Accelerating SQL on Hadoop* with Big Data Benchmark for BigBench Spec

STO Big Data Performance team
Contact us: sto-bigdata-qa-prc@intel.com
About Us

- Intel SSG STO big data performance team
- Working on performance tuning and optimization for Hadoop* ecosystem since 2011
- Contributors of the benchmark, improve the performance with the benchmark
Agenda

- Big Data Benchmark – Towards an Industry Standard
- Contribution to the benchmark
- Accelerating SQL on Hadoop* Performance with the benchmark
Big Data Performance Benchmarking

**Micro Benchmarks testing**

**End-to-End Benchmark**

- proposal accepted by TPC-TC for consideration

**Synthetic, Illustrative benchmarks suitable for Regression**

- Real-world, Instructive benchmarks suitable for End-to-End testing
- Open source standards based
- Industry consortium proposed
- Support from 10+ ecosystem partners

*Get started with the benchmark now:*

https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench

Software and Services
Benchmark Workload: Retail Section

- Cross-selling
- Customer micro-segmentation
- Sentiment analysis
- Enhancing multichannel consumer experiences

- Assortment optimization
- Pricing optimization

- Performance transparency
- Product return analysis

- Inventory management

- Price comparison

- Marketing (18.6%)
- Merchandising (16.7%)
- Operations (13.3%)
- Supply chain (6.7%)
- New business models (3.3%)
Benchmark Workload: Data Model

- Derived from TPC-DS: star schema with 6 fact tables, representing store sales, and online sales channels

- Additional big data specific dimensions to analyze user behavior

- Structured: TPC-DS + market prices
- Semi-structured: website click-stream
- Unstructured: customers’ reviews
## Benchmark Workload: Queries

<table>
<thead>
<tr>
<th>Data Sources</th>
<th>Number of Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured</td>
<td>18</td>
</tr>
<tr>
<td>Semi-structured</td>
<td>7</td>
</tr>
<tr>
<td>Un-structured</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analytic techniques</th>
<th>Number of Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics analysis</td>
<td>6</td>
</tr>
<tr>
<td>Data mining</td>
<td>17</td>
</tr>
<tr>
<td>Reporting</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Types</th>
<th>Number of Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure HiveQL</td>
<td>14</td>
</tr>
<tr>
<td>Mahout</td>
<td>5</td>
</tr>
<tr>
<td>OpenNLP</td>
<td>5</td>
</tr>
<tr>
<td>Custom MR</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query</th>
<th>Input Datatype</th>
<th>Processing Model</th>
<th>Query</th>
<th>Input Datatype</th>
<th>Processing Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Structured</td>
<td>Java MR</td>
<td>#16</td>
<td>Structured</td>
<td>Java MR (OpenNLP)</td>
</tr>
<tr>
<td>#2</td>
<td>Semi-structured</td>
<td>Java MR</td>
<td>#17</td>
<td>Structured</td>
<td>HiveQL</td>
</tr>
<tr>
<td>#3</td>
<td>Semi-structured</td>
<td>Python Streaming MR</td>
<td>#18</td>
<td>Unstructured</td>
<td>Java MR (OpenNLP)</td>
</tr>
<tr>
<td>#4</td>
<td>Semi-structured</td>
<td>Python Streaming MR</td>
<td>#19</td>
<td>Structured</td>
<td>Java MR (OpenNLP)</td>
</tr>
<tr>
<td>#5</td>
<td>Semi-structured</td>
<td>HiveQL</td>
<td>#20</td>
<td>Structured</td>
<td>Java MR (Mahout)</td>
</tr>
<tr>
<td>#6</td>
<td>Structured</td>
<td>HiveQL</td>
<td>#21</td>
<td>Structured</td>
<td>HiveQL</td>
</tr>
<tr>
<td>#7</td>
<td>Structured</td>
<td>HiveQL</td>
<td>#22</td>
<td>Structured</td>
<td>HiveQL</td>
</tr>
<tr>
<td>#8</td>
<td>Semi-structured</td>
<td>HiveQL</td>
<td>#23</td>
<td>Structured</td>
<td>HiveQL</td>
</tr>
<tr>
<td>#9</td>
<td>Structured</td>
<td>HiveQL</td>
<td>#24</td>
<td>Structured</td>
<td>HiveQL</td>
</tr>
<tr>
<td>#10</td>
<td>Unstructured</td>
<td>Java MR (OpenNLP)</td>
<td>#25</td>
<td>Structured</td>
<td>Java MR (Mahout)</td>
</tr>
<tr>
<td>#11</td>
<td>Unstructured</td>
<td>Java MR</td>
<td>#26</td>
<td>Structured</td>
<td>Java MR (Mahout)</td>
</tr>
<tr>
<td>#12</td>
<td>Semi-structured</td>
<td>HiveQL</td>
<td>#27</td>
<td>Unstructured</td>
<td>Java MR (OpenNLP)</td>
</tr>
<tr>
<td>#13</td>
<td>Structured</td>
<td>HiveQL</td>
<td>#28</td>
<td>Unstructured</td>
<td>Java MR (Mahout)</td>
</tr>
<tr>
<td>#14</td>
<td>Structured</td>
<td>HiveQL</td>
<td>#29</td>
<td>Structured</td>
<td>Python Streaming MR</td>
</tr>
<tr>
<td>#15</td>
<td>Structured</td>
<td>Java MR (Mahout)</td>
<td>#30</td>
<td>Semi-Structured</td>
<td>Python Streaming MR</td>
</tr>
</tbody>
</table>
How to Run Big-Bench Workload

- Execute benchmark with different engine
  Big-Bench/conf/userSettings.conf
  1) export BIG_BENCH_DEFAULT_ENGINE="hive"
  2) export BIG_BENCH_DEFAULT_ENGINE="spark"

- Run BigBench with -e option for specific engine

  $BB-HOME/bin/bigBench runBenchmark -m 80 -f 1000 -e spark

  $BB-HOME/bin/bigBench runQuery -e spark -q 1
Contribution to the benchmark

- Extend the mainstream Hadoop engines: Spark SQL, Hive on Spark, Impala…
- Query optimization
- Query level engine setting
Roadmap for the benchmark on Spark SQL

- Filed 14 performance and conformance JIRAs. Push and co-work with community to fix these issues. Before that, no queries can work on Spark SQL for many issues and Spark SQL performance is very poor especially join performance.

- Invited by Spark community to support Spark release with the benchmark and vote for Spark 1.4 release

- Involved since Spark 1.2.0.

- Spark SQL graduates from an alpha project since 1.3.0

<table>
<thead>
<tr>
<th>JIRA ID</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPARK-5202</td>
<td>HiveContext doesn’t support the Variables Substitution</td>
</tr>
<tr>
<td>SPARK-5237</td>
<td>UDTF don’t work with multi-alias of multi-columns as output on Spark SQL</td>
</tr>
<tr>
<td>SPARK-5364</td>
<td>HiveQL transform doesn’t support the non output clause</td>
</tr>
<tr>
<td>SPARK-5707</td>
<td>Enabling spark.sql.codegen throws ClassNotFound exception</td>
</tr>
<tr>
<td>SPARK-5855</td>
<td>'explain' command in SparkSQL don’t support to analyze the DDL ‘VIEW’</td>
</tr>
<tr>
<td>SPARK-7023</td>
<td>[Spark SQL] Can’t populate table size information into Hive metastore when create table or insert into table</td>
</tr>
<tr>
<td>SPARK-7044</td>
<td>[Spark SQL] Query would hang when using scripts in SQL statement</td>
</tr>
<tr>
<td>SPARK-7051</td>
<td>Parquet compression does not work for Spark SQL loading</td>
</tr>
<tr>
<td>SPARK-7268</td>
<td>[Spark SQL] Throw ‘Shutdown hooks cannot be modified during shutdown’ on YARN</td>
</tr>
<tr>
<td>SPARK-7119</td>
<td>ScriptTransform doesn’t consider the output data type</td>
</tr>
<tr>
<td>SPARK-7623</td>
<td>Spark prints “SelectChannelConnector@0.0.0.0:4040: java.net.BindException: Address already in use” when run 2 spark in parallel</td>
</tr>
<tr>
<td>ISSUE14 for PR 5542</td>
<td>java.lang.ClassCastException: java.lang.Double cannot be cast to java.lang.Long occurred in SET spark.sql.aggregate2=true configuration</td>
</tr>
<tr>
<td>SPARK-5791</td>
<td>Show poor performance when multiple table do join operation</td>
</tr>
<tr>
<td>SPARK-7825</td>
<td>Poor performance in Cross Product due to no combine operations for small files.</td>
</tr>
</tbody>
</table>

Pushing community to make a more stable and faster Spark SQL

Software and Services

System Technologies and Optimization
Enhance Hive on Spark with the benchmark

- Involved since Hive on Spark beta release
- Reported 10 critical conf/perf JIRA issues to Community (all fixed/verified now)
- Co-worked with community to analyze the issue and find root cause
- Delivered Hive on Spark performance evaluation report (based on the benchmark) regularly to dev

we can make the optimization much more expressive!

Software and Services  System Technologies and Optimization
Configuration and Tuning for SQL on Hadoop*

- Complex parameters tuning for Apache* Hadoop*
  - Linux* configuration
  - Java* options and GC tuning
  - HDFS parameters
  - YARN options, flags
  - Job configuration

- Target: Improve resource utilization and thus performance
Tuning JOIN for Spark SQL

- BroadcastHashJoin (AKA Map Join)
- Broadcast the small RDD to all worker nodes
- spark.sql.autoBroadcastJoinThreshold

```
SELECT *
FROM inventory inv
JOIN (SELECT i_item_id, i_item_sk
FROM item
WHERE i_current_price > $\{(hiveconf:q22_i_current_price_min}\}
AND i_current_price < $\{(hiveconf:q22_i_current_price_max}\}
) items
ON inv.i_item_sk = items.i_item_sk
JOIN warehouse w ON inv.warehouse_sk = w.warehouse_sk
JOIN date_dim d ON inv.inv_date_sk = d.d_date_sk
WHERE datediff(d.date, '{\$(hiveconf:q22_date)}') > -30
AND datediff(d.date, '{\$(hiveconf:q22_date)}') <= 30
) q22_coalition_22
```

~4.9GB

~73.5MB

~1.7MB

x 4.22 speedup
Example: Query 10

... 
INSERT INTO TABLE ${hiveconf:RESULT_TABLE}
SELECT extract_sentiment(pr_item_sk,pr_review_content) AS (pr_item_sk,
review_sentence, sentiment, sentiment_word)
FROM product_reviews;

... 
INSERT INTO TABLE ${hiveconf:RESULT_TABLE}
SELECT extract_sentiment(pr_item_sk,pr_review_content) AS (pr_item_sk,
review_sentence, sentiment, sentiment_word)
FROM ( 
SELECT pr_item_sk,pr_review_content FROM product_reviews DISTRIBUTE
BY length(pr_review_content) 
) pr

- Data is generated randomly
- Some map tasks have to process far more data
- Caused typical data skew issue that the query time
is determined by the slowest task
Next Steps

- Improve Hive Analytics/Window Functions

- Research more advanced tuning process for SQL on Hadoop*

- For more details about our tuning SQL on Hadoop:

  Contact us: sto-bigdata-qa-prc@intel.com
Notices and Disclaimers

- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.
- This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.
- The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.
- Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.
- Intel, the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

© 2015 Intel Corporation.