LDBC Social Network Benchmark

Interactive Workload

Arnau Prat
DAMA – UPC
Task Force Members

- Alex Averbuch (Neo Technologies)
- Moritz Kaufmann (TU München)
- Marcus Paradies (SAP)
- Arnau Prat (DAMA-UPC/Sparsity)
- Peter Boncz (CWI)
- Orri Erling (Google)

- And Special Thanks to many others that have contributed so far
Summary of SNB-Interactive

- Simple but challenging interactive queries on top of a social network site
  - Interactive queries
  - Flexible: Declarative and API based systems
  - Systems of different scales
  - Latency and throughput are both important
  - Easy to adopt
- All software and docs at https://github.com/ldbc
  - LDBC Datagen
  - LDBC Driver
  - Validation Sets
  - Specification
LDBC SNB Datagen

- Generates a realistic social network with the Facebook degree distribution (persons, groups, posts, likes, etc.)
  - Correlated graph → Similar people have a larger probability to connected, correlated attributes, etc.
  - Non-uniform/Spiky activity volume
  - Scalable (Apache Hadoop based)
  - Deterministic → Allows a fair comparison between SUTs and reproducibility of benchmark executions
LDBC SNB Datagen

- Scale Factors
  - 1,3,10,30,100,300,1000
  - Based on the size of the dataset on dist in CSV format

<table>
<thead>
<tr>
<th>SF</th>
<th>Relations</th>
<th>Persons</th>
<th>Messages</th>
<th>Activity</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF1</td>
<td>20M</td>
<td>11K</td>
<td>3M</td>
<td>3 years</td>
<td>1GB</td>
</tr>
<tr>
<td>SF10</td>
<td>200M</td>
<td>73K</td>
<td>30M</td>
<td>3 years</td>
<td>10GB</td>
</tr>
<tr>
<td>SF100</td>
<td>2000M</td>
<td>499K</td>
<td>300M</td>
<td>3 years</td>
<td>100GB</td>
</tr>
<tr>
<td>SF1000</td>
<td>20000M</td>
<td>3600K</td>
<td>3000M</td>
<td>3 years</td>
<td>1000GB</td>
</tr>
</tbody>
</table>

* approximate numbers
LDBC SNB Datagen

• 90% of the network is output as CSV to be bulk loaded
• The rest 10% is output as update streams
  – This guarantees the properties of the network are preserved
• Substitution parameters for each complex read query type
  – Parameter binding to reduce variability between queries
LDBC SNB Interactive queries

- 14 Complex reads
  - Interactive yet complex
  - target choke-points
  - Explores the neighborhood of a starting node or path between a pair of nodes
  - Example:
    - Query 6: Given a start Person and some Tag, find the other Tags that occur together with this Tag on Posts that were created by start Person’s friends and friends of friends
LDBC SNB Interactive queries

• 7 Short reads
  – balance read/write ratio of workload (70/30)
  – represent queries to populate the website
  – mimic user behavior around the social network
  – Example:
    • Given a start Person, retrieve their first name, last name, birthday, IP address, browser, and city of residence
    • Given a start Person, retrieve all of their friends, and the date at which they became friends

• 8 Update queries
  – Add content produced by the users, do not remove
LDBC Workload Driver

- Responsible of generating the Workload = Stream of operations
  - scheduled start time (real time)
  - type (e.g. ComplexQuery1)
  - parameters (e.g. Person ID)
LDBC Workload Driver

- **Updates**
  - substitution parameters read from datagen update streams
  - time stamps ("simulation time") read from datagen update streams

- **Complex Reads**
  - substitution parameters read from datagen files
  - scheduled start times assigned by driver as multiples of update frequency
    - Not all the queries are the same complexity \(d, d^2\) and \(d^3\). \(d =\) average degree
    - We want all the queries to take about the same time (this is vendor dependant)
LDBC Workload Driver - Example

**Query mix for SF10**

<table>
<thead>
<tr>
<th>Query</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>26</td>
</tr>
<tr>
<td>Q2</td>
<td>37</td>
</tr>
<tr>
<td>Q3</td>
<td>106</td>
</tr>
<tr>
<td>Q4</td>
<td>36</td>
</tr>
<tr>
<td>Q5</td>
<td>72</td>
</tr>
<tr>
<td>Q6</td>
<td>316</td>
</tr>
<tr>
<td>Q7</td>
<td>48</td>
</tr>
<tr>
<td>Q8</td>
<td>9</td>
</tr>
<tr>
<td>Q9</td>
<td>384</td>
</tr>
<tr>
<td>Q10</td>
<td>37</td>
</tr>
<tr>
<td>Q11</td>
<td>20</td>
</tr>
<tr>
<td>Q12</td>
<td>44</td>
</tr>
<tr>
<td>Q13</td>
<td>19</td>
</tr>
<tr>
<td>Q14</td>
<td>49</td>
</tr>
</tbody>
</table>

**Query mix for SF300**

<table>
<thead>
<tr>
<th>Query</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>26</td>
</tr>
<tr>
<td>Q2</td>
<td>37</td>
</tr>
<tr>
<td>Q3</td>
<td>142</td>
</tr>
<tr>
<td>Q4</td>
<td>46</td>
</tr>
<tr>
<td>Q5</td>
<td>84</td>
</tr>
<tr>
<td>Q6</td>
<td>580</td>
</tr>
<tr>
<td>Q7</td>
<td>32</td>
</tr>
<tr>
<td>Q8</td>
<td>3</td>
</tr>
<tr>
<td>Q9</td>
<td>705</td>
</tr>
<tr>
<td>Q10</td>
<td>44</td>
</tr>
<tr>
<td>Q11</td>
<td>24</td>
</tr>
<tr>
<td>Q12</td>
<td>44</td>
</tr>
<tr>
<td>Q13</td>
<td>19</td>
</tr>
<tr>
<td>Q14</td>
<td>49</td>
</tr>
</tbody>
</table>
LDBC Workload Driver

- **Short Reads**
  - Split into two groups: "person centric" & "message centric"
  - after each Complex Read/Update, a sequence of Short Reads is executed
    - a sequence approximates walk through network
    - at each step there is a probability of taking another step, which decreases at each step
    - steps consist of either all "person centric" or all "message centric" operations
      - e.g., (person centric operations)->(flip coin)->(message centric operations)->(flip coin)...
    - mimics user "following links"/Facebook-stalking :-)
  - substitution parameters taken from results of recent Complex Reads and Short reads
LDBC Workload Driver - Execution

• Driver schedules operations as close to their scheduled start times as possible
  - Experiments show the driver can achieve rates of hundreds of operations per second
• "Time Compression Ratio" used to configure target throughput
• Number of worker threads configurable
• Given a vendor implementation & workload, driver generates validation datasets
• Official validation datasets are provided by the LDBC SNB
  - https://github.com/ldbc/ldbc_snb_interactive_validation
LDBC Workload Driver - Rules

• Benchmark executions must meet the following rules to be valid:
  - queries must pass validation datasets
  - at most 5% of the queries actual start time can be one second greater than scheduled start time
  - must comprise at least 2 hours of simulation time
  - at any point, the test machine is disconnected and those committed must be persistent

• Performance metrics are:
  - latencies for each query
  - throughput
  - throughput/cost
  - a global benchmark score including loading time
Conclusions

- SNB Interactive on top of synthetic Social Network data
- 3 Types of queries:
  - Complex Reads
  - Short Reads
  - Updates
- The driver builds a query which mimics a user behavior
- Both latency and throughput are important. Persistence is mandatory
- All software is open source. We are open for contributions!
Progress

- Mainly focused on polishing and easing adoption
Progress

- Mainly focused on polishing and easing adoption
  - We have set up an Amazon S3 bucket with datasets
    - ldbc-snb
    - East coast region
  - From SF1 to SF1000 in CSV, CSVMergeForeign and TTL Formats
  - Set up as “Requester Pays”
    - cheap, about 0.03$ per GB
    - Datasets are compressed (about 1/3 ratio)
    - Downloading SF1000 its about 10$
Progress

- Mainly focused on polishing and easing adoption
  - We have created a Postgres compliant JDBC driver with all Interactive and BI query implementations.
    - [https://github.com/ldbc/ldbc_snb_implementations](https://github.com/ldbc/ldbc_snb_implementations)
  - Fully validated
  - The goal is to serve as the base implementation for SQL systems
Progress

• Mainly focused on polishing and easing adoption
  – Extended the LDBC driver with new requested features from vendors
    • Adjustable number of update threads
    • Skipable update stream starting point
Progress

• Mainly focused on polishing and easing adoption
  - Improved query formulation, consistent with BI queries
    • Added “limit” and “sort” sections
  - Removed unnecessary stuff that was outdated or duplicated from the github pages (from 106 to 39 pages)
Progress

- Mainly focused on polishing and easing adoption
  - New version of the data generator (v0.2.6) with new features:
    - Added data integrity tests
    - Improved performance and scalability
    - Added more configuration options to override the generation process:
      - Custom string/date formatting
      - Custom message text generation
      - Custom knows edge weight computation
    - Bug fixing (thanks to testing!)
  - See github.com/ldbc/ldbc_snb_datagen/releases/tag/v0.2.6 for a full list of changes
Current and future Work

- Towards 1.0 version
  - Missing the pricing cost model.
    - Waiting for LDBC Bylaws to be approved
  - Preparing new audited results
    - Neo4j and Sparksee are ready to be audited
Current and future Work

- Working on a reporting tool to visualize the data output by the driver
  - Just prototyping stages
Conclusions

- LDBC SNB Interactive Workload models the use of a social network site by its users
  - Complex Reads, Short Reads, Updates
- Targets systems at different scales and kinds
- Actively hearing the community, please send Feedback!
  - We are mainly working on easing the adoption
- Preparing version 1.0 with new audited results, to be sent to the Board of Directors for approval
Thank you