Check-n-Run: a Checkpointing System for Training Deep Learning Recommendation Models

Assaf Eisenman, Kiran Kumar Matam, Steven Ingram, Dheevatsa Mudigere, Raghuraman Krishnamoorthi, Krishnakumar Nair, Misha Smelyanskiy, Murali Annavaram
Recommendation Models are Important

• Use cases include:
  • E-commerce marketplaces
  • Social media platforms
  • Entertainment services

• Consumes most of AI compute cycle at Meta
  • > 50% of training compute cycle
  • > 80% of inference compute cycles
Recommendation Model Architecture

- Top MLP
- Feature Interaction
  - Bottom MLP
    - Dense Features
  - Embed Table
    - Sparse Feature
  - Embed Table
    - Sparse Feature
  - Embed Table
    - Sparse Feature
  - Embed Table
    - Sparse Feature
High Performance Training at Meta
The Criticality of Checkpointing

- Failure recovery (ensure progress)
- Migrating training jobs
- Publishing snapshots
- Transfer learning
Checkpoint Challenges

• Accuracy
• Frequency
• Write bandwidth
• Storage capacity
Check-n-Run

• Goal: a checkpointing system that significantly reduces the required write-bandwidth and storage capacity, without degrading accuracy

• What to Checkpoint?

• Decoupled Checkpointing

• Reducing write-bandwidth (WB) and storage capacity
Checkpointing Workflow

- **Training Dataset**
- **Reader**
- **Worker**
- **Master**
- **Checkpoint Storage**
- **Trainer Node**

**Arrows**:
- **→** Training Data
- **→→** Checkpoint Data
Reducing WB with Differential Checkpointing

- Motivation: model accesses are sparse
Approaches for Differential Checkpointing

- One-Shot Differential Checkpoint
- Consecutive Incremental Checkpoint
- Intermittent Differential Checkpoint
Checkpoint Quantization

• Compress checkpoint without degrading training accuracy

• Approaches:

<table>
<thead>
<tr>
<th>Uniform:</th>
<th>Non-Uniform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-0.31) (-0.05, 0.21) 0.03, 0.01</td>
<td>(-0.31) (-0.05) (0.03, 0.21) 0.01</td>
</tr>
</tbody>
</table>
Comparing Quantization Strategies

- Uniform quantization
- Non-uniform quantization using k-means
- Adaptive uniform quantization
Quantization Bit-width Selection

- Quantization error may accumulate
- Select bit-width based on the probability of a failure
Overall Reduction
Summary

• The checkpointing of large recommendation systems at scale is challenging

• Check-n-run:
  • High performance checkpointing
  • Significantly reduces the required write-bandwidth and storage capacity

• Questions? aeisenman@fb.com