

2ndQuadrant[®] 
PostgreSQL

Building Better Benchmarks

PGCon 2020



About me ...

- Employed by 2ndQuadrant
- PostgreSQL Contributor since 2005
- Director at United States PostgreSQL Association since 2011
- Portland PostgreSQL Users Group



Overview

- What is benchmarking about.
- What benchmarks are out there?
- Issues with benchmarking.
- Where do we want to go now?



Reasons for benchmarking

- Competition
 - Performance
 - Costs
- Self-assessment
 - How well am I utilizing the hardware?
 - How does this patch affect the system?



Competitive Benchmarking

Taking a look at just a of couple of industry standards bodies...



Benchmark Publications

Only a couple of published Postgres benchmarks:

- Two SPECjAppServer2004 results in 2007

<https://www.spec.org/jAppServer2004/results/res2007q3/>

- No TPC publications for PostgreSQL



SPEC

SPEC is a non-profit organization that establishes, maintains and endorses standardized benchmarks and tools to evaluate performance and energy efficiency for the newest generation of computing systems.

- develops suites of benchmarks intended to measure computer performance
- benchmarks suites are available to the public for a fee covering development and administrative costs
- publishes news and benchmark results at <http://www.spec.org/>



TPC

Transaction Processing Performance Council performs two major activity:

1. creating benchmarks
2. creating a process for reviewing and monitoring those benchmarks

<http://www.tpc.org/>



TPC Benchmarks

Many TPC benchmarks are not trivial:

- Complete benchmarking kits generally not provided
 - Database design
 - Data generation
 - Transaction profiles
- Auditing
- Publication



Summarizing Competitive Benchmarking

- Well-defined workloads
- Expensive
- Non-trivial effort to implement
- Non-trivial effort to execute
- Not intended for individuals



Needs for Self-assessment

Important data in addition to benchmark metrics:

- System statistics:
 - processor utilization
 - i/o throughput
 - etc.
- Software profiling:
 - annotated source code
 - call graphs
 - flame charts
 - etc.



OLTPBench

In many cases, researchers and developers are limited to a small number of workloads to evaluate the performance characteristics of their work. This is due to the lack of a universal benchmarking infrastructure, and to the difficulty of gaining access to real data and workloads. This results in lots of unnecessary engineering efforts and makes the performance evaluation results difficult to compare.

<http://www.vldb.org/pvldb/vol17/p277-difallah.pdf>



Adapting benchmarks

At least 12 open source projects already exist:

- <http://oltpbenchmark.com/>
- https://github.com/tvondra/pg_tpch
- <https://github.com/Percona-Lab/sysbench-tpcc>
- <https://sourceforge.net/projects/benchmarksql/>
- <http://jtpcc.sourceforge.net/>
- <https://www2.infor.uva.es/~diego/tpcc-uva.php>
- <https://github.com/pivotalguru/TPC-DS>
- <https://oss.oracle.com/projects/olt/>
- <https://www.hammerdb.com/>
- <https://github.com/2ndQuadrant/pg-tpch>
- <http://osldbt.sourceforge.net/>
- <https://github.com/jopereira/java-tpcw>



Potential Licensing Issues

While subject to change, EULA may not be agreeable. Excerpt from a previous TPC EULA:

4. Restrictions. The following restrictions apply to all use of the Materials by You.
 - a. General: You may not:
 - ...
 - iii. submit or contribute the Materials, or any part thereof, to any open source software project or as part of any third-party software or project without the express written consent of the TPC chair.

http://www.tpc.org/tpc_documents_current_versions/current_specifications5.asp



Quick Mention of Other Open Source Database Workloads

Want to discuss more in unconference session?

- Yahoo Cloud Serving Benchmark - key-value store
<https://ycsb.site>
- LDBC - graph databases <http://ldbncouncil.org/>



Summarizing Needs for Self-Assessment

Benchmarking kits are more helpful if they:

- Collect system statistics
- Profile the system
- Unencumbered licensing



Where should we go from here?



Develop Our Own Database Workloads

Focusing on just two generate types of workload:

- Online transaction processing (OLTP)
 - Multi-tier client-server
 - Electronic data processing, wholesaler supplier managing orders, brokerage firm executing customer transactions
- Business Intelligence (BI)
 - Single system
 - Business oriented ad-hoc queries, big data systems star-schema



Summary of Workload Architectures

- OLTP - Complex:
 - multi-tier: develop driver, develop or use existing connection pooler
 - Scalability of driver
 - Server vs client side transaction logic
- BI - Simple:
 - Single-tier
 - Scripts to execute queries and streams of queries.
 - Maybe we can outline a test in an unconference session?



Is there a current need for competitive benchmarking?

- Lots to consider:
 - Fair playing field
 - Enforcing fair play
 - etc.
- Further discussion in an unconference session?



A little fun: Compare a Java vs. C implementation of a TPC-C derived workload.



My hardware

HP ZBook Studio G5 Mobile Workstation:

- Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (6 cores / 12 threads)
- 64 GB RAM
- HP EX950 M.2 NVMe 2TB
- Any hardware donations available?



Test Parameters

OLTPBench vs DBT-2:

- Scale factor: 1 warehouse
- 10 terminals each with a database connection
- No keying or thinking time
- 6 Cores (SMT disabled)



Results

OLTPBench:

- Metric 21,580.49
new-order TPM
- Processor cores 80% -
90% utilized
- driver - 1 java process
100% processor utilization
- i/o 99% utilized
- 10 postgres backends 30%
processor utilization each

DBT-2:

- Metric 28,838.20
new-order TPM
- Processor cores 60%
utilized
- driver - 76% processor
utilization
- i/o 97% utilized
- 10 postgres backends 30%
processor utilization each



Thank you!

Mark Wong
Consultant, 2ndQuadrant
Contributor, PostgreSQL
mark@2ndQuadrant.com
<https://2ndquadrant.com/contact>