Incoop: MapReduce for Incremental Computations


Reviewed by Neil Satra
Why?

You are calculating PageRank at Google.
Crawling petabytes of web pages.
1% of web pages have changed every time you crawl.
Why?

Iterative
Hard to scale efficiently

Batch
Need to redo entire computation for updated data
Why?

<table>
<thead>
<tr>
<th>Iterative</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard to scale efficiently</td>
<td>Need to redo entire computation for updated data</td>
</tr>
</tbody>
</table>

Incremental Batch Data Processing
How?

Caching:

Option A: Give programmers the primitives
Option B: Do it transparently
<table>
<thead>
<tr>
<th></th>
<th>Not transparent</th>
<th>Transparent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dryad and other tools</td>
<td>Yahoo! CBP</td>
<td>DryadIncl, Nectar</td>
</tr>
<tr>
<td>MapReduce</td>
<td>Google Percolator</td>
<td>Incoop</td>
</tr>
</tbody>
</table>
How?

3 optimizations:

• Partitioning of file system
• Fine-grained Reduce phase
• Memoization-aware scheduling
How?

Figure 1: Basic design of Incoop

Source: the paper
Strengths

- Results: 10x to 1000x speedup, with a negligible processing overhead

- Evaluation: Used unmodified code for 5 realistic applications and showed improvements both quantitatively and with mathematical proofs

- Optimizations show attention paid beyond surface-level
Weaknesses

- Evaluation: No quantitative comparison with non-transparent systems (Google Percolator)
- Insufficient discussion of the memoization server, which could be a bottleneck or central point of failure. No attempt to decentralize that component.
- Storage is linear in terms of input
- Assumptions about the application
- Garbage Collection of old cache entries
- Evaluation: Replaced part of data with equal sized chunks, rather than appending new data
Summary

- Modified version of Hadoop (MapReduce)
- Efficient processing of large scale data, with incremental updates
- Works with existing code, transparently
- Memoizes computations, and tunes the operation of MapReduce to take maximum advantage of memoization
- Strong contributions, decently evaluated, number of potential concerns have been addressed

By Neil Satra


