

A SIMULATION STUDY OF ALTERNATIVES TO
UPGRADING LARGE COMPUTER SYSTEMS

by

DANIEL E. KREIMER

B.S. University of Missouri, Columbia, 1970

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

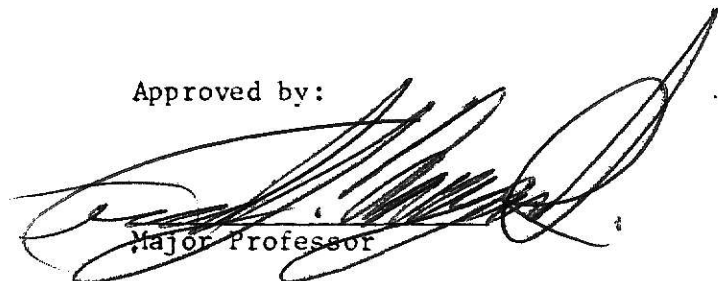
MASTER OF SCIENCE

Department of Computer Science

Kansas State University
Manhattan, Kansas

1977

Approved by:



Major Professor

Document
LD
2668
R4
1977
K74
c.2

TABLE OF CONTENTS

Acknowledgements

1.0 Introduction.....1

 1.1 Overview.....1

 1.2 Background.....1

 1.3 Increased Demand for Services.....4

 1.4 Approaches to Evaluation.....4

 1.5 Additional Processing Capability.....5

 1.6 Objectives.....6

2.0 The Model Systems.....8

 2.1 Overview.....8

 2.2 Model Requirements.....8

 2.3 General Assumptions.....9

 2.4 Basic System.....10

 2.5 Front-end Monitor.....12

 2.6 Back-end DBMS.....14

 2.7 Front-end Monitor/DBMS.....14

 2.8 Front-end Monitor, Back-end DBMS.....17

| | |
|---------------------------------|----|
| 3.0 Implementation..... | 19 |
| 3.1 Overview..... | 19 |
| 3.2 Model Description..... | 19 |
| 3.3 Communications Monitor..... | 20 |
| 3.4 Batch System..... | 22 |
| 3.5 DBMS Routine..... | 24 |
| 3.6 CPU Scheduler..... | 25 |
| 4.0 Results and Analysis..... | 26 |
| 4.1 Overview..... | 26 |
| 4.2 Test Method..... | 26 |
| 4.3 Simulation Results..... | 27 |
| 4.4 Cost Effectiveness..... | 30 |
| 5.0 Conclusion..... | 32 |
| 5.1 Summary..... | 32 |
| 5.2 Extentions..... | 33 |

| | |
|--------------------------------|----|
| References..... | 35 |
| Appendices..... | 37 |
| I. ENVIRON/1, Description..... | 37 |
| II. Program Flowchart..... | 40 |
| III. Program Results..... | 43 |
| IV. Program Listing..... | 52 |

LIST OF FIGURES

| | |
|---|----|
| 1.1 NETISA Computer System..... | 3 |
| 2.1 Basic system..... | 11 |
| 2.2 Front-end Monitor..... | 13 |
| 2.3 Back-end DBMS..... | 15 |
| 2.4 Front-end Monitor/DBMS..... | 16 |
| 2.5 Front-end Monitor, Back-end DBMS..... | 18 |
| A1.1 Communications Monitor..... | 38 |

ACKNOWLEDGEMENTS

I would like to thank Drs. Fred Maryanski, Virgil Wallentine and Myron Calhoun for serving on my committee and offering their comments and corrections. A special thanks is given to Dr. Maryanski for his help and guidance in preparing this report.

DAN KREIMER

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In this chapter some problems associated with the growth of computer systems will be reviewed. These problems are created by the increase in demand for a computer's resources. This increase requires that a system evolve from one configuration to another; that is, the system must be continuously expanded or upgraded to handle a larger workload. Two questions will be considered:

1) What is meant by throughput in a computer system and how is it measured?

2) Can this measure of throughput be used to determine the most cost-effective configuration of a computer system at any moment in its growth cycle?

1.2 BACKGROUND

This project evolved from a study performed for the Naval Education and Training Information Systems Activity (NETISA) by this author and Dr. Fred Maryanski. The purpose of the study was to determine a suitable replacement for a computer system which has reached the limits of its

processing capabilities. The system is shown in Figure 1.1.

The hardware consists of an IBM 360/65 with 2.5 MB of real memory. On-line disk storage is equivalent to 36 IBM-3330-type drives. Twelve of these drives are devoted to an on-line data base system. Several tape drives, card readers and printers are available for batch input/output operations. Approximately 150 remote interactive terminals and 10 remote batch terminals are attached through an IBM-3705-compatible communications controller. It is planned that in the future over 500 terminals will be supported with an on-line data base of possibly 30 disk units.

The operating system currently in use is OS/MVT. One fixed partition of 500 KB is used to support a telecommunications monitor, ENVIRON/1 (1), which is marketed by CINCOM INC. This monitor (described in Appendix I) provides on-line data base inquiry/update capability for the remote terminals. The TOTAL data base system (2) is used for data management and BTAM is used for terminal access.

Approximately 1.4 MB of memory is devoted to batch processing. An average of 4 batch jobs are run concurrently and the job mix is varied between compute-bound and I/O-bound jobs. Jobs which access the data base have their own separate copies of TOTAL so careful scheduling must be used to avoid conflicts between the batch jobs and the on-line programs.

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPARED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**

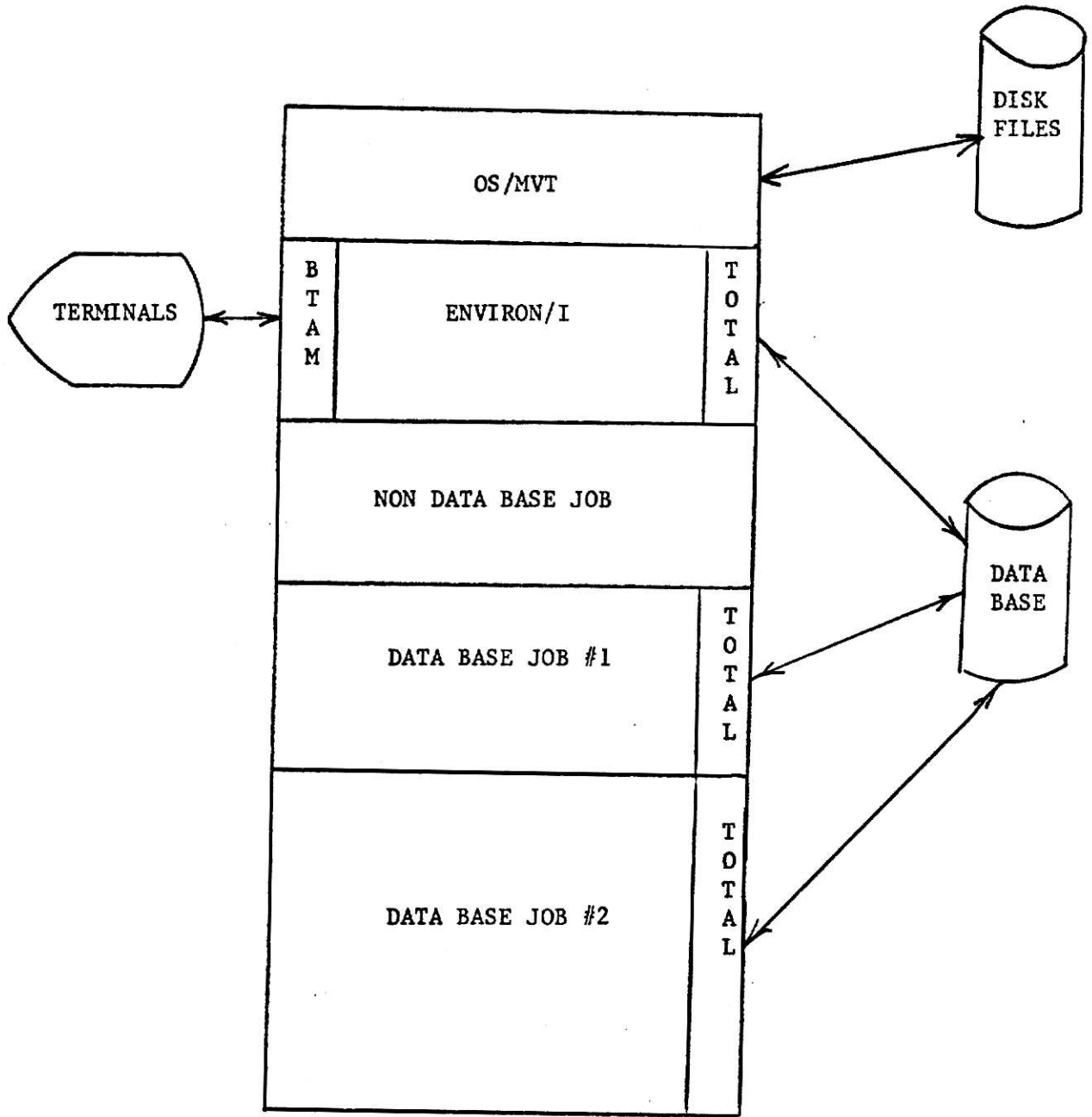


Figure 1.1. NETISA IBM 360/65

1.3 INCREASED DEMAND FOR SERVICES

The NETISA computing center was faced with increasing demands for services. In addition to the increase in the number of terminal users and in the size of the data base, more batch and real-time applications were planned. This increased demand had resulted in slower terminal response time and slower user turnaround on jobs of all types. It was evident that a system with greater throughput was needed.

In making the decision to upgrade or expand, there are few guidelines available. Since the demand on each particular system is different, there are no formulas or rules of thumb to draw upon. The decisions to expand must address the size and speed of memory, number and speed of processors, disks and drums, number and type of data channels. Indeed, the evaluations upon which to base the decisions may become a major part of the cost of expansion.

1.4 APPROACHES TO EVALUATION

Two approaches have been taken in evaluating the capability of a given computer system, analytic and simulation. In some systems the analytic method proves useful. Scherr (3) designed a simple model of the Project MAC system at MIT and Smith (4) constructed a model of a paged time sharing system. However, these models usually lack flexibility and cannot be easily changed to reflect different system configurations.

Simulation models, however, can be quite flexible as

**THIS BOOK CONTAINS
NUMEROUS PAGE
NUMBERS THAT ARE
ILLEGIBLE**

**THIS IS AS RECEIVED
FROM THE
CUSTOMER**

shown by Scherr (3), O'Connor (5) and Nielsen (6). These models are also dynamic and are better suited to investigating a constantly changing system. While there are many pitfalls in trying to simulate a computer system (6), simulation seems to be the best tool available at this time to evaluate system configurations and decisions can be based on these evaluations at a reasonable cost.

1.5 ADDITIONAL PROCESSING CAPABILITY

Faced with a backlog of batch and real-time applications, NETISA decided to upgrade to a larger system and let the major computer vendors rely on their experience to choose the correct hardware and software. A multi-processor configuration was planned. This upgrade would solve the immediate and long range problems but at an enormous price. A more cost-effective solution might be to extend the life of the existing system by off-loading some of the functions to additional processors. A simulation study could be used to evaluate these alternatives. The following possibilities exist:

- 1) Put the telecommunications monitor on a front-end processor. The monitor uses considerable CPU time for message processing, page swapping and communications overhead. In addition, the monitor requires 500 KB of main memory which could be used for new applications.

- 2) Move the data base system to a back-end processor.

This idea is not new (7); in fact, several data base systems including TOTAL have been put on minicomputers. The amount of additional throughput gained would depend on how heavily the DBMS was being utilized and also how efficient it was in processing requests. Studies have shown (8) that up to 40% of the CPU time in a large system may be devoted to DBMS processing. This solution seems very attractive when additional CPU time is needed.

3) Both the data base system and the monitor could be put on a front-end processor. It would appear that this solution would help the main processor (more memory, less CPU demand) but possibly not help the monitor.

4) Put the monitor on a front-end processor and the data base system on a back-end processor. This may be an effective but expensive solution. The use of minicomputers could make this solution more cost-effective, however.

1.6 OBJECTIVES

In this study a simulation model of the basic NETISA system and the four variations described above will be created. Each system will provide batch processing, telecommunications services and data base management. The parameters on the basic system will be adjusted so that the system resources are saturated and the throughput of each

model will be measured to determine the advantages and disadvantages of that particular configuration. This will demonstrate how simulation is an effective tool in evaluating possible system expansion and upgrades.

CHAPTER 2

THE MODEL SYSTEMS

2.1 OVERVIEW

This chapter presents the five models used in this study. The first model represents the basic system which will serve as a reference. The resources (CPU and memory) of this reference system are fully utilized.

The remaining four models are variations of the basic model. Each variation provides additional processing capability by transferring some of the load to an additional processor. The gain in capability will depend on the job mix and terminal loading of the reference model.

2.2 MODEL REQUIREMENTS

If these models are to serve as a tool in an evaluation they must meet certain requirements:

- 1) The configuration should be easily adjustable by modification of a few parameters. It should take only minor modification to change configurations.

- 2) The models should be responsive to changes in parameters. It may be necessary to make simulation runs with various job mixes and memory sizes or number of

terminals.

3) The level of detail should be such as to include the important variables yet still allow the model to run in a reasonable amount of time. This will require that activities an order of magnitude larger or smaller than this level be analyzed in a separate study or studies and the results be input to the main simulation model as parameter values.

4) The various algorithms and components of the model should be modular so they can be easily changed.

5) The models should give meaningful statistics on queue sizes and resource utilization as well as overall system throughput. This gives additional information on how the resources will be utilized in a particular configuration.

2.3 GENERAL ASSUMPTIONS

In all the models the CPU's and operating systems are assumed to be equivalent and comparable to an IBM 360/65 with OS/MVT. This assumption is made to simplify the model and may not reflect an actual system. A minicomputer which is more cost-effective would probably be used for the front-end or back-end processors. However, since the processing power of some mini's are approaching that of large machines, especially in a single application, this assumption is

justified.

The amount of main memory allocated to the operating systems is assumed to be constant. This is not strictly true since a change from one configuration to another will usually cause a change in the size of the resident operating system. However, the change should not be large enough to affect the model.

The CPU's are connected by a channel-to-channel adaptor so the transmission delay between machines is negligible. This seems to be a reasonable assumption since the DBMS requests and responses are short and sent at very high speed. A small amount of delay is included in the model for queueing .

2.4 BASIC SYSTEM (I)

The basic system is shown in Figure 2.1. It is similar to the system described in Chapter 1. A fixed partition of 500 KB is allocated to the communications monitor. This monitor is modeled after Cincom's ENVIRON/1 and is typical of many teleprocessing systems which are currently in use. The amount of load that this monitor places on the CPU is highly dependent on the number of terminals on line and the particular applications which are being performed. The parameters for the model are taken from the NETISA study and appear in Chapter 3.

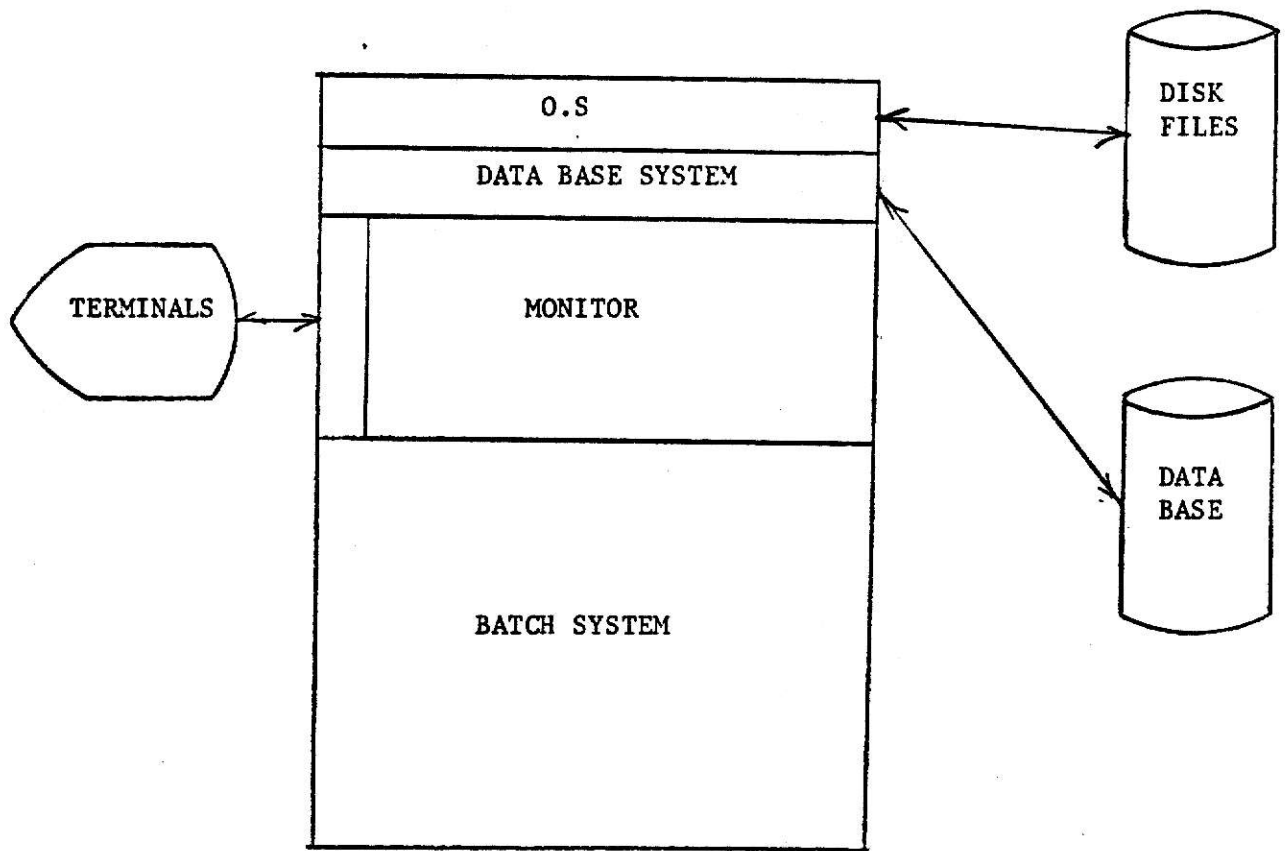


Figure 2.1 Basic System (Model I)

The data base management system model is a memory resident, re-entrant system which services both the monitor and the batch jobs. The amount of memory required for this system is assumed to be negligible compared to the batch and monitor requirements. There are two channels available for disk I/O and 12 disk drives for the data base. The channels may be shared among all the disk units.

The batch system is allocated 1.5 MB of memory. Within this area up to 15 jobs may be run concurrently and all jobs have equal priority when requesting system resources. The job mix is assumed to include both compute-bound and I/O-bound jobs.

2.5 FRONT-END MONITOR (II)

Figure 2.2 shows the first variation of the basic system. This system uses an additional CPU as a front-end to handle the communications monitor. All page swapping, checkpoint recording and communications overhead will be handled by this CPU. Data base requests must be sent to the main CPU for processing.

The advantages of this configuration would appear to be a gain of 500 KB of memory for the main CPU and additional CPU time for monitor message processing and overhead.

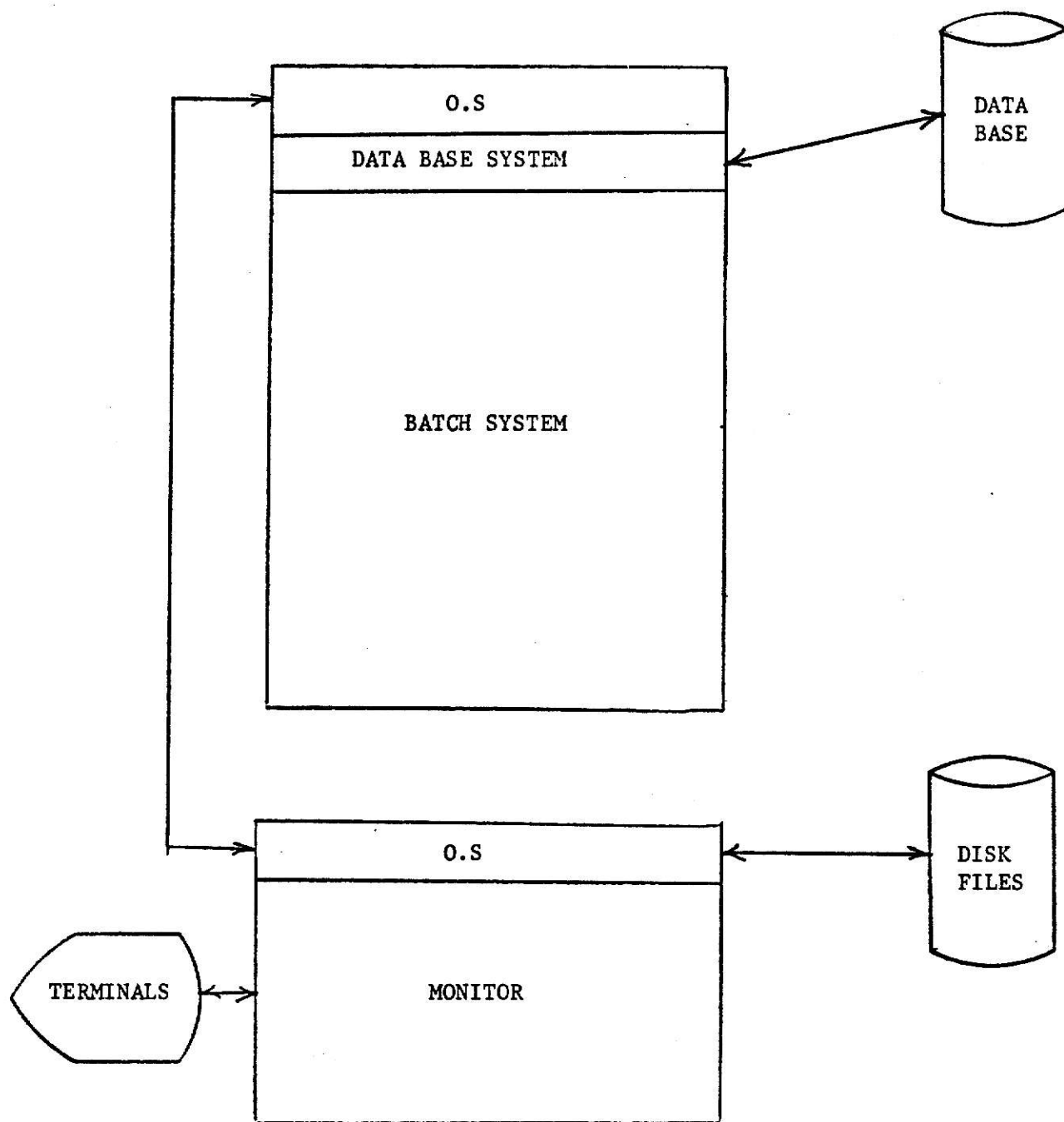


Figure 2.2. Front-end Monitor (Model II)

2.6 BACK-END DBMS (III)

Figure 2.3 shows a back-end data base system. This type of system can take considerable load off the main CPU. No additional memory is gained in this model but more CPU time for processing DBMS and disk I/O requests should be available. DBMS requests from both the monitor and batch system must be sent to this back-end processor. The gain in throughput should depend on how much of the basic system CPU time is used by the DBMS.

2.7 FRONT-END MONITOR/DBMS (IV)

The model shown in Figure 2.4 puts both the monitor and DBMS on an additional CPU with only the batch system on the main CPU. This configuration will provide 500 KB of additional memory to batch but the gain in CPU availability is not immediately clear. There will be additional processing power available but whether it can be utilized will depend on how the CPU load is distributed among the monitor, DBMS and batch systems. The batch system must send all DBMS requests to the additional CPU for processing.

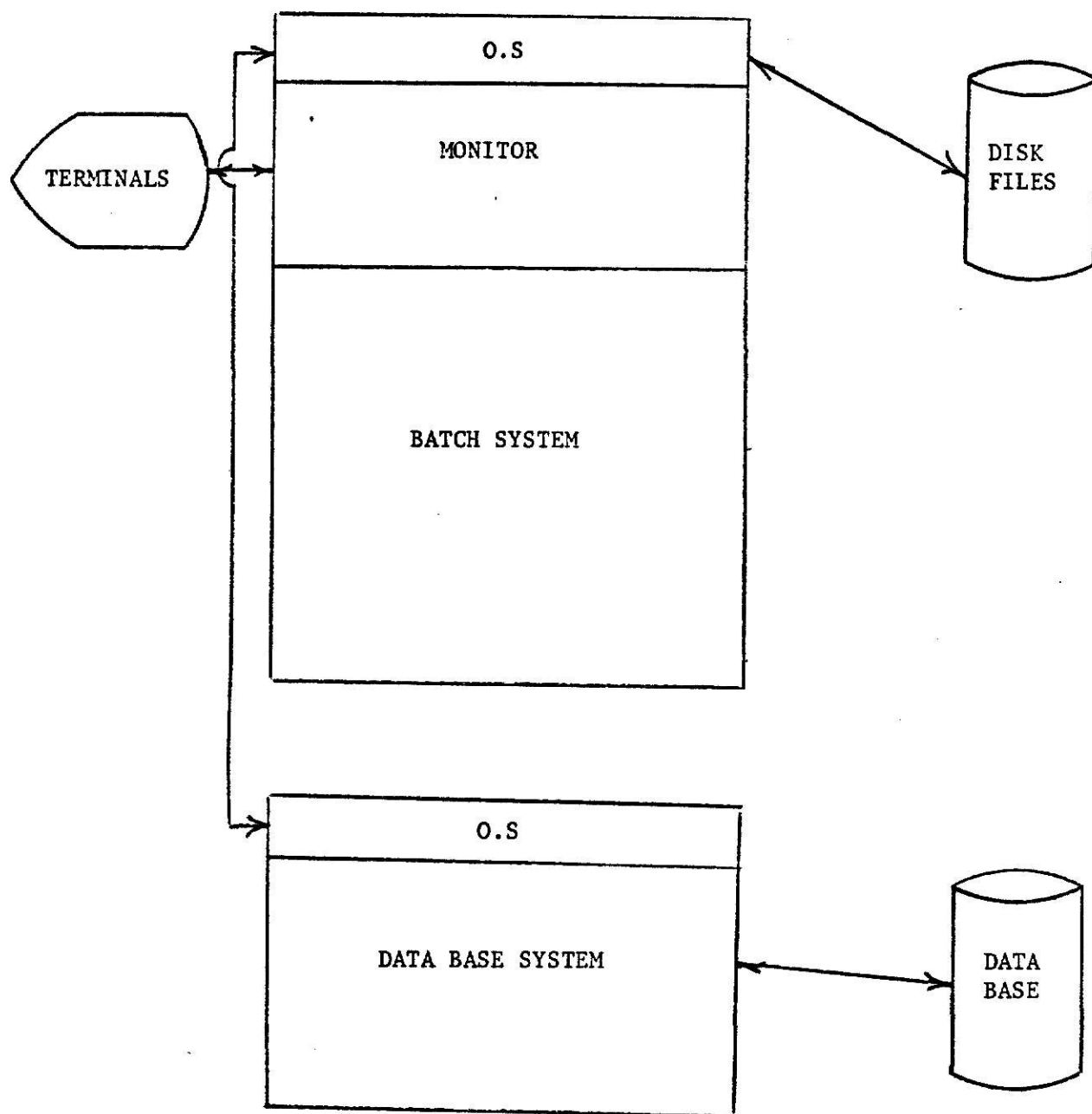


Figure 2.3. Back-end DBMS (Model III)

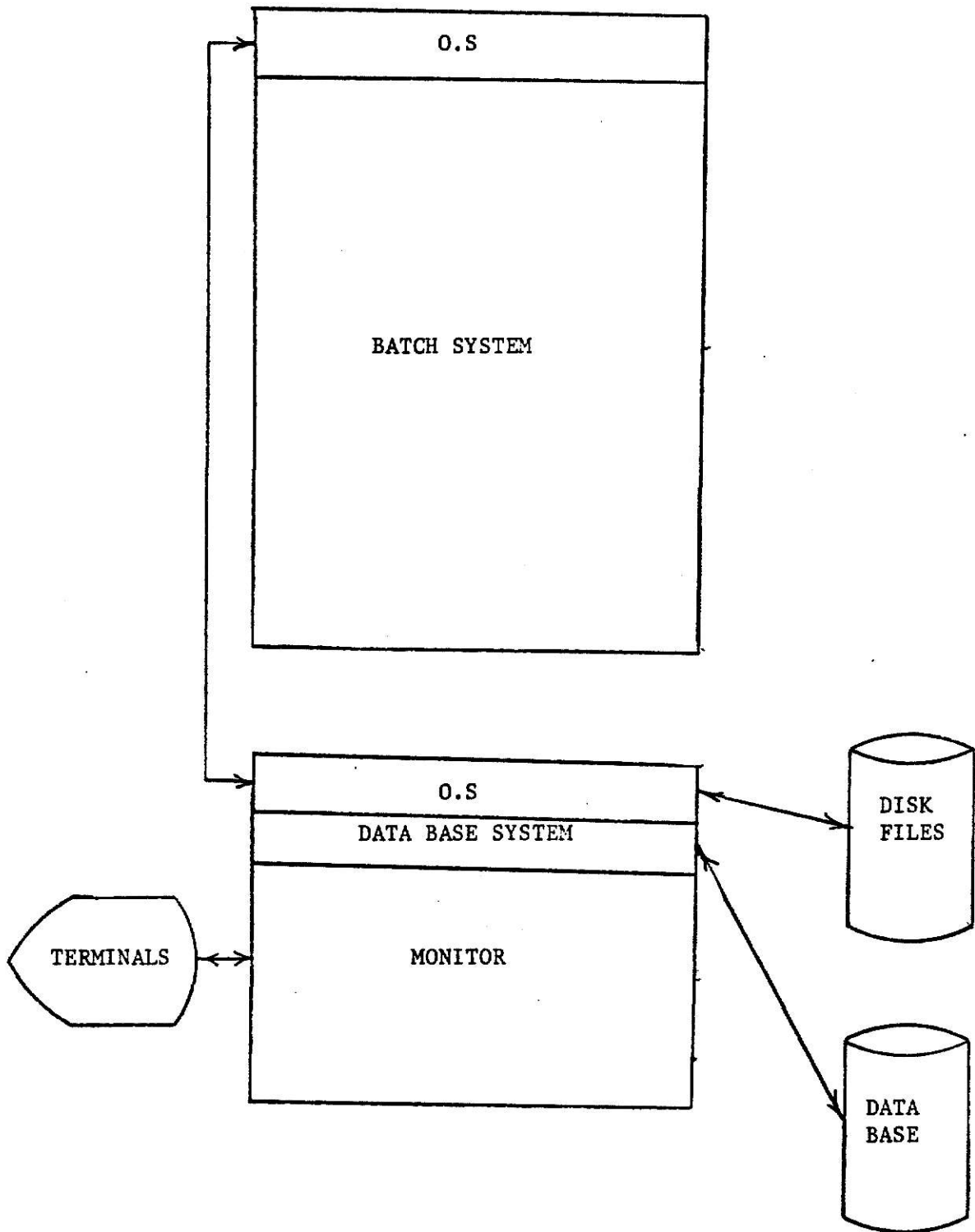


Figure 2.4. Front-end Monitor/DBMS (Model IV)

2.8 FRONT-END MONITOR, BACK-END DBMS (V)

Figure 2.5 shows a model which includes two additional CPU's. The second CPU is used as a front-end for the monitor and the third CPU is a back-end for the DBMS. There will be an additional 500 KB of memory available for batch and each major system will have its own CPU. As in configuration IV, the gain in throughput for the total system will depend on the demand for resources of each major system.

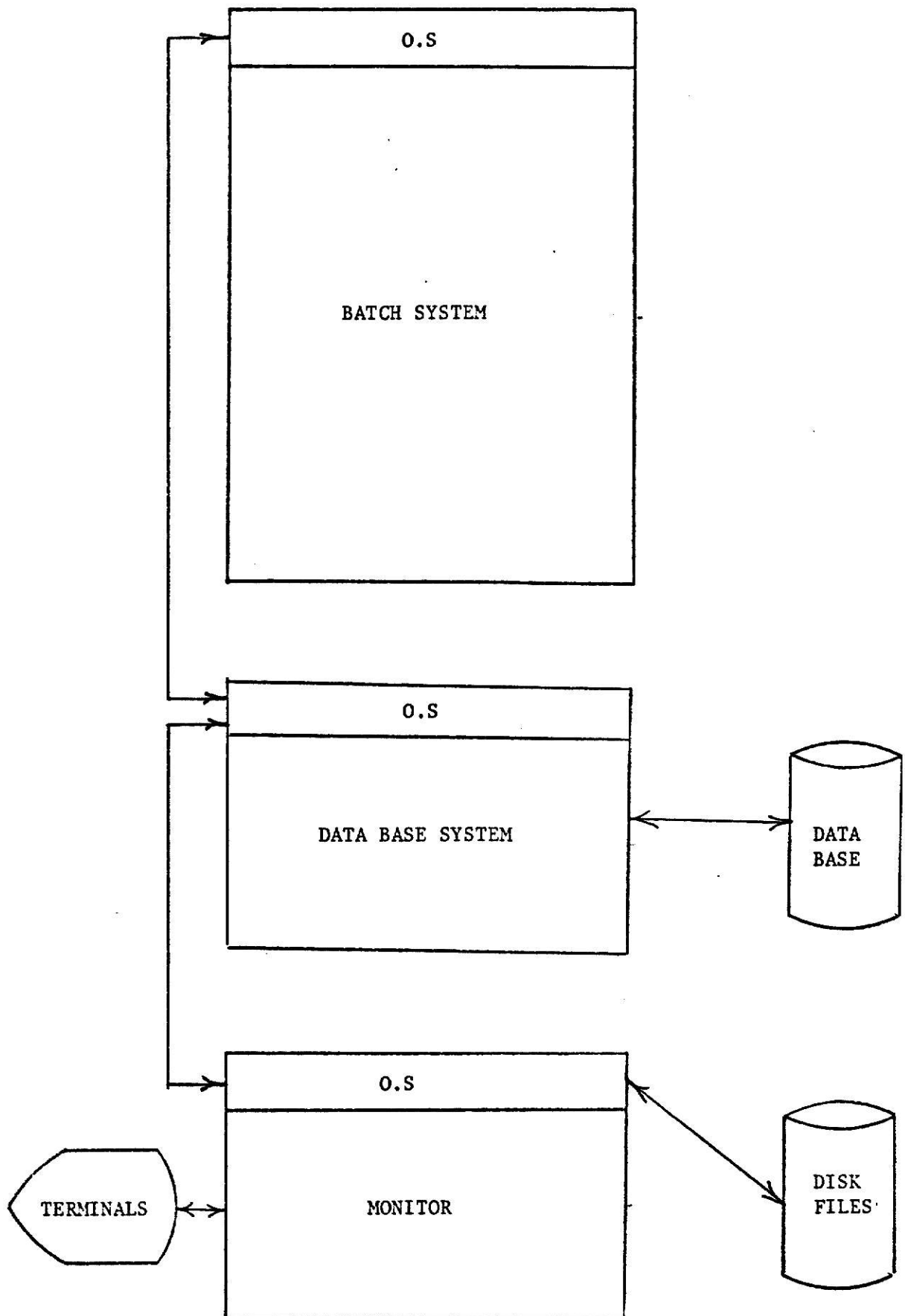


Figure 2.5. Front-end Monitor,
Back-end DBMS (Model V)

CHAPTER 3

IMPLEMENTATION

3.1 OVERVIEW

The implementation of the models will now be presented. The general assumptions and parameters used will be discussed and one model will be described in detail. Measurements of throughput in the models will be related to the real systems.

3.2 MODEL DESCRIPTION

Appendices II and IV contain flowcharts and listings of the five models. The systems were modeled in GPSS V and run on an IBM 370/158. The programs require approximately 2 minutes CPU time for 1 minute simulation time. This figure is highly dependent on program parameters. A simulation time of 1 minute was used for all runs. This time is long enough to get good random distributions and a reasonable measurement of throughput.

Each model has three sections; the monitor, the batch system and the data base system. A separate CPU scheduler is used for each CPU in the particular system. The unit of simulation time is 1 msec. This was chosen as a compromise so that both the fast CPU and the slow terminal could be studied in the same model. The primary considerations are

CPU, memory, and data base utilization and batch and monitor throughput. A 1 msec time unit is sufficient to get a good measurement of each of these items.

All five programs are similar. The change from one configuration to another is an additional CPU and/or additional memory and a delay in sending data base requests to/from a back-end or front-end processor. Therefore only Model II, the front-end monitor, is chosen for detailed explanation below.

3.3 COMMUNICATIONS MONITOR

The monitor could be modeled in detail but a paging system is very complicated and difficult to model and probably should be evaluated in a separate study. We have used parameters obtained from the NETISA study to make a simplified model of this monitor. However, it should be accurate enough to give good results. The monitor can be regarded as a batch system which executes one job (message) at a time. The jobs have a very small CPU requirement (compared to real batch jobs) and a high I/O requirement because of paging and data base requests. All communications overhead is assumed to be handled by a programmable controller and not reflected in this model.

As shown in Appendix II, terminal messages (transactions) are generated at the rate of one message per terminal every 40 seconds. There are 100 simulated terminals. The messages are assigned the following parameters and queued at the input to the monitor:

1) Number of context blocks needed to swap in. These blocks hold the non-re-entrant data for each terminal. An average of 10 blocks is used.

2) Total CPU processing time needed for this message. An average of 10 msec or 400 msec is used in separate runs.

3) Total page swaps needed during this message processing. An average of 1 swap is used but this is highly dependent on whether the terminal users are accessing different parts of the application program. A Cincom study (1) shows that in data base update/retrieval with 10% of the application program in memory, the correct page will be resident 90% of the time. A NETISA study (9) seems to confirm this low swapping rate.

4) Total data base I/O's needed to process this message. An average of 5 is used.

5) An average output message length of 200 characters is used with a line speed of 4800 Baud.

All the above parameters were obtained from the NETISA study.

When a message gets control of the monitor it must swap context blocks and do page swaps. A separate swapping disk is assumed for these operations. The message must then hold

the CPU for some compute time and perform the assigned number of data base I/O 's. In configuration II, the monitor must send the DBMS request to the main processor and receive a reply back. This requires 2 msec transmission time. The message transaction has priority over batch when requesting a facility such as the CPU or DBMS. This is done to insure fast terminal response time.

When the message has been processed, a simulated reply is sent to the terminal, the message transit time is tabulated and the transaction is terminated. The number of messages processed in a given time serves as a measure of monitor throughput.

The monitor can be viewed as a program which requires the following system resources:

- 1) 500 KB of main memory;
- 2) Some small amount of CPU time (≤ 500 msec) per message;
- 3) A large amount of disk I/O (data base and swapping).

3.4 BATCH SYSTEM

The batch model is not written to simulate a real batch job stream. It is intended to maximize the usage of system resources and provide a measure of the potential throughput of the various configurations. Parameters are adjusted on the various runs to correspond to different job mixes (i.e. compute-bound or I/O-bound jobs). If parameters from a real system were available, they could also be used.

The model generates 20 batch jobs (transactions) which

are placed in a job queue awaiting memory allocation. The average job size is such that approximately half of the jobs are on the queue and half in memory at any time. This results in high utilization of available memory and CPU resources. Each job is assigned the following parameters:

1) Total CPU time needed. An average of 250 msec per job is used. This figure is very small compared to a true batch environment but a large number of short jobs can utilize the system resources to the same degree as longer jobs. Since many short jobs will complete in the 1 minute simulation time, the number of jobs completed will give a good measure of batch throughput.

2) Total memory needed. An average partition size of 300 KB is used. This number was experimentally determined by varying the job size (i.e. degree of multiprogramming) until the maximum throughput was obtained. This partition size was found to be optimum for both I/O-bound and compute-bound jobs.

A real batch job stream would be more unpredictable than this and again real parameters could be used if available. However, since we are trying to maximize the use of the system resources, this method will provide very high memory utilization. Good comparisons can then be made between configurations with different amounts of memory.

3) Time between DBMS requests. This parameter was set