DreamShard: Generalizable Embedding Table Placement for Recommender Systems

Daochen Zha, Louis Feng, Qiaoyu Tan, Zirui Liu, Kwei-Herng Lai, Bhargav Bhushanam, Yuandong Tian, Arun Kejariwal, Xia Hu

Rice University
Meta Platforms, Inc.
Texas A&M University
Distributed Recommender System

Combining data-parallelism and model-parallelism.
Embedding Table Placement Problem

• Problem Setting
  • We consider embedding table placement on GPU devices.
  • Embedding accounts for 48% and 65% of the computation and communication costs in production model.
Embedding Table Placement Problem

(a) Random placement

(b) The existing best human expert strategy

(c) DreamShard
Key Challenges

• Challenges

• Operation fusion, which uses a single operation to subsume multiple tables, makes it hard estimate cost.
• The adopted embedding tables and the available devices can change frequently (e.g., machine learning engineers may conduct experiments with various table combinations and numbers of devices).
Formulation of MDP

- Markov Decision Process

Step t=0: Unplaced Table
Action $a_1$: Device 1

Step t=1: Unplaced Table
Action $a_2$: Device 2

Step t=2: Table to be Placed
Action $a_3$: Device 2

Step t=3: Placed to Device 1
Action $a_4$: Device 1

... Final Placement

Step t=8: Placed to Device 2
Action $a_8$: Device 1
DreamShard Framework
## Main Results

### Observations
- DreamShard outperforms baselines significantly.
- DreamShard can generalize well (test performance is similar to train performance).

| Task          | No strategy | Human Experts | RL                          |
|---------------|-------------|---------------|-----------------------------|
|               | Train       | Test          | Random | Size-based | Dim-based | Lookup-based | Size-lookup-based | RNN-based | DreamShard |
| DLRM-20 (4)   | Train 24.0±0.6 | Test 23.0±0.5 | 22.7±0.0 (+5.7%) | 21.3±0.0 (+12.7%) | 19.1±0.0 (+25.7%) | 19.1±0.0 (+25.7%) | 22.4±0.5 (+7.1%) | 18.6±0.2 (+29.0%) | 17.6±0.2 (+30.7%) |
|               | Train 41.3±0.2 | Test 41.1±0.5 | 39.6±0.0 (+4.3%) | 37.4±0.1 (+10.4%) | 35.6±0.0 (+22.9%) | 35.6±0.0 (+22.9%) | 39.2±0.7 (+5.4%) | 32.8±0.3 (+25.9%) | 32.8±0.3 (+26.9%) |
|               | Train 57.7±0.8 | Test 58.1±0.6 | 56.6±0.1 (+1.9%) | 59.6±0.1 (+2.5%) | 53.7±0.0 (+8.2%) | 49.2±0.1 (+17.3%) | 49.3±0.2 (+17.0%) | 55.1±0.9 (+4.0%) | 47.6±0.4 (+21.2%) | 47.7±0.3 (+21.9%) |
|               | Train 75.7±1.0 | Test 74.5±0.8 | 76.0±0.0 (+0.4%) | 77.7±0.2 (+1.1%) | 69.9±0.4 (+6.6%) | 64.8±0.0 (+16.8%) | 64.1±0.2 (+16.2%) | 65.3±0.1 (+15.9%) | 73.2±1.7 (+3.4%) | 62.2±0.2 (+21.7%) | 62.7±0.3 (+18.8%) |
| DLRM-100 (4)  | Train 19.8±1.7 | Test 94.5±6.5 | 94.1±0.3 (+2.4%) | 94.7±0.0 (+0.9%) | 86.7±0.3 (+5.9%) | 84.7±0.4 (+11.6%) | 81.2±0.4 (+13.1%) | 82.2±0.2 (+11.7%) | 94.5±1.0 (+7.2%) | 94.8±1.0 (-0.3%) | 78.4±0.6 (+17.1%) | 77.8±0.8 (+21.5%) |
| DLRM-40 (8)   | Train 15.6±0.4 | Test 15.2±0.2 | 14.1±0.0 (+10.6%) | 13.4±0.1 (+16.4%) | 9.8±0.0 (+59.2%) | 9.9±0.0 (+57.6%) | 16.2±0.8 (-3.7%) | 9.5±0.0 (+60.0%) | 9.9±0.0 (+59.2%) | 9.4±0.5 (+61.7%) |
| DLRM-80 (8)   | Train 25.0±0.2 | Test 25.2±1.3 | 24.0±0.0 (+4.2%) | 25.6±0.5 (-1.6%) | 21.7±0.0 (+15.2%) | 17.1±0.0 (+46.2%) | 16.7±0.2 (-50.9%) | 17.5±0.0 (+42.9%) | 16.9±0.1 (-49.1%) | 51.4±3.9 (-51.4%) | 53.4±4.6 (-52.8%) | 16.1±0.3 (+55.3%) | 16.1±0.4 (+56.5%) |
| DLRM-120 (8)  | Train 34.0±0.3 | Test 33.5±0.5 | 32.3±0.0 (+5.3%) | 35.0±0.0 (-4.3%) | 29.8±0.0 (+14.1%) | 24.5±0.0 (+38.8%) | 23.7±0.0 (+41.1%) | 25.3±0.0 (+34.4%) | 24.5±0.0 (+36.7%) | 58.6±1.7 (-42.0%) | 57.8±1.3 (-39.2%) | 23.3±0.2 (+45.9%) | 22.8±0.2 (+46.9%) |
| DLRM-160 (8)  | Train 42.8±0.3 | Test 41.1±0.0 | 41.6±0.0 (+2.9%) | 42.4±0.0 (-3.1%) | 39.0±0.0 (+9.7%) | 36.4±0.0 (+12.9%) | 32.0±0.0 (+33.7%) | 32.7±0.0 (+30.9%) | 31.6±0.0 (-30.1%) | 58.3±3.5 (-26.6%) | 59.5±3.4 (-50.7%) | 30.3±0.2 (+41.3%) | 29.6±0.2 (+38.9%) |
| DLRM-200 (8)  | Train 51.5±1.2 | Test 50.7±0.2 | 48.2±0.0 (+6.8%) | 50.0±0.0 (+0.2%) | 48.0±0.0 (+7.3%) | 44.8±0.0 (+13.2%) | 38.9±0.0 (+32.4%) | 38.6±0.0 (+31.3%) | 38.6±0.0 (+30.5%) | 68.7±2.4 (-25.0%) | 70.4±2.8 (-28.0%) | 37.2±0.2 (+38.4%) | 36.4±0.3 (+39.3%) |
| Prod-20 (2)   | Train 41.3±0.7 | Test 42.8±0.4 | 43.4±0.0 (+4.8%) | 46.1±0.0 (-7.2%) | 37.0±0.0 (+11.6%) | 44.2±0.0 (+6.6%) | 45.9±0.0 (-6.8%) | 45.8±0.0 (+9.8%) | 38.0±0.3 (+8.7%) | 39.3±0.6 (+8.9%) | 36.3±0.3 (+13.8%) | 37.5±0.2 (+14.1%) |
| Prod-40 (4)   | Train 35.1±0.3 | Test 38.3±0.3 | 39.4±0.0 (-10.9%) | 43.6±0.0 (-12.2%) | 31.3±0.0 (+12.1%) | 35.0±0.0 (+3.6%) | 37.4±0.0 (+2.4%) | 38.8±0.0 (+9.5%) | 40.1±0.0 (-4.5%) | 33.9±2.5 (+35.4%) | 36.7±2.3 (-24.4%) | 28.3±0.3 (+24.0%) | 30.4±0.7 (+26.0%) |
| Prod-80 (8)   | Train 43.2±0.2 | Test 47.7±0.4 | 44.3±0.0 (-2.5%) | 53.9±0.0 (-11.5%) | 39.0±0.0 (+10.8%) | 43.7±0.0 (+1.1%) | 46.1±0.0 (+3.5%) | 49.3±0.0 (-12.4%) | 49.6±0.0 (-3.8%) | 56.6±0.8 (-23.7%) | 55.2±0.8 (-35.5%) | 33.6±0.9 (+28.6%) | 35.2±0.8 (+35.5%) |
Takeaways

• **Summary**
  - We explore embedding table placement/sharding, a direction that has been rarely explored.
  - We propose DreamShard, which learns estimated MDP and an RL agent.
  - DreamShard significantly outperforms heuristic baselines.

Paper

Code