Yun TENG, Dawei SUN, Shipeng HU, Zhiyue LI, Guangyan ZHANG, Haidong TIAN, Rui CHANG, 2026. FastCheck: fast checkpointing and recovery for DNN training via parallel transmission and compression. *ENGINEERING Information Technology & Electronic Engineering*, 27(2):250034. https://doi.org/10.1631/ENG.ITEE.2025.0034

FastCheck: fast checkpointing and recovery for DNN training via parallel transmission and compression

Key words: Deep neural network models; Critical failures; Parallel transmission; Data compression; Checkpointing and recovery

Corresponding author: Guangyan ZHANG
E-mail: gyzh@tsinghua.edu.cn
ORCID: https://orcid.org/0000-0002-3480-5902

Motivation

- In recent years, with the rapid evolution of deep neural networks (DNNs), especially the pretrained large-scale language models (commonly known as LLMs), the number of model parameters has continuously increased. Training large-scale DNNs is prone to software and hardware failures, with critical failures often requiring full-machine reboots that substantially prolong training.
- Existing checkpointing/recovery solutions either cannot tolerate such critical failures or suffer from slow checkpointing and recovery due to constrained input/output (I/O) bandwidth. How to reduce checkpointing time and substantial recovery overhead caused by critical failures remains to be explored.

Main idea

- Multiple nodes are used to construct triple-replica placement strategy for checkpoints, multiple checkpoint shards are partitioned, and these shards are transmitted to mutiple nodes in parallel to reduce checkpointing and recovery time.
- Two compression algorithms to reduce the huge size of checkpoints: delta compression for weights and index compression for momentum.
- Lightweight and consistent health status maintenance is designed to ensure consistent tracking of node health across nodes. This protocol avoids transmitting to failed nodes, with small extra overhead.

Method

1. Each checkpoint is evenly partitioned into *N* data shards and transmitted in parallel to the memory of *N* distinct storage nodes. To improve fault tolerance, an additional parity shard is generated via XOR across the *N* data shards.

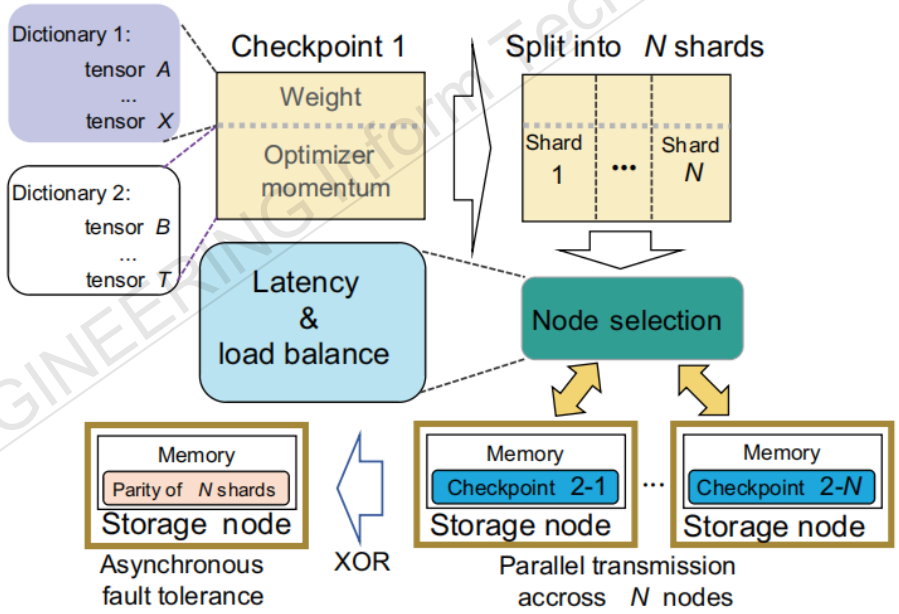

Fig. 1 Partitioning and transmission in parallel

Method (Cont'd)

2. FastCheck implements distinct compression strategies for model weights and optimizer momentum.

1) For weights, the limited changes between adjacent checkpoints make them suitable for delta compression.

2) For momentum due to a highly similar binary form of prefixes, index compression is used by representing the prefixes of momentum as indexes.

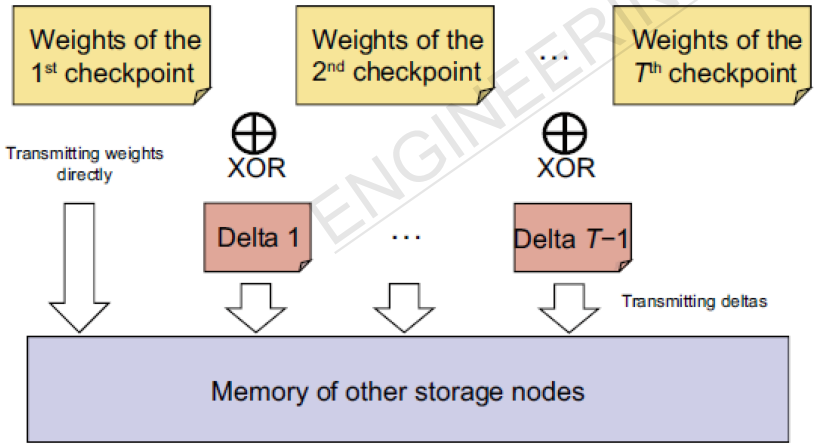

Fig. 2 The process of delta compression

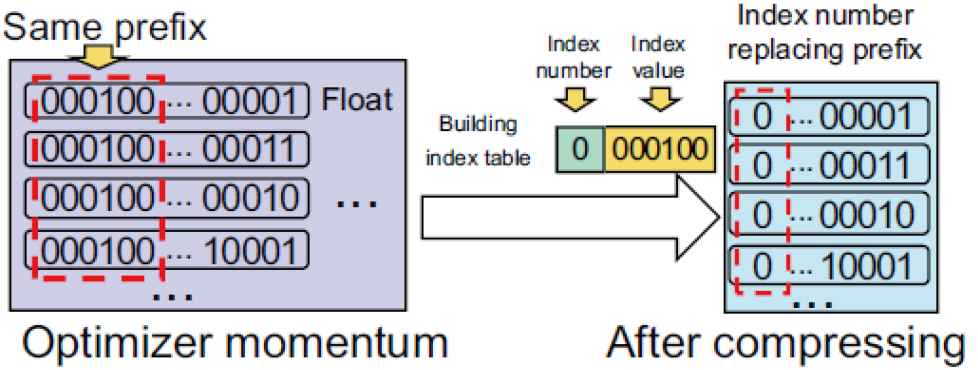

Fig. 3 The process of index compression with a simple example

Method (Cont'd)

3. For maintaining node health status, FastCheck uses a dedicated key-value (KV) store. When users query node health status, the system provides lightweight and consistent health status maintenance to ensure accurate reporting with minimal storage and time overhead.

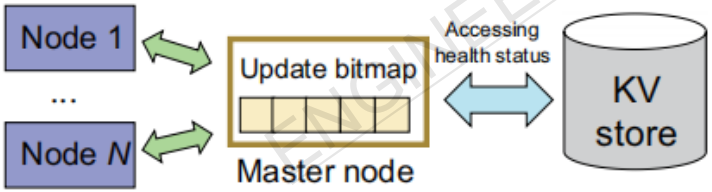

Fig. 4 The overall design of consistency guarantees for health status

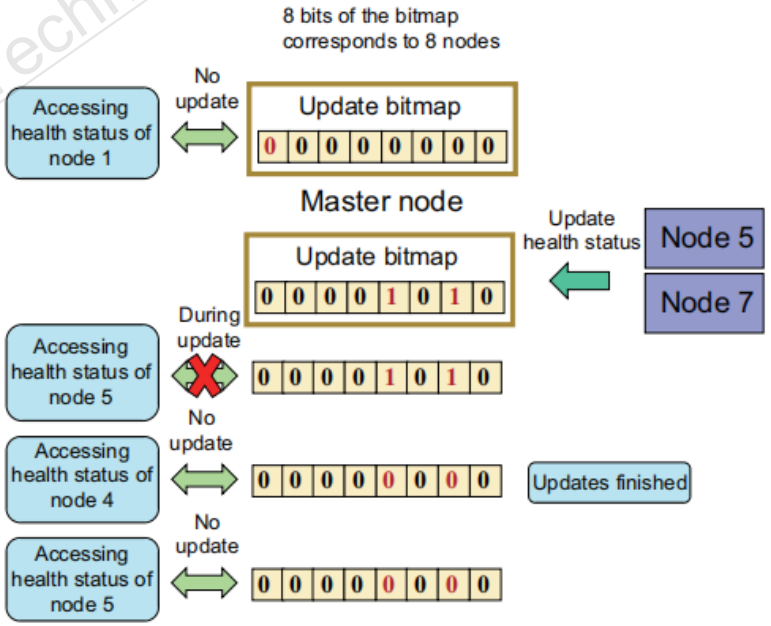

Fig. 5 An example of maintaining health status

Major results

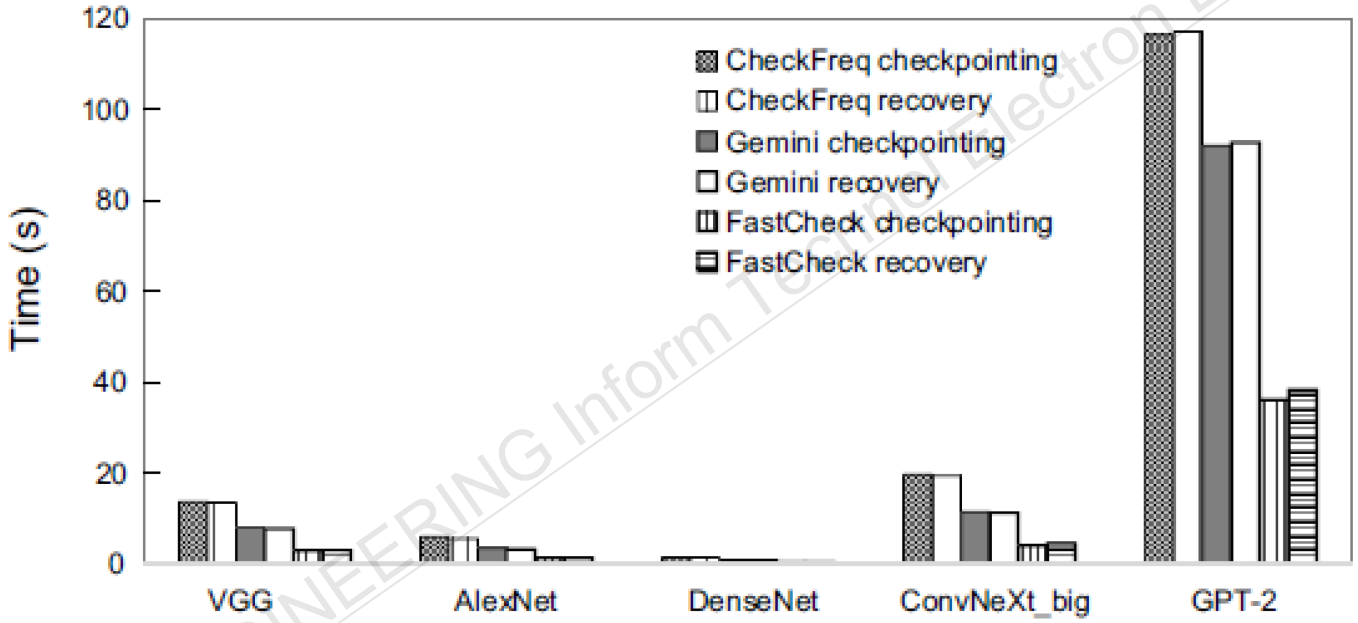

Fig. 6 The checkpointing and recovery performances of the FastCheck and two baselines for the five types of checkpoints

Major results (Cont'd)

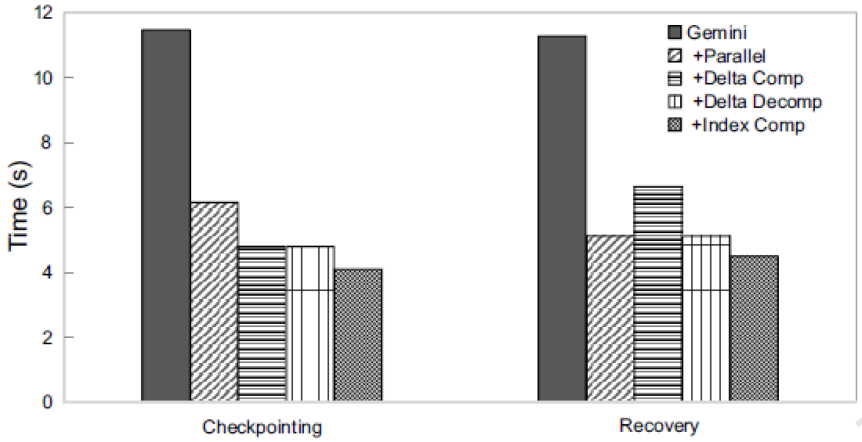

Fig. 7 Performance contributions of individual techniques

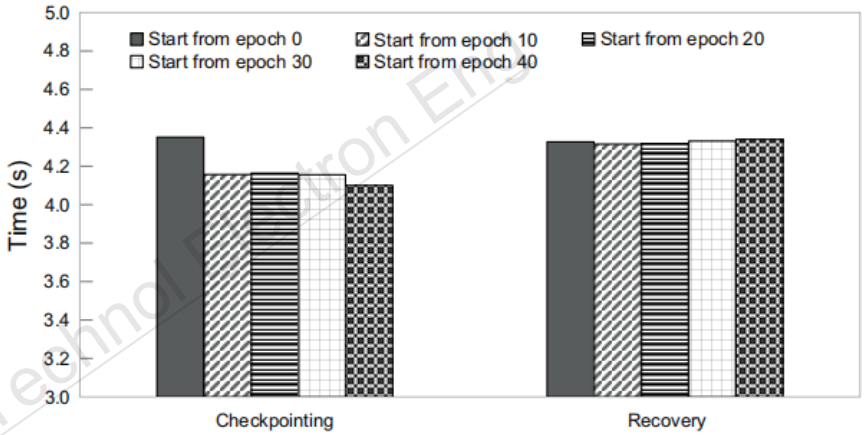

Fig. 8 Performance comparison of different start epochs indices

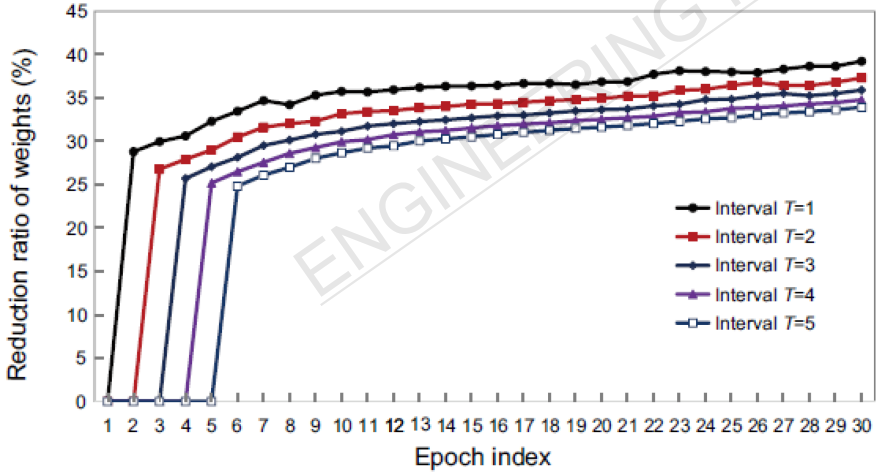

Fig. 9 Reduction ratio of weights by applying different intervals during delta compression

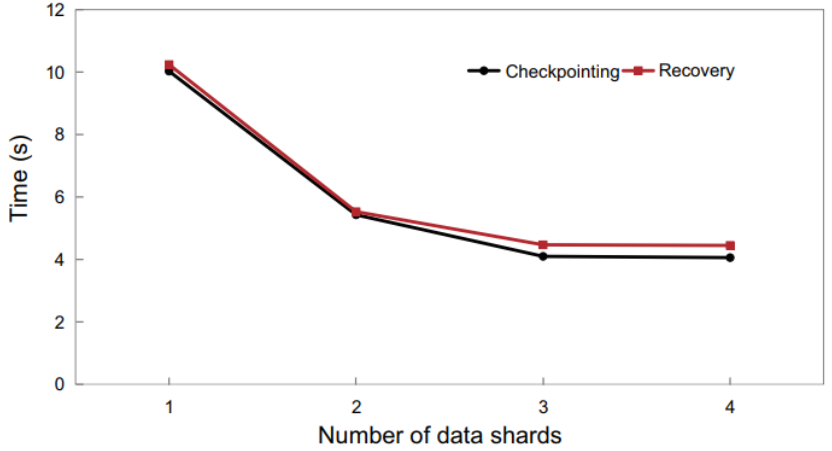

Fig. 10 FastCheck's checkpointing and recovery time for different numbers of data shards

Conclusions

- A fast checkpointing and recovery framework is proposed for DNN training, which applies parallel transmission and tailored compression (delta compression for weights and index compression for momentum).
- Lightweight and consistent health status maintenance is designed to ensure consistent tracking of node health across nodes, with small extra overhead.
- Experiments on multiple DNN models show that FastCheck achieves up to 78.42% reduction in checkpointing time when compared to existing approaches.

Yun TENG received the M.E. degree from Jilin University in 2022. Since Sept. 2022, He is currently working toward the Ph.D. degree in China University of Geosciences Beijing. His research interest is the design of storage system.

Guangyan ZHANG received the B.S., M.S., and Ph.D. degrees in computer science from Jilin University and Tsinghua University. He is an associate professor and Ph.D. advisor at Tsinghua University. His research interests include storage system, big data computing, AI computing, and distributed systems.