Statistical Study of Firm Real-Time Transactions Behavior *

S. Semghouni*, L. Amanton*, B. Sadeg* and A. Berred**
Laboratoire d'Informatique du Havre*,
Laboratoire de Mathématiques Appliquées du Havre**
Université du Havre,25 rue Philippe Lebon BP 540
76058 Le Havre Cedex, FRANCE
Tel :(33) 02-32-74-43-84 Fax : (33) 02-32-74-43-14
{samy-rostom.semghouni, laurent.amanton, alexandre.berred, bruno.sadeg}@univ-lehavre.fr

Abstract

In this paper, we study statistically firm real-time transactions behavior under a pessimistic and an optimistic concurrency control and scheduling protocols. Up to now, all the studies have been done on RTDBS performances by measuring the transactions success ratio (number of transactions meeting their deadlines over the total number of transactions), which is the well-known and accepted criterion for this purpose. Our study focuses among others, on the number of operations composing a transaction and the concurrency control protocols used. We propose a reasonable approximation of transactions success ratio behavior by a probability distribution and we suggest a method to predict the system performances. We have carried out simulations where we vary different parameters including the arrival rate, the transactions sizes and the slack factor.

1 Introduction

A real-time database system (RTDBS) can be viewed as a system which inherits mechanisms of both traditional database systems and real-time systems (RTSs). RTDBSs must guarantee simultaneously the transactions *ACID* (Atomicity, Consistency, Isolation, Durability) properties and schedule the transactions in order to meet their individual deadlines [14].

In RTDBSs, the main issue is the transactions concurrency control and scheduling (CC&S). These mechanisms are designed to cooperate tightly in order to assign transactions priorities in the case of data access conflicts. Many concurrency control and scheduling policies have been proposed to manage the real-time transactions according to their categories [1, 15]. Three classes of real-time transactions are defined according to the consequences they lead to in case of missing their deadlines: *hard, firm and soft*. Hard transactions must meet their deadlines otherwise disastrous effects may occur. Firm transactions must also meet their deadlines, but a transaction is only aborted when it misses its deadline, as it becomes useless. Soft transactions are still useful even after their deadlines, but they provide less quality of service to the application.

Up to now, almost all the studies of RTDBS performances deal with the transaction success ratio using only a simple measure: the percentage of successful transactions (transactions which have met their deadlines). To the best of our knowledge, there is no statistical study related to RTDBs performances, even if several statistical studies investigated RTSs and DBSs. For instance, Bernat et al. [3] have dealt with a probabilistic study of hard real-time systems. They have combined both measurement and analytical approaches into a model for computing probabilistic bounds of the worst case execution time (WCET). Edgar et al. [7] have proposed a statistical analysis for the problem of WCET in hard RTSs. Their paper gives an alternative analysis based on the estimation of WCET from test data within a specific level of probabilistic confidence. Other studies are devoted to periodic RTSs problems. Diaz et al. [6] have described a stochastic analysis of periodic RTS. The description contains a method for computing the response time distribution of each task in the system. That makes it possible to determine the missing deadline probability of individual tasks. Atlas et al. [2] have presented statistical rate monotonic scheduling (SRMS).

A few work has been done on transactions probabilistic/statistical analysis in traditional DBMSs. Among these researches, Singhal et al. [17] have studied the workloads in three commercial relational DBMSs, running commercial applications and production environment. Their study considers topics such as frequency of locking and unlocking, deadlock and blocking, duration of locks, types of locks, correlations between applications of lock types, two-phase versus non-two-phase locking, when locks are held and released, etc. There are non statistically studies on real-time concurrency control protocols. In RTDBS,

*This work is supported by the research ministry of France (*ACI-JC* #1055)

Notations	Definitions	Values
λ	Transaction arrival rate	0.1 to 0.8
DB-size	Number of data in the DB	1000
Slack Factor	H(x) initialization α and β	$\alpha = \overline{1,3}$ and $\beta = 0.3$
Tr-size	short/ middle/ long	5-10, 5-70, 5-160
Capacity	Capacity of execution coefficient	10 tasks/cpu-cycle
Time	Duration of one test	1000 cpu-cycle
CC&S	CC protocols & scheduling policy	(2PL-HP or Wait-50) & EDF

Table 1. Simulation parameters.

we can quote the work of Haritsa et al. [11, 10] that have studied optimistic concurrency control protocols in particular OPT-BC, OPT-Wait and Wait-50. However, they didn't propose a statistical study.

In this paper, we consider firm transactions and we propose a stochastical study of their success ratio, under several conditions corresponding to parameters such as their size (short, middle or long), the arrival process, the system load, the conflicts level and the probability of "read" and "write" operations. This study is carried out under both a pessimistic and an optimistic approach. To this purpose, we have developed a simulator including the components generally encountered in an RTDBS [12, 15]. After several simulations, we have obtained the probability density function which approximates reasonably the behavior of the transactions success ratio.

The remainder of this paper is organized as follows. In Section 2, we describe the simulator architecture and related parameters. Section 3 is devoted to simulations experiments and the interpretations. Finally, Section 4 summarizes the main ideas of the paper and outlines some aspects of our future work.

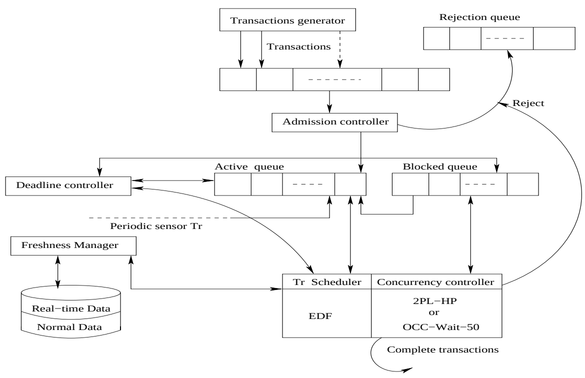


Figure 1. Simulator architecture.

2 System model

Figure 1 shows the architecture of our centralized RTDB simulator [1]. Transactions arrive in the system according to a Poisson process with average rate λ. The set of operations of each transaction is randomly generated and built according to a level of data conflicts. Given that data is selected, the transactions execute with the same probability a "read" or a "write" operation.

[1]The simulator is available on line at the following address: http://www-lih.univ-lehavre.fr/∼semghouni/Applet/PageSimulateur.html

When a transaction is submitted to the system, an admission controller mechanism (ACM) is employed to decide whether to *admit* or *reject* that transaction into the execution queue. The primary task of the ADM is to perform schedulability test of new transactions upon their arrival, to decide whether or not a new transaction should be admitted for execution [8]. Transaction deadlines are controlled by the Deadline Controller. Deadlines are verified every CPU cycle and the scheduler is informed immediately when a transaction is likely to exceed its deadline.

Scheduling is the mechanism by which a transaction is chosen to execute on the processor. There are several types of real-time schedulers. We quote for examples: *HPF (Highest Priority First)*, *LLF (Least Laxity First or Least Slack Time)* [5], *EDF (Earliest Deadline First)* [19] and *RM (Rate Monotonic)* [13]. Our Scheduler is based on *EDF algorithm* for the determination of the transactions priorities. There exist a lot of work on concurrency control and scheduling in RTDBSs [11, 18]. In our simulator, we chose the two well-known concurrency control and scheduling protocols: a pessimistic (2PL-HP protocol) [1, 4] and an optimistic (OCC-Wait-50 protocol) [9].

2.1 Conflicts level

Data conflicts emerge from the behavior of the transactions in the database. Generally, in a database, we know a priori that some data are more important than others because they are frequently requested. We thus assign to each data item a probability proportional to its importance.

Let $r_1, r_2, \ldots, r_k, \ldots, r_n$ denote the ranking of the data item D_1, D_2, ..., D_k, ..., D_n. The probability of drawing the data item *i* is given by the following formula:

$$P_i = \frac{r_i}{R_i}$$

where $R_i = \sum_{i=1}^{i=n} r_i$, is the sum of all ranks.

2.2 Deadline assignment function

The deadline assignment policy we have used to generate the transactions is given by the following formula:

$$D_T = A_T + B_T * (H(B_T) + 1)$$

where D_T denotes the deadline of the transaction T, A_T the arrival time (or release time) of T, B_T the best (minimum) estimated execution time of T and $H(x) = \alpha * (1 - \exp(-\beta * x))$ a slack function, scaled by the α and β parameters. Note that the value of the best execution time

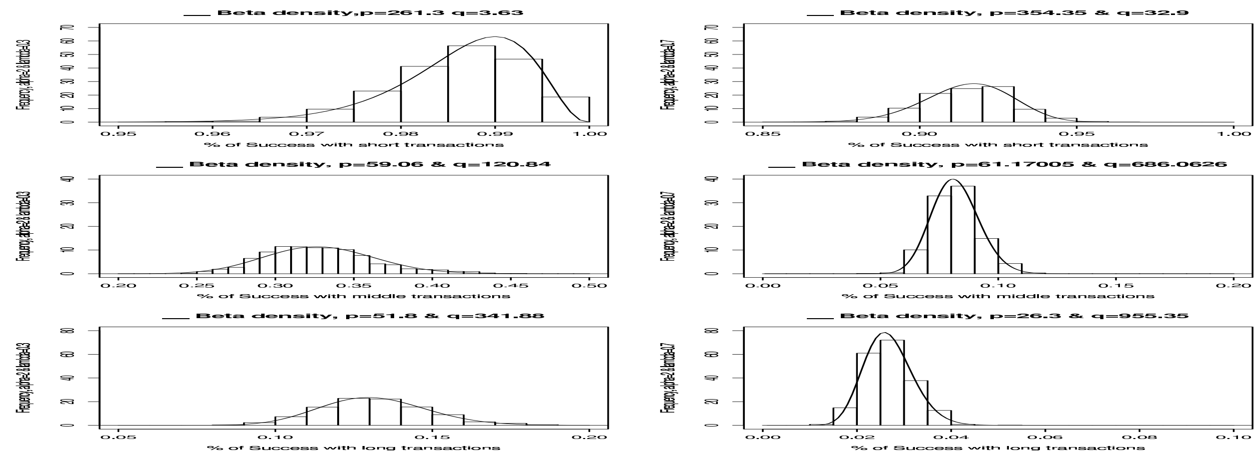


Figure 2. 2PL-HP Frequency distribution of the random success rates.

(B_T) is always greater than or equal to 0. The values of $H(B_T)$ are then ranging from 0 to α. Note also that the smallest is the best execution time, the tightest is the *Slack time*. This characteristic allows the transactions with short execution times, particularly update transactions, to be assigned the highest priorities (EDF policy), when a conflict with transactions with longer execution times occurs.

3 Simulation results

To assess the performances of our database, we carried out simulation experiments (the simulations parameters are summarized in Table 1) which evaluate the success ratio of transactions. Each experiment is repeated 1000 times. Transactions are classified into three categories: short, e.g. transaction in navigation system; middle, e.g. transaction in stock market; and long, e.g. multimedia transaction.

Figure 2 shows the frequency of short, middle and long transactions when $\lambda = 0.3$ (normal workload $\simeq 300$ transactions) and $\lambda = 0.7$ (high workload $\simeq 700$ transactions). The frequency distribution of the success ratio could reasonably be approximated by a Beta random variable [16] (see Appendix A), according to the *Kolmogorov-Smirnov* statistical test with a confidence interval equal to 95%.

Simulations results revealed that the best performances are given with short transactions, particularly with OCC-Wait-50. However, this situation is reversed rapidly in favor of 2PL-HP protocol when we use middle or long transactions. This is explained by the inherent characteristics of each protocol. With the OCC-Wait-50 protocol, conflicting transactions have to wait for the resolution of the conflict as long as their deadlines are not reached. This causes the system overload and appears clearly when we use middle and long transactions. In the case of the 2PL-HP protocol, the transactions are aborted and restarted immediately when a conflict occurs [10].The study of success ratio frequency has revealed that the RTDBSs are very complex to control. The success rate is very sensitive to the variation of slack function, size of transaction and the choice of the CC protocol. These two last parameters play a major role in the optimization of the system performances. In the following of the paper, we propose to use regression method of Beta parameters to predict the system performances.

3.1 Non-linear regression of $\hat{p}$ and $\hat{q}$ parameters

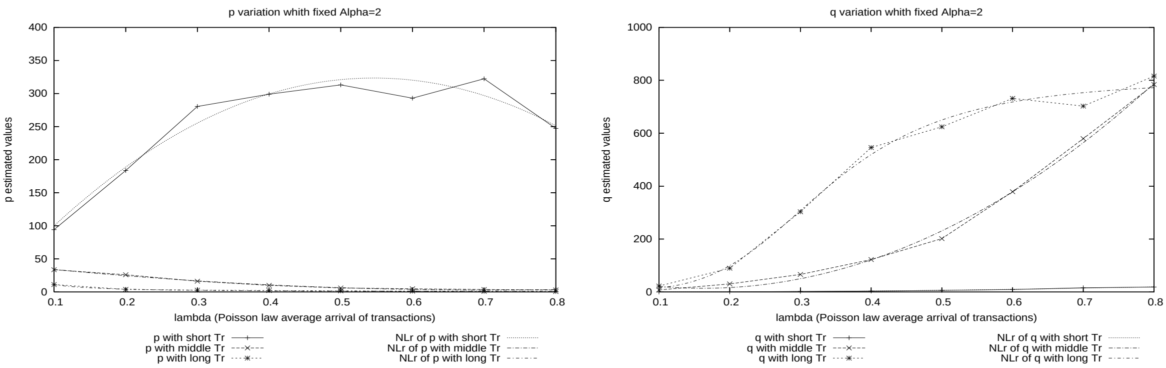
We have estimated all the values of p and q parameters (see Appendix B) of the Beta density for all results according to the simulations hypothesis (transaction size and λ). Figure 3 shows the layout of the estimated p and q values according to both the average of the arrivals λ and the transaction size when using 0CC-Wait-50. To determine the approximated values of p and q for all the values of λ, we propose to use the *Non-Linear Regression*, which we note **NLR**. The regression model is given by the following function: $F(X) = \exp(a + x * b)/(x + 1)^c + \exp(d + x * e)/(x + 1)^f$, where a, b, c, d, e, f are values adjusted continuously by the regression.

The functions obtained by the non-linear regression are used to deduce the values p and q. That gives the awaited parameters of the Beta density according to the system parameters (CC protocol and α, λ, etc.). This allows us to derive the mean of the density which is interpreted as the expected average success ratio of the system states. The best and the worst execution scenario can be deduced from the distribution. Theses indications, particularly the lower bound of the success ratio, makes it possible to see if the system overshoot[2] will not be violated.

4 Conclusion

We have studied in this paper the statistical performances of centralized RTDBS. We have particularly presented for the first time a comprehensive evaluation of real-time transactions behavior according to optimistic and pessimistic concurrency control and scheduling protocols. In this statistical study, we have modeled the success

[2]Fixed by the system manager, it represents the limit beyond which the system becomes unprofi table.



(a) $\hat{p}$ parameter (b) $\hat{q}$ parameter

Figure 3. Graphical representation of the non-linear regression of $\hat{p}$ and $\hat{q}$ with OCC-wait-50.

ratio distribution of transactions. A probabilistic density of the transactions success ratio has been estimated and confirmed by the *Kolmogorov Smirnov* test. We proposed a method based on statistical study of transactions behavior used to predict the system performances according to system parameters.

In our future work, we plan to study the behavior of updating transactions, the influence of the database size on the system performances. Moreover, we plan to test other CC&S protocols and to design and implement an hybrid system.

References

[1] R. Abbott and H. Garcia-Molina. Scheduling real-time transactions: A performance evaluation. *ACM Transactions on Database Systems*, 17(3), September 1992.

[2] A. Atlas and A. Bestavros. Statistical rate monotonic scheduling. *IEEE, RTSS*, 123, 1998.

[3] G. Bernat, A. Colin, and S. Petters. WCET analysis of probabilistic hard real-time systems. *IEEE RTSS*, 2002.

[4] P. Bernstein, V. Hadzilacos, and N. Goodman. Concurrency control and recovery in database systems. *Addison-Wesley*, 1987.

[5] C. Bonnet and I. Demeure. *Introduction aux Systèmes Temps Réel.* Hermes Science Publication., 1999.

[6] J. Diaz, D. Garica, K. Kim, C. Lee, L. Lo Bello, J. Lopez, S. Min, and O. Mirabella. Stochastic analysis of periodic real-time systems. *In Proceedings of the* 23^{rd} *RTSS*, USA, 2002.

[7] S. Edgar and A. Burns. Statistical analysis of WCET for scheduling. *IEEE, RTSS*, 2001.

[8] J. Hansson and S. Son. Overload management in RTDBs. In *Real-Time Database Systems*, pages 125–139, 2001.

[9] J. Haritsa. Transaction scheduling in firm real-time database systems. *University of Wisconsin-Madison*, August 1991.

[10] R. Haritsa, M. Carey, and M. Livny. Dynamic real-time optimistic concurrency control. *IEEE*, 1990.

[11] R. Haritsa, M. Carey, and M. Livny. On being optimistic about real-time constraints. *ACM PODS*, 1990.

[12] Y. Kim and H. Son. Supporting predictability in real-time database systems. 1996.

[13] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, (20), 1973.

[14] K. Ramamritham, H. Son, and L. Dipippo. Real-time database and data services. *Real-Time System Journal*, 28, 2004.

[15] K. Ramamritham, M. Xiong, R. Sivasankaran, J. Stankovic, and D. Towsley. Integrating temporal, real-time, and active databases. *ACM SIGMOD Record. Special Issue on RTDBS*, 25(1), March 1996.

[16] S. Semghouni, A. Berred, B. Sadeg, and A. L. On real-time DBMSs performances. In *Proceeding of The* 5^{th} *Workshop on Simulation*, St. Petersburg State University, St. Petersburg, Russia, June 2005.

[17] V. Singhal and A. Smith. Analysis of locking behavior in three real database systems. *VLDB Journal 6 (1)*, 1997.

[18] S. Son, S. Park, and Y. Lin. An integrated real-time locking protocol. *In Proceedings of the 8th IEEE International Conference on Data Engineering*, 1992.

[19] J. Stankovic, M. Spuri, K. Ramamritham, and G. Buttazzo. *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms.* Kluwer Academic Publishers, 1998.

A Beta density

A random variable *X* follows a Beta density of parameters $p > -1$ and $q > -1$ (noted $X \rightsquigarrow \beta(x,p,q)$ if its probability density is given by the following formula :

$$f(x,p,q) = \frac{x^{p-1}(1-x)^{q-1}}{\mathcal{B}(p,q)},\ x \in [0,1],$$

where $\mathcal{B}(p,q) = \int_0^1 x^{p-1}(1-x)^{q-1}dx$.

B Beta density Estimator

Lets $x_1,\ldots,x_n$ be a sample vector of probability, the parameters p and q of Beta density can be estimated according to these tow formula:

$$\hat{p} = \overline{x}\left[\frac{\overline{x}(1-\overline{x})}{s^2} - 1\right] \quad \& \quad \hat{q} = (1-\overline{x})\left[\frac{\overline{x}(1-\overline{x})}{s^2} - 1\right]$$

Where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the arithmetic mean and $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$ is the variance.