Research Proposal

Yaser Mansouri
Adelaide University
School of Computer Science
Center for Research on engineering Software Technologies (CREST)

#1

Workload-based Auto-scaling of infrastructure

Infrastructure auto-scaling helps us to optimize the performance of our applications while reducing infrastructure costs by easily and safely horizontal and vertical scaling. Auto-scaling of infrastructure allows us to automate how a cluster of computing nodes responds to changes in application demand. Auto-scaling of resources is suitable for applications in which the demand for resources is a time-varying parameter. A good example of such an application is a business application that follows three-tier architecture including (i) the presentation tier (i.e., user interface), (ii) the application tier in which data is processed, and (iii) the data tier, in which the data related with the application is stored and managed in databases.

We recently extensively evaluated the performance of the most-used distributed databases including Cassandra, MongoDB, Redis, and MySQL on hybrid clouds. In our evaluation, we pre-defined demand in the number of computing nodes that are characterized by CPU cores, RAM, and storage capacity. Such configuration is suitable for prior-known and constant workloads/loads.

We intend to design and implement auto-scaling infrastructure for unknown and time-varying workloads for distributed databases in which the number of resources can be increased and decreased on the fly as the workload changes over time (horizontal scalability). In addition, our framework can react to workload changes by empowering computing nodes (vertical scalability). For this purpose, we need to constantly monitor resource utilization in terms of CPU and RAM and accordingly, the underlying resources scale out or scale based on the defined target values. The target value represents the optimal average utilization or throughput for resources in which databases are deployed. Moreover, we explore auto-scaling against computing node failure and replace them with new computing nodes. Thus, the aims of this research are:

- Design and implementation of framework for a dynamic cluster of computing nodes
- Design and implementation of workload-based auto-scaling framework for distributed databases

The outcome of this research proposal is code development for a big-data framework at the infrastructure level as well as a scientific paper.

#2

Constrained distributed databases orchestration

Containerization is a lightweight alternative to Virtual Machines (VMs) and encapsulates applications in containers with independent operating systems. Containerized applications are beneficial in quick start, deployment, and deleting the whole system setup. With more and more born in microservices and serverless, the containerization of applications is becoming an integral part of cloud-native technology. Similarly, it tends to containerize databases to shift from large, monolithic database deployment to ones based on microservices and serverless. In addition to the advantages of containerized databases from a deployment perspective, it is claimed that application containerization has less overhead in terms of infrastructure deployment. These motivate us to investigate the performance of databases running on container virtualization.

We recently evaluated the most-used distributed databases including Cassandra, MongoDB, Redis, and MySQL on hypervisor-based virtualization in different cloud models. Due to the potential less overhead on CPU, network, and disk, we intend to evaluate the performance and resource usage of these databases on container-based virtualization cloud infrastructure. Moreover, we aim to evaluate the impact of the horizontal and vertical scalability of a containerized cluster on database performance. For this purpose, we leverage Kubernetes (sometimes referred to as K8s) to manage docker-based containers cluster. These investigations give us an insightful viewpoint of comparison between container- and hypervisor-based virtualization in terms of database performance and resource utilization. Thus, the following are the objectives of this research:

- Performance evaluation of distributed databases on container-based virtualization
- The impact of horizontal and vertical scalability of the container-based cluster on the performance of distributed databases
- Resource utilization of distributed databases running on container-based virtualization

The outcome of this research proposal is scientific papers.

#3

Design and Implementation of Hybrid Databases on Cloud Infrastructure

A hybrid database is a database system in which both on-disk and in-memory data storage are used. This combination supports high-performance data processing in the main memory and provides huge storage capacities on the physical disk. Hybrid databases are used when the system needs high performance with a small footprint that only in-memory database systems can provide. Due to the flexibility of hybrid databases, developers can then strike a balance between performance, cost, and persistence.

A hybrid database acts to do cashing in which the application server first checks whether the cache copy exists, if it does it returns the data to the application server; otherwise, it reads data from on-disk data stores and sends back the answer to the application server and one copy of data is stored in the in-memory data store (Figure 1). The well-known in-memory data stores are Redis and Memcache, and the most used and popular disk-based data stores are MongoDB, Cassandra, MySQL, etc.

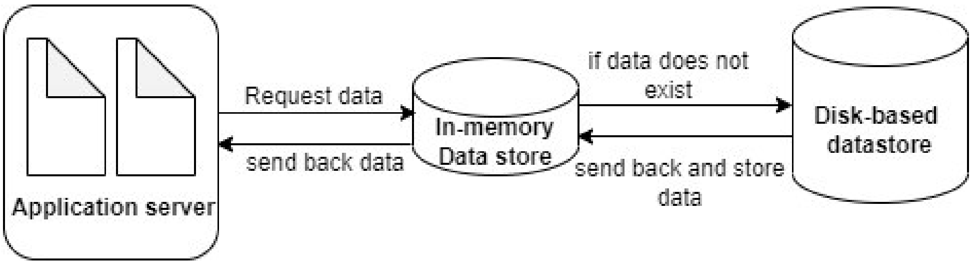

Figure 1. *A schematic of hybrid database*

We recently evaluated the performance, scalability, and resource utilization of in-memory and on-disk data stores including Redis, Cassandra, MongoDB, and MySQL as primary databases on different clous models. A combination of these databases to improve performance have not been explored. For this purpose, we intend to implement a combination of Redis with each of on-disk data stores including Cassandra, MongoDB, and MySQL. This helps us to measure the performance of these hybrid databases as well as their resource utilization. To make a comparison between Redis with another most-used in-memory data store, Memcache, we can replace this data store with Redis in the hybrid databases.

To evaluate the implemented hybrid databases, we should implement applications to generate workloads to exploit both in-memory and on-disk data stores. One solution for this challenge is to extend YCSB workloads for different workloads to store and retrieve data from hybrid databases. Alternatively, we can implement a web application to generate the desired workloads through open-source frameworks such as Express JS, Koa, FastApi.

The followings are the objective of this research:

- Design and implement of hybrid databases including Redis and Memcache as in-memory data store, and Cassandra, MongoDB, and MySQL as on-disk data stores.

- Performance evaluation and resource utilization of hybrid databases
- Make a comparison between Redis and Memcache in term of performance and resource utilization

The outcomes of this research proposal are scientific papers.