DIMMWITTED: A STUDY OF MAIN-MEMORY
STATISTICAL ANALYTICS

Shivaram Venkataraman
How to best use main memory?

Memory Bandwidth:
~60 GB/s r3.8xlarge on EC2
DESIGN SPACE

• Access method
  – Row vs. Column
  – Density
• Replication
  – Data
  – Model
ITERATIVE ALGORITHMS: ACCESS METHOD

Sample rows vs. columns
Broadly “gradient” vs “coordinate” methods.
DATA DENSITY: DENSE VS. SPARSE

Dense Linear Algebra
- More FLOPs / CPU intensive
- e.g., Matrix vector multiply: $O(n \times d)$

Sparse Linear Algebra
- Lesser FLOPs / communication intensive
- e.g., Matrix vector multiply: $O(\text{nnz} \times d)$
DIMM WITTED: ACCESS METHODS

<table>
<thead>
<tr>
<th>Data</th>
<th>Row-wise</th>
<th>Col.-wise</th>
<th>Col.-to-row</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Diagram]</td>
<td>[Diagram]</td>
<td>[Diagram]</td>
</tr>
</tbody>
</table>

- **Read set of data**
- **Write set of model**
REPLICATION

Model
- Replica per core? Similar to Spark, shared nothing
- Replica per machine? Shared memory
- Hybrid: Replica per NUMA node

Data
- Partition per core? Similar to shared nothing
- Replicate data per NUMA node?
OPTIMIZER

Inputs

- \( f_{\text{row}}, f_{\text{col}}, f_{\text{ctr}} \)
- data \( A \in \mathbb{R}^{N \times d} \)
- Initial model vector

Output

- Execution plan for each CPU
  - subset of data
  - model replica
  - access method to use
## ACCESS METHOD

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Read</th>
<th>Write (Dense)</th>
<th>Write (Sparse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row-wise</td>
<td>$\sum n_i$</td>
<td>$dN$</td>
<td>$\sum n_i$</td>
</tr>
<tr>
<td>Column-wise</td>
<td>$\sum n_i$</td>
<td></td>
<td>$d$</td>
</tr>
<tr>
<td>Column-to-row</td>
<td>$\sum n_i^2$</td>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>

- Cost Ratio: how much more expensive writes are
- Row-wise is more efficient when writes are cheap
- Column-to-row becomes more efficient at some point

![Time for Each Epoch](chart.png)
MODEL REPLICATION

(a) Number of Epochs to Converge

(b) Time for Each Epoch

Model Replication Strategies
TAKEAWAYS

- Data access patterns matters but changes based on problem

- Model / data replication design space

- “Optimizer” for ML
QUESTIONS / DISCUSSION ?