Background & Motivation

- Model size grows “beyond” the capability of a single machine.
- “Parameter Server” is distributed in-memory solution.
- Distributed in-memory is expensive and not always necessary (especially for sparse data).
- We introduce a novel solution to support “big models” within a single database server by utilizing secondary storage.

Data & Model Representation

- Data is represented as a set of arrays: 
  \[ \text{Data}(\text{index INTEGER[]}, \text{value DOUBLE[]} , \text{label INTEGER}) \]
- Model is represented as a database relation: 
  \[ \text{Model}(\text{index INTEGER}, \text{value DOUBLE}) \]

Sparsity-Aware Dot-Product Operator

- The ubiquitous dot-product: \( \mathbf{x}_i \cdot \mathbf{w} \)
  - \( \mathbf{x}_i \) is the training example; \( \mathbf{w} \) is the model
  - Example: gradient in logistic regression 
    \[ \sum_i e^{-y_i \mathbf{x}_i^T \mathbf{w}} (y_i \mathbf{x}_i) \]
- Requirements for parallel dot-product operator:
  - Support sparse data: keep only working sets in memory
  - Compute dot-product in a non-blocking fashion to support stochastic gradient descent (SGD)

Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Dim</th>
<th># Example</th>
<th>Data Size</th>
<th>Model Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>splice</td>
<td>13 M</td>
<td>50 M</td>
<td>3 TB</td>
<td>100 M</td>
</tr>
<tr>
<td>zipf</td>
<td>125 M</td>
<td>100 K</td>
<td>3.7 GB</td>
<td>1 GB</td>
</tr>
</tbody>
</table>

Design Of Dot-Product Operator

- Parallel Dot-Product Operator
  - Data reordering with locality sensitive hashing

Evaluation of Dot-Product Operator

- 16 cores @ 2GHz; 20 GB RAM; 100 MB/s I/O throughput

- Comparison between dot-product solutions
  - 10x faster than blocking join in PostgresSQL, 40x faster than non-blocking indexed join in PostgresSQL

Evaluation of stochastic gradient descent (SGD)

- Iteration execution time is not drastically affected when decreasing memory budget.
- Memory-bound execution when budget is low, but still able to scale with multi-thread parallelism.

University of California, Merced
Email: Chengjie Qin cqin3@ucmerced.edu, Florin Rusu frusu@ucmerced.edu