Developing a MapReduce Application

Oguzhan Gencoglu
Outline

1. MapReduce Paradigm
   - What is MapReduce
   - MapReduce Workflow

2. Job Tracker
   - Hadoop Default Ports

3. Example
   - Word Count
   - Job Tracker
   - Key Points
Outline

1. MapReduce Paradigm
   - What is MapReduce
   - MapReduce Workflow

2. Job Tracker
   - Hadoop Default Ports

3. Example
   - Word Count
   - Job Tracker
   - Key Points
MapReduce is a software framework for processing (large) data sets in a distributed fashion over several machines.

Core idea

<key, value> pairs
What is MapReduce

MapReduce is a software framework for processing (large) data sets in a distributed fashion over several machines.

Core idea

• Almost all data can be mapped into key, value pairs.
MapReduce is a software framework for processing (large) data sets in a distributed fashion over several machines.

**Core idea**

\[ \langle \text{key}, \text{value} \rangle \text{ pairs} \]

- Almost all data can be mapped into key, value pairs.
- Keys and values may be of any type.
Outline

1. MapReduce Paradigm
   - What is MapReduce
   - MapReduce Workflow

2. Job Tracker
   - Hadoop Default Ports

3. Example
   - Word Count
   - Job Tracker
   - Key Points
MapReduce Workflow

- Write your map and reduce functions
MapReduce Workflow

- Write your map and reduce functions
- Test with a small subset of data
MapReduce Workflow

- Write your map and reduce functions
- Test with a small subset of data
- If it fails use your IDE’s debugger to find the problem
MapReduce Workflow

- Write your map and reduce functions
- Test with a small subset of data
- If it fails use your IDE’s debugger to find the problem
- Run on full dataset
MapReduce Workflow

■ Write your map and reduce functions
■ Test with a small subset of data
■ If it fails use your IDE’s debugger to find the problem
■ Run on full dataset
■ If it fails Hadoop provides some debugging tools
MapReduce Workflow

- Write your map and reduce functions
- Test with a small subset of data
- If it fails use your IDE’s debugger to find the problem
- Run on full dataset
- If it fails Hadoop provides some debugging tools
  - e.g. IsolationRunner: runs a task over the same input which it failed.
MapReduce Workflow

- Write your map and reduce functions
- Test with a small subset of data
- If it fails use your IDE’s debugger to find the problem
- Run on full dataset
- If it fails Hadoop provides some debugging tools
  - e.g. IsolationRunner: runs a task over the same input which it failed.
- Do profiling to tune the performance
Outline

1. MapReduce Paradigm
   - What is MapReduce
   - MapReduce Workflow

2. Job Tracker
   - Hadoop Default Ports

3. Example
   - Word Count
   - Job Tracker
   - Key Points
Hadoop Default Ports

- Handful of ports over TCP.
- Some used by Hadoop itself (to schedule jobs, replicate blocks, etc.).
- Some are directly for users (either via an interposed Java client or via plain old HTTP)

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Default Port</th>
<th>Configuration Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Namenode</td>
<td>50070</td>
<td>dfs.http.address</td>
</tr>
<tr>
<td>Datanodes</td>
<td>50075</td>
<td>dfs.datanode.http.address</td>
</tr>
<tr>
<td>Secondarynamenode</td>
<td>50090</td>
<td>dfs.secondary.http.address</td>
</tr>
<tr>
<td>Backup/Checkpoint node?</td>
<td>50105</td>
<td>dfs.backup.http.address</td>
</tr>
<tr>
<td>MR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jobtracker</td>
<td>50030</td>
<td>mapred.job.tracker.http.address</td>
</tr>
<tr>
<td>Tasktrackers</td>
<td>50060</td>
<td>mapred.tasktracker.http.address</td>
</tr>
</tbody>
</table>

? Replaces secondarynamenode in 0.21.
Outline

1. MapReduce Paradigm
   - What is MapReduce
   - MapReduce Workflow

2. Job Tracker
   - Hadoop Default Ports

3. Example
   - Word Count
   - Job Tracker
   - Key Points
Task: Counting the word occurrences (frequencies) in a text file (or set of files).

\[ \langle \text{word, count} \rangle \text{ as } \langle \text{key, value} \rangle \text{ pair} \]

**Mapper:** Emits \[ \langle \text{word, 1} \rangle \] for each word (no counting at this part).

*Shuffle* in between: pairs with same keys grouped together and passed to a single machine.

**Reducer:** Sums up the values (1s) with the same key value.
The overall MapReduce word count process

Input: Deer Bear River, Car, Car, River, Deer, Car, Bear

Splitting: Deer Bear River → Car, Car, River → Deer, Car, Bear

Mapping: Deer, 1 Bear, 1 River, 1 → Car, 1 Car, 1 Car, 1 → Deer, 1 Deer, 1 → River, 1 River, 1

Shuffling: Bear, 1 Bear, 1 → Bear, 2
Car, 1 Car, 1 Car, 1 → Car, 3
Deer, 1 Deer, 1 → Deer, 2
River, 1 River, 1 → River, 2

Reducing: Bear, 2 Car, 3 Deer, 2 River, 2

Final result: Bear, 2 Car, 3 Deer, 2 River, 2
Outline

1. MapReduce Paradigm
   - What is MapReduce
   - MapReduce Workflow

2. Job Tracker
   - Hadoop Default Ports

3. Example
   - Word Count
   - Job Tracker
   - Key Points
130 Hadoop Map/Reduce Administration

Cluster Summary

<table>
<thead>
<tr>
<th>Maps</th>
<th>Reduces</th>
<th>Total Submissions</th>
<th>Nodes</th>
<th>Map Task Capacity</th>
<th>Reduce Task Capacity</th>
<th>Avg. Tasks/Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4.00</td>
</tr>
</tbody>
</table>

Running Jobs

Completed Jobs

<table>
<thead>
<tr>
<th>Jobid</th>
<th>User</th>
<th>Name</th>
<th>Map % Complete</th>
<th>Map Total</th>
<th>Maps Completed</th>
<th>Reduce % Complete</th>
<th>Reduce Total</th>
<th>Reduces Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>job_201411172241_0003</td>
<td>hadoop-user</td>
<td>streamjob16751.jar</td>
<td>100.00%</td>
<td>2</td>
<td>2</td>
<td>100.00%</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>job_201411172241_0004</td>
<td>hadoop-user</td>
<td>streamjob28687.jar</td>
<td>100.00%</td>
<td>2</td>
<td>2</td>
<td>100.00%</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Failed Jobs

<table>
<thead>
<tr>
<th>Jobid</th>
<th>User</th>
<th>Name</th>
<th>Map % Complete</th>
<th>Map Total</th>
<th>Maps Completed</th>
<th>Reduce % Complete</th>
<th>Reduce Total</th>
<th>Reduces Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>job_201411172241_0001</td>
<td>hadoop-user</td>
<td>streamjob64235.jar</td>
<td>100.00%</td>
<td>2</td>
<td>2</td>
<td>100.00%</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Local logs

Log directory, Job Tracker History

Hadoop, 2014.
### Completed Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Complete</th>
<th>Status</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Errors</th>
<th>Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>task 200904110811_0003_m_000043</td>
<td>100.00%</td>
<td>hdfs://ip-10-250-110-47.ec2.internal/user/root/input/ncdc/all/1949.gz:0+220338475</td>
<td>11-Apr-2009 09:00:06</td>
<td>11-Apr-2009 09:01:25 (1mins, 18sec)</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>task 200904110811_0003_m_000044</td>
<td>100.00%</td>
<td>Detected possibly corrupt record: see logs.</td>
<td>11-Apr-2009 09:00:06</td>
<td>11-Apr-2009 09:01:28 (1mins, 21sec)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>task 200904110811_0003_m_000045</td>
<td>100.00%</td>
<td>hdfs://ip-10-250-110-47.ec2.internal/user/root/input/ncdc/all/1970.gz:0+208374610</td>
<td>11-Apr-2009 09:00:06</td>
<td>11-Apr-2009 09:01:28 (1mins, 21sec)</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
Name Node

NameNode '130.230.16.37:9000'

Started: Tue Nov 18 18:09:31 PST 2014
Version: 0.18.0, r686010
Upgrades: There are no upgrades in progress.

Browse the filesystem

Cluster Summary

25 files and directories, 28 blocks = 53 total. Heap Size is 5.98 MB / 992.31 MB (0%)
Capacity : 23.73 GB
DFS Remaining : 21.42 GB
DFS Used : 529.41 KB
DFS Used% : 0 %
Live Nodes : 1
Dead Nodes : 0

Live Datanodes : 1

<table>
<thead>
<tr>
<th>Node</th>
<th>Last Contact</th>
<th>Admin State</th>
<th>Size (GB)</th>
<th>Used (%)</th>
<th>Used (%)</th>
<th>Remaining (GB)</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>hadoop-desk</td>
<td>2</td>
<td>In Service</td>
<td>23.73</td>
<td>0</td>
<td></td>
<td>21.42</td>
<td>28</td>
</tr>
</tbody>
</table>

Dead Datanodes : 0

Oguzhan Gencoglu | Developing a MapReduce Application
Outline

1. MapReduce Paradigm
   - What is MapReduce
   - MapReduce Workflow

2. Job Tracker
   - Hadoop Default Ports

3. Example
   - Word Count
   - Job Tracker
   - Key Points
Key Points

- Test mapper and reducer outside hadoop.
Key Points

- Test mapper and reducer outside hadoop.
- Copy your MapReduce function and files to DFS.
Key Points

- Test mapper and reducer outside hadoop.
- Copy your MapReduce function and files to DFS.
- Test mapper and reducer with hadoop using a small portion of the data.
Key Points

- Test mapper and reducer outside hadoop.
- Copy your MapReduce function and files to DFS.
- Test mapper and reducer with hadoop using a small portion of the data.
- Track the jobs, debug, do profiling