Practical Distributed Machine Learning Systems

Luo Mai

University of Edinburgh
Foundation Models

NLP Models:
- GPT-3 (175 billion parameters)
- Switch Transformer (1 trillion parameters)

CV Models:
- Vision Transformers (10s billions parameters)

[1] On the Opportunities and Risks of Foundation Models, 2021
Synchronising Gradients is Network-Intensive

Worker 1
- Model -> Gradients -> Averaged Gradients

Worker 2
- Model -> Gradients -> Averaged Gradients

Worker 3
- Model -> Gradients -> Averaged Gradients

GPT-3 model: 175 billion parameters
175G x 4-byte float = 700 GB

Compute averaged gradients
1. Collect local gradients
2. Broadcast averaged gradients to all workers
Bandwidth is Limited in Data Centres

Bandwidth over-subscription ratio = 1:2

A real-world data centre

Over-subscription ratio = 1:2
Bandwidth-efficient Allreduce

Initial State

Worker 1: 2 4 6
Worker 2: 1 2 3
Worker 3: 4 8 12

Final State

Worker 1: 7 14 21
Worker 2: 7 14 21
Worker 3: 7 14 21

Allreduce

7 = 1 + 2 + 4 (first partition)
14 = 2 + 4 + 8 (second partition)
21 = 3 + 6 + 12 (third partition)
Step 1 (Reduce)

Worker 1: 3 4 6
Worker 2: 1 10 3
Worker 3: 1 8 18

Step 2 (Reduce)

Worker 1: 3 14 6
Worker 2: 1 10 21
Worker 3: 7 8 18

Step 3 (Broadcast)

Worker 1: 3 14 21
Worker 2: 7 14 21
Worker 3: 7 14 21

Step 4 (Broadcast)

Worker 1: 7 14 21
Worker 2: 7 14 21
Worker 3: 7 14 21

Discussion
- Stragglers
- Failures
Questions?
Common Collective communication operators

- Collective communication libraries: NVIDIA Collective Communications Library (NCCL), OpenMPI
- Integration with ML frameworks: PyTorch Distributed, Horovod, KungFu
ML Server Architecture

- Memory hierarchy
  - GPU memory bandwidth: ~2000 GB/s
  - System memory bandwidth: ~1600 GB/s
  - SSD: ~20 GB/s

- High-bandwidth networks
  - GPU-GPU direct: NVLink: 600 GB/s
  - CPU-GPU PCIe Switch: 64GB/s
  - Server-Server InfiniBand: 25 GB/s

- ML data placement
  - How to improve data locality?
  - How to reduce data access latency?

Future Systems for Foundation AI Models

• Automatic model parallelism
  • Parallelism cost model + parallelism strategy solver [1]

• Memory-efficient runtime
  • Memory swapping (CPU memory + GPU memory) [2]

• Efficient optimisers for giant AI models
  • Lookahead Optimizer [3]

[3] Lookahead Optimizer: k steps forward, 1 step back, 2020
Questions?
Recommender Systems

Digital Content
2.7 Billion Monthly Active Users

E-Commerce
2 Billion Digital Shoppers

Social Media
3.8 Billion Active Users

Digital Advertising
4.65 Billion User Targeted
Deep Learning Recommender Models

Recommender system

- History of Interactions / Candidate Items
- 1000M x 100 User Embedding
- 10M x 100 Item Embedding
- User Context Embedding
- Item Context Embedding

- Recommendations / Rank

- 100s GB – 10s TB
- 10s GB – 100s GB

- Tremendous memory cost
- Small computation cost
- Embeddings are sparsely updated

- Example: Deep learning recommendation models [1]

Parameter Servers

CPU-Only Machine

Push embedding table A (Optional)

Dataset

Neural Network

GPU-CPU Machine

Parameter Server 1

Parameter Server 2

Parameter Server 3

Parameter Server 4

Parameter Server 5

Pull embedding table A

A

Batch A

A
Users/Items Follow Power Law Distribution

Imbalanced workload is common in distributed computing systems
Handling Hot Spots via Data Replication

Parameter Server Replication Group

- Replica
- Replica
- Replica

Dynamically choose replica

Load balancer

Pull embedding table

Worker 1

Worker 2

Pull embedding table
Future Recommender Systems

• Protecting user privacy
  • Federated Learning (FL)
    • Google Input Method with FL [1]
  • Trusted Execution Environments (TEEs)
    • x86 TEEs (AMD & Intel)
    • GPU TEE (Nvidia), coming soon

• Adaptive recommender engines
  • Updating models in real-time: Ekko [2]

Summary

• Foundation AI models
• Collective communication systems
  • Bandwidth over-subscription, stragglers, failures, high-speed networks
• Deep learning recommendation systems
• Parameter servers
  • Data skews, replications
Reading

- Optional reading
  - *Dive into deep learning – computational performance*
Questions?

Largs-Scale Computer Systems Group
https://luomai.github.io