ISSN No. 0976-5697

Volume 8, No. 3, March – April 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Distributed Database for Bookstore Application

Dr. D. Asir Antony Gnana Singh
Department of Computer Science and Engineering
Anna University, BIT-Campus,
Tiruchirappalli, India

Dr. E. Jebamalar Leavline
Department of Electronics and Communication Engineering
Anna University, BIT-Campus,
Tiruchirappalli, India

Srinivasan Malini
Department of Computer Science and Engineering
Anna University, BIT-Campus,
Tiruchirappalli, India

Abstract: In recent past, the bookstores generate more data such as customer details, transaction details, book details, etc. for analyzing them to make data-driven decisions. These data are stored in a centralized database. In centralized database system, the data are stored in a single site or location and that are managed. The major drawback is that if the centralized system gets crashed the entire data are lost and also it makes access bottleneck when more users need to access the database. In order to overcome these limitations, the distributed database system is used to store the business data by locating them in different databases that are located in different sites in different location. The distributed database maintains the replication and duplication that ensure the availability and fault tolerance on the data. The distributed system also provides the high accessibility to the users. Thus, this paper presents a distributed database for a bookstore to improve availability, fault tolerance, and accessibility. This distribution also maintains the transparency and supports the database properties such as atomicity, consistency, inheritance and durability (ACID). Furthermore, this distributed database is designed with four sites and each site keeps the bookstore data for access and analyses. This distributed database is constructed and accessed using structured query language (SQL) with MySQL relational database management system (RDBMS). It also provides the security and work even in offline mode.

Keywords: distributed databases; query processing; relational database management; structured query language; bookstore database;

I. INTRODUCTION

Nowadays, the bookstores tremendously generate the data that deal with customer information, transactional information, book details, vendor details, etc. These data are stored in a centralized database to make the data-driven decisions by analyzing these data. The main limitation of the centralized database system is that when the centralized database gets crashed, the entire data that are stored in the database is lost. Moreover, this system leads to access bottleneck since the more number of users need to access the data at same time.

In order to overcome these limitations, the distributed system is used to store the data. In distributed database, the data are stored in the different sites and different location. The distributed database ensures the replication and duplication of data. The replication of data ensures if a data is updated in a site that must reflect in all other sites that maintains the same copy of data. The duplication of data ensures the backup of the database that resides in the different sites. Therefore, these two properties such as replication and duplication increase the availability and provide fault tolerance. The distributed database system has many sites to store the data and the sites are located in different places [1].

Therefore, more users can access the database simultaneously from different sites thereby the access bottleneck is eliminated. The distributed database also improves parallelism and prevent deadlock. This database stores the whole data as well as information not in a single or same site. But it uses more than one, even many sites to store huge amount of data. Different sites provide the speed access of the data by allowing parallel processing. This distributed database is used in many areas including banking, real estates, organizations, companies, etc. [2]. In general, the distributed query processing is performed to carry out the transactions. This database is very efficient to work and deal with its operations by increasing the interfaces to reduce work burden. The secured mining protocol is adopted in order to provide the security in knowledge discovery from data and to maintain the high level of privacy and also to check the dependencies between data. This database also reduces the overall communication cost and computational cost for various transactions [3].

This paper presents a distributed database with four sites to maintain the bookstore data. The rest of this paper is organized as follows: Section 2 reviews the literature. Section 3 presents the distributed system with 4 sites for bookstore application. Section 4 provides the implementation details. Section 5 discuses the results. Section 6 concludes this paper with future scope.

II. LITERATURE REVIEW

This section reviews the various research works that are carried out by many researchers related to the distributed system. Claudio A. Ardagna and Ernesto Damiani proposed a configuration-independent score-based benchmark for distributed databases. This database provides a methodology score-based benchmark for SQL and NoSQL. This methodology overcomes the features of reduced latency in one user side and speed at another end. It also reduces the complexity of usage [4].

Yingpeng Sang et al presented a privacy-preserving tuple matching system in distributed database. This system preserves privacy in distributed database while identification of duplicate tuple. Moreover, they reduce the computation and communication costs using their proposed method [5]. Tamit Tassa proposed a secure mining of association rules in horizontally distributed databases.

This secure mining protocol is derived from Kantarcioglu and Clifton approach and fast distributed mining (FDM). This protocol combines two secure multi-party algorithms to perform secure mining of association rules [6]. Umut Tosun designed an evaluator algorithm-based distributed database. In this design, a set of quadratic assignment optimization algorithm is used to solve the data allocation problems and proposed a hybrid genetic multi-start tabu search algorithm to get the optimal solution [7]. Stavros Papastavrou et al presented a framework to access the web-based distributed database based on java-based mobile agents. The proposed framework is known as database management system (DBMS) - Aglet framework. This framework also supports autonomous, portable and light-weight clients [8].

R. Reinsch described the relationship between the distributed database and system application architecture (SAA). This paper also imparts the effectiveness of the distribution of structured query language and the common programming interface [9]. Masahiro Tsuchiya et al presented a distributed database management model to simulate the buffer management and usage of the database. The requirements for database access, location of the file, and other information are stored in several tables. This model is executed for managing the buffer in the distributed database [10]. Sunil K. Sarin et al presented a system to improve the high availability in distributed database and provided fault tolerance in distributed system [11].

William Perrizo presented a scheme for processing queries in distributed databases in order to achieve low response and transmission time [12]. Xianlun Dong et al provided a solution for the distributed databases using MySQL [13]. From the literature, it is observed that the distributed databases are used in various applications for efficient retrieval of data with efficient query processing. Moreover, MySQL can be the better choice for the development of distributed database. Thus, this paper presents a distributed database for bookstore application using the MySQL relational database management system.

III. DESIGN OF DISTRIBUTED DATABASE FOR BOOKSTORE

The distributed database for a bookstore is designed with 4 sites as shown in Figure 1 namely Site1 (S_1), Site2 (S_2), Site3 (S_3) and Site4 (S_4). Each site has database (DB) named as s_1, s_2, s_3, s_4, respectively.

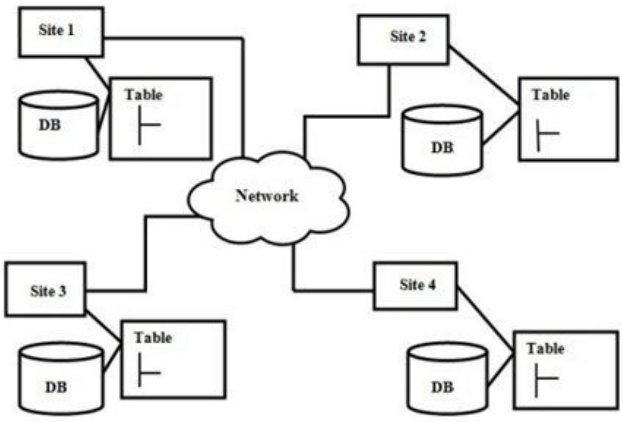


Figure 1. Distributed database architecture for bookstore with four sites

Each database s_i contains 3 relations (tables) namely Books, Bookstore and Stock. The Books relation consists of the five attributes (fields) namely ISBN, Primary author, Topic, Total stock, and Price. The Bookstore relation consists of five attributes namely Store no., City, State, Zip, and Inventory value. The Stock relation consists of three attributes namely Store No., ISBN, Qty. The Totalstock can be the total number of books in stock and inventoryvalue can be the total inventory value for the store in dollars.

IV. WORKING PRINCIPLE

This section explains the working principle. The distributed database for a bookstore is designed with 4 sites namely Site1 (S_1), Site2 (S_2), Site3 (S_3) and Site4 (S_4) and each site has database (DB) named as s_1, s_2, s_3, s_4, respectively. Each database s_i contains 3 relations (tables) namely Books, Bookstore and Stock that are interconnected with one another to pass information or messages among them and communicate with one another. Site1 has database s_1 and has 3 tables namely books, bookstore and stock. Site 2 has database s_2 and has 3 tables that are books, bookstore, and stock. Similarly s_3 and s_4 has same number of databases and tables. The query processing is performed such as insert and display details in each table, find the total number of books in stock where price is between \$15 to \$55, update the book price of no= 1234 from \$ 45 to \$55 at Site S_3, and find the total number of book at Site S_2

V. IMPLEMENTATION DETAILS

This section presents the implementation details of the distributed database for bookstore application. This distributed system is implemented using MySQL Server version 5.7 with the system configuration of operating system : Windows 8, Processor: Intel(R) Core (TM) i5-3470 CPU @ 3.20 GHz, Hard disk : 500 GB, RAM : 4 GB.

A. Implementation Procedure

The following steps are carried out in order to develop the distributed database for the bookstore application

Step 1: Install the MySQL installer and set password.
Step 2: Open the MySQL installer.
Step 3: Create a database with the name s_1 with required fields and create 3 tables using that database with name of Bookstore, Books, Stock.
Step 4: Insert values in the table.
Step 5: Similarly, create 3 more tables with the same syntax and also insert needed values.
Step 6: Perform the query processing as follows.
Step 7(a): Find the number of books in stock between \$15 and \$55.
Step 7(b): Update the price of book with number 1234 with price \$45 to \$55 in Site S3.
Step 7(c): Find total number of books in Site 2 only.

Install the MySQL installer and set password for security. Open the MySQL installer and enter the password then performs the following steps. Create a database 1 with the name s_1 with required fields and create 3 tables using that database with name Bookstore, Books, stock. Insert the needed values in the table by using the command insert into "table name" (attribute name attribute type). Create another a database 2 with the name s_2 with required fields and create 3 tables using that database with name Bookstore, Books, Stock. Insert values similar to the database1 s_1 for database2 s_2, database3 s_3, and database 4 s_4. Now, carry out query processing one by one such as find the number of books in stock between \$15 and \$55, update the price of book with number 1234 with price \$45

to $55 in Site S_3, and find total number of books in Site 2 only, etc.

VI. Outputs And Discussion

Figure 2 shows the sample image of displaying inserted details of each table. Figure 3 shows the output for the query "find the number of books in stock between $15 and $55 in each Site". Figure 3 displays the total number of books in Site 1, Site 2, Site 3, Site 4, respectively. Figure 4 shows the output for the query "update the price of book with number 1234 with price $45 to $55 in Site S3". Figure 4 displays the visualizing updation in Site 3. Figure 5 shows the output for the query "find total number of books in Site 2 only". Figure 5 shows visualizing the total data in Site 2 and visualizing the sum of total stock in Site 2.

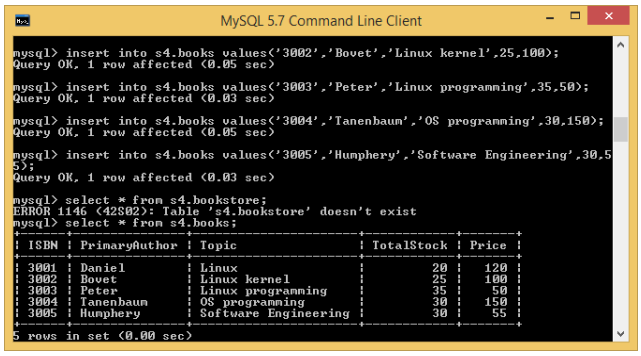
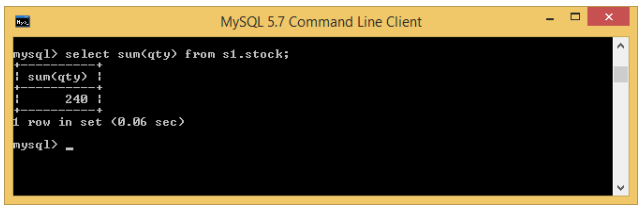
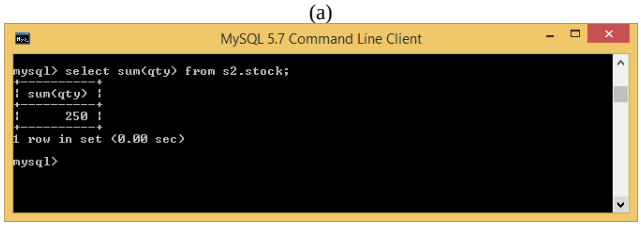


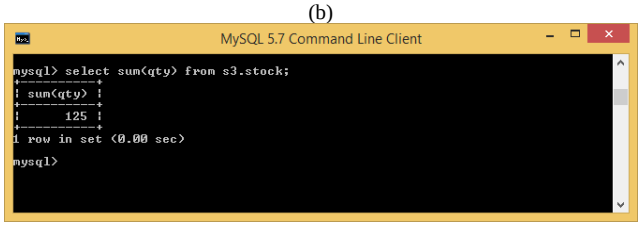
Figure 2. Displaying inserted details in each table.



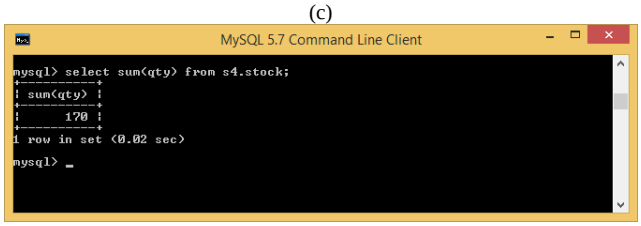
(a)



(b)

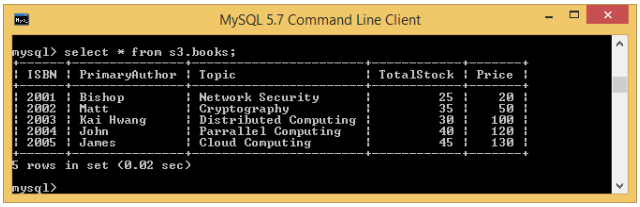


(c)

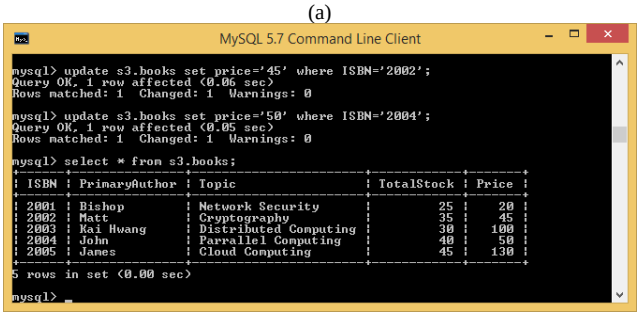


(d)

Figure 3. (a) Displaying total number of books in Site 1, Figure 3 (b) Displaying total number of books in Site 2, Figure 3(c) Displaying total number of books in Site 3, Figure 3(d) Displaying total number of books in Site 4.

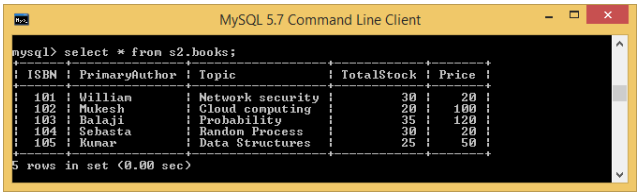


(a)

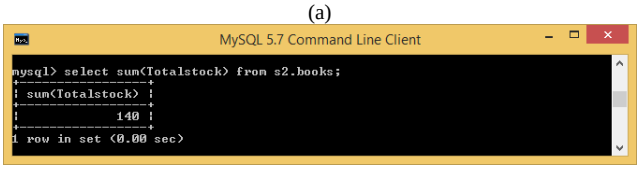


(b)

Figure 4. Figure 3(a) Visualizing updation in Site 3, Figure 3(b) Visualizing after updation in Site 3.



(a)



(b)

Figure 5. Figure 4(a) Visualizing the total data in Site 2, Figure 4(b) Visualizing the sum of total stock in Site 2

VII. Conclusion

This paper presented a distributed database for bookstore application with four sites using MySQL relational database management systems. The databases were distributed among all the sites. This database provides more flexibility, fault tolerance, and high availability since the data are replicated in each site. Moreover, it support ACID properties and it prevents sudden failures since the copy of data is stored in more than one sites. The query processing is carried out to retrieve the data from various sites. In the future work, the number of sites can be increased to provide the database management operations.

VIII. References

[1] Özsu, M. T. And Valduriez, P. Principles of Distributed Database Systems 3 Ed. Springer, pp. 1-30, 2011.

[2] Corbett, James C., et al. "Spanner: Google's globally distributed database." ACM Transactions on Computer Systems (TOCS) vlo. 31, pp.350-450, 2013.

[3] Donald Kossmann, "The state of the art in distributed query processing" ACM Computing Surveys (CSUR), Vol. 32, pp. 422-469, Dec. 2000.

[4] Claudio A. Ardagna, Ernesto Damiani, A Configuration-Independent Score-Based Benchmark For Distributed Databases , IEEE Transactions On Services Computing, vol. 9, no.1, pp.123 - 137 , Jan.-Feb. 2015.

[5] Yingpeng Sang, Hong Shen, and Hui Tian, Privacy-Preserving Tuple Matching in Distributed Databases, IEEE Transactions On Knowledge And Data Engineering, Vol. 21, no. 12, pp.1767 - 1782, Dec. 2009.

[6] Tamir Tassa, Secure Mining of Association Rules in Horizontally Distributed Databases, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, no.4, pp. 970 – 983, April 2014.

[7] Umuttosun, Distributed Database Design Using Evolutionary Algorithms, Journal of Communications And Networks, Vol. 16, no.4, pp. 430 - 435, Aug. 2014.

[8] Stavros Papastavrou, George Samaras, Evaggelia Pitoura, Mobile Agents For World Wide Web Distributed Database Access , IEEE Transactions on Knowledge And Data Engineering, Vol. 12, no.5, pp. 802 - 820, Sep/Oct 2000.

[9] R. Reinsch, Distributed Database For SAA , IBM Systems Journal, Vol. 27,no.3, pp. 362 - 369 27, April 2010.

[10] Mashiro Tsuchiya, Michael. Mariani, James. D. Brom, Distributed Database Management Model And Validation, IEEE Transactions On Software Engineering, Vol. SE-12, no. 4, pp. 511 - 520, Sept. 2012.

[11] Sunil . K . Sarin, Barbara .T. Blaustein, System Architecture For Partition-Tolerant Distributed Databases, IEEE Transactions On Computers, Vol. C-34, no. 12, pp. 1158 - 1163, Sept. 2012.

[12] William Perrizo, A Method For Processing Distributed Database Queries, IEEE Transactions On Software Engineering, Vol. SE-10, no.4, pp. 466 - 471, May 2009.

[13] X. Dong and X. Li, A Novel Distributed Database Solution Based on MySQL, 2015 7th International Conference on Information Technology in Medicine and Education (ITME), Huangshan, pp. 329-333, 2015,