Tim Kiefer, Wolfgang Lehner

Private Table Database Virtualization for DBaaS

Erstveröffentlichung in / First published in:

Fourth IEEE International Conference on Utility and Cloud Computing. Melbourne, 05.12 – 08.12.2011. IEEE, S. 328-329. ISBN 978-1-4577-2116-8

DOI: https://doi.org/10.1109/UCC.2011.52

Final edited form was published in "Fourth IEEE International Conference on Utility and Cloud Computing. Melbourne 2011", S.328-329, ISBN 978-1-4577-2116-8
https://doi.org/10.1109/UCC.2011.52

Private Table Database Virtualization for DBaaS

Tim Kiefer, Wolfgang Lehner
Dresden University of Technology
Database Technology Group
Dresden, Germany
{tim.kiefer, wolfgang.lehner}@tu-dresden.de

***Abstract*—Growing number of applications store data in relational databases. Moving database applications to the cloud faces challenges related to flexible and scalable management of data. The obvious strategy of hosting legacy database management systems (DMBSs) on virtualized cloud resources leads to suboptimal utilization and performance. However, the layered architecture inside the DBMS allows for virtualization and consolidation above the OS level which can lead to significantly better system utilization and application performance. Finding an optimal database cloud solution requires finding an assignment from virtual to physical resources as well as configurations for all components. Our goal is to provide a *virtualization advisor* that aids in setting up and operating a database cloud. By formulating analytic cost, workload, and resource models performance of cloud-hosted relational database services can be significantly improved.**

***Keywords*-Database-as-a-Service; DBaaS; Database; Cloud**

I. Introduction

Growing number of applications store their data in relational databases. A common policy is to provide a dedicated machine per database server, which in turn needs to be provisioned for peak load and may be underutilized often. Cloud computing leads to outsourcing many applications to service architectures (IaaS, SaaS). Motivations for moving to the cloud range from high flexibility and scalability to low costs and a pay-as-you-go pricing model. At the same time, from a provider's point of view, consolidation of applications on shared infrastructure leads to increased infrastructure utilization and lower operational costs. Applications that rely on relational databases can only be hosted in the cloud when flexible and scalable data management systems can be provided, too.

Relational Database Management Systems (RDBMSs) present unique challenges as regards to virtualization and consolidation. Virtual Machines (VMs) fail to adequately support database workloads and lead to suboptimal performance, as we will detail in Section II. To achieve good performance, a light-weight solution for database virtualization needs to be chosen and implemented.

Once decided in favor of a virtualization technique, the following challenges remain: (1) assigning logical resources to physical ones; (2) configuring each physical system (e.g., database design, tuning parameters, ...); (3) finding stable solutions that require no re-assignments in the near future

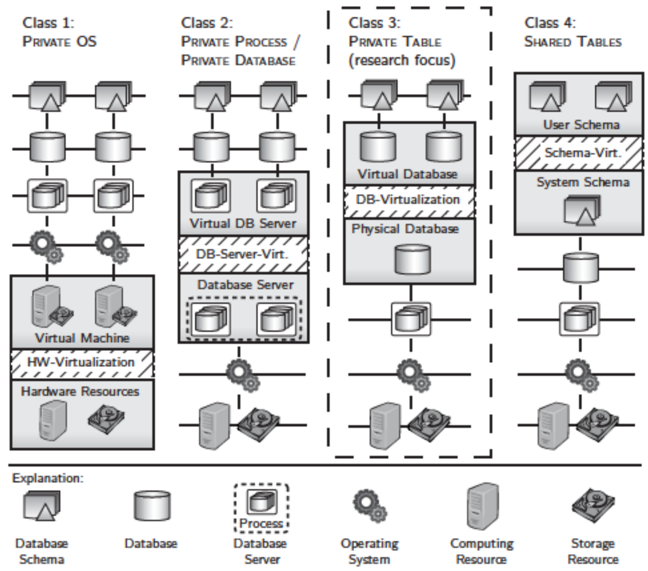

Figure 1. Possible implementations for database clouds

(motivated by the possibly high migration costs for data-intensive computations).

This paper and the corresponding poster give a brief overview of database virtualization techniques and their advantages and disadvantages (Section II). We argue in favor of PRIVATE TABLE virtualizations and propose a *virtualization advisor* (Section III) to automate setup and configuration. The main contributions of this work are an overview of the requirements of the advisor as well as a proposed architecture for implementing it.

II. Database-as-a-Service Implementations

The layered system stack of a DBMS – from the database schema down to the operating system – allows for consolidation at different levels. Assuming that only one resource is virtualized at a time, we consider four classes of database clouds [1], Figure 1 (from left to right):

*Class 1 (*PRIVATE OS*):* Class 1 virtualizes DBMSs using VMs, e.g., VMware or XEN. This class provides the highest isolation with respect to security, performance, and failover while at the same time presenting the highest resource requirements per application caused by redundancy

Final edited form was published in "Fourth IEEE International Conference on Utility and Cloud Computing. Melbourne 2011", S.328-329, ISBN 978-1-4577-2116-8
https://doi.org/10.1109/UCC.2011.52

of the system stack. Experiments showed that existing virtualization technologies are not well suited to fit the needs of common DBMSs for several reasons [2]; (1) DBMSs make strong and static assumptions regarding the characteristics and performance of the underlying physical system, (2) DBMSs are designed to use all the resources they are given, which makes it hard for virtualization layer to estimate the true resource requirements. Hence, Class 1 database clouds (although simple to implement and independent from the DBMS) are only useful in scenarios where isolation is more important than performance and utilization. Soror et al. investigate Class 1 DB clouds [3].

*Class 2 (*PRIVATE PROCESS/PRIVATE DATABASE*):* In Class 2 database clouds, the database server is virtualized either as a private process per application or as a private database on a shared server. Implementing this class requires fewer resources per application (compared to Class 1), but isolation between tenants is weaker. The DBMS has to provide private processes (or databases) to implement this class – a feature found in many products.

*Class 3 (*PRIVATE TABLE*):* Databases are virtualized in Class 3 database clouds. All applications get access to private tables in virtual databases which are in turn projected to a few physical databases. This class presents a good trade-off between isolation, complexity, scalability, and resource utilization. Our research concentrates on Class 3 database clouds as they are well-balanced and well-suited for various applications.

*Class 4 (*SHARED TABLES*):* Shared table database clouds [4] are mainly used as multi-tenant databases for SaaS offerings, where a large number of tenants share a core application (and database schema). The slightly different schemas seen by the various users are projected to shared system schemas in the virtualization layer. This class provides the least isolation (all tenants' data is stored in the same table!) but best scalability to large numbers of tenants. Since it requires a similar database schema among tenants, it is not suitable for general purpose database clouds.

III. DATABASE VIRTUALIZATION ADVISOR

Our goal is to develop a *virtualization advisor* for PRIVATE TABLE database clouds. This includes methods to assist with design and configuration of the cloud such that machine utilization is maximized, over-all resource needs are minimized, and service quality is ensured. Figure 2 shows how different components of the advisor interact.

Any solution comprises (1) the *assignment* of logical to physical resources and (2) the *configuration* of the resources. Our advisor uses workload descriptions and a cost model to find and evaluate solutions. Thereby, different objectives (e.g., performance and utilization) need to be weighted.

The main opportunity for consolidation stems from the observation that loads of different production servers are phased; high at times, idle most of the time, and low on average. Our novel cost model for a shared database considers the SQL workload as a function of time in order to predict the system utilization.

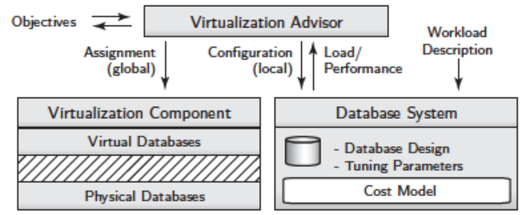

Figure 2. Virtualization Advisor Components

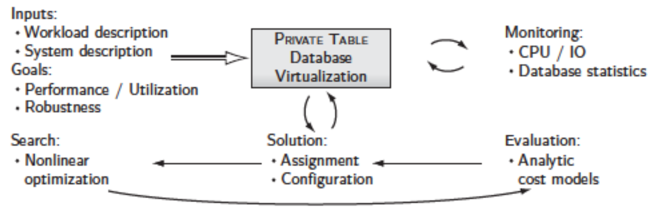

Figure 3. Virtualization Advisor Principles

The principles shown in Figure 3 convey the basic ideas of our envisioned virtualization advisor. The inputs (workload and systems descriptions) and optimization goals need to be defined and described properly. A solution (assignment and configuration) needs to be found (using, e.g., non-linear and combinatorial optimization) and evaluated (using analytic cost models). Once implemented, the database cloud needs to be constantly monitored to check for service degradations.

We are currently working on a research prototype, implemented on top of the PostgreSQL DBMS, to test and demonstrate different parts of the virtualization advisor.

IV. CONCLUSION

We outlined that a relational database cloud (or DBaaS) is a key component of cloud computing. DBMSs present unique challenges to any database cloud implementation. PRIVATE TABLE implementations are promising, but require algorithmic support for configuration in order to lead to cost reduction and improved system utilization while preserving query performance and throughput.

REFERENCES

[1] D. Jacobs and S. Aulbach, "Ruminations on multi-tenant databases," in *BTW '07*, 2007.

[2] C. Curino *et al.*, "Workload-aware database monitoring and consolidation," in *SIGMOD '11*, 2011.

[3] A. A. Soror *et al.*, "Automatic Virtual Machine Configuration for Database Workloads," *ACM TODS*, vol. 35, no. 1, Feb. 2010.

[4] S. Aulbach *et al.*, "Multi-tenant databases for software as a service: schema-mapping techniques," in *SIGMOD '08*, 2008.