Design and Application of a Containerized Hybrid Transaction Processing and Data Analysis Framework

Ye Tao, School of Information Science & Technology, Qingdao University of Science and Technology, Qingdao, China

Xiaodong Wang, Department of Computer Science and Technology, Ocean University of China, Qingdao, China

Xiaowei Xu, Department of Computer Science and Technology, Ocean University of China, Qingdao, China

ABSTRACT

This article describes how rapidly growing data volumes require systems that have the ability to handle massive heterogeneous unstructured data sets. However, most existing mature transaction processing systems are built upon relational databases with structured data. In this article, the authors design a hybrid development framework, to offer greater scalability and flexibility of data analysis and reporting, while keeping maximum compatibility and links to the legacy platforms on which transaction business logics run. Data, service and user interfaces are implemented as a toolset stack, for developing applications with functionalities of information retrieval, data processing, analyzing and visualizing. A use case of healthcare data integration is presented as an example, where information is collected and aggregated from diverse sources. The workflow and simulation of data processing and visualization are also discussed, to validate the effectiveness of the proposed framework.

KEYWORDS

Container-as-a-Service, Containerized Computing, Data Integration, Data Interfaces, Hybrid Framework, NoSQL, Service Integration, Workflow Orchestrator

INTRODUCTION

The amount of structured and unstructured transaction data has grown dramatically over the past few years, in the fields of e-commerce, smart city (Chen et al., 2016), digital home, mobile/wearable devices and *etc*. Due to the large number of service resources, different computing capabilities, diversified data formats and processing algorithms, data extraction, transformation and loading (ETL), as well as reporting, analysis and visualization have become extremely sophisticated.

Nowadays, more and more traditional systems and applications are migrated to cloud-based services offered by a variety of providers. Generally, a traditional data preparation and analyzing process includes extracting data from heterogeneous transaction systems, cleaning duplicated/incomplete items, transforming data types, reorganizing and loading tables into data warehouses, and analyzing its subsets by using business intelligent tools. However, data volumes increase in rates that has not been seen before, and this results in the requirements to handle big data analysis workloads

DOI: 10.4018/IJGHPC.2018070106

and processing on a scalable platform, pursuing for high scalability, high availability, and high fault-tolerance (Chen et al., 2016).

As a popular distributed computing framework, Apache Hadoop (Greeshma & Pradeepini, 2016) provides a software library for reliable, and scalable computing solution to store, access and process vast amounts of data in-parallel on large clusters. Based on HDFS (Karun & Chitharanjan, 2013) and MapReduce (M/R) modules (Palanisamy et al., 2015), Hive (Yin et al., 2014) enables data warehousing tasks on those distributed data storage systems, which converts SQL statements to M/R jobs. All these above components form a "stack" of open-source modules to support analytical workloads.

However, the aforementioned Hadoop ecosystem is not designed for online transaction processing (OLTP) workloads, as Hive does not provide insert/update operations at row level. Most existing mature transaction processing systems (TPS) and programming frameworks/modules (Ding et al., 2017) only support relational databases (RDB), and so far, few are compatible with Hadoop/Hive as their back-end data providers. It is complex and verbose to integrate and migrate interactive operations, services and data from legacy systems. On the other hand, leveraging unstructured and dynamic schemas, NoSQL databases take advantages for operational storage of heterogenous and high-dimensional Big Data, to implement data analysis and knowledge discovery. Therefore, a complete set of tool-kit for data integration, analysis and visualization is required, to simplify complex data preparation, transformation and analytics tasks, with configurable and visible interfaces.

In this article, a hybrid framework is proposed, to bridge the gap between traditional TPS (applications and structured data) and big data analysis systems (distributed algorithms for large volume of unstructured data). There are several advantages that could be of interest to both end users and application developers: 1) the proposed framework supports operations from various sources (e.g. tables/files) where the original descriptive information is kept; 2) it adapts to massive data storage based on HDFS and is capable of processing a large number of records in a parallel manner by running multiple data processing and analysis tasks in a batch of M/R jobs; 3) it provides a flexible service-based access mechanism for both relational and document-based data model, to simplify programming and improve performance; 4) it integrates with existing business modules and RDB technologies that have already been deployed.

The rest of the paper is organized as follows: Section 2 briefly introduces the state-of-the-art of inter-operation techniques between RDB and NoSQL databases, as well as their applications in the field of healthcare management. Section 3 gives an overview of the architecture of the framework and details the key modules within it. Section 4 presents an example use case of healthcare data integration from different providers, showing the process of aggregating, synchronizing and visualizing massive data from different sources. Section 5 describes the simulation environment, and discusses the experiment results. Finally, Section 6 concludes the paper and discusses further research directions.

RELATED WORK

Recently, the combination of transaction and analytical processing has been studied, as service computing models and NoSQL database technologies made it possible to run transactions and analytics within a unified solution (Alami & Bahaj, 2017). A schema conversion model for transforming SQL database to NoSQL was proposed in (Zhao et al., 2014). Authors in (Lawrence, 2014) presented a generic architecture that allows seamless integration of MongoDB and any software supporting JDBC. In (Xiu-Jun & University, 2014), authors discussed a hybrid storage strategy with MongoDB and MySQL, to provide data cloud storage services and query operations of e-government data, saving the storage space and enhancing the scalability of backend database. And in a more recent research (Gyorödi et al., 2016), the authors presented a comparative study between the usage capabilities of MongoDB and MySQL, as a back-end for an online platform. (Lomotey & Deters, 2015) proposed a data analytics framework that enables information retrieval and filtering from document-based NoSQL. (Liao et al., 2016) presented mechanisms for data query, transformation and synchronization