Real-time Data-processing Systems

A METHODOLOGY FOR DESIGN AND COST/PERFORMANCE ANALYSIS

SAUL STIMLER

Manager, Advanced Systems Analysis Radio Corporation of America, Cherry Hill, New Jersey

McGRAW-HILL BOOK COMPANY

New York San Francisco Toronto London Sydney

REAL-TIME DATA-PROCESSING SYSTEMS

 Library of Congress Catalog Card Number 68-28422

61480

234567890 MAMM 7543210

Real-time Data-processing Systems

Preface

This book is designed for data-processing practitioners who are interested in calculating and optimizing the performance per dollar of real-time systems typified by message-switching and automated airline-reservation systems. The main objectives of this book are:

- To introduce the data-processing practitioner with batch-processing experience to the operation of real-time systems.
- To present a practical and broadly applicable methodology which the practitioner can use to calculate and optimize the performance and performance per dollar of his real-time system.
- To illustrate the method by applying it.

The degree of success achievable in the calculation of meaningful numerical measures for the performance per dollar of real-time systems does and will probably continue to depend upon the mixture of inspiration, perspiration, knowledge, and experience brought to bear on the problem. This book contains no substitute for any of these necessary ingredients. I hope it will, however, shorten and make easier the acquiring of real-time-system knowledge and experience. To help achieve this objective, an effort was made to omit material which did not bear directly on the subject matter. Photographs of equipment, for example, are not included.

I developed most of the material in this book over an eight-year period. It reflects the tangible and intangible influence of many of my associates at the International Telephone and Telegraph Company's Federal Laboratories and The Radio Corporation of America. I am particularly indebted to Mr. Henry O'Conner, who developed the dynamic tables and application-program estimates used in Chapter 3; Dr. K. A. Brons, who was the mathematical consultant for the book, developed Appendix C, and made invaluable contributions in probability and queuing problems; Mrs. Betty Ringleb, who tirelessly typed subscripts, superscripts, and parentheses to generate the drafts; and A. G. Daubert, who encouraged me onward during the difficult initial phases.

Most of all I am indebted to my wife Ellie and my children Mark and Lynn, without whose understanding and willingness to give up much time together, this book would not have been possible.

S. Stimler

Contents

CHAPTER ONE

Introduction

1.1 Chapter in brief

The goal of this chapter is to describe and place in perspective the subject, objectives, approach, and results of the book. One objective of this book is to present a methodology which will help the data-processing practitioner reduce the time and effort he expends in the calculation of meaningful numerical values for, and the optimization of, performance, cost, and performance per dollar. To help meet this objective, special attention is given to carefully defining technical terms and criteria for performance and cost, and explicitly stating assumptions. In addition, technical terms, their definitions, and the equations in which they are used are listed in Appendix A. It is highly improbable that a set of equations can be developed which would be practically applicable to the wide variety of different hardware-software and applications available today or to those of the future. Therefore, the effort is concentrated on developing a broadly applicable methodology which will help develop cost and performance relationships for each application, identify and use those factors in each application which significantly affect cost and performance, as well as identify and disregard those factors which do not. To aid in the presentation of the methodology, the design of a message-switching system is started in Chap. 2 and carried on in Chaps. 3 to 5. A more difficult automated stock brokerage system within which three real-time programs are operating in real time is developed in Chaps. 7 to 9. The results developed in each chapter are presented and described at the beginning of the chapter. The derivation of the results is carried out in the remainder of the chapter and the appendixes. This organization is intended to give the reader an overview of each topic before getting into the detailed examination. With the exception of some probability and queuing relationships, the book is self-contained. At least one point, usually labeled T, is calculated in detail for every graph and table presented. The reader may calculate another point on the graph when he wishes to try a practice calculation. This method can be applied very usefully during the early stages of a design when such factors as traffic, file organization, program running time, and terminal device type are usually not well defined. Calculation of performance within ± 30 percent of that finally achieved is considered excellent at this stage, ± 50 percent is very useful. The relative performance calculated while varying different system parameters is usually well within ± 30 percent and is particularly valuable in setting the direction of the design. The extensive use of graphs in the book is intended to help the system designer and evaluator to understand better the trade-offs available.

1.2 A first look at the real-time system analyzed in this book

Figure 1.1 illustrates a typical configuration of the real-time system considered in this book. The system, usually classified as *commercial*, consists of three major subsystems, the *processor subsystem*, the *communications network*, and the *terminal subsystem*. The processor subsystem includes the processor, peripheral equipments, software, and application programs. The communications network contains all the communication channels and channel terminations. The terminal subsystem includes the terminal devices, system representatives, and customers. In operation, a customer may, at any time, request a system representative for such information as the availability of a seat on a particular flight, the price of a stock, or the status of his bank account, depending upon the system application. The system representative enters the request into his terminal device in a format acceptable to the system. The request is transmitted to the processor through the communications network. The processor generates and transmits a reply to the system representative who in turn relays the reply to the customer, ending the servicing of

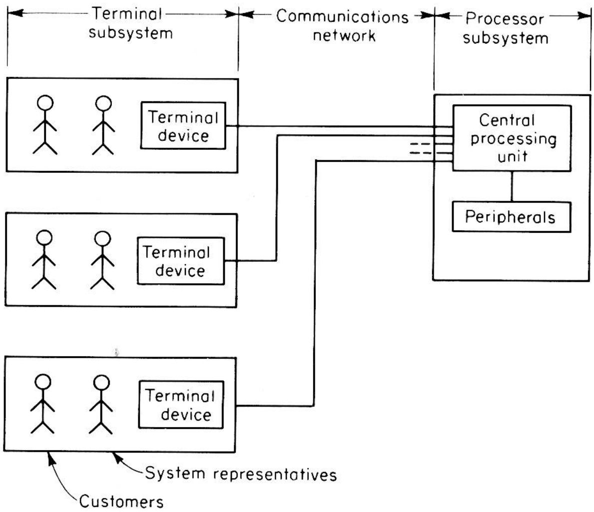

fig. 1.1 *Major components of the real-time system.*

that task. One of the essential aspects of real-time system operation is:

> Each system should be capable of servicing simultaneously up to a maximum specified number of customers, completing each customer service within a time specified for that system.

A large part of this book is devoted to developing methods for calculating the maximum number of customers a system and subsystem can service simultaneously within the service-time constraints specified for that system. A first definition of a real-time system is:

> A real-time system consists of a terminal subsystem, communications network, and processor subsystem operating in such a coordinated manner that up to a specified maximum number of customers may be serviced simultaneously, each receiving service within a specified time.

To permit us to describe system and subsystem performance more precisely, *input message*, *throughput-rate capability*, and *response time* are now described and defined.

An input message is the customer's request for service in a format which can be processed in the processor subsystem.

Figure 1.2 illustrates the general form of the input message which consists of a header, text, and *end of message* (EOM). The header and end of message are for ease of manipulation and bookkeeping in

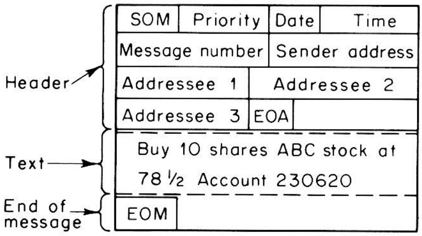

fig. 1.2 *Outline of message format.*

SOM = start of message
EOA = end of address
(This string of characters identifies the end of the header portion of the message.)

the processor. Typically, the header contains a specific string of characters to indicate such information as the *start of message* (SOM), address of the sender, address(es) of the addressee(s), the date, and the message number. The text section contains the customer's request placed in the format required by the processor. The end of message is a string of characters signifying the end of that input message.

Throughput-rate capability is the maximum number of fixed-length-input messages arriving at a uniform input rate* that the system or subsystem can completely service per hour in a no-error environment.

An input message may require that replies be sent to one or more addressees. *One-message throughput* is the receipt and processing of the one-input message and the generation and delivery of replies to all addressees.

Response time is the time between entering the last character of an input message and receipt of the first character of a reply.

The *gross-timing diagram* of an automated stock brokerage system, Fig. 1.3, illustrates the relationship between throughput-rate capability, response time, and system operation. Assume this system has only one broker and one terminal device active during the entire period. A customer starts his transaction with the broker at t_0. The broker receives the entire request 20 sec later at t_1. He has entered the input message and just initiated transmission to the processor at t_2, 40 sec after the start of the transaction. Forty seconds later at t_3 the broker receives the first character of the processor reply. Twenty seconds later at t_4, the broker starts giving the reply to the customer. One hundred and twenty seconds after the start of the transaction at t_5 the broker com-

* This is equivalent to the maximum throughput rate achievable if the system were operating in a batch-processing mode with all input messages sequentially located on one magnetic tape and all output messages written to another magnetic tape, the tapes being fast enough not to limit the throughput. Response time is not a consideration in throughput-rate capability.

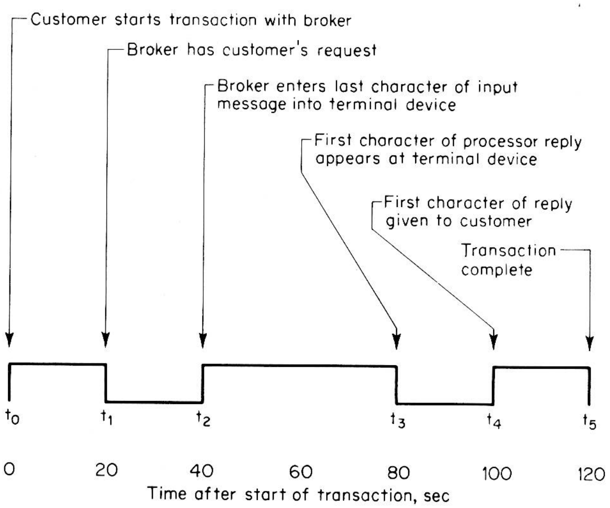

fig. 1.3 *Typical gross-system-timing diagram.*

pletes this transaction and starts the next transaction. The throughput-rate capability of this one-broker–one-terminal system is 30 messages per hr.*

The response time of the terminal device, communications network, and processor subsystem from the broker's viewpoint is $t_3 - t_2$, which equals 40 sec. From the customer's viewpoint the response time of the broker is $t_4 - t_1$ which equals 80 sec.

The terminal device, communications network, and processor subsystem can process a message in the 40-sec period between t_2 and t_3. Therefore, the throughput-rate capability S_M of this portion of the system is 90 messages per hr,† which is three times the 30-message-per-hr throughput-rate capability for the overall system with one terminal device. One way to use the system more fully is to connect three terminal devices to the communications network as shown in Fig. 1.1.

In broad terms, the approach to calculating and optimizing performance and performance per dollar of such systems consists of:

- Establishing criteria for cost and performance
- Developing a generally applicable methodology for:
  - Calculating both cost and performance
  - Developing cost/performance trade-offs
  - Optimizing the cost/performance ratio

* (3,600 sec per hr)/(120 sec per message throughput) = 30 messages throughput per hr.

† (3,600 sec per hr)/(40 sec per message throughput) = 90 messages per hr.

In applying the method, cost/performance trade-off graphs are prepared for each of the three subsystems assuming each subsystem is acting alone and is the limiting subsystem. The cost/performance trade-off for each of the subsystems is then used as input to develop cost/performance for the overall system. The cost/performance ratio for the overall system is then optimized to meet particular operation specifications. Having looked briefly at the overall system operation, each of the three subsystems are now briefly examined.

1.3 The processor subsystem

The processor subsystem is analyzed in Chaps. 2 to 4. In Chap. 2 the operational models to be used in Chaps. 3 and 4 are developed and described. In Chap. 3 the method of calculating throughput-rate capability is developed. In Chap. 4 the effects of nonuniform input rates and variable message lengths are considered. The method is illustrated using a message-switching system. The random-access device can have a significant effect on the processor-subsystem performance; therefore, a solid-state, fixed-head drum and movable-head disk are analyzed as random-access devices. In a message-switching system the text received in the input message is transmitted in the output messages. The processor operates on the header and end-of-message parts of an input message but not on the text portion. In other types of systems the processor generally operates on all portions of the message. For example, in an automated stock brokerage system, not only must the header and end-of-message portion be processed but the text must be processed to determine what type of stock transaction is to be carried out for what account. This usually requires additional accesses to the random-access device and additional processing over that required in the message-switching system. From a methodology viewpoint it is an easy extension* from the message-switching-system analysis to those applications requiring operation on the text portion as well as the header and end of message. For this reason the message-switching system is used in the illustrative examples in Chaps. 2 to 4, and the automated stock brokerage system is considered in Chaps. 7 to 9.

Figure 1.4 illustrates the processor-subsystem components which are important from a system-analysis viewpoint. The processor is assumed to have the capability of interleaving hardware-controlled transfer of characters from one or more peripheral equipments† with instruction execution. Viewed over a period such as seconds, the subsystem appears

* In practice the extension usually requires a more detailed analysis and development of the file structure and file-access method, which is not easy.

† Communication lines are here included in "peripherals."

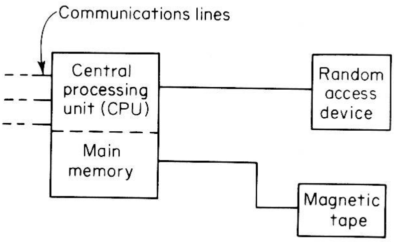

fig. 1.4 *Processor subsystem components.*

to transfer data between main memory and one or more peripherals and to execute instructions "simultaneously." In addition, the subsystem is assumed to be so designed that after a preset number, or after a particular set of characters is transferred between main memory and a peripheral device, the peripheral-device controller can signal the processor that the transfer has been completed by generating an *interrupt signal* or *interrupt* in the processor. The operation in process is interrupted and control is given to the softwave system,* which determines the cause of the interrupt. After service of the interrupt is completed, the next operation in the system may be a continuation of the operation interrupted or a new operation depending upon the preset schedule of priorities. This type of operation, where changes in operations are initiated by interrupts, is frequently called *interrupt-driven operation.* The methodology, as applied to the processor subsystem, is now outlined below.

Outline of processor-subsystem analysis method

1. Develop an operating model for the system in sufficient detail to calculate the software, application-program, and peripheral equipment time per message throughput.
2. Assume real-time-only operation.
3. Assume no-error operation with fixed-length messages arriving at a uniform input rate.
4. Develop an equation relating average time per message throughput to the software time, application-program time, and peripheral-equipment time per message throughput.
5. Use the average time per message throughput to calculate throughput-rate capability S_M.
6. Prepare throughput-rate-capability trade-offs.

Typical trade-off parameters are those which relate throughput-rate capability to the size of main memory required and type of peripheral equipment needed. At this point response time is not a trade-off parameter.

* Hardware interrupt systems are considered a special case of the software interrupt handling system and are not considered.