Hadoop Map-Reduce – Tuning and Debugging

Arun C Murthy
acmurthy@apache.org
Yahoo! CCDI
Topical Matters…

• Who doesn’t know Map-Reduce?!  
• Peek inside your MR application…  
• Tuning  
• Debug (god forbid!)
Counters ...

- Often MR applications have countable ‘events’
- For e.g. the Map-Reduce framework ‘counts’ the bytes read/write on HDFS and the local filesystem
- To define your own:
  - static enum Counter {C1, C2}
  - reporter.incrCounter{Counter.C1, 1}
Counters continued...

<table>
<thead>
<tr>
<th>Counter</th>
<th>Map</th>
<th>Reduce</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local bytes read</td>
<td>15,436,320,026</td>
<td>8,575,518,710</td>
<td>24,011,838,736</td>
</tr>
<tr>
<td>Local bytes written</td>
<td>17,333,083,926</td>
<td>8,575,518,710</td>
<td>25,908,602,636</td>
</tr>
<tr>
<td>HDFS bytes read</td>
<td>5,093,892,056</td>
<td>0</td>
<td>5,093,892,056</td>
</tr>
</tbody>
</table>
| HDFS bytes written            | 0              | 31,139,543,728,535 | 31,139,543,728,535 |}

| Job Counters                  |                |                |                  |
| Launched map tasks            | 0              | 0              | 727              |
| Launched reduce tasks         | 0              | 0              | 724              |
| Data-local map tasks          | 0              | 0              | 602              |
| Rack-local map tasks          | 0              | 0              | 96               |

| Map-Reduce Framework          |                |                |                  |
| Map input records             | 57,300,102     | 0              | 57,300,102       |
| Map output records            | 57,300,102     | 0              | 57,300,102       |
| Map input bytes               | 5,093,891,958  | 0              | 5,093,891,958    |
| Map output bytes              | 8,005,032,069  | 0              | 8,005,032,069    |
| Combine input records         | 0              | 0              | 0                |
| Combine output records        | 0              | 0              | 0                |
| Reduce input groups           | 0              | 33,668,268     | 33,668,268       |
| Reduce input records          | 0              | 57,061,253     | 57,061,253       |
| Reduce output records         | 0              | 833,247,066,047| 833,247,066,047  |
Debugging – Oh no!

• Advanced technology
  stderr – Hold on! Where do we find it?
Debugging continued...

- Run job with ‘Local Runner’
  - Set `mapred.job.tracker` to "local"
  - Runs application in single process/thread
- Run on a single-node cluster i.e. your dev-box, with sampled data
- Set `keep.failed.task.files` to `true` and use the `IsolationRunner`
Profiling

• Set `mapred.task.profile` to `true`.

• Use `mapred.task.profile.{maps|reduces}`.

• Hprof support is built-in.

• Use `mapred.task.profile.params` to set options for the debugger.

• Possibly `DistributedCache` for the profiler’s agent.
Tuning

- Tell HDFS and Map-Reduce about your network!
  - Rack locality script: `topology.script.file.name`

- Number of maps
  - Data locality

- Number of reduces
  - You don’t need a single output file!
Tuning continued…

- Amount of data processed per Map
  - Consider fatter maps
  - Custom input format
- Combiner
  - With 0.18 onwards we have multi-level combiners at both Map and Reduce
  - Check to ensure the combiner is useful!
Tuning continued...

- Map-side sort (brr... the voodoo art)
  - `io.sort.mb`
  - `io.sort.factor`
  - `io.sort.record.percent`
  - `io.sort.spill.percent`
Tuning continued...

• Shuffle
  – Map-side
    • Compression for map-outputs
      – mapred.compress.map.output
      – mapred.map.output.compression.codec
    • lzo via libhadoop.so
• tasktracker.http.threads
Tuning continued...

• Shuffle
  – Reduce-side
    • mapred.reduce.parallel.copies
    • mapred.reduce.copy.backoff
    • mapred.job.shuffle.input.buffer.percent
    • mapred.job.shuffle.merge.percent
    • mapred.inmem.merge.threshold
    • mapred.job.reduce.input.buffer.percent
Tuning continued…

- Compress the job output
- Miscellaneous
  - Speculative execution
  - Heap size for the child
  - Re-use jvm for maps/reduces
- Last, not least: Raw Comparators
Questions?