Journal of Information Technology Management

ISSN #1042-1319

A Publication of the Association of Management

IS DEPLOYMENT OF DISTRIBUTED PROCESSING FOR BUSINESS APPLICATIONS A SOUND MANAGEMENT DECISION: PERFORMANCE ANALYSIS AND SIMULATION

PAUL SAFONOV
SAINT CLOUD STATE UNIVERSITY
safonov@stcloudstate.edu

DENNIS GUSTER
SAINT CLOUD STATE UNIVERSITY
guster@stcloudstate.edu

RENAT SULTANOV
SAINT CLOUD STATE UNIVERSITY
sultanov@bcrl.stcloudstate.edu

NIKOLAI KRIVULIN
ST. PETERSBURG STATE UNIVERSITY, RUSSIA
Nikolai.Krivulin@pobox.spbu.ru

ABSTRACT

Like network connectivity, the volume of data that must be stored and transferred across a network has increased dramatically. This trend has necessitated splitting databases and distributing them across multiple network nodes with potential impact on network performance. There are many ways to predict performance in computer networking applications, but one of the most popular is through simulation because it provides fairly accurate results and does not require the massive hardware resources of a network test-bed. This paper uses simulation to predict the performance gain that might be realized by employing this methodology. The results indicate that distributing a database among multiple nodes can significantly reduce network response delay to client requests, and that having adequate network bandwidth is imperative. It is suggested that further research should focus on the efficiency of the algorithm used to distribute the inquiries among the database nodes.

Keywords: LAN, WAN, Distributed Processing, Networking Performance, Database Performance, Network Simulation.

INTRODUCTION

Distributed processing has been around for over 20 years, but has not dominated the computer landscape, especially in business related applications. There are several reasons for this lack of response. For example, the technology of distributed processing is more mature for scientific applications that are computational intensive. Generally speaking, in the scientific world, there are certain classes of computing problems so complex or massive that they require the resources of more than just a single computer. However, in the business world, the problems tend to be more input-output intensive. This means that the bottleneck occurs on the disk rather than at the CPU level. Therefore, although there is some carryover of basic logic, it is difficult to transfer whole parallel or distributed algorithms directly from science to business, as shown in Guster et al [5].

This in effect means that distributed applications in business will have to evolve independently. This added cost of programming in parallel has led many business software developers to question the value of pursuing distributed processing at all (Johnson [9]. In fact, many proponents of distributed processing, such as Anthes [1], admit that the deployed systems have barely moved beyond scientific, engineering and mathematical/statistical applications. There have been, however, successes at building distributed databases using the client/server architecture. For example, Roussopoulos, Economous & Stamenas [14] developed an advanced data management system at the University of Maryland in 1993. Performance of distributed database client/server systems has recently been examined by Kanitkar & Delis [10]. They determined that distributed databases can offer significant performance advantages if the system is large enough in terms of number of users, and they found that it takes about 40 users to reach this threshold.

Besides the number of users, other variables that influence performance (according to Ghandeharizadeh & DeWitt [4]) are physical layout and speed of network links. In other words, it is critical that the network does not become a communication bottleneck. There have also been recent successes in web-based e-commerce applications. Elnikety et al [3] was able to improve throughput by ten percent and decrease workstation response time by a factor of 14. Furthermore, the potential of distributed databases has been embraced and implemented by vendors. In fact, Townsand & Tsai [17] report in a white paper for Oracle that with distributed processing their database product can now scale to support millions of transactions per minute. Although these works are encouraging there is still a lot of work that is needed in the field. Specifically, Smith et al [15] state that there is a need for more performance evaluation research over more and larger databases.

It therefore seems logical to look at potential benefits in relation to their costs. This cost/benefit ratio is important for all businesses, but it is especially crucial for smaller ones that do not have the massive resources of Fortune 500 companies. According to Buyya [2], the most obvious advantage is speed of processing, although there are others such as improved backups and improved network access. The disadvantages might include these: added hardware cost, added software cost, added programming costs and added management complexity. However, if the main potential benefit (improved processing speed) cannot be realized then the remainder of the discussion is not worth pursuing. Therefore, the primary purpose of this paper will be to test, through simulation, whether or not distributed processing can solve a common business-related problem: speeding up disk I-O by distributing a data base across multiple nodes.

This paper further elaborates on a previous study by Guster, Safonov, Hall, and Sundheim [7], that evaluated the performance of distributed web servers through simulation and experimentation. Specifically, it examined the utilization rates of three different hardware configurations: a single web server, dual web servers, and dual web servers with mirrored file systems.

The results were encouraging: through simulation it was determined that the server utilization rates in the dual server model were about half of the single server model. Furthermore, the overhead generated by the file server mirroring process only about doubled the utilization on each of the dual servers, which matched the level of the single server model. When these models were tested with live systems the results were fairly close. Specifically, the live system was loaded about 10% higher in the single and dual models, but the dual/mirrored values were almost identical. However, the simulation determined that delay to end-user work stations in the dual processor models was about 200 milliseconds longer than with the single server model. This would make sense if one considers the additional communication that must take place in splitting the workflow and the intercommunication between the dual servers.

The ramifications of these findings follow. First, distributed processing can be used to lessen the load on a single server. Second, mirrored file systems can be deployed with acceptable processing overhead. Third, the modeling techniques used by Guster et al [7] offer promise because the output was validated by a real time experiment. Fourth, distributed processing may speed up processing, but because of communication overhead the ultimate performance result (delay to end-user workstations) may not be in the desired direction.

Although our earlier study [7] provided some useful baseline information about distributed processing and methodology, it did not answer the primary question that we put forward in this paper: can the disk I-O performance of a given business application be speeded up by distributed processing? More specifically, what are the performance gains of distributing a database across multiple nodes? Other research such as Elnikety et al [3] has shown the basic model works, but often the data was gleaned from a test bed or used high end networking resources to validate the experimental results. In this study the experimental focus is to determine how well the technology would fit in a typical smaller company that may have a fairly high speed LAN (100Mbs), but is still relying on T1 (1.544Mbs) WAN access.

METHOD AND PURPOSE

The performance of any application making I-O calls to a database is a function of a number of variables. However, there are four basic factors that are easy to measure and control. They are the number of nodes on which the database will be stored, how frequently each database node will be queried, how many workstations will be trying to access the database and at what speed are those workstations are connected.

Theoretically from a queuing perspective, the more nodes the database is stored upon the less a given workstation will need to wait. Like parallel processing, communication overhead minimizes the potential advantage to something less than a linear progression, as demonstrated by Sultanov and Guster [16], and Guster and Madison [8]. Under ideal conditions one would like to have the probability that any one database request will be evenly distributed among the processor nodes supporting the distributed database activity. This may be possible if each database node is storing a replica of a single database, but less likely in a relational split. Although the replica model may work fairly well for relatively static databases, it is severely limited by the overhead needed to synchronize the replicas in dynamic applications.

Therefore, for applications involving large number of updates splitting the entire database into subparts and storing each subpart on a separate processing node is attractive. In this situation it is very difficult to split the database in such a way that access to it can be equally divided among the nodes allocated (symmetrically split). However, in splitting a database one expects some kind of interaction with network access. That interaction may occur among the database nodes or in the manner end user work stations gain access to the database cluster.

Generally speaking, an end user workstation will gain access in one of two ways. Either they will be connected via a LAN or remotely through a WAN. More than likely there will be vast speed difference between the two. Based on commonly deployed technology one would expect the LAN to offer speeds in the 100Mbs range and the WAN in the 1.5Mbs range. So therefore it is important to test any distributed data base model in regard to its suitability for both LAN and WAN based end-user workstations.

In this study, a business oriented LAN/WAN will be programmed through simulation and the following will be varied: number of nodes storing the database, number of workstations accessing the database, the speed at which the workstations are connected and the probability of any given request being sent to a given database node. The data collected in these simulations will be used to test the following null hypotheses:

H1. The number of nodes a database is stored upon has no effect on the responses back to the initiating workstations.

H2. The number of end-user workstations making inquiries to a distributed database has no effect on the delay of the responses back to the initiating workstations.

H3. The probability distribution that controls inquiry assignment to database nodes has no effect on the delay of the responses back to the initiating workstations.

H4. Whether an end user work station is connected to a LAN or a WAN causes no difference in the delay of the responses back to the initiating workstation.

For Hypothesis 1 one would expect that as the number nodes increases, delay back to the workstation would lessen. However at some point it would be expected that the communication overhead would result in a point of diminishing returns. This might give rise to an inverted U-shaped relationship, but probably not at just the 8 node level. Within Hypothesis 2, it seems logical to expect that as the number of workstations is increased delay will increase as well. The key however, will be to be able to use the simulation process to identify where potential bottlenecks might occur. For Hypothesis 3, given the simplicity of the problem one would probably expect the symmetrical algorithm to be superior. However, it may be questionable whether the round robin formula employed in the simulation will adequately challenge the distribution algorithm. In the last Hypothesis one would expect that the LAN would offer superior performance. However, it would be expected that the major factor would be the inherent line speed differences.

The software for the basic simulation was written to mimic end-user workstations coming from a WAN or LAN and making inquires to a single logical database (which may be stored on multiple nodes). The splitting of data was not done through replica's, rather it was done be breaking the data base in to relational subparts. A drawing of these basic models is included in Figures 1 and 2.

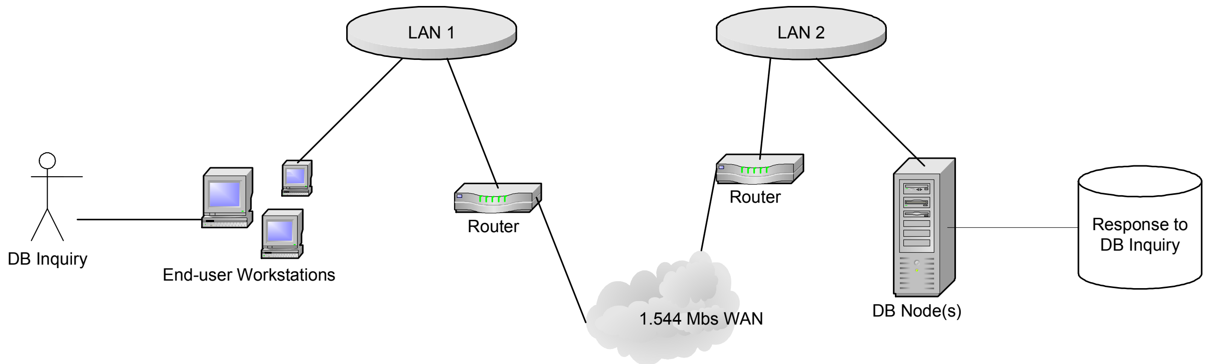

Figure 1: WAN Model Used in the Simulation

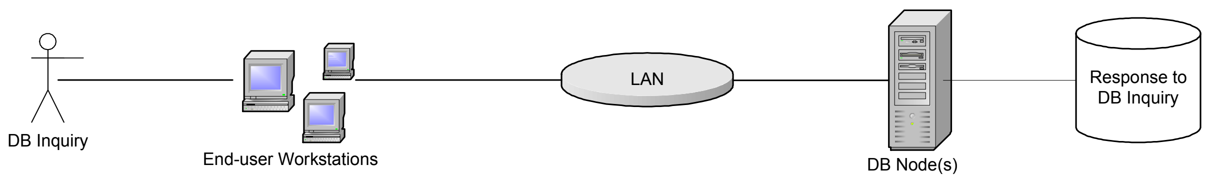

Figure 2: LAN Model Used in the Simulation

The basic model is rather simplistic. For the WAN based model, a database inquiry is generated on an end-user workstation. The inquiry in turn is sent over a LAN to a router, which sends it across a WAN to another router. After traveling via a LAN and arriving at one of the database node(s), the inquiry is processed and the response sent back to the end-user. For the LAN based model, the workstations are connected to the LAN that contains the data base nodes. A series of simulations were undertaken by varying this basic model in regard to the number of end-user workstations, the number of database nodes, the speed the nodes were connected and the probability distribution of any given inquiry being routed to a given database node.

The simulation was programmed with *Network II.5* software. This point and click tool allows the programmer to select various network devices and connect them together with a variety of network architectures and uses standard queuing theory to determine network performance delay.

The critical part of using this software is determining packet inter-arrival distributions. This is difficult because some research, like Partridge [13], has shown that packet-inter-arrival rates often do not follow any know distribution. Krzenski [11] shows that it is critical that the correct distribution be utilized; otherwise, the validity of the simulation could well be suspect. To ensure a representative workstation distribution was utilized in the simulation, samples of packet data from a distributed database application were recorded on the authors' network and converted to a tabular distribution (a table that provides the probability of any given packet inter-arrival time occurring). This tabular distribution is listed below in Table 1. Note that this is the probability for packet inter-arrivals not for the database inquiries. In fact an inquiry would generate multiple packets. The representativeness of these values to other systems is not known as are the transferability of the results contained herein. Fortunately the process is easily transferrable to other systems. It is just a matter collecting the appropriate packet data from the target system and converting it to a tabular distribution.

But first, the primary goal of this study -- to determine if the modeling technique yields useful results -- must be realized. Ex[SE1]perimental runs from 10 to 90 seconds were tried. It was found that the initial setup procedure within the system tended skew workstation delay values. So therefore, a one second warm-up period was added to the simulation process. Further, fairly consistent results were obtained at the 45 workstation level with 10, 30 and 90 second simulation runs. However, while at the

100 workstation level the CPU and network (LAN) utilization remained fairly consistent, the workstation delay doubled when the run time was increased from 10 to 90 seconds. This value increased at an even quicker rate at the 250 workstation level (2.5 times larger from 10 to 30 second runs), but again the utilization rates remained fairly constant.

At first, the lack of consistency within the 100 and 250 workstation levels was a concern, but an analysis of the results revealed that the WAN link in each case had reached almost 100% utilization, which after several seconds of the run created an ever increasing backlog. To see if consistency could be reached, the line speed of that link was increased from a T1 (1.5Mbs) to a T3 (40Mbs). The results indicated that there were consistent readings as the run time was increased from 10 to 30 seconds at the 250 workstation level. Furthermore, even at the 10 second run level in excess of 5,000 packets were generated. When coupled with the warm up, this number would tend to provide further support for the assumption that a steady state was attained. Therefore, all data reported are based on a one second warm-up and a ten second simulation period.

Table 1: Tabular Distribution of Packet Inter-Arrival Times

Probability	**Time (in seconds)**
.357	0.1
.588	0.2
.752	0.3
.806	0.4
.862	0.5
.885	0.6
.889	0.7
.914	0.8
.923	1.0
.942	1.2
.949	1.4
.953	1.6
.956	1.8
.957	2.0
.959	2.5
.962	3.0
.964	3.5
.967	4.0
1.000	10.0

RESULTS

The data that was used to evaluate Hypothesis 1 (*The number of nodes a database is stored upon has no effect on the delay to the end-user workstations*) is displayed in Table 2. This data indicates that splitting the data base across multiple nodes reduces the delay of the response to the workstations. In the case of the WAN example it was reduced by a factor of about 4 (374/86). In the case of the LAN example it was reduced by a factor of 36 (1986/55). The difference in these factors can be explained by the line speed associated with each of these architectures. The WAN connected workstations were constricted to 1.5Mbs while the LAN workstations were provided with a 100Mbs link to the database nodes. The link utilization rates provide some perspective about the findings in Table 2. For the WAN, link utilization averaged about 50% while for the LAN the utilization rate was only about 3%. Thus, Hypothesis 1 is rejected.

Table 2: Average Delay to the Workstation in Milliseconds
(run on 1.2 GHz database nodes). 45ws WAN vs. 100ws LAN

Type	1 node	2 nodes	4 nodes	8 nodes
45ws WAN	374	357	108	86
100ws LAN	1986	216	93	55

Table 3 provides the data that was used to evaluate Hypothesis 2 (*The number of end user workstations making inquiries to a distributed database has no effect on the delay of the responses back to the initiating workstations*).

Table 3: Average Delay to the Workstation in Milliseconds
(run on 1.2 GHz database nodes). 45ws WAN vs. 100wsWAN vs. 250ws WAN

Type	1 node	2 nodes	4 nodes	8 nodes
45ws WAN	374	357	108	86
100ws WAN	2385	2116	2071	1554
250ws WAN	5008	4972	4892	4840

In all cases as the number of workstations increased so did the message delay of the message response to the workstation. For all node columns, the delay increased by at least a factor of 14 from the 45 to the 250 workstation level. Furthermore, as workstations are added, the efficiency of splitting the data across multiple nodes lessens. Specifically, it varies from a factor of about 4 (374/86) with 45 workstations to 1.5 at 100 workstations and culminates at 1.03 at the 250 workstation level. Once again this disparity can be explained by the WAN link utilization rates which were: 50% with 45 workstation, but 98% at both the 100 and 250 workstation levels. Hence, Hypothesis 2 is rejected.

The data used to evaluate Hypothesis 3 (*The probability distribution that controls which inquiry will be assigned to which database node has no effect on the delay of the responses back to the initiating workstations*) is displayed in Table 4.

Table 4: Average Delay to the Workstation in Milliseconds
(run on 500 MHz database nodes) Symmetrical vs. Asymmetrical Allocation Strategies

Type	2 nodes		4nodes		8 nodes	
	Sym	*Asym*	*Sym*	*Asym*	*Sym*	*Asym*
10ws WAN	525	505	430	386	315	347
100ws WAN	7610	6423	5207	4438	2698	2115
1000ws WAN	11619	11244	10437	10877	10416	10258

Varying the distribution revealed mixed results, the symmetric algorithm (contains equal probabilities such as 50/50 for a 2 node model) was not always the most efficient. In fact, in the majority of cases the asymmetric model (contains unequal probabilities such as 80/20 for a 2 node model) was superior. Specifically, there were mixed results at the lightly and heavily loaded levels, but consistent results at the moderately loaded level in favor of the asymmetric model.

There are several possible explanations for these unexpected results. Perhaps, the random generator distribution process within the simulation inadvertently created a sampling error. It was suspected that link utilization might have an effect since the WAN utilization for both the 100 and 1000 workstation simulations was very high, but the load at the 10 workstation level was less than 10%. A similar test was performed on a 100Mbs LAN with 10 workstations and a similar pattern evolved. Certainly, based on the data herein the magnitude does not appear sufficient to warrant the extra complexity. However, this was a very simple problem with a round-robin allocation scheme. There may be, however, more complex problems with characteristics that might lend themselves to an asymmetrical distribution. Investigation of such problems is beyond the scope of this exploratory paper, but the authors recognize the need for such research. The equivocal and unsystematic results do not support rejection of Hypothesis 3.

The data used to evaluate Hypothesis 4 (*Whether an end user workstation is connected to a LAN or WAN causes no difference in the delay of the responses back to the initiating workstation*) is displayed in Table 5. In both cases, LAN and WAN, splitting the problem across multiple nodes reduces delay. However, the efficiency of the LAN is dramatically better. Once again this can be attributed to the difference in link speed and the resulting differences in link utilization 3% for the LAN and 98% for the WAN. In fact, to test this assertion the WAN link speed was doubled in the 8 node experiment to about 3Mbs this reduced the utilization down to about 60% and the average delay to 73 milliseconds. Our results support rejecting Hypothesis 4.

Table 5: Average Delay to the Workstation in Milliseconds
(run on 1.2 GHz database nodes). 100ws WAN vs. 100ws LAN

Type	1 node	2 nodes	4 nodes	8 nodes
100ws WAN	2385	2116	2071	1554
100ws LAN	1986	216	93	55

Analysis of the Hypotheses

H1: The data from Table 2 does not support this null hypothesis. Adding data base nodes resulted in improved performance in all cases presented.

H2: This null hypothesis is not supported by the data in Table 3. It was clear that adding workstations increased delay in all categories.

H3: This null hypothesis is supported by the data in Table 4. There was no clear superiority, in comparisons of the symmetric and asymmetric algorithms.

H4: The data from Table 5 does not support this null hypothesis. There is a clear advantage in delay in favor of the LAN (of course, this is primarily based on line speed).

CONCLUSIONS AND RECOMMENDATIONS

Although distributed processing offers exciting possibilities, it definitely has limits. Our results make it clear that splitting a database across multiple nodes can offer distinct advantages. However, the results suggest splitting has to be done on a network that is properly loaded. If the intensity of the database inquiries is too light, it is not worth the extra overhead of distributing the database. However, if the network is too heavily loaded by this traffic an interaction between network and database access times may occur resulting in equivocal results and little or no gain from distributing the database. On the moderately loaded WAN an 831 ms gain was observed. This could have a positive influence on a number of business functions: such looking up a product on an e-commerce site or conducting a credit card transaction at a retail store. Furthermore, these results reinforce the need to treat the entire IT design process as a coordinated effort. Without the proper network configuration the results of implementing distributed databases may be unpredictable. This paper looks at a fairly complex queuing theory problem; in this case speeding up one part of the system may not result in a gain. In fact, in some situations distributing databases may overload another part of that system and result in a performance loss.

The results scaled well when additional database nodes were added within light- and moderately-loaded systems. However, there is a tradeoff in adding additional nodes. Given the wealth of research in parallel processing in which inter-processor communication was a limiting factor, one would expect it would also be a limiting factor in distributing databases as well.

It would be expected that a point of diminishing returns would be reached where the overhead of managing 16 or 32 database servers is not trivial. At this point, further distribution should be considered only if a substantial performance gain can be realized – especially if that gain translates to financial gain as well. One specific objective might be trying to find an optimum (maximum recommended load level) value for the WAN. From the results we obtained, it is likely that the value is somewhere around fifty workstations (based on the author's tabular distribution). For example, adequate performance was obtained for the 45 workstation experiment. When the WAN link was doubled (such as in 2 load balanced T1s) in the 100 workstation experiment, the performance specifications were similar to the 45 workstation experiment. Hence, based on the parameters of this simulation one can conclude it takes approximately 1.5Mbs of bandwidth per 50 workstations.

These results raise a very interesting question about the feasibility of implementing a distributed database in a WAN world. In actuality, it is more of a line speed question than LAN versus WAN. In this study the commonly used architecture for each category selected was 100Mbs for the LAN and 1.544Mbs for the WAN. The question is this: if the line speeds were similar, would the LAN and the WAN perform at the same level, even though one expects more propagational delay on the WAN level? Thus, the real question becomes whether a business contemplating employing a distributed database can afford the bandwidth necessary to make it work. In the LAN world, 100Mbs is already widely deployed and based on our results will provide most adequate performance. The additional cost within a LAN would be minimal -- just connections for the database nodes. However, in the WAN world, an upgrade from a 1.544Mbs WAN is a major expense. For example, upgrading a 100 mile point-to-point link from 1.544Mbs to 44.736Mbs would jump from about $1,300 to $24,000 a month, as indicated by [12]. This certainly is an expense that only fairly large companies could justify. However, if bandwidth requirements are modest, then multiple T1s in a load balance format could be used. In the example cited earlier, a 100 workstation WAN could obtain the required 3Mbs bandwidth for $2,600 a month.

A number of points raised by this research should spur subsequent investigation. Because this study only included up to eight database nodes, it is not possible to say how the problem would scale beyond that point. Therefore, more definitive results could be obtained by extending the number of nodes to 16 or even 32. Also, providing more than just 3 categories in the number of workstations would be useful.

More research is needed on the symmetric versus asymmetric distribution issues. Our data provides inconclusive evidence. Perhaps more sophisticated simulations should be undertaken. Determining the optimal distributed database allocation algorithm and the rest of the study should be replicated on a computer test-bed in which everything is programmed into the various hardware elements, but the workload is generated by script file instead of actual end users (and is thus controllable). Depending on those results additional recommendations could be made to guide companies contemplating a production distributed database model. The results of our research presented here and subsequent studies could be used to help companies model their own distributed database network using their specific workload/server parameters resulting in objective decision making data.

However, these preliminary results are encouraging. First, in one experiment, delay was reduced by a factor of 36 by splitting the data across 8 nodes. Second, a bandwidth requirement of 50 workstations per 1.5Mbs of bandwidth was identified for the load levels tested. Last, and most important, the simulation method we devised could serve as an excellent tool to support what-if scenarios., For example, what if the company expanded to 500 workstations? Different link options, database node levels, and distribution algorithms could be tested to find an acceptable solution. Therefore, it appears that simulation can provide useful objective performance data when planning the design of a distributed database system.

REFERENCES

[1] Anthes, G. (2003). "Grids Extend Reach", *Computerworld*, 37(41), pp. 29-30.

[2] Buyya, R. (1999). *High Performance Cluster Computing* , Prentice-Hall.

[3] Elnikety, S. et al. (2004). **"**A method for transparent admission control and request scheduling in e-commerce web sites", Proceedings of the 13th international conference on World Wide Web*,* pp. 276-286.

[4] Ghandeharizadeh, S. & DeWitt, D. (1994). "MAGIC: A Multiattribute Declustering Mechanism for Multiprocessor Database Machines", *IEEE Transactions on Parallel and Distributed Systems*, 5(5).

[5] Guster, D., Sultanov, R & Chen, Q. (2003). **"**Adaptation of a Parallel Processing Technique Used to Solve a Physics Problem to a Computer Network Management Application**"**, *Proceedings of the Information Resources Management Association International Conference*, Philadelphia, PA.

[6] Guster, D., Sohn, C., Robinson, D. & Safonov, P. (2003). "A Comparison of Asynchronous Transfer Mode (ATM) and High Speed Ethernet and the Network Design Implications to a Business Organization", *Journal of Information Technology and Decision Making*, 2(4).

[7] Guster, D., Safonov P., Hall C., Sundheim R. (2003). "Using Simulation to Predict Performance Characteristics of Mirrored Hosts Used to Support WWW

Applications", *Issues in Information Systems*. 4 (2), 2003.

[8] Guster, D. and Madison, D. (1991). "Integrating Parallel Processing into the Microcomputer Education Curriculum**",** A paper presented at the Midwest Decision Sciences Institute. Indianapolis, IN.

[9] Johnson, M. (2003). "Gridlock Reality", *Computerworld*, 37(41), p 24.

[10] Kanitkar, V. & Delis, A. (2002). "Distributed Query Processing on the Grid", *IEEE Transactions on Computers*, 51(3), pp. 269-278.

[11] Krzenski, K. (1998). "The Effect of Varying the Packet Interarrival Distribution in the Simulation of Ethernet Computer Networks", Unpublished graduate research project, St. Cloud State University.

[12] http://www.bandwidthsaving.com/T3prices.cfm

[13] Partridge, C. (1993). "The End of Simple Traffic Models**"** (Editor's Note), *IEEE Network*, 7(5).

[14] Roussopoulos, N., Economous, N. & Stamenasm, A. (1993). "A Testbed for Incremental Access Methods", *IEEE Transactions on Knowledge and Data* Engineering, 5(5) pp.762-774.

[15] Smith, J. (2003). "Distributed Query Processing on the Grid", *International Journal of High Performance Computing Applications,* 17(4), pp.353-367.

[16] Sultanov, R. and Guster, D. (2003). "Parallel Computing for Semiquantal Few Body Systems in Atomic Physics", *Springer-Verlag Lecture Notes in Computer Science.* 2667, pp. 568-576.

[17] Townsand, M. & Tsai, J. (2003). "Oracle 9i New Features", Oracle Corporation, Redwood City, CA.

AUTHOR'S BIOGRAPHY

Paul Safonov earned his Ph.D. in applied mathematics at the Russian Academy of Sciences in the Institute of Control Sciences, Moscow in 1995. He is Assistant Professor of Business Computer Information Systems at St. Cloud State University, Minnesota, USA since 2002. He was visiting scientist and professor in Center for European Economic Research, Mannheim, Germany, University of Warsaw, Poland, University of Campinas, Brazil, University of Versailles, France, and lately worked as Senior Researcher in Free University of Brussels, Belgium. His interests include simulation and optimization methods, computer modeling and information systems in ecological economics, transportation, and computer networking.

Dennis Guster earned his Doctorate at the University of Missouri, St. Louis in 1981. He has 25+ years teaching experience in higher education and has been Professor in the Computer Science and Statistics Departments before joining the Business Computer Information Systems Department at St. Cloud State University in 2001. Dennis has served as a consultant and provided industry training to organizations such as Compaq, NASA, DISA, USAF, Motorola, and ATT. As Director of the Business Computing Research Laboratory at SCSU he has organized numerous sponsored research projects and has published in the areas of network design, network performance analysis, and computer network security.

Renat Sultanov currently is Research Scientist at the Business Computer Research Laboratory at St. Cloud State University, MN. He obtained his Ph.D. in computational nuclear physics in 1990, worked in Bonn University as research fellow of the Alexander von Humboldt Foundation, Bonn Bad-Godesberg, Germany, and thereafter held research positions in the University of Sao Paulo, Brazil, Texas A&M University in College Station, and in the University of Nevada, Las Vegas.

Nikolai Krivulin is Associate Professor in the Department of Mathematics and Mechanics at St. Petersburg State University, Russia, where he obtained his M.Sc. in operations research and Ph.D. in computational mathematics and optimal control. In 1999-2002 he was Chair of the Department of Information Management of the School of Management at St. Petersburg State University. His research interests include mathematical modeling and simulation of computer information systems and networks. He is an author of more than 50 papers in journals and conference proceedings. A member of St.Petersburg Mathematical Society, AMS, and SIAM.