Hadoop Tuner: Tune Configuration Parameters of Hadoop System

National Taiwan University & Industrial Technology Research Institute of Taiwan

Chi-ou(Ray) Chen
Who am I?

• Graduate student from National Taiwan University
  – Advisor: SW Liao

• Researcher from Industrial Technology Research Institute

• What I do?
  – To optimize the performance of Hadoop MapReduce jobs
Hadoop MapReduce Configuration

• A Hadoop job $j = \langle p, d, c, r \rangle$, $p$ is program, $d$ is input data, $c$ is configuration parameters and $r$ is cluster resource

• Configuration parameters of Hadoop:
  – Specify how the job will be executed
  – Is influential in the performance (execution time)

• Misconfiguration will degrade the performance

• For Hadoop MapReduce, misconfiguration accounts for 60 percent of the frequency of error event and degradation [1]

Hadoop MapReduce Configuration (Cont’d)

• Configuration parameters mainly control four performance-related factors:
  – Parallelism
  – Memory capacity
  – Job flow trigger point
  – Data Compression

Appendix I: Execution Flow of a MapReduce Job
How to tune parameters?

• Hadoop tuner: Machine Learning-based Tuner
  – Predictor:
    • Predict execution time by machine learning models
  – Optimizer:
    • Search the approximately optimal parameters
Hadoop Tuner Architecture
What’s Task Feature?

• Task Feature
  – Stand for the task behavior, and influence the job execution time
  – Select median and deviation execution time of map\reduce as task feature
  – And also the number of waves
Hadoop MapReduce System

Job Execution Time

- Master Node
  - TaskTracker
  - JobTracker
- Slave Node
  - Map Slot
  - Map Task
  - Reduce Slot
  - Reduce Task

2 Map Waves
1 Reduce Wave
Distribution of Map Tasks Execution Time

Configuration Parameter Sample 1:

Median: 6339.5ms, Deviation: 675.4289ms

Median: 6343ms, Deviation: 690.9816ms

Median: 6339ms, Deviation: 679.822ms
Distribution of Map Tasks Execution Time

Configuration Parameter Sample2:

Median: 9969ms, Deviation: 1457.947ms

Median: 10252.5ms, Deviation: 1511.778ms

Median: 10256ms, Deviation: 1460.173ms
Distribution of Map Tasks Execution Time

Configuration Parameter Sample 3:

Median: 13883ms, Deviation: 1867.57ms

Median: 13871.5ms, Deviation: 1871.261ms

Median: 13867ms, Deviation: 1943.13ms
Machine Learning Algorithm in Predictor

• Our case:
  – The features interact in complicated and non-linear way

• Ensemble Regression:
  – e.g. random forest regression, extra trees regression
  – Python machine learning toolkit (sklearn)
  – Regression tree sub-divide the interactions

=> The interactions are more manageable
Optimizer

• Optimizer is to search optimal parameters by enumerating parameters and invoking predictor repeatedly and strategically

\[ c_{opt} = \arg\min_{c \in PS} F(p, d, c, r) \]
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Recursive Random Search

2 Dimensional Parameter Space
Evaluation

• Benchmark:
  – **WordCount**: input data from random text writer (Hadoop native application)
  – **TeraSort**: input data from teragen (Hadoop native application)

• Environment:
  – 8 nodes
  – Intel i7-4770 core, 32GB RAM and 2TB disk space
  – Hadoop-1.2.1
Accuracy of Predictor

• Metric:

  – MAPE (Mean Absolute Percentage Error)

  \[
  MAPE = \left( \frac{1}{N} \sum \frac{|\text{Actual} - \text{Predicted}|}{\text{Actual}} \right) \times 100\%
  \]

  – Pred(25) = \frac{\# \text{of observations with relative error} \leq 25\%}{\# \text{of total observations}} \times 100\%

• Accuracy:

<table>
<thead>
<tr>
<th></th>
<th>TeraSort</th>
<th>WordCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPE</td>
<td>11.72%</td>
<td>9.71%</td>
</tr>
<tr>
<td>Pred(25)</td>
<td>90.12%</td>
<td>93.86%</td>
</tr>
</tbody>
</table>
Compare with Default Configuration

**TeraSort**

**WordCount**
Q&A
Appendix I: Execution Flow of a MapReduce Job
Appendix III: Importance of Map Parameters

Map_Config

TeraSort
WordCount

MapHeapSize  MapTaskMax  SplitSize  SortMB  SortPer  RecordPer  JVMReuse
Appendix III: Importance of Reduce Parameters

Reduce_Config

TeraSort
- RedHeapSize
- RedSlowstart
- OutputCompress

WordCount
- RedTaskMax
- inMemMergeThreshold
- ShuffleInputPer
- JVMReuse

- RedTaskNum
- ShuffleMergePer
- ReduceInputPer