**Semantic Publishing Benchmark**

Full Disclosure Report

Report template date: 3 Nov 2014

***<UUID of the Test Instance>***

*<Date>*

Table of Contents

[Preface 4](#_Toc416271968)

[1 General Terms 5](#_Toc416271969)

[2 Clause 1: Dataset and Bulk Load Description 6](#_Toc416271970)

[2.1 Dataset Description 6](#_Toc416271971)

[2.2 Bulk Load Description 6](#_Toc416271972)

[3 Clause 2: RDF Database Description 7](#_Toc416271973)

[3.1 Vendor Name 7](#_Toc416271974)

[3.2 RDF Database Name 7](#_Toc416271975)

[3.3 Version Number 7](#_Toc416271976)

[3.4 Database Engine Configuration 7](#_Toc416271977)

[4 Clause 3: Platform Description 8](#_Toc416271978)

[4.1 Operating System and File System 8](#_Toc416271979)

[4.2 CPU Type and Count 8](#_Toc416271980)

[4.3 Number of Threads 8](#_Toc416271981)

[4.4 Number of Cores 8](#_Toc416271982)

[4.5 Memory 8](#_Toc416271983)

[4.6 Number of Disks, Type of Disks, Storage Configuration 8](#_Toc416271984)

[4.7 Total Disks capacity 8](#_Toc416271985)

[4.8 System Configuration 9](#_Toc416271986)

[5 Clause 4: Benchmark Test Driver Description 10](#_Toc416271987)

[5.1 Test Driver Configuration 10](#_Toc416271988)

[5.2 Test Driver Reference Data 10](#_Toc416271989)

[6 Clause 5: Performance Metrics and Execution Rules 11](#_Toc416271990)

[6.1 Scale Factors 11](#_Toc416271991)

[6.2 SPB Primary Metric 11](#_Toc416271992)

[6.3 Update Rate 11](#_Toc416271993)

[6.4 Interactive Query Metric 12](#_Toc416271994)

[6.5 Analytical Query Metric 12](#_Toc416271995)

[6.6 Measurement Window and Minimum Duration 12](#_Toc416271996)

[6.7 Query Execution Report 13](#_Toc416271997)

[7 Supporting File Index 14](#_Toc416271998)

***Abstract***

This document specifies what information from the LDBC Semantic Publishing Benchmark run will be gathered and described in it. Most of this information is produced by the benchmark tools themselves, some is produced by a test sponsor.

This document can be used as a template, where the parts in *italic font face* are replaced with the information about the actual test results being disclosed.

Preface

The Linked Data Benchmark Council (LDBC) is established as an independent authority responsible for specifying benchmarks, benchmarking procedures and verifying/publishing results for software systems designed to manage graph and RDF data. LDBC aims to make insightful the critical properties of graph and RDF data management technology that allows information technology practitioners to make informed technology choices and stimulate progress through competition.

**LDBC Semantic Publishing Benchmark Overview**

The *Semantic Publishing Benchmark* (SPB) is a LDBC benchmark for RDF database engines inspired by the Media/Publishing industry. It simulates the management and consumption of RDF metadata and is designed to reflect a scenario where a large number of aggregation agents provide the heavy query workload, while at the same time a steady stream of update operations are in progress. This benchmark targets RDF database systems, which support at least basic forms of semantic inference.

LDBC SPB evaluates the performance of RDF database systems by execution of sets of queries and updates under controlled conditions. Two types of concurrently running agents are performing a workload against the RDF database:

* aggregation agents - executing a mix of queries such as: search, aggregation, full-text-search, geo-spatial, analytical, faceted search;
* editorial agents - executing insert, update and delete operations;

All of the information is gathered into a single file per test execution accompanied by a set of supporting files also provided or made available for download.

Benchmark results are highly dependent on the systems design, implementation, workload and specific application requirements. Relative system performance will vary as a result of these factors, therefore LDBC-SPB should not be used as a substitute for a specific customer application benchmarking when critical capacity planning and product evaluation decisions are contemplated.

Further information is available at <http://ldbcouncil.org/benchmarks/spb>.

1. General Terms

**Test Sponsor**

*A statement identifying the benchmark sponsor and other participants must be provided.*

**Test Start Time**

*This is a date time for placing the test run in time, the specific start times of all components (test drivers) are specified.*

**SUT Name**

*System under test – a short name of the hardware platform, e.g. manufacturer plus model number.*

**RDF Database Name**

*Short name of the RDF Database, e.g. vendor, product name and version.*

**Flag Indicating Whether This is a Cluster Result**

*Provide a flag which clearly indicates that results produced by the benchmark are coming from a cluster configuration or a single database instance.*

**UUID of the Test Instance**

*A single unambiguous identifier for the test. It is recommended that the code represents a short description of the run including benchmark name, version, scale factor, database name and environment. Version of the benchmark test driver can be found in the manifest of the distribution file. For instance: SPB v0.1.e7ff928, SF1 – 50M, basic qmix, GraphDB v6.0, AWS c3.4xlarge*

1. Clause 1: Dataset and Bulk Load Description

The following items are recorded for description of the generated dataset and the bulk load of generated data.

* 1. Dataset Description
* Scale Factor

*Specify the scale factor used, e.g. SF1. See section 6.1 for details.*

* Data Format

*Specify data serialization format, e.g. N-quads, TriG etc. Note that data serialization format should be context aware, i.e. data input files should provide information about the Named Graph or the Context of each statement; this is the fourth element of a statement <s,p,o,c>.*

* Data generator version

*Specify the version of the Benchmark Test driver. Version of the Benchmark Test Driver is contained in the manifest of the distribution file.*

* Date Time Start of data generation
* Count of generated Creative Works and the exact number of explicit statements produced

*Details about those entities in generated data is produced by the Data Generator.*

*Listings must be provided of all ontologies and reference datasets used. By default all reference datasets and ontologies are included with the benchmark distribution.*

* 1. Bulk Load Description
* Start of bulk load

*Provide the start date and time of the bulk load.*

* End of bulk load

*Provide the end date and time of the bulk load.*

1. Clause 2: RDF Database Description

The items described below may be extracted with appropriate system information calls to the SPARQL end-point or other means.

* 1. Vendor Name

*The name of the RDF database vendor.*

* 1. RDF Database Name

*The name of the RDF database being tested.*

* 1. Version Number

*The full version and revision numbers of the RDF Database.*

* 1. Database Engine Configuration

*Full disclosure about the configuration of the engine to make possible reproduction of the test results. It should include at least the following parameters, if those are relevant for the engine:*

* *Cache configuration (usually multiple parameters);*
* *Transaction isolation level and or model;*
* *Reasoning strategy and configuration, e.g. “backward-chaining RDFS” or “materialization, OWL 2 RL” or “hybrid, RDFS--";*
* *What triple/quadruple indices are switched on/maintained, e.g. POS, PSO, CPSO, etc.;*
* *What additional indices are used, e.g. FTS, geospatial, etc.;*
* *What other configurable optimizations are used, e.g. owl:sameAs optimization.*

*.*

1. Clause 3: Platform Description

The following lists the features of the platform used to run the LDBC benchmark. All examples given below are suitable for execution on a Linux system, but the corresponding metrics may be extracted from most other systems.

* 1. Operating System and File System

*Provide the result of: "uname -a" command for Linux systems or appropriate one for others.*

*Similar system information for the file system should be provided, e.g. ZFS or EXT4.*

*Provide the result of: "df --print-type"*

* 1. CPU Type and Count

*Provide heading of the CPU Description in /proc/cpuinfo*

* 1. Number of Threads

*The count of CPU description entries in /proc/cpuinfo. In case of Intel Xeon processors with hyperthreading, this figure is double the number of the physical cores.*

* 1. Number of Cores

*The siblings count in a CPU description in /proc/spuinfo. In case of cloud infrastructure the number of the virtual CPUs should be provided; for instance vCPUs for AWS.*

* 1. Memory

*The total amount of physical RAM in MBytes.*

* 1. Number of Disks, Type of Disks, Storage Configuration

*The number of distinct /dev/sd?\*, count distinct letters in ?.*

*Different storage interfaces should be disclosed, e.g. SATA, SAS, etc.*

*Different storage technologies should be disclosed, e.g. HDD, SDD, hybrid.*

*If RAID or HBA controller is used, its name should be disclosed along with the specific configuration, e.g. RAID 0, 1, 11, 5, 6.*

* 1. Total Disks capacity

*The sum of total space from "df" command, excluding nfs mounted file systems and RAM based file systems, e.g. tmpfs.*

* 1. System Configuration

*Contents of system configuration file /etc/sysconf*

In case of a cluster configuration, this description is repeated for each of the cluster nodes. These will typically be identical but may not always be so.

1. Clause 4: Benchmark Test Driver Description
   1. Test Driver Configuration

*Provide basic test driver configuration details:*

* *total number of read threads (aggregation agents);*
* *total number of write threads (editorial agents).*

*All of the benchmark's test driver configuration parameters are contained in two files:*

* *test.properties - defines the general behavior of the benchmark test driver, number of read and write threads, reference datasets, etc.*
* *definitions.properties - defines various allocations related to data generation, execution of query mixes, benchmark behavior, etc.*

*Listed configuration files above must be provided within the supporting files set for each of the benchmark test drivers.*

* 1. Test Driver Reference Data

*Reference data, ontologies and query templates are located at the 'data/' directory of the SPB benchmark's software distribution. A list of the reference datasets and ontologies used in the test must be disclosed. Any modification to files e.g. query templates, reference datasets or ontologies must be disclosed and provided within the supporting files set.*

1. Clause 5: Performance Metrics and Execution Rules
   1. Scale Factors

The unit scale of SPB is 64M statements (the unit scale is denoted as SF1). Benchmark results may be reported at any scale greater than this. It is recommended to use scale factors that double in size the previous one, e.g. SF2 - 128 million statements, SF3 - 256 million statements etc.

It is important to note that, when generating synthetic data for a certain scale factor, the number of generated statements should be calculated by subtracting the number of explicit statements in reference data which currently is about 25M statements. e.g. thus for SF1 generated synthetic data will be 64M - 25M (ref. data) = 39M statements.

* 1. SPB Primary Metric

SPB reports the following for a run of the complete workload and its primary metric is a composite of the four following quantities:

* Query rate for the interactive mix in queries per second *(iq/s);*
* Query rate for the analytical mix in queries per hour, multiplied by the scale factor *(aq/h\*SF);*
* Update rate operations per second;
* Scale Factor (SF).

The metric is of the form *iq/s, aq/h, u/s at scale* e.g.: 12.2 iq/s, 17.2 aq/h, 70 u/s at 200 million triples (SF 3).

The metrics are all measured during one execution of the benchmark. The test sponsor may configure the test drivers to obtain different balances of the components of the metric by adjusting the update rate and the number of simulated users (test driver threads) driving each type of operation. An increase in one component of the metric will generally lead to a decrease in another component.

* 1. Update Rate

A qualifying SPB execution must sustain a minimum number of updates per unit of time, proportional to the scale factor. A compliant run may report this rate or any update rate greater than this. The update driver is configured to maintain a given update frequency. A run is disqualified if the number of updates actually completed at the end of the benchmark execution is less than 95% of the target number of updates per second times the duration of the measurement interval.

The minimum update rate is 7 updates per second at the unit scale (SF1). The minimum number of updates per second at other scales is given by *7 \* log (SF) \* log (1.1)*, where *log* means the natural logarithm.

The update concurrency is at the discretion of the test sponsor, thus a throughput of 7 updates per second may be obtained with a single test driver configured to do 7 per second or two test drivers each configured to do 3.5 per second, for example. In any case, update execution is interleaved with query execution and updates are to be submitted at the specified rate throughout the execution.

Due to the specifics of the simulated application scenario **updates should not be batched**. This is related to the fact that media businesses are extremely sensitive with respect to timely and consistent access to up to date databases. The benchmark test driver performs validations to detect if batching is used and will report errors when this is detected. Test results from runs that involve batching of updates should be considered invalid.

* 1. Interactive Query Metric

The test sponsor influences the interactive queries per second metric by starting a variable number of threads submitting queries from the interactive query mix. Each such thread runs at the maximum throughput of the SUT, so that the next query is immediately submitted after a test driver thread receives the result of the previous result.

The test sponsor may choose which one of the two interactive query mixes to use for the benchmark run: the one distributed with basic or the one distributed with the advanced version of the benchmark test driver. Chosen interactive query mix should be disclosed.

The metric is calculated by dividing the count of interactive query executions whose start and finish fall within the measurement window by the length of the window in seconds.

* 1. Analytical Query Metric

The analytical query metric is the number of query executions from the analytical query mix divided by the length of the measurement window in hours and multiplied by the scale factor. Rationale: The interactive query times are generally in a logarithmic relation to scale factor, hence larger scale does not significantly affect the query duration. Analytical queries generally traverse a large, fraction of the database and are generally between *O(n)* and *O(n\*log(n))* in complexity. Hence processing double the data takes roughly double the time. In order for the metric not to drop as scale increases, the reported number is multiplied by the scale factor. This makes results at different scales easier to compare.

The test sponsor may choose not to measure the performance of the analytical component - that should be stated by adding a "N/A" string next to the analytical component metric.

* 1. Measurement Window and Minimum Duration

A SPB measurement window consist of an integer number of completed analytical query mixes. Concurrently with these, the update throughput target for the scale factor must be sustained and there must be at least one thread continuously driving the interactive workload at the maximum speed of the SUT (no think times between queries).

The analytical workload may be driven by one or more threads, each submitting the next query immediately following the completion of the previous one. The measurement window starts at the start of the first query of the first analytical mix and finishes at the completion of the last query of the last analytical mix.

The test sponsor may set the number of analytical mixes to be performed during the window. The window must not contain partial executions of the analytical mix. For example, if the test sponsor runs three analytical mixes on concurrent threads, these may finish at different times but the threads which finish first will not start a second query mix. The window finishes at the completion of the slowest thread driving the analytical mix.

The SUT may be warmed up before the start of the measurement window by running the interactive query workload (no updates and no analytical queries) for a test sponsor determined amount of time.

The minimum duration of the measurement window depends on the scale factor. For each scale, there is a minimum number of analytical mixes and a minimum duration in real time. If the prescribed number of analytical mixes finishes before the minimum duration, the test sponsor must add more analytical mixes in order to run for the minimum amount of real time. In any case, the window always contains an integer number of completed analytical mixes.

The minimum number of analytical mixes is given by the formula:

*1+ floor(log (SF) / log (1.5))*

The minimum duration of the measurement window is 10 minutes of real time per each required analytical mix. If analytical mixes complete faster than this, then more mixes have to be added to meet the minimum duration criterion.

If no analytical query mixes are to be executed during the benchmark run, then the minimum time duration of measurement window is defined by same rules as described in this section.

* 1. Query Execution Report

In addition to the primary metric of *iq/s, aq/h \* SF, u/s at SF* the numerical quantities summary contains the following data:

* Duration of bulk load (HH:MM:SS);
* Duration of measurement window (HH:MM:SS);
* Number of complete analytical mixes performed;
* Number of complete interactive mixes performed;
* Number of complete update operations performed.

**Note:** the measurement window may stop in mid update or mid interactive query mix. Partial updates and interactive query mixes are not counted.

For each operation, i.e. query of interactive, analytical mix or update operation, the following numbers are provided:

* Arithmetic mean of execution time;
* Minimum execution time;
* Maximum execution time;
* 90th percentile execution time;
* Count of executions.

The execution times are in milliseconds measured at the test driver. The time starts before sending the request and finishes after the complete response from the SUT is available to the application. The times include any network latencies and client side times for parsing the reply from the SUT.

1. Supporting File Index

The following table describes a list of supporting files that should be provided together with each test disclosure report. Supporting files can be made available for download. The table may be expanded with more items if necessary.

|  |  |  |
| --- | --- | --- |
| **Clause** | **Description** | **Archive file name** |
| Clause 1 | RDF Database load scripts if any | Archive1.zip |
| Clause 2 | RDF Database configuration if more details are available | Archive2.zip |
| Clause 3 | Operating System and Hardware settings | Archive3.zip |
| Clause 4 | Benchmark test driver configuration files and any modified query templates and (or) ontologies / reference datasets | Archive4.zip |
| Clause 5 | All of the log files generated by the benchmark test driver | Archive5.zip |