ISSN 2229-5518

IMPROVEMENT ON CONCURRENCY CONTROL IN A DISTRIBUTED DATABASE

C.C OGBONNA[1] , P.O ODO[2],
1,2: DEPARTMENT OF ELECTRONIC ENGINEERING, UNIVERSITY OF NIGERIA NSUKKA
Ogbonnachidubem32@gmail.com;polycarp.odo@unn.edu.ng

ABSTRACT: This paper presents an Improvement on concurrency control in a distributed database. Concurrency control approach provides rules, methods, design methodology and theories to maintain the consistency of components operating concurrently while interacting and thus the consistency and correctness of the whole system. It is shown that the problem of existing architecture of a distributed database uses bus topology that causes data collision and poor management of data transaction. The proposed architecture of the distributed database uses a star-wired topology with Intelligent Ethernet managed switches Network. This improved architecture ensures consistent and correct transaction, efficient data collection and management and recoverability of data in a distributed database. The mathematical analysis that uses Petri-nets and simulated results are also presented in the paper.

Keywords: Distributed database management System, Concurrency Control, Petri-nets

1 INTRODUCTION

Computer and its diverse applications have in recent times witnessed a revolution. Its enormous success is due largely to the flexibility and reliability that computer system offer to potential users [1]. According to [2], the last decade has seen the creation and rapid expansion of the field of distributed computing systems. This has been driven by both technical and social forces and it seems likely that the pressure from both will continue for some time yet. For over 20 years, businesses have been moving their Data processing activities on line. Many businesses, such as airlines and banks, are no longer able to function when their on-line computer system are down. Their on-line database must be up-to-date and correct at all times.

Managing large scale distributed data and activities gets difficult as the amount of heterogonous data grows. Therefore classic techniques for distributed design and query processing in relational database system have been revisited to address dynamic issues in high performance computing and flexibility challenges of xml document [3].Concurrency control is the activity of coordinating the actions of processes that operate in parallel, access shared data, and therefore potentially interfere with each other [4]. According to [5], in mobile computing environment, clients can access data irrespective of their physical location. Data is shared among multiple clients and can be updated by each client independently, thus, leading to inconsistency of data. Due to limitations of mobile computing environment, traditionally techniques cannot be used. As stated by [6], the protocol structure for our distributed database is based on the transaction oriented high level protocol, supported by the top three layers of the ISO reference model namely; the application layer, presentation layer, and the session layer. In most existing distributed databases, database function are incorporated at the application layer [7-8].The performance of such protocol is not satisfactory because of high level time-out, and excessive communication delay. The failure detection facility is usually established at the application layer in most distributed database system, which is triggered off by a high level time-out that is quite time consuming. In order to enhance the performance of the system, the failure detection facility is encapsulated in the

ISSN 2229-5518

session layer. It detects the site status of the cooperative system at allow cost and with enhanced performance [9].

Maintaining consistency requires the imposition of the events within a system. The substantial insight of [10] is that events in a distributed system only define a partial order rather than a total order. Required orderings can be achieved by extending existing centralized mechanism, such as locking, or using-time stamp based algorithms.

The study on [11] presented that a Petri Nets (PN) is a graphical tool for the formal description of the flow of activities in complex systems. With respect to other more popular techniques of graphical system representation (like block diagrams or logical trees), PN are particularly suited to represent in a natural way logical interactions among parts or activities in a system. Typical situations that can be modeled by PN are synchronization, sequentially, concurrency and conflict. The main advantages include: the graphic nature, the conciseness in comparison with state graphs, the possibility of implementing analysis techniques.

2 METHOD

The MySQL relational database was employed in collecting data for concurrency control protocols. Petri-nets, a graphical tool for the analysis of sequence of operations in the distributed database was also used in this thesis. Finally, MATLAB was applied to simulate the operation of concurrency control protocol in a distributed database

2.1 Architecture of a distributed Database management system (DBMS)

The Architecture of an existing distributed database server [12] is shown in Figure 1. This architecture uses bus topology as its Network configuration. The problem with this topology is that only one computer at a time can send data. The more computers you have on the network, the more computers that will be waiting to send data. This therefore poses a great difficulty in achieving consistent and correct data transaction due to data collision that may occur when two or more data are sent at the same time. The IEEE 802.11 DcF protocol which is used in the multi-hop N/Ws based on the Bus Topology doesn't provide the sufficient through put for each node because of the exposed/hidden node problem or collision [13]

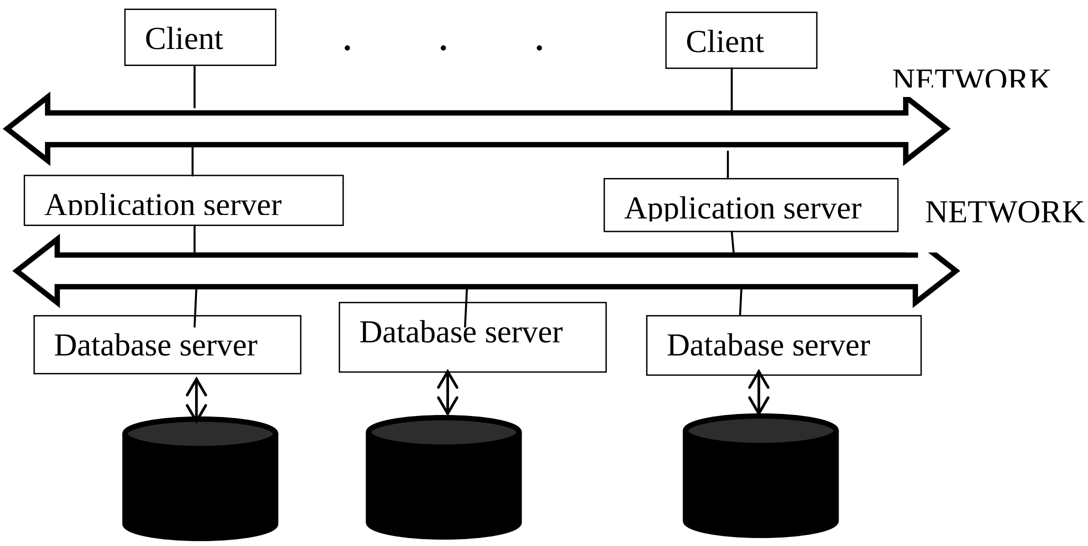


Fig 1: The Distributed Database server Architecture [12]

2.2 An Improved Architecture for concurrency control in a distributed database

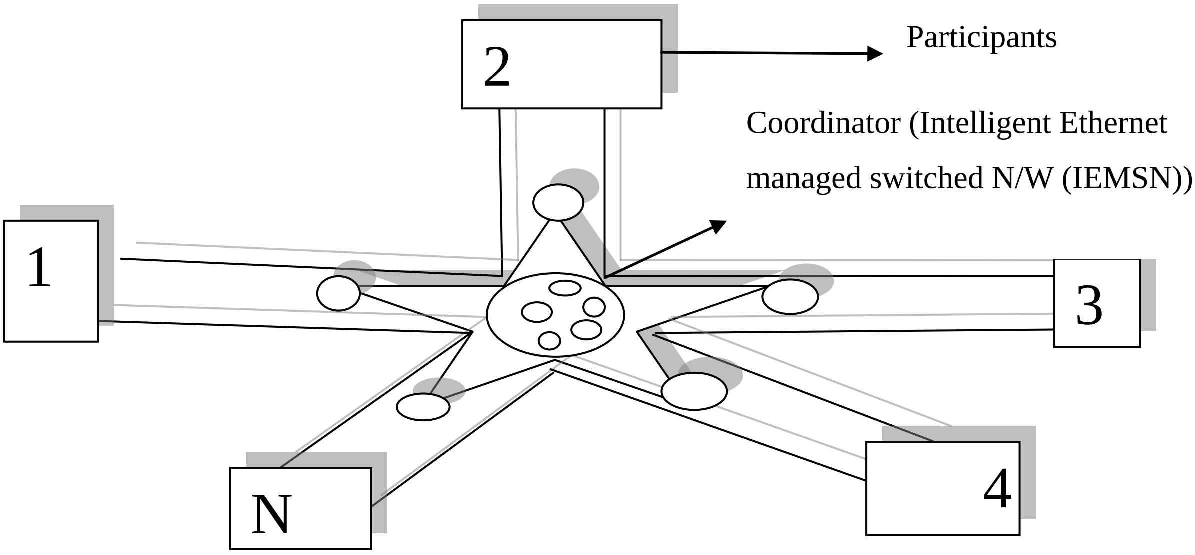


Figure 2: An improved architecture for a distributed database

The diagram in Figure 2 is an improved architecture for a distributed database. It is a star-wired topology with Intelligent Ethernet managed switched Network that is resided in the Coordinator. The architecture consists of N nodes Participants and Coordinator (Intelligent Ethernet managed switched network (IEMNS)) . Each node has a unique number from 1, 2, 3, 4, N. In star configuration, data passes through the coordinator to reach other participants on the Network. The intelligent Ethernet Switches has the ability to manage multiple traffic. Instead of flooding that traffic to all users, they use Internet Group management protocol (IGMP) to direct the traffic only to the desired recipients [14].This reduces the chances of data collisions that occur during data transaction

2.3 Petri net design approach

A Petri net is defined by the triple PN = (NET, W, M0), where NET = (P, T, F), P = {p1, p2, ,pn} is a finite non-empty set of places, T = {t1, t2, …, tm} is a finite non-empty set of transitions, and P and T are the disjunctive sets, i.e., P∩T=∅ (empty set). F is the set of ordered pairs consisting of a place (transition) at the first position and a transition (place) at the second one. F is called a flow relation .W is the weight function given as W:F→N+ where N+ is a set of positive integer [15]. The function M:P→N is called the marking of a Petri net.

PETRI- NET EQUATION

The state equation of a Petri net is shown in equation 1

$$\boldsymbol{M' = m + uD} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad .1$$

Where M' is the final state vector, M is the initial state vector, u is the firing vector and D is the incidence matrix [16].

2.3.1 Existing One-way Petri net diagram for concurrency control

The diagram shown in Figure3 is an existing One-way Petri net diagram for concurrency control. This consists of six places and four transitions. The P_k is the coordinator and P_l to P_{nm}

is the Participants, T_1 to T_n are the transitions. The Figure 3 is a forward direction approach and any attempt for backward direction will result to data collisions.

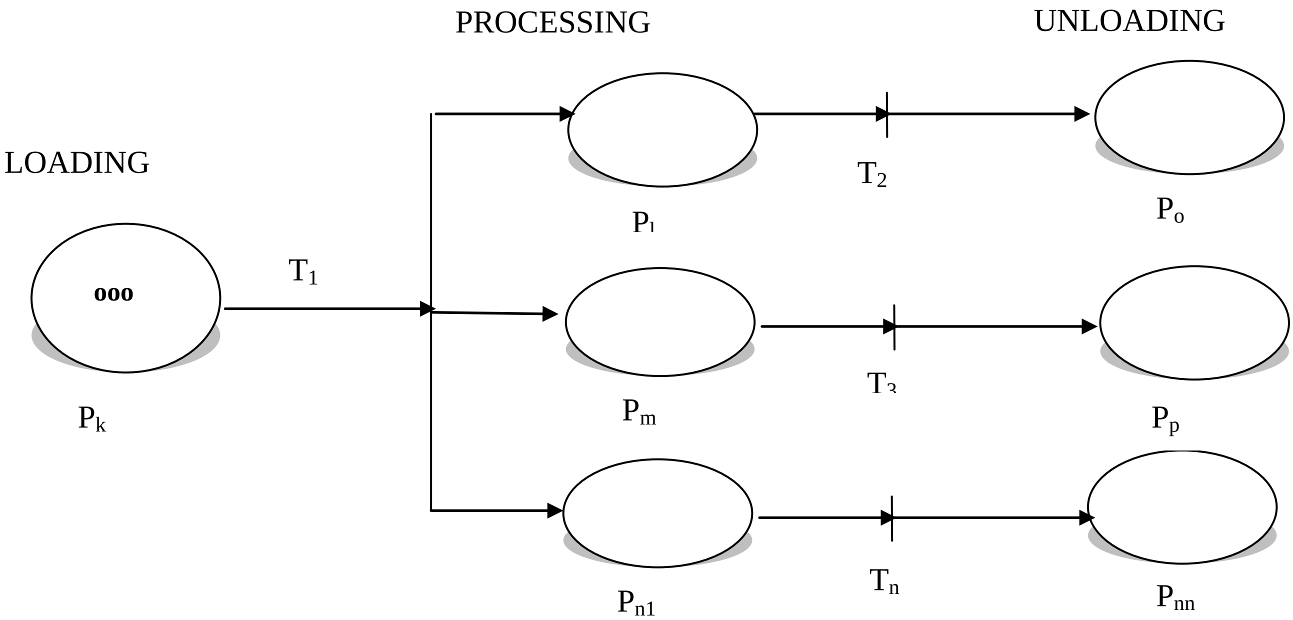


Figure 3: Existing One-way Petri net diagram for concurrency control

2.3.2 Mathematical Model For An Existing One-Way Concurrency Control

The state transition mechanism in PN is realized by moving tokens between places [16]. When the necessary conditions for a transition are satisfied, this transition can trigger. And when a transition triggers, tokens are removed from input places and added to output places. For a transition to trigger and to change the system state, the required conditions for that transition should be met [17]. The mathematical model is derived from Equation 1 and Fig.3.

$$\begin{bmatrix} Mk \\ Ml \\ Mm \\ Mn \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Pk \\ Pl \\ Pm \\ Pn \\ Po \\ Pp \\ Pq \end{bmatrix}$$

$$+ \begin{bmatrix} -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$Mk = -Pk + Pl + Pm + Pn + M0$ 2a
$Ml = 0 + 0 + 0 + 0 + M0$ 2b
$Mm = 0 + 0 + 0 + 0 + M0$ 2c
$Mn = 0 + 0 + 0 + 0 + M0$ 2d
$Mo = 0$ 2e
$Mp = 0$ 2f
$Mq = 0$ 2g

ISSN 2229-5518

2.3.3 Improved Two-way Petri net diagram for concurrency control

The diagram shown in figure 4 is an improved two-way Petri net diagram for concurrency control. This is a two stages Petri net analysis that consists of fourteen places and Eleven Transitions. The $P_{1a}$and P_{1b} is the coordinator (Intelligent Ethernet managed switched network (IEMNS)) and P_2 to P_f are the Participants.. The P_{1a} contains the initial tokens for the First stage process and it fires simultaneously to different participants during an enabling state. P_{1b}contains the initial tokens for the second stage process. The second stage starts from P_a, P_g and P_e respectively. These approach solve the problem of data collision in Figure 3.

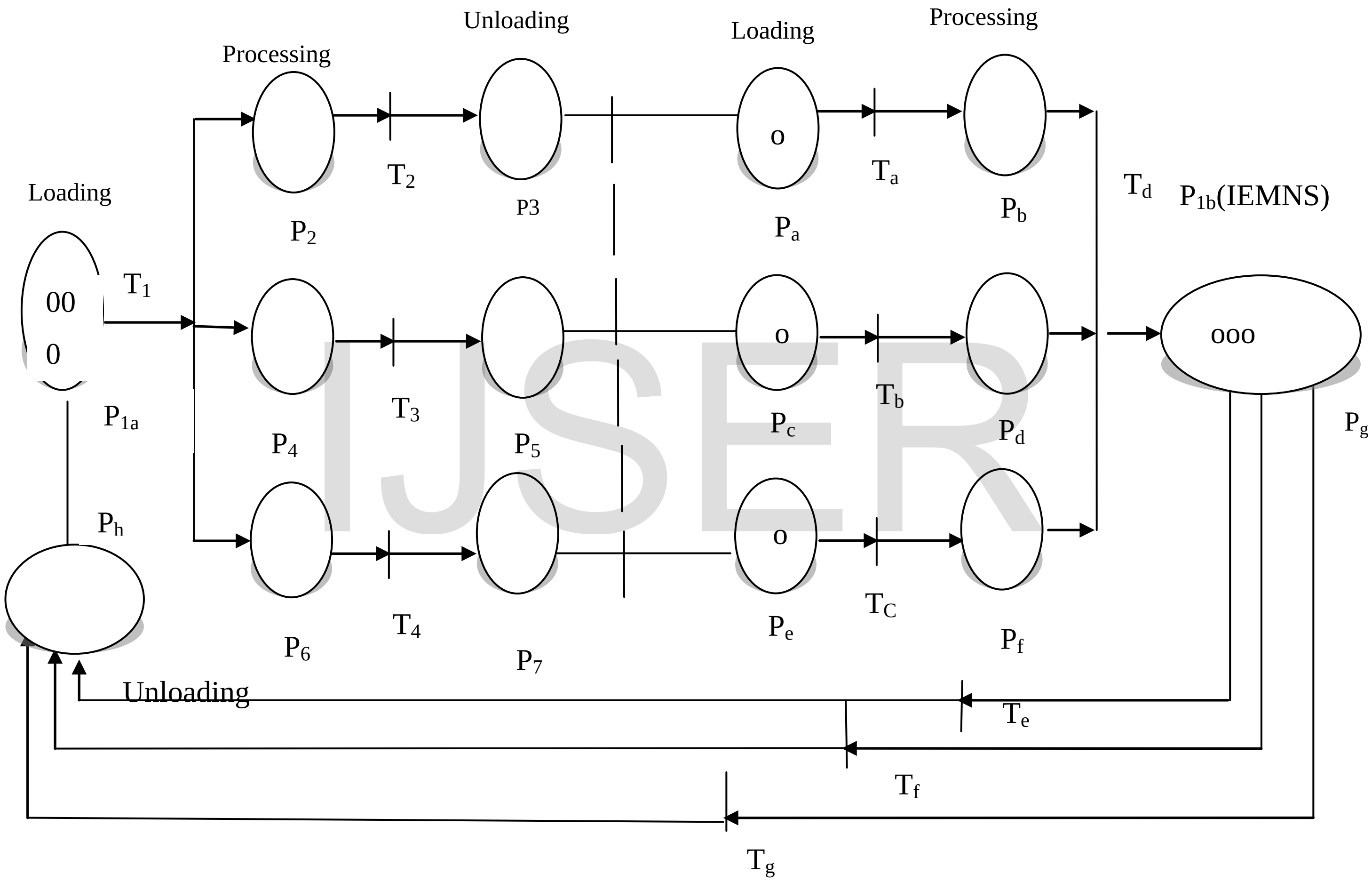


Figure 4: Improved Two-way Petri net diagram for concurrency control with buffer

2.3.4 Mathematical Model For The Improved Two- Way Forward Direction

The mathematical model in equation 2 below is derived from Equation 1 and Fig. 4.

ISSN 2229-5518

$$\begin{bmatrix} M1 \\ M2 \\ M3 \\ M4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$M1 = -P1 + P2 + P3 + P4 + Mo$ 3a
$M2 = -P1 + P3 + P4 + P5 + M0$ 3b
$M3 = -P1 + P2 + P3 + P4 + M0$ 3c
$M4 = -P1 + P5 + P6 + P7 + M0$ 3d
$M5 = 0$ 3e
$M6 = 0$ 3f
$M7 = 0$ 3g

2.3.5 Mathematical Model For The Improved Two Way Backward Direction

The mathematical model is derived from equation 1 and Fig. 4.

$$\begin{bmatrix} Ma \\ Mb \\ Mc \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} Pa \\ Pb \\ Pc \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$Ma = -Pa + Pb + M0$ 4a
$Mb = -Pa + Pg + M0$ 4b
$Mg = -Pa + Ph + M0$ 4c
$Mh = 0$ 4d

3 RESULTS AND DISCUSSION

The Results obtained by the Simulation method are shown on table 1, 2 and 3 respectively

Table 1: Results of the Existing concurrency control

M_o	1 0 0 0 0 0 0
M_k	0 1 1 0 0 0 0
M_l	1 1 1 0 0 0 0
M_m	0 0 0 0 0 0 0
M_n	0 0 0 0 0 0 0
M_o	0 0 0 0 0 0 0
M_p	0 0 0 0 0 0 0
M_q	0 0 0 0 0 0 0

3.1 Graph of the Existing concurrency control in a distributed database

The graph in figure 5 is the simulation result got from the existing concurrency control in a distributed database. The three straight lines represent the three participants database management system projected from coordinator database management system. The lines have

ISSN 2229-5518

intercept which signify incorrect and inconsistent operations of a distributed database management system.

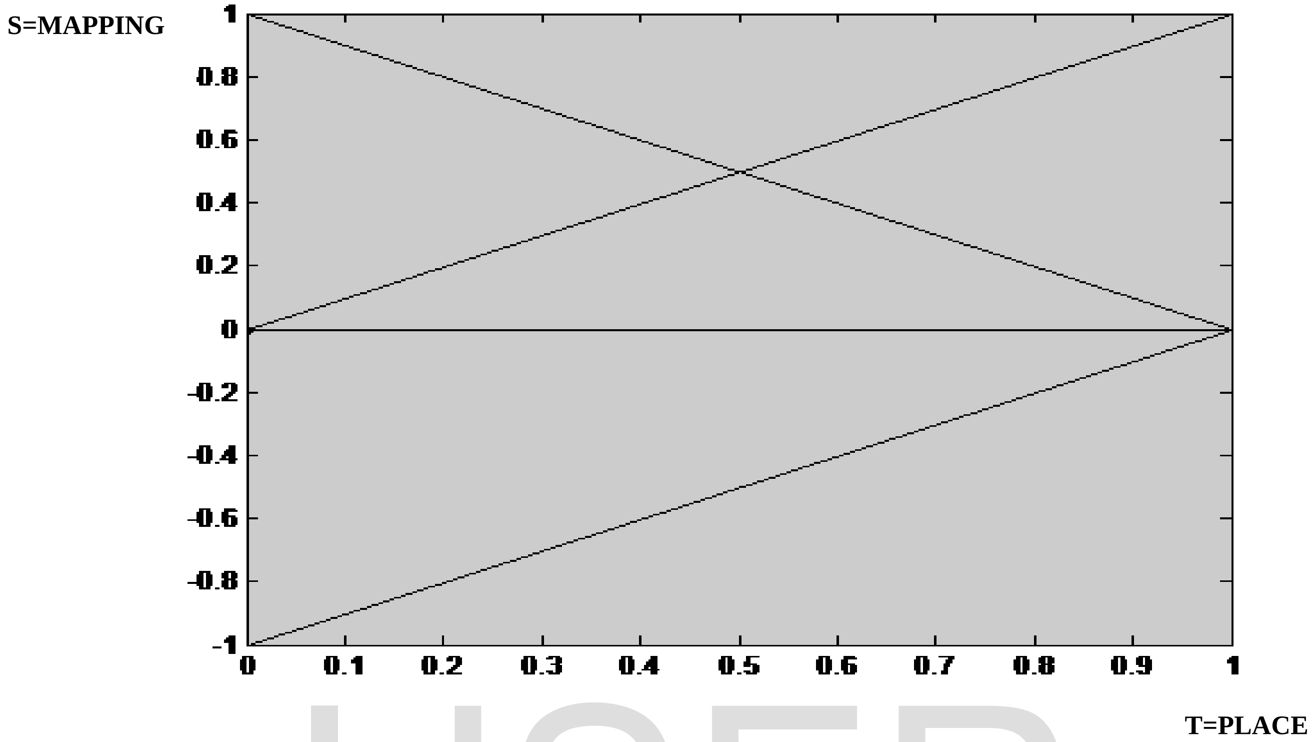


Figure 5: Graph of existing concurrency control in a distributed database

Table 2: Results For the Improved Two way Forward Direction

M_0	1 0 0 0 0 0 0
M_1	0 1 1 1 0 0 0
M_2	0 0 1 1 1 0 0
M_3	0 0 0 1 1 1 0
M_4	0 0 0 0 1 1 1
M_5	0 0 0 0 0 0 0
M_6	0 0 0 0 0 0 0
M_7	0 0 0 0 0 0 0

The graph in figure 6 is the simulation result got from the improved Two- way forward direction of concurrency control in a distributed database. The three straight lines represent the three participants database management system projected from coordinator database management system. The lines have no intercept which signify correct operations of a distributed database.

ISSN 2229-5518

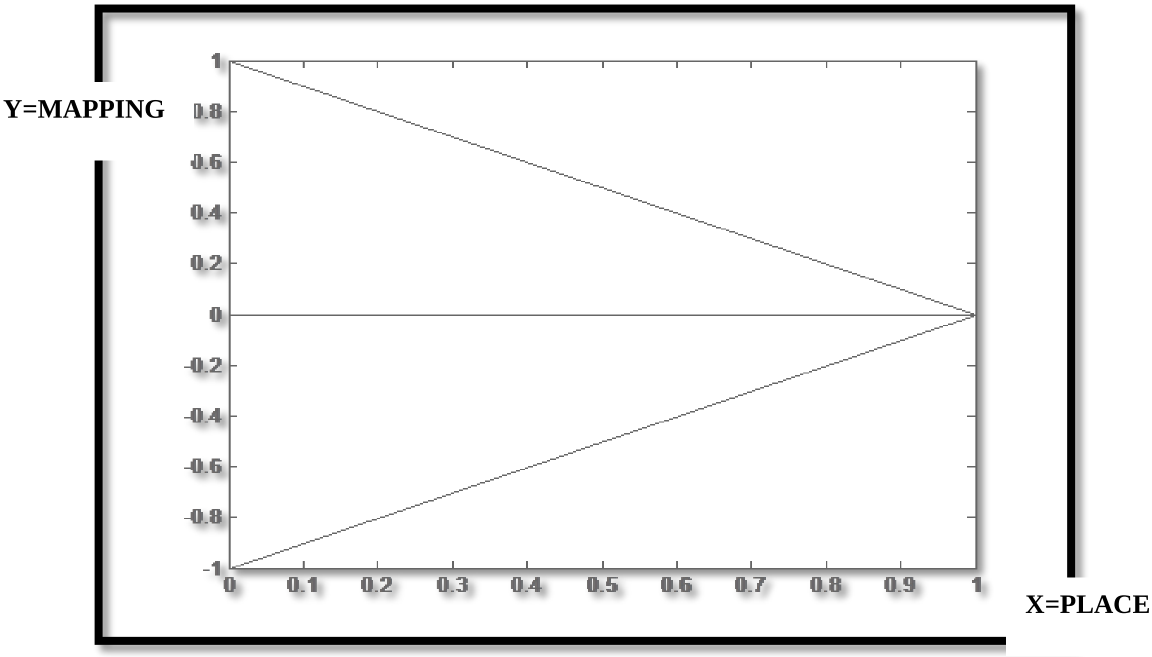


Fig. 6. Plotted graph for the Improved TWO-WAY Forward direction

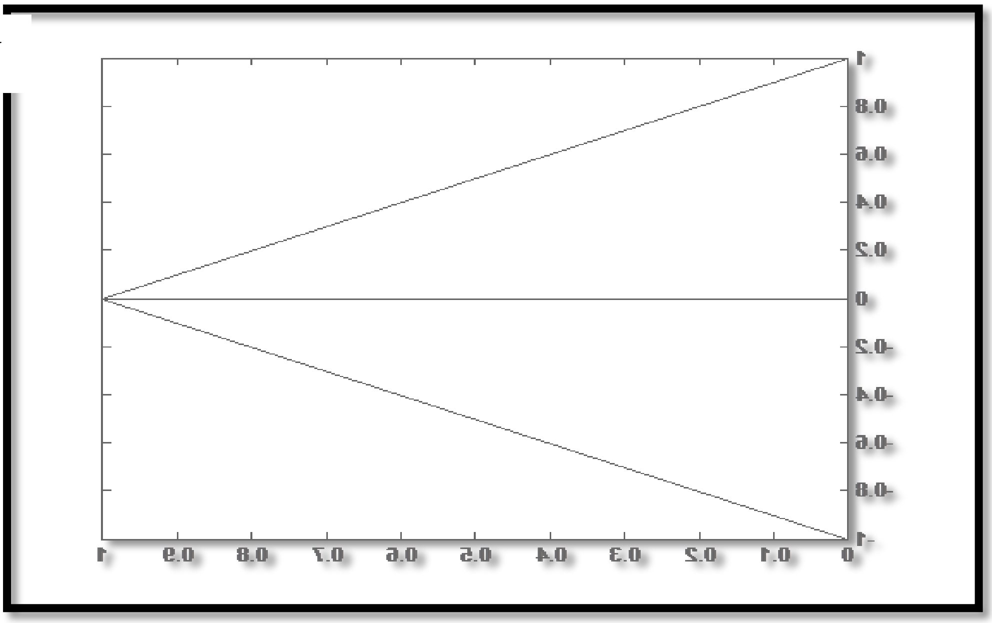
Table 3. Results of Improved Two way Backward Direction

M_o	1 0 0 0
M_a	0 1 0 0
M_b	0 0 1 0
M_g	0 0 0 1
M_h	0 0 0 0

3.2 Improved TWO-WAY backward direction

The graph in figure 7 is the simulation result got from the improved Two- way backward direction of concurrency control in a distributed database. The three straight lines represent the three participants database management system projected from coordinator database management system. The lines have no intercept which signify correct operations of a distributed database

Y=MAPPING



X=PLACE

Fig.7. Plotted graph for the Improved TWO-WAY Backward direction f

The summary of the improvement is presented in table 4,

Assumption: Let '1' represent improvement and '0' represent no improvement

Table 4: Requirements on improvement on concurrency control

Requirement(R)	Improvement mark(IM)	% improvement (IFR)
Data collection	1	
Data duplication	0	
Data distribution	1	
Distributed serializability	1	
Data consistent	1	
Complexity	0	
Cost	0	
Recoverability	1	
Availability	1	
∑R=9	∑IM =6	66.7%

% improvement (FR) = ((∑IM ⁄ ∑R)*100⁄1) =(6/9*100/1)=66.7%

4 CONCLUSION

The general purpose of concurrency control is to provide rules, methods, design methodology and theories to maintain the consistency of components operating concurrently while interacting, and thus, the consistency and correctness of the distributed database system**.** This work has studied on the improvement of concurrency control in a distributed database. From the analysis and results, it can be seen that, the concurrency control in a distributed database has been improved by 66.7%. The research results indicated that improvements on concurrency control in a distributed database is achievable.

REFERRENCES

[1] Anderson P.H.G: Use of a PC printer port for control and data acquisition, technology interface. The Electronic journal for Engineering Technology fall, 1996.
[2] Jonathan P.B and Tim, J.G; Distributed operating system. 1990
[3] Marta, m; Vanessa, B; Alexandra A.B and Lima L; Distributed database Research at coppe/ UFRJ. Journal of information and data management, Vol 2, No.2, 2011 pp 1
[4] Bernstein, P.A; Hadzilacos, V, and Godman, N. Concurrency control and Recovery in Database System. Addison Wesley, 1987
[5] Mohammed K.N and DrA.S, Data count driven concurrency control scheme with performance gain in mobile Environments: Journal of Emerging Trends in computing and information. Vol 2, No2, 2011, pp 106-112
[6] Iso/Dis/7498; Information processing system- open Interconnection- Basic Reference model, International standard organisation, Geneva, Switzerland 1982.
[7]Stonebraker, M; Concurrency control and consistency of multiples copies of data of distributed INGRES. IEEE Transactions on software Engineering, No 5, 1979 pp 188-194
[8] Bernstein, P: A sophisticated Introduction to distributed database concurrency control. Proceedings of the International conference on very large databases, No 8, 1981, pp 62-76
[9] Ma, Z; Universe status monitoring. Internal note 1526, Department of computer science, University College, London, uk. 1984.Acm 11 1968 pp 341-346
[10] Lamport, L.I; Time, Clocks, and ordering of events in a distributed System communication of the Acm. 1978
[11] Andrea b; System modelling with Petri nets: System Reliability Assessment, Kluwer p.c.,pp 102-143, 1990
[12] Tamer M.O and Patric V, Principles of Distributed Database System, Springer, 2011
[13] Shabnam A and ToorajS.B., Distributed scheduling for mesh Networks with chain Topology. International Journal of Advanced Research in Engineering and Applied Sciences, vol.2, No.10, 2013, ISSN :2278-6252
[14]Cisco system .Inc. "Ethernet, Hubs, switches, and the Evolving Factory Network",www.cisco.com/go/offices, pp.1 – 2. 1992-2002
[15] Hruz B., Zhou M.C., Modeling and Control of Discrete-Event Dynamic Systems: With Petri Nets and Other Tools,
[16] Murata T., Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE, Vol.77, No.4, 1989, pp. 541-580.
[17] Cassandras C.G., Lafortune S., Introduction to Discrete Event Systems, Kluwer Academic Publishers, 1999. Springer, 2007.