International Research Journal of Applied and Basic Sciences. Vol., 3 (5), 1052-1058, 2012
Available online at http:// www. irjabs.com
ISSN 2251-838X ©2012

Simulating Impact of Impatient Users on Database Server Performance

Mohammad Shabanali Fami[1*], Elham Shabanali Fami[2], Mohammad Ali Montazeri[3] and Mohammad Taghi Isaai

1- Arak Branch, Islamic Azad University, Arak, Iran.
2- Isfahan University of Technology, Isfahan, Iran.
3- Isfahan University of Technology, Isfahan, Iran.
4- Sharif University of Technology.

*__Corresponding Author Email__: emfami@gmail.com

Abstract

Database server is the major part of information systems. In this paper, we investigate the behaviour of impatient users in the server performance parameters with modelling database system. We use multiclass queuing network for this simulation. The user can behave impatiently in the form of optional cancellation or cancellation forced by the system. By analysing the simulation results, we indicated that with efficient user interface design that informs users sufficiently we can prevent performance losses due to the cancellation of impatient users. Considering a suitable cancellation time, will guarantee a constant response time for the system and will increase the system throughput and reduces the costs.

Keywords: database performance, impatience user behaviour, multiclass queuing network, simulation.

Introduction

Database servers are one of the key elements in providing IT services and their performance is important to service delivery. A major point that should be considered is impact of the behaviour of IT service users in server's performance. User's impotency and their retrying to get service can lead server to spend the time on the cancelled services or resent requests. In a real network, if the response time increases up to a threshold, the user's behaviour will change the system behaviour. Because of different levels of user patience, after a period of time correspond with their degree of patience, if they didn't get the complete service, they may cancel the service request or retry to send request again.

Boland and Roberts (Bonald and Roberts, 2001) presented a model that avoid congestion by adequate provisioning, overload can clearly occur on certain network links. They survey user impatience and reattempt behaviour in overloaded link accounting by some simple early models. They reported (Bonald and Roberts, 2003) throughput performance is generally satisfactory as long as demand is only slightly less than capacity. And in overload, some flows must be abandoned. Hoxmeier investigated that slow system response time leads to dissatisfaction and determine if experience influences response time tolerance.

Altman and Yechiali (Altman and Yechiali, 2006) analyse queue networks with impatience user and server vacations. They show the proportion of customer abandonments under the single-vacation regime is smaller than that under the multiple-vacation discipline. Gromoll (Gromoll et al., 2008) investigated a processor sharing queue with renewal arrivals and generally distributed service times. His fluid model captures many essential features of the underlying stochastic model, and it is used to analyse the impact of impatience in processor sharing queues.

Perel (Perel and Yechiali, 2010) analysed customers' impatience in three Markovian models: the single server case, the multiple server case and the infinite-server case. They calculated the mean total number of customers in the system.

Thus, Evaluation of user satisfaction, response time and system performance and resource consumption of the service when users are impatience is very important.

Database performance modelling research published in 1984 by Lazowska (Lazowska, 1984). But first researches perform by Sevcik (Sevcik, 1981) who built a layered queuing model to predict the performance of database systems. Later, this model was used by Casas and Sevick to analyse the influence of buffer management in the performance. Researches on performance modelling of database system divide to two way: research on database design performance and database server performance.

Lin (Lin and Lieh-san, 1989) compared the performance of DBMS with two transaction processing techniques: one in which the transaction is partially processed in the client and other in which the transaction is mainly processed in the server.

Zhu and Lu (Lazowska, 1984) developed an open multiple-class queuing network model that combines web and database servers. In this model response time calculated and in the results can see when arrival rate increase the queue length increases.

Garcia (Garcia, 2010) present a simple model and similar to model that present in (Zhu et al., 2000) that is so near to the real results of running TPC-C on a server with for Xeon processor and windows server 2003 platform.

In this paper, we introduce the TPC-C benchmark in section 2, and we describe the Garcia model. In section 3, we discuss on the details of implementation and in section 4, we will compare the results and analyses them. Finally the conclusion and feature works is presented in section 5.

TPC Benchmark™ C (TPC-C)

TPC benchmark is an OLTP workload that is developed to simulate complex OLTP application environment and consisted of a mixture of read-only and update intensive transactions. This benchmark is a subcommittee member of some companies like Amdahl, Bull, CDC, DEC, DG, Fujitsu/ICL, HP, IBM, Informix, Mips, Oracle, Sequent, Sun, Sybase, Tandem, and Unisys (TPC, 2010).

TPC introduce a company to simulate the company transactions. This company is wholesale suppliers that their branches and warehouses are distributed geographically. More expands in company business, more established warehouses (TPC, 2010).

Company's customers command a new order or status request of existing order. The length of each order is ten lines in average. There is a case that current warehouse has not the stock and it should be provided from another warehouse. This case will occur in one percent of the situations (TPC, 2010).

Besides, there are other types of orders in such a company. For example, customers pay their stock cost, system processes delivery status and investigate stock level to determine potential shortage in supplying stocks (TPC, 2010).

Table 1. Characteristics of the transactions of the TPC-C benchmark

Transaction Class	Minimum % of mix	Minimum Keying Time	Minimum Mean of Think Time Distribution	90 Percentile Response Time Constraint
New-Order	45	18s	12s	5s
Payment	43	3s	12s	5s
Order-Status	4	2s	10s	5s
Delivery	4	2s	5s	5s
Stock-level	4	2s	5s	20s

Customers randomly select one of the classes so that the existing transactions in the total mix of selected transactions at any time that follow the distribution shown in column "Minimum% of mix" of table I (TPC, 2010).

User Behaviour

Customer spends time for think time and keying time, based on the average contingency that follows Table 1 and then they receive the desired service. One of the scenarios that may happen is that all their customers to be patient and then to get into the queue waiting to complete their service. This is the simplest scenario that used by Garcia in his model (Garcia, 2010). But in reality it is very unlikely and illogical that applicants be patience to get full service without latitude. There are cases where clients are machine. They are expected to be prolonged. In these cases the results can be extended to the forced to cancellation.

To model this behavior in terms of settling time that a random exponential distribution with mean m at the beginning of the applicant's entry to the server is allocated to him. This time the user will have to wait. And after this period. Applicant at any point of the service will leave the server and the service will be deterred. In this paper three cases that the patient with low, medium and large patience used. the average waiting time of

25 and 100 and 250 units of time were considered to be the effect of the behavior of users to compare the value of this parameter in the performance also be addressed.

Method

Implementation Details

In this paper, due to lack of Garcia's model data and unavailability of their simulation software, at first their model simulated based on information given in the papers (Garcia, 2010) and (Zhu et al., 2000), and then we calculate system performance parameters by adding user impatience behaviour and finally, we compare these two situations. Of course with professional judgement we found that, outputs of Garcia's model relative to increasing number of shops displays performance parameters that are consistent with our implementation. Logically when time passes, the number of stores increases. When the entrance rate of system raised, we expect to have queues longer than before and response time more than before that observed in (Lin and Lieh-san, 1989) (Zhu et al., 2000).

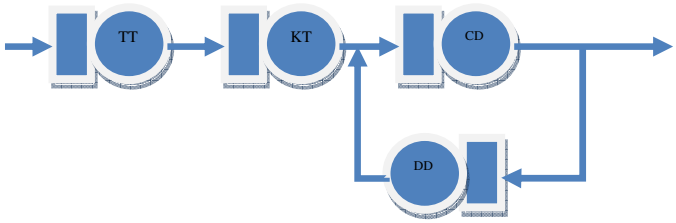

Figure 1. Queuing network model. It is a multiclass queuing model which considers CPU and disk service time.

Garcia in his model (Garcia, 2010) considered a server according to queuing network in Figure 1. In this model TT is a queue for thinking time of customers and TK is a queue which models keying time of them. DC is the queue that simulate processor server. This queue has 4 parallel

Server with quantum processing time equal to 0.05 .In queue the applicant spends some times to process the request. The service time of processor queue is randomly distributed with Weibull distribution with mean variable SC and the shape parameter 10 (Hoxmeier, 2000). In each stage of processing, SC will obtain according to the formula SC = DC / VC (Jain, 1991). DC is the required time for processor and VC is the rate of visiting processor. These variables obtain from the operational low (Zhu et al., 2000) Dc = UC / X و VC=VD+1 when processing service is received, disk service may be required for applicants. So, randomly some applicants wait in DD queue that is a queue for disk service, and the others applicant's services complete and transfer to output of system. Circular queue in Garcia's model is simulated with sequentially entrance of applicant and their exit after service delivery. This behaviour is not suitably fit with real user's behaviour in online servers but due to continually coming requests we can assume that some applicant will request services permanently. According to this logic, both models will be identical in this case. This model assumes that the server after processing the request may need to forward request to disk and after a processing time it will be completed. This is evident from the formula VC = VD +1. The promise is true because we can reasonably assume all applicants required the disk, will get all of its need in one time and after returning to processor it will not need the disk again. The service complete in the second stage. From another perspective it

This can be analysed in an infinite loop in processor and disk is a software bug. So an applicant can get two kinds of services, one processor usage or two processor usages along with one disk requirement.

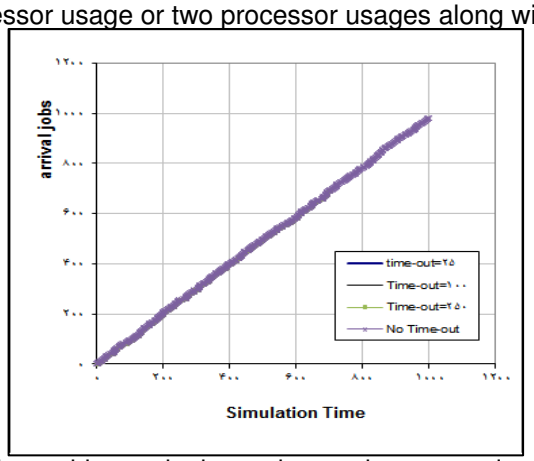

Figure 2. jobs arrival rate.this graph shows that as time passed, new jobs will arrive.

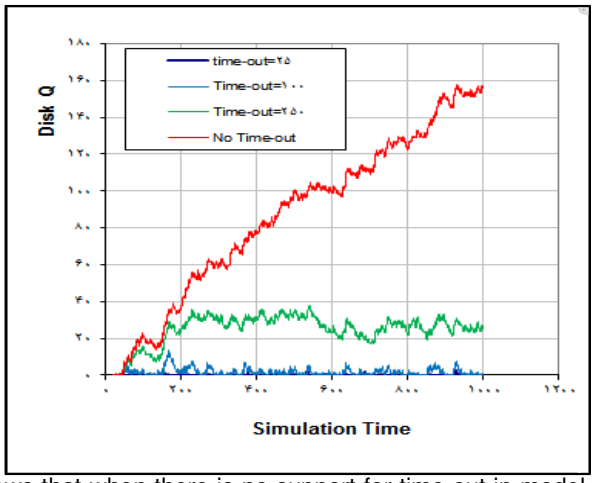

Figure 3. This graph shows that when there is no support for time-out in model, the number of elements in disk queue increase strictly.

Service time ratio of disk is according to Weibull distribution with shape parameter equal to 10 and the mean is adjusted to SD. SD will obtain based on performance parameters with respect to the previous state of the system using of operational low (Zhu et al., 2000) and the formula presented in the Garcia model (Hoxmeier, 2000). SD = DD / VD DD = UD / X and VD = XD / X where X is total throughput and XD is the throughput of Disk. UD is the utilization on Disk.

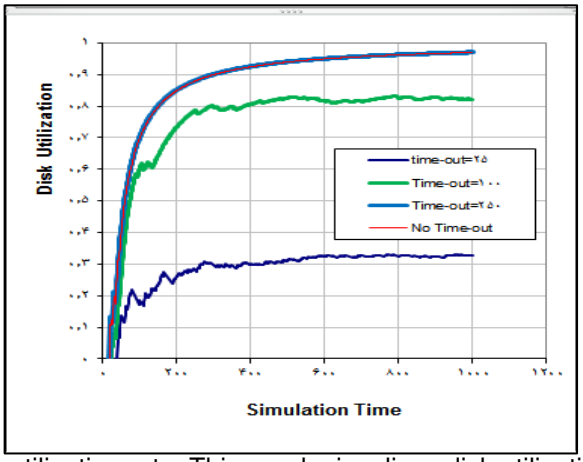

Figure 4. Disk utilization rate. This graph visualizes disk utilization behaviour.

Experimental Results

According to table 1 we can find that a mixture of the first 100 job composed of all transaction has an average of 560 time units for completion. This means that at this time 100 transactions will complete. According to the operational low (Zhu et al., 2000), system throughput can calculate with following equation: X = C / T, where C is the number of completed jobs on time T. so X in the eq.1 at the start of simulation, can set to 0.178571. Similarly, SC, and SD values in the beginning of simulation can be calculated for each transaction class. We simulate this system using Matlab tools. In our simulation applicant will enter to system with entrance ratio m. it means that applicants can login to system with exponential distribution with average of m that called arrival rate. At the beginning timestamp set to applicant for cancellation time. And we study outputs of the system. As can be seen over time, response time of applicants is increased.

At first without time labeling for cancellation time, we analyses system outputs. As you see, when time passed, the response time increased. Similar to Garcia's model, there are some customer waiting in disk queue and it is the reason of rising response time (Garcia, 2010; Lin and Lieh-san, 1989; Zhu et al., 2000). When we look at disk and processor utilization, we found that after a time, disk will service the request with all of its capacity but the processor busy time is only 25%. On the other hand, system througthput after a time period, approach to a special number around 60%. In the second phase of our simulation, we add cancellation time labels to requests and we run our simulation for cancellation time equal to 25,100,250. When to=25, disk utilization decreased to 25% and also system throughput will decrease to 10%. But when

to=100, disk utilization and system throughput respectively approach to 80% and 70%. In the case of to=250, disk utilization is about 100% and system throughput will achieve 70%. In all of these situations, response time will achieve to a constant time while when we did not consider timeout, it will increase for ever.

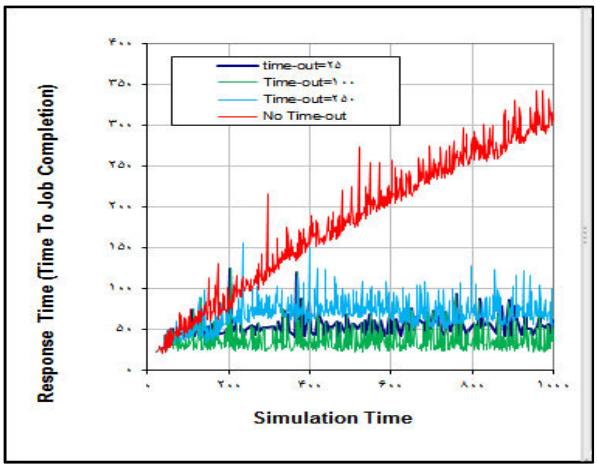

Figure 5. Response time rate. This graph shows when there is no time-out, the response time strectly increase.

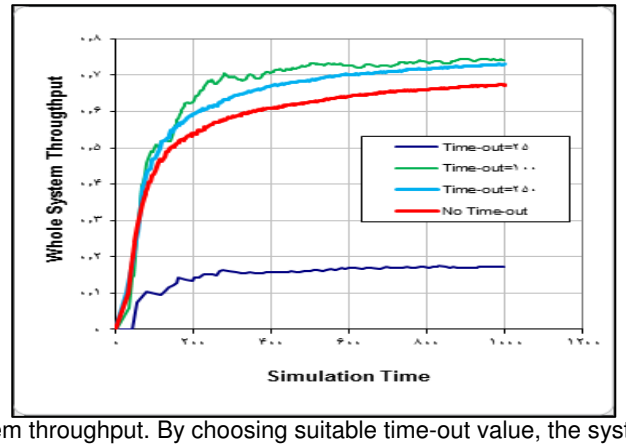

Figure 6. Whole system throughput. By choosing suitable time-out value, the system throughput can be improved.

Then we measure time interval to get service, to determine if user behavior effects on that. With Duncan test (Hinkelmann and Kempthorne, 2007) we analyzed relationship between these four cases. The results are shown in table 2. It can be observed that between no timeout case and to=100 and also between to=25, to=100 and to=250, there are meaningful relationship. These relationships claimed that with selecting a suitable timeout interval, service time will not change.

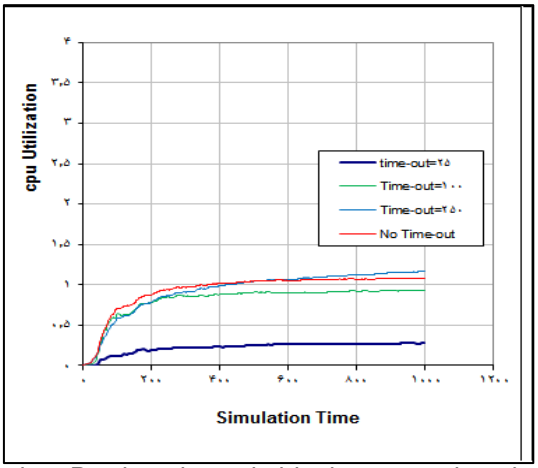

Figure 7. CPU Utilization. By choosing suitable time-out value, the system saves CPU utilization.

But in a special arrival rate, if the cancellation time will be chosen unsuitably, system throughput will increase while there is no change in service time. Response time will approach to a constant while the utilizations of system resources decreased.

Table 2. Results of Duncan test for compare the service time

Experiment Type

Duncan[a,b]

Experiment Type	N	Subset for alpha = 0.05	
		1	2
TO=25	193	1.8951	
TO=250	693	2.2102	
TO=100	726	2.3426	2.3426
No Time-out	618		2.7294
Sig.		.052	.077

Means for groups in homogeneous subsets are displayed.

a. Uses Harmonic Mean Sample Size = 415.806.

b. The group sizes are unequal. The harmonic mean of the group sizes is used. Type I error levels are not guaranteed.

Conclusion

According to obtained results, the following solution can be achieved. First, the impatient user behaviour is a real phenomenon and considering it makes the model to be more realistic. More user cancellation time, more decrease in system performance parameter. With choosing a forced timeout in system, in addition of raise in the system throughput and achieving a constant response time, we can decrease resource utilization while increasing the client satisfaction. Also we can decrease the cost of service delivery due to decrease in resource usage. In the future someone can investigate the cancellation time effects. because when the applicant ignore their request, system resources usage will decrease and it seems that a research in this area can present a suitable solution in user interface design and its effect on database system performance.

References

Altman E., Yechiali U, 2006. Analysis of customers' impatience in queues with server vacations. *Queuing Syst. Theory Appl.* 52:261-279.

Bonald T., Roberts J, 2001. Performance modeling of elastic traffic in overload. *SIGMETRICS Perform. Eval. Rev* 29:342-343.

Bonald T., Roberts J, 2003. Congestion at flow level and the impact of user behaviour. *Computer Network* 42:521-536.

Garcia D. F, 2010. Performance Modeling and Simulation of Database Servers. *OJEEE* 2:183 – 188.

Gromoll H., ROBERT P., ZWART B, 2008. Fluid Limits for Processor-Sharing Queues with Impatience. *Math. Operational Res* 33:375-402.

Hinkelmann K., Kempthorne O, 2007. Design and Analysis of Experiments, Introduction to Experimental Design, Wiley, New York.

Hoxmeier J. A, 1996. System response time and user satisfaction: An experimental study of browser-based applications. *Association of Information Systems Americas Conf.*, pp. 10-13.

Jain R. K, 1991. *the Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, Wiley, New York.

Lazowska E. D, 1984. Quantitative System Performance Computer System Analysis Using Queuing Network Models, Prentice-Hall.

Lin P., Lieh-san, 1989. Modeling and simulation of LAN DBMS performance. *22th annual symposium on Simulation*, p 61--68.

Perel N., Yechiali U. 2010. Queues with slow servers and impatient customers. *European Journal of Operational Research* 201:247 – 258.

Sevcik K.C, 1981. Database system performance prediction, using an analytical model. 1 7th Very Large Database Conference (VLDB'81), Cannes, France, September 1981, P 182-198.

TPC BENCHMARK™ C, 2010. Transaction Processing Performance Council.

Zhu, Y and Lu, K. J, 2000. "Performance Modeling and Metrics of Database-Backed Web Sites. 11th International Workshop on Database and Expert Systems Applications, PP.494.