2

COMPUTER DESIGN AND ARCHITECTURE SERIES

Computer Performance Measurement and Evaluation Methods: Analysis and Applications

Liba Svobodova
Massachusetts Institute of Technology

ELSEVIER
New York/Oxford/Amsterdam

Computer Performance Measurement and Evaluation Methods: Analysis and Applications

ELSEVIER COMPUTER SCIENCE LIBRARY

Operating and Programming Systems Series Peter J. Denning, *Editor*

1. *Halstead*
   A Laboratory Manual for Compiler and Operating System Implementation

Spirn
Program Behavior: Models and Measurement (in prep.)

Programming Languages Series Thomas E. Cheatham, *Editor*

1. *Heindel and Roberto*
   LANG-PAK—An Interactive Language Design System
2. *Wulf et al.*
   The Design of an Optimizing Compiler
3. *Maurer*
   The Programmer's Introduction to SNOBOL

Cleaveland and Uzgalis
Grammars for Programming Languages (in prep.)
Hecht
Global Code Improvement (in prep.)

Theory of Computation Series Patrick C. Fischer, *Editor*

1. *Borodin and Munro*
   The Computational Complexity of Algebraic and Numeric Problems

Computer Design and Architecture Series Edward J. McCluskey, *Editor*

1. *Salisbury*
   Microprogrammable Computer Architectures
2. *Svobodova*
   Computer Performance Measurement and Evaluation Methods: Analysis and Application

Wakerly
Error-Detecting Codes, Self-Checking Circuits and Applications (in prep.)

Artificial Intelligence Series Nils J. Nilsson, *Editor*

1. *Sussman*
   A Computer Model of Skill Acquisition
2. *Shortliffe*
   Computer-Based Medical Consultations: MYCIN

Contents

Preface

Analysis and evaluation of computer system performance has become an important and demanding field. Important because performance is one of the prime considerations in evaluating a computer system. Demanding because it requires a deep understanding of the inner system mechanisms, both hardware and software, knowledge of the processing requirements, understanding of the user's habits, adaptability to system changes and preferences, and the effects of poor performance on the user's productivity and well-being. As a result of demand for more powerful and flexible computing services and better performance per cost, computer systems have become increasingly complex, and system performance increasingly difficult to assess. Performance evaluation it not just determining whether or not a system meets certain objectives; it is also understanding if and how system performance can be improved. A computer system analyst must master a number of techniques to ascertain important factors and their effect on system performance. In the vast literature on computer performance and performance evaluation, only a few works discuss the relative utility of different techniques, their proper application and inter-relation. Thus, it is hard to acquire the needed information on applicable methods and tools.

The purpose of this book is to develop a better understanding of the problem of performance evaluation and to analyze available techniques within this concept. The book is directed to present and future computer analysts and designers. The readers are assumed to be familiar with concepts of hardware organization, system architecture, and operating systems. The readers are exposed to many different ideas and trends, but the level of detail is kept relatively low; for details, the readers are directed to appropriate publications.

This work is a result of a three-year experience in measuring and analyzing the performance of a developing system as well as teaching the subject at Stanford and Columbia Universities. It is based on my dissertation completed in June 1974 carrying the same name. However, after an additional year and a half of research and teaching, the original material was largely restructured and enriched with new ideas. While not intended to be a textbook, this book can nevertheless serve as a basic text for a self-contained course or seminar on computer performance evaluation, supplemented from the numerous references where more depth is desirable. At Stanford, the material was concentrated into a four-week course offered as a part of a special summer program to qualified people from outside companies. At Columbia University, the material was used in a one-semester graduate seminar with summary lectures supplemented by students' presentations of selected papers.

The book has two parts. The first part presents the fundamental concepts (Chapters 1 and 2); the second part examines different techniques and tools in detail (Chapters 3 to 6). Chapter 1 presents a brief overview of the need for performance evaluation, describes the levels of a computer system, and defines the boundary between the system and the system environment. Chapter 2 defines and examines the performance evaluation problem: specification of system workload, selection of performance measures, determination of the quantitative values of performance measures, and evaluation of performance. Chapter 3 is dedicated to system models. A functional model provides a framework for system analysis; a performance model describes the effect of the system structure and the system workload on system performance. A performance model can be obtained analytically, by simulation, or by direct measurement. Chapter 4 discusses workload models for measurement and simulation experiments. Chapter 5 looks at simulation techniques. Chapter 6 evaluates measurement techniques and tools. Chapter 7 deals with the problems of computer measurability and performance control. A case study of system instrumentation for measuring CPU utilization profile, multiprogramming effect, and utilization and efficiency of program modules is presented in the Appendix. Each chapter has its own bibliography that includes selected articles, reports, and books relevant to the covered subject.

I would like to express my deep gratitude to the editor of this series, Professor E.J. McCluskey of Stanford University, who suggested a publication of this material in book form. His encouragement and constructive comments were always greatly appreciated. Many valuable comments and suggestions came from Professor Forest Baskett of Stanford University. I am also indebted to my students who helped me to clarify many of the concepts presented here. I owe many thanks to Joanne Knowlton for her excellent typing job and cheerful support in the iterative process of polishing the manuscript. Finally, my warm thanks belong to my husband for his understanding and his supportive belief that one day this book will indeed come out.

Chapter 1

Introduction

> In the history of science, there are two converging avenues along which flows the potential of progress: the avenue of ideas and the avenue of techniques. It is the confluence of these that has made possible the marvels of modern civilization.
>
> P.P. Schodeberk
> *Management Systems*

Computer performance evaluation is a complex process that involves many analyses and decisions. At the very root of the problem lies the question: What are the legitimate measures of performance? Once these measures are specified, it is then necessary to decide what techniques to use in order to determine their values, analyze what factors have the greatest effect on performance, and qualitatively assess performance given the values of performance measures.

The computer as a system can be analyzed and evaluated by standard techniques used to analyze and evaluate systems in other fields of science and engineering. Such techniques include simulation, queuing theory, statistical methods of sampling, estimation, regression analysis, experimental design and hypothesis testing, optimization techniques, control theory, decision making methods, etc. The true information about a system's behavior in a given environment can only be obtained by measurement. Without measurement, no performance hypothesis can be fully validated. Measurement is thus viewed as a fundamental technique.

This book is concerned mainly with the effectiveness of various techniques and tools available to a computer performance evaluator. Since, like other disciplines, the discipline of computer hardware and software engineering had to develop its own measurement techniques and tools, special attention will be given in this book to the design of measurement tools and system instrumentation.

1.1 NEED FOR PERFORMANCE EVALUATION

The development of computer performance evaluation as a separate discipline is a natural result following the development of powerful and complex computer systems.

The early computers were designed to be operated by the programmer himself, who in fact could actually watch what was going on in the computer during program execution. In the early days of computers software was practically nonexistent, and the fundamental design decisions were concerned with the word size together with the composition and implementation of the machine instruction set. Computer evaluation and comparison were based on such parameters as the CPU cycle time and the execution times of the basic instructions (usually the ADD instruction). Various software aids such as assemblers and compilers only became feasible with the subsequent development of larger and less expensive memories. Together with better peripherals, this led to the introduction of batch processing controlled by a resident supervisor. Software was discovered to have a great impact on the ease of programming. Mere instruction speeds and instruction repertoire could not sufficiently describe qualities of new computer systems. Software characteristics had to be taken into consideration too. Together the system overhead, the compile speed, and the execution speed determined how long a specific program would take to run. To make programming easier was not the only reason behind the efforts in software development. As more powerful (and more expensive) hardware became available, greater pressure was applied to increase the efficiency of computer operations and to achieve better utilization of various computer system units. Independent I/O channels communicating with the central processor via an interrupt system allowed concurrent usage of several machine resources (overlap of CPU operations with I/O operations). The amount of overlap was not easily detectable, yet it was an important parameter in determining system throughput. Multiprogramming further increased the chance of good system utilization, but it became increasingly difficult to understand and to follow what was happening inside the system. The question 'How long does it take to execute a given job' no longer had a simple answer. Job execution time under a heavy over-all system load could be several times that achieved when such a job was the only one in the system. More had to be learned about internal system operations, such as scheduling algorithms, resource allocation policies, etc. Time-sharing, interactive processing and real-time processing only enhanced the need for further analysis.

Early performance evaluation studies were concerned with comparing the capabilities of different computers. This still remains one of the major purposes of performance evaluation, but such a study now requires more sophisticated methods. Because of the large investments in installed computer equipment and the high cost of running a computer center, today considerable attention is paid to the possibilities of increasing system efficiency through hardware configuration changes and software changes. The high cost of software development and hardware design creates pressure for evaluating proposed systems and system changes before the actual implementation begins.

The frequently cited classification scheme by Lucas [LUCA71] divides performance evaluation undertakings into selection evaluation, performance monitoring, and performance projection. Lucas' classification scheme closely reflects the reasons given in the preceding paragraph. For a discussion of *methods* of performance evaluation, it is generally better to use a classification scheme that differentiates between comparative evaluation and analytic evaluation. These differing approaches are defined as follows.

1. **Comparative Evaluation** Performance of a particular computer system is evaluated relative to the performance of another computer system. The purpose of such an evaluation may be to:

(a) Lease or purchase new hardware and software,
(b) Select a supplier of computing services,
(c) Classify existing systems,
(d) Evaluate changes in system hardware or software (i.e., compare performance of the modified system with performance of the system prior to modification).

2. **Analytic Evaluation** Performance of a computer system is evaluated with respect to various system parameters and system workload. The purpose of such evaluation may be to:

(a) Improve performance of an existing system (performance tuning),

(b) Maintain performance of an operating system within specified limits (performance control),

(c) Design and implement new systems.

1.2 COMPUTER AS A MULTI-LEVEL SYSTEM

A system is a structure that is composed of elements connected according to some interconnection rules. An element of a system can itself in turn be a system that is composed of some lower-level elements. It is necessary to decide a level at which to analyze the system. The elements of a system are then those systems that are considered to be indivisible at the chosen level.

Computer hardware can be discussed on several levels [BELC71]: the circuit level, the register-transfer (RT) level, the programming level, and the processor memory switch (PMS) level. The elements of the RT level are circuits representing registers, shifters, adders, etc. The systems arising from this level are controllers, processors, and interfaces. The programming level, sometimes called the instruction set processor (ISP) level, adds to the RT system the capability of executing instructions from a stored program. Finally, the elements of the PMS level are complex hardware units such as processors, memories, controllers, and switches. These elements are linked together by switches that govern flows of information.

For a user who is interested in the computer as a problem-solving tool and data processing utility, bare hardware does not represent a complete system. The desired services are provided by the operating software (OS) level that is an extension to the hardware hierarchy. The quality of services is, of course, influenced by the hardware system. Hardware speed limits the speed with which users' requests can be processed and indirectly imposes boundaries on what services are feasible. The ISP and PMS levels provide an interface between the hardware that does the actual work and the software that maps different applications into executable tasks. The operating software is again a hierarchy of several levels [TSIC74]. The kernel of the operating software controls the PMS switches. Additional levels of the operating software add progressively more power and flexibility:

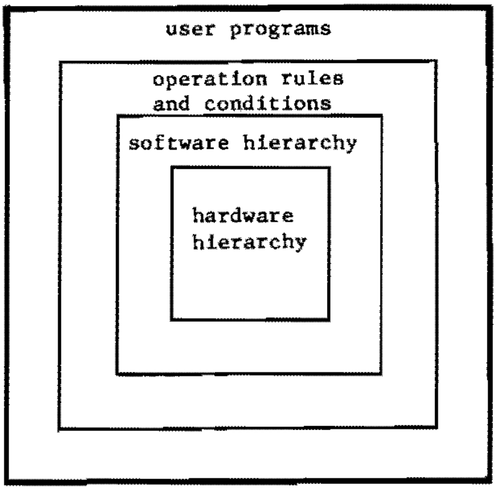

Figure 1.1. Levels of the information processing system.

automatic scheduling and resource allocation, data base management, compiling, linking, and loading of programs, command structure. The highest level of consideration is the level at which the user communicates with the system. At this level, the system elements are the processes that carry out the functions requested by the user.

Performance of a system is determined by the performance of individual system elements and the way these elements are connected into a system. Thus, any and all of the described levels contribute toward the performance seen by the user. In order to meet specific performance objectives, all of these levels have to be taken into consideration in an evaluation of performance.

In addition to specifying the level at which the system is to be evaluated, it is necessary to make a clear distinction between the system and its environment, that is, to define the system boundary.[1] For example, the system may be defined merely as the hardware operating in the environment created by the software. Or the system of interest may be a specific hardware or software subsystem operating in the environment of other hardware and software subsystems

[1] The system boundary can change during the course of system evaluation. An instructive example can be found in [STIM74], where the boundary is called the evaluation interface.

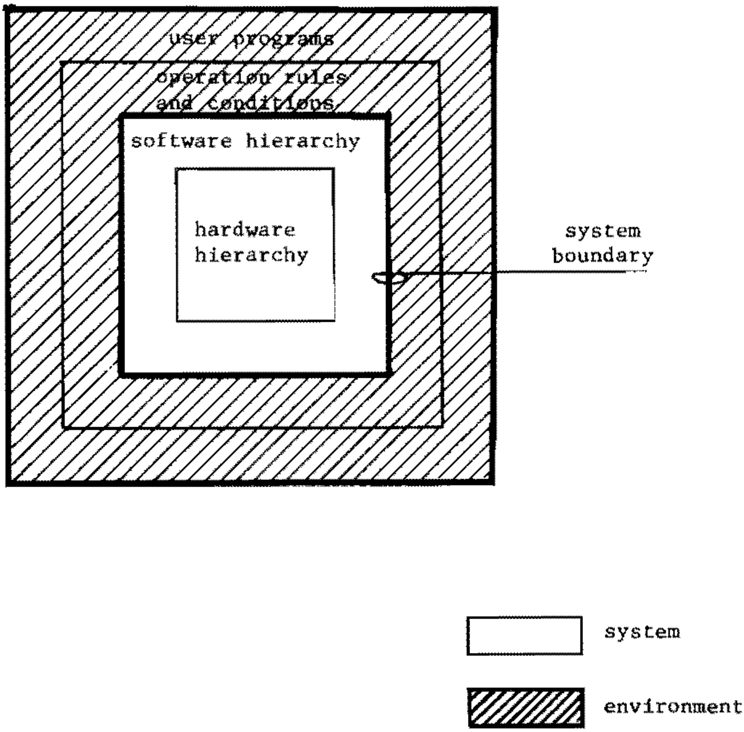

Figure 1.2. Evaluated system and its environment.

with which it interacts. Or the system can be perceived as a total complex organization of hardware, software, user programs, users, programmers and operators, together with the operation rules and conditions governing the interactions of the human with the non-human elements (job submitting policies, handling of tapes and disk packs requested by a job, equipment layout, etc.). The system hierarchy has to be thus extended by additional levels as shown in figure 1.1. This complete system will be called the information processing system. The information processing system is in turn an element of a higher level organization such as a company, a university, or a bank. Unless otherwise specified, the performance of a computer system will be discussed with respect to the user interface as shown in figure 1.2.