

Mean Value Analysis of a Database Grid Application

D. R. Thompson and S. Shin
drt@uark.edu

Computer Science and Computer Engineering Department
University of Arkansas, Fayetteville, AR, U.S.A.

Abstract

The mean value analysis algorithm is used to model a database application that runs over a grid of computers. A bound on the maximum throughput is modeled by assuming a uniform distribution of requests across the grid. Then, a non-uniform distribution of requests was modeled. Also, the number of clients was varied. Then, the model was modified to reflect a proposed change in the application. Finally, the batch and interactive requests were segregated to improve the response time.

1. Introduction

An application is analyzed using the Bard-Schweitzer approximate mean value analysis (MVA), which is an analytical model. The application is a database that is distributed across a grid of computers. Each computer is devoted to a portion of the database for doing processing. Therefore, a computer is devoted to this application and the amount of overhead processing is minimal.

1.1. Database application

A high-level diagram of the database application is shown in Figure 1. Each client sends a block of one or more records to a grid of directors. The director grid is a designated group of computers from a grid of computers. The computer in the director grid servicing the particular client receives the block of records and splits the records according to a key. The director then sends each record to the portion of the database grid that is servicing the key of particular the record. The database grid is logically partitioned to serve different keys and there are 70 computers in the database grid. The Common Object Request Broker Architecture (CORBA) is used for application-to-application communication. CORBA is middleware software that allows application programs to communicate independent of the particular hardware, software, and networking platforms.

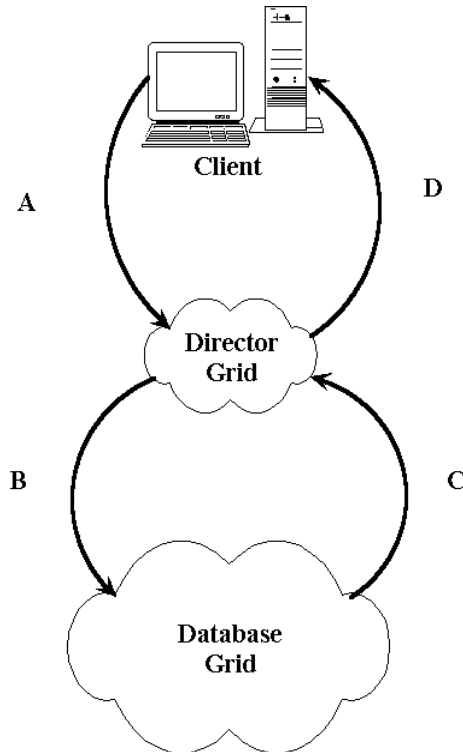


Figure 1 Database application

1.2. Background

A system can be represented as network of queues to analyze its performance. Each queue represents a service center such as a CPU or a network device. The performance of a queueing network is measured by the throughput and mean delay. The throughput is the average number of customers serviced per unit of time such as records per second. The mean delay is the average delay per customer.

There are two general types of queueing networks [1]. In an open queueing network, customers arrive to the system from the outside, receive service, and then leave the system. In a closed queueing network, the number of customers remain constant in the queueing network and circulate through the system of queues. It is common to analyze a system as a closed queueing network even if it is

an open queueing network because there are efficient analysis techniques. A closed queueing network is used to model the application.

Buzen developed a technique called convolution to solve a closed queueing network exactly, but it is prone to numerical instabilities and does not scale well to a large number of queues [2]. Then, Reiser and Lavenberg developed an efficient method to calculate the mean queue lengths at each of the devices to determine the performance of the system [3]. This method is called the exact mean value analysis (MVA). It calculates the mean queue lengths, mean throughput, and mean delay per record. But, if the number of classes is greater than 10 or for large populations per class it is not practical to use because of execution time and memory requirements [4].

Bard presented the first approximate MVA [5]. Schweitzer then developed the most popular approximate MVA sometimes called the Bard-Schweitzer proportional estimation algorithm [6]. The Bard-Schweitzer algorithm is practical for networks up to 100 classes and about 1000 queues [4]. Its maximum queue length error can be as high as 15% [4]. An empirical study found that the maximum throughput and system response time (mean delay) error for a system with populations between 2 and 10 was less than 3% [4]. For a system with populations between 20 and 100, the maximum throughput and system response time error was less than 1%.

Reiser analyzed the window flow control algorithm used in computer networks with the Bard-Schweitzer algorithm [7]. Each session was treated as a separate class of customers. The number of customers in each class was equal to the window size of each session. In this paper, this technique was used to model the database application. Each client was treated as a different class and the number of customers in each class was set to the block size used by the client. The block size is the number of records that is submitted by the client.

2. Model

The database application was modeled with the queueing network shown in Figure 2 and solved with the Bard-Schweitzer algorithm. This model is used to capture the important characteristics of the system. In the model, each client is represented as a separate class of customers each with the number of customers in the class equal to the block size. The block size is the number of records that is submitted by the client. For example, twenty clients with a block size of 1150 are represented by twenty classes with each class having 1150 records that constantly circulate through the system.

A client is assigned to a computer called a director. If the number of directors is less than the number of clients, then the clients are assigned uniformly. The directors route each record to the specific database grid computer that is assigned to a key of the database. The

database grid computer services a record by modifying it and sending the record back to the director. The director then sends the record back to the client. So the flow of the records for a single client is from the client to the director, from the director to the database grid, from the database grid back to the director, and from the director back to the client. Each client or class may share a director and associated network and do share the database grid and associated network.

2.1. CPU and network demand

The Bard-Schweitzer algorithm requires the demand per record to calculate the mean number of records in the queue, the mean throughput, and the mean delay. The demand is defined as the average time to service a record given that a record is at a queue times the average number of times that the record visits a queue [1]. The record size is 500 bytes. Also, the data link layer (Ethernet), network layer (Internet protocol (IP)), and the transport layer (transmission control protocol (TCP)) each add overhead to a record to transmit it over the network. The standard amount of overhead for Ethernet, IP, and TCP, is 26, 20, and 20 bytes, respectively. Therefore, the mean service demand at a CPU was set to 4.528×10^{-6} seconds/record (s/record) based on the approximate total size of 566 bytes and the CPU speed of 1 GHz. This service demand was used for the clients, the computers in the director grid, and the computers in the database grid.

$$(566 \text{ bytes/record})(8 \text{ bits/byte})(1/(1 \times 10^9 \text{ bits/s})) = 4.528 \times 10^{-6} \text{ s/record}$$

The mean service demand of the network was set to 4.528×10^{-5} s/record based on the approximate total size of 566 bytes and a network rate of 100×10^6 bits per second (100 Mbps). All networks were assumed to be 100 Mbps since the majority of the network cards in the grid are 100 Mbps Ethernet.

$$(566 \text{ bytes/record})(8 \text{ bits/byte})(1/(100 \times 10^6 \text{ bits/s})) = 4.528 \times 10^{-5} \text{ s/record}$$

Bottleneck analysis was used to approximate the maximum obtainable throughput of the system. It was calculated by inverting the resource that has the maximum demand, which in this system is the 100 Mbps Ethernet network. The maximum attainable throughput is equal to 79.5 Mega records/hour.

$$(1/(4.528 \times 10^{-5} \text{ s/record})) \times (3600 \text{ s/hr}) / (10^6) = 79.5 \text{ Mrecords/hr}$$

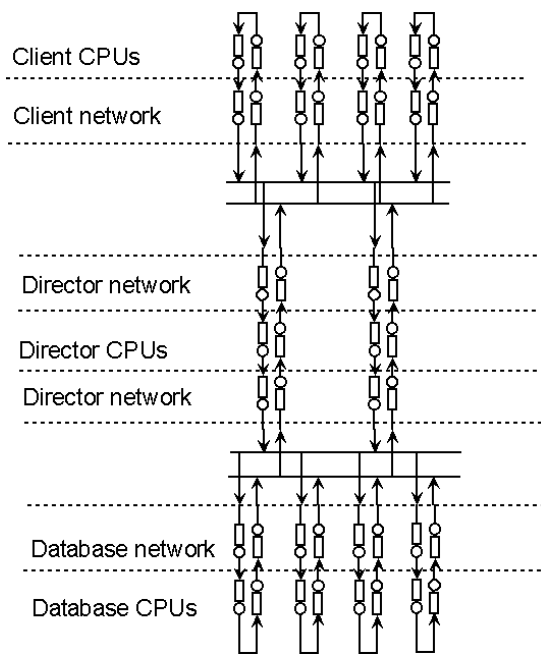


Figure 2 Queuing model

2.2. Block size

The clients send blocks of records to the director grid for processing. Most of the requests are batch. Presently, the block size for the batch class is set to 1150 records, which was determined empirically. Intuitively, the block size needs to be large enough to overcome the overhead associated with CORBA. In the model, the block size was varied to determine its affect on the performance. Initially the efforts were focused on improving the throughput, but it is also recognized that these large blocks of records will affect the mean delay of the interactive clients that are sending blocks of one record. Separate directors are used to serve interactive clients to help avoid this problem. Therefore, the system was modeled with some of the clients sending blocks with 1150 records and some sending blocks with one record to demonstrate the tradeoff between high throughput and low delay.

3. Scenarios

In this section, several scenarios are described. The parameters of the system are varied to show their affect on the performance of the system. First, a uniform distribution of records is assumed to determine the maximum throughput of the system. Second, the distribution of records submitted to the system is changed from a uniform distribution to a non-uniform distribution. The non-uniform distribution degrades the performance of the system. Then, the number of clients is varied to demonstrate that the number of directors should be approximately equal to the number of clients. Then, a

proposed change to the database application is modeled that does not appear to significantly decrease the performance. Finally, the existing production system is modeled. In this model, the batch and interactive records are segregated at the director level.

3.1. Uniform distribution

In the first model, a uniform distribution of records was applied to the database application to determine a bound on the system. This should give the maximum throughput of the system. Each record was equally likely to go to any of the computers in the database grid. So the demand for the database grid computers was set to 4.528×10^{-6} s/record divided by the number of computers in the database grid. The number of clients was set to 20 and the number of computers in the database grid was set to 70. Then the block size was varied for a system with 10, 15, and 20 directors. The clients were assigned uniformly to the directors. Therefore, two clients were assigned to each director when there were 10 directors and 20 clients. The demand for the director grid computers was set to 4.528×10^{-6} s/record to model the configuration that each client is assigned to a single director. The overall throughput of the system in Mega records/hour (Mrecords/hr) and the mean delay per record in seconds/record (s/record) for varying block sizes are shown in Figures 3 and 4.

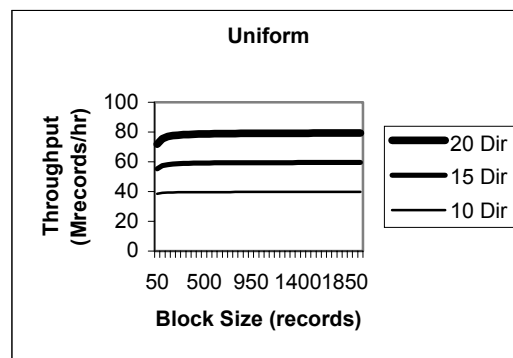


Figure 3 Throughput

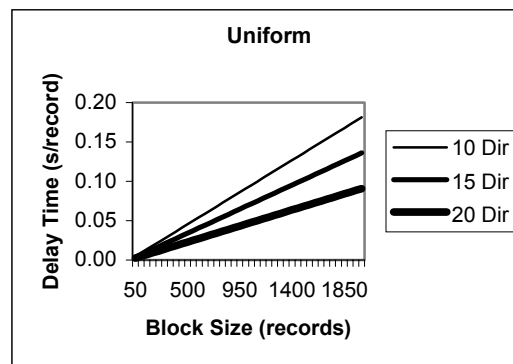


Figure 4 Mean delay per record

Notice that the maximum throughput predicted by the bottleneck analysis of 79.5 Mrecords/hr is obtained for 20 directors since there are 20 clients. The routing of the records from the clients through the directors restricts the throughput if the number of directors is less than the number of clients. The throughput is limited by the 100 Mbps Ethernet network that is serving the directors, not the directors. The capacity of the network at the director grid and the database grid would need to be increased to overcome this bottleneck.

3.2. Distribution of records

Another important parameter that affects the performance of the system is the distribution of records that are sent to the database grid. With the database grid, it is more efficient if the clients submit records that are uniformly randomized. If all clients distribute their requests uniformly across the database grid, the overall efficiency of the system improves.

To demonstrate this affect on the performance, a uniform distribution of demands was compared with a non-uniform distribution of demands both using a block size of 1150. The non-uniform distribution of demands was created by assuming that 80% of the requests from clients (16 clients) went to 20% of the database grid (4 computers). The remaining 20% of the requests (4 clients) went to the remaining 80% of the database grid (56 computers). The overall throughput of the system and the mean delay per record for the uniform and non-uniform demands with 10, 15, and 20 directors are shown in Table 1.

There are several important results represented in Table 1. First, since there are 20 clients that are assigned to directors, the only way to reach the maximum throughput of 79.5 Mrecords/hr is to have the number of directors be equal to the number of clients. For a system with 10 and 15 directors, the non-uniform traffic does not affect the throughput or mean delay when compared with the uniform traffic. This is caused by the restriction of having 20 clients all share less than 20 directors and more importantly 20 network connections. The bottleneck device in this system is

Table 1 Throughput and delay for uniform and non-uniform demands with block size of 1150

Throughput (Mrecords/hr)	10 Directors	15 Directors	20 Directors
Uniform	40	59	79
Non-uniform	40	59	71
Delay time (s/records)	10 Directors	15 Directors	20 Directors
Uniform	0.1043	0.0783	0.0523
Non-uniform	0.1043	0.0783	0.0581

the 100 Mbps Ethernet network. The non-uniform traffic does not affect the performance until the number of directors approaches the number of clients (20 clients). With 20 clients and 20 directors the non-uniform traffic causes the throughput to drop from 79 Mrecords/hr to 71 Mrecords/hr and the delay to increase from 0.0523 s/record to 0.0581 s/record.

3.3. Number of clients

Then, the performance of the system with 20, 40, and 60 clients with the batch class was analyzed. A uniform distribution of records to the database grid is assumed and the clients were assigned uniformly. The number of computers in the database grid was set to 70. The performance with 10, 15, and 20 directors was calculated. The overall throughput and mean delay per record for block size 1150 with 10, 15, and 20 directors are shown in Table 2.

Notice how the throughput decreases significantly when more clients are added to the system. It may seem obvious that adding more clients reduces the throughput because of the added requests, but the maximum theoretical throughput is still 79.5 Mrecords/hr as predicted by the bottleneck analysis. Examine the system with 20 directors as shown in Table 2. The maximum obtainable throughput of 79.5 Mrecords/hr is nearly reached when there are 20 clients and 20 directors. But when the number of clients is doubled from 20 to 40, the throughput decreases by approximately 50% from 79 to 40 Mrecords/hr and the delay doubles from 0.05 to 0.10 s/record. When the number of clients is increased from 20 to 60, the throughput decreases by approximately 30% from 79 to 26 Mrecords/hr and the delay triples from 0.05 to 0.16 s/record. It is possible to increase the throughput of the systems with 40 and 60 clients if the number of directors is increased to be approximately equal to the number of clients.

Table 2 Throughput and delay for 20, 40, and 60 clients with block size of 1150

Throughput (Mrecords/hr)	20 Clients	40 Clients	60 Clients
10 Directors	40	20	13
15 Directors	59	30	20
20 Directors	79	40	26
Delay time (s/record)	20 Clients	40 Clients	60 Clients
10 Directors	0.1043	0.2084	0.3126
15 Directors	0.0783	0.1433	0.2084
20 Directors	0.0523	0.1043	0.1564

Table 3 Throughput and delay comparison between original and proposed application

Throughput (Mrecords/hr)	10 Directors	15 Directors	20 Directors
Original Application	40	59	79
Proposed Application	39	57	77
% decrease	2.50	3.32	2.49
Delay time (s/record)	10 Directors	15 Directors	20 Directors
Original Application	0.1043	0.0783	0.0523
Proposed Application	0.1095	0.0809	0.0549
% increase	4.98	3.32	4.97

3.4. Proposed change to application

There was a proposed change to the database application in which a small percentage of the records required service from two different database grid computers. In the proposed application change, the record from a client was routed to one of the directors as before. Then, if the record belonged to a special class, the record was sent to two database grid computers. It was estimated that approximately 1% of the records would cause the directors to send a record to two database grid computers. This proposed change was modeled by having 5% of the clients (1 client out of 20) require demand from two database grid computers. This is worse than the 1% that was predicted. In this model, there were 20 clients, 70 database grid computers, and the block size was set to 1150 records. The number of directors was set to 10, 15, and 20. The throughput and delay for the original and proposed application are shown in Table 3.

With 5% of the records being sent to two database grid computers the throughput decreases by less than 3.5% and the delay increases by no more than 5%. Therefore, the proposed change to the application does not appear to significantly decrease performance.

3.5. Batch and interactive classes segregated

The scenario discussed in this section is the best model for the present production application. There are 12 batch clients and 8 interactive clients. The batch clients are being serviced by 12 directors and the interactive clients are being serviced by 4 directors for a total of 16 directors. Therefore, the batch and interactive records are segregated at the director level. But batch and interactive records still share the resources of the database grid. The block size for the batch class is set to 1150 records and for the interactive class is set to 1 record. There are 70 grid computers in

Table 4 Throughput and delay of a system with batch and interactive clients with segregated directors

	Throughput (Mrecords/hr)
Interactive	3.4
Batch	47.5
Total	50.9
	Avg. Delay (s/record)
Interactive	0.0002
Batch	0.0314
Total	0.0316

the database grid. The throughput and delay are shown in Table 4.

In this system, 4 directors are devoted to the 8 clients running interactive. Therefore, the interactive records are segregated from the batch records at the director level. This reduces the mean delay per record to better serve the interactive clients. The decreased response time lowers the overall throughput of the system from 79.5 Mrecords/hr to 50.9 Mrecords/hr.

4. Conclusions

An analytical model of a distributed database application was developed and several realistic assumptions were made to test it to provide examples of the type of analysis that can be performed. The block size, the distribution of the submitted records, the number of computers in the director grid, and the number of clients were varied in the model. Also, a proposed change to the database application was modeled. The proposed change did not significantly degrade the performance of the system. Finally, the production database application method of segregating the batch and interactive records at the director level was modeled. The segregation of the two types of records decreased the mean delay of a record at the expense of decreased throughput.

First, the number of directors should be approximately equal to the number of clients to obtain the maximum throughput of the system. This is logical because the records from the clients are routed through the director grid. The bottleneck device is not the director CPUs, but the network between the clients and the director computers and the network between the director computers and the database grid. Still, the closer the match between the number of clients and the number of director computers increases the throughput.

Second, the bottleneck device in this system is the network. A higher capacity network would increase the performance of the network. A faster network would not remove the bottleneck, but would just move it to another device. For example, the bus of the computer may become the bottleneck device if it is slower than the network.

Also, the proposed application change that caused 5% of the records to require service from two database grid computers did not significantly decrease the performance of the system. This is an example of the type of analysis that this analytical model can provide without having to implement and test the system. The strength of this model is that it can be used as a planning tool.

Segregating the batch and interactive records at the director level causes the response time of the interactive requests to decrease. The decreased response time comes at the price of lowering the overall throughput of the system. As discussed, the model can be used to determine the trade offs of decreased response time versus increased throughput.

5. References

- [1] D. A. Menasce, V. A. F. Almeida, and L. W. Dowdy, *Capacity planning and performance modeling: from mainframes to client-server systems*, New Jersey: Prentice Hall, 1995.
- [2] J. P. Buzen, "Computational algorithms for closed queueing networks with exponential servers," *Communications of ACM*, vol. 16, no. 8, Sep. 1973, pp. 527-531.
- [3] M. Reiser and S. S. Lavenberg, "Mean-value analysis of closed multichain queueing networks," *Journal of the ACM*, vol. 27, no. 2, Apr. 1980, pp. 313-322.
- [4] H. Wang and K. C. Sevcik, "Experiments with improved approximate mean value analysis algorithms," *Performance Evaluation*, vol. 39, 2000, pp. 189-206.
- [5] Y. Bard, "Some extensions to multiclass queueing network analysis," in M. Arato, A. Butrimenko, E. Gelenbe (Eds.), *Performance of Computer Systems*, North-Holland, Amsterdam, 1979, pp. 51-62.
- [6] P. J. Schweitzer, "Approximate analysis of multiclass closed networks of queues," in *Proceedings of the International Conference on Stochastic Control and Optimization*, Amsterdam, Netherlands, 1979, pp. 25-29.
- [7] M. Reiser, "A queueing network analysis of computer communication networks with window flow control," *IEEE transactions on Communications*, vol. COM-27, no. 8, Aug. 1979, pp. 1199-1209.

D. R. Thompson and S. Shin, "Mean value analysis of a database grid application," in *Proceedings of the 3rd International Conference on Networking*, Guadeloupe, French Caribbean, March 1-4, 2004.