International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 7, Issue 5, September - October 2018 ISSN 2278-6856

Testing of serializability of a schedule in Distributed Database System

Dr. Anil Kumar Singh

Jagran Institute of Management

620-W block Saket Nagar, Kanpur-208014 (U.P.), India

1.0 Abstract

When wedesign concurrency controls systems, we must display that schedules produced by the system are serializable. To do that, we must first know how to regulate, given a particular schedule, whether the schedule is serializable. In this paper we will be test that the schedule is serializable or not with the help of two scenarios.

This paper will help to our students, research scholar and database designer that how to test schedule is serializable or not.

Key Words: Schedule, Transaction, serializability, concurrency, Node, Edge, Vertex

2.0 Introduction

As we know that the schedule is an order of the read or writes operations of numerous transactions as they are executed in the Database System [1]

Serializability is the traditional concurrency scheme. It ensures that a schedule for executing concurrent transactions is equivalent to one that executes the transactions serially in some order. It assumes that all accesses to the database are done using read and write operations. A schedule is called ``correct'' if we can find a serial schedule that is ``equivalent'' to it [2].

If the concurrent execution of transactions leaves the database in a state that can be achieved by their serial execution in some order, problems such as lost update will be resolved [3].

3.0 Method

There will be a directed graph G(V,E) consisting of a set of nodes V = {T_1,T_2,T_3,T_4…….Tn} and a set of directed edges E {E_1,E_2,E_3,E_4…….Em} construct with the following rules:

1. Create a node for each transaction
2. Create a directed edge T_I ->T_j ,if T_j writes a value of an object afterT_i read it.
3. Create a directed edge T_I ->T_j , if T_j writes a value of an object after $T_i$wrote it.
4. Create a directed edges T_I ->T_j , if T_j reads the value of an object written by T_i [4]

T_i write (x)	T_i write (x) or read (x)
Rule No. 4	Rule No. 2 and 3
T_j (earliest) Read (x)	T_j write (x)

If the precedence directed graph is acyclic then only the schedule is serializable.

Now with the help of following examples we will test the schedule that is serializable or not

Example: - 1

Schedule – A

Time	Tran. -1	Tran. -2	Tran. -3
t_1	Read (x)		
t_2		Read (y)	
t_3	x:=F_1 (x)		
t_4			Read(x)
t_5		Y:=F_2(y)	
t_6		Write(y) (1)	
t_7			Z:=F_3 (z)
t_8			Write (z) (2)
t_9	Write (x) (3)		
t_{10}			Read (y) (1)
t_{11}		Read (x)(3)	
t_{12}		X:= f_4 (x)	
t_{13}	Read (z) (2)		
t_{14}		Write (x)	
t_{15}	Z:=F_5 (z)		
t_{16}			Y:=F_6 (y)
t_{17}			Write (y)

STEP – 1 In above example there are 3 transaction i.e. Transaction1, Transaction2 and transaction3. These are represented as follows as per Rule 1.

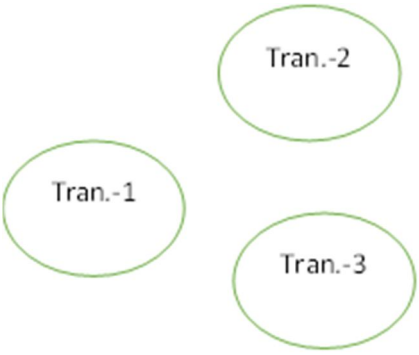

Fig. 1.1

Step 2- Transaction 3 reads the value of data item 'y' which is previously written by transaction 2. Therefore the directed edge will be created as

Trans2---> Trans3

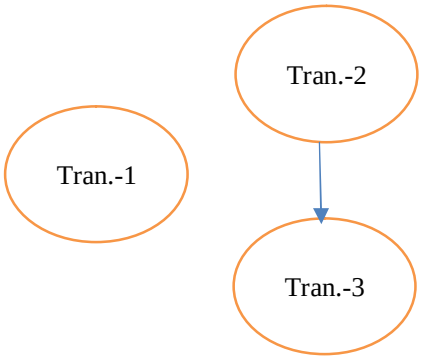

Fig. 1.2

Step 3 – Transaction1 reads the value of data item 'z' which is previously written by Transaction3. So the directed edge will be created as

Trans3---> Trans1

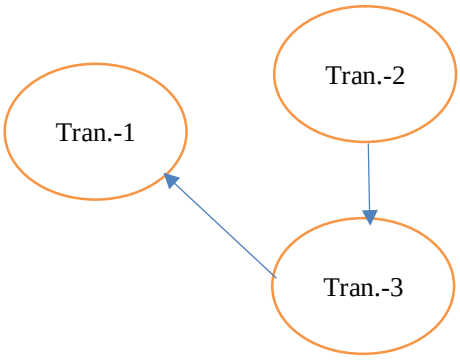

Fig. 1.3

Step 4 – Transaction2 reads the value of data item 'x' which is previously written by Transaction1. So the directed edge will be created as

Trans1---> Trans2

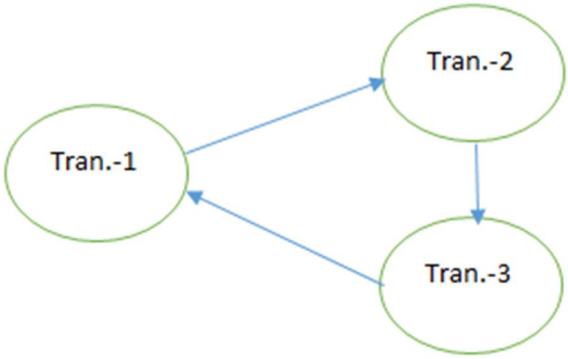

Fig. 1.4

Result - Fig. 1.4 shows that the precedence directed graph is cyclic. Therefore, the schedule will not be serialised

Example -2

Schedule – B

Time	Tran.-1	Tran.-2	Tran.-3
t_1	Read (x)		
t_2	$X:=F_1(x)$		
t_3	Read (z)		
t_4	Write (x) (1)		
t_5	$X:=F_2(z)$		
t_6		Read (y)	
t_7	Write (z) (2)		
t_8		Read (x) (1)	
t_9			Read (z) (2)
t_{10}		$Y:=F_3(y)$	
t_{11}		Write (y) (3)	
t_{12}			$C:=F_4(z)$
t_{13}			Read (y) (3)
t_{14}			Write (z)
t_{15}		$x:=F_5(x)$	
t_{16}		Write (x)	
t_{17}			$Y:=F_6(y)$
t_{18}			Write (y)

Step -1 In above example there are 3 transactions i.e. Transaction1, Transaction2 and transaction3. These are represented as follows as per Rule 1.

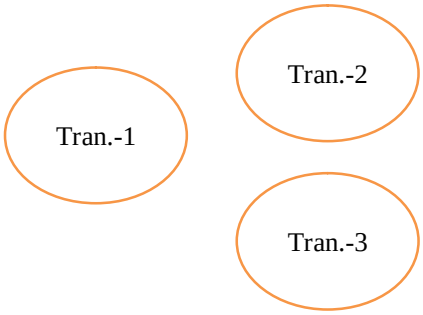

Fig. 2.1

Step -2 Transaction2 reads the value of data item 'x' which is previously written by transaction1. Therefore the directed edge will be created as
Trans1---> Trans2

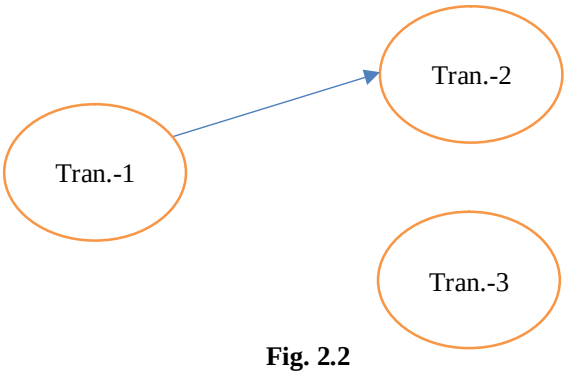

Fig. 2.2

Step -3 Transaction3 reads the value of data item 'z' which is previously written by transaction 1. Therefore the directed edge will be created as
Trans1---> Trans3

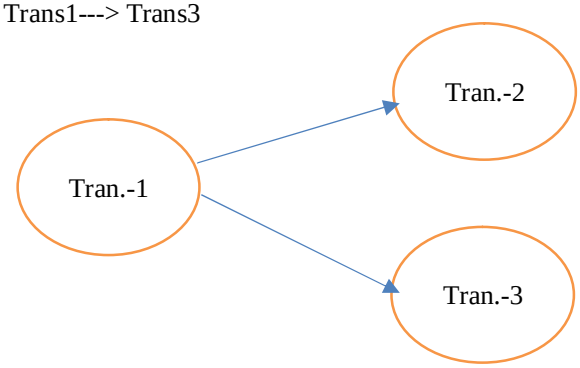

Fig. 2.3

Step – 4 Transaction3 reads the value of data item 'y' which is previously written by transaction 2. Therefore the directed edge will be created as
Trans2---> Trans3

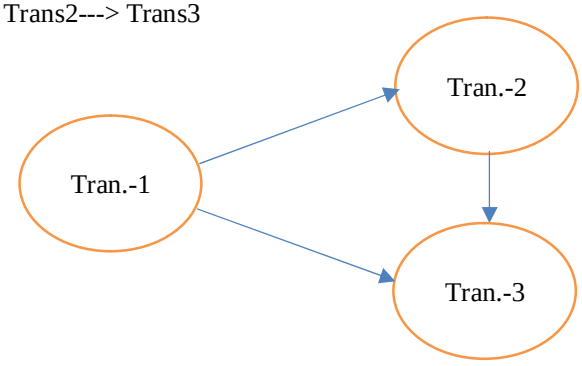

Fig. 2.4

Result – Fig. 2.4 shows that the precedence directed graph is acyclic. Therefore, the schedule will be serialised

4.0 RESULT AND DISCUSION

While going through Example-1, Schedule A, it is found that the precedence directed graph is cyclic as per the Fig. 1.4. Therefore, the schedule will not be serialised. For serialization the precedence directed graph should be a cyclic.

While going through Example-2 Schedule B, it is found that the precedence directed graph is acyclic. Hence the schedule will be serialised.

References

[1] Schedules: SERIAL and SERIALIZABLE http://webspace.cs.odu.edu/~ibl/450/pdf/view-on-line/serial/serial3.pdf
[2] Prasun Dewan https://www.cs.unc.edu/~dewan/242/s01/notes/trans/node3.html
[3] M. Tamer Ozsu, Patric Valduriez, " Principles of Distributed database system", Pearson Education, ISBN 81-7758-177-5, p.p. 250
[4] GreeksforGreeks, A computer Science portal for greaks., https://www.geeksforgeeks.org/ precedence-graph-testing-conflict-serializability/

AUTHOR

Dr. Singh has 18 years experience in Academic and Research. He is a Post Graduate in Computer Science and an MCA. He has received the Ph.D. degree in Information Technology from Mahatama Gandhi Chitrakoot Gramodaya Vishwavidyalaya, Chitrakoot in 2012. He has to his credit several research papers published in National and International journals and conferences. Presently he is working in Jagran Institute of Management, Kanpur as Associate Professor.