深度学习模型训练过程检查点访问性能优化方法

滕云[1]，张广艳[2]，孙大为[1]，田海东[3]，常锐[3]
1. 中国地质大学（北京）人工智能学院，北京 100083；
2. 清华大学计算机科学与技术系，北京 100084；
3. 中兴通讯股份有限公司，江苏 南京 210012

摘要
随着大模型应用越来越广泛、规模逐渐增大，目前大模型训练面临出错概率高、检查点访问性能差等问题。总结了已有检查点访问性能优化方法的优缺点，提出了一种新的检查点访问性能优化方法。观察检查点数据模式可知，相近检查点的模型权重数据变化较小，适合增量压缩。基于多台互联训练节点实现了增量压缩，并基于真实的深度学习模型训练时产生的检查点数据进行了实验测试。结果表明，在训练周期内，增量压缩对大多数检查点具有较好的压缩效果。此外，提出在增量压缩中使用动态间隔来平衡压缩率与存储开销，并对动量数据特征进行分析。文章对已有方法的分析及对检查点访问性能的优化为大模型训练加速提供了指导。

关键词
大模型；检查点；数据压缩；性能提升

中图分类号：TP302 文献标志码：A doi：10.11959/j.issn.2096-0271.2026029

Checkpoint accessing performance optimization method for the deep learning model training process

Teng Yun[1], Zhang Guangyan[2], Sun Dawei[1], Tian Haidong[3], Chang Rui[3]
1. School of Artificial Intelligence, China University of Geosciences Beijing, Beijing 100083, China
2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
3. ZTE CORPORATION, Nanjing 210012, China

Abstract
As LLMs become more widely used and their scale continues to expand, currently LLMs training faces issues such as high error rate and poor performance of checkpoint accessing. This paper reviews the strengths and weaknesses of existing methods for optimizing checkpoint accessing performance and introduces a novel method for optimizing checkpoint accessing performance. Based on the observation of data patterns in checkpoints, where the model weights change between adjacent checkpoints are minimal, making them suitable for delta compression. The proposed method implements delta compression across multiple interconnected training nodes and conducts experimental tests using real checkpoints generated during deep learning model training. The results demonstrate that, during the model training,

delta compression has good compression effect for most checkpoints. Furthermore, the paper introduces dynamic intervals in delta compression to balance compression ratio and storage overhead, while also analyzing the characteristics of momentum datas. The analysis of existing methods and the optimization of checkpoint accessing performance offer insights for accelerating LLMs training.

Key words
LLM, checkpoint, data compression, performance improvement

0 引言

近年来，大模型的应用日益广泛，其参数量不断增加，训练数据规模持续增大。然而，大模型训练过程中仍面临核心挑战——在大规模分布式训练中大模型出错概率高、检查点的存储与访问性能差。这些问题严重制约了训练效率和资源利用率。针对检查点访问性能优化这一关键问题，本文系统梳理了已有各类优化方法的优势与不足，并在此基础上提出了一种新的检查点访问性能优化方法。在已有优化检查点访问性能的研究中，基于调度策略优化检查点访问性能的方法无法解决单个检查点数据量大的问题，传统的数据压缩方法没有考虑检查点的数据特性，因此压缩效果有限。为弥补已有方法的不足，本文设计并实现了基于多台互联训练节点的增量压缩方法，利用真实深度学习模型训练任务生成的检查点数据进行了详尽的实验评估。实验结果表明，在完整的训练周期内，大多数的增量压缩能取得良好的压缩效果，显著减少了存储和传输开销。此外，本文提出在增量压缩中使用动态间隔来均衡压缩率与存储开销，并对动量数据的特征进行分析。本文对已有方法进行深入剖析、对检查点访问路径进行优化设计，旨在为大模型训练的高效加速策略提供切实可行的理论指导和实践参考。

1 研究背景及相关工作

1.1 研究背景

目前人工智能技术广泛应用于多个领域，如图像检测、博弈类决策、自然语言处理等[1]。人工智能技术和深度学习[2]的逐步发展，使大模型的应用范围逐渐变大。2019年2月推出的GPT-2的参数规模达15亿，而2020年推出的GPT-3的参数规模则达1 750亿，由此可见大模型是未来的研究热点，大模型时代即将到来[3]。

大模型拥有较强的数据处理能力和较高的预测精度[4]，成为推动人工智能技术突破的重要力量。大模型具有较庞大的参数和较复杂的结构，能够学习较多的数据特征和模式，从而在许多任务中表现出较好的性能。随着计算硬件的发展和分布式训练技术的应用，模型参数规模的扩大变得更加容易[5-7]。大模型通常能够处理较大规模的数据集、输出较精确的预测结果和具有较好的泛化能力。大模型在许多领域取得了显著的成果。例如，使用大型卷积神经网络（convolutional neural network，CNN）模型，计算机视觉任务中的图像分类、目标检测和图像生成等研究取得了重大突破。在自然语言处理领域，Transformer模型的引入显著提升了机器翻译、文本生成和语言理解等任务的效率。

1.2 存在的问题

目前大模型应用广泛，与人类的生产生活息息相关。随着相关研究不断取得进展，大模型应用的发展趋于更多参数、更高精度，但更大的模型意味着需要存储更多的数据，这给数据存储带来了挑战，普通的集群难以提供让大模型正常工作的存储空间和性能，因此不能忽视对大模型的存储优化。

大模型的参数规模正在不断扩大，PaLM具有540×10^9的参数量，相比于GPT-2增长了360倍，如此庞大的参数量意味着模型训练需要海量的计算资源。在实际的训练集群中，研究者大多采用分布式训练提高模型的训练速度与性能[8-9]。高计算量会增加计算时间，从而导致大量的软硬件错误。以10万卡规模的“神威·太湖之光”人工智能平台训练万亿参数量的模型为例，训练过程中平均每小时会发生一次错误，这种万亿级参数量模型需要存储的模型参数检查点约为12 TB，同时写入检查点需要耗费大量时间，未经优化的一次写检查点需要3 h。大模型训练过程中频繁发生的软硬件错误受到越来越多学者的重视。

此外，软硬件错误的发生往往会带来巨大的恢复开销，OPT-175B在训练中发生了软硬件错误，造成了178 000 h的GPU时间损失[10]。训练发生错误时，可以通过恢复检查点重新训练，检查点保存了大模型在某个时刻的参数信息，发生错误时不需要从头训练，只需要加载最近一个保存的检查点即可继续训练，如**图1**所示。在模型训练一段时间后，将当前的权重和优化器状态持久化；当出现错误时，加载最近一次保存的检查点并开始继续训练。在频繁出错的大规模训练场景下，检查点对训练十分重要，对检查点数据的快速读写将减少检查点过程对训练过程的干扰，提高检查点访问的性能变得更加重要。因此，本文针对已有方法进行分析，总结其优缺点，并提出新的优化方法和进一步优化的方向。

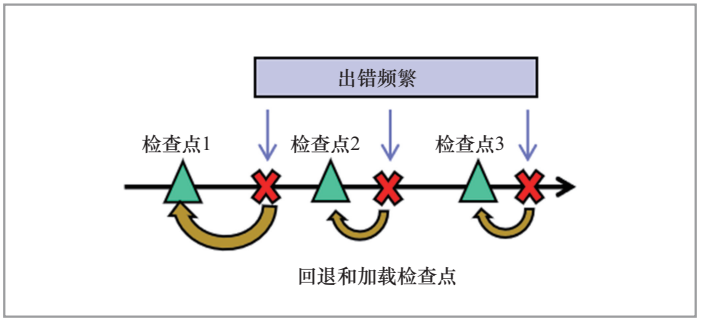

图1 检查点出错恢复

2 已有检查点访问性能优化方法

2.1 基于调度策略的检查点访问性能优化方法

当前存在一些方法利用异步策略优化检查点写入性能，如DeepFreeze[11]实现了异步保存深度神经网络（deep neural network，DNN）检查点，但它只考虑CPU集群。当使用较先进的图形处理单元（graphics processing unit，GPU）进行训练时，它不考虑在内存中复制模型状态的成本。已有方法表明，在现代机器学习优化的服务器上，从GPU复制到CPU的成本是显著的，这些方法通过流水线计算和传输，并使用备用GPU资源来实现快速快照。此外，DeepFreeze需要人工干预来调整给定模型、硬件、软件和训练环境的检查点频率，而CheckFreq[12]避免用户设置这些参数，能够自动分析识别检查点的最佳参数。与使用静态检查点频率的

DeepFreeze不同，CheckFreq在共享集群设置中比较有效，因为它根据其他作业造成的内存和存储干扰来调整检查点频率，以最大限度地减少检查点为训练过程带来的延迟。先前在高性能计算（high-performance computing，HPC）中的工作使用异步检查点来掩盖I/O延迟。DNN检查点与传统HPC检查点的一个关键区别在于，由于GPU的计算能力越来越强，其从GPU到CPU执行同步模型状态的内存复制开销较大。CheckFreq通过在可能的情况下利用空闲的GPU内存和计算能力来执行快速快照，进一步减少了检查点的延迟。其基于故障预测模型规则，在高性能计算应用[13]中使用自适应的思想来决定检查点的时间。

然而，上述方法都是基于调度或已有架构进行检查点访问性能优化的，仍需要将检查点写入持久性存储介质中，访问速度较慢；如果使用访问速度较快的内存存储检查点，可靠性较低，如何综合利用内存的高速访问特性和底层存储的持久性提供更好的检查点读写性能是值得研究的问题。

2.2 基于内存介质的检查点访问性能优化方法

内存存储相对于底层持久性存储具有高带宽、低时延的特点，在近些年的研究和实际生产应用中，内存存储逐渐受到了广泛青睐，究其原因主要有两个：一是内存存储具有更高的带宽，二是内存存储具有更低的访问时延。一种可行的检查点访问性能优化方法是利用内存介质存储检查点数据，相对于传统检查点技术，其增加了缓存层，是对传统检查点技术的一种优化。因此Gemini[14]考虑基于内存介质的检查点策略，为了提高检查点读取性能，在内存中存储检查点副本，当存在软件错误时，读取本地内存中的检查点副本；当出现硬件错误时，使用其余节点中的副本恢复检查点；当两个节点中的副本都不可用时，从远端持久存储中拉取检查点进行恢复。然而该方法只将检查点数据以多副本的形式存储在单个节点中，读写带宽有限。

相比于两个节点之间的数据传输，已有方法针对内存的并行读写去容错，充分利用并行性进行并行读写。EC-Cache[15]通过额外发送一定数量冗余的请求，选择最快返回的k个请求，得到整个纠删码条带的数据；ECWide-H[16]基于内存KV存储的纠删码存储系统，将写入的数据发送到多个内存节点的内存中。

2.3 基于数据压缩的检查点访问性能优化方法

随着模型规模的增大，检查点的存储和传输成为瓶颈，分别对底层存储空间和节点间传输效率产生影响。检查点的数据规模随着模型规模的增大而增大，因此使用数据压缩减少数据量能够提高检查点访问速度。传统的数据压缩方法可以分为两类，一类是高压缩率方法（如GZIP、LZMA和ZSTD），另一类是高速压缩方法（如Snappy、LZ4[17]）。压缩算法的选择取决于应用场景的需求。如果需要实现高压缩率可以选择GZIP或LZMA等压缩方法，这些方法在文件压缩和网络传输的应用中较常见。但是这些压缩方法都是针对特定的场景而设计的，包括数据使用场景和数据特点等，不适用于检查点数据。

一些方法使用数据压缩技术对检查点数据进行优化，Check-N-Run[18]主要针对推荐模型，采用差异化检查点技术，仅存储模型中修改过的部分模型参数，并结合量化技术来减少浮点数的表示位数，进

而减小检查点的大小。通过这种增量检查点技术，Check-N-Run能够在特定类型的模型（如推荐模型）中应用，因为这些模型并不会在每个小批量（mini-batch）中更新所有参数。但是，Check-N-Run的差异化检查点技术不适用于大模型的预训练，因为大模型通常表现出密集的更新模式，几乎所有参数都会被频繁更新。Just-In-Time Checkpointing[19]在深度学习训练过程中发生错误时进行即时检查点，将浪费的GPU工作限制为仅一个小批量的重放，避免了频繁检查点带来的额外成本。但是该方法在真实的大规模GPU集群中表现出脆弱性，当同一备份组中的多台机器同时发生故障时，可能会导致不可接受的灾难性状态丢失。一些方法对检查点进行量化压缩，如QD-Compressor[20]，然而这会对模型的精度产生影响。

上述方法没有考虑检查点数据的特性。本文对模型训练的检查点数据进行分析，制定匹配检查点数据特性的压缩策略。

3 检查点访问性能优化方法以及未来优化方向

3.1 检查点数据规律观察

深度学习模型训练时产生的检查点包含大量的数据，主要包括两个部分：模型权重数据和优化器动量数据。在模型逐渐收敛的过程中，相邻的检查点中权重的参数变化较小，因此可以考虑使用增量压缩方法记录两个检查点权重的差值；而优化器动量数据不具备这样的规律，变化较随机，但是动量数据本身数值较小，因此在二进制表示时，会有较多相似前缀，可以将这些相似前缀表示成索引来简化其二进制的表示。

3.2 增量压缩优化检查点访问性能

在训练过程中，模型权重将根据梯度在前一个检查点的基础上进行更新，相邻模型权重之间的差异大多较小[21]，因此使用增量压缩可能会获得较大收益。首先，增量压缩需要存储第一个完整的模型权重数据，将其作为增量压缩的基准文件。增量压缩仅存储第一个版本的全量检查点作为基准，存储第二个检查点，增量压缩采用异或操作，计算其与第一个检查点对应的模型参数的差值。类似地，所有检查点均计算与上一个版本检查点的异或值，该值即增量。具体的异或方式如**图2**所示，将两个权重矩阵对应位置的参数使用异或计算增量结果，增量结果构成了与权重矩阵相同规模的增量矩阵，将增量矩阵存储到对应的存储节点中；为了计算与上一轮检查点模型权重的差值，需要将第二个检查点的权重存入内存，此时第一个检查点的内容可以从内存中删除；当检查点逐步生成时，当前检查点均与上一个检查点计算差值。

为了验证增量压缩是否可行，本文统

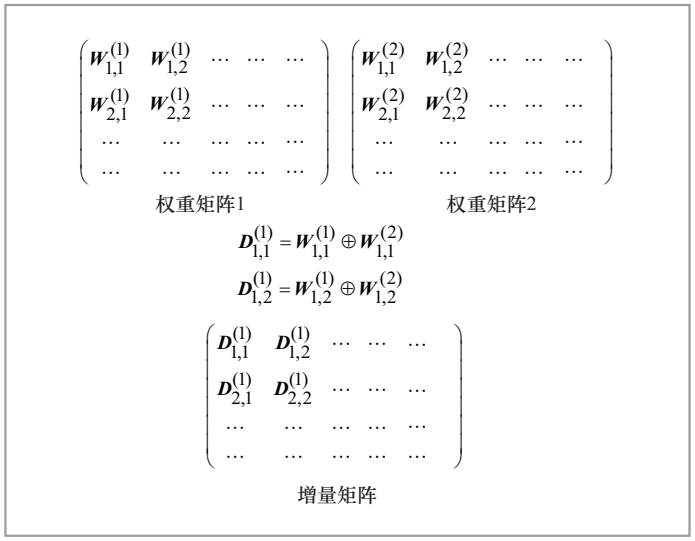

图2 权重矩阵增量计算

计了多个模型训练产生的检查点数据的压缩效果，其中包括大模型GPT-2。本文定义压缩率为原始数据量除以压缩后的数据量，即压缩率越高越好。**图3**基于4种模型训练生成的检查点数据下，统计了权重数据增量压缩的压缩率，在所有模型的检查点数据中，增量压缩的压缩率均高于1.3，其中GPT-2的压缩率最高，达3.23，该结果表明增量压缩对于权重数据具有较好的压缩效果。此外，为了验证增量压缩相对于传统压缩方法的优势，本文对比了通用压缩方法GZIP，测试了GZIP压缩方法的压缩率，在多种模型检查点下，GZIP的压缩率均不超过1.1（如**图3**所示），说明传统压缩方法没有充分考虑检查点的数据特性，导致压缩效果有限。

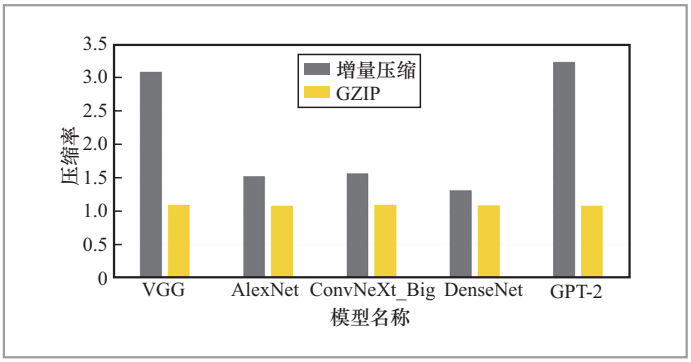

图3 多种模型下权重数据压缩率

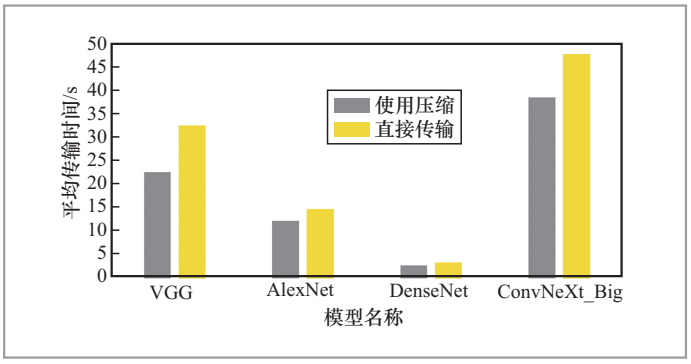

图4 检查点平均传输时间对比

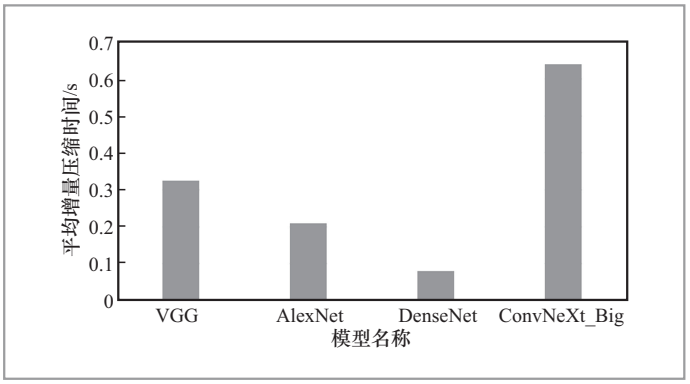

图5 检查点平均增量压缩时间对比

除了考虑压缩降低的数据量，也需要考虑压缩带来的时间开销。作为初步探索，本文搭建了4台机器间的权重数据传输环境，4台机器单独训练模型并向其他节点传输检查点，本文测试了检查点压缩前后4台机器的平均传输时间，并测试了4台机器的平均增量压缩时间，如**图4**、**图5**所示。相比于压缩节省的传输时间，压缩时间可以忽略不计，说明增量压缩方法可用性较强。同时，在解压过程中，解压时间与压缩时间相差不大。

图6统计了在连续的训练轮次中进行增量压缩的压缩率变化，表明随着训练轮次逐渐增加，增量压缩的压缩率趋于稳定并保持较高水平，说明在整个训练过程中，增量压缩的有效范围较大，收益比较明显。

3.3 优化器动量数据访问性能优化

根据已有工作分析[21]，优化器动量数据包括每个参数的一阶矩和二阶矩，其数值较小，因此在二进制表示过程中前缀较相似。为了验证这一特性，本文在ConvNeXt_Big的检查点下测试了优化器动量数据不同前缀长度的种类数量，如**图7**所示。当前缀长度从7位增长到9位

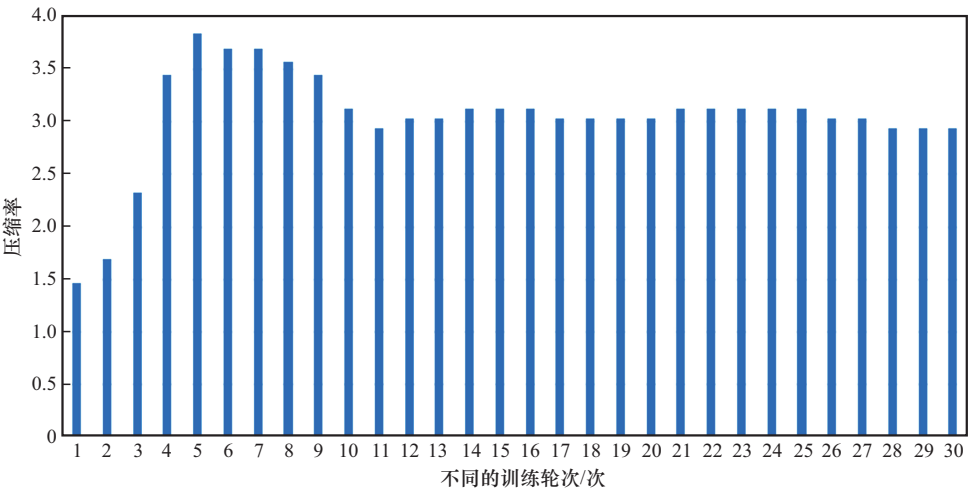

图6 不同训练轮次的增量压缩效果

时，前缀种类数量从28种增长到101种，说明前缀种类较少。

上述现象是因为32位浮点数前缀由符号位和指数位构成，如果数值较小，其指数位差别不明显，大量数据具有相同的前缀，因此可以将前缀编码为更短的位数表示，以降低优化器动量数据的大小。本文统计了不同模型下的优化器动量数据压缩率，如**图8**所示，压缩率最高可达1.22。

此外，本文根据优化器动量数据的相似性，将前缀相似程度高的数据分为一组。最佳情况是每一组的前缀只有一种，编码时可以用1位的前缀长度表示该前缀，进一步减小优化器动量数据的占用空间。

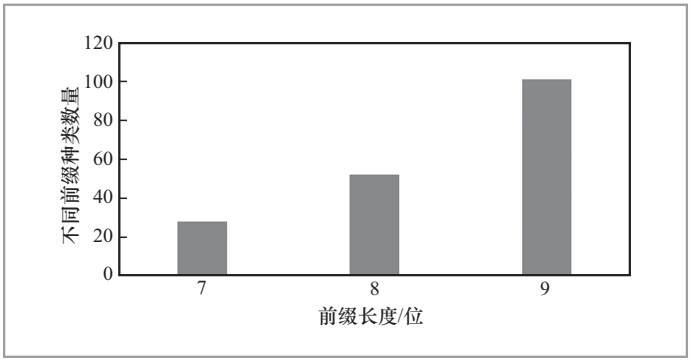

图7 优化器动量数据的前缀种类数量

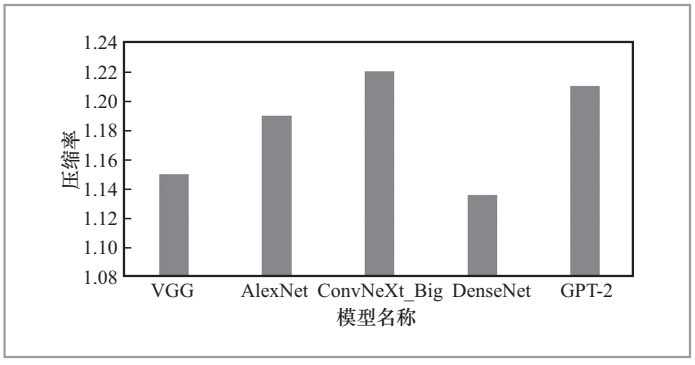

图8 优化器动量数据的压缩率

3.4 增量压缩动态间隔优化

为了减少存储端的增量开销，设置动态的检查点增量压缩间隔是必要的。默认的增量压缩方法是将第K个检查点与第$K-1$个检查点计算增量，当恢复第K个检查点时，需要读取第一个基准检查点及$K-1$个增量文件，其大小远超单个检查点，因此可以考虑调整计算增量的间隔T，即将第K个检查点与第$K-T$个检查点计算增量，

减小合并增量的次数，T值的选择需要根据压缩率的变化幅度确定，既要满足开销减少的要求，又要保证压缩效果没有明显降低，当T值增大时，进行增量计算的检查点间隔增大，恢复时参与计算的检查点数量减少，减小恢复开销，然而在计算增量时，更大的间隔会导致参数变化增大，降低压缩率。

4 结束语

本文主要分析了当前广泛使用的大模型中检查点访问性能差的问题，分类总结了已有方法，并分析其不足；从数据压缩角度，进行数据特征观察并给出了检查点访问性能优化方法和未来可能优化的方向。本文主要总结了3类检查点访问性能优化的方法：基于调度策略的检查点访问性能优化方法、基于内存介质的检查点访问性能优化方法和基于数据压缩的检查点访问性能优化方法。本文对基于数据压缩的检查点访问性能优化方法进行分析，实现了增量压缩减少存储和传输开销，并给出了未来可能的优化方向：首先，根据已有工作和本文实验测试的结果，证明了增量压缩在模型训练的权重数据上具有较高的压缩率，并统计了在训练不同过程中压缩效果的动态变化，说明了增量压缩的可行性和有效性；其次，给出了优化器动量数据压缩可能的优化方向。本文为未来的大模型检查点访问性能优化提供了方向指引。

参考文献：

[1] 张鹤译，王鑫，韩立帆，等. 大语言模型融合知识图谱的问答系统研究[J]. 计算机科学与探索，2023，17(10)：2377-2388.
Zhang H Y，Wang X，Han L F，et al. Research on question answering system on joint of knowledge graph and large language models[J]. Journal of Frontiers of Computer Science & Technology，2023，17(10)：2377-2388.

[2] 韩炳涛，刘涛，唐波. 深度学习的10年回顾与展望[J]. 中兴通讯技术，2022，28(6)：75-84.
Han B T，Liu T，Tang B. Deep learning：past decade and future[J]. ZTE Technology Journal，2022，28(6)：75-84.

[3] 桑基韬，于剑. 从ChatGPT看AI未来趋势和挑战[J]. 计算机研究与发展，2023，60(6)：1191-1201.
Sang J T，Yu J. ChatGPT：a glimpse into AI's future[J]. Journal of Computer Research and Development，2023，60(6)：1191-1201.

[4] 田海东，张明政，常锐，等. 大模型训练技术综述[J]. 中兴通讯技术，2024，30(2)：21-28.
Tian H D，Zhang M Z，Chang R，et al. A survey on large model training technologies[J]. ZTE Technology Journal，2024，30(2)：21-28.

[5] Brown T B，Mann B，Ryder N，et al. Language models are few-shot learners[C]// Proceedings of the Annual Conference on Neural Information Processing Systems 2020. California：NIPS，2020：1877-1901.

[6] 李戈，彭鑫，王千祥，等. 大模型：基于自然交互的人机协同软件开发与演化工具带来的挑战[J]. 软件学报，2023，34(10)：4601-4606.
Li G，Peng X，Wang Q X，et al. Challenges from LLMs as a natural language based human-machine collaborative tool for software development and evolution[J]. Journal of Software，2023，34(10)：4601-4606.

[7] 鞠天杰，刘功申，张倬胜，等. 自然语言处理中的探针可解释方法综述[J]. 计算机学报，2024，47(4)：733-758.
Ju T J，Liu G S，Zhang Z S，et al. A re-

view of probe interpretable methods in natural language processing[J]. Chinese Journal of Computers, 2024, 47(4): 733-758.

[8] Lian X, Jacobs S A, Kurilenko L, et al. A flexible and efficient distributed checkpointing system for large-scale DNN training with reconfigurable parallelism[C]//Proceedings of the USENIX Annual Technical Conference (USENIX ATC 25). Berkeley: USENIX Association, 2025: 1519-1534.

[9] Strati F, Friedman M, Klimovic A. PCcheck: persistent concurrent checkpointing for ML[C]//Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1. New York: ACM, 2025: 811-827.

[10] Zhang S S, Roller S, Goyal N, et al. OPT: open pre-trained transformer language models[PP]. arXiv preprint, 2022, arXiv: 2205.01068.

[11] Nicolae B, Li J L, Wozniak J M, et al. DeepFreeze: towards scalable asynchronous checkpointing of deep learning models[C]//Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). Piscataway: IEEE Press, 2020: 172-181.

[12] Mohan J, Phanishayee A, Chidambaram V. CheckFreq: frequent, fine-grained DNN checkpointing[C]//Proceedings of the 19th USENIX Conference on File and Storage Technologies. Berkeley: USENIX Association, 2021: 203-216.

[13] Lan Z L, Li Y W. Adaptive fault management of parallel applications for high-performance computing[J]. IEEE Transactions on Computers, 2008, 57(12): 1647-1660.

[14] Wang Z, Jia Z, Zheng S, et al. Gemini: fast failure recovery in distributed training with in-memory checkpoints [C]//Proceedings of the 29th Symposium on Operating Systems Principles. New York: ACM, 2023: 364-381.

[15] Rashmi K V, Chowdhury M, Kosaian J. EC-Cache: load-balanced, low-latency cluster caching with online erasure coding[C]//Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2016: 401-417.

[16] Hu Y, Cheng L, Yao Q. Exploiting combined locality for wide-stripe erasure coding in distributed storage[C]//Proceedings of the 19th USENIX Conference on File and Storage Technologies. Berkeley: USENIX Association, 2021: 233-248.

[17] Beneš T, Bartík M, Kubalík P. High throughput and low latency LZ4 compressor on FPGA[C]//Proceedings of the 2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig). Piscataway: IEEE Press, 2020: 1-5.

[18] Assaf E, Kiran K M, Steven I, et al. 2022. Check-N-Run: a checkpointing system for training recommendation models[C]//Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). Berkeley: USENIX Association, 2022: 929-943.

[19] Gupta T, Krishnan S, Kumar R, et al. Just-In-Time checkpointing: low cost error recovery from deep learning training failures[C]//Proceedings of the Nineteenth European Conference on Computer Systems. New York: ACM, 2024: 1110-1125.

[20] Zhang S Y, Wu D L, Jin H Y, et al. QD-Compressor: a quantization-based delta compression framework for deep neural networks[C]//Proceedings of the 2021 IEEE 39th International Conference on Computer Design (ICCD). Piscataway: IEEE Press, 2021: 542-550.

[21] Li W S, Chen X H, Shu H, et al. ExCP:

extreme LLM checkpoint compression via weight-momentum joint shrinking[C]// Proceedings of the 41st International Conference on Machine Learning (ICML 2024). New York: ACM. 2024: 27575-27588.

作者简介

滕云（1998- ），男，中国地质大学（北京）博士生，主要研究方向为存储系统和分布式处理。

张广艳（1976- ），男，博士，清华大学计算机科学与技术系副教授、博士生导师，中国计算机学会杰出会员，国家杰出青年科学基金获得者，主要研究方向为大数据计算、存储系统和分布式处理。

孙大为（1985- ），男，博士，中国地质大学（北京）人工智能学院教授、博士生导师，主要研究方向为大数据计算、分布式系统。

田海东（1981- ），男，中兴通讯股份有限公司先进计算存储架构师，主要研究方向为计算存储系统架构演进、存算一体及异构内存体系。

常锐（1978- ），男，中兴通讯股份有限公司先进计算存储架构师，主要研究方向为大模型训推系统、数据流支撑平台、安全可信计算。

收稿日期: 2025-11-06

通信作者: 张广艳，gyzh@tsinghua.edu.cn

基金项目: 国家自然科学基金项目(No.62025203)

Foundation Item: The National Natural Science Foundation of China (No.62025203)