A Simulation Model for Evaluating Initiator Structures

H. Pat Artis

Bell Laboratories
Piscataway, New Jersey 08854

Abstract: The design and implementation of a discrete simulation model of initiator structures for IBM's MVS operating system is presented. This model was developed for evaluating the service characteristics of alternative initiator structures for processing testing workloads.

1. INTRODUCTION

An initiator is a multi-class server that selects jobs for processing by a computer system. By specifying the number of initiators and the job classes each serves, the user may limit the system's multiprogramming level, provide different levels of service to various job classes[1] and allow the operating system to perform the majority of the job scheduling functions. It has been the author's experience that the most extensive use made of initiator structures is to provide various levels of service to job classes in testing environments. For this study, service is defined as the length of the delay between a job's arrival and its selection by an initiator. The purpose of this investigation was to design a trace-driven simulation model for evaluating the service characteristics of alternative initiator structures for testing workloads.

The most important decision in the development of the model was the selection of the level of detail at which the initiator process would be modeled. Basically, this was a choice of whether or not to model the execution (server holding time) of jobs in detail. Using highly complex queuing models, other investigators [2,3] have concentrated on the effects of job interactions on execution time. However, in a testing environment, job execution times are usually quite small in comparison to their input queuing delay. Hence, this investigator chose to assume that the execution time of jobs would be unchanged when executed under a new initiator structure with a multiprogramming level less than or equal to the environment in which the trace data for the simulation was collected. This assumption allowed the investigation to concentrate on modeling the input queuing delay phenomenon.

Although this assumption initially seemed quite weak, investigations of average execution times of a number of resource-limited job classes revealed only small variations in execution time for a variety of multi-programming levels. Moreover, actual experiences with the model, which will be discussed in later sections, have shown the effects to be negligible when the multi-programming level decreases or is unchanged, provided that each class consists of jobs that have relatively small homogeneous resource demands.

2. DATA COLLECTION

The data for the modeling effort was collected from the Systems Management Facility (SMF) [4] log file normally produced by the MVS operating system. In particular, the JES Job Purge (Type 26) records were used for the model.[2] The JES Job Purge record contains the following items that are required for the simulation model:

- Job Name,
- Priority,
- Job Class,
- Reader Start Time and Date, T_1,

1. A technique for establishing resource-limited job class structures has been presented in a companion paper previously published by the author. [1]

2. Although the data contained in the type 26 record is sufficient for modeling an installation's current job classes, additional information must also be collected by the analyst if a new set of job classes is to be developed. These additional variables are described in papers previously published by the author. [1,5]

CH1437-3/79/0047-0051$00.75

INITIATOR SIMULATION MODEL

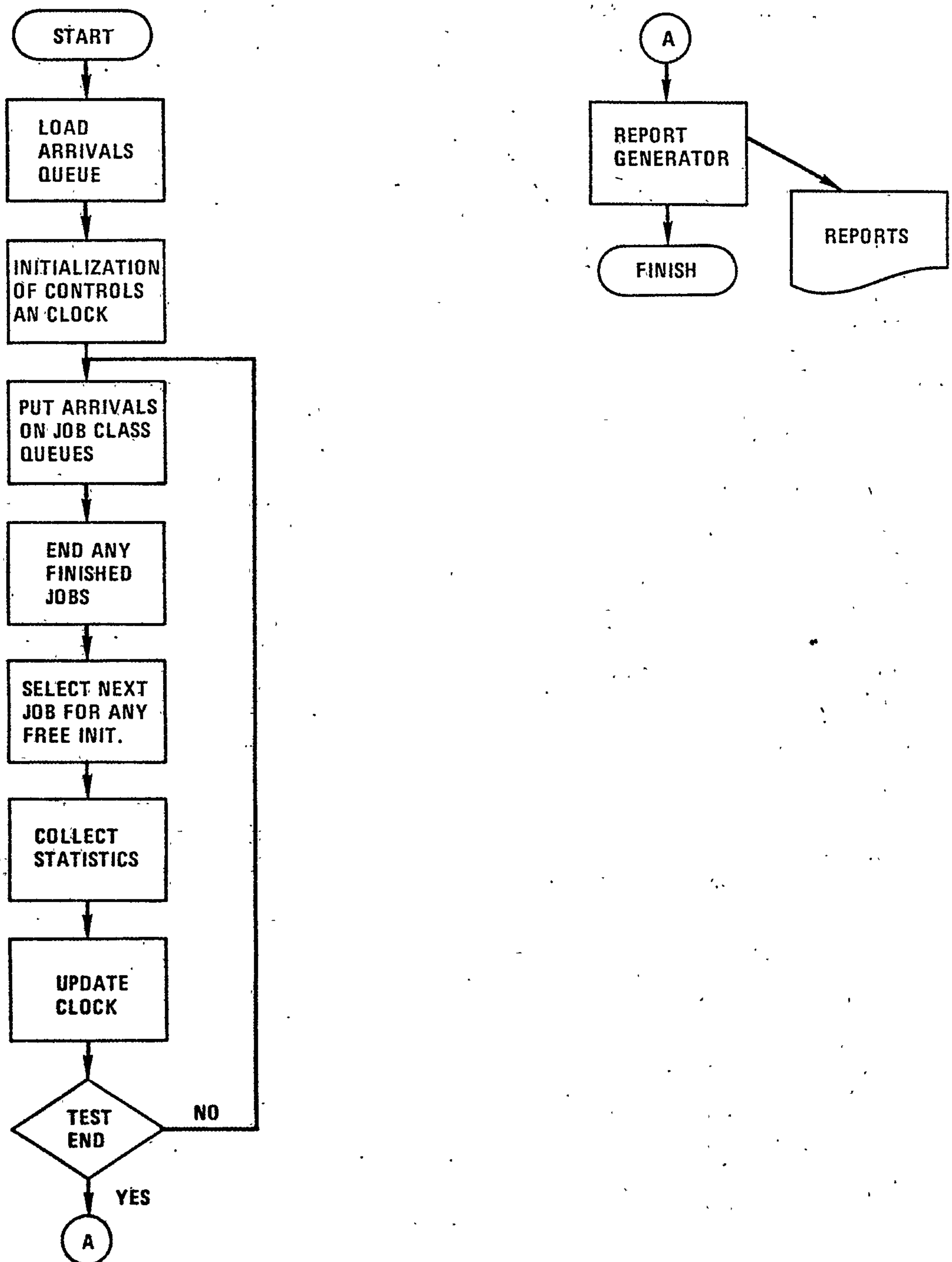

FIGURE I

- Execution Start Time and Date, T_2,

- Execution End Time and Date, T_3.

When this data is sorted in ascending order by reader time, it provides a stream of arrivals for driving a simulation model. A job's actual queuing delay, T_2-T_1, and server holding time, T_3-T_2, may be calculated from these values. As previously stated, the simulation model assumes that a job's server holding time remains unchanged, and the model determines the queuing delay under the new initiator structure.

The reader should note that there are a number of influences on initiator and job activity that are not recorded in the SMF log file. These influences are introduced by operator interactions with the system. For instance, the operator may:

- modify the priority of a job to change its position in the input queue,

- put a job into a "hold" status to prevent or delay its execution,

- modify the initiator structure.

Generally, these operator interactions are minimal in a well managed system. For any given system, an analyst may determine the exact extent of these activities by investigating the console log where operator actions are recorded.

3. MODEL DESIGN

The discrete simulation model was implemented in PL/1. A flowchart of the model is shown in Figure 1. The principal modules in the model are:

LOAD ARRIVALS — The module reads the trace file produced by the data collection program.

INITIALIZATION — This module loads the user input control data that specifies the number of initiators and the job classes that each serves. It also sets the clock and statistics collection arrays to their initial values.

ARRIVALS — This module simulates arrivals by queuing all jobs that arrive at the current clock value and placing them on the input queues that are maintained by job class and priority.

SELECTION — This module attempts to schedule a job for each free initiator. It selects the highest priority job in the first non-empty job queue which is served by the initiator. When a job is selected, an end event for the initiator is scheduled based on the job's actual execution time which was determined by the data collection program.

STATISTICS — This module collects statistics on initiator utilization, job queue lengths and simulated input queue delay time.

UPDATE CLOCK — This module updates the simulation clock. A one second clock step is used for this simulation.

REPORTS — This module summarizes and reports on the statistics collected by the program.

Typical run times for the simulation are from twenty to sixty CPU seconds on a 370/168 for an eight hour study period.

4. VALIDATION

A series of tests were conducted using synthetic jobs on an IBM 370/168 to validate the model. The tests were conducted in a stand-alone environment to avoid any influences that might be introduced by other jobs or operator activities.

The synthetic jobs used in the study were quite simple in design. The program executed by the jobs issued a "wait" for a period of time specified by the input control to the program. Thus, an exact server holding time could be specified for each job.

For the validation study, three classes of jobs were used. They were:

- CLASS A, jobs with a holding time of four minutes,

- CLASS B, jobs with a holding time of two minutes,

- CLASS C, jobs with a holding time of one minute.

A total of three tests were conducted. During each test, a total of twelve groups of jobs were submitted to the system at five minute intervals. Each group contained one Class A job, two Class B jobs and seven Class C jobs. SMF data was collected for the entire testing period and was reduced using the data collection program described in a previous section.

Since both sources of potential error (operator activities and job interactions) were carefully excluded from the test environment, the simulated and actual results were exactly the same for each of the three experiments. In the each test, three initiators were started in the system. In the first test, the first initiator served Class A jobs, the second served Class B jobs and the third served Class C jobs. The total time between the arrival of the first group of jobs and the completion of the final job was 84 minutes. The results of the experiment are shown in the following table:

Test I	Actual	Simulated
Length of Study (min)	84	84
Initiator 1 Busy (%)	57.14	57.14
Class A Jobs Processed	12	12
Initiator 2 Busy (%)	57.14	57.14
Class B Jobs Processed	24	24
Initiator 3 Busy (%)	100.00	100.00
Class C Jobs Processed	84	84
Avg Wait Class A (min)	0.00	0.00
Standard Deviation	0.00	0.00
Avg Wait Class B (min)	1.00	1.00
Standard Deviation	1.02	1.02
Avg Wait Class C (min)	14.00	14.00
Standard Deviation	7.23	7.23

As shown in the table, the Class C jobs were poorly served even though initiator three was always busy.

For the second test, initiator one was changed to serve both Class A and C jobs. The results of this experiment are shown in the following table:

Test II	Actual	Simulated
Length of Study (min)	66	66
Initiator 1 Busy (%)	100.00	100.00
Class A Jobs Processed	12	12
Class C Jobs Processed	18	18
Initiator 2 Busy (%)	72.73	72.73
Class B Jobs Processed	24	24
Initiator 3 Busy (%)	100.00	100.00
Class C Jobs Processed	66	66
Avg Wait Class A (min)	0.00	0.00
Standard Deviation	0.00	0.00
Avg Wait Class B (min)	1.00	1.00
Standard Deviation	1.02	1.02
Avg Wait Class C (min)	7.00	7.00
Standard Deviation	2.95	2.95

The results of this experiment indicate the improved service provided to the Class C jobs by the modification of the first initiator. However, the second initiator was still not completely utilized.

In the final test, the second intitator was changed to serve both Class B and C jobs. The results of the this experiment are shown in the following table:

Test III	Actual	Simulated
Length of Study (min)	60	60
Initiator 1 busy (%)	100.00	100.00
Class A Jobs Processed	12	12
Class C Jobs Processed	12	12
Initiator 2 Busy (%)	100.00	100.00
Class B Jobs Processed	24	24
Class C Jobs Processed	12	12
Initiator 3 Busy (%)	100.00	100.00
Class C Jobs Processed	60	60
Avg Wait Class A (min)	0.00	0.00
Standard Deviation	0.00	0.00
Avg Wait Class B (min)	1.00	1.00
Standard Deviation	1.02	1.02
Avg Wait Class C (min)	2.57	2.57
Standard Deviation	1.50	1.50

As shown in the table, the service to the Class C jobs was once again improved by the modification to initiator two. Moreover, the minimum possible run time for a three initiator experiment, sixty minutes, was achieved since all of the servers were 100% utilized.

The three experiments discussed above show that the model is capable of identically simulating the input queue delay times and the initiator utilizations in environments where the model's assumptions are ideally met. It has been the author's experience, that in actual use the model is usually about five percent off due to operator activities and unexpected job interactions. Moreover, these differences are almost always accountable to operator activities rather than significant changes in job execution times due to job interactions.

5. APPLICATIONS

During the last eighteen months various versions of the simulation model have been used in approximately fifteen studies. The experience gained in these studies has led to a number of enhancements of the model and has provided a basis for confidence in the model's results. These studies can be divided into three principal groups. They are:

- very short studies, usually conducted with only one or two days of data, to evaluate the potential effects of a minor modification to an existing initiator structure. The primary objective of these studies is not to evaluate the change under a variety of workloads but to provide a "check" that the proposed change will not severely degrade the system's current service level,
- detailed analysis of a proposed major revision of an initiator structure using a system's current job classes. These studies usually require two to four weeks of historical data so that the proposed initiator structure can be evaluated for a wide variety of loads,
- studies in which new job classes and initiator structures are developed simultaneously. These studies usually require four to six weeks of data and use techniques for developing new job classes that are described in the references [1,5]. These studies are the most complex since changes to the job class structure redefine the classes, interarrival and average server holding times for the simulation model. Typically, these studies have taken one to two person-weeks to complete.

To date, these studies have never resulted in an initiator structure that failed to provide approximately the service levels predicted by the model. Although the predictions are subject to the previously discussed limitations, they provide a significant comparative advantage to the trial and error evaluation of new initiator structures in a live environment.

ACKNOWLEDGEMENTS

The author would like to acknowledge the contributions of Mr. Andrew Kalicak of Illinois Bell Telephone who initially suggested the development of the model. He was also intrumental in the testing and validation of the model.

REFERENCES

[1] Artis, H.P., A Technique for Establishing Resource Limited Job Class Structures, CMG X Proceedings, Dec. 1979.

[2] Buzen, J.P., A Queuing Network Model of MVS, ACM Computing Surveys, Vol. 10, No. 3, Sept. 1978.

[3] Roode, J. D., The Operational Analysis Approach to Computer System Modeling, Symposium on Computer Resource Performance Management, Pretoria, South Africa, April 1979.

[4] Systems Management Facility, IBM, July 1977, EC28-0706.

[5] Artis, H.P., Capacity Planning for MVS Computer Systems, published in Performance of Computer Installations, ICPCI 78, D. Ferrari Ed., North Holland Publishing, 1978.