

PERFORMANCE MODELING STUDY OF A CLIENT/SERVER SYSTEM ARCHITECTURE

Ji Shen

AMS Center For Advanced Technologies
American Management Systems, Inc.
4050 Legato Road
Fairfax, Virginia 22033, U.S.A.

Shahla Butler

AMS Center For Advanced Technologies
American Management Systems, Inc.
4050 Legato Road
Fairfax, Virginia 22033, U.S.A.

ABSTRACT

This paper presents a model developed during the course of an investigation conducted at The AMS Center for Advanced Technologies, in which IBM's REsearch Queuing Modeling Environment (RESQME) was used to perform discrete-event simulation of a large client/server (C/S) system for a major corporation in the health-care products field. RESQME provides a wide range of tools for applying discrete-event simulation to application architecture, system sizing, and the configuration of computer systems. By applying simulation modeling early in the system design process, system developers can avoid the performance bottlenecks and potentially costly mistakes in system design. The RESQME analysis identified an initially unsuspected system bottleneck and permitted stress-testing of the proposed system without physical implementation of hardware and software.

1 INTRODUCTION

As Client/Server (C/S) technology gains momentum, performance has been recognized by more and more information technology professionals as one of the most important issues in C/S system design. The distribution of system functions and data in C/S systems, along with the large number of hardware and software components, make it a complex task to determine the appropriate application architecture, system sizing, and configuration of such systems. Simple linear spreadsheet modeling and

rule-of-thumb estimation techniques pose unacceptable risks in the C/S development scenario.

This paper gives an account of a project in which a development team at American Management Systems Center for Advanced Technologies (AMSCAT) used IBM's REsearch Queuing Modeling Environment (RESQME) as the primary performance modeling tool to aid in the design of a large C/S system for a major client in the health-care field. RESQME employs queuing theory and discrete-event simulation technology to represent planned system configuration and application design and produce outcome data for performance measures; e.g., response time, throughput and resource utilization levels. Properly used, this tool can assist system architects in designing better application architecture, identifying potential bottlenecks, and sizing equipment (Gordon et al., "An Extensive Visual Environment" 1991). Modeling and simulation allow us to avoid making oversimplified assumptions about complex systems and help identify unexpected problem areas. Sensitivity analysis of the results enables us to stress-test systems without physical implementation of hardware and software. The projected performance can be used as a criterion for sizing decisions.

2 INITIAL INVESTIGATION

The system that was modeled in this study is a client/server (C/S)-based system that tracks information on clinical test results for a healthcare products corporation. The system must be highly responsive and

available while supporting a large number of concurrent user transactions.

The system design includes an online database server, an automated call distribution (ACD) unit, an Ethernet LAN and a large number of client workstations. Users access the system by calling in on touch-tone or rotary telephones. The ACD unit greets the users and prompts them to enter their personal identification information. This information is then transferred from the ACD unit to the database server through full-duplex serial communication lines. After the users listen to their testing results as returned by the database server, they may listen to prerecorded messages or choose to speak with a service representatives who is seated at a workstation. The representatives are connected to the database server via the Ethernet LAN. Data about callers is retrieved from the database server and presented on the workstation screens just before users' calls are forwarded to the service representatives. Figure 2.1. illustrates the system's configuration.

The application architecture for this system includes a set of complex process flows among the three major components: i.e., the database server, the ACD unit, and the PCs. Messages pass to and from each component via the serial lines and/or the Ethernet LANs.

In attempting, before actual implementation, to identify potential problems in architecture and system configuration the development team asked the following questions:

- Are the Ethernet and serial lines potential bottlenecks?
- Is it appropriate to choose HP 9000/H60 as the database server platform? What kind of performance can be expected from this server in our planned configuration?
- How will the system perform as transaction volume increases?

It was obvious from the beginning that the performance of each component would depend to some extent on that of the other components. Interaction among the components precluded the use of static or merely linear modeling techniques because these would fail to estimate the dynamic and nonlinear characteristics of the system. Benchmarking would have provided credible results; however, the purchase of equipment was predicated on estimating the capacity needed, and it was

therefore not feasible to conduct any effective benchmark tests. Discrete-event simulation remained as the natural choice for this task.

3 DISCRETE EVENT SIMULATION AND RESQME

Discrete-event simulation models the evolution of a system over time by both graphically and mathematically representing instantaneous changes in state variables at separate points in time. When a discrete-event model of a computer system has itself been implemented on a computer, analysts can collect information from the model as if it were a "real" system. Thus a proposed configuration's future or potential characteristics can be evaluated (Law 1991).

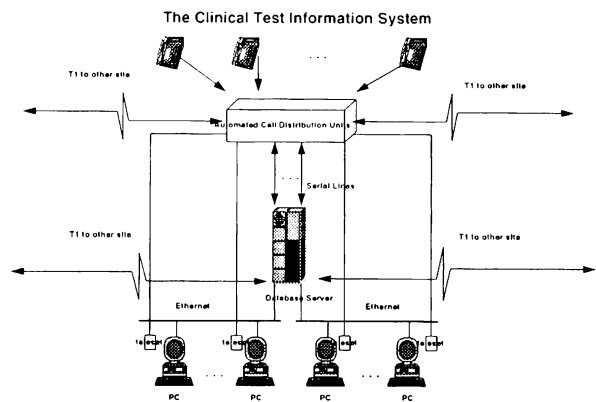


Figure 1. Configuration of the Modeled Information System

In order to build a RESQME model, three categories of information must be provided:

- User behavior or "think time" specification: This information tells the model how often user requests arrive at the system. We chose a probability distribution that would allow us to specify this information by assigning interarrival times. In this model, this information is determined from incoming call volumes.
- Work Demand specification: Work Demand specifies the amount of processing that is required. In this model, the Work Demand equals the number of CPU instructions required.
- Processing environment characterization: This information is also required at every model component.

It specifies the service rate for the queue and equals the number of instructions the CPU can execute per second.

Other types of information required are process flow logic and branching probabilities.

4 QUANTITATIVE ASSUMPTIONS

Before the model can be built, the required model parameter values have to be determined. This section of the document will state what these parameters were in our study and how they were estimated.

4.1 Incoming Call Volumes

The incoming call volumes were based upon the following assumptions:

- Total sales volume of the clinical test kits is 7 million per year with the potential of growing to 14 or 20 million per year.
- The actual call volume data from two similar systems was used to estimate the average number of incoming calls at peak load at around 8,130 calls per hour.
- Standard deviation of incoming call rate was assumed to be 10% of the average number of incoming calls per hour.
- The above estimates of peak load were based on a spreadsheet model. No actual data was available to verify these estimates. On the basis of the project team's knowledge about call distributions, we selected a standard distribution to characterize the process of random arrival of calls.

4.2 Aspect Call Distribution Unit

The ASPECT Call Distribution unit has four kinds of ports: 1,200 agent ports, 1,200 trunk ports at maximum, 96 voice ports at maximum (each of which can serve up to 20 callers simultaneously), and 3 other types of voice-processing cards. At any given time, at maximum, 1,888 ports can be available without blocking. However, when an incoming call is forwarded to a customer service representative's workstation, that particular call will tie

up 2 ports, which means that in practice fewer than 1,888 ports will be available. A gross estimate of the number of calls that will be transferred to a workstation is around 10% of the total incoming calls.

Our model did not attempt to capture the details of the ASPECT's inner workings. Instead, our emphasis was on finding out the maximum number of calls the ASPECT could process without turning away any call. We based this estimate on the time each call could spend in different components of the system (e.g., serial lines, HP, Ethernet and the workstations). A very detailed ASPECT model would also significantly stress the capacity of the PC on which the simulation model was running.

The model assumed that each ASPECT is capable of taking 1,888 incoming calls simultaneously without blocking. This assumption does not take into account the fact that some calls will tie up two ports instead of one. The estimated capacity of the system based on this assumption will therefore be slightly larger than it would be if based on a less optimistic assumption. A simulation run was made to predict the effect of having 2 ASPECT units with a total of 3,776 ports at maximum.

4.3 Serial Lines

There are totally 5 serial lines connecting each ASPECT call distribution unit and the HP workstation. When multiple ASPECT units are present in the system, the total number of serial lines equals the number of ASPECT units multiplied by 5. All serial lines are full-duplex with 9600 bits per second bandwidth in each direction.

We assumed that 80% of the calls in each direction will go through Serial Line #1, and that the remaining 20% will be evenly distributed across the remaining 4 lines.

4.4 HP and Oracle DBMS

As we have indicated, we were not able to do any benchmarking specific to the application. Industry standard benchmark data was applied to the HP model. Had it been feasible, benchmarking tests would have

been desirable to collect more accurate input data and validate preliminary modeling results.

The first-stage objective in the modeling study was to get estimates of the transaction volume on the SQL server (Oracle DBMS) on HP. An unconstrained HP model that could process transactions with no delay was built for this purpose. Throughputs for different types of transactions specific to this application were estimated by this model.

The unconstrained throughput, along with the transaction profile information, was given to the vendor for recommendation of an appropriate server platform. The vendor recommended a multi-processor HP 9000/H60.

The HP 9000/H60 has been estimated to be capable of processing 14.5 TPC-C transactions per second (the H60 has 40% to 50% more processing power than the H50, which has a throughput of 613.8 TPC-C transactions per minute.) TPC-C is a benchmark designed by the Transaction Processing Council for C/S systems. This application's transactions are simpler than the TPC-C order entry transactions, the worst-case estimate being that each of this application's transactions is equivalent to 1 TPC-C transaction. This mapping between the two was based on the particular application architecture design. Should the architecture change, the mapping would also have to be re-estimated. These assumptions served as the input to the model for estimating key system activity response times.

4.5 Ethernet

The Ethernet was modeled on the Media Access Control (MAC) layer with 10 megabits per second bandwidth, using the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. The length of physical wire was assumed to be exactly 1 kilometers (Sauer 1983).

5 THE PERFORMANCE MODEL

A model was built, using RESQME, to simulate the system's performance. The model represented the workload, the system hardware and software environment, and the application architecture. RESQME

allows us to build a layered hierarchy of models and submodels. We designed the model with multiple layers and multiple functional components in each layer. In the following sections of this document, we will explain each layer (i.e., each submodel) in more detail.

5.1 Top-Layer Model

The top-layer model mainly includes submodel components that are almost identical to the functional blocks in the real system. We had no intention of keeping this one-to-one mapping relationship inside the submodels. To capture all such details of a real-life system in a simulation would almost certainly require more computing resources than are available on a regular PC or a workstation. The top-layer model is presented in Figure 5.1. Among the five submodels on this layer, the Ethernet submodel was a direct re-implementation of one previously built (Sauer 1983), the HP database server submodel is a slight modification of a previous model that we built in an earlier project, and the rest are newly developed. The newly developed and modified submodels will be stored in our submodel repository for future reuse.

The node \square is defined as a source node. It represents the process of arrival of requests, referred to as jobs, at the system. A probability distribution function can be employed to characterize the arrival process by specifying the interarrival time. In this particular model, a distribution supplied by RESQME, called Standard Distribution, was chosen. The average interarrival time was estimated to be 0.44 seconds (= seconds in an hour/average number of arrivals per hours = 3600/8129).

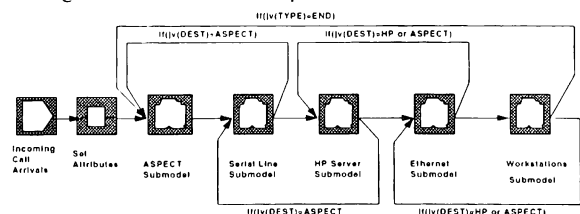







Figure 2. The Top Layer of the Performance Simulation Model

The node  assigns attribute values for each arrived job. RESQME provides a mechanism called job variables (abbreviated as JV) to associate attribute values with jobs. In our model, this node specifies the percentage breakdown of different types of incoming calls. The breakdown will determine the process flow and system resource required to process each job.

Five submodels are represented by the icon . They are the ASPECT Call Control Unit submodel, the Serial Line submodel, the HP9000 Database Server submodel, the Ethernet LAN submodel, and the Workstation submodel. As their names suggest, each of these simulates a real system component. Of course these submodels do not implement any of the real system's functionality; rather, they estimate the time required for jobs to be processed in each component. Details of these submodels will be explained below.




All the possible process flows in this top layer have been graphically illustrated in Figure 5.1. Each job may follow a different route depending upon such attributes as destination, source, and type. JVs are assigned at the assignment node  and inside some of the submodels before the job is transferred.



For syntactic reasons, every submodel in the following sections includes two sets of nodes,  and , so that the submodels can be properly connected with the upper-layer model.

5.2 Aspect Call Control Unit Model

The ASPECT Call Control unit is the interface between customers and the system. Its main function is to interact with customers, process the information they enter, and pass this information on to the HP over the serial communication lines. The unit also coordinates system operations when a customer requests to speak with a representative. Figure 5.2 is the RESQME submodel built for the ASPECT unit. The submodel has been slightly modified to ensure both confidentiality and simplicity. The detailed inner workings of the ASPECT unit are not simulated. The emphasis is on the application process flows.

The key component of this submodel is the pool of available ports represented by a RESQME passive queue

. As stated earlier, a total of 1,888 ports per unit are available at any given time without blocking. These ports are represented as a number of tokens in the pool. When a customer calls in, one port is allocated to him at the allocate queue . The port is not released at the release node  until the customer hangs up.

The customer listens to a greeting and to option menus, and enters his/her personal PIN number after being prompted. These activities are simulated by an infinite server queue  because there is no waiting for other customers in this process. The work demand at this queue was estimated by taking actual measurement from the unit. The information entered is transferred to the HP so that customer data can be retrieved. The information concerning message size and destination is assigned at the assignment node .

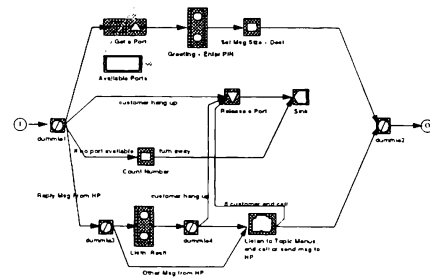


Figure 3. The ASPECT Call Control Unit Submodel

For various reasons, the time that a customer spends in the system ranges from less than 1 minute to 30 minutes. There is a significant possibility that late-arriving customers may find all ports held by others. These customers are turned away. The number of customers turned away is counted at an assignment node labeled Counter Number.

A number of other activities also occur in the ASPECT. Most of these are simulated in another submodel nested inside the ASPECT submodel. This is important for the accuracy of project deliverables but not necessary for illustration purposes. We therefore pass over these details in this paper.

5.3 Serial Line Model

Each ASPECT unit can be connected to the HP by 5 serial communication lines. The serial line submodel is presented in Figure 5.3. The model following, for the sake of simplicity, includes only one full-duplex serial line. In practice all 5 lines were modeled.

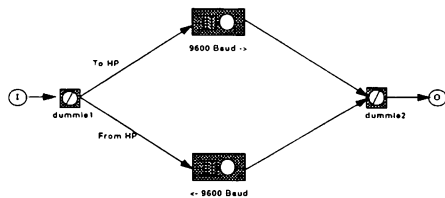


Figure 4. The submodel for a full duplex serial communication line

The serial line has a bandwidth of 9600 baud (9600 bits per second) in each direction. Both directions are simulated by First Come First Served (FCFS) single-server queues. Each queue has the following specific characteristics:

- Service Rate: constant(960) bytes per second. This is simply the serial line's transfer speed. The unit given is bytes instead of bits because all message sizes in the model are specified in bytes.
- Work Demand: constant(jv(msg_size)). This is the amount of traffic generated on the serial line by each job. Jv(msg_size) is a job variable specifying message size.

5.4 Database Server Model

This section describes the database server model. The server platform consists of HP 9000 workstations running Oracle 7. A model was built to predict the performance of this configuration under various workloads and further determine which HP 9000 platform, namely H50, H60, or H70, would be the most appropriate for this application. The model is represented in Figure 5.4.

The model simulates the database server on a fairly high level. Processes at the level of the operating system (OS), for example CPU and memory access and I/O

activities, are not directly reflected in this model. Numerous factors on the OS level can affect the server's overall performance. To simulate such system activities with sufficient detail would require very intimate and extensive knowledge of the specific OS and Database Management System (DBMS) products. It will also require tremendous computing resources to complete the simulation. We decided to simulate only the overall transaction processing performance, omitting excessive details. Industry-standard benchmarks were employed to specify the server's processing characteristics and the matching work demands. (See section 3.4 for details.)

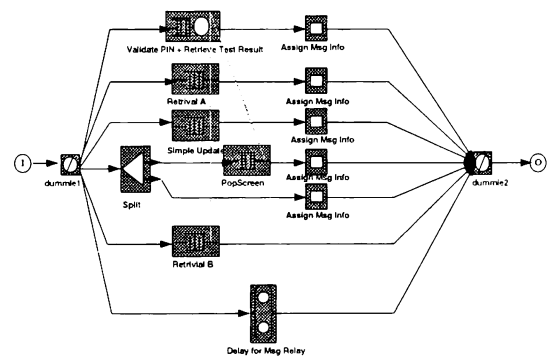



Figure 5. The HP9000 Database Server Submodel

The active queue represents the database server and the waiting line for jobs requesting PIN number validation and information retrievals. Other queues represent waiting lines for 4 other types of transactions. The shaded lines connecting them to the active queue indicate that all of the transactions will be processed by the database server. The parameter values are specified as follows:

- Service Rate:
 - 10.2 TPC-C transactions per second for HP9000/H50
 - 14.5 TPC-C transactions per second for HP9000/H60
 - 20.2 TPC-C transactions per second for HP9000/H70
- Work Demand: exponential(1).

Each application-specific transaction is considered equivalent to 1 TPC-C transaction. This work demand parameter value applies to jobs at all 5 waiting lines.

- All waiting lines have FCFS service discipline.

When the HP database server receives a message from the ASPECT unit requesting transfer of calls to a representative, the server extracts data to populate the representative's PC screen and sends a confirming message back to the ASPECT. This process is simulated with a split node, , which spawns a child process whenever the parent job arrives. Message sizes and destination information are again specified by assignment nodes.

5.5 Ethernet LAN Model

A critical task for C/S application designers is to estimate the impact of network processes on overall system performance. We have implemented a RESQME submodel devised by Charles H. Sauer and Edward A. MacNair for the Ethernet LANs in this system (Sauer 1983). A graphical representation of this submodel may be found in Figure 5.5.

Most of the application's messages have fairly small sizes. The majority are smaller than 150 bytes. Only a few are greater than 1 KB in size. We decided not to simulate the packetizing process on each PC's network adapter card.

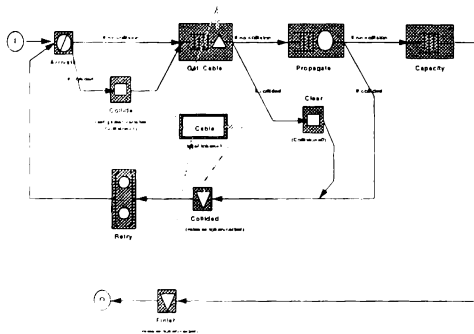




Figure 6. The Ethernet LAN Submodel with CSMA/CD Protocol

It is beyond the scope of this paper to explain the CSMA/CD protocol. However, it is important to take into consideration the fact that excessive packet collisions on the network may increase transmission time exponentially because of the way in which CSMA/CD works. The submodel we use does estimate the potential impact of excessive collision on network performance.

We believe the graphical representation in Figure 5.5 to be self-explanatory. For more detail on CSMA/CD protocol, please refer to network-related literature.

5.6 Client Workstation Model

The client workstations are responsible for receiving information from the database server and populating the representatives' screens. They also take user input via a Graphical User Interface (GUI) and submit the requested transactions to the database server. The model simulates the effect of having 300 workstations, including 133 workstations for service representatives and 167 for counselors. The client workstation submodel is presented in Figure 5.6.

The application is designed in such a way that no representative can speak with more than one customer at a time. That means that customers who come in later will be put on hold until a representative is available. Figure 5.6 has illustrated how this potential waiting process has been modeled using two wait nodes represented by the icon . The two submodels  nested in this submodel contain the simulation of detailed client processes. We used active queues with multiple servers and FCFS waiting lines. The detailed content of the two nested submodels is skipped.

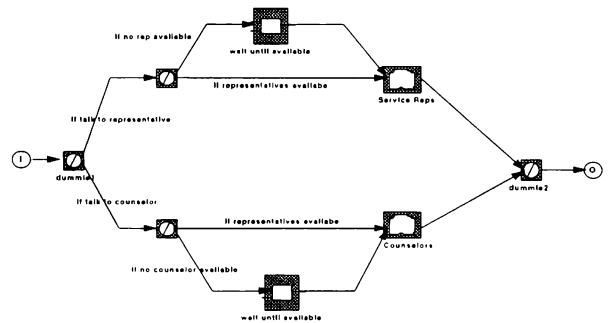


Figure 7. The Client Workstation Submodel

6 CONCLUSIONS AND RECOMMENDATIONS

We made numerous simulation runs, and compiled and analyzed the outcome statistics. Our major conclusions are as follows:

- Neither the serial lines nor the Ethernet LANs are performance bottlenecks in the system. The level of

utilization of the serial lines was estimated to be under 40% for any sales volume of fewer than 14 million test kits. The level of utilization of the Ethernet LANs was estimated to be less than 5%.

- The HP9000/H60 workstation appears to have sufficient power to support the current transaction workload; The estimate of performance for different HP workstation models supporting a wide range of workloads is presented in Figure 5.1.
- The ASPECT Call Control unit was the key limiting factor in system performance. With one ASPECT unit, approximately 15% of customer calls would be turned away based on the 7 million test kit sales volume, and the percentage of calls turned away would increase to 53% as sales volume doubled. With two ASPECT units, the corresponding percentages are 0% and around 18%. In order to assure high availability, the system needs at least two ASPECT units.

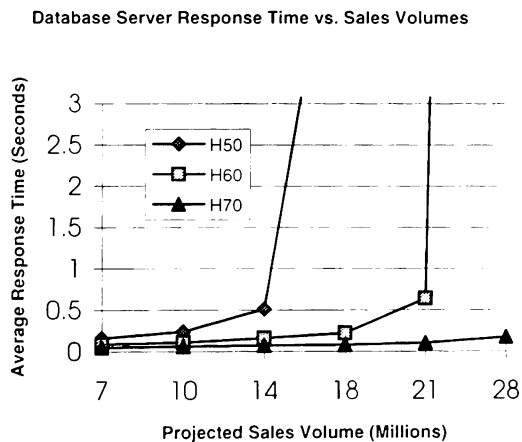


Figure 8. Projection of Response Time for Different HP9000 Models

ACKNOWLEDGMENT

We would like to acknowledge Pete Glosser for editing this paper.

REFERENCES

Anelli, Pascal and M. Soto. 1993. *A General-Purpose Network Simulation Tool*. In Proceedings of the 1993

Summer Computer Simulation Conference, ed. Joel Schoen, 584–589. Boston: The Society for Computer Simulation.

Gordon, K.J., J.F. Kurose, R.F. Gordon, and E.A. MacNair. 1991. *An Extensive Visual Environment for Construction and Analysis of Hierarchically-Structured Models of Resource Contention Systems*. In *Management Science* Vol. 37, No. 6 (January): 714–732.

Gordon, R.F., P.G. Lowner, and E.A. MacNair. 1991. *The Research Queueing Package Version 3 Language Reference Manual*. Yorktown Heights, N.Y.: IBM Thomas J. Watson Research Center.

Law, Averill M and W.D. Kelton. 1991. *Simulation Modeling and Analysis*. New York: McGraw-Hill.

Sauer, Charles H. and E.A. MacNair. 1983. *Simulation of Computer Communication Systems*. Englewood Cliffs, N.J.: Prentice-Hall.

AUTHOR BIOGRAPHIES

JI SHEN is a senior analyst at the AMS Center for Advanced Technologies. He received a B.S. in Mechanical Engineering from Harbin Institute of Technologies in China in 1988, and he received a M.S. degree in Operations Research and Applied Statistics from George Mason University in 1991. Mr. Shen is completing his Ph.D. dissertation at George Mason University in Information Technology. His research interests include simulation, computer performance analysis, computational statistics, 3D visualization, and virtual reality.

SHAHLA BUTLER is Associate Director of the AMS Center for Advanced Technologies and the Director of its Performance and Measurement Lab. Her current research interests include simulation, computer system architecture, software reuse practices and organizational change management. Dr. Butler received her B.S. in Chemistry from University of Michigan in 1968, and her Ph.D. in Quantum and Statistical Mechanics from University of Chicago in 1974.