多级检查点。SCR 作为能够扩展的检查点，能够将检查点写入并行文件系统、计算节点 RAM、闪存以及磁盘中，具有运行高效、负载轻、成本低等特点。

2.3 高性能计算检查点技术的发展

为了能够满足大规模应用的根本需要，HPC 系统逐步增设更多组件，进而满足实际需求。为了满足大规模应用要求，需要 HPC 系统具有更高的容错率，因此，需要对检查点技术加以应用。对检查点技术来说，可将其分为系统层检查点和应用层检查点，其中，系统层检查点不仅可以对进程的状态进行保存，还能对进程上下文环境加以存储。应用层检查点可以对制定程序状态与上下文信息进行保存，该功能的使用需要相应的编程语言或模型加以支持。在 20 世纪 80 年代，威斯康星大学麦迪逊分校开发了异构分布式系统，该系统能够对进程进行任务迁移，并运用检查点技术可以实现集群主观能的进程级任务迁移，保证负载均衡，进一步提高其资源使用率。在 2005 年至 2013 年之间，很多系统层检查点工具逐步出现，如，BLCR、DMTCP、CRIU 等，同时，SCR、VeloC 等应用层检查点工具也相继得到使用。在 HPC 系统中，其应用主要是将 MPI 作为并行模式，其中，MPI3.1 标准并未解决其存在的进程故障问题，而 ULFM 能够保证 MPI 程序从故障状态进行恢复，进而避免出现程序重新启动问题，有效降低高性能计算成本。虽然 ULFM 可以提高容错率，但很难将其直接应用于一些程序中，因此，技术人员还应设置基于并行编程的检查点机制，从而提升其容错率。当多功能存储系统逐步出现，多级检查点技术能够提高资源使用率，但会给 HPC 系统带来较大的 I/O 口流量。为了降低 I/O 口开销，需要对检查点性能加以优化。随着 AI 技术与 HPC 系统之间的有效融合，技术人员在超算中心设置了 GPU 集群，因此，怎样对异构计算资源进行管理十分关键，如何减少检查点成为当下重要研究方向，以适应高性能计算的未来发展 [2]。

3 高性能计算检查点技术的应用

3.1 并行计算容错

对 HPC 系统而言，为了达到提高系统性能的目的，应持续增加系统的计算节点，并丰富其核心数量，但系统的长时间稳定运行与 MPI 应用程序易被节点故障问题带来负面影响，若缺乏有效的容错技术，很容易导致系统计算结果数据丢失，且重新进行计算具有较高的经济成本。因此，技术人员可运用检查点技术与回滚恢复技术使系统能够从故障状态恢复为一致状态。目前，将弹性算法融合于已有编码中具有较大困难，不仅需要解决弹性算法本身的复杂问题，还应将独立的新策略与原有的容错技术加以结合。为确保并行应用程序的检查点保持一致，应借助回滚恢复算法加以实施。其一，非协调检查点技术。当系统出现故障问题时，若是运用非协调检查点技术加以修复，需要在日志文件内部保留信息痕迹，才能修复到之前状态，以免造成多米诺效应。该检查点技术可以单独保留每一个进程的实际状态，能够对消息日志的大小加以控制。若是该技术以悲观回滚恢复协议作为基础时，会导致延迟性增加，该问题主要是由于每个节点之间网络带宽存在差异所导致，所以在多核系统中接收器的信息保存能力会变得更低。其二，协调检查点技术。若是使用协调检查点技术进行恢复时，可将全部进程恢复成全局状态。因 I/O 存在拥堵问题，通过对检查点的协调会造成程序执行速度变慢。因此，当系统出现故障问题时，所有程序进程会由最后检查点开始重新启动。当进程数量不断增多时，会重新执行很多计算任务，进而导致计算成本较高。其三，半协调检查点技术。若是使用半协调检查点技术，能够有效降低容错强制开销的数量，只需对进程组之间的相互作用进行保留即可，可应用于规模增长方面。总之，运用检查点技术可以提高并行计算容错率，有效减少资源消耗，降低计算成本 [3]。

3.2 深度学习容错

在高性能计算过程中，深度学习作为其中的重要形式，需要运用海量数据集与深度神经网络，对 HPC 系统具有较高要求。在深度学习过程中进行检查点技术的运用十分关键，不仅可以解决容错问题，还能确保模型训练时的忠实重放，不仅可以有效提高模型的基本性能，还能提高其生产率，进一步提升其鲁棒性，有效促进安全审计工作的开展。很多深度学习模型训练过程中会运用根检查点技术，此种技术会因前进进度滞后与阻塞语义带来负面影响。为解决此问题，可运用多检查点技术进行模型训练，不仅可以提高其扩展性，还能在特有的框架下加以更改。对深度学习模型训练而言，其非确定性特点为终始重放带来较大困难，即便运用较为固定的随机种子进行训练，也会导致同一个训练管道内部的不同运行性能之间存在较大差异，且使用原有的基础架构无法对训练过程进行重视重放。为解决此问题，可在其中加入随机数生成机制，该技术能够在数据并行计算的基础上生成一致的随机数，并运用较为新颖的分析方法对忠实重放需要的一组状态变量进行确定，且该变量可以保存于检查点内，也可利用选择性执行技术重新生成。总而言之，在高性能计算工作中，技术人员可运用检查点技术对深度学习模型训练提供支持，能实现确定性重放目标，有效提高深度学习容错计算效果 [4]。

3.3 HPC 的调度和迁移

现阶段，在 HPC 集群中开展的作业需要和开始运行节点之间进行绑定，从而对集群资源的实用性交流与调度带来较多限制。与此同时，随着 AI 技术的快速发展，将其与高性能计算加以结合，从而导致很多 GPU 资源被应用于高性能计算工作中，使其运行作业会存在较多动态变化。当出现用户数目、硬件状态、作业规模等出现变化时，会造成运行时间得以延长，使资源的使用率逐渐降低。为有效解决该问题，技术人员可运用检查点技术来进行 HPC 的调度与迁移，有效提高资源使用率。对 GPU 集群资源管理工作来说，主要是通过静态资源隔离与划分的方式进行管理，无法提高其资源使用率。因此，技术人员可打造开放的资源管理机制，实现跨用户调度作业资源来提升其资源使用率。为达到此目标，可将用户级检查点机制嵌入到资源管理器内部，可以让作业的迁移过程变得更加透明，为调度工作奠定基础，进一步提升其容错性。此种基础的设置对将来超大规模的 HPC 集群十分关键，随着调度的复杂程度不断提升，对其并行程度具有更高要求。高性能计算中会因作业规模及服务能力的变化导致完成任务的时间得以延长，为有效解决该问题，可引入分析方法来进行检查点调度算法的设计，可以在多任务情况下降低作业时间。与此同时，技术人员还可利用虚拟化技术提高资源使用率，运用远程 GPU 虚拟化技术来提升其灵活性。除此之外，技术人员利用迁移技术可以将 GPU 部分迁移至其他 GPU 处，实现不同用户 GPU 的动态分配，进一步提高集群的资源利用率，还能确保性能隔离与公平。总之，利用检查点技术可以进行 HPC 的调度与迁移，有效提高其资源利用率。

3.4 FPGA 的调试

FPGA 和 HPC 之间具有较为紧密的关联，FPGA 不仅具有良好的可重配置性能，还拥有高效性优势，进而成为 HPC 系统的协处理器与加速器，还能对该系统进行建模、评估等工作，是 HPC 系统的基础。对 FPGA 而言，调试工作的效率十分重要，当下 FPGA 项目的 51% 时间主要是进行调试与验证。同时，当 FPGA 项目的设计变得越来越复杂时，用于调试和验证的时间也会变得越来越多。在调试工作中，主要是在仿真叶片上进行调试，此种方法虽然具有一定的可观察性，但无法找到故障的实际原因。基于此，需要对具有可观察能力的软件加以开发。State Mover 作为一个能够在 FPGA 与模拟器之间加以移动的调试器，能够对设计状态加以移动，可以通过全硬件速度加以运行，直至达到感兴趣的点。但该调试器无法对具有外部 I/O 接口的系统进行调试，无法将片外状态移动至模拟器中。State Link 作为一个基于事务的系统仿真框架，能够保证系统的一部分在模拟器中加以运行，还能保证其硬件和其他系统组件之间加以交互，该系统仿真框架可以将其状态移动至模拟器中，并保证硬件部分处于活跃状态，可以对具有 I/O 接口的系统进行调试，有效提高其仿真速度。总之，在进行 FPGA 的调试过程中，可运用 State Link 等基于检查点的协同仿真技术加以调试，从而降低其验证难度，助力 FPGA 产品的研发 [5]。

4 结语

综上所述，在高性能计算工作中，检查点技术的应用具有重要意义，不仅可以提高程序的容错性，还能降低计算成本。检查点技术主要包括系统层检查点工具和应用层检查点工具两大部分，可将其应用于并行计算容错、深度学习容错、HPC 的调度和迁移、FPGA 的调试等工作中，从而确保高性能计算工作的稳定。在未来，可对检查点工具进行优化，扩大其检查范围，拓展其应用领域，为高性能计算提供技术支持。

参考文献

[1] 刘扬,许建飞,许黄超,吴璨,胡泰源,原惠峰,高凌云,梁文昊,董盛,马英晋,李瑞琳,赵永华.基于超级计算机的高性能计算应用发展现状及趋势研究[J].数据与计算发展前沿(中英文),2025,7(2):68-85.

[2] 杨敏,何芸,许涛,景少军.高性能GPU计算集群应用体系建设[J].信息系统工程,2025(3):102-105.

[3] 郑宏兴.“高性能计算在电波传播研究中的应用”专题前言[J].电波科学学报,2025,40(3):405-405.

[4] 陈筱琳,张亚强,史宏志.面向多样计算场景的检查点技术综述[J].计算机应用,2025,45(6):1922-1933.

[5] 陈轶阳,王小宁,闫晓婷,李冠龙,赵一宁,卢莎莎,肖海力.基于CRIU的高性能计算容器检查点技术研究[J].计算机科学,2024,51(9):40-50.