Models of 2PL Algorithms with Timestamp Ordering for Distributed Transactions Concurrency Control

Svetlana Zhelyazkova Vasileva, Aleksandar Petrov Milev

Abstract—In this paper simulation models of two-phase locking in distributed database systems are presented. A mechanism of timestamp ordering (“wait – die” method) is embedded in the modeling algorithms to prevent deadlocks. The results of running model of Centralized, Distributed and Primary copy Two-phase locking algorithms are gathered and analyzed and represented. The main characteristics of transaction processing in distributed database management systems such as throughput, response time and probability service are given.

Index Terms — Simulation models, GPSS transactions, Distributed transactions, Two-phase locking, Timestamp ordering.

I. INTRODUCTION

The concurrency control of the simultaneously running transactions which are intended to process the common data is one of the most problems in database management systems (DBMS). Solving this problem becomes too hard in distributed database management systems (DDBMS) where the reliability of the system is increasing as an impact of data fragmentation and replication.

The paper researches the developed transaction concurrency control algorithms in DDBMS [1], [2], [3]. Main topic of our research is concurrency control algorithms for distributed transactions based on the method of two-phase locking (2PL) in the distributed databases (DDB): Centralized 2PL, Primary Copy 2PL and Distributed 2PL. Additional methods for deadlock finding and resolving or avoidance are needed when DBMS with 2PL algorithms is considered. This paper depicts the algorithms modeling 2PL in DDBMS where a mechanism of timestamp (TS) ordering is using to avoid the deadlocks and its specific version “wait-die”.

II. Timestamp Ordering Algorithms providing protection of deadlocks Figures and Tables

There are some advantages of the method for protection of deadlocks - timestamp ordering transactions: deadlocks are not possible ([1], [6], [7] and [8]) and it is easy to be implemented. Moreover the method can’t afford cycle restart of discarded transaction because of the time marker.

In addition to this every transaction will be older one during the processing time and it won’t be restarted for resource contention [1].

There are two implementation of timestamp ordering deadlocks protection [1] and [6]:

A. Method „wait – die”

When a resource conflict occurs at this method and if the transaction Ti is “older” than transaction Tj, which holds the element lock (TS(Ti)<TS(Tj)), then Ti waits its release. If Ti is “younger” then Tj (TS(Ti) > TS(Tj)), transaction Ti restarts. The algorithm of lock manager handling transaction Ti, and its requesting locking of the data element x is shown on fig.1.

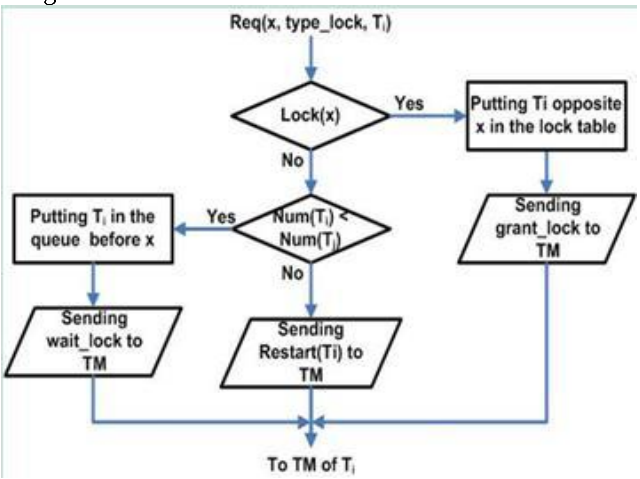

FIG. 1 method „wait – die” locking request handling of the element x by transaction Ti

B. Method „wound - wait”

This method is contrary to the previous one. If the candidate transaction Ti is “older” than used one transaction Tj [1] and [6] transaction Ti “wounds” Tj when a resource conflict occurs. It could lead to restart of Tj. In that case the transaction Tj “survives” and there is not a rollback. If Ti is “younger” than Tj, then Ti is permitted to be set in state waiting for locking. A scheme of algorithm for timestamp ordering by the method „wound – wait” is shown on the fig. 2, in which the lock manager handles locking request of the element x by the transaction Ti.

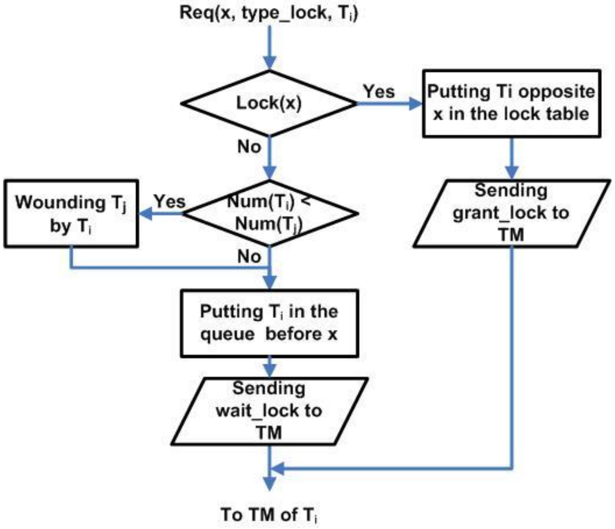

FIG. 2 method „wound - wait locking request handling of the element x by transaction Ti

Manuscript received September, 2013.

Svetlana Vasileva, College - Dobrich, Konstantin Preslavsky University of Shumen, Dobrich, Bulgaria.

Aleksandar Milev, Faculty of Mathematics and Informatics, Konstantin Preslavsky University of Shumen, Shumen, Bulgaria.

If the method „wound – wait" is used and a half job is done in the local databases by the "younger" transaction it could be restarted. This is very difficult problem for implementation in distributed system. The method "wait – die" is more efficient for distributed database management system.

This work presents simulation models of Centralized 2PL, Primary Copy 2PL and Distributed 2PL with embedded mechanism of timestamp ordering by the "wait – die" method.

III. Basic Elements of the GPSS Simulation Models

The suggested Centralized 2PL, Primary copy 2PL and Distributed 2PL algorithms are investigated with the help of the simulation environment GPSS World Personal Version. The presented simulation models use generated streams of transactions which simulate global transactions in DDB systems. Such simulation models are presented in [5]. Here we will emphasize the use of timestamps in 2PL to avoid DL. They are all in parallel streams and their intensity □ is given in tr per sec (number of transactions per second).

The structural schemes of modeling algorithms for distributed transactions management in „embedding" of timestamps in Centralized, Primary Copy and Distributed 2PL algorithm in DDB are shown correspondingly in fig. 3, fig. 4 and fig. 5.

A. Parameters of the GPSS Transactions

P1 – Number of transaction. The value is a sum of System Numeric Attribute MP2 (The subtraction between the relative model time and the content of the second parameter of GPSS transaction) and the number of the site;

P2 – Number of the site, where the transaction is generated. The value is a number from 1 to <number of stream transactions>;

P$Nel – Length of the modeled transaction. The value of that parameter in the constructed models is 1 or 2 chosen by probability defined by the function FN$BrEl respectively 0.30 and 0.70. It is supposed that long transactions get in the system more frequently then short ones in that model;

P$El1 – Number of the first element, which the generated transaction will read or write. The value is a random number and is uniformly distributed in the interval [1, NumEl];

P$El2 – Number of the second element, which will be processed by the generated transaction;

P3 – Type of the requested lock for the first element, which will be processed by the generated transaction;

P4 – Type of the requested lock for the accessed second element;

P5 – Value 0, if the transaction is in 1st phase – occupation of the locks and value 1, if the transaction finishes its work and has to release the locks;

P$Prim1 – Number of the primary site of the first element, which the generated transaction will read or write (in the Primary Copy 2PL model – fig. 4);

P$Prim2 – Number of the primary site of the second element, which will be processed by the generated transaction (in the Primary Copy 2PL model – fig. 4);

P$CHTN1 and P$CHTS1 – In the situation when P3 = 1 the transaction only " reads" the element with number P$El1. This is possible if the element is not free and the lock is permissible. According to these facts the parameters P$CHTN1, P$CHTS1 record accept respectively the number of the previous transaction which had blocked the element and the number of the site generated it;

P$CHTN2 and P$CHTS2 – In the situation when P4 = 1 the transaction only " reads " the element with number P$El2. This is possible if the element is not free and the lock is permissible. According to these facts the parameters P$CHTN2, P$CHTS2 record accept respectively the number of the previous transaction which had blocked the element and the number of the site generated it;

P6 and P7 – In them there are correspondingly recorded the number of the site, where it is the nearest copy of the data element and the number of the site, where it is the second replica of the first data element, processed by transaction. Correspondingly in parameters P8 and P9 we have the nearest copy of the second data element and the number of the recorded site, where it is the second replica of the second data element;

P11 – number of the user' s list where the corresponding sub-transaction waits for the release of the copy data element (in the Distributed 2PL model).

B. Basic Steps in the Suggested Modeling Algorithms of Centralized 2PL and Primary Copy 2PL with TS

The basic steps in the algorithm modeling Primary Copy 2PL with TS are:

When the transaction TP2P1 comes in the transaction manager TMP2 its length is checked (1 or 2 data elements will be processed) - operation 1 on fig. 4 and the transaction is prepared to be split (operations 8 on fig. 4).

With the operations 9 values of the parameters of the sub-transactions are acquired – the numbers of the data managers DMP6, (DMP7), (DMP8 and DMP9), where the sub-transactions TP2,P6P1, (TP2,P7P1), (TP2,P8P1 and TP2,P9P1) have to execute the operations of reading/recording of the copies of data elements El1 and El2.

After the primary processing in the transaction coordinator TCP2 the requests for locking El1 and El2 are transmitted through the net to the corresponding primary lock managers LMprim1 and LMprim2 (operations 2 and 5 on fig. 4).

LMprim1 and LMprim2 check in the lock tables LTprim1 and LTprim2 if the lock of El1 and El2 is allowed (operations 3 and 6 on fig. 4). If the lock of El1 (and El2) is allowed, the corresponding record is put opposite the number of the element in LTprim1 (and LTprim2).

The transaction receives confirmation messages about the lock of El1 (operation 4) and if two data elements are being processed, TMP2, through the transaction coordinator TCP2 sends the request for lock of El2 to LMprim2 (operation 5).

If the lock of the corresponding element is not possible, the number of the transaction is check if it is smaller than the number of the transaction which has put the lock:

- if the sub-transaction is not going to continue and is not going to restart, it waits the release of the element in user chain, whose number is the number of the element;
- if the sub-transaction has not received the lock of the element it restarts (operation 4/operation 7 is a restart operation). After it has arrived in TMP2, the restarted lock request (operation

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

16) is transmitted to LMprim1/LMprim2 (the repeated (successful) attempt for lock element 1/element 2 is presented with operations 16, 17 and 18).

Transaction which has finished with the operation read/write releases the element in LTprim1 (and LTprim2) – operations 21 and 22 on fig. 4. The requests for release of the lock of the elements are transmitted to the corresponding primary lock manager with operations 19 and 20.

After the release of the lock of an element, the transaction which is first in the waiting list heads to the lock manager. If it is a group of sub-transactions then they receive a shared lock of the element.

Receiving a confirmation for a lock of the elements of the GPSS transaction being allowed, a modeling global transaction splits. After that the sub-transactions are transmitted through the net to the data managers for executing the read/write operations (operations 10 and 11 on fig. 4).

The sub-transactions of TP1P2 execute read/write in local databases LDBP6, LDBP7, LDBP8 and LDBP9 with the corresponding replicas of El1 and El2 (operations 12 on figure. 4).

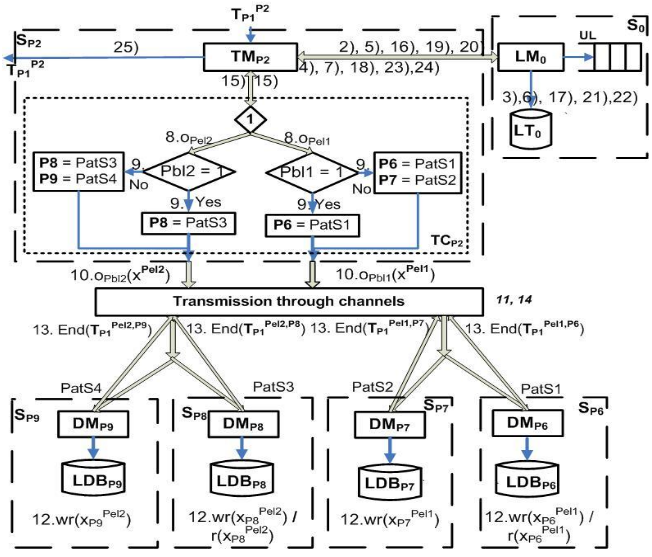

FIG. 3 Scheme of centralized 2PL with timestamps

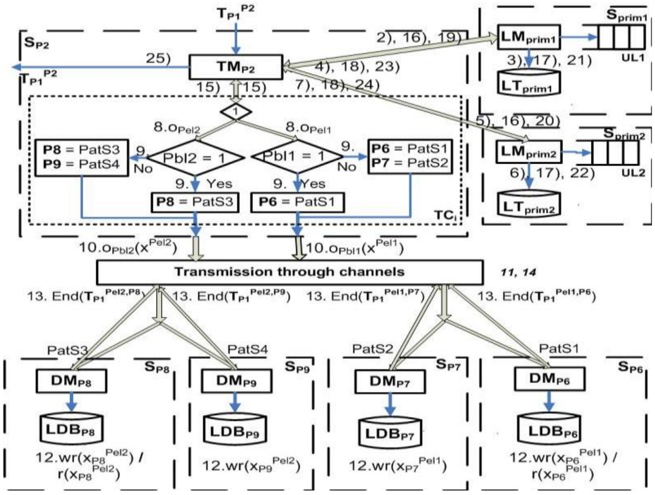

FIG. 4 SCHEME OF PRIMARY COPY 2PL WITH TIMESTAMPS

After that they are transmitted to the transaction manager TMP2 (operations 13 and 14). If a transaction renews a data element, the sub-transactions recording the corresponding copies wait for each other and get united (operations 15), before a request for release of the lock of the element is sent to LMprim1 (and LMprim2).

Transaction TP1P2 quits the system (operation 25 on fig. 4) as soon as sub-transactions TP1Pel1 and TP1Pel2 finish their process (modeled with operations 23 and 24 on fig. 4).

The transfer through the network to primary lock managers LMprim1 (and LMprim2) and to the sites-executors, where are the data managers DM is simulated with retention.

The lock manager (LM0) and the lock table (LT0) are only one when centralized 2PL (fig. 3) is considered. All operations 2, 3, 4, 5, 6, 7, 16, 17, 18, 19, 20, 21, 22, 23 and 24 are referred to the central node where central lock manager LM0 is situated.

C. Basic Steps in the Suggested Modeling Algorithm of Distributed 2PL with TS

When the transaction TP2P1 comes into transaction manager TMP2 its length is checked (1 or 2 data elements will be processed) – operation 1 (fig. 5) and the GPSS transaction is repaired to splitting – operations 2. With operations 3 the values of the sub-transactions parameters are converted – the numbers of the lock managers LMP6, (LMP7), (LMP8 and LMP9), where the sub-transactions TP2,P6P1, (TP2,P7P1), (TP2,P8P1 and TP2,P9P1) have to receive the lock for read/write replicas of data elements El1 and El2. In the common case it is executed transferring of requests for locking data elements replicas through network to the executor nodes (operations 4 and 10).

In the executor nodes SP6, (SP7) the lock managers LMP6, (LMP7) check in the lock tables LTP6, (LTP7) with operation 5 in fig. 5 the possibility for presenting the locking of the replicas of the element El1 to the sub-transactions TP2,P6P1, (TP2,P7P1). The decision for presenting a locking of an element is accepted by the lock managers LM in conformity with the table of the compatibility of the locking shown in [1] and [2]. If the locking is allowed (operations-messages 6) the sub-transactions are split and their heirs (with operations 7) come back in the node-initiator for transmitting the confirmation for the locking of El1 before the sub-transactions TP2,P8P1 and TP2,P9P1 so that it would be possible that the global transaction continues its first “expanding” phase, and “the parents” TP2,P6P1, (TP2,P7P1) continue to execute the operations read/write (operations 9) on the replicas of the element El1 in the local databases LBDP6, (LBDP7). In most cases the locking of the copies of the element El1 is submitted.

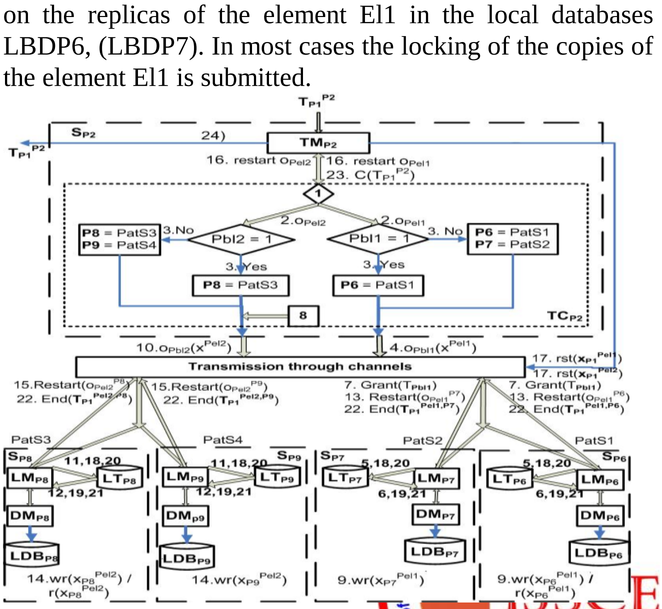

Fig. 4 A Frame Scheme Of Global Transaction Execution By The Algorithm Of Distributed 2pl With Timestamps (In Two Data Elements Update)

Receiving the confirmation for locking of the element El1 (operation 8 on fig. 5) the transaction manager TP2P1 - TMP2 transmits the sub-transaction that processes the element El2 to the transaction coordinator TCP2. There is a great probability that (from the range of 0,8) the transaction updates El2, and therefore the sub-transaction is split (operation 10) and the sub-transactions TP2,P8P1 and TP2,P9P1 are transmitted through the channels of the network the corresponding lock managers LMP8 and LMP9. LMP8 and LMP9 through the lock tables check the possibility for taking the replicas of the element El2 (operations 11). Submitting the locking (operations 12) the sub-transactions continue to the data managers DMP8 and DMP9 for execution of operations read/write of the element El2 (operations 14 on fig. 5).

If the locking is impossible it is checked if the number of the sub-transaction is smaller than the number of the sub-transaction that has put the locking: if the sub-transaction does not continue and is not going to then it queues before the replica of El2 in the lock tables LTP8 and LTP9. The waiting is modeled by user chains with number P11 (of the parameter P11 of each of the sub-transactions TP2,P6P1, (TP2,P7P1), TP2,P8P1 and TP2,P9P1 is given a value before it enters the corresponding lock manager); if the sub-transaction has not received the locking and is not going to wait for its submission, it is restarted (operations 13 / operations 15 in fig. 5). After arriving in TMP2 (operation 16), the restarted transaction is transmitted again to the lock managers (operations 17 in fig. 5). The second (successful) attempt for locking the element 1 / element 2 is shown with the operations 18 and 19. The corresponding lock managers LM put the record for the element lock in the lock tables. After the execution of operations read/write (operations 9 and 14), the locking of the replicas of the elements is released (operations 20 – request and operations 21 – confirmation for removing the locking).

The confirmation about the finish of reading/writing of the element El2 is transmitted to the transaction manager TMP2 (operation 22). The sub-transactions TP2,P8P1 and TP2,P9P1, and before that TP2,P6P1 and TP2,P7P1 (if the element El1 has been updated) are merged in the sub-transactions processing El2 and El1 respectively. The confirmation about the end of the corresponding sub-transactions is transmitted to the transaction manager TMP2. The sub-transactions that processed El1 and the element El2 respectively, and the parent-transaction, waiting for them in the transaction manager TMP2 are merged (operation 23). After gathering the necessary statistics about these which have finished their work GPSS transactions, they leave the system (operation 24).

IV. ANALYSIS OF SIMULATION RESULTS

A simulation of the algorithm for centralized 2PL with TS for equal intensities of ingress flows is performed and the results are given on figure 6. The shown data present system behavior through the period of conducted experiments in seconds. The dense line on the graph shows the results for 6 flows with average intensity 4.17 tr/s for everyone of them. It is the situation for minimum system loading. The dotted line on the graph shows the results for 6 flows with intensity 8.33 tr/s which is considered as average system loading. The third line present the values for 6 flows with average intensity 16.67 tr/s which is the maximum system loading.

Throughput (TP) of the system can be calculated as number requests handled per time unit by using the formula:

TP = <number fixed transactions for the time Tn> / Tn.

The results of the throughput given on figure 6 show that the throughput of centralized 2PL wit TS model is increasing for all three types of system loading when the system running time is increased at the same time. A stable mode is appeared after a period of time where the throughput gets constant value and is equal to the whole intensity of the ingress flow. This is the perfect characteristic.

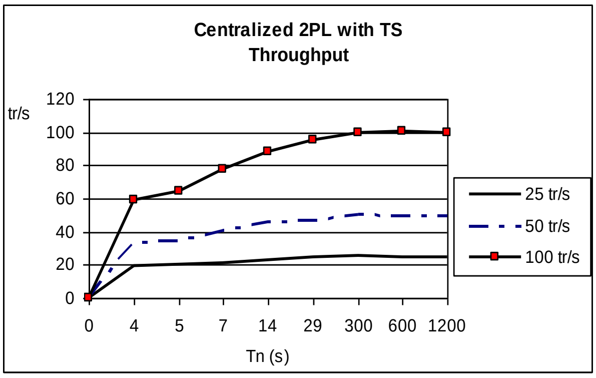

FIG. 6 Throughput in the model of Centralized 2PL

The throughput simulation results for primary copy 2PL system are given in the figure 7. All data are for the same intensity flows as those given on figure 6.

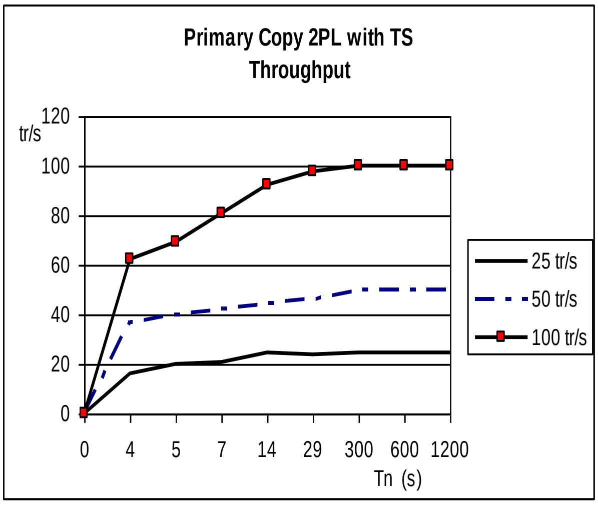

FIG. 7 Throughput in the model of Primary copy 2PL

It could be seen that the graphs shown in figure 7 have the same behavior as those given on figure 6. There is a slight difference at the moment of stable mode where the value has not change. It could be seen that stable mode for simulations of Primary copy 2PL occurs a little bit earlier, for example 29 sec.

The throughput of Distribute 2PL for the all three modes of loading is given on the figure 8.

The lines marked with a little figure of square on the fig. 6, fig. 7 and fig. 8 show that throughputs of the centralized 2PL, primary copy 2PL and distributed 2PL system for maximum loading are very similar. Furthermore, they are almost without deflection from the throughput for system managing database given at [4]. It is possible to claim that results for simulated centralized 2PL, primary copy 2PL and distributed 2PL models for different loadings of DDBMS are trustworthy. It could be said that 2PL algorithms with timestamp mechanism are effectively enough for concurrency control in the DDBMS.

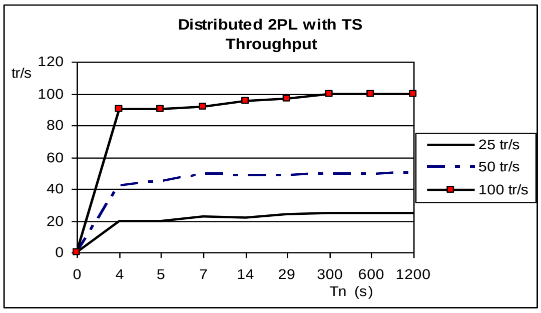

FIG. 8 Throughput in the model of Distributed 2PL with Timestamps mechanism

The lines marked with a little figure of square on the fig. 6, fig. 7 and fig. 8 show that throughputs of the centralized 2PL, primary copy 2PL and distributed 2PL system for maximum loading are very similar. Furthermore, they are almost without deflection from the throughput for system managing database given at [4]. It is possible to claim that results for simulated centralized 2PL, primary copy 2PL and distributed 2PL models for different loadings of DDBMS are trustworthy. It could be said that 2PL algorithms with timestamp mechanism are effectively enough for concurrency control in the DDBMS.

A comparison of the results for throughput of 2PL algorithms in DDBMS is given on fig. 9. There are shown the simulations of the developed timestamp model algorithm for ordering in DDBMS considering the same input flow intensity and the same data element replicas distributions.

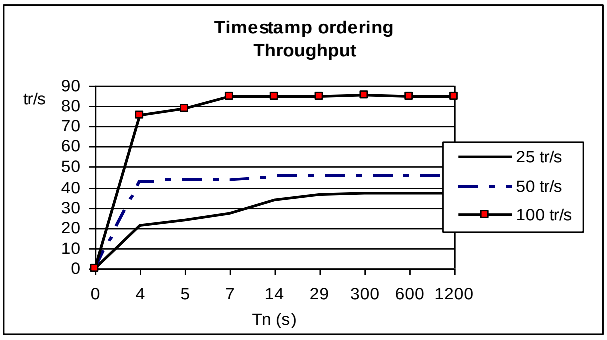

FIG. 9 Throughput in the model of distributed timestamp ordering

The frequency distribution of response time (RT) for the three models (centralized 2PL, primary copy 2PL and distributed 2PL) is given on fig. 10, fig. 11 and fig. 12. The results are gotten for a long period of monitoring time for about 28800 model units and input intensity 100 tr/s. The graphs are generated by GPSS World according to the tables of frequency distribution, which is constructed automatically for every relevant simulation. The number of GPSS transactions is given on axis Y vs. the relevant process time for the proper interval on axis X is represented by the three histograms.

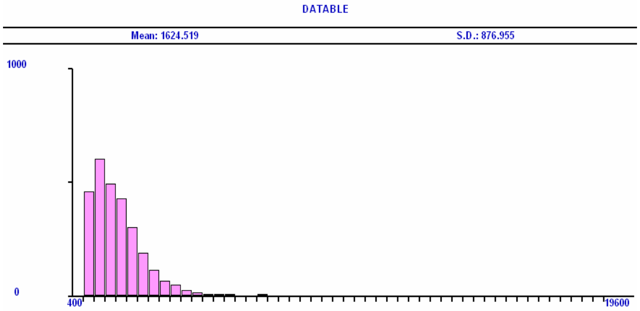

FIG. 10 Frequency distribution of transaction RT in centralized 2PL model for input intensity 100 tr/s. Time intervals on axis X have length 400 ms

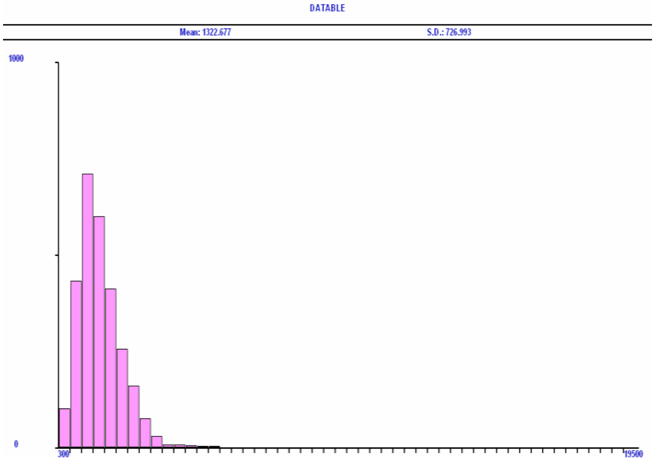

FIG. 11 Frequency distribution of transaction RT in primary copy 2PL model for input intensity 100 tr/s. Time intervals on axis X have length 400 ms

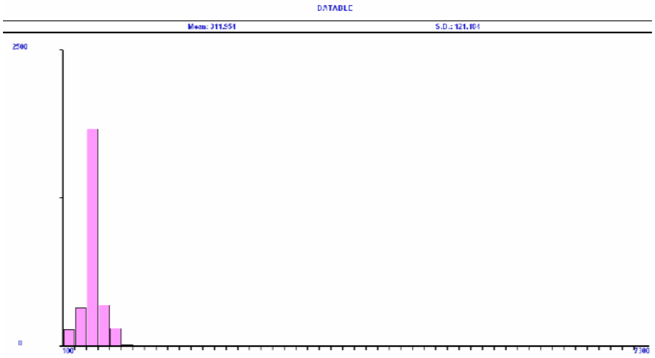

FIG. 12 Frequency distribution of transaction RT in distributed 2PL model for input intensity 100 tr/s. Time intervals on axis X have length 200 ms

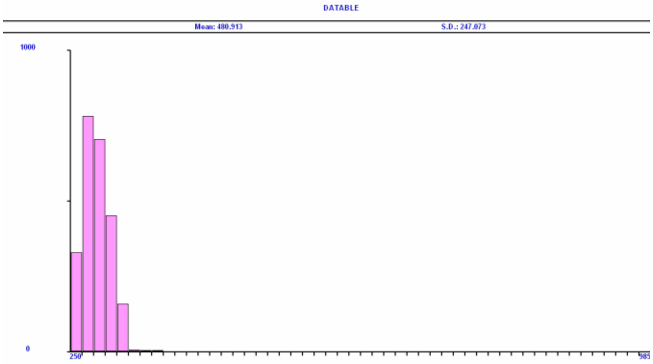

FIG. 13 Frequency distribution of transaction RT in Distributed timestamp ordering model for input intensity 100 tr/s

Fig. 13 shows the diagram of frequency distribution of response time (in case of: summary input intensity 100 tr/s, modeling time 28800 model units, i.e. just before stationary regime). The diagram in fig. 13 corresponds to the stereotyped graphic of response time, shown in [4].

V. Conclusions

Simulation models of Two-phase locking in DDBMS for centralized 2PL, for primary copy 2PL and for distributed 2PL with embedded mechanism of timestamp by “wait – die” method are developed with deadlock avoidance of distributed transactions. The models are limited for the number of input flows and length of processed transactions because of the limitation of other similar models compared to them.

It is necessary to develop simulation models of 2PL protocols with much complexity by the meaning of the number of element replicas and transaction length in number of elements.

The models of 2PL protocols with timestamp show results similar to the graphics of the throughput and frequency distribution of transaction RT in [4].

The advantages of embedded timestamp mechanism are experimentally proved about the algorithm of two-phase locking in DDBMS.

The efficiency of the 2PL method in systems full with conflicts is proven.

More results and statistics of simulations are needed to evaluate the effectiveness of 2PL algorithms based on the criteria of throughput, response time and service probability.

REFERENCES

[1] T. Connolly, C. Begg, Database systems: Addison-Wesley, 2002.

[2] C. J. Date, Chris J. Introduction to Database Systems. 7th edn. Reading, MA: Addison-Wesley, 2000

[3] N. Krivokapic, A. Kemper, E. Gudes, Deadlock detection in distributed database systems: A new algorithm and a comparative performance analysis [Online]. Available: http://masters.donntu.edu.ua/2005/fvti/kovalyova/library/d1.pdf.

[4] TPC BenchmarkTMC. Revision 5.11 February 2010. [Online]. Available: http://www.tpc.org/tpcc/spec.

[5] S. Z. Vasileva, A. P. Milev, “Simulation Models of Two-Phase Locking of Distributed transactions” International Conference on Computer Systems and Technologies: V.12-1-V.12-6, ACM, New York, 2008.

[6] I. Bodyagin. “Deadlocks. Что такое взаимоблокировки и как с ними бороться”, RSDN Magazine #5, 2003, [Online]. Available: http://sasynok.narod.ru /index.htm?omvs.htm.

[7] Simeonov, S., Ts. Tswetanov Network Flow Security Baselining IT-Incidents Management IT-Forensics - IMF, pp. 143-156, 2008

[8] Simeonova, N, S. Simeonov, A. Iliev Concepts for Creating Operating Systems with Special Purpose, UNITECH’12, Gabrovo, 16-17 November 2012, pp 377-381, ISSN 1313-230X.

Svetlana Vasileva: PhD of Informatics. Works on simulation modeling of systems and its application in student education..

Her research interests include distributed databases and distributed transaction concurrency control.

Education details: Master of Science in Engineering (Saint Petersburg Electro-technical University "LETI"). Membership: John Atanasoff Society of Automatics and Informatics, Union of scientists in Bulgaria.

Prof. Aleksandar Milev PhD of Informatics Works on simulation of systems and its application in networks. His research interests include communication systems, wireless communication technologies, operation systems and network security.

Education details: Master of Science in Engineering (Military University "V. Levski"). Membership: John Atanasoff Society of Automatics and Informatics, Union of scientists in Bulgaria.