Machine Learning Problems

- Many models have \(O(1)\) blocks of \(O(n)\) terms (LDA, logistic regression, recommender systems)
- More features than what fits into RAM (10^{11} dimensions) (personalized CTR, large inventory, action space)
- Unreliable infrastructure (preemption, failure, slowdown)
- Local model typically fits into RAM
- Data needs many disks for distribution (100TB and more)
- Decouple data processing from aggregation
- Sweet spot - optimize for 80% of ML

Parameter Server

- Clients process data shards
- Clients have local view of global state and purely local state
- Parameter server has full global state
- Updates are via push / pull

### Sparse Logistic Regression

\[
\text{minimize } \sum_{i=1}^{m} \log(1 + \exp(-y_i (w \cdot x_i))) + \lambda \|w\|_1
\]

- Compute gradient on (subset of data) on each client
- Send gradient from client to server asynchronously
- Proximal gradient update on server per coordinate
- Server returns parameters

### Architecture Overview

- Chord style key layout (keys and machines in ring)
- Replication along chain (a la Ouroboros)
- Recovery from failure by hot failover
- Multiple virtual servers per server for load balancing and efficient recovery
- Dynamic scaling for free!
- Consistency via vector clocks for ranges
- Key / value compression
- Dynamic task dependency graph and scheduler

Communication

- Convergence speed depends on communication efficiency
- Sending (key,value) pairs is inefficient
- Send only values (cache key list & checksum) instead
- Send only (key,value) pairs that client needs
- Sending small gradients is inefficient
- Send only sufficiently large ones instead
- Randomize and compress accuracy of values
- Updating near-optimal values is inefficient
- Send only large violators of KKT conditions
- Filters to allow clients / servers to self-censor
- Avoid need for fancy scheduler

### Consistency and Vector Clocks

- Different consistency requirements for different models
- Use dependency graph to accommodate all of them (as special cases)
- Task controller sends subtasks to workers
- Flexible adjustment at runtime as needed
- Vector clocks
- Per (key,value) pair costs \(O(\text{clients})\) storage (fatal)
- Apply for each range and partition only as needed

Performance

- Logistic regression (100TB CTR data)

<table>
<thead>
<tr>
<th>System</th>
<th>Method</th>
<th>Consistency</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>L-BFGS</td>
<td>Sequential</td>
<td>10,000</td>
</tr>
<tr>
<td>B</td>
<td>Block PG</td>
<td>Sequential</td>
<td>30,000</td>
</tr>
<tr>
<td>Parameter Server</td>
<td>Block PG</td>
<td>Bounded Delay</td>
<td>300</td>
</tr>
</tbody>
</table>

- Sketches (15 machines on 40 Gbit/s net)

  Peak inserts per second: 1.3 billion
  Average inserts per second: 1.1 billion
  Peak network bandwidth per machine: 4.37 GB/s
  Time to recover a failed node: 0.8 second

- Topic models (4 Billion documents, 60k cores, 1M tokens)