CS 744: MARIUS

Shivaram Venkataraman
Fall 2022
- Midterm grades out!
- Regrade requests: In-person (strongly preferred)
  - Thu: After class, Roger’s OH
  - Mon: Shivaram’s OH, Roger’s OH
  - Tue: After class
- Course Project: Check in by Nov 23\textsuperscript{th}
PROJECT CHECK-INS

One page document that includes the following

- What have you done so far
- Any challenges that you have faced so far
- Your timeline (from now till end of the semester)
- Things you need help from the course staff
- Any other comments/remarks
Scalable Storage Systems

Datacenter Architecture

Resource Management

Computational Engines

Machine Learning

SQL

Streaming

Graph

Applications

SQL

Streaming

Graph

Machine Learning
EXAMPLE: LINK PREDICTION

Task: Predict potential connections in a social network

Find K-nearest neighbors
Score function
Capture structure of the graph given source, destination embedding

Loss function
Maximize score for edges in graph
Minimize for others (negative edges)

$$\mathcal{L} = \sum_{e \in G} \sum_{e' \in S'_e} \max(f(e) - f(e') + \lambda, 0))$$
TRAINING ALGORITHM

SGD/AdaGrad optimizer

Sample positive, negative edges

Access source, dest embeddings for each edge in batch

for $i$ in range(num\_batches)
    $B = \text{getBatchEdges}(i)$
    $E = \text{getEmbeddingParams}(B)$
    $G = \text{computeGrad}(E, B)$
    $\text{updateEmbeddingParams}(G)$
CHALLENGE: LARGE GRAPHS

Large graphs $\rightarrow$ Large model sizes

Example

3 Billion vertices, $d = 400$
Model size = 3 billion $\times$ 400 $\times$ 4 = 4.8 TB!

Need to scale beyond GPU memory, CPU memory!
DGL-KE: Sample edges, embeddings from CPU memory

Pytorch-BigGraph: Partition embeddings so that one partition fits on GPU memory. Load sequentially

One epoch on the Freebase86m knowledge graph
MARIUS

I/O efficient system for learning graph embeddings

Marius Design
- Pipelined training
- Partition ordering
PIPELINED TRAINING

CPU Memory

Edges

Node Embedding Parameters

Load

Update

Transfer

GPU Memory

Transfer

Compute

Relation Embedding Parameters
OUT OF MEMORY TRAINING

Key idea: Maintain a cache of partitions in CPU memory

Questions
Order of partition traversal?
How to perform eviction?

Partitions in Buffer
\[ c = 3 \]

Partitions on disk
\[ p = 6 \]
**BETA ORDERING**

Initialize cache with $c$ partitions

Swap in partition that leads to highest number of unseen pairs

Achieved by fixing $c-1$ partitions and swap remaining in any order
Graph Embeddings: Learn embeddings from graph data for ML

Marius: Efficient single-machine training
  Pipelining to use CPU, GPU
  Partition buffer, BETA ordering
DISCUSSION

https://forms.gle/uNAKsPsZp56Cc1Vz9
How does the partitioning scheme used in this paper differ from partitioning schemes used in PowerGraph and why?
<table>
<thead>
<tr>
<th>System</th>
<th>Deployment</th>
<th>Epoch Time (s)</th>
<th>Per Epoch Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marius</td>
<td>1-GPU</td>
<td>727</td>
<td>0.61</td>
</tr>
<tr>
<td>DGL-KE</td>
<td>2-GPUs</td>
<td>1068</td>
<td>1.81</td>
</tr>
<tr>
<td>DGL-KE</td>
<td>4-GPUs</td>
<td>542</td>
<td>1.84</td>
</tr>
<tr>
<td>DGL-KE</td>
<td>8-GPUs</td>
<td>277</td>
<td>1.88</td>
</tr>
<tr>
<td>DGL-KE</td>
<td>Distributed</td>
<td>1622</td>
<td>2.22</td>
</tr>
<tr>
<td>PBG</td>
<td>1-GPU</td>
<td>3060</td>
<td>2.6</td>
</tr>
<tr>
<td>PBG</td>
<td>2-GPUs</td>
<td>1400</td>
<td>2.38</td>
</tr>
<tr>
<td>PBG</td>
<td>4-GPUs</td>
<td>515</td>
<td>1.75</td>
</tr>
<tr>
<td>PBG</td>
<td>8-GPUs</td>
<td>419</td>
<td>2.84</td>
</tr>
<tr>
<td>PBG</td>
<td>Distributed</td>
<td>1474</td>
<td>2.02</td>
</tr>
</tbody>
</table>
What are some shortcomings of Marius? What could the authors do to further improve the system?
NEXT STEPS

Next class: Distributed GNNs
Project check-ins by Nov 23th