BigBench V2: The New and Improved BigBench

Ahmad Ghazal, Todor Ivanov, Pekka Kostamaa, Alain Crolotte, Ryan Voong, Mohammed Al-Kateb, Waleed Ghazal, Roberto V. Zicari
Contents

- **Background**
  - BigBench and TPCx-BB
  - Other Benchmarks
  - BigBench Shortcomings

- **BigBench V2**
  - Simplified Data Model
  - New Generator
  - New Workload specification

- **Evaluation**
  - Proof of concept on Hive
  - Some queries on Spark and Drill
Background - BigBench

- **End to end benchmark**
  - On top of TPC-DS (decision support on retail business)
    - Add semi-structured and un-structured data
  - *Focus on:* Parallel DBMS and MR engines

- **Literature:**
  - Initial work presented at 1st WBDB, San Jose
    - Full spec at 3rd WBDB, Xian, China
  - Collaboration with Industry & Academia
    - Teradata, University of Toronto, InfoSizing, Oracle
  - SIGMOD 2013 paper:
    - 214 citations “google scholar” and 44 on “ACM DL”

- **Adopted by TPC as TPCx-BB**
  - Based on HIVE HQL
Background – BigBench – Data Model
Background – BigBench – Workload

- **30 queries**
  - Business problems: retail big data analytics “McKinsey report”
    - Marketing
    - Merchandising
    - Operations
    - Supply chain and Reporting (customers and products)
  - Technical dimensions:
    - Data Source: structured, semi-structured and un-structured
    - Processing type dimension: Declarative (SQL, HQL), Procedural and Both
  - Analytic technique dimension
    - Statistical analysis: correlation analysis, time-series, regression
    - Data mining: classification, clustering, association mining, pattern analysis and text analysis
    - Simple reporting
Background – BigBench – Limitations

- **Data Model Limitations:**
  - The structured component from TPC-DS
    - 26 tables
    - Complex snowflake-like schema.
    - Big Data Models: simple star schema
  - Semi-structured web-logs
    - Treated as structured table.
    - In real life, web-logs are modeled as key-value pairs with unknown schema.
    - Schema known at query time “late binding”

- **Workload Limitations:**
  - Eleven (out of thirty) queries from TPC-DS queries.
  - Queries are complex SQL on structured data
  - Not typical of big data workloads.
BigBench V2 – Simplified Data Model

• 1 – many relationship
• Semi-structured: key-value WebLog
• Un-structured: Product Reviews
BigBench V2 – Simplified Data Model

- **Structured Part**
  - 2 fact tables: store sales and web sales
  - Medium table: user
  - 3 dimension tables: store, product and web page

- **Semi-structured Part:**
  - Key-value pairs representing user clicks
  - Keys corresponding to structured part and random keys and values
  - Example:
    - `<user,user1> <time,t1> <webpage,w1> <product,p1>`
    - `<key1,value1> <key2,value2> ... <key100,value100>`

- **Unstructured Part**:
  - mostly same as original BigBench
BigBench V2 – Data Generator

- Generator developed for simplified data model
- **Weblogs and Web Sales**:
  - driven by user sessions & users
  - Users: registered and guest
  - Browsing, abandoned shopping carts and orders
  - Weblogs key-vale produced as JOSN
- **Linear growth by scale factor**:
  - User, store sale, web sale, weblogs and product reviews
- **Sub-linear by scale factor**:
  - products and stores
- **Static**:
  - webpage
- **Configuration file**
BigBench V2 – New Workload

- **Main goal**
  - De-emphasize structured part of data
    - Remove all 11 DS queries
    - Remove 2 queries using “sale returns”
  - Mandate late binding in query execution

- **New Queries**
  - 13 new queries
    - Mostly on weblogs
  - 17 old queries from BigBench
    - Re-written on simplified schema
BigBench V2 – New Workload

- 13 New Queries
  - About
    - products viewed and purchased
    - user behavior/sessions
  - Examples
    - Q_{5}: Find the 10 most browsed products.
    - Q_{6}: Find the 5 most browsed products that are not purchased.
    - Q_{7}: List users with more than 10 sessions. A session is defined as a 10-minute window of clicks by a user.
    - Q_{9}: Find the average number of sessions per registered users per month. Display the top ten users.
# BigBench V2 – New Workload

- **Business Category**
  No major/intended change

- **Query Type**
  More mix of declarative and procedural

- **Data Source**
  More focus on semi-structured

<table>
<thead>
<tr>
<th>Business Category</th>
<th>No. of queries</th>
<th>Percentage</th>
<th>BigBench V2</th>
<th>No. of queries</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>18</td>
<td>60.0%</td>
<td>20</td>
<td>69.0%</td>
<td></td>
</tr>
<tr>
<td>Merchandising</td>
<td>5</td>
<td>16.7%</td>
<td>3</td>
<td>10.3%</td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td>4</td>
<td>13.3%</td>
<td>2</td>
<td>6.9%</td>
<td></td>
</tr>
<tr>
<td>Supply chain</td>
<td>2</td>
<td>6.77%</td>
<td>1</td>
<td>3.3%</td>
<td></td>
</tr>
<tr>
<td>New business models</td>
<td>1</td>
<td>3.3%</td>
<td>4</td>
<td>13.8%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Type</th>
<th>No. of queries</th>
<th>Percentage</th>
<th>BigBench V2</th>
<th>No. of queries</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declarative</td>
<td>10</td>
<td>33.3%</td>
<td>7</td>
<td>24.1%</td>
<td></td>
</tr>
<tr>
<td>Procedural</td>
<td>7</td>
<td>23.3%</td>
<td>4</td>
<td>13.3%</td>
<td></td>
</tr>
<tr>
<td>Declarative &amp; Procedural</td>
<td>13</td>
<td>43.3%</td>
<td>19</td>
<td>65.6%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Source</th>
<th>No. of queries</th>
<th>Percentage</th>
<th>BigBench V2</th>
<th>No. of queries</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured</td>
<td>18</td>
<td>60.0%</td>
<td>5</td>
<td>16.7%</td>
<td></td>
</tr>
<tr>
<td>Semi-Structured</td>
<td>7</td>
<td>23.3%</td>
<td>20</td>
<td>66.7%</td>
<td></td>
</tr>
<tr>
<td>Unstructured</td>
<td>5</td>
<td>16.7%</td>
<td>5</td>
<td>16.7%</td>
<td></td>
</tr>
</tbody>
</table>
BigBench V2 – Late Binding

- **Late binding**: Schema at query time
  - Weblogs has 1000’s of different keys
  - Hard to parse up-front
  - Most keys are not required

- **BigBench V2 mandates late binding unlike BigBench**
  - No pre-parsing or pre-processing weblogs
  - Data generator produce weblogs as simple JSON format
  - Produce relational format of specific keys from weblogs

- **Various “late binding” implementations**
  - SparkSQL and Drill have native support for JSON and can parse web-logs directly.
  - Hive needs an internal or external user-defined function (UDF) to parse web-logs.
Proof of Concept

- **Objective is to**
  - Show feasibility of benchmark:
    - no serious tuning effort
  - Different ways of implementing late binding

- **Setup**
  - Benchmark on **Hive**
  - 30 Queries in HQL
  - Hardware
    - Cluster with 4 nodes
    - Each: 6 cores, 32 GB and 1 TB disk
  - Software
    - Ubuntu Server 14.04.1
    - Cloudera Distribution of Hadoop (CDH) versions 5.5.1
    - Hive 1.1.0
  - Data Generation : SF = 1
Proof of Concept - Implementation

- SF=1 data produced in 8 files
  - 6 for structured tables
  - File with JSON format for weblogs
  - File for product reviews with text for reviews
    - No change from BigBench

- Structured tables created as Hive tables and loaded from files

- DDL example for user table
  
  ```sql
  CREATE TABLE IF NOT EXISTS user
  ( u_user_id bigint, u_name string)
  ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' 
  STORED AS TEXTFILE
  LOCATION 'hdfsDataPath/user';
  ```
Proof of Concept – Implementation continued

- Weblogs implemented as external table with one text field
  
  ```
  CREATE EXTERNAL TABLE IF NOT EXISTS web_logs (line string)
  ROW FORMAT DELIMITED LINES TERMINATED BY '\n'
  STORED AS TEXTFILE
  LOCATION 'hdfsPath/web_logs/clicks.json';
  ```

- Late binding implemented through UDF json parser
  
  - Json_tuple
    - Input : record number and key
    - Output : value
    - `json_tuple (web_logs.line, 'wl_webpage_name')`
Proof of Concept – Implementation continued

- **Q16 Hive QL**
- **Find number of page visits by page name.**
  
  Select `wl_webpage_name`, `count(*)` as `cnt` from
  
  ```
  web_logs
  lateral view
  json_tuple (web_logs.line, 'wl_webpage_name') logs as `wl_webpage_name`
  ```

  Where `wl_webpage_name` is not null
  
  group by `wl_webpage_name`
  
  order by `cnt` desc
  
  limit 10;

- **Other Options for Late binding**
  - Hive Streaming in combination with Python scripts

- **Procedural constructs**
  - Native UDF for sessionize and path functions
Proof of Concept - Experiments

- queries shows variation in run time
- 20 queries require late binding
Proof of Concept – Other Engines

- **SparkSQL & Drill**
  - Have native support for json
  - We ran few queries for exercising variety in late binding

- **Q16 Drill:**
  
  ```sql
  select
  wl_webpage_name, count(*) as cnt
  from
  /* using late binding */
  hdfs.'/hdfs_path/clicks.json'
  Where wl_webpage_name is not null
  group by wl_webpage_name
  order by cnt desc
  limit 10;
  ```
Summary

- **BigBench V2 - a major rework of BigBench**
  - Separate from DS and take care of late binding

- **Data Model**
  - New data model and generator reflect Big Data simple data models and late binding requirement.
  - Custom made scale factor-based data generator for all components

- **Workload**
  - All 11 TPC-DS queries are replaced with new queries in BigBench V2.
  - New queries with similar business questions - focus on analytics on the semi-structured web-logs.

- **Proof of concepts**
  - Rigorous/complete proof of concept on Hive.
  - Illustrates the feasibility and self-containment of the benchmark.
  - Highlights cost of late binding and variations among different engines.
Future Work

- **Share BigBench V2 with community**
  - Open source
  - Connect with WBDB community

- **Propose enhancing TPCx-BB using BigBench V2**
  - Collaborate on making the necessary changes.

- **Add streaming to BigBench**
  - On going work
  - Velocity not covered
  - Appropriate for web sales and weblogs
  - Support real time analytics
    - Monitoring number of visits and abandoned shopping carts
    - Monitoring sales of a hot item to measure operation flows.
Thank you

www.huawei.com

Copyright©2011 Huawei Technologies Co., Ltd. All Rights Reserved.
The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.