# NAVAL POSTGRADUATE SCHOOL

## Monterey, California

ANALYSIS OF COMPUTER PERFORMANCE

IN MULTIPROGRAMMED PROCESSING

by

Norman F. Schneidewind

May 1975



Prepared for: Naval Weapons Center,
China Lake, California

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Isham Linder
Superintendent

Jack R. Borsting
Provost

The work reported herin was supported by the Naval Weapons Center, China Lake, California.



This report was prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>NPS55Ss75051 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)*<br>Analysis of Computer Performance in Multiprogrammed Processing | | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical Report<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR*(s)*<br>Norman F. Schneidewind | | 8. CONTRACT OR GRANT NUMBER*(s)* |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Naval Postgraduate School<br>Monterey, California 93940 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>N60530 72PO-2-0056 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS | | 12. REPORT DATE<br>May 1975<br>13. NUMBER OF PAGES |
| 14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)* | | 15. SECURITY CLASS. *(of this report)*<br>Unclassified<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT *(of this Report)*<br>Approved for public release; distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)* | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)* | | |

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

An analysis was made of the correlation between performance and resource usage variables of a given computer job and between the performance variables of a given job and the resource usage variables of other jobs in a multiprogramming environment. This analysis was performed in order to: (1) determine the mix and characteristics of jobs which lead to high performance, (2) provide regression equation predictors of performance for given resource utilizations, and (3) provide performance and resource

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

utilization coefficients for use in a linear programming resource allocation model. The linear programming model is used to select an optimum job mix subject to production, resource usage and budgetary constraints.

TABLE OF CONTENTS

# INTRODUCTION

## Objective

The basic objective of computer center management is to provide high computer system performance at a reasonable cost under conditions of fluctuating workload and fixed computer resources. In order to satisfy this objective, it is necessary to forecast the performance and resource utilization which would result from a permanent and significant change in workload, if resources remain unchanged. If projected performance is unsatisfactory or the anticipated resource utilization is low, the forecast provides a warning that resources must be expanded or contracted, respectively. Once the condition of saturation or under utilization has been anticipated, it is necessary to forecast the performance and resource utilization which would be obtained when resources are changed. In addition to providing the above forecasts, the requirement to achieve high performance, when constrained by computer capacity, budget, schedule and workload restraints, suggests the need for a model of computer performance and resource allocation. Such a model would provide a tool for producing some of the information which computer center management requires for achieving high computer performance. Thus, this paper involves two major analytical efforts: (1) the development of performance and resource usage forecasting equations, and (2) the development of a model for analyzing computer performance and resource allocation.

Scientific computer center management has little control over the variables which determine computer center performance. Unlike management data processing, the scientific center deals with external users who program a variety of problems in a variety of languages and with varying degrees of

skills. In addition, inputs cannot be scheduled or sequenced to any significant degree in order to take advantage of the processing characteristics of various jobs. Usually, computer management can only influence performance by using a charging algorithm which discourages large jobs during the prime shift. Tuning of the operating system is done reluctantly, if at all, because some computer vendors will not support a modified system or will refuse to provide the information needed for making operating system changes. Management may be able to make hardware changes or replace the computer system when the workload threatens to saturate the system.

A model for system optimization would have to be compatible with the above limitations. In view of the peculiarities of the scientific computing environment, the modelling approach which has been developed seeks to use charging algorithms and system modifications as the primary system improvement alternatives and to employ changes to the operating system scheduler and dispatcher as a possible secondary remedy. The key control variable which has been identified as a means of attempting system optimization is the job mix. An explanation of the use of this variable appears in the next section.

## Job Mix

One variable which has considerable influence over computer performance is job mix. This refers to the composition of a group of jobs, by type of application, programming language or use of resources, which is input to a computer during a specified interval of time. Job mix is of no interest to the individual user; his measure of performance is turnaround or response time. However, job mix is of interest to computer center management because it must view the user workload and allocation of computer resources in total. A good job mix, from the standpoint of computer center management, has jobs

which are compatible, i.e. jobs which do not compete to a great extent for the use of the same resources at the same time and, hence, are non-interfering jobs. Equivalently, jobs are compatible if the execution time of each does not depend on the resource usage of the other jobs.

It should be noted that a good job mix, from the standpoint of computer center management, may be antithetical to the needs of some users. Therefore, when job mix is used as a performance factor, we should also consider its effect on user turnaround time. User job production constraints could be used in the model as a means of satisfying this requirement. If compatible and incompatible job mixes could be identified, this information could be used to implement one or more of the following performance improvement techniques:

- Adjust the job charging algorithm so that incompatible jobs will be charged more than compatible jobs during peak computer usage and at the regular rate during non-peak periods. This selective charging technique can be applied by job type, such as FORTRAN compilations, or on the basis of resource usage, independent of job type. An advantage of the first method is that the customer would possess definitive information about computer costs in advance of job execution because the charge would be based on type of job. A disadvantage is that, because customer relations must be considered, it may be infeasible for computer center management to selectively charge in such a way that all jobs in specified categories are penalized for prime shift usage, regardless of resource usage. An advantage of the second method is that it is more equitable because pricing is based only on resource usage. However, the user may not be aware of resource usage prior to job execution. The resource usage would be known, once the job is executed; however, in scientific computing, many jobs are executed only once.

• Eliminate or reduce job mix as a major performance consideration by expanding or changing the hardware configuration so that performance is not critically dependent on job mix. Of course, the implementation of this alternative may involve a considerable expenditure for hardware.

• Modify the operating system so that compatible jobs will have a higher priority for job initiation and execution than incompatible jobs. It should be noted that, whereas the job charging approach attempts to influence job mix prior to the input of jobs to the computer center, this approach attempts to change the input job mix to a compatible mix, after jobs are read into the input queue. Whereas the use of a different charging algorithm may produce a permanent change in job mix, a change in the operating system will only affect job mix during short time intervals, since low priority jobs cannot be queued indefinitely. In addition, if the input queue becomes too long, jobs must be temporarily transferred to auxiliary storage and later returned to main memory, resulting in an increase in processing overhead. Job priority determination for scheduling and dispatching purposes may be made by job type or resource usage. An advantage of the former method is that priority determination is unambiguous because job type will be known to the operating system. A disadvantage is that the priority assignment is not selective, i.e. entire jobs are assigned a single priority, regardless of resource usage. Assignment of job priority by resource usage would be superior to assignment by job type, if resource usage were known or could be estimated accurately in advance of execution. However, this information is not known for a high percentage of jobs until after the first execution. Once the job has been executed, the user can request resources based on the resource usage of the last execution. However, these requests are often so conservative that there may be a big

difference between requested and actual resource usage, resulting in an inefficient allocation of resources by the operating system.

It is natural to speculate on ways to identify the ideal job mix, where ideal may be defined as a job mix which will optimize some performance variable, such as job elapsed time, subject to user and computer management specified constraints. If an ideal mix can be identified, existing performance can be evaluated with respect to the ideal. The ideal mix could be used as the goal for future performance achievement, and the ideal mix could be used as a measure of the usefulness of alternative performance improvement approaches. The next section describes the scope of this paper with respect to modelling and quantifying the relationships which have been discussed.

## SCOPE

The scope of this paper is limited to the following three items:

1. Construct a model for computing the ideal job mix. The ideal job mix would be compared with the actual job mix in order to determine which changes in computer service charging policy, operating system usage or hardware configuration may be desirable. Only the mathematical formulation of the model is presented in this paper. The numerical solution of the model for a particular computer center operation is outside the scope of this paper. In order to obtain a numerical solution of the model for an operation with many types of jobs, it is necessary to estimate the values of many coefficients and parameters. Work is in progress to complete the estimation of these factors for the Naval Weapons Center (NWC), China Lake, California, Computer Center UNIVAC 1108.* The complete model numerical solution pertaining to this installation

---

*The analysis of data in this report pertains to a UNIVAC 1108 which was installed at NWC during the period of data collection. This installation was subsequently upgraded to a UNIVAC 1110.

will be the subject of a subsequent paper. Also to be included in this future paper will be a detailed analysis of the problems of implementing a performance improvement in terms of a charging algorithm, operating system modification or hardware change, should such an improvement appear necessary after obtaining a numerical solution to the model.

2. Analyze the statistical correlations among performance and resource usage variables in order to estimate the degree of association among these variables. This information is useful for identifying relationships which can be used in computer center resource management and for the development of computer center performance and resource usage estimating equations.

3. Formulate regression equations for forecasting computer center performance and for estimating performance coefficients which are used in the model. These equations would provide the capability of estimating the effect on performance of changes in resource usage, job mix or workload. Regression equations are also required for estimating resource usage and resource coefficients which are used in the model. The formulation of the resource usage equations was beyond the scope of the paper.

Although each of the above items involves a separate analysis, the items are related because the regression equations are used to provide estimates of model coefficients and the correlation coefficients are used to develop the regression analysis.

Items 2 and 3 involved the analysis of system log data collected from the NWC. The sample data used in the analysis is summarized in Appendix I.

## COMPUTER CENTER RESOURCE ALLOCATION MODEL

### Operating Environment

The type of operating environment for which this model is applicable is batch and terminal processing with multiprogramming. Using the terminology of NWC, jobs executed from a terminal will be called demand jobs. The types of programs which are executed can be divided roughly into three classes: compilations, execution of compiled programs (production) and the use of a variety of utility programs. Each of these categories is executed in both batch and demand modes and the two modes are used on the same shift.

### Job Mix

Job mix as a controllable performance variable has been discussed in previous sections. On what basis should the ideal mix be chosen? One approach is to attempt to find that mix which will simultaneously satisfy job production, computer center budget, resource capacity and utilization constraints and minimize total elapsed time over all jobs. Job mix during time periods of equal duration (say one hour) can be represented by the number of jobs of program type $j$

$$x_1,\ldots,x_j,\ldots,x_n \tag{1}$$

which are executed during the specified time period. Alternately, the mix can be represented by

$$x_1/X_T,\ldots,x_j/X_T,\ldots,x_n/X_T \tag{2}$$

where $X_T$ is the total number of jobs executed in the designated time period. For our purposes (1) will be the number of jobs executed of a given program type (FORTRAN compilations) per hour.

Performance Variables

The achievement of user performance objectives should result in the maximization of user satisfaction or utility, overall, taking into consideration various user and computer center constraints. A model which employs the concept of relative value to a user as a function of system response time and utilization is described in [1]. Ideally, measures of user utility or value should be used in a performance model. However, this type of data is very difficult to collect. The reference does not explain how this data was or could be obtained. Lacking a direct measure of the value to the user of computer system performance, a surrogate variable, elapsed time, will be utilized. It will be assumed that value to the user is inversely proportional to elapsed time. The minimization of this variable over all jobs, is equivalent to maximizing value to the users as a whole. For batch and demand jobs elapsed time is defined as the interval of time between the initiation and termination of CPU activity on a job. Elapsed time involves all CPU and I/O file activity and wait times between the time that the CPU starts and terminates job execution. The elapsed time excludes the following: time required to read the job from the card reader or terminal, time the job resides in internal storage prior to the start of execution by the CPU and time required to print job output. Job output is spooled to peripheral devices during execution and printing is performed as a separate task. For purposes of this model and the analyses which appear in subsequent sections of this paper, elapsed time per program type $j$, $T_j$, will be the total of all elapsed time for program type $j$ during a one hour period. A primary reason for using this particular definition is that this is the form in which data is recorded by the system logging function at the Naval Weapons Center Computer Center. This system captures total elapsed time and resource

usage data by program type for each hour of operation but does not record these data by individual job. However, the number of jobs executed per hour by program type, $x_j$, is recorded so that the mean elapsed time per job, $t_j$, may be computed. With respect to program type $j$, elapsed time

$$T_j = C_j + W_j \tag{3}$$

where $C_j$ is CPU active time and $W_j$ is CPU inactive or wait time. Wait time

$$W_j = W_{j1} + W_{j2} + W_{j3} \tag{4}$$

where $W_{j1}$ is the total wait time per hour incurred by program type $j$ while waiting for its own I/O operation to complete and no CPU processing is occurring; $W_{j2}$ is the total wait time per hour incurred by program type $j$ while waiting for the CPU to become available; and $W_{j3}$ is the total wait time per hour incurred by program type $j$ while waiting for an I/O device to become available and no CPU processing is occurring. It should be noted that, in general, there is an overlap of CPU and I/O operations; $W_{j1}$ is the amount of I/O time of program type $j$ which is not overlapped.

With respect to the components of elapsed time $T_j$, the following functional relationships exist:

- $C_j$ is determined by the CPU requirements of program type $j$ which are determined in part by $x_j$ and job core requirements.
- $W_{j1}$ is determined by the I/O requirements of program type $j$ which are determined in part by $x_j$, $C_j$ and job core requirements. Job mix has no effect on $C_j$ and $W_{j1}$.
- $W_{j2}$ is determined by the CPU requirements of other program types.
- $W_{j3}$ is determined by the I/O requirements of other program types.

Job mix does have an effect on $W_{j2}$ and $W_{j3}$. Thus job mix can be used as a control variable to control the $W_{j2} + W_{j3}$ component of elapsed time. The NWC data recording includes $T_j$ and $C_j$, from which $W_j$ can be computed; however, $W_{j1}$, $W_{j2}$, and $W_{j3}$ are not individually recorded, so that control of $W_{j2} + W_{j3}$ has to be exercised indirectly via $W_j$.

Other candidates for performance variable are CPU time and total wait time. Neither of these variables alone accounts for both CPU activity and the delays resulting from resource usage conflicts created by the presence of multiple jobs or the delays caused by non-overlapped CPU and I/O activities.

Optimal Job Mix

The optimal job mix is defined to be that mix which will result in the minimization of elapsed time over all jobs executed in a specified time period. If performance and resource usage vary considerably over a number of time periods, it would be necessary to identify an optimum job mix for each time period. However, this consideration is not a critical issue with respect to the approach used to develop the model. It is a secondary issue concerned with estimating model coefficients for different time periods and employing them in the appropriate time period solution. If the set

$$x_1^*, \ldots, x_j^*, \ldots, x_n^* \tag{5}$$

constitutes the optimal mix, given in units of jobs per hour for program type $j$, then the ratio

$$M = \sum_j \hat{t}_j x_j \Big/ \sum_j t_j x_j \tag{6}$$

is a figure of merit, where $\hat{t}_j$ is the estimated mean elapsed time in seconds per job for program type $j$, and $t_j$ and $x_j$ are the actual

elapsed times and number of jobs, respectively. The closer $M$ is to one, when various constraints are considered, the better the performance. The goal of various performance improvement approaches, such as charging algorithms, operating system modification and hardware changes, would be evaluated with respect to their ability to achieve the optimal job mix given by (5). As mentioned previously, there may be a different optimal mix for each time period. It should be noted that it may be possible to have more than one optimal job mix. In addition, total elapsed time may be insensitive to job mix. It will be possible to analyze these factors, once numerical solutions to the mathematical program are available.

## MATHEMATICAL PROGRAM

### Objective Function

The use of mathematical programming for computer equipment selection and computer center resource allocation has been demonstrated in [2]. In order to estimate the optimal job mix, a mathematical program is formulated based on the minimization of $T_j$ and the satisfaction of constraints involving the $x_j$ and resource usage $A_{ij}$, where $i$ and $j$ are the resource and program type, respectively. For the purpose of generality of notation, the form $A_{ij}$ will be used to designate all resource usages, including CPU time $C_j$. Recalling (3) and (4) and the statements which followed concerning the functional relationships involving $C_j$ and $W_j$, we can write

$$T_j = f_j(x_1,\ldots,x_j,\ldots,x_n;A_{11},\ldots,A_{i1},\ldots,A_{m1};$$

$$A_{1j},\ldots,A_{ij},\ldots,A_{mj};A_{1n},\ldots,A_{in},\ldots,A_{mn}) \qquad (7)$$

The $A_{ij}$ are a function of the use of other resources $A_{kj}$ and the $x_j$. An example is the number of I/O references per hour as a function of CPU time per hour (a measure of program duration), core usage per hour (a measure of program size) and number of jobs per hour (a measure of resource usage). Thus,

$$A_{ij} = g_j(A_{1j},\ldots,A_{kj},\ldots,A_{mj};x_j), \quad \text{where} \quad k \neq j. \tag{8}$$

Certain $A_{ij}$, such as core usage, are a function of $x_j$ directly. In other cases the $A_{ij}$ can be made a function of $x_j$ indirectly via the relationship between $A_{kj}$ and $x_j$. An example is that CPU time per hour $(A_{ij})$ is a function of core usage per hour $(A_{kj})$. Both are a function of number of jobs per hour $(x_j)$, so that CPU time can be related to jobs per hour alone. Thus (8) becomes

$$A_{ij} = g_j(x_j). \tag{9}$$

Although the degree of association between dependent and independent variables may be higher in (8) than in (9), the use of (9) is required because the $A_{kj}$ are not really independent variables; their values depend upon the nature of the computer operation to be performed. In addition, the $A_{kj}$ are not known in advance of a computer run; they must be estimated from the $x_j$. In contrast, the $x_j$ are determined external to the computer system by the users. Their values and the associated job mix represent the demands placed on the computer center. In view of (9) equation (7) reduces to

$$T_j = f_j(x_1,\ldots,x_j,\ldots,x_n). \tag{10}$$

A discussion of the degree of linear association among the variables in (9) and (10) and the appropriateness of a linear model is presented in the correlation and regression sections of this paper. For the present we will be interested in (9) and (10) for the purpose of formulating a mathematical programming model.

Since we wish to minimize the elapsed time per hour over all program types, the objective function in the mathematical programming formulation becomes

$$\text{Minimize } Z = \sum_j T_j = \sum_j f_j(x_1,\ldots,x_j,\ldots,x_n) \tag{11}$$

Constraints

1. Resource Usage

Total resource usage must not exceed the available resource in a specified time period. Thus, using (9)

$$\sum_j A_{ij} = \sum_j g_j(x_j) \le b_i, \qquad i = 1,\ldots,m$$

where $b_i$ is the capacity of the $i^{\underline{th}}$ resource, such as the number of CPU seconds available per hour. Since in certain instances, it may be infeasible or of no interest to account for all types of programs which are executed at a computer center, a load factor $\ell_i$, representing the fraction of the total capacity $b_i$ which is used by the $n$ program types (excluding operating system load), is applied as follows:

$$\sum_j A_{ij} = \sum_j g_j(x_j) \le \ell_i b_i, \qquad i = 1,\ldots,m. \tag{12}$$

2. Production

User requirements in terms of number of jobs of program type $j$ executed per hour, $N_j$, must be satisfied

$$x_j \ge N_j, \qquad j = 1,\ldots,n \quad \text{and} \quad N_j \ge 0. \tag{13}$$

The production constraints may be viewed as an indirect way of specifying priority by program type in the model.

3. The computer center should receive revenue per hour at least equal to the total budget available per hour, $B$, to operate the computer center. Thus

$$\sum_j \sum_i a_{ij} p_i x_j \ge B \tag{14}$$

where $a_{ij}$ is the units of resource $i$ per job used by program type $j$ and $p_i$ is the price per unit of resource $i$ which is charged to the users.

4. Utilization

The computer center may be interested in maintaining a minimum CPU utilization $U_j$ for each program type. If this is the case, a constraint would be involved of the form $\frac{A_{1j}}{T_j} \geq U_j$ or $U_j T_j - A_{1j} \leq 0$, where $A_{1j}$ is the CPU resource usage. Since $A_{1j}$ and $T_j$ can be expressed in terms of $x_j$, we have

$$U_j f_j(x_j) - h_j(x_j) \leq 0, \quad j = 1,\ldots,n. \tag{15}$$

Thus, (11) through (15) constitute a mathematical program, where the solution variables are the $x_j$ and we wish to find the optimum job mix $x_1^*,\ldots,x_j^*,\ldots,x_n^*$. This formulation does not depend upon the existence of linear functions in (11) through (15), although the feasibility of obtaining numerical solutions depends upon the existence of linear or separable (non-linear but no product of $x_j$ terms) functions. Assuming for the present that a linear function can be used to estimate $T_j$ with sufficient accuracy, (10) can be written as

$$\hat{T}_j = \hat{v}_{j0} + \hat{v}_{j1}x_1 + \hat{v}_{j2}x_2 + \ldots + \hat{v}_{jk}x_k + \ldots + \hat{v}_{jn}x_n \tag{16}$$

where the $\hat{v}_{jk}$ are estimated regression coefficients and $1 \leq j \leq n$. When (16) is divided by $x_j$, we obtain $\hat{t}_j$, the estimated elapsed time per job for program type $j$. Using (16) in the objective function (11), we obtain

$$\text{Minimize } Z = \sum_j \sum_k \hat{v}_{jk}x_k. \tag{17}$$

Once the $x_j^*$ are determined, they can be used with the $\hat{t}_j$ to obtain the figure of merit of (6).

Similarly, if we assume that (9) can be estimated with sufficient accuracy by

$$\hat{A}_{ij} = \hat{\alpha}_{ij} + \hat{\beta}_{ij} x_j, \tag{18}$$

where $\hat{\alpha}_{ij}$ and $\hat{\beta}_{ij}$ are estimated regression coefficients, (12) can be expressed by

$$\sum_j \hat{\beta}_{ij} x_j \leq \ell_i b_i - \sum_j \hat{\alpha}_{ij}, \qquad i = 1,\ldots,m. \tag{19}$$

If (18) is divided by $x_j$, we obtain $\hat{a}_{ij}$, the estimated resource usage per job for program type $j$. The $\hat{a}_{ij}$ are used as estimates of $a_{ij}$ in (14).

Applying (16) and (18), (15) becomes

$$\hat{U}_j \sum_k \hat{v}_{jk} x_k - \hat{\beta}_{1j} x_j \leq \hat{\alpha}_{1j} - \hat{U}_j \hat{v}_{j0}, \qquad j = 1,\ldots,n, \tag{20}$$

where $\hat{U}_j$ is the sample mean utilization for program type $j$ and the $\hat{\alpha}_{1j}$ and $\hat{\beta}_{1j}$ constitute the estimated regression coefficients of (18) for CPU usage.

Thus, (13), (14) with $\hat{a}_{ij}$ substituted for $a_{ij}$, (17), (19) and (20) constitute a linear program.

Fortunately, it is not necessary to obtain an integer solution for the $x_j$ because, as actually measured, the number of jobs per hour of program type $j$ is, in general, a non-integer value, because some jobs are in process at the termination of each one hour measurement period. A fraction of each job in process is recorded against the period just concluded and the remaining fraction is recorded against the next period.

After obtaining the optimal solution (5), the quantities listed below could be computed. These calculations would indicate how close the actual computer center operation is to the optimal operation. Of course, lack of optimality would only exist in the context of the mathematical definition of the linear program. Computer center management may wish to use different sets of constraints and definitions of optimality.

(1) Difference Between Total Actual and Total Optimal Elapsed Time

$$\sum_j T_j - \sum_j (\hat{v}_{j0} + \sum_k \hat{v}_{jk} x_k) \tag{21}$$

(2) Difference Between Actual and Optimal Production

$$x_j - x_j^*, \qquad j = 1,\ldots,n \tag{22}$$

(3) Difference Between Actual and Optimal Revenue

$$\sum_j \sum_i A_{ij} p_i - \sum_j \sum_i \hat{a}_{ij} p_i x_j^* \tag{23}$$

(4) Difference Between Actual and Optimal Resource Usage

$$\sum_j A_{ij} - \sum_j (\hat{\alpha}_{ij} + \hat{\beta}_{ij} x_j^*), \qquad i = 1,\ldots,m \tag{24}$$

(5) Difference Between Actual and Optimal CPU Utilization

$$U_j - \hat{A}_{ij}/\hat{T}_j = U_j - (\hat{\alpha}_{1j} + \hat{\beta}_{1j} x_j^*)/(\hat{v}_{j0} + \sum_k \hat{v}_{jk} x_k^*), \qquad j = 1,\ldots,n \tag{25}$$

If a significant difference between actual and optimal operations exists, a sensitivity analysis could be performed with respect to the parameters: computer capacities ($b_i$), user production requirements ($N_j$), budget (B), charge rates ($p_i$), and utilizations ($U_j$), in order to ascertain the effect of parameter changes on total elapsed time and job mix.

Solutions Obtained by Distinguishing Among Hourly Operating Periods

In order to achieve sufficient accuracy in the estimates of linear program coefficients, as obtained from regression equations, it may be necessary to make the estimates for each hour of the daily operation. This corresponds to using twenty-four sets of observations, with each set corresponding

to all observations in the given one hour period over all the days which comprise the sample. This is in contrast to the above formulation which makes no distinction of observations according to hour of the day.

The linear program would be changed by modifying (17), (13), (14), (19), (20) and (5), as follows, respectively

$$\text{Minimize } Z = \sum_h \sum_j \sum_k \hat{v}_{jkh} x_{kh}, \tag{26}$$

$$x_{jh} \geq N_{jh}, \qquad j = 1,\ldots,n; \quad h = 1,\ldots,24 \quad \text{and} \quad N_{jh} \geq 0, \tag{27}$$

$$\sum_h \sum_j \sum_i \hat{a}_{ijh} p_{ih} x_{jh} \geq 24B, \tag{28}$$

$$\sum_h \sum_j \hat{\beta}_{ijh} x_{jh} \leq 24\, \ell_i b_i - \sum_h \sum_j \hat{\alpha}_{ijh}, \qquad i = 1,\ldots,m, \tag{29}$$

$$\hat{U}_{jh} \sum_k \hat{v}_{jkh} x_{kh} - \hat{\beta}_{1jh} \leq \hat{\alpha}_{1jh} - \hat{U}_{jh} \hat{v}_{jOh}, \qquad j = 1,\ldots,n; \quad h = 1,\ldots,24. \tag{30}$$

where $h$ refers to the hour of the day. The optimal solution becomes

$$x^*_{11},\ldots,x^*_{1h},\ldots,x^*_{124};\ x^*_{j1},\ldots,x^*_{jh},\ldots,x^*_{j24};\ x^*_{n1},\ldots,x^*_{nh},\ldots,x^*_{n24}. \tag{31}$$

This approach would obviously require significantly more computation and definition of parameters but would provide greater solution accuracy and allow a distinction to be made, by hour, with respect to production requirements $(N_{jh})$ and resource prices $(p_{ih})$.

## CORRELATION ANALYSIS

The degree of linear association between performance and resource usage variables and among resource usage variables can be measured by simple and multiple correlation coefficients. Although no cause and effect relationships can be attributed to the existence of high correlation coefficients, the measures do have several uses. One use is to indicate which variables should be related in regression equations for the purpose of using linear functions to forecast performance or resource usage. Secondly, since this paper is concerned with multiprogrammed systems, it is of interest to identify job mixes which contain types of programs with highly correlated variables. This information identifies the job mixes for which linear functions can be used to forecast performance and resource usage, when the mix is executed in a multiprogrammed system. Similarly, job mixes which contain variables with low correlations cannot be analyzed with linear functions. Finally, the correlation coefficients indicate whether the performance of a given program type can be forecasted with linear functions by using only its variables, or whether it is necessary to use other program variables. This determination is made by comparing correlation coefficients within program types to correlation coefficients between program types. The existence of job mixes with high correlations or low correlations suggests but does not prove the existence of non-compatible and combatible mixes, respectively.

The correlation analysis is related to the mathematical program of the previous section because it provides a basis for developing the regression equations which are used to obtain estimates of the mathematical program coefficients. The regression analysis is described in the next section. Only relationships which are linear in the independent variables are discussed in

this section. The applicability of polynomial functions is considered in the regression analysis section. Initially, it was of interest to examine the applicability of linear functions because: (1) if this approach proved adequate, the functions could be used in the linear program, (2) linear functions may suffice as an adequate approximation to the true performance and resource usage, and (3) greater forcasting accuracy may be attainable by using many independent variables in a linear regression equation as compared to using fewer variables in a polynomical regression equation. Functions which are non-linear in the parameters were not used because a significant number of data values were zero, since all program types are not executed during each time period. The absence of these values must be recorded as zeros when performing regression analysis with respect to the many programs which constitute the job mix in a multiprogramming mode. In order to perform a logarithmic transformation of the variables, when zero values are present, it would be necessary to use the form log (x+a), where "a" is a constant. This is not a problem when program types are analyzed individually because it is not necessary to account for missing values. The focus of the investigation was on the development of a multiple variable model and the correlation and regression analysis of multiple program types in order to treat the variables as they exist in the actual multiprogramming environment. However, if it can be shown that only a limited number of variables is required to adequately forecast performance and resource usage, the complexity and computational effort involved in solving the mathematical program and regression equations can be reduced.

A description of the sample which was used in the correlation and regression analysis appears in Appendix I. A definition and identification of variables appears in Appendix II. A brief description of the NWC computer

configuration and the types of programs which were used in the analysis appear in Appendix IV.

The data used in the correlation and regression analysis were produced by the system logging function at the NWC Computer Center.

Sample correlation coefficients were computed by using the computer program "Correlation with Transgeneration, BMD02D" [9]. Prior to the use of the correlation routines, scatter plots were obtained for identifying the high correlation variables.

The results which will be discussed are only a sample of the entire analysis which was performed. Correlation coefficients were obtained for six types of programs, both batch and demand. The data shown in the tables is intended to illustrate the type of analysis which has been performed and to indicate the applications of the data. The data shown is typical of the pattern of sample correlation coefficients.

Correlation of Variables Within a Program Type

The data presented in Tables 1 and 2 are typical of the results obtained for the twelve program types which were analyzed. That is, sample correlations are high except for the correlations between elapsed time and other variables, where the values are considered acceptable but not outstanding. This anomaly is attributed to the high percentage of wait time which is part of elapsed time (refer to Tables 13 and 14). As mentioned previously, wait time is a function of several variables. Some of these variables are associated with other program activities (wait for another program's CPU or I/O activities to complete). Thus, we would not expect to find a high correlation coefficient when only the resource usages of the given program type are related to elapsed time. None of the relationships between performance and resource usage or between resource usages should necessarily be linear.

Table 1

Sample Correlation Coefficient Matrix - FORTRAN (Batch Programs)

N = 108 hours

| FORTRAN | Jobs | CPU Time | Core | I/O Refs | I/O Words | Elapsed Time |
|---|---|---|---|---|---|---|
| Jobs | 1.0 | .962 | 1.0 | .991 | .996 | .709 |
| CPU Time | .962 | 1.0 | .962 | .977 | .970 | .707 |
| Core | 1.0 | .962 | 1.0 | .992 | .996 | .709 |
| I/O Refs | .991 | .977 | .992 | 1.0 | .997 | .716 |
| I/O Words | .996 | .970 | .996 | .997 | 1.0 | .710 |
| Elapsed Time | .709 | .707 | .709 | .716 | .710 | 1.0 |

Table 2

Sample Correlation Coefficient Matrix - FORTRAN (Demand Programs)

N = 108 hours

| FORTRAN | Jobs | CPU Time | Core | I/O Refs | I/O Words | Elapsed Time |
|---|---|---|---|---|---|---|
| Jobs | 1.0 | .816 | 1.0 | .955 | .979 | .732 |
| CPU Time | .816 | 1.0 | .817 | .941 | .908 | .650 |
| Core | 1.0 | .817 | 1.0 | .955 | .979 | .730 |
| I/O Refs | .955 | .941 | .955 | 1.0 | .993 | .713 |
| I/O Words | .979 | .908 | .979 | .993 | 1.0 | .745 |
| Elapsed Time | .732 | .650 | .730 | .713 | .745 | 1.0 |

For example, CPU time per hour should be positively correlated with the amount of core used per hour and number of jobs executed per hour.* However, the jobs of a given program type do not use equal amounts of CPU time, nor do programs of equal core size use equal amounts of CPU time. Thus, all correlation coefficients are intended only to provide an indication of the feasibility of using a linear function as an approximation to the true function.

In contrast to elapsed time, resource usage, as measured by CPU time, core usage, I/O references and I/O words transferred, is a function only of the variables of the given program type. Thus data of the type depicted in Table 1 and Table 2 are the only data necessary for indicating the degree of linear association among resource usage variables. For elapsed time, the correlations of elapsed with other program variables must also be examined. This topic is discussed in the next section.

Correlation of Variables Across Program Types

The data in Table 3 and Table 5 suggest that there is a greater degree of linear association between the elapsed time of a given program type (FORTRAN) and its variables than there is between elapsed time and other program variables. This is true for both batch (Table 3) and demand (Table 5) programs. In addition, there is greater correlation between batch elapsed time and batch program variables than between batch elapsed time and demand program variables (Table 3). Similarly, there is greater correlation between demand elapsed time and demand program variables than between demand elapsed time and batch program variables (Table 5). Thus, the primary (high correlation) variables for an elapsed time linear regression equation would come from the given program type.

---

*It should be noted that certain of the program types analyzed, including FORTRAN compilations, use a constant amount of core storage (See Appendix I).

Table 3

Sample Correlation Coefficients of FORTRAN (Batch) Elapsed Time with Resource Usage of Various Programs

N = 108 hours

| | | Resource Usage | | | |
|---|---|---|---|---|---|
| Batch Programs | Jobs | CPU Time | Core | I/O Refs | I/O Words |
| FORTRAN | .709 | .707 | .709 | .716 | .710 |
| Other | .519 | .358 | .454 | -.141 | -.114 |
| FURPUR | .368 | .259 | .365 | .175 | .153 |
| MAP | .632 | .570 | .622 | .601 | .619 |
| PMD | .069 | .132 | .083 | .092 | .116 |
| COBOL | .227 | .111 | .226 | .076 | .086 |
| Demand Programs | | | | | |
| FORTRAN | .400 | .379 | .399 | .402 | .411 |
| Other | .449 | .402 | .438 | .339 | .337 |
| FURPUR | .486 | .456 | .486 | .430 | .430 |
| MAP | .425 | .404 | .424 | .399 | .423 |
| PMD | .177 | .064 | .170 | .059 | .063 |
| COBOL | .212 | .273 | .212 | .258 | .261 |

Table 4

Sample Correlation Coefficients of FORTRAN (Batch) Wait Time with Resource Usage of Various Programs

N = 108 hours

| | | Resource Usage | | | |
|---|---|---|---|---|---|
| Batch Programs | Jobs | CPU Time | Core | I/O Refs | I/O Words |
| FORTRAN | .653 | .646 | .653 | .659 | .653 |
| Other | .475 | .346 | .417 | -.141 | -.110 |
| FURPUR | .339 | .239 | .337 | .158 | .137 |
| MAP | .586 | .523 | .576 | .556 | .573 |
| PMD | .048 | .103 | .060 | .068 | .091 |
| COBOL | .209 | .103 | .208 | .069 | .078 |
| Demand Program | | | | | |
| FORTRAN | .385 | .364 | .384 | .386 | .395 |

Table 5

Sample Correlation Coefficients of FORTRAN (Demand) Elapsed Time with Resource Usage of Various Programs

N = 108 hours

| | | Resource Usage | | | |
|---|---|---|---|---|---|
| Batch Programs | Jobs | CPU Time | Core | I/O Refs | I/O Words |
| FORTRAN | .218 | .232 | .218 | .224 | .222 |
| Other | .355 | .214 | .273 | -.212 | -.111 |
| FURPUR | .229 | .183 | .229 | .106 | .226 |
| MAP | .405 | .304 | .403 | .323 | .259 |
| PMD | -.018 | .179 | -.017 | .175 | .167 |
| COBOL | .278 | .136 | .276 | .074 | .092 |
| Demand Programs | | | | | |
| FORTRAN | .732 | .650 | .730 | .713 | .745 |
| Other | .614 | .623 | .562 | .553 | .487 |
| FURPUR | .595 | .507 | .596 | .510 | .583 |
| MAP | .606 | .579 | .606 | .565 | .580 |
| PMD | .041 | .043 | .048 | .037 | .040 |
| COBOL | .404 | .373 | .403 | .378 | .388 |

Table 6

Sample Correlation Coefficients of FORTRAN (Demand) Wait Time with Resource Usage of Various Programs

N = 108 hours

| | | Resource Usage | | | |
|---|---|---|---|---|---|
| Batch Programs | Jobs | CPU Time | Core | I/O Refs | I/O Words |
| FORTRAN | .211 | .225 | .211 | .217 | .214 |
| Other | .349 | .213 | .267 | -.209 | -.109 |
| FURPUR | .226 | .183 | .227 | .107 | .228 |
| MAP | .398 | .297 | .396 | .316 | .351 |
| PMD | -.017 | .176 | -.017 | .175 | .166 |
| COBOL | .275 | .134 | .273 | .072 | .090 |
| Demand Program | | | | | |
| FORTRAN | .721 | .632 | .720 | .699 | .732 |

However, in order to obtain sufficient regression equation forecasting accuracy, it may be necessary to include secondary (lower correlation) variables from other program types. The effects of the secondary variables are an indication of the impact of job mix and conflicting resource usage on performance.

Similar results were obtained with respect to the use of wait time as a performance variable, as shown in Table 4 for batch FORTRAN compilation and in Table 6 for demand FORTRAN compilation. The indication is that elapsed time will provide a more accurate predictor of performance than wait time when using a linear regression function. This is due to the fact that wait time varies in a complex manner because of the influence of other program activities under multiprogramming, whereas elapsed time contains CPU time which will later be seen to vary more linearly than wait time with resource usage.

An examination of Tables 3, 4, 5 and 6 indicates that in most cases, for FORTRAN compilations, there appears to be no appreciable difference in correlations by resource usage for the batch and demand program categories. This examination involves comparing the correlation coefficients across columns within the batch and demand groups. Also, there appears to be no significant difference among resource usage correlations per given program type (examine the data within rows). As a consequence of the absence of difference in correlation by variable, we are led to consider the use of the given program's number of jobs as the primary variable for estimating elapsed time and the use of other programs' number of jobs as the secondary variables. The primary variable is related to the CPU time and wait time attributed to the execution of the given program and the secondary variables are related to the wait time attributed to the execution of other programs. Other reasons for using number of jobs to forecast elapsed time are:

(1) it is not necessary to estimate this variable from other variables, and

(2) these are the solution variables $(x_j)$ of the linear program.

Certain correlations, which may appear to be significant, require careful interpretation. An example is the correlation between FORTRAN compilation elapsed time and MAP CPU time (Table 3). This value (.570) may appear to be related to the delay in FORTRAN compilation caused by MAP use of the CPU, whereas the more likely explanation is the correlation between the two independent variables (.795 between batch FORTRAN number of jobs and batch MAP number of jobs). This coefficient does not appear in Table 3. Thus, a large amount of FORTRAN elapsed time occurs during the same hourly interval as a large amount of MAP CPU time because a large number of FORTRAN and MAP jobs are executed during the same period. A possible method for removing the effect of number of jobs would be to normalize the variables to be correlated by dividing the values by number of jobs. However, this is infeasible because many number of job values are zero. If values are restricted to non-zero quantities, the intersection of the non-zero values of two or more program types is a sample size which is too small for meaningful statistical analysis.

Correlation of Elapsed Time with Number of Jobs

For reasons which have been given previously it is highly desirable to use only number of jobs as the independent variable in the regression equations. The data in Table 7 provide one indication of the feasibility of this approach. The following characteristics emerge:

Table 7

Sample Correlation Coefficient Matrix - Elapsed Time vs. Number of Jobs

N = 108 hours

| | Batch Jobs | | | | | | Demand Jobs | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FORTRAN | OTHER | FURPUR | MAP | PMD | COBOL | FORTRAN | OTHER | FURPUR | MAP | PMD | COBOL |
| Batch Elapsed Time | | | | | | | | | | | | |
| FORTRAN | .709 | .519 | .268 | H.632 | L.069 | .227 | .400 | .339 | .486 | .425 | .177 | .212 |
| Other | .354 | .504 | .377 | H.381 | .273 | .177 | .364 | .335 | .333 | .312 | .313 | L.145 |
| FURPUR | .372 | H.569 | .553 | .555 | .432 | .377 | .253 | .291 | .258 | .238 | L.045 | .237 |
| MAP | .540 | H.679 | .511 | .743 | .206 | .562 | .452 | .482 | .483 | .455 | L.121 | .402 |
| PMD | .156 | H.239 | .156 | .130 | .401 | .192 | .149 | .298 | .176 | .243 | L.002 | .346 |
| COBOL | .082 | .176 | .177 | H.226 | L.033 | .528 | .170 | .194 | .184 | .120 | -.039 | .189 |
| Demand Elapsed Time | | | | | | | | | | | | |
| FORTRAN | .218 | .355 | .229 | .405 | L.018 | .278 | .731 | H.614 | .595 | .606 | .041 | .404 |
| Other | .441 | .447 | .263 | .460 | L.068 | .207 | .709 | .850 | H.816 | .800 | .313 | .626 |
| FURPUR | .399 | .369 | .191 | .474 | L.007 | .214 | H.714 | .710 | .827 | .706 | .151 | .559 |
| MAP | .272 | .330 | .228 | .373 | L.017 | .160 | .556 | .647 | H.738 | .748 | .078 | .633 |
| PMD | .064 | .027 | -.070 | L.004 | -.083 | -.087 | .141 | .135 | .276 | H.284 | .221 | .140 |
| COBOL | .202 | .345 | .196 | .374 | L.015 | .256 | .542 | .603 | .629 | H.700 | .181 | .787 |

H Signifies highest correlation other than with own jobs.

L Signifies lowest correlation other than with own jobs.

- Correlations of elapsed time with the given program's number of jobs are generally higher than with other programs' number of jobs.
- Batch programs have higher correlations with batch number of jobs than with demand number of jobs. Similarly, demand programs have higher correlations with demand number of jobs than with batch number of jobs.
- As was mentioned in the previous section, the given program will be the source of the primary variable in an elapsed time regression equation and the other programs will be the source of the secondary variables.

Correlation of Elapsed Time with CPU Time

A comparison of Tables 7 and 8 reveals that in many instances the correlation of elapsed time with CPU time is higher than with number of jobs. For this reason, it may be possible to achieve greater accuracy in the forecasting of elapsed time for some program types by using CPU time as the "independent" variable. Since CPU time is one of the variables to be forecasted, its value must be estimated from number of jobs or from this variable and the amount of core storage used. The correlations of CPU time with these variables are shown in Table 9.

Correlation of Resource Usage Variables

A. CPU Time

The data in Tables 9 through 12 are presented in order to show the relationships among variables which are required in order to forecast four resource usage variables: CPU time, core storage usage, number of I/O references and number of I/O words transferred. In the discussion which follows all references to variables are with respect to a given program type. It is hypothesized that CPU time per hour is primarily a function of two variables: number of jobs per hour and amount of core used per hour. The former is a measure of the frequency of use of the CPU and the latter is a measure of the duration of CPU use as indicated by program size. As mentioned previously,

Table 8

Sample Correlation Coefficients of Program Elapsed Time with that Program's CPU Time

N = 108 hours

| | Batch Programs | Demand Programs |
|---|---|---|
| FORTRAN | .707 | .650 |
| Other | .714 | .849 |
| FURPUR | .738 | .707 |
| MAP | .808 | .785 |
| PMD | .718 | .576 |
| COBOL | .803 | .809 |

Table 9

Sample Correlation Coefficients of Program CPU Time with that Program's Number of Jobs and Core Usage

N = 108 hours

| | Batch Programs | | Demand Programs | |
|---|---|---|---|---|
| | Jobs | Core | Jobs | Core |
| FORTRAN | .962 | .962 | .816 | .817 |
| Other | .519 | .459 | .832 | .829 |
| FURPUR | .711 | .714 | .753 | .758 |
| MAP | .905 | .916 | .963 | .963 |
| PMD | .423* | .472 | .227* | .242 |
| COBOL | .753 | .755 | .886 | .887 |

*Small number of jobs.

Table 10

Sample Correlation Coefficients of Program Core Usage with that Program's Number of Jobs*

| | Batch Programs | Demand Programs |
|---|---|---|
| Other | .937 | .949 |

*All other programs use approximately a constant amount of storage per job. Therefore, correlation coefficients $\approx 1$.

this is not a linear function. However, a linear function appears to be a satisfactory approximation to the true function. For certain program types, such as compilations, the amount of core storage is a constant. In this case the known values of core storage and number of jobs can be used directly in the regression equation. In the case of production programs, the core storage usage is a variable and must be estimated from number of jobs. The correlations of CPU time with number of jobs and core storage usage are shown in Table 9.

B. Core Storage Usage

The correlations of production program core storage usage (OTHER) with number of jobs are shown in Table 10. Since there is a great variety of production jobs at NWC with a variety of core storage requirements, the high linearity between core storage usage and number of jobs is surprising. This is possibly explained by the allocation of core storage to a program in fixed size blocks, in which the number of region sizes is limited. This procedure reduces the core storage size variability.

C. I/O References

It is hypothesized that number of I/O references per hour is primarily a function of CPU time per hour and core storage used per hour. The former is the time during which it is possible to initiate an I/O reference. The latter is a measure of frequency of occurrence of I/O instructions as indicated by program size. Again, this function would not be strictly linear because I/O references do not occur in strict proportion to program duration and I/O commands do not appear in programs in strict proportion to program size. The correlations between I/O references and CPU time are shown in Table 11. Correlations between I/O references and core storage usage were not consistently high.

Table 11

Sample Correlation Coefficients of Program I/O References with that Program's CPU Time

N = 108 hours

| | Batch Programs | Demand Programs |
|---|---|---|
| FORTRAN | .977 | .941 |
| Other | .186 | .531 |
| FURPUR | .745 | .805 |
| MAP | .959 | .997 |
| PMD | .921 | .987 |
| COBOL | .969 | .994 |

Table 12

Sample Correlation Coefficients of Program I/O Words Transferred with that Program's CPU Time

N = 108 hours

| | Batch Programs | Demand Programs |
|---|---|---|
| FORTRAN | .970 | .908 |
| Other | -.086 | .510 |
| FURPUR | .753 | .848 |
| MAP | .966 | .991 |
| PMD | .943 | .980 |
| COBOL | .969 | .993 |

The CPU time which would be used in a regression equation for forecasting I/O references would be estimated from number of jobs and amount of core storage required.

The production program (OTHER) correlations in Table 11 are low. A relatively low correlation coefficient for production batch jobs also appears in Table 9. These low values appear to be caused by the heterogeneous characteristics of production jobs as contrasted to the more uniform characteristics of a compiler or utility program.

D. I/O Words Transferred

Since words are transferred each time an I/O reference is made, the two variables would be positively correlated. In fact, the sample correlation coefficients are high except for the OTHER category. These coefficients are not shown in the tables. Since I/O references would be forecasted by using CPU time, as indicated above, the correlations between CPU time and I/O words transferred are shown in Table 12.

Distribution of Resource Usage

Tables 13 and 14 show mean values of CPU time, wait time, elapsed time and CPU utilization on a per hour and job basis, respectively. These tables demonstrate that the "other" category (production programs) dominate computer usage. This is unfortunate because it was not possible to obtain high correlation coefficients for this category due to the heterogeneity of production programs.

Table 13

N = 108 hours

Mean Values

| Batch | CPU Time Sec/hr | Wait Time Sec/hr | Elapsed Time Sec/hr | CPU Utilization |
|---|---|---|---|---|
| FORTRAN | 34.11 | 266.68 | 300.79 | .113 |
| Other | 1,203.45 | 5,825.80 | 7,029.25 | .171 |
| FURPUR | 6.12 | 461.79 | 467.91 | .013 |
| MAP | 27.06 | 213.84 | 240.90 | .112 |
| PMD | 3.80 | 5.69 | 9.49 | .400 |
| COBOL | 36.09 | 148.53 | 184.62 | .195 |
| | 1,310.63 | 6,922.33 | 8,232.96 | .159 |
| Demand | | | | |
| FORTRAN | 10.58 | 343.70 | 354.28 | .030 |
| Other | 127.83 | 3,196.36 | 3,324.19 | .038 |
| FURPUR | 5.82 | 1,608.04 | 1,613.86 | .004 |
| MAP | 14.27 | 220.70 | 234.96 | .061 |
| PMD | .84 | 5.44 | 6.28 | .134 |
| COBOL | 26.35 | 176.72 | 203.07 | .130 |
| | 185.69 | 5,550.96 | 5,736.65 | .032 |
| | 1,496.32 | 12,473.29 | 13,969.61 | .120 |

Table 14

N = 108 hours

Mean Values

| Batch | CPU Time Per Job (sec) | Wait Time Per Job (sec) | Elapsed Time Per Job (sec) |
|---|---|---|---|
| FORTRAN | 1.43 | 11.19 | 12.62 |
| Other | 72.02 | 348.64 | 420.66 |
| FURPUR | .42 | 32.05 | 32.47 |
| MAP | 2.99 | 23.60 | 26.59 |
| PMD | 1.87 | 2.80 | 4.67 |
| COBOL | 16.33 | 67.21 | 83.54 |
| Demand | | | |
| FORTRAN | 1.60 | 51.84 | 53.44 |
| Other | 8.83 | 220.90 | 229.73 |
| FURPUR | .27 | 75.57 | 75.84 |
| MAP | 2.35 | 36.30 | 38.65 |
| PMD | 1.58 | 10.27 | 11.85 |
| COBOL | 8.67 | 58.13 | 66.80 |

# REGRESSION ANALYSIS

## Previous Studies

Reference [6] makes the point that a major difficulty in the evaluation of computer systems has been the inability to provide a valid quantification of the relationship between performance and workload. It is also mentioned that the problem of the presence of a large number of variables is compounded when the interactions of these variables must be taken into account. It is further stated that although the application of regression analyses to computer performance has not been extensive, it will become a major analysis tool. A regression equation can serve as a predictor of performance and resource usage, within the range of the variables which were used to estimate the regression coefficients. A disadvantage results from treating computer functions as black boxes, where the regression variables are black box inputs and outputs. This macro approach fails to deal with the internal structure of computer functions. The characteristics of the internal structures may be important determinants of computer performance. Also, it is possible to have a very good fit between dependent and independent variables without a cause and effect reason for the relationship. The goodness of the fit may mislead one to believe that the mathematical relationship implies a physical relationship.

Regression analysis has been applied or studied for the evaluation of computer systems in several instances. In [3] CPU utilization was the dependent variable and number of instructions executed/number of bytes transferred

and CPU channel overlap were the independent variables which provided the best fit. Data were collected by hardware monitor on the IBM 360/85. Reference [4] reports several regression relationships: percent non-overlapped time between CPU and I/O versus CPU to I/O time ratio; channel/channel overlap versus number of initiators; unoverlapped disk seek time versus number of channels and number of initiators; and channel busy time versus supervisor CPU time and number of channel program executions. No information was provided concerning the application of these equations to actual computer systems. Two interesting efforts [5, 8] were concerned with estimating CPU overhead, average throughput and maximum throughput in a time-sharing environment (CP-67). The CPU time consumed by the operating system was related to various event counts, such as number of I/O instructions issued, number of interrupts and number of pages read. Reference [7] describes the use of regression analysis to estimate the effect on system reaction time of the number and size of core regions used for APL programs in an IBM 360/91 operating under OS/MVT. Reaction time is made a function of workload variables such as number of conversational inputs, log ons and large CPU requests per hour; CPU utilization; and number and size of core workspaces. According to this study, the workspace size was more important than number of workspaces as a determinant of system reaction time. An interesting aspect of this study was the use of regression to produce a cumulative distribution function of system reaction time.

The above studies provide useful background for analyzing the subject of this paper. However, the thrust of the studies is the measurement and forecasting of a system (CPU overhead) performance variable as a function of various system or user activities. In addition, the studies make no distinction among the various types of programs with regard to performance and

resource usage. This paper is concerned with the measurement and forecasting of user performance variables (elapsed time or wait time) as a function of resource usage and input load for each type of program in a multiprogrammed mode.

Development of Regression Equations

As a result of the correlation analysis, certain high correlation variables emerged as candidates for the regression equations. Usually one or two "independent variables" dominate in the sense of being the major contributors to the explained variation about the mean due to regression (multiple correlation coefficient $r^2$). The following results of the correlation analysis provided guidelines for developing the regression equations:

- Higher correlations were achieved within batch or demand program categories than between these categories.
- Within the batch or demand program categories, higher correlations were achieved for variables within a program type than between program types.
- Higher correlations were achieved for elapsed time than for wait time.
- Number of jobs and CPU time were identified as the primary variables for predicting elapsed time by means of a regression equation.
- Overall, the best single independent variable is number of jobs because its value would be known and would not have to be estimated.

Only regression equations for the performance variable, elapsed time, were developed. This variable was emphasized because it proved to be the most difficult variable to forecast, since (1) the correlation coefficients were not high, and (2) elapsed time is a function of the variables of many program types. In most cases the resource usage regression equations will be easy to develop because (1) the correlation coefficients are high, and (2) resource usage is a function only of the variables of the given program type.

The following approaches to regression analysis were utilized:

(1) Linear regression. This represented a subset of stepwise linear regression results, wherein the regression equation obtained up to a point in the stepwise routine was utilized. This approach was used to determine whether adequate forecasting accuracy could be obtained with only a few variables.

(2) Stepwise linear regression using computer program BMD02R [9]. This approach was used to investigate the possibility of improving the forecast by using many variables.

(3) Polynomial regression using computer program BMD05R [9].

The last approach was used to determine whether a single variable equation which was non-linear in the independent variables could provide better accuracy than a linear equation with multiple variables.

A summary of the regression equations which were developed appears in Table 15. The details of a subset of these equations (those with $r > .7$) appears in Appendix III.

Several measures of the adequacy of the regression equations for forecasting purposes were utilized as follows:

- Sample multiple correlation coefficient $r$.
- Coefficient of determination $r^2$, the proportion of total variation about the mean of the dependent variable explained by the regression.
- Sum of the squared residuals SS.
- Residual mean square error of residuals SS/df, where df is the degrees of freedom. If the regression model is correct, this statistic provides an estimate of the error variance.
- An examination of residuals (difference between observed and predicted values of elapsed time) in order to determine whether the following two assumptions of a regression model are satisfied:

  (1) whether the residuals are normally distributed,

  (2) whether the residuals have constant variance.

- If $r \geq .7$ is used as one of the criterions for an acceptable regression equation, only some of the program types have regression equations which are acceptable.

Table 15

Sample Regression Equation Data

Elapsed Time as a Function of Resource Usage and/or Number of Jobs

N = 108 hours

Batch Programs

| I.D. | Program | Reg Type | Indep Var Type | No of Indep Var | r | $r^2$ | SS $\times 10^{-6}$ | (SS/df) $\times 10^{-3}$ | F |
|---|---|---|---|---|---|---|---|---|---|
| 1 | FORTRAN | L | 1 | 1 | .709 | .503 | 6.843 | 64.55 | 107.36 |
| 2 | FORTRAN | L | 3 | 1 | .716 | .513 | 6.710 | 63.30 | 111.58 |
| 3 | FORTRAN | S | 1,4 | 7 | .731 | .534 | 6.423 | 64.23 | 16.35 |
| 4 | FORTRAN | S | 1,2,3, 4,5,6 | 9 | .733 | .537 | 6.377 | 65.07 | 12.63 |
| 5 | FORTRAN | P2 | 1 | 1 | -- | -- | 6.209 | 59.13 | 63.96 |
| 6 | FORTRAN | S | 1,4 | 10 | .743 | .551 | 6.180 | 63.71 | 11.92 |
| 7 | FORTRAN | S | 1,2,3, 4,5,6 | 14 | .754 | .569 | 5.935 | 63.82 | 8.77 |
| 8 | Other | P2 | 1 | 1 | -- | -- | 1,739 | 16,562 | 26.34 |
| 9 | Other | P3 | 1 | 1 | -- | -- | 1,730 | 16,634 | 17.67 |
| 10 | Other | S | 1,4, 5,6 | 14 | .624 | .390 | 1,594 | 17,142 | 4.24 |
| 11 | Other | S | 1,4 | 11 | .633 | .401 | 1,565 | 16,301 | 5.84 |

I.D.: Regression equation identification used in Appendix III. Equations with $r > .7$ are shown in Appendix III.

Regression Type: L = linear; PX = polynomial of degree x; S = stepwise.

Independent Variable Type

1. This program's input load (no. of jobs, core usage).
2. This program's CPU time.
3. Wait for this program's I/O to complete.
4. Other program's input load (no. of jobs, core usage).
5. Wait for CPU to become available.
6. Wait for I/O to become available.

r: Sample coefficient of multiple correlation.

SS: Sum of squared residuals.

df: Degrees of freedom.

SS/df: Mean square error of residuals.

F: Ratio of sum of squared deviations due to regression to sum of squared residuals.

Table 15
(Continued)
N = 108 hours

Batch Programs

| I.D. | Program | Reg Type | Indep Var Type | No of Indep Var | r | $r^2$ | SS $\times 10^{-6}$ | (SS/df) $\times 10^{-3}$ | F |
|---|---|---|---|---|---|---|---|---|---|
| 12 | FURPUR | S | 1,4 | 7 | .680 | .462 | 13.796 | 137.96 | 12.26 |
| 13 | FURPUR | S | 1,4 | 12 | .683 | .466 | 13.680 | 142.50 | 7.63 |
| 14 | MAP | S | 1,4 | 2 | .765 | .586 | 3.144 | 29.94 | 74.21 |
| 15 | MAP | S | 1,4 | 7 | .778 | .605 | 2.996 | 29.96 | 21.90 |
| 16 | MAP | S | 1,4 | 10 | .791 | .626 | 2.842 | 29.30 | 16.20 |
| 17* | MAP | S | 1,3 | 3 | .831 | .690 | 1.742 | 22.93 | 56.51 |
| 18 | MAP | S | 1,3 | 3 | .876 | .767 | 1.765 | 16.97 | 114.40 |
| 19 | PMD | S | 1,4 | 10 | .604 | .365 | .042 | .43 | 5.58 |
| 20 | COBOL | S | 1,4 | 12 | .577 | .333 | 12.407 | 130.60 | 3.95 |

Demand Programs

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 21 | FORTRAN | S | 1,4 | 11 | .790 | .624 | 15.240 | 158.75 | 14.44 |
| 22 | Other | S | 1,4 | 12 | .884 | .781 | 364.046 | 3,832 | 28.16 |
| 23 | FURPUR | S | 1,4 | 12 | .849 | .721 | 161.119 | 1,696 | 20.48 |
| 24 | MAP | S | 1,4 | 11 | .789 | .622 | 6.504 | 67.750 | 14.36 |

*N = 80

In most cases the use of a large number of variables is not warranted, based on :

(1) the small increases in $r^2$ achieved (Table 15) by adding variables, particularly variables from other program types, and

(2) the fact that minimum SS/df does not necessarily occur when the largest number of independent variables is used (Table 15).

Analysis of Residuals

An analysis of residuals is shown in Table 16. The assumptions of the regression model are that the residuals are normally distributed with zero mean and constant variance. An approximate check of normality was made (Table 16) by determining the percentage of residuals which were within one and two standard deviations of the mean. The data in Table 16 indicates non-normality of the residuals at one standard deviation. The lack of normality prevents the use of F tests for testing:

(1) the hypothesis that all regression coefficients are zero or, equivalently, that the regression equation is no better for forecasting elapsed time than the mean value of elapsed time (using the F values in Table 15), and

(2) the hypothesis that individual regression coefficients are zero by calculating $F = (\text{regression coefficient})^2/(\text{standard error of regression coefficient})^2$ from the data in Appendix III.

As indicated in Appendix III, in many instances the standard error is much greater than the regression coefficient, indicating that the associated variable should not be included in the regression equation. Fortunately, this is usually not the case for the primary (high correlation) variables.

Computer plots were made of residuals versus the independent variables. These plots show that the residuals increase with increasing values of the independent variable of the given program type. Thus, the constant variance assumption is violated. This is caused by the increase in the variance of wait time (the major component of elapsed time) which occurs when a large number of jobs is resident in the system. The build up in queues causes an increase in the variability of wait time. The data in Table 16 also indicate that the regression equations consistently overestimate the actual values of elapsed time.

Table 16

Analysis of Residuals

N = 108 hours

| I.D. | Program | Reg Type | No. of Indep. Var. | S.D. | % Residuals Bet ±1S.D. | % Residuals Bet ±2S.D. | % Negative Residuals |
|---|---|---|---|---|---|---|---|
| 3 | FORTRAN Batch | S | 7 | 253 | 86 | 94 | 73 |
| 4 | FORTRAN Batch | S | 9 | 255 | 86 | 94 | 75 |
| 5 | FORTRAN Batch | P2 | 1 | 243 | 84 | 94 | 75 |
| 6 | FORTRAN Batch | S | 10 | 252 | 85 | 94 | 74 |
| 7 | FORTRAN Batch | S | 14 | 253 | 85 | 95 | 73 |
| 16 | MAP Batch | S | 10 | 171 | 84 | 96 | 61 |
| 21 | FORTRAN Demand | S | 11 | 398 | 80 | 94 | 63 |
| 22 | Other Demand | S | 12 | 1958 | 76 | 97 | 58 |
| 23 | FURPUR Demand | S | 12 | 1302 | 81 | 95 | 56 |
| 24 | MAP Demand | S | 11 | 259 | 93 | 97 | 51 |

I.D.: Regression equation identification used in Table 15.

Regression type: S = stepwise; P2 = polynomial of degree 2.

S.D. = standard deviation of residuals.

In order to produce more accurate forecasts of $T_j$, one of the following approaches for obtaining the regression equations could be employed:

(1) Use a linear weighted least square model for the purpose of achieving constant variance residuals.

(2) Use a model which is non-linear in the parameters in order to provide a better fit to the data (reduce the large number of negative residuals).

(3) Decrease the variability of elapsed time by partioning the observations into 24 sets, where each set is all the observations in a given one hour period over all the days which constitute the sample. The variability of observations should be reduced by using this method since the input to the system varies considerably by hour within each day, but the daily pattern of usage is consistent. However, the amount of computation is expanded significantly, being multiplied by a factor of 24. This procedure would correspond to using the linear programming formulation of (26) through (31).

(4) Since it was observed from Table 9 that CPU time $A_{1j}$ can usually be forecasted more accurately than elapsed time $T_j$, a CPU time regression equation $\hat{A}_{1j} = h_j(x_j)$ could be developed and used to forecast $T_j$ from the relationship $\hat{T}_j = h_j(x_j)/\hat{U}_j$, where the CPU utilization for program type $j$, $U_j$, is estimated from $\hat{U}_j = \sum A_{1j}/\sum T_j$ over the jobs which constitute the historical sample for program type $j$.

After obtaining $\hat{T}_j$, by one of the above methods, and $\hat{A}_{1j}$, it would be of interest to <u>forecast</u> $U_j$ from the relationship $\hat{U}_j = \hat{A}_{1j}/\hat{T}_j$.

## SUMMARY

### Research Efforts

Three major efforts have been discussed. The first effort dealt with the development of a mathematical programming model for determining the optimal mix of jobs to run in a computer center. The purpose of the second effort, correlation analysis, was to estimate the degree of linear association among variables. This was done in order to identify, for a multiprogramming operation, the high correlation variables and to confirm or reject intuitive notions regarding the linear association of variables. The identification of high

correlation variables led to the use of these variables in the third effort, regression analysis.

Applications

There are two applications of the regression analysis. One application involves using the regression equations in the formulation of the linear program. In this instance, the linear program objective function and constraints are not deterministic with constant coefficients. Rather, the objective function and constraints are functions of random variables. These functions are derived statistically in the form of regression equations, where the estimates of the dependent variables (performance or resource usage variables) are least squares estimates of the objective and constraints functions. The regression equation coefficients are the coefficients of the objective function and constraints. These coefficients are not constants as they would be in the ordinary linear programming formulation. Rather, these coefficients are least square estimates of random variables.

The second application of the regression equations is to make various forecasts and estimates as indicated below.

- Forecast the effect on performance of various resource usages and job mixes.
- Forecast resource usage as a function of input load $x_j$.
- Estimate the input load which would cause resource usage to equal or exceed resource capacity (saturation).
- Forecast the input load and resource usage which would cause unacceptable performance.

Once the linear program is available, the following types of analyses could be made:

- Compute the optimal job mix. If it is feasible to implement this job mix, improved computer performance should be possible.

- Attempt to validate the model by comparing the actual total elapsed time of job mixes which approximate the optimal job mix with the actual total elapsed time of non-optimal job mixes.
- Compare the actual job mix with the optimal job mix and compare actual total elapsed time with optimal total elapsed time. If the actual time exceeds the optimal time by a large amount, use the linear program to estimate the effect on performance and job mix of hardware changes (resource capacities), budget changes and differential pricing. In order to ascertain the effect of pricing which is based both on type of resource and time of use, it would be necessary to use the model given by (26) through (31).
- Determine the effect on optimal job mix and performance of changes in user production requirements.

Although the effect on performance of changes in the operating system cannot be evaluated by using the linear program, it may be possible to use the optimal job mix as the objective to be achieved by a change in operating system job scheduling and task dispatching priorities.

One of the benefits of the above analyses is to estimate the effect of a change in operation prior to the implementation of the change. For example, in view of the expense of making hardware changes, it would be advantageous to use the model as a means of estimating future performance and of justifying expenditures when equipment requests are made to the Navy's Automatic Data Processing Equipment Selection Office. In addition, since the Government Accounting Office has made CPU utilization a primary means of evaluating the effectiveness of Federal computer centers, the model could be used to estimate the effect of various changes on CPU time and elapsed time and, hence, on CPU utilization. In addition, the CPU times and elapsed times which correspond to the minimum CPU utilization constraints in the linear program, would be obtained from a solution of the linear program. Furthermore, when utilization constraints are not specified, optimal values of CPU utilization can be estimated from the optimal values of CPU time and elapsed time obtained from solving the linear program.

It would be important to reformulate the regression equations and linear program, by using new sample data, when significant changes in the computer operation occur, such as a change in hardware configuration.

## Future Research Efforts

In order to complete this model, additional work must be done to obtain more accurate regression equations for the performance variables. The techniques for achieving this result include (1) linear weighted least squares, (2) non-linear regression, (3) a reduction in variability by using more homogeneous sets of data, and (4) estimation of elapsed time from the CPU time regression equation and CPU utilization. If non-linear regression equations are used, they must be separable functions in order to make the numerical solution of the mathematical program feasible.

In addition to providing more accurate performance regression equations, it will be necessary to develop regression equations for the resource usage variables.

## Acknowledgment

The substantial assistance provided on this project by Mr. Robert Stirton of the Naval Weapons Center is gratefully acknowledged.

## REFERENCES


[1] D. N. Streeter, "Scientific Computing Service Evaluation," IBM System Journal, Vol. 11, No. 3, 1972, pp. 219-233.

[2] N. F. Schneidewind, "Analytical Model for the Design and Selection of Electronic Digital Computing Systems," University of Southern California, 1966, 308 pages.

[3] Walter Freiberger (ed.), Statistical Computer Performance Evaluation, Academic Press, New York, 1972, A. C. Yeh, "An Application of Statistical Methodology in the Study of Computer System Performance," pp. 287-328.

[4] Walter Freiberger (ed.), Statistical Computer Performance Evaluation, Academic Press, New York, 1972, M. P. Racite, "The Use of Pure and Modified Regression Techniques for Developing Systems Performance Algorithms," pp. 347-370.

[5] Walter Freiberger (ed.), Statistical Computer Performance Evaluation, Academic Press, New York, 1972, Y. Bard and D. V. Suryanarayana, "On the Structure of CP-67 Overhead," pp. 329-346.

[6] Walter Freiberger (ed.) Statistical Computer Performance Evaluation, Academic Press, New York, 1972, Ulf Grenander and Rhett F. Tsao, "Quantitative Methods for Evaluating Computer System Performance: A Review and Proposals," pp. 3-24.

[7] Gerlad Waldbaum, "Evaluating Computing System Changes by Means of Regression Models," First Annual SIGME Symposium on Measurement and Evaluation, 26-28 February 1973, pp. 127-135.

[8] Y. Bard, "Performance Criteria and Measurement for a Time Sharing System," IBM Systems Journal, Vol. 10, No. 3, 1971, pp. 193-216.

[9] W. J. Dixon (ed.), Biomedical Computer Programs, University of California Publication in Automatic Computations, No. 2, University of California Press, 1970.

[10] "NWC Central Computing Facility Computer Utilization and Accounting Guide," Naval Weapons Center, China Lake, California (undated).

[11] "Your Guide to the NWC Computer Center," Naval Weapons Center, China Lake, California, 17 April 1972, updated 1 December 1972.

[12] "UNIVAC 1108 User Reference Manual," Naval Weapons Center, China Lake, California, 2 October 1972.

[13] "UNIVAC 1100 Series Operating System," Sperry Rand Corporation, 1971.

APPENDIX I

Sample Data and Statistics

N = 108 hours

A. Number of Jobs in Sample

| Batch | Total Jobs(1) | Percent | Jobs per Hour Mean | Jobs per Hour Standard Deviation |
|---|---|---|---|---|
| FORTRAN | 2,574 | (19.8) | 23.82 | 26.94 |
| Other | 1,805 | (13.9) | 16.71 | 12.94 |
| FURPUR | 1,556 | (12.0) | 14.41 | 14.63 |
| MAP | 979 | (7.5) | 9.06 | 8.32 |
| PMD | 219 | (1.7) | 2.03 | 1.97 |
| COBOL | 239 | (1.8) | 2.21 | 2.95 |
| | 7,372 | (56.7) | | |
| Demand | | | | |
| FORTRAN | 716 | (5.5) | 6.63 | 8.88 |
| Other | 1,563 | (12.0) | 14.47 | 16.98 |
| FURPUR | 2,298 | (17.7) | 21.28 | 22.95 |
| MAP | 657 | (5.1) | 6.08 | 7.95 |
| PMD | 57 | (.5) | .53 | .92 |
| COBOL | 329 | (2.5) | 3.04 | 4.83 |
| | 5,620 | (43.3) | | |
| Total batch and demand | 12,992 | | | |

(1) Total number of jobs executed during five day period for each program. The sample data were collected during the following dates of operation in 1973:

July 6-7
July 10-11
July 17-18
July 20-21
July 27-28

APPENDIX I (continued)

N = 108 hours

B. CPU Time in Sample

| Batch | Total Time(1) (secs) | Percent | Secs per Hour: Mean | Secs per Hour: Standard Deviation | Secs per Job: Mean |
|---|---|---|---|---|---|
| FORTRAN | 3,684 | (2.3) | 34.11 | 38.82 | 1.43 |
| Other | 129,972 | (80.4) | 1,203.45 | 787.20 | 72.01 |
| FURPUR | 661 | (.5) | 6.12 | 7.59 | 1.09 |
| MAP | 2,922 | (1.8) | 27.06 | 24.43 | 2.98 |
| PMD | 410 | (.2) | 3.80 | 7.78 | 1.87 |
| COBOL | 3,898 | (2.4) | 36.09 | 50.38 | 16.31 |
| | 141,547 | (87.6%) | | | |
| Demand | | | | | |
| FORTRAN | 1,142 | (.7) | 10.58 | 18.21 | 1.59 |
| Other | 13,805 | (8.5) | 127.83 | 176.69 | 8.83 |
| FURPUR | 628 | (.4) | 5.82 | 8.79 | .27 |
| MAP | 1,541 | (.9) | 14.27 | 17.76 | 2.35 |
| PMD | 91 | (.1) | .84 | 4.01 | 1.60 |
| COBOL | 2,846 | (1.8) | 26.35 | 40.65 | 8.66 |
| | 20,053 | (12.4) | | | |
| Total batch and demand | 161,600 | | | | |

(1) Total CPU time used during five day period by each program.

APPENDIX I (continued)

N = 108 hours

C. Core Usage in Sample

| Batch | Total Kilo Words[1] | Percent | Kilo-words per Hour | | Kilo-words per Job |
|---|---|---|---|---|---|
| | | | Mean | Standard Deviation | Mean |
| FORTRAN | 110,690 | (30.7) | 1,024.91 | 1,158.67 | 43.00[2] |
| Other | 59,855 | (16.6) | 554.22 | 460.79 | 33.16 |
| FURPUR | 19,298 | (5.4) | 178.69 | 181.51 | 12.40[2] |
| MAP | 25,687 | (7.1) | 237.84 | 217.90 | 26.24[2] |
| PMD | 1,656 | (.5) | 15.33 | 14.89 | 7.56[2] |
| COBOL | 8,987 | (2.5) | 83.21 | 110.70 | 37.60[2] |
| | 226,173 | (62.8) | | | |
| Demand | | | | | |
| FORTRAN | 30,699 | (8.5) | 284.25 | 381.34 | 43.88[2] |
| Other | 44.787 | (12.5) | 414.70 | 511.88 | 28.65 |
| FURPUR | 28,204 | (7.9) | 261.14 | 281.74 | 12.27[2] |
| MAP | 17,253 | (4.8) | 159.75 | 208.89 | 26.26[2] |
| PMD | 427 | (.1) | 3.96 | 6.96 | 7.49[2] |
| COBOL | 12,351 | (3.4) | 114.36 | 181.46 | 37.54[2] |
| | 133,721 | (37.2) | | | |
| Total batch and demand | 359,894 | | | | |

(1) Total core usage during five day period by each program.

(2) These programs have approximately constant core usage per job and usage is the same for batch and demand.

APPENDIX I (continued)

N = 108 hours

D. Input/Output References in Sample

| Batch | Total Refs(1) | Percent | Refs per Hour: Mean | Refs per Hour: Standard Deviation | Refs per Job: Mean |
|---|---|---|---|---|---|
| FORTRAN | 169,167 | (2.1) | 1,566.36 | 1,798.23 | 65.76 |
| Other | 5,091,488 | (64.2) | 47,143.41 | 42,091.38 | 2,821.27 |
| FURPUR | 529,459 | (6.7) | 4,902.39 | 6,668.04 | 340.21 |
| MAP | 383.433 | (4.9) | 3,550.31 | 3,083.04 | 391.87 |
| PMD | 11,845 | (.1) | 109.68 | 203.14 | 54.03 |
| COBOL | 247,523 | (3.1) | 2,291.88 | 3,526.57 | 1,037.05 |
| | 6,432,915 | (81.1%) | | | |
| Demand | | | | | |
| FORTRAN | 61,112 | (.8) | 565.85 | 815.77 | 85.35 |
| Other | 729,423 | (9.2) | 6,753.91 | 10,802.05 | 466.75 |
| FURPUR | 272,274 | (3.4) | 2,521,06 | 4,309.85 | 118.47 |
| MAP | 235,269 | (3.0) | 2,178.42 | 2,691.00 | 358.29 |
| PMD | 4,482 | (.1) | 41.50 | 171.24 | 78.30 |
| COBOL | 193,264 | (2.4) | 1,789.48 | 2,726.97 | 588.64 |
| | 1,495,824 | (18.9%) | | | |
| Total batch and demand | 7,928,739 | | | | |

(1) Total number of I/O references made during five day period by each program.

APPENDIX I (continued)

N = 108 hours

E. Input/Output Transfers in Sample

| Batch | Total Kilo-Words(1) | Percent | Kilo-Words per Hour: Mean | Kilo-Words per Hour: Standard Deviation | Kilo-Words per Job: Mean |
|---|---|---|---|---|---|
| FORTRAN | 183,511 | (4.6) | 1,699.18 | 1,956.51 | 71.33 |
| Other | 2,450,657 | (61.4) | 22,691.27 | 32,668.43 | 1,357.95 |
| FURPUR | 310,467 | (7.8) | 2,874.70 | 3,838.91 | 199.49 |
| MAP | 313,617 | (3.3) | 1,218.67 | 1,081.69 | 134.51 |
| PMD | 3,552 | (.1) | 32.89 | 58.09 | 16.20 |
| COBOL | 127,758 | (3.2) | 1,182.94 | 1,813.47 | 535.27 |
| | 3,207,562 | (80.4) | | | |
| Demand | | | | | |
| FORTRAN | 61,370 | (1.5) | 568.24 | 782.98 | 85.71 |
| Other | 321,211 | (8.0) | 2,974.18 | 5,048.76 | 205.54 |
| FURPUR | 212,046 | (5.3) | 1,963.39 | 2,899.68 | 92.26 |
| MAP | 88,077 | (2.2) | 815.53 | 1,018.94 | 134.13 |
| PMD | 1,212 | (.1) | 11.23 | 44.39 | 21.19 |
| COBOL | 98,958 | (2.5) | 916.28 | 1,398.79 | 301.41 |
| | 782,874 | (19.6) | | | |
| Total batch and demand | 3,990,436 | | | | |

(1) Total words transferred during five day period by each program.

# APPENDIX II

## Variable Identification

### Program Type j

| Program | Batch | Demand |
|---|---|---|
| FORTRAN | 1 | 7 |
| Other | 2 | 8 |
| FURPUR | 3 | 9 |
| MAP | 4 | 10 |
| PMD | 5 | 11 |
| COBOL | 6 | 12 |

### Resource i

| | | |
|---|---|---|
| CPU Time | 1 | Seconds per hour |
| Core Usage | 2 | Kilowords per hour |
| I/O References | 3 | References per hour |
| I/O Words Transferred | 4 | Kilowords per hour |

### Elapsed Time Variables

Total elapsed time in seconds per program type $j$ per hr. = $T_j$

Elapsed time in seconds per program type $j$ per hr. per job = $t_j$

### Resource Usage Variables

Total use of resource $i$ by program type $j$ per hr. = $A_{ij}$

Use of resource $i$ by program type $j$ per hr. per job = $a_{ij}$

### Job Variables

Number of jobs of program type $j$ per hr. = $x_j$.

APPENDIX III

Sample Regression Equations

N = 108 hours

| I.D. | Dep Var | r | Independent Variables: Var | Type | ρ with Dep Var | ρ with Own Jobs | Reg Coeff | Std Err Reg Coeff | Inter-cept |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $T_1$ | .709 | $x_1$ | 1 | .709 | 1.0 | 9.448 | .912 | 75.62 |
| 2 | $T_1$ | .716 | $A_{31}$ | 3 | .716 | .991 | .143 | .014 | 76.99 |
| 3 | $T_1$ | .731 | $x_1$ | 1 | .709 | 1.0 | 7.356 | 1.596 | 70.77 |
| | | | $x_2$ | 4 | .519 | 1.0 | 10.890 | 7.677 | |
| | | | $x_3$ | 4 | .368 | 1.0 | 4.670 | 3.267 | |
| | | | $x_4$ | 4 | .632 | 1.0 | .757 | 2.344 | |
| | | | $x_5$ | 4 | .069 | 1.0 | -9.243 | 13.596 | |
| | | | $x_6$ | 4 | .227 | 1.0 | -5.516 | 10.468 | |
| | | | $x_7$ | 4 | .400 | 1.0 | -3.286 | 3.815 | |

I.D.: Regression identification referred to in Table 15. Only equations with $r > .7$ are shown here.

r: Sample coefficient of multiple correlation.

ρ: Sample coefficient of correlation.

Independent Variable Type

1. This program's input load (no. of jobs, core usage).
2. This program's CPU time.
3. Wait for this program's I/O to complete.
4. Other programs' input load (no. of jobs, core usage).
5. Wait for CPU to become available.
6. Wait for I/O to become available.

APPENDIX III (continued)

| I.D. | Dep Var | r | Independent Variables: Var | Type | ρ with Dep Var | ρ with Own Jobs | Reg Coeff | Std Err Reg Coeff | Inter-cept |
|---|---|---|---|---|---|---|---|---|---|
| 4 | $T_1$ | .733 | $A_{11}$ | 2 | .707 | .962 | -.759 | 3.802 | 49.76 |
| | | | $A_{21}$ | 1 | .709 | 1.0 | .083 | .256 | |
| | | | $A_{31}$ | 3 | .716 | .991 | .268 | .206 | |
| | | | $A_{41}$ | 3 | .710 | .996 | -.178 | .217 | |
| | | | $x_4$ | 4 | .632 | 1.0 | 10.579 | 32.817 | |
| | | | $A_{14}$ | 5 | .570 | .905 | -4.948 | 4.331 | |
| | | | $A_{24}$ | 4 | .622 | .995 | -.412 | 1.299 | |
| | | | $A_{34}$ | 6 | .601 | .926 | -.011 | .052 | |
| | | | $A_{44}$ | 6 | .619 | .959 | .206 | .235 | |
| 5 | $T_1$ | -- | $x_1$ | 1 | .709 | 1.0 | 16.038 | 2.194 | 13.98 |
| | | | $x_12$ | 1 | -- | -- | -.074 | .023 | |
| 6 | $T_1$ | .743 | $x_1$ | 1 | .709 | 1.0 | 6.224 | 1.691 | 48.75 |
| | | | $x_2$ | 4 | .519 | 1.0 | -3.412 | 3.955 | |
| | | | $x_4$ | 4 | .632 | 1.0 | 13.434 | 7.829 | |
| | | | $x_5$ | 4 | .069 | 1.0 | -2.198 | 13.864 | |
| | | | $x_6$ | 4 | .227 | 1.0 | -6.925 | 10.488 | |
| | | | $x_8$ | 4 | .339 | 1.0 | -1.034 | 3.799 | |
| | | | $x_9$ | 4 | .486 | 1.0 | 2.379 | 3.241 | |
| | | | $x_{10}$ | 4 | .425 | 1.0 | 7.467 | 8.340 | |
| | | | $x_{11}$ | 4 | .177 | 1.0 | 21.727 | 28.740 | |
| | | | $x_{12}$ | 4 | .212 | 1.0 | -11.115 | 9.112 | |
| 7 | $T_1$ | .754 | $A_{11}$ | 2 | .707 | .962 | .925 | 4.056 | 65.70 |
| | | | $A_{21}$ | 1 | .709 | 1.0 | .074 | .266 | |
| | | | $A_{31}$ | 3 | .716 | .991 | .278 | .208 | |
| | | | $A_{41}$ | 3 | .710 | .996 | -.207 | .223 | |
| | | | $x_2$ | 4 | .519 | 1.0 | -8.705 | 7.574 | |
| | | | $A_{22}$ | 4 | .454 | .937 | .059 | .173 | |
| | | | $x_4$ | 4 | .632 | 1.0 | 3.060 | 35.777 | |
| | | | $A_{14}$ | 5 | .570 | .905 | -5.521 | 4.688 | |
| | | | $A_{24}$ | 4 | .622 | .995 | .175 | 1.390 | |

APPENDIX III (continued)

| I.D. | Dep Var | r | Independent Variables: Var | Type | ρ with Dep Var | ρ with Own Jobs | Reg Coeff | Std Err Reg Coeff | Inter-cept |
|---|---|---|---|---|---|---|---|---|---|
| | | | $A_{44}$ | 6 | .619 | .959 | .154 | .164 | |
| | | | $x_8$ | 4 | .449 | 1.0 | -6.320 | 6.136 | |
| | | | $A_{28}$ | 4 | .438 | .949 | .240 | .175 | |
| | | | $A_{29}$ | 4 | .486 | 1.0 | .075 | .265 | |
| | | | $x_{10}$ | 4 | .425 | 1.0 | 3.590 | 7.891 | |
| 14 | $T_4$ | .765 | $x_4$ | 1 | .743 | 1.0 | 19.875 | 2.403 | 16.44 |
| | | | $x_6$ | 4 | .562 | 1.0 | 20.019 | 6.785 | |
| 15 | $T_4$ | .778 | $x_1$ | 4 | .541 | 1.0 | -.528 | 1.090 | -11.38 |
| | | | $x_2$ | 4 | .679 | 1.0 | 2.750 | 2.605 | |
| | | | $x_3$ | 4 | .511 | 1.0 | .512 | 1.601 | |
| | | | $x_4$ | 1 | .743 | 1.0 | 15.251 | 5.243 | |
| | | | $x_5$ | 4 | .206 | 1.0 | 4.508 | 9.285 | |
| | | | $x_6$ | 4 | .562 | 1.0 | 18.801 | 7.149 | |
| | | | $x_7$ | 4 | .452 | 1.0 | 3.402 | 2.231 | |
| 16 | $T_4$ | .791 | $x_2$ | 4 | .679 | 1.0 | 2.470 | 2.686 | -17.58 |
| | | | $x_4$ | 1 | .743 | 1.0 | 15.225 | 4.321 | |
| | | | $x_5$ | 4 | .206 | 1.0 | 3.156 | 9.383 | |
| | | | $x_6$ | 4 | .562 | 1.0 | 19.720 | 6.726 | |
| | | | $x_7$ | 4 | .452 | 1.0 | 3.298 | 3.375 | |
| | | | $x_8$ | 4 | .482 | 1.0 | -.816 | 2.632 | |
| | | | $x_9$ | 4 | .483 | 1.0 | -1.928 | 2.315 | |
| | | | $x_{10}$ | 4 | .455 | 1.0 | 2.259 | 5.631 | |
| | | | $x_{11}$ | 4 | .212 | 1.0 | 27.371 | 19.599 | |
| | | | $x_{12}$ | 4 | .402 | 1.0 | 10.503 | 5.922 | |
| 17[a] | $T_4$ | .831 | $x_4$ | 1 | .743 | 1.0 | -32.931 | 7.003 | 21.59 |
| | | | $A_{34}$ | 3 | .799 | .926 | -.124 | .034 | |
| | | | $A_{44}$ | 3 | .830 | .959 | .789 | .124 | |
| 18[b] | $T_4$ | .876 | $x_4$ | 1 | .743 | 1.0 | -31.945 | 5.867 | 9.00 |
| | | | $A_{34}$ | 3 | .799 | .926 | -.117 | .027 | |
| | | | $A_{44}$ | 3 | .830 | .959 | .768 | .101 | |

[a] N = 80

[b] N = 108

APPENDIX III (continued)

| I.D. | Dep Var | r | Independent Variables: Var | Type | ρ with Dep Var | ρ with Own Jobs | Reg Coeff | Std Err Reg Coeff | Inter-cept |
|---|---|---|---|---|---|---|---|---|---|
| 21 | $T_7$ | .790 | $x_1$ | 4 | .218 | 1.0 | -8.916 | 2.588 | 60.79 |
| | | | $x_2$ | 4 | .355 | 1.0 | -8.379 | 6.350 | |
| | | | $x_3$ | 4 | .229 | 1.0 | 1.224 | 3.809 | |
| | | | $x_4$ | 4 | .405 | 1.0 | 35.546 | 11.916 | |
| | | | $x_5$ | 4 | -.0182 | 1.0 | -6.944 | 22.424 | |
| | | | $x_7$ | 1 | .732 | 1.0 | 42.200 | 7.978 | |
| | | | $x_8$ | 4 | .614 | 1.0 | 6.695 | 6.174 | |
| | | | $x_9$ | 4 | .595 | 1.0 | -7.331 | 5.417 | |
| | | | $x_{10}$ | 4 | .606 | 1.0 | 36.727 | 13.248 | |
| | | | $x_{11}$ | 4 | .041 | 1.0 | -30.695 | 46.875 | |
| | | | $x_{12}$ | 4 | .404 | 1.0 | -35.377 | 14.620 | |
| 22 | $T_8$ | .884 | $x_1$ | 4 | .441 | 1.0 | 4.479 | 13.457 | 562.98 |
| | | | $x_2$ | 4 | .447 | 1.0 | -25.097 | 31.201 | |
| | | | $x_3$ | 4 | .263 | 1.0 | 4.526 | 18.721 | |
| | | | $x_4$ | 4 | .460 | 1.0 | 45.363 | 63.307 | |
| | | | $x_5$ | 4 | -.068 | 1.0 | -189.611 | 110.226 | |
| | | | $x_6$ | 4 | .207 | 1.0 | -70.285 | 81.423 | |
| | | | $x_7$ | 4 | .709 | 1.0 | 24.131 | 38.225 | |
| | | | $x_8$ | 1 | .850 | 1.0 | 143.376 | 30.410 | |
| | | | $x_9$ | 4 | .816 | 1.0 | -8.997 | 26.614 | |
| | | | $x_{10}$ | 4 | .800 | 1.0 | 89.684 | 65.092 | |
| | | | $x_{11}$ | 4 | .313 | 1.0 | 751.111 | 230.484 | |
| | | | $x_{12}$ | 4 | .626 | 1.0 | 49.987 | 72.032 | |
| 23 | $T_9$ | .849 | $x_1$ | 4 | .399 | 1.0 | -8.781 | 8.952 | -146.98 |
| | | | $x_2$ | 4 | .369 | 1.0 | -26.495 | 20.757 | |
| | | | $x_3$ | 4 | .191 | 1.0 | -12.489 | 12.454 | |
| | | | $x_4$ | 4 | .474 | 1.0 | 76.253 | 42.781 | |
| | | | $x_5$ | 4 | -.007 | 1.0 | 103.769 | 73.329 | |
| | | | $x_6$ | 4 | .214 | 1.0 | -56.268 | 54.168 | |
| | | | $x_7$ | 4 | .714 | 1.0 | 38.284 | 26.095 | |
| | | | $x_8$ | 4 | .710 | 1.0 | -19.235 | 20.231 | |

APPENDIX III (continued)

| I.D. | Dep Var | r | Independent Variables: Var | Type | ρ with Dep Var | ρ with Own Jobs | Ref Coeff | Std Err Reg Coeff | Inter-cept |
|---|---|---|---|---|---|---|---|---|---|
| | | | $x_9$ | 1 | .827 | 1.0 | 100.251 | 17.705 | |
| | | | $x_{10}$ | 4 | .706 | 1.0 | -17.431 | 43.304 | |
| | | | $x_{11}$ | 4 | .151 | 1.0 | -55.856 | 153.333 | |
| | | | $x_{12}$ | 4 | .550 | 1.0 | -51.721 | 47.921 | |
| 24 | $T_{10}$ | .789 | $x_1$ | 4 | .272 | 1.0 | -2.954 | 1.788 | -3.65 |
| | | | $x_3$ | 4 | .228 | 1.0 | 1.472 | 2.451 | |
| | | | $x_4$ | 4 | .373 | 1.0 | 6.697 | 7.405 | |
| | | | $x_5$ | 4 | -.017 | 1.0 | -3.233 | 14.441 | |
| | | | $x_6$ | 4 | .160 | 1.0 | -11.309 | 10.826 | |
| | | | $x_7$ | 4 | .556 | 1.0 | -6.103 | 5.202 | |
| | | | $x_8$ | 4 | .647 | 1.0 | -4.913 | 3.887 | |
| | | | $x_9$ | 4 | .738 | 1.0 | 11.165 | 3.436 | |
| | | | $x_{10}$ | 1 | .748 | 1.0 | 25.473 | 8.633 | |
| | | | $x_{11}$ | 4 | .080 | 1.0 | -35.915 | 30.483 | |
| | | | $x_{12}$ | 4 | .633 | 1.0 | -1.075 | 9.570 | |

## APPENDIX IV

### Naval Weapons Center Computer Center

The NWC Computer Center performs both batch and demand (interactive) processing on an open shop basis for a variety of scientific and management applications. The Center has approximately 700 users in total, serves 300 to 400 users each week and processes 4,000 to 5,000 jobs per week on a five day per week, three shift basis [10]. About four jobs are usually processed concurrently.

The data which were used in the correlation and regression analysis were collected from the Center's UNIVAC 1108 system. This system operated under the control of EXEC 8, level 28. The data were collected during a five day period in July 1973. A description of the sample appears in Appendix I. Since this time, the Center has upgraded the system to a UNIVAC 1110.

Description of Hardware [11, 12]

The major components of the hardware system, corresponding to the time period of the sample data, will now be described.

1. Central Processing Unit (CPU)

   There is one CPU with 128 control registers and a 125 nanosecond access time. There are 155 instructions, many of which execute in 750 nanoseconds. The CPU uses 11 input/output channels.

2. Main Memory

   There are three modules of 750 nanosecond cycle time core memory. Each module consists of two banks of 32,768 thirty-six bit words for a total of 196,608 words. Two modules (4 banks) or 131, 072 words are available for user programs. A single user program is normally limited to 65,000 words. One module is reserved for the storage of the resident portion of EXEC 8.

3. Auxiliary Storage

FH-432 Drum Units

- There are six flying head drum units which are used for the storage of non-resident portions of EXEC 8 and frequently used system processors, such as FORTRAN. These drums have a storage capacity of 262,144 thirty-six bit words, or 1,572,864 words in total, an average access time of 4.25 milliseconds and a transfer rate of 240,000 words per second.

FH-1782 Drum Unit

- This unit is used for the storage of catalogues, user and production programs and data. Drum storage capacity is 2,097,152 thirty-six bit words, with an average access time of 17 milliseconds and a transfer rate of 240,000 words per second.

AMPEX Disc Units

- These disc units emulate the UNIVAC Fastrand drum format but operate at a higher speed than the UNIVAC units. These units are used for the storage of programs and data which are referenced and created during job processing. There are eighteen disc units, with each unit capable of storing 7,340,032, thirty-six bit words, or a total of 132,120,576 words. The average access time is 30 milliseconds and the transfer rate is 52,000 words per second.

  The variables "I/O References" and "I/O Words Transferred," which were used in the correlation and regression analysis, pertain to all of the above auxiliary storage units.

UNISERVO Tape Units

- These tape units provide permanent storage for user programs and data. There are thirteen 7 track drives with 200/556/800 bpi and one 9 track with 800/1600 bpi. Both tape units operate at 120 in./sec.

4. UNIVAC 9300 Computers and Data Communications Terminals

One UNIVAC 9400 at the central site and two remotely located computers serve as I/O interfaces for card readers, card punches and high speed printers. There are also three remotely located Data Communication Terminals for interfacing card readers, card punches and printers.

5. Communications Subsystem

For demand processing, a communications subsystem is provided which consists of a Communications Terminal Module Controller for multiplexing terminal communication lines, and a variety of terminals (approximately 64 in number), including UNISCOPE 300, teletype, crt and graphics terminals.

Description of Programs [12, 13]

The following is a brief description of the types of programs which were used in the correlation and regression analysis. Six types of programs were analyzed. The same types were analyzed for both batch and demand processing. These programs constitute approximately two-thirds of the total computing load of the NWC Computer Center.

1. FORTRAN

A language processor which compiles FORTRAN V source language statements into relocatable binary code. This processor can also be used to store or update source statements.

2. OTHER

This is the name given to scientific and management production programs. These are executable programs which have been previously compiled or assembled by one of the language processors.

3. FURPUR

This is a set of file utility routines which is used to perform file copying, deletion, listing, positioning, marking, name changing, rewinding, closing

and punching. File operations involve the use of magnetic tape and mass storage units. Since a number of file options are available to the user, the execution characteristics of FURPUR are not constant but vary considerably as a function of the operation to be performed.

4. MAP

A system processor for collecting and combining relocatable program elements which have been generated by a language processor into executable (absolute) program elements. The absolute elements are structured so that the loader can place the absolute elements in execution. It is possible to save and reexecute the absolute elements many times. Recollection is only required when the relocatable elements have been modified.

5. PMD (Postmortem Dump Processor)

At program termination, the PMD writes the final contents of a program's main storage area into the diagnostic file and then edits and prints the data.

6. COBOL

A language processor which compiles COBOL source language statements into relocatable binary code. This processor can also be used to store or update source statements.

The above programs were executed under the control of the EXEC 8 operating system which, in total, requires two million words of storage.

INITIAL DISTRIBUTION LIST

| | Copies |
|---|---|
| Dean of Research<br>Code 023<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 2 |
| Library (Code 0212)<br>Naval Postgraduate School<br>Monterey, California 93940 | 2 |
| Library (Code 55)<br>Naval Postgraduate School<br>Monterey, California 93940 | 2 |
| Mr. Lee Lakin | 1 |
| Mr. Robert Stirton | 1 |
| Dr. Delbert Zilmer<br>Naval Weapons Center<br>China Lake, California 93555 | 1 |
| W. R. Church Computer Center<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| Capt. Stephan Ruth<br>Code 0451<br>Supply Systems Command<br>Department of the Navy<br>Washington, D. C. 20376 | 1 |
| Naval Electronics Laboratory<br>Center Library<br>271 Catalina Boulevard<br>San Diego, California 92152 | 1 |
| Mr. Michael Griswold<br>FCDSSA<br>270 Catalina Boulevard<br>San Diego, California 92147 | 1 |
| Mr. Marvin Denicoff<br>Office of Naval Research<br>Department of the Navy<br>Arlington, Virginia 22217 | 1 |

CDR J. S. Prokop 1
Automatic Data Processing Equipment
Selection Office
Department of the Navy
Washington, D. C. 20376

Professor D. Gaver 1
Professor G. Bradley 1
Professor G. Howard 1
Professor A. McMasters 1
Professor G. Barksdale 1
Professor U. Kodres 1
Professor V. M. Powers 1
Naval Postgraduate School
Monterey, California 93940

Professor N. F. Schneidewind 50