Low memory Map-Reduce

Hrishikesh Amur, Karsten Schwan, Georgia Tech.
Wolf Richter, Athula Balachandran, Erik Zawadzki,
Dave Andersen, CMU
Michael Kaminsky, Intel
Datasets are growing

People see value (even a little) in storing data rather than throwing it away
Map-Reduce

Map

Map

Map

Red.

Red.

Red.
Map-Reduce
Map-Reduce

Map

Map

Red.

Red.
Map-Reduce

Map

Red.

Map

Red.

Map

Red.

Map

Red.
Data transmitted over network can be reduced!
Aggregation is critical...
Aggregation is critical...

• Useful data is small (selection problems)
Aggregation is critical...

- Useful data is small (selection problems)
- Aggregate smaller than sum of parts (aggregation problems)
Aggregation is critical...

- Useful data is small (selection problems)
- Aggregate smaller than sum of parts (aggregation problems)
- Networks usually oversubscribed
... as others have said

- Parallel databases allow aggregation, but queries become complex

- Dryad, MapReduce and Hadoop.
Pre-aggregation in Hadoop
Pre-aggregation in Hadoop
Pre-aggregation in Hadoop

Can aggregation be performed in memory-constrained environments?
Why memory-constrained?
Why memory-constrained?

- Energy
Why memory-constrained?

• Energy
• Decreasing memory per core
Why memory-constrained?

- Energy
- Decreasing memory per core
- Fun :)}
Pre-aggregation in Hadoop

Map  Sort  Add
Pre-aggregation in Hadoop
Pre-aggregation in Hadoop
Pre-aggregation in Hadoop
Pre-aggregation in Hadoop

In-memory sort limits aggregation
Minni: Low-memory Map-Reduce
Minni: Low-memory Map-Reduce

- Memory-efficient
Minni: Low-memory Map-Reduce

- Memory-efficient
- Performance scales with available memory
Minni: Low-memory Map-Reduce

- Memory-efficient
- Performance scales with available memory
- External aggregation using SSDs
Partial Aggregation Object (PAO)

Key, Value

<table>
<thead>
<tr>
<th>User-defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>create(key, value)</td>
</tr>
<tr>
<td>destroy()</td>
</tr>
<tr>
<td>merge(PAO)</td>
</tr>
<tr>
<td>serialize()</td>
</tr>
<tr>
<td>deserialize()</td>
</tr>
</tbody>
</table>

Distributed Aggregation for Data-Parallel Computing: Interfaces and Implementations, Yu et. al., SOSP'09
Grouping by Hashing

Map  Sort  Add
Grouping by Hashing
Grouping by Hashing

Map — Hash

Aggregate as you hash
Grouping by Hashing

But the hash table might not fit in memory

Aggregate as you hash
External Aggregators

- Bucketing
- External Sort
- External Hash
Bucketing

Map → Hash → Part. → Files on SSD

Hash
Bucketing

Map → Hash → Part.

Cap: 10 keys

Files on SSD

Hash
Bucketing

100 keys

Map → Hash → Part.

Cap: 10 keys

Files on SSD

Wednesday, October 19, 2011
Bucketing

Map → Hash → Part.

Files on SSD

100 keys

Cap: 10 keys

12 buckets

Wednesday, October 19, 2011
Bucketing

100 keys

Map → Hash → Part.

Hash

Files on SSD

Has <10 keys

12 buckets
Bucketing

100 keys

Map → Hash → Part.

Files on SSD

Has <10 keys

Hash

Can aggregate in memory!

12 buckets

Wednesday, October 19, 2011
Bucketing

Technique:
SSDs can support writes to many files
Bucketing

Technique:
SSDs can support writes to many files

But, how many?
Bucketing

Technique:
SSDs can support writes to many files

But, how many?

100 keys
Map
Hash
Part.
Cap: 10 keys

Files on SSD

12 buckets

Wednesday, October 19, 2011
Bucketing

Technique:
SSDs can support writes to many files
External Sort

Map → Hash → Overflow File

Overflow File → Ext. Sort → Add
External Sort

• Technique: Trade-off memory consumption for extra CPU work
External Hash

18
External Hash

Use random read capabilities of SSDs
Pipelining

- Aggregators implemented as pipelines in Intel Threading Building Blocks (TBB)
Effects of token size (bucketing)

Map time (s)

Average memory (MB)

Token size (MB)

Wordcount:
8G dataset
7 B/key
1 mil keys
Comparisons

Map time (s) vs Memory (MB)

- Bucketing
- External Sort
- Hadoop

Wordcount:
- 8G dataset
- 7 B/key
- 1 mil keys
Recap of Techniques
Recap of Techniques

• Use SSD capabilities
  • Parallel writes to multiple files
  • High random read capabilities
Recap of Techniques

• Use SSD capabilities
  • Parallel writes to multiple files
  • High random read capabilities
• Trade-off latency for low memory consumption
Recap of Techniques

• Use SSD capabilities
  • Parallel writes to multiple files
  • High random read capabilities
• Trade-off latency for low memory consumption
• Trade-off CPU work for low memory consumption
Questions & Suggestions