OPTIMAL LOAD CONTROL IN COMBINED BATCH-INTERACTIVE COMPUTER SYSTEMS

by

Samuel T. Chanson and Prem S. Sinha

Dept. Of Computer Science

University of British Columbia

April 1980

TR 80-5

1. Introduction

Most large scale computer systems employ some form of load control to maintain a high throughput rate and/or to provide an acceptable level of service to the users. For paging systems this is often accomplished by manipulating the degree of multiprogramming or equivalently the size of the resident sets of the active processes to keep the system from becoming saturated. Previous work directed towards this end is primarily represented by the development of the working set policy [1,2,3,4]. More recently, efforts to optimize the system work capacity lie mainly in keeping some measures related to program behavior (usually paging behaviour) within some predetermined bounds [5,6,7,8]. The 50% criterion [7] for example, aims at maintaining the utilization of the paging device to around 0.5. The L=S criterion [6] proposes to keep the system life time to approximately that of the page swap time. The knee criterion [5,8] suggests that the mean resident set size of each process should be maintained at the value associated with the primary knee of its life time function. Though the most robust of the three, the knee criterion is also the most costly to implement and involves the largest amount of overhead.

Though these criteria are not based on mathematical models and cannot be proved to be optimal, they aim at increasing the throughput rate by loading the system up to the point when the measured indicator suggests further increase in system load may cause 'thrashing'. The methods cannot be applied to non-paged systems. Furthermore, for interactive systems and combined batch-interactive systems, one is interested not only to maximize the system throughput rate but also to guarantee good response times to the interactive jobs (possibly at the expense of the batch jobs). Landwehr [9] studied a combined batch-interactive system and proposed a scheme to activate batch jobs based on the terminal load. The emphasis of the study, however, was on model formulation and validation. There was no attempt to prevent the system from saturation or to optimize performance. As well, there is no easy or systematic way of determining the values of the break points. Hine et.el. [10] studied the problem from a slightly different viewpoint. Their goal was to control the main memory allocation for each class of jobs to provide different response times to each while maximizing the CPU utilization. They employed a mathematical model but optimization was achieved by an exhaustive search technique. A heuristic was also given which provides good but not optimal results.

In this paper we study the performance of a combined batch-interactive computer system using the operation analysis technique [14]. The control algorithm determines from time to time the number of batch jobs (if any) to be activated from the batch queue. The control criterion aims at keeping the system from saturation (to be defined in the next section) while minimizing the mean number of jobs waiting to be activated. The effect is to maximize the throughput of the system while maintaining good response time for interactive jobs.

2. Estimation of System Saturation

Definitions of system saturation have been proposed [12,13,14]. Invariably the system is considered saturated at the point the response time vs system load curve starts to rise rapidly. Kleinrock [12], for example, using the number of active terminals as the load, defined the system's saturation point to correspond to the intersection of the mean normalized response time curve asymptote and the horizontal line corresponding to the minimum response time (i.e., when there is only one active terminal). (See Figure 1). If a system is not allowed to get saturated according to this definition, the mean response time of the active jobs will not exceed an acceptable level. However, the implicit assumption is that the program population considered is both homogeneous and stationary. Our approach is to compute the system saturation load at small intervals (such as a few seconds) during which the stationary assumption is justified. The homogeneous assumption is discussed below.

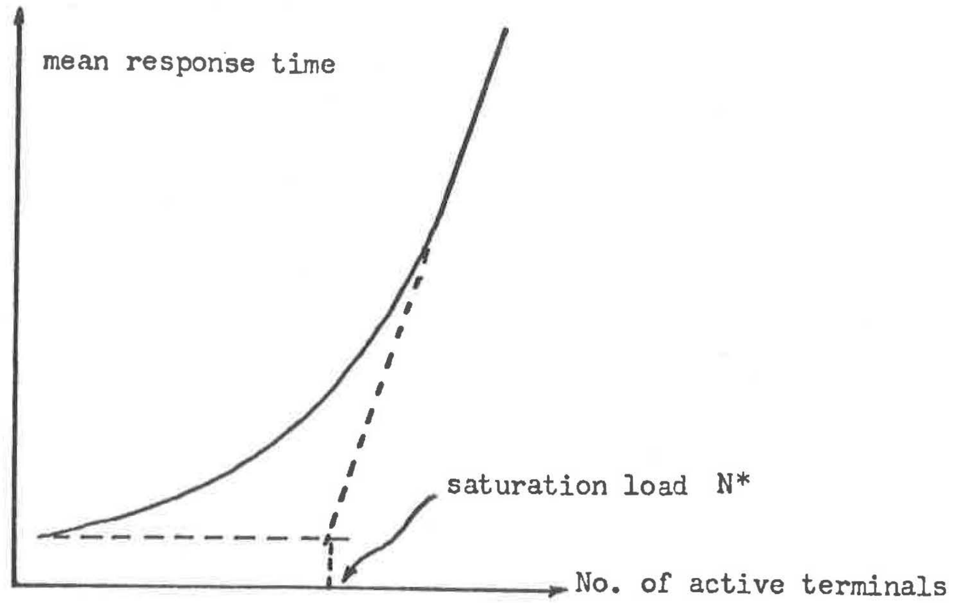

Figure 1. Mean response time vs. the number of active terminals.

The computer system is often modelled as a central-server model [11,13]. Consider such a model with M service centres and a degree of multiprogramming equal to N. Each service centre consists of a device and its associated queues (Fig 2.)

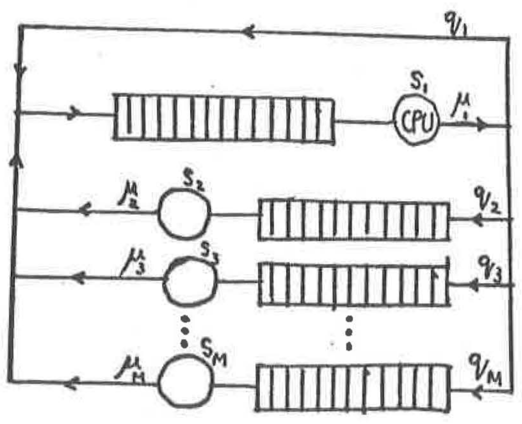

Figure 2. Central-server model with M service centres.

The service centre S1 is the CPU service centre (the central server). On completion of the CPU service a job either leaves the system or joins another service centre. A job leaving service centre S_i, $i=2,3,\dots M$ must join the central server.

2.1 Notations

The quantities defined in section 2.1 and computed in section 2.2 are mean values within an observation period and as such are functions of time which is omitted for clarity.

T : observation period

C_i : observed number of completions at centre S_i during T

B_i : the total amount of time during which the service centre S_i is busy during T

R_i : observed number of requests for centre S_i during T

q_i : request frequency, the fraction of jobs proceeding next to service centre S_i on completing a service request at the central server

$=R_i/C_1$, $i \neq 1$

q_1 is the fraction of jobs leaving the system on completing a service request at the CPU. $\left(\sum_{i=1}^{M} q_i = 1\right)$.

2.2 Operational quantities

mean service rate of server $S_i = \mu_i = C_i/B_i$

utilization of server $S_i = \rho_i = B_i/T$

system throughput rate $\bar{T} = \dfrac{C_1 q_1}{T}$

$$= \frac{C_1}{B_1} \cdot \frac{B_1}{T} \cdot q_1$$

$$= \mu_1 \rho_1 q_1 \qquad \text{———} \qquad (1)$$

utilization $\rho_i = B_i / T$

$$= \frac{B_1}{T} \cdot \frac{C_1}{B_1} \cdot \frac{R_i}{C_1} \cdot \frac{B_i}{R_i}$$

$$= \rho_1 \mu_1 q_i \cdot \frac{B_i}{R_i} \qquad i \neq 1$$

Under the job flow balance assumption $C_i = R_i \quad i = 2, 3, \ldots M$

$$\therefore \rho_i = \rho_1 \mu_1 q_i \frac{B_i}{C_i} \qquad i \neq 1$$

$$= \frac{\rho_1 \mu_1 q_i}{\mu_i} \qquad i \neq 1 \qquad \text{———} \qquad (2)$$

When there is only one job in the system

$$\sum_{i=1}^{M} \rho_i = 1$$

from (2) $$\rho_1 + \sum_{i=2}^{M} \frac{\rho_1 \mu_1 q_i}{\mu_i} = 1$$

$$\Rightarrow \quad \rho_1 \left(1 + \sum_{i=2}^{M} \frac{\mu_1 q_i}{\mu_i} \right) = 1 \qquad \text{———} \qquad (3)$$

We use Little's law to compute the mean response time

$$\bar{R}(N) = N / \bar{T} = \frac{N}{\mu_1 q_1 \rho_1}$$

$$= \frac{N q_i}{q_1 \rho_i \mu_i} \qquad \text{———} \qquad (4)$$

from (3) $$\bar{R}(1) = \frac{1}{\mu_1 q_1} \left(1 + \sum_{i=2}^{M} \frac{\mu_1 q_i}{\mu_i} \right) \qquad \text{———} \qquad (5)$$

The equation of the asymptote (as N approaches infinity) is more difficult to derive. Let us first consider the simple case of a non-virtual memory system. The asymptote occurs at the point when the utilization of a service centre (i* say) reaches unity (i.e., it becomes the system's first bottleneck).

From Buzen's analysis [11], i* is that service centre which has the highest utilization in an interval (i.e., i* may vary from interval to interval as the work load characteristics change). If it is the CPU, the equation of the asymptote is simply

$$\bar{R}(N) = \frac{N}{\mu_1 q_1} \qquad \text{———} \quad (6)$$

Otherwise, using equation (4) and noting that μ_i as well as the ratio (q_i/q_1) remains unchanged as N increases, the equation of the asymptote is

$$\bar{R}(N) = \frac{N q_{i*}}{\mu_{i*} q_1} \qquad \text{———} \quad (7)$$

For a paging system, the eventual bottleneck as N approaches infinity must be the paging device but it need not be the first device to saturate.

Case (i), the paging device is not the first to saturate.

In this case, as the system is saturated before the paging device is fully utilized, the asymptote should be computed based on the first device to reach saturation and equation (7) is still valid (see Figure 3).

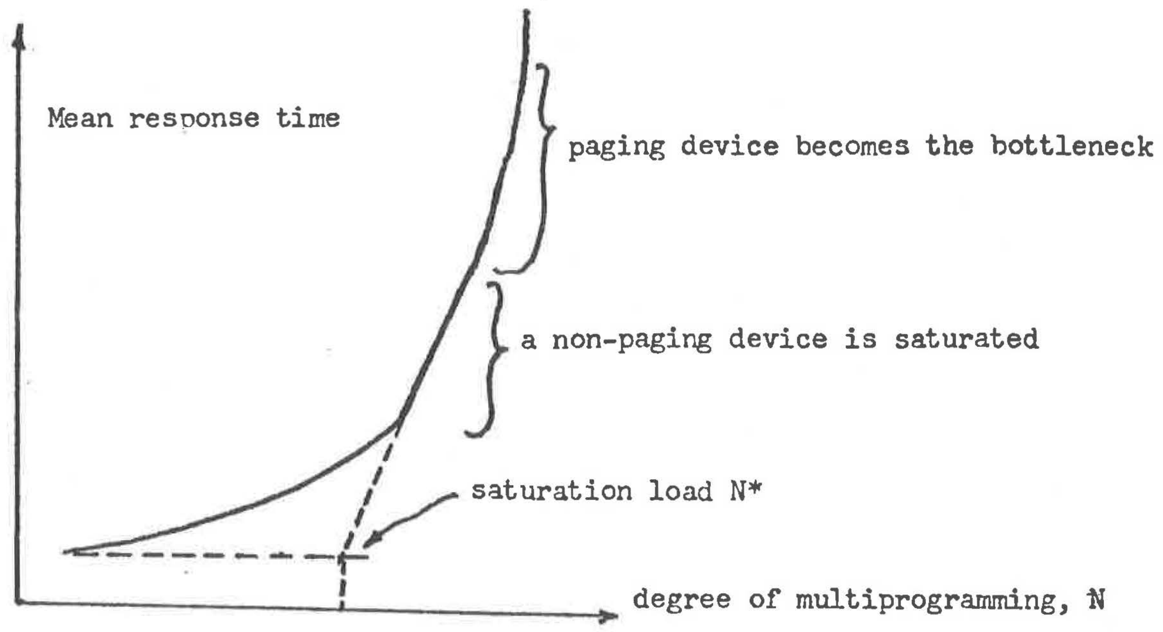

Figure 3. Mean response time vs. N, a non-paging device is the first to be saturated.

Case (ii), the paging device is the first to saturate.

The ratio q_{i*}/q_1 continues to increase as N increases and approaches infinity as N approaches infinity. A realistic approach consistent with the one used in Case (i) is to use the value of q_{i*}/q_1 corresponding to the point the paging device first becomes fully utilized. However, this ratio is not easy to estimate. The observed value of q_{i*}/q_1 can be used if the system is close to saturation (i.e., $N^* \doteq N$, see below) when the parameters are measured. Otherwise the saturation load will be under-estimated. This is not a problem when the work load is light. As can be seen subsequently, if the system work load then gets heavy, the control policy will adjust to it and the observed ratio will again approach the desired value. The saturation load N* is thus the intersection of equations (5) and (6)

$$N^* = 1 + \sum_{i=2}^{m} \frac{\mu_1 q_i}{\mu_i} \qquad \text{———} (8)$$

if the CPU is the first device to saturate. Otherwise it is the intersection of equations (5) and (7).

$$N^* = \frac{\mu_{i*}}{\mu_1 q_{i*}} \left(1 + \sum_{i=2}^{M} \frac{\mu_1}{\mu_i} q_i\right) \text{———} (9)$$

All of the above equations can also be derived using queuing theory.

Most proposed schemes assume a fixed saturation load. The Michigan Terminal System [15] for example computes the values of five load factors at fixed intervals and if one or more exceeds the corresponding predetermined static saturation value, the system is assumed to be saturated. For the 50% criterion, the saturation point corresponds to when the utilization of the paging device exceeds 0.5+c, where c is some small positive constant. The L=S criterion to certain extent assumes the system to be saturated when the system life time is below the page swap time, which is fixed for a given paging device.

In a previous report [15], we have shown that the saturation load is really a function of the characteristics of the current work load and cannot be very well represented by some constant measures. For the present model, these work load characteristics are q_i and μ_i, i=1,2,..,M. Any model which does not take this into consideration will sometimes over-estimate and sometimes under-estimate the system saturation load. The fact that the over-estimation on the average is equal to the under-estimation provides no comfort when the goal is to optimize performance at all times.

3. Load control

The first criterion for load control is to keep the system from saturation. From Figure 1, it is seen that the mean response time increases rapidly beyond this point. Furthermore Denning [16] has shown that 'thrashing' (and thus reduced throughput rate) occurs when the paging device is saturated. For multiprogrammed paging computer systems, the simplest way to accomplish this is to keep the number of active jobs below N^* given in equation (8) or (9). Since the system throughput rate is a non-decreasing function of N *before* the system saturates [8], activating N^* jobs whenever possible will also maximize the throughput. There are three cases to consider:

(i) the system is saturated (i.e. $N > N^*$),

(ii) the system is under utilized (i.e., $N \ll N^*$),

(iii) the system is close to but not yet saturated.

Case (iii) is the interesting case since if the system is under utilized, it is unnecessary to apply any control measure but to activate each job as it arrives until the condition for case (iii) is reached. If the system load is then properly controlled, the system should attain saturation (case(i)) infrequently and only for brief periods. The control when the system is saturated could simply consist of not activating any more batch job until the system comes out of saturation. If the system is in the saturation state frequently and for extended durations then it is highly probable that the hardware is inadequate to handle the normal work load and should be upgraded. Thus we shall consider only case(iii) in this paper. We note that many systems (e.g.,the Michigan Terminal System [15]) do not apply any control until saturation is detected. This, in our opinion, does not constitute proper load control.

4. Optimization

We define the following variables all of which are functions of time t which is omitted for clarity :

- I : number of batch jobs waiting to be activated,
- A_t: number of terminal requests in the interval,
- E_t: expected number of terminal arrivals in the next interval,
- D : expected number of job departures (both batch and terminal) in the next interval,
- N_b: optimal number of batch jobs that should be activated,
- N_t: optimal system capacity (measured in terms of the number of jobs) to be reserved for expected incoming terminal jobs.

S : remaining system capacity (defined as the number of additional jobs that can be accommodated without saturating the system).

S can be approximated by

$$S = N^{*} - N + D$$

The problem is to determine how many of the S jobs should be filled by waiting batch jobs (if any). Our objective is to maximize the mean system throughput rate without saturating the system. This is equivalent to minimizing the expected number of jobs that have to wait at each interval because admitting them would saturate the system. We shall minimize a weighted sum of the waiting batch and terminal jobs which is a more general problem.

Let the weights be C1 and C2 for batch and terminal jobs respectively. The optimization problem is therefore

$$\text{Min} \left\{ C_1 (E_t - N_t) + C_2 (I - N_b) \right\} \text{———} (10)$$

$$\text{subject to} \begin{cases} N_t + N_b \leq S \\ N_t \leq E_t \\ N_b \leq I \end{cases} \text{———} (11)$$

The problem is equivalent to

$$\text{MAX} \left\{ Z = C_1 N_t + C_2 N_b \right\}$$

subject to (11).

If C1 > C2 (i.e., terminal jobs are favoured), it is easy to see that the solution to the above optimization problem is

$$\left.\begin{array}{ll} N_t = E_t \text{ if } E_t < S & \Rightarrow N_b = S - E_t \\ N_t = S \text{ if } E_t \geqslant S & \Rightarrow N_b = 0 \end{array}\right\} \text{——(12)}$$

In the above computation ,it is assumed that E_t and D are available at the beginning of the interval. If E_t<D then we may have $N + N_b > N^*$ for a short period at the beginning of the interval. This problem can be alleviated by spreading out the activation of the N_b batch jobs throughout the interval instead of all at once at the beginning. It remains to show how E_t and D can be computed using smoothed statistical estimates.

Let P_t be the expected prediction of the parameter for the period $[t, t+1]$. Let x_t be the observed value of the parameter at time t. P_t can be expressed as

$$P_t = (1-\beta)\left\{x_t + \beta x_{t-1} + \beta^2 x_{t-2} + \cdots\right\}$$

$$= (1-\beta)\sum_{j=0}^{\infty} \beta^j x_{t-j} \qquad \text{———— (13)}$$

where the exponential weight factor β is a constant between zero and one. For t-1,

$$P_{t-1} = (1-\beta)\sum_{j=0}^{\infty} \beta^j x_{t-j-1}$$

Now let the error made at time (t-1) in predicting x_t be ϵ_t, then

$$\epsilon_t = x_t - P_{t-1} \qquad \text{———— (14)}$$

substituting in equation (13),

$$P_t = P_{t-1} + (1-\beta)\,\epsilon_t \qquad \text{———— (15)}$$

Now if ϵ_t is relatively small we do not recompute the value of β. If ϵ_t is large, we find a new value of β which will minimize the sum of the squares of errors given by

$$\sum_{t=t_0}^{-\infty} \left\{ x_t - (1-\beta) \sum_{j=0}^{\infty} \beta^j x_{t-j-1} \right\}^2 \qquad \text{———— (16)}$$

In practice, the summation in (16) does not have to involve many (k,say) terms before β^K approaches zero. β does not have to be very accurate and standard techniques exist for its efficient computation.

To summarize, the control procedure consists of the following steps:

1. During an interval T, observe D, A_t, N, q_i, μ_i, i=1,2,...M.
2. Compute N* using equation (8) or (9).
3. Estimate the expected number of terminal arrivals and total departures in the next interval using equation (15).
4. Compute the number of batch jobs N_b to be activated in the

next interval using equation (12).

5. Terminal jobs are immediately activated upon arrival.

5. Conclusion

A model to estimate the saturation of a computer system has been presented which is capable of adjusting to varying work load characteristics. Based on this, the number of batch jobs that should be activated to minimize a weighted sum of the number of jobs that will have to wait upon arrival without saturating the system in a combined batch-interactive environment is computed using simple optimization theory. The approach thus provides good mean response time to the terminal jobs and maximizes the system throughput rate under that condition.

The second level of load control - that of the selection of the type of jobs to be activated has not been discussed in this paper. Work has started in this direction. However, because of the difficulty of predicting accurately the resource demands of a job before it is executed and because of the adaptiveness of the proposed scheme which is capable of correcting itself it may be that equally good results can be achieved without it.

The use of the number of jobs to characterize the system load is of course not precise as jobs do not necessarily have the same resource demand characteristics. However, it has been shown to produce useful results [17,18]. It is made even more acceptable for this application because of the built-in adaptiveness of the proposed policy. If the average resource demands of the activated jobs in the next time interval is lighter than those of the previous interval (on which the remaining system capacity was estimated), then the updated remaining system capacity will increase and more jobs will be activated in the next interval. On the other hand, if they are heavier, then the remaining system capacity will decrease and fewer (or no) jobs will be activated in the next interval.

The model as proposed is very simple and all the parameters required can be measured directly.

<u>REFERENCE</u>

[1] Denning,P.J.,"The working set model for program behaviour",<u>Comm.</u> <u>of</u> <u>ACM</u>15,5 (May 1968),323-333.

[2] Denning,P.J.,"Virtual memory",<u>Computing</u> <u>Surveys</u> 2,3 (Sept. 1970),153-189.

[3] Denning,P.J.,"Third generation computer systems", <u>Computing</u> <u>Surveys</u> 3,4 (Dec. 1971),175-216.

[4] Rodriguez-Rosel,J. and Dupuy,J.P.,"The design ,implementation, and evaluation of a working set dispatcher", <u>Comm.</u> <u>of</u> <u>ACM</u> 16,4 (April 1973),247-253.

[5] Denning,P.J., Kahn,K.C., Leroudier,J. Potier,D. and Suri,R.,"Optimal multiprogramming", <u>Acta</u> <u>Informatica</u>,7 (1976),197-216.

[6] Denning,P.J. and Kahn,K.,"An L=S criterion for optimal multiprogramming",<u>Proc.Int'l.Symp.</u> <u>on</u> <u>Computer</u> <u>Performance</u> <u>Modeling,Measurement</u> <u>and</u> <u>Evaluation</u> , (March 1976),219-229.

[7] Leroudier,J., Potier,D.,"Principles of optimality for multiprogramming",<u>Proc.Int'l.Symp.</u> <u>on</u> <u>Computer</u> <u>Performance</u> <u>Modeling,Measurement</u> <u>and</u> <u>Evaluation</u> (March 1976),211-218.

[8] Graham,G.S. and Denning,P.J.,"On the relative controllability of memory policies",<u>Proc.Int'l.</u> <u>Symp.</u> <u>on</u> <u>Computer</u> <u>Performance,Modeling,Measurement</u> <u>and</u> <u>Evaluation</u>, (August 1977),411-428.

[9] Landwehr,C.E.,"An endogenous priority model for load control in combined batch-interactive computer systems",<u>Proc.Int'l.Symp.</u> <u>on</u> <u>Computer</u> <u>Performance</u>,

Modeling,Measurement and Evaluation,(March 1976),282-287.

[10] Hine,J.H., Mitrani,I. and Tsur S.,"The control of response times in multi-class systems by memory allocation",Comm. of ACM 22,7 (July 1979),415-424.

[11] Buzen,J.p.,"Analysis of system bottlenecks using a queuing network model",Proc. ACM-SIGOPS Workshop on System Performance Evaluation (April 1971)82-103.

[12] Kleinrock,L., "Certain analytic results for time-shared processors", Information Processing (Proc. IFIP Congress 68),(1968),838-845.

[13] Ferrari,D.,Computer Systems Performance Evaluation, Prentice Hall,1978.

[14] Denning,P.J. and Buzen,J.P.,"The operational analysis of queuing network models",Computing Surveys 10,3(Sept. 1978),225-261.

[15] Chanson,S.T.,"Saturation estimation in interactive computer systems", Technical Report 79-7, Dept. of Computer Science, University of British Columbia, (June 1979).

[16] Denning,P.J.,"Thrashing: its causes and prevention", Proc. FIPS,33,(FJCC, 1968),915-922.

[17] Scherr,A.L.,"An analysis of time shared computer systems",Ph.D.Thesis, Dept. of Electrical Engineering,M.I.T., Cambridge, Mass. (June 1965).

[18] Chanson,S.T. and Ferrari,D.,"A deterministic analytic model of a multiprogrammed interactive system", NCC, AFIPS Conference Proc.,43(1975),645-652.