Task Fusion: Improving Utilization of Multi-user Clusters

Robert Dyer
rdyer@iastate.edu

Iowa State University

The research and educational activities described in this talk was supported in part by the US National Science Foundation (NSF) under grants CCF-13-49153, CCF-11-17937, and CCF-08-46059.
Submit Task

Computing Cluster

MapReduce, Hadoop, etc

FIFO Queue

T1 T2 T3 T4 T5

Task Result
Computing Cluster
Submit Task
MapReduce, Hadoop, etc
Time Sharing
T1 T2 T3 T4 T5
Task Result
Solutions?

- **Scale the hardware**
  - Expensive
  - Not always feasible (small businesses, MOOCs, researchers, etc)

- **Optimize the software**
  - Optimize individual tasks
    - standard program optimizations
    - chain folding [MinerShook12], sibling/MSCR fusion [Chambers10]
  - Optimize multiple tasks
    - manual job merging [MinerShook12]

---

[Chambers10] Craig Chambers et al., “FlumeJava”, PLDI 2010
[MinerShook12] Donald Miner and Adam Shook, “MapReduce Design Patterns”, O’Reilly, 2012
Key Insights

1) People analyze similar data
   - NCDC, NCCS
   - 1k Genomes Project
   - SDSS
   - US Census

2) Data-intensive computing
   - Loading GB/TB/PB of data takes time

Insight: **Load data once, run multiple analyses**
Research Questions

1. Can we *automatically* merge related tasks from *different users*?

   Answer: Task Fusion

2. Does *Task Fusion* decrease user wait times in shared computing clusters?
Submit Tasks → Task Fusion → Individual Task → MapReduce Cluster → Task Result 1 → Task Result N
Technical Challenge: \textit{map output == side effect}
Solution: modify maps to output composite keys

Custom partitioner ensures proper routing
Research Prototype

Task Fusion implemented for Boa

- Large-scale software repository mining
- SourceForge data (700k projects)
- Automatically parallelizes queries
## Early Results

<table>
<thead>
<tr>
<th>Task Size</th>
<th># of Tasks</th>
<th>Times No Task Fusion</th>
<th>Times Task Fusion</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small(^1)</td>
<td>21</td>
<td>8.1m</td>
<td>0.8m</td>
<td>10.8X</td>
</tr>
<tr>
<td>Medium(^2)</td>
<td>22</td>
<td>2.3h</td>
<td>1.8h</td>
<td>1.3X</td>
</tr>
<tr>
<td>Large(^2)</td>
<td>18</td>
<td>4.6h</td>
<td>3.9h</td>
<td>1.2X</td>
</tr>
<tr>
<td>Mixed(^3)</td>
<td>9</td>
<td>1.3h</td>
<td>0.9h</td>
<td>1.4X</td>
</tr>
</tbody>
</table>

[1] queries on project and revision metadata only
[2] queries on metadata and millions of source files
[3] 3 small, 3 medium, 3 large
Early Results

User Wait Time Improvement

Time (seconds)

Number of Tasks Fused
1. No shared state
2. No dependency conflicts
   
   **Idea:** Separate class spaces (a la OSGi)

3. Controllable side effects
   
   **Idea:** Automated program transformations

Assumption: Relax Assumptions