Evaluating Distributed Transactional Database System

Pengwei Ma
China Academy of Information and Communications Technology
52 garden north road
Haidian District, Beijing, China
0086-15201365784
mapengwei@caict.ac.cn

Kai Wei
China Academy of Information and Communications Technology
52 garden north road
Haidian District, Beijing, China
0086-18600832598
weikai@caict.ac.cn

Chunyu Jiang
China Academy of Information and Communications Technology
52 garden north road
Haidian District, Beijing, China
0086-18500188116
Jiangchunyu1@caict.ac.cn

ABSTRACT

With the rapid development of mobile Internet and electronic commerce, massive transactional operations under new data modules becomes the dominated workload of the database systems. In order to meet the functional and performance requirements of this kind of business, new distributed transactional databases system evolve fast. However, because of the distinct technical differences between the new distributed transactional database and traditional centralized transactional database, such as technical component architecture and principles of computing and storage, the evaluation framework and method for traditional database cannot directly apply to distributed transactional database. For example, the existing benchmarks can't evaluate high availability, scalability, and distributed consistency issues which are important characteristics after distribution. It is urgent for the industry to construct evaluation methods for distributed transactional databases so as to comprehensively measure the technical maturity and capability of these products. In this paper, we first summarize the definition and characteristics of distributed transactional database system. After evaluating the existing test benchmarks for distributed transactional database, we propose a new evaluation framework and test methods for distributed transactional database system. This paper provides a new guidance for testing and selecting distributed transactional database in the world.

CCS Concepts

• **Software and its engineering ~ Formal software verification**

Keywords

distributed transactional database, evaluating, financial.

1. INTRODUCTION

The time of electronic commerce is coming. I Internet finance and mobile Internet make payment habit change from cash to mobile payment, which results in a large number of live users, data and concurrent volume of business scenarios [1]. These business scenarios require transactional databases to support their correctness and performance. According to statistics, Taobao's "double Eleven" turnover increased 3000 times in eight years, that is, from 52 million yuan in 2009 to 168.2 billion yuan in 2017. However, in the 2011-2014 "double Eleven" process, there were congestion in 8 of the 16 listed banks. The traditional database get limitation to support new business scenario like mobile payment.

At present, large enterprises, such as financial banks, mostly adopt centralized transactional database architecture. This architecture adopts scale-up mode, that is, increasing the hardware performance of a single machine to increase the upper limit of processing power. However, with the gradual failure of Moore's law, the scale-up scheme began to reveal performance bottlenecks. Due to cost and performance reasons, most of the emerging Internet enterprises began to explore the road of distributed transactional database architecture. And through many years of practical experience, it is proved that the distributed transactional database architecture can support the large-scale and high concurrency business scenario [2]. Taking Alipay as an example, the whole line of IBM minicomputers and EMC storage devices were removed in 2013, and a new distributed database was developed to replace Oracle databases in 2014. Oceanbase and x86 servers architecture successfully passed the 2015-2017 three years of online business test. The corresponding hardware and software costs are reported to have fallen by about 60%, while performance has improved dozens of times. Therefore, the distributed transactional database will be the evolution direction of the transactional database in the new business environment.

However, the development of distributed transactional database technology is still at the beginning stage, so there are still many problems [3]. The main problems are listed below: 1) there was not a unanimous definition about distributed transactional database. Each company develop their own product based on their requirements and understandings. 2) the development of distributed transactional database technology is still isolated, which means that the related achievements and experiences of industry have not been efficiently shared. 3) distributed transactional database gets new features, the industry call for new benchmark tools to measure all kinds of technical realization.

In order to solve the above problems, this paper firstly studies the technology and industry of distributed transactional database, and gives the definition of distributed transactional database. At the same time, a comparative study of various distributed transactional database technology routes is carried out, and the technical characteristics are compared and summarized. Then, the shortcomings of the current transactional database testing benchmark are analyzed. and we developed the functional and performance evaluation methods of distributed transactional database which covers all kinds of technical features. This paper

ICCMS 2019, January 16–19, 2019, Melbourne, VIC, Australia
ACM ISBN 978-1-4503-6619-9/19/01…$15.00
DOI: https://doi.org/10.1145/3307363.3307364

makes some suggestions for the development of distributed transactional database.

2. OUTLINE

The rest of this paper is structured as follows: In Sect.3 we introduce the standard definition and typical technical characteristics of distributed transactional database. Section 4 analyzes the advantages and disadvantages of the current distributed transactional database technology architectures. Next, Sect.5 analyzes the shortcomings of the current transactional database evaluation methods in the current environment. Sect.6 gives the implementation principles of the distributed transactional database testing benchmark and introduces the technical specifications of our functional evaluation methods for distributed transactional database. Finally, we describe how the evaluation methods of distributed transactional database will be used in the future in Sect.7.

3. DEFINITION OF DISTRIBUTED TRANSACTIONAL DATABASE

At present, there is no uniform definition of distributed transactional database in the industry. This paper summarizes the distributed transactional database into three core attributes and six core capabilities after summarizing the views of academia and industry.

3.1 Three core attributes

The three core attributes of a distributed transactional database are as follows:

a) Transactional database properties

The database core data logical management model has the transaction ability, conforms to the complete ACID (atomicity, consistency, isolation, persistence) attribute request.

b) Massive Online transaction processing capability

A concerned database management system that can handle massive online transaction processing loads.

c) Distributed architecture attribute

Both computing and storage make effective use of the parallel capabilities through multiple nodes.

3.2 Six core capabilities

The six core capabilities of a distributed transactional database are listed as follows:

a) Computing and storage can be implemented in hardware based on industrial standards such as X86

The new distributed transactional databases are based on the standard X86 industry standard hardware, but not on the original mainframe and minicomputer proprietary hardware. This is helpful to control the budget and unify the development method, so that it can be widely popularized in the industry.

b) Comply with the relational model of the transactional database and fully meet the ACID requirements of the transaction

Distributed transactional databases are mainly used to support transactions. Distributed transactional databases need to be able to fully support the ACID attribute.

c) Ability to support highly concurrent OLTP workloads

With the rapid development of mobile Internet, the number of OLTP services supported by distributed transactional databases will increase exponentially. The distributed transactional database needs not only to ensure the correctness of the business, but also to be able to support high concurrent performance requirements, so as to truly meet the requirements of the business development.

d) Support for standard SQL (standard query language) data manipulation

At present, most of the database services are written in the SQL language. If the distributed transactional database can't support complete standard SQL, then the stock business will face large-scale transformation, which is intolerable to the database application. Therefore, the ability of supporting standard SQL in distributed transactional database is the precondition of database distributed transformation.

e) High availability guarantee for automatic Fault Self-Healing and switching

After the transactional database is distributed, the pressure of computing and storage is shared by each node. At this time, if a single node fails, the database system needs to perform the switching process quickly to ensure the continuity of business.

f) Support scale-out ability across data center levels

Due to the continuous growth of transactional business scale, the distributed transactional database will be put forward higher and higher performance requirements. Because distributed mainly depends on scale-out ability to improve processing performance, distributed transactional database needs horizontal expansion ability at the level of data center to meet the needs of future business growth of transaction processing.

4. ANALYSIS OF CURRENT DISTRIBUTED TRANSACTIONAL DATABASE

This chapter will analyze the current development status of distributed transactional database from two aspects of technology and industry.

Since the distributed transactional database is still in the early stage of development, and each database product is gradually developed based on its own business practice, different technical routes and theoretical frameworks have been formed. Database theorists have put forward a new data management model, trying to fundamentally solve related problems. However, the database industry tries to find a breakthrough in the database design architecture through the practice of various engineering research and development methods [4].

Combining the academic and industrial achievements in the field of distributed transactional databases in the past 10 years, we can sort out the development direction of several kinds of distributed transactional databases.

Table 1. Table captions should be placed above the table

Primary classification	Secondary classification	Tertiary classification	Technical characteristics
Based on centralized transactional database	Depending on the application end to complete the transformation		The business layer intrudes strongly into the business layer in order to realize the distributed correlation characteristics.
	Middleware based	Implementation of	Limited intrusion into the business

	transformation	distributed characteristics in the Intermediate layer embedded in the Application layer	layer
		Implementation of distributed feature based on Independent Middleware	Limited intrusion into the business layer
Based on New Database Theory			New Computational Storage algorithm and Engineering implementation of distributed correlation, weak intrusion to the Service layer

The detailed descriptions of three distributed transactional database technology solutions are as follows:

a) Business distributed transaction Transformation Scheme based on single transaction Relational Database

This scheme can effectively utilize the maturity advantage of the original single computer transaction processing database. By establishing the rules of data sharding and data routing in the application layer, a composite distributed transactional database architecture is established [5].

This architecture requires the DBA (Database Administrator) to manually control the distribution and management of application traffic. Therefore, the intrusion into the business layer is more serious, but it can alleviate the disadvantage of the lack of storage and computing power of centralized database.

b) Solution of distributed Management component based on single transactional database

By improving the original architecture of single transactional database engine layer, the independent distributed control logic unit is used to implement the automatic routing strategy. The consistency problem of distributed transactional database is solved by matching theory and algorithm of consistency. This method is mainly divided into embedded middleware mode and independent middleware mode two kinds of solutions.

Embedded middleware uses DAL (Data Access layer) to manage data fragmentation rules and routing strategies. The DAL here is not a standalone middleware. It is a middle-tier pattern embedded in the business side and is typically provided to the application system for invocation in the form of a Jar package.

Independent middleware manages data fragmentation rules and routing policies by proxy server mechanism. Independent middleware is usually a separate logical and physical processing system architecture. It is isolated from the business layer and the data storage and computing layer, focusing on the data distribution management function.

c) New distributed Database Theory and Engineering implementation Architecture

This kind of database is usually composed of four parts: database-driven, computing node cluster, data node cluster and global management node [6]. Multiple copies of the data are stored on multiple nodes of the data cluster in a customized or automatically optimized strategy. The data nodes do not share the data, and the synchronization between the replicas is realized by the automatic consistency algorithm. The computing node is responsible for distributed query processing and global transaction control. The global management node is a necessary component to realize the consistency of distributed transactions and to maintain the partition and topology of the sub-table. The nodes communicate with each other through a high-speed interconnected network, thus completing the rapid processing and response to the application data requests.

5. ANALYSIS OF CURRENT TRANSACTIONAL DATABASE BENCHMARKS

Transactional database has accumulated a lot of test experience in the long process of development [7]. Many organizations and individuals rely on this testing experience to create common testing tools and benchmarks to verify the functionality and performance of transactional databases. First of all, this chapter briefly introduces the transactional database testing tools which are commonly used in the industry[8].

a) Sysbench.

Sysbench is an open source multithreaded performance testing tool that performs performance tests in areas such as CPU / memory / thread / IO / database. The database currently supports MySQL / Oracle / PostgreSQL. It is a very popular dba pressure testing tool.

However, it lacks scenario support and can only be used to test common resource performance limits.

b) TPCC

TPC-C was developed by the TPC Institute to test the performance of a typical complex OLTP system. TPC consists of five processing logics. It is a pressure measuring tool which is close to the business of e-commerce platform. It is the most commonly used benchmark tool in testing transactional database at present. Its execution logic includes New-Order, payment, Order-Status, delivery, Stock-Level.

However, TPCC can not fully reflect the ability of distributed transactional database because of its slow data generation, single business scenario, and lack of distributed characteristics to measure dimensions and so on.

c) Mysqlslap

Mysqlslap is an official stress testing tool that has been available since version 5. 1. 4. Stress tests are performed by impersonating multiple concurrent clients accessing MySQL, while providing a more detailed data performance report. It can be used to compare the concurrent pressure performance of multiple storage engines in the same environment.

However, mysqlslap still lacks distributed features to measure dimensions, and the scenario is single. At the same time, it is not compatible with non-database of mysql class.

d) tcpcopy

TCP copy is a real-time replication tool for tcp traffic, version 1. 0 developed and maintained by NetEase engineer @ tcpcopy. It is generally used to copy the online traffic of the production environment to the test environment in real time for testing. The advantage is that the test data is close to the real level and relatively simple to implement.

However, it is complex to configure, and involves the online environment, so the risk is high.

Generally speaking, current testing schemes and tools commonly used in industry cannot be applied directly to distributed transactional databases. Based on the advantages of the current common test tools, the industry needs to supplement the distributed characteristics and develop functional and performance evaluation methods and tools suitable for distributed transactional database.

6. FUNCTIONAL EVALUATION METHODS OF DISTRIBUTED TRANSACTIONAL DATABASE

The computing and storage mode of distributed transactional database is different from that of traditional centralized transactional database [9]. Therefore, the traditional centralized transactional database evaluation methods can't be directly applied to the distributed transactional database.

According to the characteristics of distributed transactional database, this paper puts forward the functional test method of distributed transactional database from six dimensions, so as to measure the capability of distributed transactional database in all directions. It provides guidance for the development and application of distributed transactional database.

The evaluation methods in detail is as follows.

6.1 Basic Function

Distributed transactional database should be able to have the basic functions of traditional centralized transactional database [10]. Thus, after the distributed transformation, the adaptation work of the business system and the personnel can be simplified to the maximum extent, and the normal operation of the existing business of the database can be guaranteed. At the same time, distributed transactional database needs to be able to reflect the basic distributed capabilities. This prepares the technology for performance extensions and new architectures such as operations and maintenance.

Basic functional indicators include:

a) Support capabilities for common data types, operators, character sets, functions, SQL syntax, etc. The ability of data slicing, partitioning, distributed transaction, isolation level setting.
b) Support ability of data fragmentation, partition operation, distributed transaction, isolation level setting.

c) Ability to execute plans, table partitions, and indexes.

6.2 Compatibility

Compatibility represents the ability of a distributed transactional database to connect with existing business systems and common database hardware and software tools [11].

This class of indicators measures the scale of impact on the existing business ecosystem after the implementation of a distributed transactional database.

Compatibility indicators mainly include:

a) Support ODBC, JDBC and other common connections.

b) implicit conversion of data types.

c) Total migration of heterogeneous data in traditional databases.

d) Ability to use other databases for remote connection operations.

e) The supporting ability of Common hardware such as x86.

6.3 Management

Complete and friendly management ability can provide effective support for the operation and maintenance of distributed transactional database. This helps database DBA managers, database business leaders, and related application support teams to have better database management and control capabilities.

Specific indicators of management capacity include:

a) Easy-to-use installation deployment, configuration management, and upgrade capabilities.

b) Distributed transactional database is capable of real-time monitoring and statistical analysis of nodes and data. At the same time, distributed transactional database can notify and alarm all kinds of information in a variety of forms.

c) Distributed transactional database can realize no deadlock or automatic processing of deadlock.

d) Distributed transactional databases can manage, view, and process common types of logs.

e) Distributed transactional databases can support online database schema definition languages and management commands.

f) Distributed transactional databases are capable of managing, viewing, and processing common types of logs in the system.

g) Distributed transactional database can backup and restore data in many forms, such as timing, increment and full quantity.

h) Distributed transactional database can customize the isolated grouping of resources within the cluster.

6.4 High Availability

Distributed transactional database is the infrastructure of a variety of core businesses, which may lead to a large number of economic property losses when failure occurs. Therefore, the high availability must be able to meet the corresponding requirements, thus ensuring the continuity and stability of the business.

Specific indicators of high availability include:

a) Distributed transactional database can provide normal database service in hardware accidents such as power outage, hard disk failure, network flashover, computer room disconnection and so on.

b) Distributed transactional database can provide normal database service under the failure of primary and secondary center data backup and synchronization.

c) Distributed transactional database can provide normal database service under CPU resource overload, IO resource overload, memory resource overload, disk space overload and other operating system software failures.

d) Distributed transactional database can provide normal database service under database service failure such as data file corruption, data node corruption, system table corruption and so on.

e) Distributed transactional database can provide normal database service in the case of application connection loss, peripheral system error and so on.

6.5 Scalability

Distributed transactional databases must provide online horizontal scaling capabilities [12]. It needs to meet the needs of customer business growth on the basis of customer business expansion without obvious awareness, so as to solve the bottleneck in disk capacity and computing performance of the existing centralized transactional database system.

Specific references to capacity expansion include

a) Computing resources for read and write services can be automatically balanced across nodes in a cluster.

b) The stored data can be balanced according to the customized requirements of the business side.

c) The cluster can realize online expansion operation without affecting the normal operation of the business.

d) Cluster can be configured to achieve online capacity reduction, so as to achieve reasonable planning and utilization of resources.

6.6 Safety

In order to keep corporate business away from legal and business risks, distributed transactional databases need complete security capabilities. For example, to prevent sensitive data disclosure and eliminate the wrong operation behavior and so on.

Specific indicators of security capacity include:

a) Distributed transactional databases are able to verify permissions for data operations at the library level and at the table level.

b) Distributed transactional database can authenticate access users. Such as implementing whitelist and blacklist reviews.

c) Distributed transactional database can audit and trace all kinds of operations performed, so as to locate and analyze the key problems.

d) Distributed transactional database can be used for flow control, so as to prevent the whole system failure caused by a certain burst of traffic which will affect the normal operation of other services.

7. CONCLUSION

In this paper, the distributed transactional database is studied deeply. Firstly, this paper proposes a comprehensive definition and feature definition of distributed transactional database. Secondly, this paper analyzes the current distributed transactional database of all kinds of technical architecture and product characteristics. Based on the analysis of the shortcomings of the distributed transactional database evaluation method, this paper proposes the distributed transactional database function and performance evaluation method. Based on the distributed feature, the method measures the special ability of distributed transactional database from multi-dimension. These works have very important guiding significance for the development of distributed transactional database. In the future, our team will automate the evaluation methods in the form of tools, and provide more convenient services for the selection and evaluation of distributed transactional databases.

8. REFERENCES

[1] Corbett J C, Dean J, Epstein M, et al. Spanner: Google's globally-distributed database[C]// Usenix Conference on Operating Systems Design and Implementation. 2012:251-264.

[2] Cheung D W, Ng V T, Fu A W, et al. Efficient mining of association rules in distributed databases[J]. Knowledge & Data Engineering IEEE Transactions on, 1996, 8(6):911-922.

[3] Zsu M T, Valduriez P. Principles of distributed database systems[J]. Springer Berlin, 1999, 24(1):31-41.

[4] Bernstein P A. Concurrency Control in Distributed Database Systems[J]. Acm Computing Surveys, 1989, 13(2):185-221.

[5] Romano P, Romano P. A uto P lacer : Scalable Self-Tuning Data Placement in Distributed Key-Value Stores[J]. Acm Transactions on Autonomous & Adaptive Systems, 2014, 9(4):19.

[6] Kumar R, Gupta N, Charu S, et al. Manage Big Data through NewSQL[C]// National Conference on Innovation in Wireless Communication and NETWORKING Technology. 2014.

[7] Kumar R, Parashar B B, Gupta S, et al. Apache Hadoop, NoSQL and NewSQL Solutions of Big Data[C]// International Journal of Advance Foundation and Research in Science & Engineering. 2014.

[8] Ardekani M S, Terry D B. A self-configurable geo-replicated cloud storage system[C]// Usenix Conference on Operating Systems Design and Implementation. USENIX Association, 2015:367-381.

[9] Ling G U, Weng H X, Hua-Liang H U, et al. OceanBase schema design for OLAP application[J]. Journal of East China Normal University, 2014.

[10] Ramanathan S, Goel S, Alagumalai S. Comparison of Cloud database: Amazon's SimpleDB and Google's Bigtable[C]// International Conference on Recent Trends in Information Systems. IEEE, 2011:165-168.

[11] Sukhwani B, Min H, Thoennes M, et al. Database analytics acceleration using FPGAs[C]// International Conference on Parallel Architectures and Compilation Techniques. IEEE, 2017:411-420.

[12] Abbas Q, Shafiq H, Ahmad I, et al. Concurrency control in distributed database system[C]// International Conference on Computer Communication and Informatics. IEEE, 2016:1-4.