SIMULATION AIDED EVALUATION OF REMOTE BATCH EQUIPMENT

T. W. Dudley and B. E. Herring
Auburn University, Auburn, Alabama

1. INTRODUCTION

The purpose of the procedure described in this paper was the development of an instrument to permit in-depth evaluation of non-interactive (batch) pieces of computer equipment used at remote locations. A prime concern in the evaluation process was the ability to predict the effect of utilizing alternative items of remote equipment, i.e., faster or slower input/output devices as well as to be able to determine the impact of supporting a variable number of remote sites.

Auburn University's Computer Center (AUCC) was the subject for the evaluation process. An expansion of capability to support remote batch equipment has been a relatively recent addition at Auburn. Figure 1 depicts AUCC's equipment configuration.

Input/output equipment consisted of one highspeed reader and two printers located at the site of the IBM 370/155 CPU and four remote stations each having a lower speed reader and printer with half-duplex operation, i.e., the equipment cannot read and print simultaneously. The usual complement of tape drives, disk drives, and a console completed the primary set of equipment.

Effectiveness evaluation of such a configuration of equipment was a difficult task. When the unique demands of academic users were added, direct analytic evaluation became impossibly complex.

Techniques for measuring and evaluating equipment and software come under the broad topic of computer measurement and evaluation (CME). Lucas (1) cited three purposes for evaluation of a computer system: selection evaluation, performance projection, and performance monitoring. Bell (2) discussed six reasons for computer performance analysis: general control, hypothesis generation, hypothesis testing, equipment change, sizing, and system design. Bell also discussed five methods or tools for performance evaluation and their applicability to each of his six reasons for performance evaluation. The approaches are: personal inspection, accounting systems, monitors (simulation and analytic models). Bell felt that simulation techniques were well suited to evaluating equipment changes.

Lucas identified eight evaluation techniques: (1) comparison based on memory access time, CPU cycle time, and add time; (2) comparison based on instruction mixes; (3) kernel programs; (4) benchmarks; (5) synthetic programs; (6) analytic models; (7) simulation models; and (8) performance monitoring. Kimbleton (3) stated that analytic models and simulation models provide a means for considering various modifications to the system.

Simulation techniques are a well accepted method for testing changes in hardware or equipment because once a good model is designed, pieces of equipment can be added or subtracted within the simulation model with very little effort. According to Maguire (4), simulation is an effective method for testing proposed changes to computer systems. He felt that simulation, compared with other methods of analysis, is more realistic, more easily understood, and more conclusive. The results are easier to understand, therefore the conclusions receive wider acceptance.

2. RELATED EFFORTS

Much effort has been devoted to the development of various techniques of evaluating computer performance and equipment. These range from very simple methods to elaborate and detailed procedures. They have been developed for many different purposes and many different systems and machines. Review of reports of this work points to a trait common to all. In order to develop a workable model, one should avoid the temptation to include every conceivable element. References 5 through 11 underscore this point.

3. MODEL STRUCTURE

In keeping with the notion of simplification, Figure 2 illustrates the equipment configuration included in the model. The primary concern of the evaluation procedure was to determine the effectiveness of various configurations of remote equipment. For that reason, tape and disk drives and their operations were omitted from the model. The card punch and one of the centrally located printers were also omitted.

Conceptual simplifications were made to the computer system to enable it to be more easily modeled. These simplifications, or constraints were:

a. Disk and tape usage was not considered since these were not relevant to the operation of the remote batch equipment.

b. The true multiprogramming environment was not simulated. Instead, up to five jobs were allowed to be executing at the same time, provided core was available, and each executed for a given execution time. Therefore execution time in this model did not depend on other jobs in the machine at the same time.

c. Jobs were allowed to run if the amount of core available was greater than or equal to the amount of core necessary to run a job of that class. However, core was not kept in segments and the problem of core fragmentation was not considered.

d. When an initiator was freed and more core was available, the job with the highest priority was executed. If there was not enough core available for the job with the highest priority, all jobs in the queue waited until either there was enough core or a job with a higher priority came along.

e. In the actual system jobs that require greater than 240k or fifteen minutes of execution time were automatically put into a hold status and had to be released by an operator to be run. This model did not place any jobs in hold.

While the imposition of such constraints could significantly revise the situation being studied, analysis of model validation efforts yielded no serious deviations from the actual system.

4. DETAILS OF THE MODEL

The major elements of the model have been organized in flow diagram form and are contained in Figure 3. The General Purpose Simulation System language (GPSS) was the vehicle chosen to actually build and run the model. The empirical distribution functions used to describe the stochastic processes being modeled were constructed from SMF (system management facility) data collected during a one month period. Distributions which were necessary included:

a. Job interarrival times for each input location.

b. The various job classes submitted at each input location.

c. Execution times by job classes.

d. The quantity of cards per job submitted at each input location.

e. The quantity of lines printed per job submitted at each input location.

The various blocks of the flow diagram (Figure 3) represent the following:

a. Job Arrival - GPSS generate blocks coupled with the appropriate interarrival functions provide a flow of jobs into the system model.

b. Parameter Assignment - transaction parameters are given values which provide each job a custom identity consisting of its input location, job class, quantity of cards to be read, execution time requirement, number of lines to be printed, and output location.

c. Job Input - once the job attributes have been determined and assigned, the incoming job is placed in a queue to await reader availability. Reader processing time is computed from the deck size (quantity of cards) attribute determined earlier.

d. Execution Priority Assignment - execution priorities are selected from an inverse ranking of core and execution time attributes, i.e., jobs with small core and time requirements receive a high execution priority. AUCC, being in an academic environment, processes a large number of jobs having very small core and time needs. These are FORTRAN jobs which use the Waterloo FORTRAN compiler (WATFIV). Such jobs are read as submitted, but are batched for processing in a special class several times a day.

e. Job Execution - after all job attributes and priority assignments are made, the job is placed into a queue to await the availability of a job initiator (five maximum) and sufficient core. Jobs which are in a special class; very small needs or very large needs, execute only on a periodic basis. Very small jobs (WATFIV) are handled by a monitor which enters several times a day while the large jobs are started only when no other jobs are waiting. Preemptive priority rules are not a part of the model. While actual job execution times vary depending on in-core job mix, the model ignores mix variability and retains (advances) the job for the predetermined time period.

f. Job Output - Following job execution, printer wait queues are formed. Queue positions are based on the previously determined number of output lines - the smaller the amount, the earlier in the line. In addition, a routing exercise may be necessary to return the finished job to a different print location, i.e., a faster printer or default to the printer at the location of job origin.

5. MODEL VALIDATION

Validation of a model is important if the model is to be useful as a basis for predicting the effect of changes in a workload mix or alternative equipment assignments. Visual comparison of actual versus simulated system operation yielded acceptable results (see Figures 4 and 5). Statistical analysis (chi-square test) was not as good. Distributions of most of the interarrival times and the elapsed execution time yielded by the model were statistically found to not be from a distribution different from the actual data. However, the simulated distributions for turnaround time, reader time, printer time, job input location and job class mix did not yield good statistical results.

While statistically significant results were not always obtained, the results were felt to be successful. Sample sizes were large (nearly 25,000 jobs), thus causing deviation from the actual distribution to be significant according to the chi-square test. Additionally, in most cases, class frequencies deviated only two percent or less (the worst case was only five and a half percent). With these minor deviations, the model was judged to be useful for the original purposes.

6. USE OF THE MODEL

System parameters related to a different type of remote reader/printer unit were introduced into the model for trial at the more heavily used remote stations. The alternative equipment, as compared with that used in the original model, utilized a slower card reader, a faster printer, and a buffer which permitted near simultaneous reader/printer operation. While mean turnaround time was not significantly reduced, the variance of turnaround time as tested by the F-ratio was definitely decreased.

According to Weinberg (12), the decrease in the variance of turnaround time can be as desirable to the user as a reduction in mean turnaround time. It was felt the major reason no significant reduction was made in the mean turnaround time was the lack of a heavy workload at any of the remote stations.

REFERENCES

1. Lucus, J. C., Jr., Performance Evaluation and Monitoring, Computing Surveys 3, 3(Sept. 1971), pp 79-91.

2. Bell, T. E., Performance Determination - The Selection of Tools, if any, Proc. AFIPS 1973 National Computer Conference, pp. 31-38.

3. Kimbleton, S. R., The Role of Computer Models in Performance Evaluation, Comm ACM 15, 7(July, 1972), pp. 586-590.

4. Maguire, J. N., Discrete Computer Simulation Technology and Applications - The Next Ten Years, Proc. AFIPS 1972 Spring Joint Computer Conference, pp. 815-826.

5. MacDougall, M. H., Computer System Simulation: An Introduction, Computing Surveys 2, 3(Sept. 1970), pp. 191-209.

6. Noe, J. D. and Nutt, G. J., Validation of a Trace-Driven CDC 6400 Simulation, Proc. AFIPS 1972 Spring Joint Computer Conference, pp. 749-757.

7. Bowden, E. K., Mamrak, S. A., and Salz, F. R., Simulation - A Tool for Performance Evaluation in Network Computers, Proc. AFIPS 1973 National Computer Conference, pp. 121-131.

8. Katz, J. H., An Experimental Model of System/360, Comm. ACM 10, 11(Nov. 1967), pp. 694-702.

9. Katonak, P. R., Performance Analysis of the Multiprocessor/65 Through Simulation, Computer Measurement and Evaluation, Vol. 1, Oct. 1967 - Oct. 1971, pp. 265-281.

10. Bokenbaum, M., Is Simulation Any Good Without Measurement (Validation and Verification of a GPSS Model Which Uses SMF Data for Local Description), Computer Measurement and Evaluation, Vol. 2, Nov. 1971 - Dec. 1973, pp. 375-380.

11. Kernodle, J. F., Jr., A Simulation Model of a Batch-Processing Job Stream, Master's Thesis, University Of North Carolina at Chapel Hill, 1973.

12. Weinberg, G. M., The Psychology of Computer Programming, New York, N. Y.: Van Nostrand Reinhold Company, 1971.

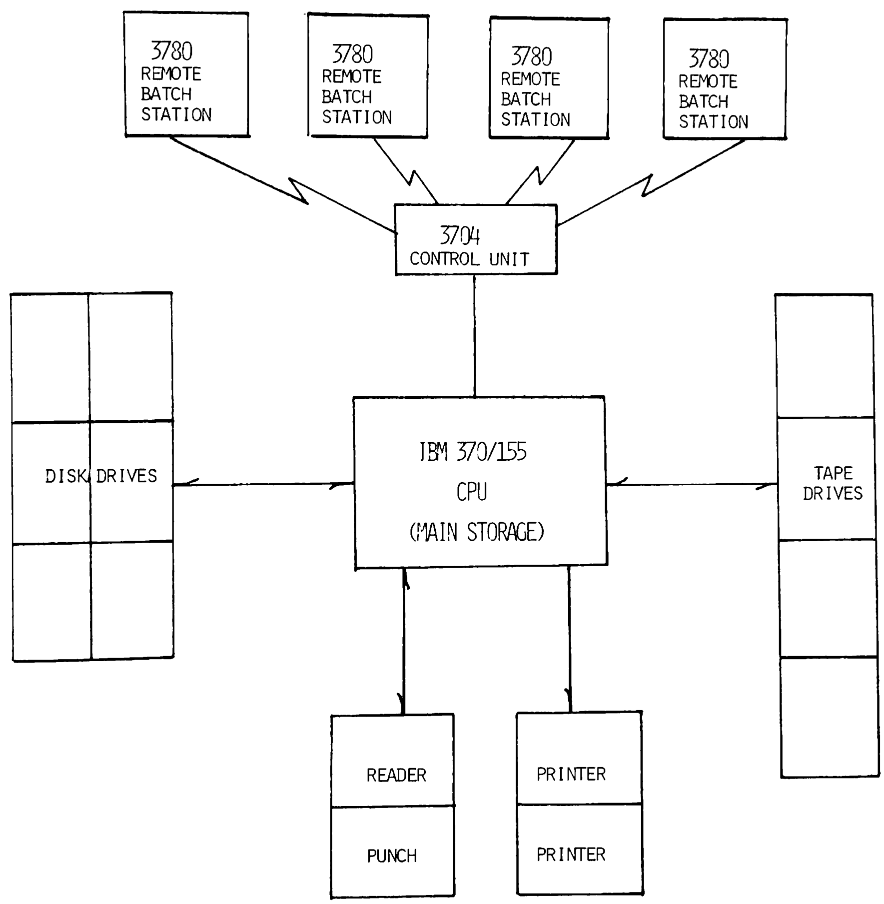

Figure 1. Auburn University Computer Center Equipment

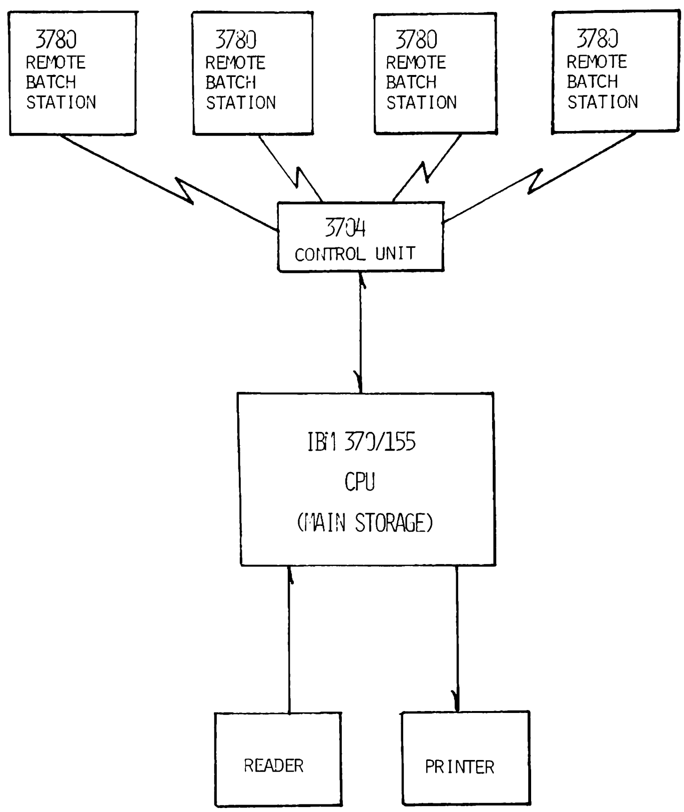

Figure 2. Equipment Considered Within the Model.

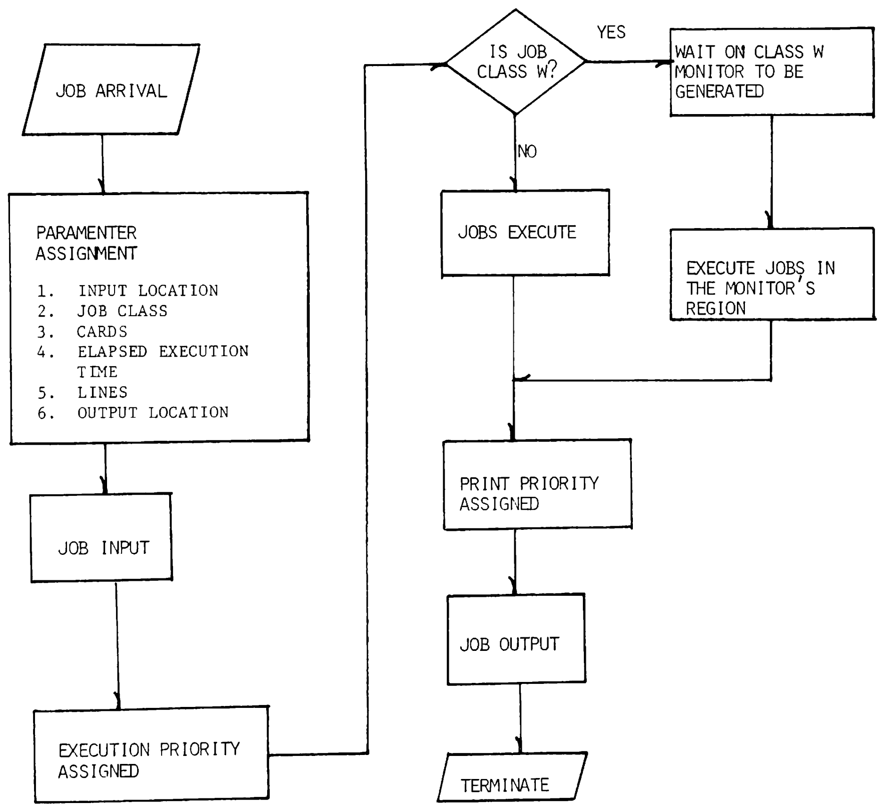

Figure 3. Flow Diagram of the Model.

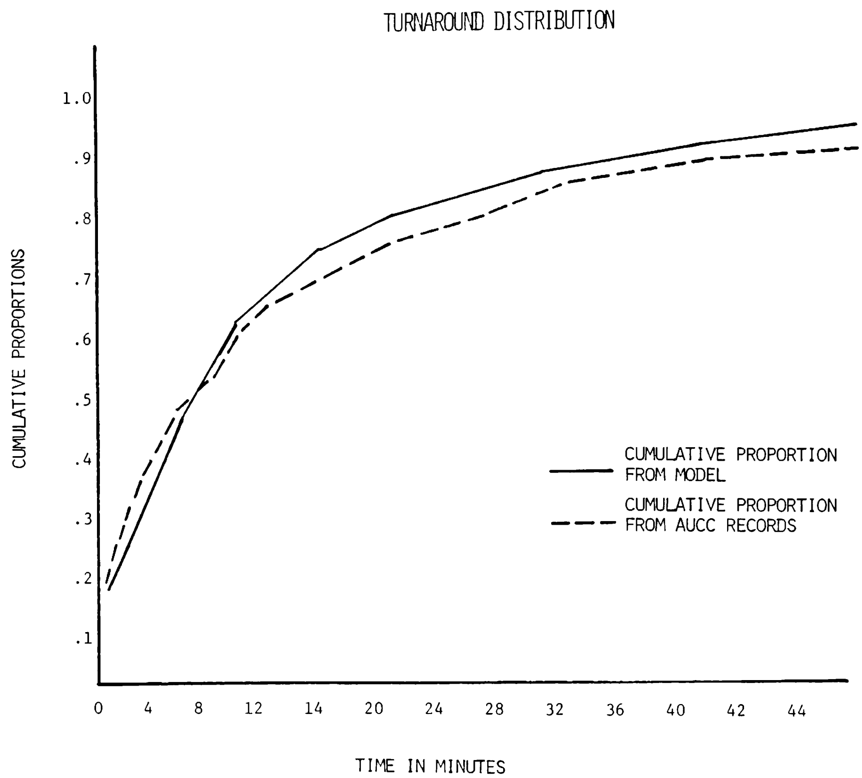

Figure 4. Cumulative Distributions of Job Turnaround Times.

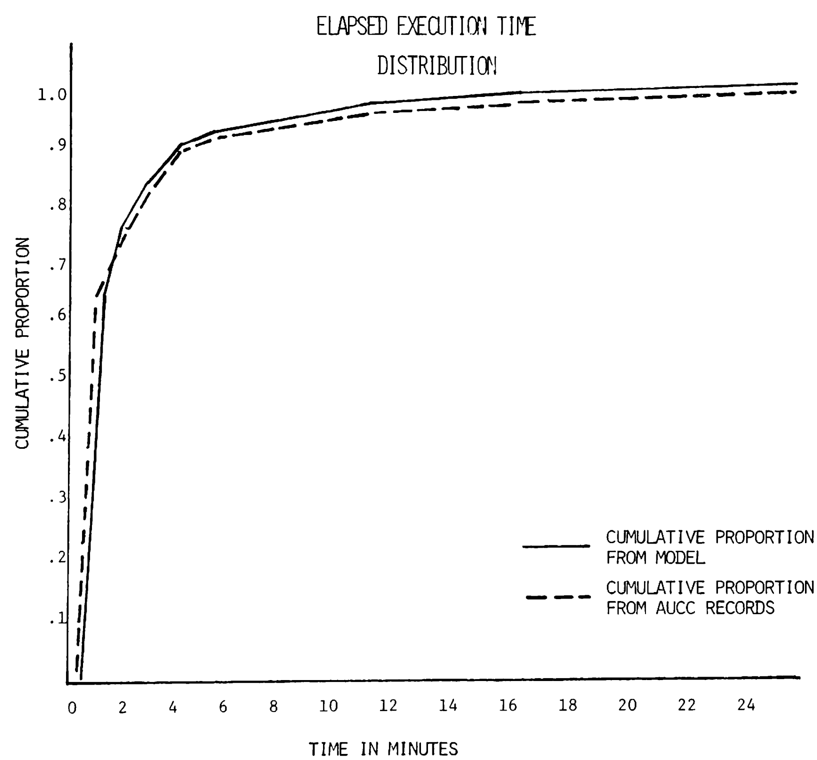

Figure 5. Cumulative Distributions of Job Execution Times.