Performance Evaluation Of Computer Systems In Terms Of Time Delay, Throughput, And Throughput Per Unit Of Cost: Single Processor System, Multiple Processor Network System, And Uniprocessor With Time Sharing Capacity.

by

Sridar Ramamurti

Computer Science Department
Oregon State University
Corvallis, OR. 97331
(503) 754-3273

In partial fulfillment of the requirements for a Masters degree in Computer Science at Oregon State University

MASTER OF SCIENCE

JUNE 1978

APPROVED:

Theodore G. Lewis, Associate Professor
Department of Computer Science
in Charge of Major

Date presented: May, 1978

CONTENTS

A. INTRODUCTION

The information processing industry is one of the fastest growing, and most dynamic industries on the scientific as well as business scene, today [1]. Progress in designing and applying computing systems has out raced progress in evaluating their performance. In order to circumvent this trend, there should be a simultaneous development of guidelines for measuring the performance of a computing system. These guidelines are called measures of performance.

The two basic measures of performance are turnaround time (time delay), and throughput [1]. Turnaround time is defined as the delay between the presentation of input to a system and the receipt of output from it. Throughput is defined as the steady state work capacity of the system. Both throughput and turnaround time depend on internal CPU speed, and each is dependent on the other. It is often possible to increase throughput at the expense of turnaround time or to decrease turnaround time at the expense of throughput. From these two basic measures of performance stem another measure of performance called throughput per unit of cost. Throughput per unit of cost is found by dividing the throughput by the cost of the system.

This paper chooses three types of systems and evaluates their performance in terms of turnaround time (time delay), throughput, and throughput per unit of cost. The three types of systems are single processor system (SP), multiple processor network system (MPN), and uniprocessor with time sharing capacity (UPTS).

The significance of choosing these three systems is to study the improvement of time delay and throughput in a multiple processor network system and uniprocessor system with time sharing capacity over single processor centralized system. Queuing Theory techniques are used to determine time delay and throughput.

The single processor system is modelled in terms of M/M/1 [2]. Multiple server network system is modelled in terms of M/M/K queuing discipline [2]. Uniprocessor system with time sharing capacity is modelled in terms of Round-Robin queuing discipline. At the end a cost function equation is developed for each system and throughput per unit of cost is determined for MPN and UPTS.

B. REPRESENTATION OF SYSTEMS

a. Single Server System:

Figure (1) represents a configuration of a single server system. The single server is the CPU. The arrival rate and service rate are governed by the exponential distribution. The exponential arrival rate (Poisson Arrival) mechanism is singled out for two reasons [3]: it often, but not always, corresponds to actual arrival patterns. Secondly, it is one of the few for which some queuing models can be solved analytically.

Users are selected for service in order of arrival, and each user is given service until the job is processed before the next user is accomodated. This type of system is represented in Queuing Theory as M/M/1 type model where M represents Poisson arrival rate (exponential arrival rate), M represents exponential service rate, and 1 represents single server (CPU). The queuing discipline is first in first out.

b. Multiple Server System

Figure (2) represents a configuration of multiple servers (K servers) system. The system maintains a queue containing K users, each able to have access to one of the K servers. The capacity of each server (CPU) in this system is $1/K^{th}$ of the total capacity C. This type of system is represented in Queuing Theory as M/M/K type model where M represents Poisson arrival rate

(exponential arrival rate), M represents exponential service rate, and K represents K servers (CPU).

c. Round-Robin System

Figure (3) represents a configuration called Round-Robin System which is equivalent to uniprocessor (CPU) with time sharing capacity. Users join a First Come First Serve queue upon entering the system. When they enter the CPU, each user is allowed a quantum of service and are fed back to the queue if they are not finished. In this way every user cycles through the loop (having attained a quantum of service for each cycle) until a required processing time is attained. Once any user has attained the required processing time, the system ejects out that user from the cycle.

C. DERIVATION OF TIME DELAY FOR SP (M/M/1):

The following are the terms needed to derive the formula for the SP system.

λ : arrival rate of users

$1/\mu C$: service rate of users

ρ : traffic intensity factor which is the ratio of arrival rate to service rate

T : Time Delay

L : average length of queue

The average length of the queue, the arrival rate, and the time delay are connected by Little's Formula [2] by the following equation

$$L = \lambda T$$

$$\text{Average length of queue} = \frac{\rho}{1-\rho} \qquad \text{where} \quad = \lambda/\mu C$$

Substituting in Little's equation, we can get the value of T.

$$\text{i.e.} \quad T = L/\lambda$$

$$= 1/\lambda \cdot \frac{\lambda/\mu C}{1-\rho}$$

$$= \frac{1/\mu C}{1-\rho}$$

Referring to Figure (4), we see a graph that shows the relationship between ρ (traffic intensity factor) and time delay. As ρ increases, one notices degradation of time delay (increase in T). This is so because of the fact that under a constant service rate, as arrival rate approaches closely toward the service rate (which is a constant for a particular processor), it is difficult to keep up with the congestion and the result is degradation of time delay. Another graph depicting the relationship between ρ and throughput (1/T) is shown in Figure (5). The same argument holds good for the degradation of throughput.

D. DERIVATION OF TIME DELAY FOR MPN:

The following are the terms needed to derive the formula for the MPN system.

p_K : the probability that the system is busy (system having K servers)

p_o : the probability that the system is empty (system having K servers)

Referring to MPN, the probabilities are expressed as [2]

$$p_K = \frac{p_o (K\rho)^K}{(1-\rho)\ K!}$$

$$p_o = \left[\frac{(K\rho)^K}{(1-\rho)\ K!} + \sum_{n=o}^{K-1} \frac{(K\rho)^n}{n!} \right]^{-1}$$

From these probabilities, and using Little's result as before

$$T = \frac{K}{\mu C} + \frac{P_K}{\mu C(1-\rho)}$$

$$= \frac{K}{\mu C} + \frac{\left[\dfrac{(K\rho)^K}{(1-\rho)K!} + \sum_{n=o}^{K-1} \dfrac{(K\rho)^n}{n!}\right]^{-1} \cdot (K\rho)^K}{K!\ \mu C \cdot (1-\rho)^2}$$

Assuming an interconnection overhead (interprocessor communication overhead), the above formula for T can be written as

$$T = \frac{K_{EFF}}{\mu C} + \frac{\left[\dfrac{(K_{EFF}\cdot\rho)^{K_{EFF}}}{(1-\rho)\ \gamma(K_{EFF})} + \sum_{n=o}^{K-1} \dfrac{(K_{EFF}\cdot\rho)^n}{n!}\right]^{-1} \cdot (K_{EFF}\cdot\rho)^{K_{EFF}}}{\gamma(K_{EFF}) \cdot \mu C \cdot (1-\rho)^2}$$

where K_{EFF} is the value of K such that the effective value of Time Delay includes the interconnection overhead. A Fortran Program to compute the effective value of T for different overhead percentages (from 10 percent to 50 percent in increments of 10 percent) is included in this paper. Figure (6) shows a graph which provides a reasonable justification to show that $\gamma(K_{EFF})$ is a good approximation. In this graph, all the cross marks (x) which correspond to the time delay using $\gamma(K_{EFF})$, fall closely on the 20 percent overhead line. A similar result can be shown for different percentages.

Figure (7) shows a graph which depicts the relationship between ρ and Time Delay. Again, we can use the same explanation for degradation for time delay as for a single server system. Figure (8) shows a graph which depicts the relationship between ρ and throughput.

E. DERIVATION OF TIME DELAY FOR UPTS

The following are the terms needed to derive the formula for the RR system.

$B(x)$: required service time distribution

λ : arrival rate

$n(x)$: average density of customers still in the system who have so far attained x sec of service.

Wo : average waiting time

$T(x)$: Time Delay for a user requiring x secs of processing.

Using Little's result

$$n(x) = \lambda[1-B(x)] \frac{dT(x)}{dx} \qquad (1)$$

Rate of completing service

given an attained service

of X secs $(M(x)) = \frac{b(x)}{1-B(x)}$ (2)

The problem is to find $n(x)$. Assume after one pass through the queue, jobs receive service for a quantum of service 4x. Jobs with X secs of attained service after one pass will have attained X+4x secs of service.

Now $n(x+4x) = n(x)\ [1-M(x)4x] + 0.4x$

where $[1-M(x)4x]$ is the probability that a user requires more service time than x+4x.

As $4x \to o$

$\frac{dn(x)}{dx} = - M(x)n(x)$ which is a first order differential equation. Solving this equation (using equation (2))

$$n(x) = n(o)\ [1-B(x)]$$

Equating the above equation with equation (1) (Little's result)

$$\lambda\ [1-B(x)]\ \frac{dT(x)}{dx} = n(o)\ [1-B(x)]$$

$$\frac{dT(x)}{dx} = \frac{n(o)}{\lambda}$$

$$T(x) = \frac{n(o)}{\lambda} \cdot x \qquad (3)$$

In determining n(o), Kleinrock [2] proposes the following solution. Examine the response time for the case of very large service times. Users who arrive at the system, while a user having a long test job resides in the system, are served to completion before the long test job departs. That is, the test job is assumed the lowest priority job in a preemptive Head of the Line Priority (HOL) system.

Accordingly

$$T(x) = \frac{x(1-\rho) + W_o}{(1-\rho)^2}$$

$$\lim_{x \to \alpha} T(x) = \frac{x}{1-\rho}$$

Using the above equation and equation (3), we finally get the time delay

$$T = \frac{x}{1-\rho} \qquad \text{where x is the required processing time.}$$

$$x = n.1/\mu C \qquad \text{where n is the number of cycles}$$

a user must pass through to complete processing. n is at most (K-1) because during the process, some users might be ejected out of the system after they attain a required service with the result that the number of cycles required at the maximum is K-1. (K is number of users). Thus with K users in the system

$$T = \frac{(K-1)\ 1/\mu C}{1-\rho} \qquad K > 2$$

Assuming a scheduling overhead, the above formula for T can be written as

$$T = \frac{(K_{EFF} - 1)\ 1/\mu C}{1 - \rho} \qquad K_{EFF} > 2$$

where K_{EFF} is the value of K such that the effective value of Time Delay includes the scheduling overhead. The same relationship between ρ and T holds here, and the graphs showing the relationship between ρ and T looks the same

as that for a single processor system except they are magnified by $(K_{EFF}-1)$.

F. COMPARISON OF TIME DELAY FOR SP WITH

the time Delay for MPN and UPTS:

Consider first the time delay for single processor (SP) and MPN system when used by K users.

$$T_{MPNS} = \alpha.K.T_{SP}$$

The significance of this equation can be explained as follows - On the left hand side of the equation, T_{MPNS} expresses the time delay of the system being used by K users. On the right hand side of the equation, $K.T_{SP}$ expresses the fact that K users are using the single processor (by forming a queue). In other words after establishing identical situation in both systems, we are comparing:

$$\alpha = \frac{T_{MPNS}}{K.T_{SP}}$$ i.e. α is the ratio of the time delay of MPNS to the time delay of the SP systems used by K users. This α is a useful factor in evaluating the performance advantage of multiple processor network system (MPNS) as compared to the single processor system.

Consider finally the time delay for SP system and UPTS when used by K users.

$$T_{UPTS} = \beta.K.T_{SP}$$

Again the significance of this equation can be explained in the same way as for the SP system and MPN system.

$$\beta = \frac{T_{UPTS}}{K.T_{SP}}$$

Figure (9) represents the table showing the value of α and β for values of K

ranging from 2 to 10. Figure (10) gives a graph showing the relationship between K (number of users) and the ratios α and β . As K increases, the ratio α decreases. This is interpreted as follows. As the number of users increases, the time delay for the MPNS decreases when compared with the time delay for a SP system being used by K users. With regard to β, as K increases, the ratio β increases. This is so because when K increases, the number of cycles for a user to make in order to get required processing time increases, which is evident from the time delay expression for the UPTS. We note that the cut off point occurs at slightly beyond K = 2.

G. PERFORMANCE COST EQUATION

The following terms are defined in developing the cost equation.

P : is the processor capacity in terms of number of operations per second. Knight [4] indicates that ρ depends upon the internal processing speed of the computer (tc). tc once again depends upon the instruction mix of the user job which in turn depends upon the types of instructions in an instruction mix, and the frequency of their occurrence.

C_c : is communication cost

C_p : processing cost

C_m : is the memory cost to support processor operation.

a. Single Processor System:

The cost equation is expressed as

$$C = n.P.C_p + n.C_m$$ where n is the number of individual single processors.

Figure (11) gives a graph for n versus cost and it is linear (as n increases, the cost increases).

b. Multiple Processor Network System:

The cost equation is expressed as

$C = C_c + n.P.C_p + n.C_m$ where n is number of processors.

Figure (12) gives a graph for n versus C and it is linear.

c. Single Processor With Time Sharing Capacity:

The cost equation is expressed as

$C = C_c + n.P.C_p + n.C_m$ where n is the number of individual single processors with timesharing capacity. Figure (13) gives a graph for n versus C and it is linear.

H. PERFORMANCE IN TERMS OF THROUGHPUT PER UNIT OF COST

Having seen the general cost equation, the next item is to compute the throughput per unit of cost for MPNS and UPTS.

The general equation for cost function for a MPNS is

$$C = C_c + K.P.C_p + K.C_m$$

for the model under consideration in this paper, the power of a single processor (P) is distributed equally over K sites so that the power of a processor at each site is $1/K^{th}$ of the power of a single centralized processor. Hence the cost function equation is

$C = C_c + K.P/_K . C_p + K.C_m$ where P/K is the power of a single processor in MPNS, and K is the number of processors used by K users. The throughput per unit of cost is determined by (1/T)/C where T is the time delay for the MPNS. A Fortran Program to compute this is included in the appendix.

For the UPTS, the cost equation is

$C = C_c + P.C_p + C_m$. The throughput per unit of cost is determined by (1/T)/C where T is the time delay for the UPTS used by K users. A Fortran Program to compute this is included in this paper.

Figure (14) represents a graph for K_{EFF} versus throughput per unit of cost. The cut off point occurs at K_{EFF}=. After this (for values of $K_{EFF} > 3.4$), the throughput per unit of cost for the MPNS is higher compared with throughput per unit of cost for the UPTS. Figures (15), and (16) represent graphs for K_{EFF} versus throughput per unit of cost under different cost parameters.

Figure (17) shows a graph for overhead percentages versus the optimum throughput per unit of cost. As the OH percentage increases, the optimum throughput per unit of cost decreases. Figure (18) shows a graph for K_{EFF} versus the optimum throughput per unit of cost.

Figure (19) and (20) represent a graph for K versus the ratios of cost parameters (C_p/C_c and C_m/C_c) at the optimum value of K (=3.4) as seen in Figure (14).

I. CONCLUSION

Three types of queuing models have been introduced to represent three types of systems: M/M/1 model for SP System, M/M/K model for MPNS, and Round-Robin System for UPTS. From these models, the performance in terms of time delay (turn around time) and hence throughput for a MPNS is better than for a UPTS or a SP system. The main concept that is derived out of this paper is that it is advantageous performance wise to distribute the computing power rather than having a centralized computing power.

J. REFERENCES

1) Peter Calingaert. System Performance Evaluation: Survey and Appraisal. CACM 10,1 (Jan. 1967) pp. 12-18.

2) Leonard Kleinrock. Queuing Systems. Vol. II. Computer Applications. Wiley-Interscience Publications. John Wiley & Sons, New York, 1976.

3) Herbert Hellerman and Thomas F. Conroy. Computer System Performance. McGraw-Hill Book Company, New York 1975.

4) Kenneth E. Knight. Changes in Computer Performance. Datamation September, 1966. pp. 40-54.

K. APPENDIX

Program listings for the cost models used in this study.

-Service as referee for numerous journals and conferences in computer science and management science

-Chairman of dissertation committee (principal advisor) for four doctoral dissertations completed during 1976-78, member of two other completed dissertation committees. Currently member of three dissertation committees (chairman of one) and two masters thesis committees

-Principal investigator on current NSF grant No. MCS77-02715, "Dataflow Organization of Databases"

Publications

1. An Introduction to Information Processing Language V. Communications of the ACM, Vol. 3, No. 4, April 1960, (with A. Newell).

2. Summary of a Heuristic Line Balancing Procedure. Management Science, Vol. 7, No. 1, 1960, (reprinted in Feigenbaum and Feldman), Computers and Thought, McGraw-Hill, 1963).

3. A Heuristic Program for Assembly Line Balancing. Prentice-Hall, Inc., 1961.

4. The Use of Heuristic Programming in Management Science. Management Science, Vol. 7, No. 3, 1961, (reprinted in Starr, Executive Readings in Management Science, MacMillan Co., 1965).

5. Empirical Explorations of a Hypotheses-Testing Model of Binary Choice Behavior. In Hoggatt, A.C. and F.F. Balderston, Ed., Symposium on Simulation Models: Methodology and Applications to the Behavioral Sciences, South-Western Publishing Co., 1963, (with J. Feldman and H. Kanter).

6. Balancing Assembly Lines Using the General Problem Solver, in Hoggatt and Balderston, eds. Symposium of Simulation Models: Methodology and Applications to the Behavioral Sciences, South-Western Publishing Co., 1963.

7. Information Processing Language V Manual. 2nd ed., Prentice-Hall, 1964, (with A. Newell, E. Feigenbaum, B. F. Green, Jr., and G. Mealy).

8. Assembly Line Balancing Using Probabilistic Combinations of Heuristics. Management Science, May 1965 (reprinted in Hottenstein, Models & Analysis for Production Management, International Textbook Company, 1968).

9. QUICKSCRIPT: A SIMSCRIPT-Like Language for the G-20. Communications of the ACM, June 1965, (with P. Keller and A. Newell).

10. A Look Ahead - Coming Developments in Production and Inventory Control. American Production and Inventory Control Society, Quarterly Bulletin, 1965, 6, No. 2.

11. Some Reflections on a Survey of Research Problems in Computer Science. RM-4467-PR, The RAND Corporation, 1965.

12. Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming Problems. Management Science, Vol. 12, No. 7, 1966, (with G. Thompson and S. Zoints).

13. A View of Artificial Intelligence. Proceedings of the ACM National Meeting, 1966.

14. ICS-1 Laboratory Manual. University of California, Irvine, 1966, (with J. Feldman).

15. A Simple Scheme for Formalizing Data Retrieval Requests. The RAND Corporation Research Memorandum, RM-5150-PR, 1967.

16. A section in Computers in Education, Ralph W. Gerard (ed.), on the UCI Computer-Assisted Instruction Conference, McGraw-Hill, 1968.

17. Design of a Programming Language & System for Computer Assisted Learning. IFIPS Congress '68, Vol. 1, pp. 1115-1119, North Holland Publishing Company, 1968.

18. Hierarchical Aspects of Computer Languages, in L. L. Whyte, A. G. Wilson, D. Wilson, eds., Hierarchical Structures, American Elsevier Publishing Company, Inc., 1969.

19. Using Computers in Teaching about Computers, presented at the DLJ Foundation Curriculum Workshop on Using Computers in Management Education, Harvard University, February 1969, proceedings to be published in 1969).

20. Threading for Endorder Traversal. University of Hawaii, Honolulu, The Aloha System, Technical Report A70-6, April, 1970.

21. Hardware Mapping of Tables. Proceedings of Fifth Hawaiian Conference on System Sciences, 1972.

22. Contributor to Ashenhurst, R. L. (ed.) "Curriculum Recommendations for Graduate Professional Programs in Information Systems," Communications of the ACM, May 1972.

23. Reflections on Computer-Oriented Curricula for Management. Journal of Contemporary Business, Vol. 1, No. 2, Spring 1972.

24. Reflections on the Problem of Characterizing Information Systems. Invited discussion, the Wharton Conference in Research on Computers in Organization, appears in Data Base Vol. 5, Nos., 2, 3, and 4, Winter 1973.

25. A Note on Response Time and Saturation. University of California Technical Report #40, January 1974, Information and Computer Science.

26. Procedures and Procedure-Followers, (with Julian Feldman). McGraw-Hill Publishing Co., 1975.

27. Data Representation and Synthesis. University of California, Irvine, Technical Report #63, Information and Computer Science, 1975, (with Lawrence A. Rowe).

28. Storage Structures Formalism. University of California, Irvine, Technical Report #64, Information and Computer Science, 1975, (with Lawrence A. Rowe).

29. Expressions for Time and Space in a Recursive Realization Parallelism. University of California, Irvine, Technical Report #79, Information and Computer Science, 1976.

30. A Selection Algorithm for Coalesced Implementation Structures, (with Lawrence A. Rowe), 1976.

31. Algorithms for the Synthesis of Implementation Structures, University of California, Irvine (with Lawrence A. Rowe), IEEE Transactions on Software Engineering, November 1978. (Also available as TR #91, ICS Dept., UCI).

32. The Last Ten Years. In Instructional Computing in the University: The Second Decade, Feldman, J., Mosmann, C., eds., 1977, San Francisco Press.

33. Pipelining Performance of Structured Networks, University of California, Irvine Technical Report #117, 1977 (submitted to IEEE Transactions on Computers).

34. A Parametric Disk Scheduling Policy (with Ram P. Singhania), 1978 (submitted to Communications of the ACM).

35. Structured Process Description (with Robert S. Barton and Richard M. Cowan), University of California, Irvine, Technical Report #130, Information and Computer Science, 1979.