Towards a Framework for Orchestrated Distributed Database Evaluation in the Cloud

Daniel Seybold
Institute of Information Resource Management
Ulm University
Ulm, Germany
daniel.seybold@uni-ulm.de

Abstract

The selection and operation of a distributed database management system (DDBMS) in the cloud is a challenging task as supportive evaluation frameworks miss orchestrated evaluation scenarios, hindering comparable and reproducible evaluations for heterogeneous cloud resources. We propose a novel evaluation approach that supports orchestrated evaluation scenarios for scalability, elasticity and availability by exploiting cloud resources. We highlight the challenges in evaluating DDBMSs in the cloud and introduce a cloud-centric framework for orchestrated DDBMS evaluation, enabling reproducible evaluations and significant rating indices.

CCS Concepts • **Information systems → Database performance evaluation**;

Keywords benchmarking, distributed database, cloud, NoSQL, NewSQL, orchestration

1 Introduction

In the last decade, database management systems (DBMSs) have evolved by focusing as well on distributed database management systems (DDBMSs) as evolving application domains such as the Web or Big Data impose new challenges to Online Transaction Processing (OLTP) [13]. Thus, a plethora of new DDBMSs have appeared on the DBMS landscape, which can be classified into NoSQL and NewSQL. These DDBMSs are built on a shared-nothing architecture and promise to cater for non-functional requirements such as scalability, elasticity, availability by running on commodity hardware or even on cloud resources. As cloud computing offers on-demand resource provisioning, the cloud seems to be the preferable solution to operate DDBMSs.

Yet, with the vast number of available DDBMSs and the heterogeneous cloud resource offerings, the selection of a DDBMS and its operation in the cloud becomes a challenging task. Assuming a DDBMS is required to continuously store social media data based on actual events. The data is read from a varying amount of users and periodically queried by an analytics engine. Hence, the DDBMS needs to *scale horizontally* in case of growing workloads and provide *elasticity* to handle sudden workload peaks, created by social media events and the resulting user requests. Cloud resources are used to run the DDBMS, ensuring the dynamic resource allocation on demand. Yet, as cloud resources can fail, the DDBMS needs to provide *availability* in case of cloud resource failures. Evaluating these requirements of existing DDBMS is a common approach to guide the DDBMS selection process. Yet, current evaluation frameworks (EFs) do not explicitly consider the usage of heterogeneous cloud resources and lack the support for orchestrated evaluation scenarios [11] with respect to scalability, elasticity and availability.

2 Problem Statement

Our research targets the enhancement of the DDBMS selection by providing a cloud-centric EF for the orchestrated scalability, elasticity and availability evaluation. In the scope of our research, we highlight the following key challenges:

Cloud Resource Characteristics needs to be considered by the EF as cloud resources tend to become more heterogeneous, from virtual machines to container technologies, various storage technologies and resource locality options. Thus, our EF will be aware of these characteristics and enables the access to cloud resources in a unified way, easing comprehensive, significant and reproducible results.

Orchestrated Evaluation Scenarios enable the evaluation of elasticity and availability in a comparable and reproducible way, by adapting the DDBMS topology by managing cloud resources. Consequently, predefined evaluation scenarios can be applied to generic DDBMS and cloud resource templates (CRTs) that will be executed by the EF.

Workload Domains such as OLTP or Hybrid Transaction and Analytical Processing (HTAP), are required for realistic evaluation scenarios. Thus, our EF will support domain-specific workload creation, based on synthetic and trace-based workloads.

Rating Indices on a DDBMS basis need to be computed based on the raw evaluation results to ease the DDBMS selection. While for established features such as performance, rating indices are available, comparable rating indices as elasticity and availability still need to be defined.

3 Related Work

One of the main drivers of non-functional DBMS feature evaluation is the Transaction Processing Performance Council (TPC)[1], providing EFs for the OLTP domain. Yet, TPC rather focuses on the performance of relational DBMS than on DDBMSs features such as elasticity, availability or the usage of cloud resources. DDBMSs centric EFs such as the Yahoo Cloud Serving Benchmark (YCSB) [5] and its extensions YCSB++ [10] and YCSB+T [6] support the evaluation of scalability and consistency based on synthetic workloads. Advanced workload domains are addressed by BG [2], LinkBench [1] and OLTP-Bench [7], yet without the explicit consideration of cloud resources. Evaluating the scalability and elasticity of DDBMSs by using cloud resources is presented by [9]. Yet, the evaluation relies on synthetic workloads and does not consider multi-cloud scenarios. A first attempt to evaluate availability is presented by UPB [8] by measuring the performance impact in case of node failures, while advanced scenarios including cloud resource failures, DDBMS failover and recovery capabilities are not considered.

[1] http://www.tpc.org/default.asp

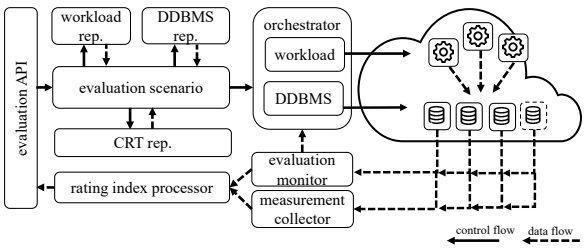

Figure 1. High-level evaluation framework architecture

While existing EFs rely on static evaluation scenarios, which either focus on advanced workloads for evaluating performance or using synthetic workloads without considering heterogeneous cloud resource configurations, we propose a novel EF enabling the orchestrated evaluation of scalability, elasticity and availability for DDBMSs based on heterogeneous cloud resources and multiple workload types.

4 Approach

Our initial DDBMSs evaluation addresses scalability and elasticity of DDBMSs in the cloud [12]. Our results show significant differences with respect to elasticity and the need for orchestrated DDBMS evaluation in order to provide adaptive and reproducible evaluation scenarios. Consequently, we analyze existing EFs with the focus on their evaluation scenarios and their consideration of cloud resources [11]. As existing EFs do not yet support orchestrated evaluation scenarios, elasticity and availability evaluation lacks dedicated support. In addition, the impact of heterogeneous cloud resource is not addressed by existing EFs.

Hence, we propose a novel EF, enabling orchestrated evaluation scenarios with the focus on scalability, elasticity and availability of DDBMSs in the cloud. Its architecture is depicted in Figure 1. The *evaluation API* enables the specification of *evaluation scenarios* for scalability, elasticity and availability. Each evaluation scenario comprises the properties workload (synthetic/OLTP/HTAP); a CRT defining providers, locations and resource dimensions; and a DDBMS template provided by the *DDBMS/CRT repository*. The DDBMS template exposes a unified set of non-functional configuration options to ensure comparable evaluation of different DDBMSs. Each evaluation scenario can specify adaptation actions for *scalability*, *elasticity* and *availability* for evaluating their correlation with cost-, performance- or locality-optimized CRTs. The execution of the specified evaluation scenario is enabled by the *orchestrator* component, which is realized by a cloud orchestration tool [3]. The orchestrator unifies the cloud resources access, provisions the required cloud resources and orchestrates the DDBMS, the workload and the DDBMS adaptations at run-time. During the evaluation, system and DDBMS specific monitoring data is collected and stored by the *evaluation monitor*. Based on this monitoring data, the orchestrator is able to adapt the DDBMS automatically, according to the specified adaption actions of the evaluation scenario. The evaluation-specific metrics are collected by the *measurement collector* and retrieved by the *rating index processor* to compute significant rating indices.

5 Evaluation

The evaluation will follow a two-dimensional approach. The first dimension will evaluate the framework's features against distinguished guidelines of DBMS evaluation [4]. The second dimension comprises industry-driven evaluation scenarios for >= 5 DDBMSs based on cost-, locality and performance-optimized CRTs. The resulting scalability, elasticity and availability insights are analyzed towards their significance in the DDBMS selection process.

6 Conclusion

The vast DDBMS landscape requires new evaluation framework concepts, considering heterogeneous cloud resources and the orchestration of the evaluation, including the workload generation and the DDBMS deployment and adaptation. Thus, we propose a novel evaluation framework for the orchestrated DDBMS evaluation of scalability, elasticity and availability by explicitly addressing the usage of cloud resources.

7 Acknowledgements

This work is done under the supervision of Prof. Dr.-Ing. Stefan Wesner. The author would like to thank Dr. Jörg Domaschka for his support. The research leading to these results has received funding from the EC's Framework Programme HORIZON 2020 under grant agreement number 644690 (CloudSocket) and 731664 (MELODIC).

References

[1] Timothy G Armstrong, Vamsi Ponnekanti, Dhruba Borthakur, and Mark Callaghan. 2013. LinkBench: a database benchmark based on the Facebook social graph. In *ACM SIGMOD*.
[2] Sumita Barahmand and Shahram Ghandeharizadeh. 2013. BG: A Benchmark to Evaluate Interactive Social Networking Actions.. In *CIDR*.
[3] Daniel Baur, Daniel Seybold, Frank Griesinger, Athanasios Tsitsipas, Christopher B Hauser, and Jörg Domaschka. 2015. Cloud Orchestration Features: Are Tools Fit for Purpose?. In *IEEE/ACM UCC*.
[4] David Bermbach, Jörn Kuhlenkamp, Akon Dey, Sherif Sakr, and Raghunath Nambiar. 2014. Towards an extensible middleware for database benchmarking. In *TPCTC*.
[5] Brian F Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. 2010. Benchmarking cloud serving systems with YCSB. In *ACM SoCC*.
[6] Akon Dey, Alan Fekete, Raghunath Nambiar, and Uwe Rohm. 2014. YCSB+T: Benchmarking web-scale transactional databases. In *IEEE ICDEW*.
[7] Djellel Eddine Difallah, Andrew Pavlo, Carlo Curino, and Philippe Cudre-Mauroux. 2013. Oltp-bench: An extensible testbed for benchmarking relational databases. *VLDB Endowment* (2013).
[8] Alessandro Gustavo Fior, Jorge Augusto Meira, Eduardo Cunha de Almeida, Ricardo Gonçalves Coelho, Marcos Didonet Del Fabro, and Yves Le Traon. 2013. Under pressure benchmark for ddbms availability. *JIDM* (2013).
[9] Jörn Kuhlenkamp, Markus Klems, and Oliver Röss. 2014. Benchmarking scalability and elasticity of distributed database systems. *VLDB Endowment* (2014).
[10] Swapnil Patil, Milo Polte, Kai Ren, Wittawat Tantisiriroj, Lin Xiao, Julio López, Garth Gibson, Adam Fuchs, and Billie Rinaldi. 2011. YCSB++: benchmarking and performance debugging advanced features in scalable table stores. In *ACM SoCC*.
[11] Daniel Seybold and Jörg Domaschka. 2017. Is Distributed Database Evaluation Cloud-Ready?. In *ADBIS*.
[12] Daniel Seybold, Nicolas Wagner, Benjamin Erb, and Jörg Domaschka. 2016. Is elasticity of scalable databases a Myth?. In *IEEE Big Data*.
[13] Michael Stonebraker. 2012. Newsql: An alternative to nosql and old sql for new oltp apps. *Communications of the ACM. Retrieved* (2012).