IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-10, Oct. 2018 E-ISSN: 2347-2693

A study of Concurrent transaction execution and their problems in Distributed Database System

Anil Kumar Singh

Department of Information Technology, Jagran Institute of Management, Kanpur -208 014, India

Corresponding Author: anil.sysadmin@gmail.com

Available online at: www.ijcseonline.org

Accepted: 10/Oct/2018, Published: 31/Oct/2018

Abstract- More than one transactions are executing simultaneously is known as concurrent executions. It implies interleaving execution of operations of a transaction. In this paper we will study the benefits of concurrent transactions and problems with concurrent transactions. We will study above with the help of the suitable examples. This paper will help to students and research scholars to understand the concurrent transactions executions.

***Keywords*:** Concurrent Transaction**,** throughput, response time, concurrency

I. INTRODUCTION

As we know that we can execute the transactions in two ways i.e. serially and parallel. While executing transactions serially may involve consistency, safety and like features, we cannot compromise time on serial execution. We need to apply the concept of simultaneous execution of transactions much more in a multi-users system. This has to be done with additional care because of the problems that would happen in some cases.

The paper is organized as follows:
Section I contains the Introduction of Concurrent transaction execution and their problems in Distributed Database System, Section II Contains some benefits of concurrent transaction execution, Section III contains different types of problems arises in concurrency control. In this section problems are explained and their justification is given with the help of suitable example, Section IV concludes the research work with future direction.

II. BENEFITS OF CONCURRENT TRANSACTION EXECUTION

A. Enhanced throughput

When 'N' number of transactions that can be executed in a given amount of time 'Tn' is called as throughput. When we execute multiple transactions concurrently then transaction through will increase.

A better throughput is accomplished by increasing the number of cycles that can be completed in a given time than by increasing the number of simultaneous transactions processing in the same period of time [1].

B. Proper Resource utilization

It is about accessing different system resources like disks and CPUs simultaneously. As we know that most of the times these resources are not utilized appropriately. When a transaction is using CPU we may permit other transaction to use disk Input /Output to obtain the data from disks to main memory. With the help of this approach the utilization of system resources can be increased.

C. Reduced waiting time

Concurrent execution of transactions would decrease the waiting time of other transactions. Consider a circumstance where all transactions are executed serially. Transactions may be of different sizes. When transaction executing serially, one long transaction is executing and a very small transaction may be on the queue for its turn. This increases the waiting time of small transaction. Therefore, if we execute transactions all together, the waiting time would be much reduced when compared to the serial execution [2].
One more important factor being assessed is waiting time of each transaction. It is calculated as follows: Waiting time = time of transaction execution- length of transaction. Length of transaction is the totality of time of performing the instruction of transaction [3].

III. PROBLEMS OF CONCURRENCY CONTROL

When concurrent transactions are executed in an uncontrolled way then several problems may occur. The concurrency control has main 3 problems which are as follows:

- Lost updates.
- Dirty read
- Unrepeatable read (or inconsistent retrievals).

A. *Lost Update Problem (Write-Write Conflict):*

A lost update problem occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of the database item incorrect.

Example -1 Consider the situation given in figure:

Tran.-T1	Timestamp	Tran. -T2	Value of A
Read (A)	t0		100
A=A-10	t1		90
	t2	Read (A)	100
	t3	A=A+20	120
Write (A)	t4		90
	t5	Write (A)	120

In the above example transaction T1 reads the value 100 of object A, at time t0.
At time t1, Transactions-T1 decrease the value of A by 10 at this time transaction T1 holds the value of A=90
At time t2,Transactions-T2 read the value 100 of A object on the basis of value seen at time t0
At time t3,Transactions-T1increase the value of A by 20 at this time transaction T1 holds the value of A=120.
At time t4, Transaction-T1 writes value of A=90 on the basis of the value seen at time t1.

At time t5, Transaction-T2 writes value of A=120 on the basis of the value seen at time t3.

So, update of Transactions-T1 is lost at time t5because Transactions-T2 overwrites it without looking at its current value. Such type of problem is referred as the Update Lost Problem, as update made by one transaction is lost here.

B. *Temporary update - Dirty Read Problem (Write- Read Conflict):*

A dirty read problem arises when one transaction updates a database item and then the transaction fails for some reason. But its updated database item is accessed by another transaction before it is changed back to the original value. In other word Dirty read is, the reading of uncommitted data of some other concurrent transaction [4].

For example, a transaction T1 updates a record, which is read by the other transaction T2.
Then T1 terminates and T2 now has values which have never formed part of the stable database.

Example -2Consider the situation given in figure:

Tran.- T1	Time	Tran.-T2	Temporary Update value	Actual value
Read (A)	t0		100	
A:=A+20	t1		120	
Write (A)	t2		120	
	t3	Read (A)	120	100
	t4	A:=A+30	150	130
	t5	Write (A)	150	130
Read (B)	t6			
Rollback	t7			
	t8	Commit	150	130

At time t0, Transactions-T1 reads the value of A=100.
At time t2, Transactions-T1 writes value of A.

At time t3, Transactions-T2 reads the value of A which is previously written by transaction T1.
At time t7, Transactions-T1 rollbacks. So, it changes the value of “A” back to that of prior to t0. So,Transaction-T2 now has value which has never become part of the stable database.

Such type of problem is referred as the Dirty Read Problem, as one transaction reads a dirty value which has not been committed.

C. *Incorrect Summary Problem:*

If one transaction T2 is calculating aggregate summary function on a number of records, while other transaction is updating some of these records, the aggregate function may calculate some values before they are updated and others after they are updated. Thus result comes to incorrect summary [5].

In other word though one transaction takes a summary over the values of all the occurrences of a recurrent data-item, a second transaction updates some occurrences of that data-item. Thus the resulting summary does not reflect an accurate result [6].

Example – 3Consider the situation given in figure:
Let's A=500, B=1000 and Sum=0

Tran.- T1	Time	Tran.- T2	Value
	t0	Read (Sum)	0
	t1	Read (A)	500
	t2	Sum=Sum+A	Sum = 0 + 500
	t3	Read (B)	1000
	t4	Sum=Sum+B	Sum= 500+1000 =1500
Read (B)	t5		1000
B=B+100	t6		1100
Write (B)	t7		1100

Transaction-T2 is summing with the value of A and B; while, Transaction-T1 is updating the value of B. Here, the result produced by Transaction-T2 is 1500,which is incorrect. If this result is written in database, database will be in inconsistent state, as actual sum 1600 because at time of t6 the value of data item B has increased by 100 and it is written in database as 1100 instead of 1000.Here, Transaction-T2 has seen an inconsistent state of database, and has performed inconsistent analysis.

IV. CONCLUSION

In distributed database system, we can execute the transactions in two ways either serially or parallel/ concurrent. It is found that with the help of concurrent execution we may improve the throughput, reduce the waiting time and proper resource utilization. But it has some problems i.e. dirty read problem, lost update problem and incorrect summary problem.

The limitation of this paper is that I have studied concurrency transactions and their problems in distributed database systems. In future we may work on how to control the concurrency problems i.e. dirty read problem, lost update problem and incorrect summary problem.

REFERENCES

[1]. Kaspi Samuel and Sitalakshmi Venkatraman, "Performance Analysis of Concurrency Control Mechanisms for OLTP Databases", International Journal of Information and Education Technology, Vol. 4, No. 4, p.p. 313-318, August 2014

[2]. http://www.exploredatabase.com/2016/04/ advantages-of-concurrent-execution-of-transactions-in-database.html

[3]. Karami Ali and Ahmad Baraani-Dastjerdi, "A Concurrency Control Method Based On Commitment Ordering in Mobile Databases", International Journal of Database Management Systems (IJDMS) Vol.3, No.4, p.p. 39-53 November 2011

[4]. Laiho Martti, et. al., "SQL Transactions – Theory and Hands-on Exercises", Version 1.4 of the first edition 2013, pp 16, ISBN 978-952-93-2421-7

[5]. https://tutorialink.com/dbms/concurrency--control.dbms

[6]. Jitendra Sheetlani and Gupta V.K., "Concurrency Issues of Distributed Advance Transaction Process", Research Journal of Recent Sciences, Vol. 1 (ISC-2011), p.p., 426-429, (2012), ISSN 2277-2502

AUTHOR'S PROFILE

Dr. Anil Kumar Singh is Postgraduate in Computer Science, M.Sc., MCA, PGDCA (1st Position in CSJM University, Kanpur) and Doctorate in Information Technology and having a large 18 years' experience in Academic and Research. Dr. Singh has presented and published more than 20 papers in various National and International Journals and Conferences. His area of expertise is in Computer Network, Database Management, RDBMS, Wi-Fi Technology, Cyber Security, Client/Server Computing, Linux, CISCO and Ethical Hacking. He is a professional life member of Indian Science Congress Association etc. Dr. Singh has a vast experience in academic field and served as Head Computer Center in Dr. GHS-IMR, Kanpur for more than 4 years and presently working as Associate Professor and Academic Head in JIM, Kanpur since year 2005. Dr. Singh has participated various workshops and Short Term Courses organised by the prestigious institutes like I.I.T., Kanpur, I.I.T., Delhi and various technical universities.

Moreover, he has organised various workshops related with Computer Networking, Security and Ethical hacking. Dr. Singh Chaired in the Technical Sessions of International Conferences like IEEE, ICSPICC2016, organized by SSBT College of Engineering, Jalgaon, Maharashtra, IEEE, 2nd ICCIT, organized by Siddhant College of Engineering, Pune, Maharashtra.