PERFORMANCE PREDICTION OF A VIRTUAL MACHINE

Dr. M. K. Rajaraman

University of New Mexico
Albuquerque New Mexico 87131

ABSTRACT

Modeling and simulation of computer systems have two main objectives. First, to evaluate the performance of a given configuration of a machine and second, to derive a mechanism for prediction of performance when configuration parameters change. This paper addresses the second issue and reports the result of a recent investigation of a Virtual Memory Computer. The results indicate which variables or combination of variables have significant effect on the performance and which do not.

INTRODUCTION

In recent years performance evaluation of computer systems has played a significant role in the field of computer science. This is mainly a result of the complexity of current systems and the speed with which computer technology is changing. Even before a system is put together and becomes operational new technology could change the entire concept with which the original system was designed. So, designers are using performance evaluation to study the effects of alternate configurations of the systems. On the other hand, the needs of the user are also changing fast, perhaps because of new technology and also because of better system capabilities available at a lower cost. This makes it essential that the system planner determine as to which variable or combination of variables affect the performance significantly and which do not have significant effect.

For example, a current system may be capable of being upgraded in the near future by faster (and cheaper) memory, faster peripherals, better operating system and so on. In order to determine which of these would significantly improve performance of an existing configuration one could use a simulation model of a computer system. This is the second aspect of performance evaluation, namely, performance prediction.

In this paper we report the results of a recent study made by using a simulation model of a virtual memory system. The results of the study indicate the sensitivity of the system configuration and the nature of impact that different variables have on the performance of the system.

BACKGROUND

The model that is studied is that of a Virtual Memory Computer System. A number of studies have reported various individual subsystem models and their performance. In this study we brought together the known subsystem models and integrated them into a homogeneous system model of a Virtual Memory Computer. It is one of the comprehensive models where a number of different aspects of performance evaluation and prediction have been studied.

In modeling the system three major subsystem behaviors have to be considered and their interrelationships modeled. They are: Program Behavior, Memory Allocation under Multiprogramming and Secondary Storage configuration. The subsystems are then linked together and their interactions also modeled. These are the major subsystems that have been studied by others and modeled extensively as individual systems. Our model is based on these well-known results.

The program behavior is critical in the study of the Virtual Memory machine. The concept of virtual memory is derived from the basic behavior of programs in a computer. By program behavior is meant the nature of address references that a program makes during its executions. It is this behavior that determines which part of address space is in main memory and which is in secondary. The principle of locality characterizes the behavior of programs. Denning et.al. [5] and Shedler and Tung [10] studied this aspect extensively. Coffman and Varian [3] have repotted experimental data on program behavior. Results of Coffman and Ryan [2] also corroborate earlier experimental results. Our model is based on these results on page reference patterns.

Under a multiprogramming, paging environment memory allocation and management become very critical. The concept of working set characterizes the optimal allocation. The size of memory allocated to a program that will minimize the occurrence of page faults is the working set.

This is quite difficult to determine under dynamic conditions. However, one could experiment with different types of memory allocation and choose the scheme that gives the least rate of page fault.

Denning [4] has given some theoretical characterization of working set. Rodriguez-Roselle [8] report some experimental results on working set size. However, these are either too specific or too simplistic to be useful in a realistic modeling of a virtual machine.

Queueing models are most commonly used for the paging device (secondary storage). Models exist for Disks and Drums. Markov chain analysis is another approach for modeling the secondary storage. Seaman et.al. [9] studied a simple exponential queueing model. Abate et.al. [1] studied tandem queueing for disk storage. Teorey and Pinkerton [12] report secondary storage behavior under different scheduling algorithms. In our study we assume a single channel queueing and test the effect of scheduling rules.

Though a number of studies exist that model individual subsystems, few studies are reported where they are integrated to represent a virtual memory computer. Two studies are of importance in this respect. Kuck and Laurie [7] used models of paging memory allocation and secondary storage. Smith [11] also modeled these systems and included such variables as job mix. But he considered only a limited number of variables. The study reported here used some of the submodels and parameters reported in the above studies.

THE MODEL

The model studied consists of three subsystems. We use a combination of probabilistic and task sequence descriptions for the components of the model. In describing the external environment of the model we consider the incoming job mix.

A stream of jobs arrive for service by the machine on a first come, first served basis. In order to eliminate the effect of arrival patterns on the dynamic performance we consider a saturated queue for the incoming jobs. The jobs are classified as either 'Batch' or 'Interactive' jobs and the job mix is described by the percentage of interactive jobs in the queue. The model maintains the average job mix at a given level by generating new jobs to join the input queue intermittantly.

Once the jobs are in the queue they contend for memory space. When a currently executing job finishes its quantum of CPU time (called execution internal) another job from input is admitted for CPU service. Initially all jobs are assigned a fixed number of pages in memory (usually three pages). Thereafter, the job accumulates additional pages as execution proceeds. There is a maximum limit to the number of pages a job can occupy. After that,pages have to be swapped. During a page replacement,the job is 'inactive' and does not receive CPU service.

Explicit I/O is not considered, since a job requesting explicit I/O will be 'blocked' and does not receive its CPU quantum until it is able to obtain memory space. Such jobs are recycled back to input queue. The page request rate of the jobs is simulated by the program-behavior submodel. When a job requests a new page the request is queued to a single channel secondary I/O device where it is served according to a scheduling rule. The simulation starts with a saturated queue with periodic replacement for jobs that finish processing. The simulation is continued for a given length of simulated time and this process is repeated five times for each combination of parameters.

The program behavior submodel generates page faults for the program based on the relationship.

$$t(p) = a \cdot p^b$$

where $t(p)$: mean time to reference p pages

a and b: are constants

The values of a and b we used were 1.1 and 3.4 as reported in the study by Kuck and Lurie [7] mainly using data by Fine et.al. [6].

The total CPU time that a job requests depends on the type of job; i.e., whether it is a batch or interactive job. We used a hyper-exponential distribution which is of the type

$$p(t = \tau) = w\, e^{-\tau/\alpha} + (1-w)e^{-\tau/\beta}$$

where the left hand side represents the probability that a job requests τ units of CPU time. w is the weighing factor for the job mix, α and β are parameters for the two classes of job. Smith [11] estimates the values of α and β to be 88 and 40,7000 instructions, respectively, for typical programs.

The multiprogramming level was varied between 2 and 6 in the study and data collected for each level. In memory allocation we considered the static and dynamic memory allocation schemes. In the static scheme each program was allocated a fixed partition within which it can accumulate its pages. The partition size was varied and data collected for each. In the dynamic scheme each program accumulates pages based on its page fault rate and the process proceeds until the entire memory is filled. In both cases data was collected for different memory sizes.

The model for the secondary storage included a general model, a Disk Model and a Drum Model. In the general model we considered only the average page transfer rate. Page transfer rates varying from 2.5 m secs. to 15 m secs. were considered. This range includes most commercially available paging device rates (except extended core which was not simulated). The disk model contained specifications for cylinders, tracks, rotational speed, transmission rate, page density on the tracks and rate of read-arm movement over the cylinder. We assumed that for each page request the cylinder number, track number and page number are uniformly distributed random variables.

The drum system model had specifications for number of tracks, pages per track, rotational speed and transmission rates. Only fixed head drums with single head per track was considered. The page request was served based on two rules: first come, first served and nearest page next.

Two measures of performance were considered: The CPU-Utilization and Job Throughput (the number of jobs to leave CPU per unit time). Six different independent variables were used. They are:

1. Job Mix
2. Multiprogramming Level
3. Page Replacement Speed
4. Type of Memory Management (Static and Dynamic)
5. Page Service Rule
6. Type of Paging Device

METHODOLOGY FOR PREDICTION

One of the main aims of the study was to determine the effect of the different variables and their interactions on the performance indexes. This involves two aspects: first, to determine which variables and interactions are significant and second, to express the performance indexes as functions of these. The first involves a statistical determination of variables that have significant influence on the performance indexes. The second is determining a prediction equation as a function of the variables. This equation can be used to obtain the values of the performance indexes for different values of the variables within the ranges used in the study. One could obtain different sets of equations for different configurations.

In this study we used the method of step-wise Multiple Regression Analysis for fitting a multidimensional surface for the performance index to be predicted. The surface of non-linear equations expresses the relationship between the performance index and the set of variables. The equation can be considered as a relationship between the dependent variable and a set of independent variables or as an inferential equation which expresses the relative influence of various variables on the performance index. The latter approach implies that an inference about the relationship is deduced based on sample data obtained in the simulation. In the former we simply use the equation to describe the observed relationship.

A major assumption made is that the effect of the significant variables and interactions on the performance index is additive. In the Multiple Regression Analysis we find the least square fit for the performance parameter in the form of an nth degree polynomial of the variables. The step-wise procedure selects the variables or their interactions in the order of their significance and includes them in the final equation. If y' represents the performance index and $X_1 X_2 \ldots X_n$ represent the variables and their interaction, then prediction equation is of the form

$$\begin{aligned} y' = {} & a_0 + a_1 X_1 + a_2 X_1^2 \ldots\ldots\ldots + a_n X_1^n \\ & + b_1 X_2 + b_2 X_2^2 \ldots\ldots\ldots + b_n X_2^n \\ & + r_1 X_m + r_2 X_m^2 + \ldots\ldots + r_n X_m^n \\ = {} & a_0 + \sum_{i=1}^{n} a_i X_1^i + \sum_{i=1}^{n} b_i X_2^i + \ldots + \sum_{i=1}^{n} r_i X_m^2 \end{aligned}$$

where $a_0, a_1, a_2 \ldots a_n, b_1, b_2 \ldots b_n \ldots r_1 \ldots r_n$ are constants to be determined. If y is the observed value of the index, then the least square polynomial regression finds the constants in such a way that

$$\Sigma (y-y')^2$$

is minimized, where the summation is over the values of y and y'.

If $\bar{y}$ is the mean value of the observed performance index, then the ration

$$\sqrt{\frac{\Sigma (y'-\bar{y})^2}{\Sigma (y-\bar{y})^2}}$$

is called the coefficient of correlation and is a measure of the strength of the relationship between y' and X_i's. The accuracy of the polynomial equation in predicting the performance index within the given ranges of the variables is measured by the parameter called the Standard Error of the Estimates, given by

$$\sqrt{\frac{\Sigma (y-y')^2}{N}}$$

where N is the sample size. The accuracy of the contribution made by each variable to the performance index can also be measured. If b_i is the coefficient of the variable, X_j^i is the prediction equation, then the ratio

$$\frac{\Sigma (y-y')^2}{(N-2)\,(\Sigma (X_j^i - \bar{X}_j)^2}$$

is called the standard error of b_i (where $\bar{X}_j^i$ is the mean of X_j^i values).

The Multiple Regression Analysis was done using SPSS (Statistical Package for Social Science) package on CDC 6600 computer. As a part of the results we get

(i) the coefficients of the prediction equation
(ii) coefficient of correlation
(iii) standard error of the estimates
(iv) standard error of the coefficients.

In addition, we also get the results of an F-test that shows if there is any significant lack of fit of proposed prediction equation to the population from which the data was sampled.

We developed prediction equations for two performance indexes: CPU-Utilization (percentage of time that CPU was idle) and Throughput (the number of jobs leaving CPU per CPU second). Six different configurations were studied. They were termed DISK-1, DISK-2, DISK-3, DRUM-1, DRUM-2, and DRUM-3. The configurations with the name 'DISK' used Disk as the secondary device and those named 'DRUM' used Drum as the device. The suffix 1, 2 and 3 represent different software alternatives. They are:

DISK-1, DRUM-1:	Page Service - FCFS and Static Memory Management
DISK-2, DRUM-2:	Page Service - Nearest Page Next and Static Memory Management
DISK-3, DRUM-3:	Page Service - Nearest Page Next and Dynamic Memory Management.

These combinations were selected because they had the most influence on the performance indexes.

For each configuration the effects of the following variables were studied.

1. Job Mix (Symbol-X: Range: 0.1 to 0.7)
2. Multiprogramming Level (Symbol-M; Range: 2 to 8)
3. Memory available. For DISK-1, DISK-2, DRUM-1, DRUM-2 this is expressed as the maximum number of pages (1K-Page size). (Symbol: P; Range 3 thru 33 in increments of 3). For DISK-3 and DRUM-3 this is expressed as maximum core available in number of words (Symbol: C; Range: 50K to 300 K in increments of 50K).
4. Paging Speed (Symbol: S) For Disks this is expressed as cylinder to cylinder seek time. (Range: 24 m sec to 30 m sec in increments of 1 m sec). For Drums this is expressed as average page transfer time. (Range: 700 m sec to 2200 m sec in increments of 300 m sec).

RESULTS

The polynomial prediction equation had

(i) Constant Terms
(ii) Linear Terms in each variable
(iii) Product Terms for variables taken two at a time
(iv) Terms containing the square of each variable
(v) Product Terms for variables taken three at a time
(vi) Terms containing the cubes of each variable.

Experiments showed that with these types of terms the prediction equation was able to predict performance indexes fairly accurately. Table-1 shows the equations for the CPU-Utilization. Table-2 shows the equation for Throughput. The following observation can be made based on these results.

For Disks, the memory allocated (P or C) and Multiprogramming level (M) have more significant influence on CPU-Utilization than the job mix (X). For Drums, paging speed (S) and memory allocated (P or C) have more significant effect on CPU-Utilization than the other variables.

For Disks, the job mix and multiprogramming level have more effect on the throughput than other variables. For Drums, the paging speed and memory allocated still have an effect on Throughput.

These equations can be used to arrive at the CPU-Utilization and Throughput values for any given configuration and a combination of values for the variables. The study showed that mainly three different configurations are of importance. It also indicated, for the two alternate devices, which variable significantly influences which performance index.

The equations also indicate that more terms are needed to predict throughput than CPU-Utilization. Also, Throughput has less accuracy of prediction (correlation coefficient is smaller) than CPU-Utilization.

TABLE 1

CPU-Utilization Equations

DISK-1

Correlation coefficient = 0.96021

STD. Error = 0.07261

Prediction Equation

$$y = - 0.1609 + 0.0397M + 0.0432P - 0.01749X.M.P + 0.001526P.S - 0.001137M.P - 0.004173\ M^3 + 0.0.00241X.P.S$$

DISK-2

Correlation coefficient = 0.97265

STD. Error = 0.05346

Prediction Equation

$$y = - 0.67103 + 0.1606P + 0.03355 M - 0.006656P^2 + 0.001173M.P - 0.001644X.M.P + 0.0008668 X.P. - 0.0004077 M^e$$

DISK-3

Correlation coefficient = 0.84646

STD. = 0.71650

Prediction Equation

$$y = 0.22704 + 0.01208P - 0.01181X.M - 0.00659M^2 + 0.0004033M.P - 0.0001254 X.M.P$$

DRUM-1

Correlation coefficient = 0.98407

STD. Error = 0.04622

Prediction Equation

$$y = - 0.30262 + 0.1756P + 0.04844M - 0.02843X^2 - 0.05526X - 0.007822P^2 + 0.003963X.P + 0.008068X.M.$$

DRUM-2

Correlation coefficient = 0.99043

STD. Error = 0.03541

Prediction Equation

$$y = 0\ 0.27459 + 0.178P - 0.2784X^3 + 0.04292M + 0.08496X^W - 0.007991 P^2 - 0.001461 X.M.P + 0.006812 X.P + 0.004968X.M.$$

DRUM-3

Correlation coefficient = 0.87260

STD. Error = 0.09524

Prediction Equation

$$y = 0.42236 + 0.10774M + 0.01277C - 0.02955 M^2 + 0.001479M^3 + 0.003827X.M - 0.004576 X^2 - 0.005875 X$$

Legend:
- y: CPU-Utilization
- M: Multi-Programming Level
- P: No. of Pages of Memory
- S: Paging Speed (m-sec)
- X: Job Mix Percent
- C: No. of Words of Memory (in K)

TABLE 2

Throughput Equations

DISK-1

Correlation Coefficient = 0.74337

STD. Error = 3.857

Prediction Equation

$$Z = 37.3428 + 1.3345X.M + 2.6515M\ 14.823 X^3 - 2.1158.S - 0.3386X.S + 0.18909X.P + 0.01422X.P.S - 0.09443X.M.P. + 0.09252P - 0.06537M.S + 0.07955X.M.X$$

DISK-2

Correlation Coefficient = 0.77212

STD. Error = 2.883

Prediction Equation

$$Z = - 11.7283 + 0.4208P + 1.9523X.M + 9.1154 X^3 - 0.8377X.P. + 0.5432.S - 0.3132M^2 + 1.3153M + 0.04018X.P.S + 0.02272 M.P - 0.03458 X.M.S - 0.05516X.S + 0.02028M^3$$

DISK-3

Correlation Coefficient = 0.7724

STD. Error = 2.523

Prediction Equation

$$Z = - 39.148 + 24.7023X^2 - 11.4278 X + 2.443S + 0.20404X.P + 0.9268M - 9.1417X.S + 0.01926.P + 0.05929X.M.S - 0.0301M.P - 0.05188 M.S$$

DRUM-1

Correlation Coefficient = 0.8512

STD. Error = 1.813

Prediction Equation

$$Z = 3.0733 + 1.4114X.M + 8.9236 X^3 - 1.5165 M + 3.3171 X + 0.4502M^2 + 0.4635.P + 0.02667X.M.P - 0.066898 X.P. - 0.01802M.P - 0.028958M^3$$

DRUM-2

Correlation Coefficient = 0.92372

STD. Error = 1.4405

Prediction Equation

$$Z = -3.0591 + 30.8241X^3 + 1.3222.P + 3.2817\ XM - 11.5977.X + 0.055026\ X.M.P - 0.059479P^2$$

DRUM-3

Correlation Coefficient = 0.75131

STD. Error = 3.36743

Prediction Equation

$$Z = 4.8829 + 44.3342X^3 + 0.00124\ X.M.C + 0.04877\ X.C + 0.023718C - 0.003622\ M^3 + 0.001363M.C$$

Legend: Z = Throughput (no. of jobs/unit time)
M = Multiprogramming Level
P = No. of Pages of Memory (1K. Page Size)
S = Paging Speed (in sec)
X = Job Mix (Percent)

REFERENCES

1. Abate, J., H. Dubner, and S. B. Weinberg, "Queuing Analysis of the IBM 2314 Disk Storage Facility", JACM, 15, 4 (1968).
2. Coffman, E. A. and T. A. Ryan, "A Study of Storage Partitioning Using Mathematical Model of Locality," CACM, Vol. 15, No. 3, 1972.
3. Coffman, E. G. and L. C. Varian, "Further Experimental Data on the Behavior of Programs in a Paging Environment", CACM, Vol. 11, 5, 1968.
4. Denning, P. J., "The Working Set Model for Program Behavior", CACM, 11, 5, 1968.
5. Denning, P. J., J. E. Savage and J. R. Spirn, "Models of Locality in Programs Behavior," TR-107, Dept. of Electrical Engineering, Princeton University (1972).
6. Fine, G. H., C. W. Jackson and P. V. McIssac, "Dynamic Program Behavior Under Paging", Proc. Natl. ACM, 21st, (1966).
7. Kuck, D. J. and D. H. Laurie, "The Use and Performance of Memory Hierarchies: A Survey", Software Engineering, Vol. 1, Julius Ton, Ed., Academic Press (1970).
8. Rodriguez-Rosell, J., "Experimental Data on How Programs Behavior Affects Choice of Scheduling Parameters", Proc. 3rd ACM Symp. on o/s Princ. (1971).
9. Seaman, P. H., R. A. Laird and T. L. Wilson, "On Teleprocessing System Design. Pt. IV.--Analysis of Auxiliary Storage Activity", IBM System J., Vol. 5, No. 3, 1966.
10. Shedler, G. S. and C. Tung, "Locality in Page Reference Strings", SIAM J. Comput. 1, 3 (1972).
11. Smith, J. C., "Multiprogramming Under Page on Demand Strategy", CACM, 10 (1967).
12. Teorey, T. J. and T. B. Pinkerton, "A Comparitive Analysis of Disk Scheduling Policies", Proc. 3rd Symp. O/S Princ. (Oct. 1971).

A NOTE ON COMPUTER SYSTEM....

(continued from page 56)

shows periodic behavior. To obtain unbiased measurements the"interarrival times" between sampling points should be exponentially distributed: Poisson measurement process.

Another issue that deserves mentioning is sequential sampling. For instance, since σ in Orchard's eq. (3) is unknown, one may start sampling, compute an estimate s^2, substitute this estimate into eq. (3) continue sampling, update s^2, etc. This more efficient approach(and several variants) is discussed at length in Kleijnen (1975, pp. 479-506). Note that sequential sampling also applies to binary variables.

Stratified sampling briefly discussed by Orchard, is further analyzed in Kleijnen (1975, pp. 110-133). However, other variance reduction techniques may be more attractive, e.g. control variates; see Kleijnen (1975, p.p. 105-285).

References:

1. Kleijnen,J.P.C., STATISTICAL TECHNIQUES IN SIMULATION.(In two volumes) Marcel Dekker Inc., New York, 1974/1975.

2. Orchard, R.A., A new methodology for computer system data gathering. PERFORMANCE EVALUATION REVIEW, 6, no.4, Fall 1977, pp. 27-41.