THE EFFECTS OF PROGRAM SEGMENTATION ON JOB COMPLETION TIMES IN A MULTIPROCESSOR COMPUTING SYSTEM

Richard K. Dumas
UNIVAC Federal Systems Division
St. Paul, Minnesota

This paper is the result of a study to determine the relationship between job completion times in a multiprocessor system and the manner in which the component programs of the jobs are segmented.

The nature of jobs consists of a set of functionally related programs some of which could or must be executed in parallel. Thus the work for the multiprocessor system is described in the form of a network of programs. A typical network may appear as in Figure 1. The nodes in Figure 1 act as "and" nodes for this study. For example, program E cannot be executed until programs B and C have been completed. Several networks may be in various stages of execution at any given time. In particular the study was concerned with minimizing the completion time of the highest priority network.

Program segmentation divides a program into executable units of work constrained to be less than or equal to a fixed time called the segment time. When each segment is completed there is a return to the executive to determine on a priority basis which segment is to be executed next. Thus, when new work appears in the system and this work is of higher priority than work currently being executed, a maximum of one segment time unit will pass before the system begins execution on the segments of the new work. It is apparent that the segmenting scheme affects the completion times of the highest priority network in two unique ways:

1. There are delays due to processors not being able to transfer control to higher priority jobs immediately, but having to wait a maximum of one segment.

2. There are delays due to the periodic forced scan of the segment list by the executive.

Upon initiation of the study it was assumed that if program segment time was plotted against network completion time, the resulting curve would exhibit a minimum. With reference to this minimum, as segments became shorter than the optimum time, network completion time would increase because of executive overhead. Similarly, for segment times longer than optimum, network completion time (for high priority jobs) would again increase since the segment list is examined less frequently, thereby increasing the average time between a program being eligible to be executed and its actual execution.

A model of a multiprocessor system was constructed using the UNIVAC Simulation Package (USP), a discrete simulation language similar to GPSS. Only the processors and their interfacing with the networks of programs via the segment list are considered in the simulation. The programs are represented as units of traffic (transactions) and the processors as single server devices. The model allows segment time, executive overhead, the number of processors, and the networks of programs to be easily changed.

It was decided that coarse initial simulations would be used to establish direction in the study. Then other implicit statistical conclusions would be uncovered by more comprehensive simulation. The comprehensive simulations required the input of a large number of networks. To meet this requirement a network generator was constructed within the framework of USP. This generator interfaces directly with the multiprocessor model allowing the user to vary the above system variables and consider the effects of the variations on a large number of sets of networks during one USP compilation. The paper describes the model and the network generator in detail.

The initial attempt to verify that the simulation model correctly represented the simulated system involved comparing the resulting simulation network completion times with hand-calculated Gantt charts of the processors' activities. The first model was created such that the simulated executive system control was performed by the internal USP control. Discrepancies between the simulation results and Gantt charts pointed out implicit problems in using the internal USP control in this manner. Therefore another model was written in which the executive system control was implemented by the programmers use of USP instructions. In other words, the simulated executive system control was removed from USP control. Both models were verified again through comparisons with Gantt charts. Then more runs were made using the same inputs to both models. Identical outputs gave confidence that the executive system was being properly modeled.

The study results indicate that for the highest priority network, the relation between network completion times and segment times is described by a parametric family of curves as is shown by Figure 2. The executive overhead $\emptyset$ incurred in executing a segment is the parameter. Each curve may be approximated by a sum of one of the curves of Figure 3 and the curve of Figure 4. Figure 3 shows the relation between segment time and total executive overhead incurred in executing the complete network. Again, $\emptyset$ is the parameter. Figure 4 relates, for $\emptyset = 0$, the increases in network completion to increases in segment times. As was expected, the increases in segment times result in delays in processing on the highest priority network, since the processors may be occupied for increasing periods of time on lower priority work and thus not scan the segment list as frequently.

The values of $\emptyset$ for which optimum segment times exist and the value of the optimum segment time may both be found from simulation results. The study shows, however, that the form of the curve of Figure 4 is dependent on the lower priority networks. Thus the relations of Figure 2 may not be determined simply from the properties of the highest priority network.

User guidelines in scheduling work under this program segmentation scheme cannot be finalized until remaining areas of study are complete. These include:

1. Determination of meaningful measures of network characteristics.
2. Variations in the method of removing segments from the segment list.
3. Allocating a unique segment time to each network.
4. Investigation of the dependency of optimal segment times on variations in program execution times for fixed networks of programs.
5. Effects of changing the number of processors.

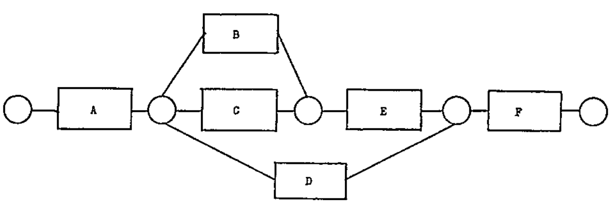

Figure 1 - Network of Programs

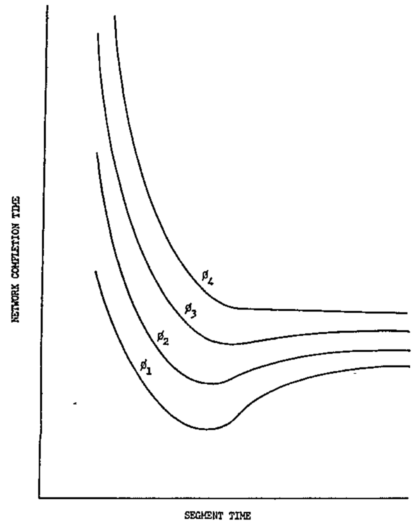

Figure 2 - Optimal Segment Times

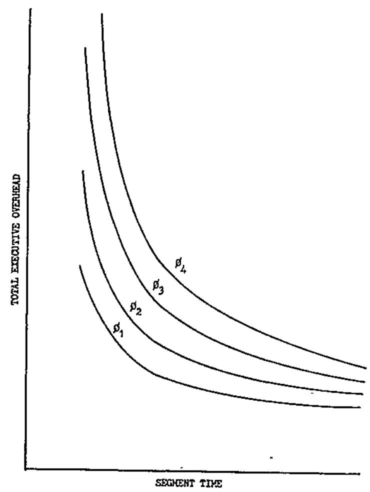

Figure 3 - Delay Due to Processor Unavailability

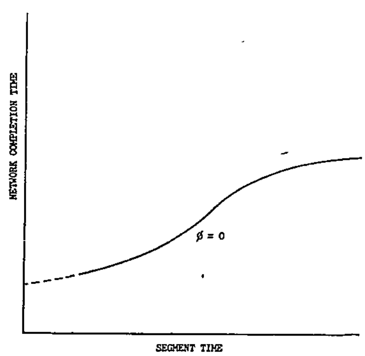

Figure 4 - Delay Due to Executive Overhead