Towards DBCloudBench
A Scenario-Based Database Benchmarking Framework

Andreas Bader and Oliver Kopp

University of Stuttgart, IPVS, Germany

Abstract. Many benchmarks can be used for measuring performance of different types of databases. To automate the process of benchmarking databases, this paper outlines DBCloudBench. It can be used to automatically setup a scenario and perform a benchmark run using a standards-based approach. The databases and benchmarks are stored in "cloud services archives" allowing them to be reused and combined as necessary. Each benchmark is accompanied with an adapter for running the benchmark on certain database systems while using DBCloudBench.

1 Overview

Choosing a database in a project may be done in different ways such as comparing performance [3], comparing features [1], based on existing knowledge, or by architectural decisions [17]. A performance comparison of different databases is a crucial step when choosing a database. Performance can be compared with different metrics, e. g., completed transaction per time unit, or latency of transactions. Since most databases are grouped in overlapping groups (e. g., time series databases, NoSQL databases, or relational databases), many benchmarks are made for one or more groups of databases or for a specific purpose (e. g., storing time series data). This means that a benchmark can only be executed if the database supports all features that it requires. Beside their targeted database groups, benchmarks are usually different in the creation of their queries (e. g., synthetic creation) and in the measured metrics. As a result, there exist multiple benchmarks for the same group of databases (e. g., MySQL can be benchmarked with TPC-H [15] and YCSB [3]).

When benchmarking performance, there are two possible approaches: 1) Attest and achieve a maximum performance by fine tuning the database and (optionally) the underlying system, 2) Getting insights on performance for choosing a database by benchmarking a set of databases with the same benchmark.

If using the second approach for choosing a database, one or more benchmarks are used to measure one or more metrics of several databases setup in one or more scenarios. That means that a user performing the benchmarking has to know how to setup multiple databases, multiple benchmarks, their metrics, and their required parameters. To chose a database, multiple scenarios may be required to be measured, e. g., different cluster sizes or different replication factors. Additionally, a user has to perform each benchmark run manually, which means that each setup has to done manually for each database, scenario, and benchmark. To ensure good results, the conditions for each run must be comparable [11].

The usage of cloud techniques (e. g., using an Elastic Infrastructure, EI for short) makes it possible to automate the process of performing a benchmark run, which includes setup, benchmarking, retrieving results, and cleanup. Therefore, this work focuses on the second approach, as fine tuning of a system cannot be done in an automated way yet, which makes the first approach impossible to automate. The proposed solution is a benchmark framework for databases being independent of databases and benchmarks that can be used to automatically perform benchmark runs that are setup according to a scenario definition on an EI: DBCloudBench.

A requirement for the benchmarking framework is to support as many EIs as possible and to be as independent from EI-specific code as possible. Additionally, it should be easy to use and to extent for a user. Therefore, a external solution is required that interacts with the EI and keeps DBCloudBench free from EI-specific code. These requirements were derived during the creation of a platform for a new marked role, called Decentralized Market Agent (DMA) [13], which was done in the context of the NEMAR project [9].

By sharing artifacts on how to setup a scenario using repositories, a user requires less knowledge to setup a benchmark run and execute it. A user can pick the required artifacts from the repositories, choose the components he requires for his scenario, and execute it automatically in an EI. This results in less required domain-knowledge, as a user must only know which components he needs, without deeper knowledge in the setup of the chosen database and benchmark. In other words, the aim is that a user that wants to chose a database only specifies the database, benchmark, and scenario he wants to use and the framework executes it automatically. One main part of the solution is to use the “cloud services archive” packaging format of TOSCA and the OpenTOSCA ecosystem [2] for the installation of different benchmarks and databases.

2 Related Work

There exist several benchmarks for measuring cloud performance, the most prominent are CloudBench [12] and CloudCMP [10]. Silva et al. [12] provide an overview over the rest of these types of benchmarks. All of these benchmarks have in common that they try to provide an answer to the question which EIs to use. DBCloudBench, however, has the focus to provide a general framework to meassure database performance.

HammerDB is a tool for automated benchmarking of databases, with a focus on relational databases [4]. As it does not support the automated deployment of databases and its focus is on relational databases, it cannot be used for our approach.

The Transaction Processing Performance Council (TPC) benchmarks consist of several benchmarks for measuring the performance of different business scenarios, e. g., TPC-C [14] that uses an Online Transaction Processing (OLTP) workload that emulates a wholesale company or TPC-H [15] that emulates a decision support scenario, which uses complex queries that are designed to have a long execution time. Most comparable to our approach is TPC-VMS [16], which benchmarks the performance of a database whilst running on Virtual Machines (VMs). To do this, it takes one of the other TPC benchmarks and runs it three times parallel on three identical VMs on an EI. TPC-VMS does not cover the automated deployment of the VMs used in the benchmark, as they are already setup before the benchmark starts.

Kolb and Wirtz [8] identified a similar scenario for users of Platform as a Service (PaaS). These cloud systems often come with heterogeneous environments and interfaces to work with. To assist a user in the decision for the right cloud platform and automate the deployment of user applications, they propose a unified feature description and interface for cloud platforms [7].

The BPEL/BPMN Engine Test System (betsy), a benchmark framework for testing the compliance of open-source BPEL/BPMN engines to the corresponding standards [5], was extended to use virtualization techniques (vbetsy) [6], which results in the use of VMs and their snapshot functionalities to perform benchmark runs. The idea is similar to our approach, but it lacks the support of creating a cluster of VMs depending on a scenario chosen by a user on an EI, which makes it not usable for our approach, even though it uses multiple benchmarks and test candidates (engines) for its tests.

Acknowledgments This work was created in the context of the BMWi-funded project NEMAR (03ET4018).

References

1. Bader, A., Kopp, O., Falkenthal, M.: Survey and Comparison of Open Source Time Series Databases. In: Datenbanksysteme für Business, Technologie und Web (BTW2017) – Workshopband. LNI, Gesellschaft für Informatik e.V. (GI) (2017)
2. Binz, T., Breitenbücher, U., Kopp, O., Leymann, F.: TOSCA: Portable Automated Deployment and Management of Cloud Applications, pp. 527–549. Advanced Web Services, Springer (Jan 2014)
3. Cooper, B.F., Silberstein, A., Tam, E., Ramakrishnan, R., Sears, R.: Benchmarking Cloud Serving Systems with YCSB. In: Proceedings of the 1st ACM Symposium on Cloud Computing. ACM (2010)
4. HammerDB project team: HammerDB (2017), http://www.hammerdb.com
5. Harrer, S., Lenhard, J.: Betsy – A BPEL Engine Test System. Tech. Rep. 90, Otto-Friedrich Universität Bamberg (Jul 2012)
6. Harrer, S., Röck, C., Wirtz, G.: Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines. In: 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops. IEEE (mar 2014)
7. Kolb, S., Röck, C.: Unified Cloud Application Management. In: Proc. World Congress on Services. IEEE (2016)
8. Kolb, S., Wirtz, G.: Towards Application Portability in Platform as a Service. In: Proc. 8th Symp. Service-Oriented System Engineering. IEEE (2014)

9. Kopp, O., Falkenthal, M., Hartmann, N., Leymann, F., Schwarz, H., Thomsen, J.: Towards a Cloud-based Platform Architecture for a Decentralized Market Agent. In: Informatik 2015. LNI, Gesellschaft für Informatik e.V. (GI) (2015)
10. Li, A., Yang, X., Kandula, S., Zhang, M.: CloudCmp: Comparing Public Cloud Providers. In: Proceedings of the 10$^{\text{th}}$ annual conference on Internet measurement. ACM (2010)
11. Lilja, D.J.: Measuring Computer Performance: A Practitioner's Guide. Cambridge University Press (2005)
12. Silva, M., Hines, M.R., Gallo, D., Liu, Q., Ryu, K.D., da Silva, D.: CloudBench: Experiment Automation for Cloud Environments. In: 2013 IEEE International Conference on Cloud Engineering (IC2E). IEEE (2013)
13. Thomsen, J., Hartmann, N., Klumpp, F., Erge, T., Falkenthal, M., Kopp, O., Leymann, F., Stando, S., Turek, N., Schlenzig, C., Schwarz, H.: Darstellung des Konzeptes – DMA Decentralised Market Agent – zur Bewältigung zukünftiger Herausforderungen in Verteilnetzen. In: INFORMATIK 2015. LNI, vol. P-246. Gesellschaft für Informatik e.V. (GI) (2015)
14. Transaction Processing Performance Council (TPC): TPC-C Benchmark (2017), http://www.tpc.org/TPC_Documents_Current_Versions/pdf/TPC-C_V5-11.pdf
15. Transaction Processing Performance Council (TPC): TPC-H Benchmark (2017), http://www.tpc.org/TPC_Documents_Current_Versions/pdf/tpch2.17.1.pdf
16. Transaction Processing Performance Council (TPC): TPC-VMS Benchmark (2017), http://www.tpc.org/TPC_Documents_Current_Versions/pdf/TPC-VMS-V1.2.0.pdf
17. Zimmermann, O., Wegmann, L., Koziolek, H., Goldschmidt, T.: Architectural Decision Guidance Across Projects – Problem Space Modeling, Decision Backlog Management and Cloud Computing Knowledge. In: 12$^{\text{th}}$ Working IEEE/IFIP Conference on Software Architecture. IEEE (2015)