RAIDTool: A First Step to RAID 6 in HDFS

Wittawat Tantisiriroj
Lin Xiao, Bin Fan, and Garth Gibson (CMU)
Robert Chansler (Yahoo!)

PARALLEL DATA LABORATORY
Carnegie Mellon University
Benefit: more storage capacity for users

- Reduce HDFS storage overhead from 200% down to 28%
- Provide 136% more space for users
DiskReduce refresh

• Triplicate every data block
• Encode in background
Implementation goals

• RAID 6 (double-parity)
• Deployed by Apache Hadoop/HDFS
  • As an external tool
• Space efficient
  • Grouping blocks across files within a directory[DiskReduce]
• Highly parallel encoding
  • Use Mapreduce to create parities
Deployment friendly RAID 6

HDFS
- Untouched HDFS source code

RAIDTool
- External tool
  - For encoding & repair
- FileSystem plug-in
  - For online recovery
Space efficient encoding

User actions
- Compute parities
  $\text{raidtool output-dir}$
- Reduce copies
  $\text{hadoop fs --setrep 1 output-dir}$
Mapreduce for parallel encoding

1) Gather a list of blocks & decide RAID sets

2) Compute parities

3) Write metadata (RAID set membership)

MapReduce for parallel encoding

- RAIDTool
- MapReduce
- FS API
- HDFS

A list of blocks

<table>
<thead>
<tr>
<th>f1</th>
<th>f2</th>
<th>f3</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>D2</td>
<td>D3</td>
</tr>
</tbody>
</table>

Parity files are written to HDFS

Metadata files
Plug-in enables online recovery

- **hdfs**://…/output-dir/f2
  - A client gets an exception while reading
- **raid**://…/output-dir/f2
  - ‘raid’ plug-in can detect an exception (e.g. a block is missing)
  - Recovery is done at a client side without modifying data inside HDFS
  - Transparent to a client
Offline recovery (now)

- Without a hook inside HDFS, RAIDTool cannot detect a problem unless a block is read by a client
- Require admin actions to fix problems
  - Use an existing HDFS-Fsck periodically to collect a list of corrupted files
  - Use RAIDTool to recover each corrupted file
Offline recovery (envision)

- HDFS should support pluggable repair modules
  - Recover from a replica within a cluster (current)
  - Recover from a replica from another cluster
  - Recover from a RAID set

HDFS background repair process:

1. Missing block
   - LOCAL
     - No
       - REMOTE
         - No
           - RAID
             - Failed
             - Repaired
           - Repaired
         - Repaired
       - Repaired
     - Repaired
Encoding is fast but reconstruction needs tuning

- 60 nodes (two quad-core 2.83GHz Xeon, 16GB memory, four 7200 rpm SATA 1TB disks, 10 Gigabit Ethernet)
- Dataset: 240GB (3,840 files, each 64MB in size)

<table>
<thead>
<tr>
<th>Operation</th>
<th>User Throughput (GB/s)</th>
<th>Disk I/O (GB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write (Triplication)</td>
<td>1.93</td>
<td>5.80</td>
</tr>
<tr>
<td>Encode (RAID6 8+2)</td>
<td>3.69</td>
<td>4.61</td>
</tr>
<tr>
<td>Repair (after a 2-node failure)</td>
<td>0.23</td>
<td>2.09</td>
</tr>
</tbody>
</table>
Conclusion

• Provide 136% more space for users
• No need to modify Hadoop/HDFS
• RAIDTool is available now
  • Available @ http://issues.apache.org/jira/browse/MAPREDUCE-2036
  • Planned for Hadoop 0.22
• Demo @ poster sessions