SOME GPSS OPPORTUNITIES FOR MODELING OF TIMESTAMP ORDERING IN DDBMS AND SIMULATION INVESTIGATIONS

Svetlana Vasileva
School of Computer Science
Varna University of Management
13 A Oborishte Str, 9000, Varna, Bulgaria
E-mail: svetlanaeli@abv.bg

KEYWORDS
Simulation models, distributed transactions, distributed timestamp ordering, distributed 2PL, voting.

ABSTRACT

This paper considers some opportunities of the system for simulation modelling GPSS World Personal Version for simulating algorithms for transaction concurrency control (CC) in Distributed database management systems (DDBMS). Models of Timestamp ordering (TSO) algorithm and Two-version Two-phase locking (2V2PL) in DDBMS are presented. Both approaches – two version data and timestamps (and others) are used in database management systems for avoiding transaction deadlock. Method of TSO and method of 2V2PL are still not investigated enough. However, the use of timestamps makes the CC algorithms more complex due to the restarting of the transactions from service and the additional waiting for processing. Therefore results of the implementation the simulating algorithms are showed in comparative view.

INTRODUCTION

Simulation modeling is becoming more widespread and used as system-and an extremely valuable link in the process of decision-making, so use with other software systems for making a decision in the systems to support decision making. Nowadays the systems for modeling is a powerful analytical tool in which they integrated the all newest information technologies, for the purpose of constructing models and interpretations of simulating results, multimedia and video, supporting animation in real time, object-oriented programing, Internet-solutions, etc.

The paper considers one of the most famous and universal systems for simulation modeling – GPSS World, and especially its opportunities for simulating concurrency control algorithms in distributed database systems and making simulation investigations on the implementation of these algorithms. Simulation modeling allows us to explore queuing systems to different types of input flows and intensities of arrival of requests at the entrances of systems and determine the main features of the same. All these and other specific characteristics of GPSS World make it possible to develop simulation models of a variety of simple and complex systems running on different algorithms and to carry out various types of research. Such studies are an analysis of concurrency control (CC) of distributed transactions in distributed database management systems as represented in (Culciar and Vasileva 2015). In the cited work are presented some simulation studies of the implementation of Centralized Two-Phase Locking (2PL) in Distributed database management systems (DDBMS). The paper presents a model of another approach for transaction CC in DDBMS named Timestamp ordering (Connoly and Begg 2002) and some simulation results also.

ADVANTAGES OF SIMULATION MODELING

Among the advantages of simulation (Tomashevskii and Zhdanova 2003) and others, we could cite the following: present only essential for understanding the behavior parts; the model can be built before the real system, with a much less resources; various parameters may change during the modeling; the model takes into account the random nature of the processes in the real system; to conduct modeling are not required deep knowledge of computational mathematics. This allows applying simulation as a universal approach to decision-making under conditions of uncertainty in the models with thinking of factors that are difficult to be formalized, but also to use the basic principles of the systematic approach to solve practical problems.

The problems that arise in simulation modeling (Tomashevskii and Zhdanova 2003; etc.), are: modeling results are always approximated; optimization of the modeled system is possible, but difficult and requires great computing power; need for validation and verification of the model and actual construction is complicated; needed is a specialist in modeling; there is always a risk to build an inadequate model. One of the purposes of the work is to try to show some GPSS World opportunities that could help us to make models, which we can trust when making a decision. These opportunities are shown on the example of the GPSS model of distributed transaction Timestamp ordering in distributed database management systems.

TIMESTAMP ORDERING AND DDBMS

One of the major problems in database management systems is concurrency control. There are mainly 3

Proceedings 30th European Conference on Modelling and Simulation ©ECMS Thorsten Claus, Frank Herrmann, Michael Manitz, Oliver Rose (Editors)
ISBN: 978-0-9932440-2-5 / ISBN: 978-0-9932440-3-2 (CD)

methods for CC: protocols using 2PL, protocols using timestamping and validation check up. The first two methods have monoversion and multiversion variations. The variations of CC protocols are not only these. The 2PL protocol in DDBMS can be on Centralized 2PL, Distributed 2PL, Primary copy 2PL and Voting 2PL. (Connolly and Begg 2002). And everyone ot these protocols could have monoversion, two-version and multiversion variant as presented ones in (Chardin 2005; Date 2000; etc.). Every one of the new variant of the CC protocol is needed in its study, validation and verification. That is the reason to use the simulation modeling.

In the centralized database systems the task of timestamp (TS) protocols is the global alignment of transactions so that the older transactions (which have smaller TS) in case of conflict to receive priority (Connolly and Begg 2002; Date 2000; Tanenbaum and Steen 2007; etc.). The general approach in the DDBMS is concatenation of local timestamp with the unique identifier of the node (<*local TS*>, <*node identifier*>). (Connolly and Begg 2002; etc.) The node identifier value has a smaller weight coefficient which guarantees the order of the events in accordance with the moment of their appearance.

The serving of global transaction in the distributed timestamp ordering (DTO) modeling algorithm is performed according to the algorithm of timestamps, shown in fig. 1. The schema in fig. 1 demonstrates TO algorithm in a summary, described in (Connolly and Begg 2002; etc.). The algorithm uses Tomas rules (Thomas 1979) according to which:
- The duration of service transactions has the exponential distribution with parameter m;
- To each transaction T is assigned timestamp, denoting the time of its coming into the system and the number of the site-generator. When a transaction read/write data element, it records its TS in it.
- If a transaction T wants to update data element x:

If TS(T) < readTS(x), then restart(T);
If TS(T) < writeTS(x), then ignore(T);
If TS(T) > writeTS(x), then execute(T).
- If a transaction wants to read data element x:

If TS(T) < writeTS(x), then restart (T);
If TS(T) > writeTS(x), then execute(T).

GPSS WORLD AND DISTRIBUTED TSO MODELING

GPSS blocks used in the model of Distributed TSO

In the considered GPSS model of timestamping of distributed transactions we use the following GPSS blocks (Minuteman 2010):
- Blocks for generating and cancelling of transactions: GENERATE – a main block of transactions inputting in the model; TERMINATE – to output transactions from the model.
- Blocks for management of the modeling time: ADVANCE – a block for program detention of the transactions.
- Blocks connected with devices: SEIZE – for modeling of taking an implement (device) from the transaction. In case the implement is taken, a queue is made in front of it; RELEASE - for modeling of transaction implement release by the one that has taken it.
- Blocks for multi-channel devices: ENTER – for modeling of taking one or several channels by transactions entering the block; LEAVE – for release of certain number of channels.
- Blocks for queues: QUEUE – for registration of a transaction entering a queue; DEPART – reduces the length of the particular queue with a definite number of units.
- Tables: TABULATE – the value of the argument in the table is inputted into it each time when a transaction comes into the block.
- Blocks changing the route of transaction movement: TRANSFER – a basic means of route change of a transaction; TEST – points the number of the next block for transferring the transactions in meeting/not meeting some conditions; GATE – allows to change the direction of the movement of the transactions depending on 12 logical attributes.
- Blocks changing the parameters of the transactions: ASSIGN – for appropriating number values of the transaction parameters.
- Blocks for transaction's families: SPLIT – for creating a certain number of copies of the coming transaction; ASSEMBLE – unites a given number of transactions in one family; GATHER – analogous of an ASSEMBLE block, but does not take transactions out of the model and throughputs them to the next block.
- User chains (lists): LINK – for taking a transaction out of the chain of current events and locating it in a user chain, where it waits for another transaction to take it out; UNLINK – for taking a transaction out of a user chain.

Parameters of the generated transactions in the model of Distributed TSO

The parameters of generated by GPSS transactions in the Distributed TSO model (Vasileva 2012) are the same as in (Culciar and Vasileva 2015; Vasileva and Noskov 2009). The parameters of every GPSS transaction modeling distributed transaction in DBMS, that receive their value just after they enter into the model by the block GENERATE are following (fig. 1):

P1 – Number of the transaction. The value is a sum of System Numeric Attribute MP2 and the number of the site;
P2 – number of the generating transaction site;
Pnel - number of elements processed by the transaction
Pel1 / Pel2– number of the first / second processed data element by the transaction (El1) / (El2);
Pbl1 / Pbl2 – type of the operation over the element El1 / El2: 1 (r) – if read (El1) / (El2); 2 (w) if write (El1) / (El2);

P5 – phase of the transaction processing: it takes the value of 0 in the transaction coming in the model and after the end of the operation read/write it takes the value of 1. In the Distributed TSO model P5=2, if Ignore(T); P5=3, if Rollback(T);

P6 / P7 – number of the site where the first / second copy of the first data element El1 is stored;

P8 / P9 – number of the site-executor where the first / second copy of the second data element El2 is stored;

P11 – number of the user chain where the corresponding sub-transaction waits for the release of the copy data element.

P$Vr – parameter that is used in making the decision about commit/rollback of transaction in Distributed TSO model: P$Vr=0, if the transaction has not requested the element yet; P$Vr=1, if T continues execution; P$Vr=2, if Rollback(T).

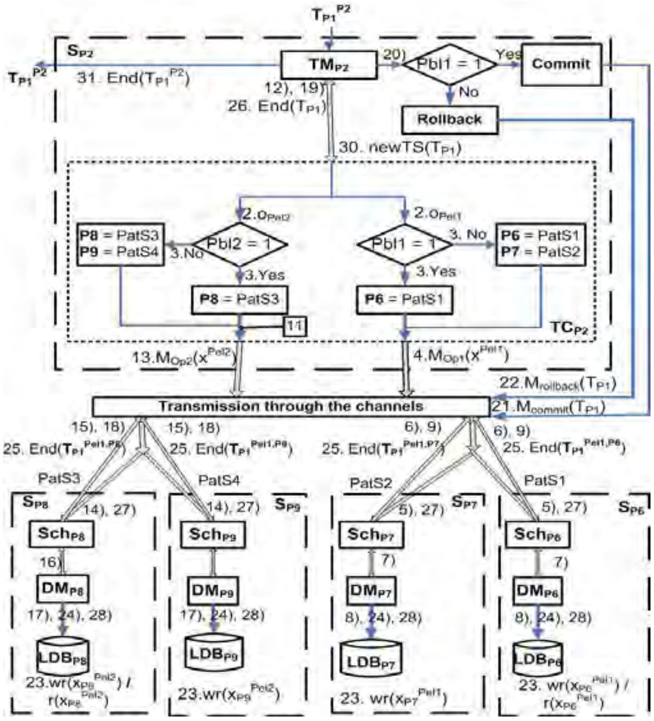


Figure 1: A scheme of distributed transaction execution by the modeling algorithm of Distributed TSO

Use of Variables in the Model of Distributed TSO

The arithmetic variables allow to calculate the arithmetic expressions which consist of operations upon SNA of the objects. Boolean variables give the opportunity to the concurrently checking of several conditions, proceeding from the assumption concerning the object's condition or the SNA values. (Minuteman 2010).

Some of the variables in the Distributed TSO model (Vasileva 2010) are:

V$ElemN1/V$Elemn2 – it calculates the number of the first/second element, which is processed from the transaction;

SiteRepl1/SiteRepl2 – it calculates the value of P6 and P7 / P8 and P9 (fig. 1);

RAZRBL1/RAZRBL01 – it determines whether the sub-transaction can block the first / second replica of El1 at the corresponding site-executor (P6/P7);

RAZRBL2/RAZRBL02 - it determines whether the sub-transaction can block the first / second replica of El2 in the corresponding site-executor (P8/P9);

SPIS2 / SPIS22 – it determines the number of the user chain where the sub-transaction processing the second replica of El1 / El2 can await the release of the corresponding replica.

Use of cells and matrices in the model of Distributed TSO

Cells and matrices are used for storing user numeric information. The record in these objects is used and read by the transactions. (Minuteman 2010)

In the considered model the cells are used mainly as counters:

X$BROITR1/X$BROITR2 - counter of transactions with length 1/2 elements;

X$BROITR - counter of generated transactions;

X$ZAVTR - counter of committed transactions

X$VOT1, X$VOT2, XVOT, XVOT12 – the counters serving in the taking of decision for continuation, ignoring or rollback of T

X$RESTRT - counter of restarted transactions.

In the example GPSS matrices are used: for the modeling distance between the nodes in DDBMS like (Culciar and Vasileva 2015) and for modeling the distributed database (DDB) (Vasileva 2012).

MX$RAZST and MX$RAZDEV are used to set the mean and standard deviation of the retention time of his transactions in the transmission of messages between the nodes of the distributed database system modeled for communication costs;

GBDA1 / GBDA2 - it models the local database (LDB) for first/second copies of El1 and El2. Each row of the GBDA1 / GBDA2 corresponds to the data element from DDB. The matrices have the following columns:

- Value of the element. This value is increased by 1 when a transaction records the data element and the value decreased by 1 in cases where the transaction has written the item but then had to rollback;
- Type of operation that the current transaction carried over elements: read, write or update. When a transaction rollbacks, it writes value 0;
- The timestamp of the transaction, that is the last recorded value of the data element. In this column is written the value the P1 parameter transaction, which recorded the new value in the first column;
- The timestamp (parameter P1) of the transaction, that is the last read the value of the data element;
- The number of the site-initiated the GPSS transaction (parameter P2), that is the last processed the data element. The value is 0 if the last transaction restarted.

Use of functions in the model of Distributed TSO

The function could be a means of giving uninterrupted or discrete functional dependence between the argument of the function and its value. Functions in GPSS are assigned to a table with operators for function description. (Minuteman 2010)

In the example (Vasileva 2012) we set following functions:
XPDIS – A standard exponential distribution function that determines the exponential distribution of inflow transactions with intensity λ = 1. In blocks GENERATE we redefine the intensity of the respective transaction inflow by variants of the operand A of the statement;
DistrS1 FUNCTION V$SiteRepl1,D6 – calculates the number of the site where is the first copy of El1 (serves to determine the value of the parameter P6);
DistrS2 FUNCTION V$SiteRepl1,D6 - calculates the number of the site where is the second replica of El1 (serves to determine the value of the parameter P7);
DistrS3 FUNCTION V$SiteRepl2,D6 - calculates the number of the site where is the first copy of El2 (it determines the value of the parameter P8);
DistrS4 FUNCTION V$SiteRepl2,D6 - calculates the number of the site where is the second replica of El2 (serves to determine the value of the parameter P9);
TransCor FUNCTION P2,D6 – determines the name of GPSS Facility modeling the transaction coordinator of the current transaction;
TraMan FUNCTION P2,D6 - determines the name of the GPSS Storage Entity modeling the transaction manager of the current transaction;
Opash1 FUNCTION P6,D6 – determines the name of queue in front of facility entity modeling scheduler of first sub-transaction that processes the first copy of El1;
Opash2 FUNCTION P7,D6 - determines the name of queue in front of the scheduler of second sub-transaction that processes the second copy of El1;
Opash3 FUNCTION P8,D6 - determines the name of queue in front of facility entity modeling scheduler of third sub-transaction that processes the first copy of El2;
Opash4 FUNCTION P9,D6 - determines the name of queue in front of facility entity modeling scheduler of fourth sub-transaction that processes the second copy of El2
BrEl FUNCTION RN4,D2 – transaction length - it is calculated in number of elements processed by transaction.

Use of queues in the model of Distributed TSO

The movement of the transaction flow could be detained due to inaccessibility of the resources. In this case the transactions make a queue. There could be defined points in the model where to gather statistics about the queues (queue registrators). Then the interpreter will gather the statistics about the queues (length, average time of the stay in the queue, etc.) automatically. (Minuteman 2010; Tomashevskii and Zhdanova 2003)
In this reason we set block QUEUE before every Facility entity and Storage entity (and block DEPART after the transaction serving by corresponding Facility/Storage) included in the model. (Vasileva 2012)
In order to collect statistics during the service transaction model is set statement QUEUE Totaltim after the segment setting the transaction length (parameter Pnel), the numbers of data elements (transaction parameters Pel1 (and Pel2)) processed by the transaction and the operations types (P3 (and P4)). This segment is after the generating transaction segment (Set the values of the parameters P1 and P2). And we set the statement DEPART Total Time before the block TERMINATE modeling, leaving the model by the current transaction.

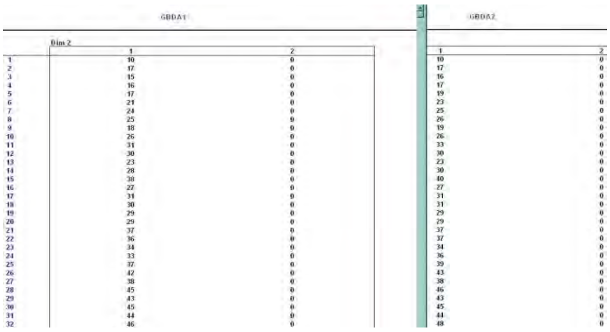
Use of tables in the model of Distributed TSO

The tables serve to gather statistics about casual quantities. They consist of frequency classes in which the number of concrete quantity hits is recorded (some of the GPSS System Numeric Attributes (SNA)). (Minuteman 2010; Tomashevskii and Zhdanova 2003)
In our model and studies on it, we used these GPSS tables:
DaTable – It serves of tabulating the time of residence of every transaction in the model (GPSS SNA M1). We set the block TABULATE DaTable before the TERMINATE block by which the current transaction leaves the model.
RespTime – Table of time of residence of the GPSS transactions in the queue TotalTim.

GPSS WORLD WINDOWS AND WATCHING SIMULATIONS

The windows on the GPSS World environment, provide excellent opportunities for observing the work of the modeled systems. Choosing which windows should be open on the screen to observe the simulation is done by choosing the command Simulation Window from the Window menu (Minuteman 2010):
- Blocks Window, which gives information about: labels and names of the blocks; number of entries in the corresponding block and the others. The window allows chronological tracking of transactions in blocks at model time;
- Facilities Window – a window of single channel devices - gives information about: Number / name of the device; Number of inputs; Rate of use, Average time of residence of the transaction in the device; State of readiness; Number of the last transaction occupying device; Number of interrupted transaction in the device; Number of interrupting device transactions; Number of transaction, pending special conditions; Number of transactions, pending the holding of the device; etc. In the example we use the Matrix window.
- Matrix Window – a window of the matrices (fig. 2 shows combined the “windows” of GBDA1 and GBDA2 matrices) - shows results in values of the data elements during a simulation. We can watch the concurrent change the values of the first and second replicas of the corresponding data element (The values

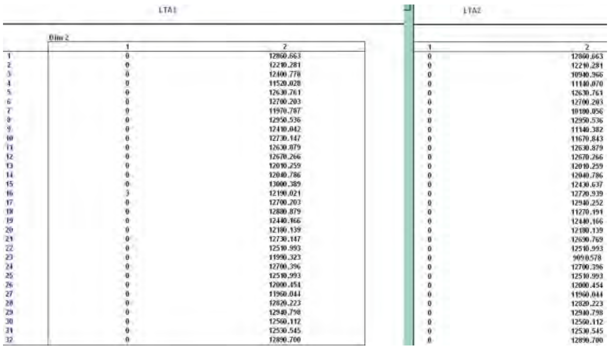
in the first column are the numbers of the elements. The values in the second column are the values of the first copies and the values in the fourth column are the values of the second replicas of the data elements.). Monitoring changes in the values of the elements in the second and the fourth columns we can make conclusions whether the modeling algorithm is executed correctly (if the algorithm is correct, the values in the second column should be the same as in the fourth column).



	GBDA1		GBDA2	
Dim 2	1	2	1	2
1	10	0	10	0
2	17	0	17	0
3	15	0	16	0
4	16	0	17	0
5	17	0	19	0
6	21	0	23	0
7	24	0	25	0
8	25	0	26	0
9	18	0	19	0
10	26	0	26	0
11	31	0	33	0
12	30	0	30	0
13	23	0	23	0
14	28	0	30	0
15	38	0	40	0
16	27	0	27	0
17	31	0	31	0
18	30	0	31	0
19	29	0	29	0
20	29	0	29	0
21	37	0	37	0
22	36	0	37	0
23	34	0	34	0
24	33	0	36	0
25	37	0	39	0
26	42	0	43	0
27	38	0	38	0
28	45	0	46	0
29	43	0	43	0
30	45	0	45	0
31	44	0	44	0
32	46	0	48	0

Figure 2: Combined view to the windows of matrices GBDA1 and GBDA2 for demonstration and tracing of the transaction service on Distributed TSO

Fig. 3 shows a combined window to monitor the parallel execution of transactions by monitoring the change of values in the lock tables of the copies of the data elements in simulation Distributed 2V2PL (Vasileva and Noskov 2009): first column – the numbers of the elements; second column – the lock types of the first replicas; third column – the numbers of transactions locked the first replicas; fourth column – the lock types of the second replicas; fifth column – the numbers of transactions locked the second replicas.



	LTA1		LTA2	
Dim 2	1	2	1	2
1	0	12860.663	0	12860.663
2	0	12210.281	0	12210.281
3	0	12400.778	0	10940.966
4	0	11520.028	0	11140.070
5	0	12630.761	0	12630.761
6	0	12700.203	0	12700.203
7	0	11970.787	0	10180.056
8	0	12950.536	0	12950.536
9	0	12410.042	0	11140.382
10	0	12730.147	0	11670.843
11	0	12630.879	0	12630.879
12	0	12670.266	0	12670.266
13	0	12010.259	0	12010.259
14	0	12040.786	0	12040.786
15	0	13000.389	0	12430.637
16	3	12190.021	0	12720.939
17	0	12700.203	0	12940.252
18	0	12880.879	0	11270.191
19	0	12440.166	0	12440.166
20	0	12180.139	0	12180.139
21	0	12730.147	0	12690.769
22	0	12510.993	0	12510.993
23	0	11990.323	0	9090.578
24	0	12700.396	0	12700.396
25	0	12510.993	0	12510.993
26	0	12000.451	0	12000.454
27	0	11960.044	0	11960.044
28	0	12820.223	0	12820.223
29	0	12940.798	0	12940.798
30	0	12560.112	0	12560.112
31	0	12530.545	0	12530.545
32	0	12890.700	0	12890.700

Figure 3: Windows of matrices LTA1 and LTA2 for tracing of the transaction service on Distributed 2V2PL

- Table Window – a window of the tables – a diagram of the frequency distribution of the tabulated transactions. (fig. 4 and fig. 5). Several windows can be open and ordered on the screen in the demonstration of a model and different aspects and elements of the modeled system can be watched in them.

We can observe the frequency distribution of the tabulated transactions as during the simulation and after modeling - finalized and this final version of the charts can be compared with published benchmarks (or be compared with other reported graphics as we could do for fig. 4 and fig. 5).

Fig. 4 and fig. 5 show frequency distribution of Response time (RT) of transactions. Frequency distribution of RT is another indicator of the performance of concurrency control algorithms. The diagrams of Frequency distribution of RT are built automatically by the formulated in the GPSS model tables (tables named DaTable in the Ditributed TSO and Distributed 2V2PL models). On fig. 4 is demonstrated the histogram of Frequency distribution of RT in modeling Distributed TSO at the total intensity of the input streams 100 tr/s (maximum load on the system) and observation time 28.8 seconds. Fig. 5 shows the results on the same conditions, but for the Distributed 2V2PL model.

Figure 4: Frequency Distribution of RT in modeling Distributed TSO at λ = 100 tr/s

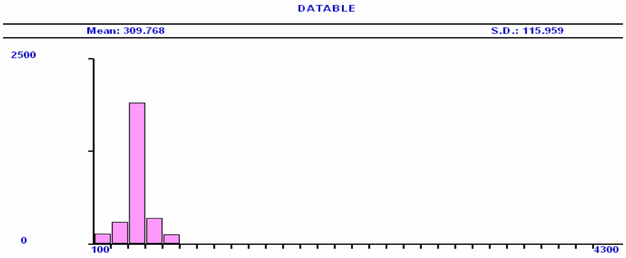


Figure 5: Frequency Distribution of RT in modeling Distributed 2V2PL at λ = 100 tr/s

The tables of Frequency distribution of RT besides that serve comparative analysis of concurrency control algorithms, serve also to assess the reliability of modeling algorithms by comparing with the template chart of Frequency distribution of RT (TPC Council 2010).

Similarly, can be compared with the template graphics and charts for throughput of fig. 6, fig. 7 and fig. 8.

SIMULATION RESULTS

Through defined in the model tables and reports generated after modeling, we can determine one of the most important features for DDBMS - the Response time. With defined in the model variables and cells, as described in Section “GPSS World and Distributed TSO Modeling” we could calculate another main characteristics of service transactions in DDBMS: Throughput (TP) and Service Probability (SP).

Throughput of one system is calculated in the number of requests serviced per unit time (Tomashevskii and Zhdanova 2003). For our model they are respectively the values of the cell X$ZAVTR and time modeling at different startups of the modeling algorithm.

Time modeling in the modeling algorithms is set in milliseconds in block GENERATE at the end of the models. All streams transactions are received upon an exponential law with a variable at different studies with an average length of the interval. In all modeling algorithms we consider 6 streams generated by GPSS transactions modeling 6 sites in distributed database system, from which Poisson law shall go global transactions.

The diagram of fig. 6 presents the results of simulations of Throughput for Distributed 2V2PL and Distributed TSO algorithms at the same intensities of input flows depending on the monitoring period (in seconds): The graph marked with a thin blue dashed line (2V2PL) and the graph indicated by a thick black line and square markers (TSO) – 6 streams, each with an average intensity 4,17 tr/s (minimum load - intensity cumulative flow 25 tr/s and operand); The graph marked with thin black line and asterisks (2V2PL) and in the graph illustrated by dashed lines with triangular markers (TSO) - 6 streams, each with an intensity of 8,33 tr/s (average load - intensity of the aggregate stream 50 tr/s); The graph indicated by the thin dotted line (2V2PL) and the graph indicated by a thin blue line (2PL TSwd) - 6 streams of medium intensity 16,67 tr/s (maximum load - intensity of aggregate stream 100 tr/s).

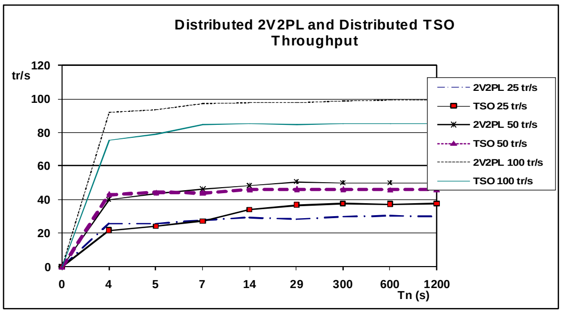


Figure 6: Throughput of the systems

To get the appropriate intensity of each of inflows in operand A in the corresponding block GENERATE we set value: 60 - maximum load; 120 - average load; 240 - minimum load. And to change the distance between the nodes of the DDBMS and conduct research on the dependence of the throughput of the system (and transaction SP and other performance indicators of concurrency control algorithms) of the distances between nodes in different modes and CC algorithms, we should change the values in the cells of the MX$RAZST matrix (and the cells of the MX$RAZDEV matrix).

On fig. 7 graphics are given the values that are obtained for TP by substituting the fixed in the receiving reports values of the cell *X$ZAVTR*. Intensity of inflows transactions are the same as the graphs of fig. 6, it was changed only the distance matrix *MX$RAZST* – their cell values are increased twice compared to models whose results are reported in the graphs of fig. 6.

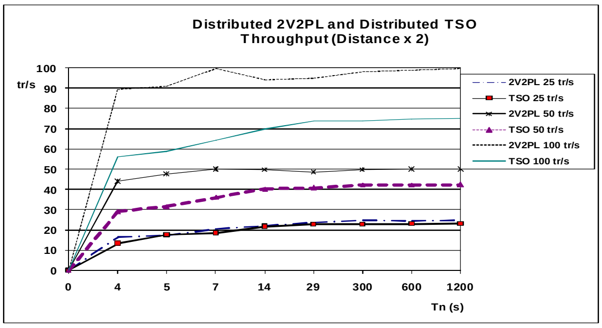


Figure 7: Throughput in the models at doubled distance between the sites in the system

From the graphs of fig. 7 and fig. 8 it can be concluded that with the increase of the distance between sites in the system, the throughput graphs are "spaced apart" more. This is very evident in the graphs at maximum load of the system (intensity of inflow 100 tr/s).

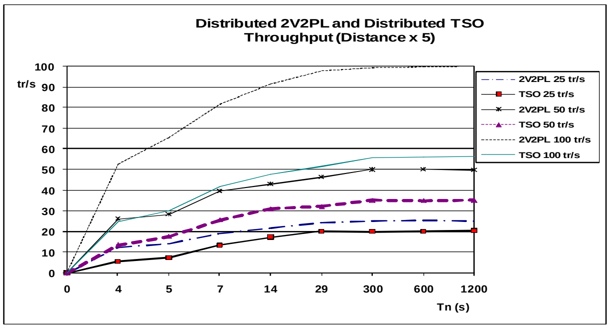


Figure 8: Throughput of the system (distance x 5)

Service probability factor or completion of service transactions serves to assess the dynamic properties of DDBMS. The probability of service P_s of distributed transactions is calculated by the formula (1) (Tomashevskii and Zhdanova 2003):

$$P_s = \frac{N_c}{N_g} \qquad (1)$$

where N_C is the total number of fixed transactions (cell value of *X$ZAVTR* after modeling, and N_g is the total number of transactions generated for the same period of time (cell value of *X$BROITR* after modeling).

Fig. 9 presents the results for the service probability of distributed transactions at simulation algorithm Distributed TSO and Distributed 2V2PL at the same intensities of inflows (as for Figure 6).

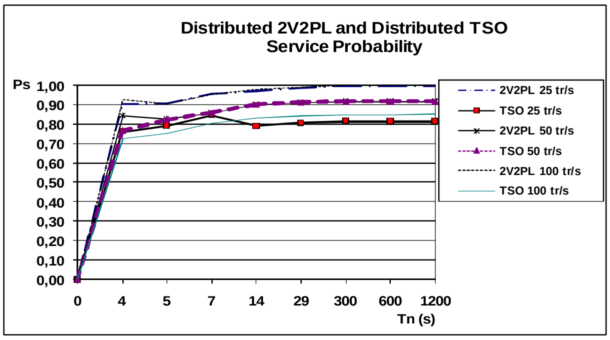


Figure 9: Service probabilities in the 2V2PL model and the TSO model

In the diagram of Figure 10 are shown in graphical form the data collected from the reports generated after the simulations conducted and reported in (Vasileva 2012). It can be seen that the graphics of RT measurements have the same kind - a rapid increase in the beginning, slow growth and stationary mode.

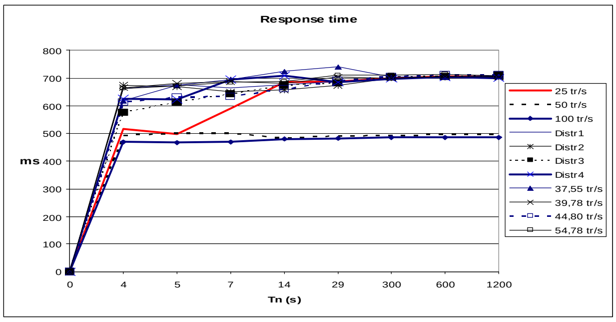


Figure 9: Response time of the Distributed TSO model in the different intensities of the incoming streams

CONCLUSION

The system of simulation GPSS World permits creation of effective simulation models of transaction concurrency control (in particular model of distributed transaction Timestamp ordering in DDBMS.

The demonstrated model developed in the simulation GPSS environment proves the great opportunities of the GPSS for simulation of algorithms for concurrency control of the distributed transactions in DDBMS. Such systems and the work algorithms of their nodes are part of the enormous field of discrete event systems.

The advantages of modeling such systems with GPSS (generating various flows of different types of requests with different in type and number parameters, realization of the service algorithm, branching out (even cyclic recurrence)) are shown in the illustrated model.

The developed simulation models (as well as their complicated analogues – for longer transactions which process two and more data elements) allow the needed statistics to be gathered with the purpose to investigate and analyze the protocols for timestamp ordering and two-phase locking in DDBMS. On the basis of the gathered data from the simulations we define the coefficients which serve to assess the effectiveness of the algorithms for concurrency control in DDBMS.

REFERENCES

Chardin, P. 2005. “Data multiversion and transaction concurrency control”. http://citforum.ru/database/articles/multiversion.

Connolly, T., C. Begg. 2002. “Database Systems”. Addison – Wesley.

Culciar, A., S. Vasileva. 2015. “Simulation studies of the implementation of Centralized two-phase locking in DDBMS”. In *Proceedings of the 29th European Conference on Modelling and Simulation ECMS 2015*, May 26th – May 29th, 2015, Albena (Varna), Bulgaria, 107-114.

Date, C. 2000. “Introduction to Database Systems”. MA: Addison – Wesley.

Minuteman Software. 2010. “GPSS World Reference Manual”. http://minutemansoftware.com/reference.

Tanenbaum, A., M. Steen. 2007. “Distributed systems. Principles and paradigms”, Pearson Prentice Hall.

Thomas, R. 1979. “A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases”. In *ACM Trans. Database Systems*, 4(2), 180-209.

Tomashevskii, V. and E. Zhdanova. 2003. “Simulations in GPSS”. Bestseller, Moscow.

TPC Council. 2010. “TPC – C Benchmark”. Standard Specification Revision 5.11. Transaction Processing Performance Council, (Feb).

Vasileva, S. 2012. “Algorithm for Simulating Timestamp Ordering in Distributed Databases”. In *Proceedings of the VI International Conference on Information Systems and GRID Technologies*, (Sofia, BG, Jun.1-3), 352-364.

Vasileva, S., Y. Noskov. 2009. “Simulation of Distributed Two-version Two-phase Locking”. In *Proceedings of the Fourth All-Russia Scientific-Practical Conference on Simulation and its Application in Science and Industry “Simulation. The Theory and Practice” (IMMOD-2009)*, Vol. II, 98-103.

AUTHOR BIOGRAPHIES

SVETLANA Zh. VASILEVA was born in Plodiv, Bulgaria and went to the Saint-Petersburg State Electrotechnical University (LETI), where she studied Engineer Process Systems "Computer Aided Design" and obtained her degree in 1994. She has also a degree as a PhD in Informatics. She worked as a programmer (1995-1996 - EF "DeytaSoft" - Plovdiv), assistant professor of Informatics (1997-2000 - Agricultural University - Plovdiv), before moving in 2001 to the University of Shumen, College Dobrich where she was a chief assistant professor in Informatics. Currently she is a chief assistant professor at Varna University of Management. Areas of scientific interests: Control systems for distributed databases; Simulation modeling of processes and systems; Application of information technologies in education.

E-mail address is : `svetlanaeli@abv.bg`.