7560068

Lecture Notes in Computer Science

Edited by G. Goos, Karlsruhe and J. Hartmanis, Ithaca

8

GI-NTG

Fachtagung Struktur und Betrieb von Rechensystemen

Braunschweig, 20.–22. 3. 1974

Springer-Verlag
Berlin · Heidelberg · New York

TP303-2
S1
1974
TP303-53
G389.2
1974
7560068

Lecture Notes in Computer Science

Edited by G. Goos, Karlsruhe and J. Hartmanis, Ithaca

8

E7560068

GI-NTG

Gesellschaft für Informatik e.V., Fachausschüsse „Rechnerorganisation (3)" und „Betriebssysteme (4)"
Nachrichtentechnische Gesellschaft im VDE, Fachausschuß „Technische Informatik (6)"

Fachtagung Struktur und Betrieb von Rechensystemen

Braunschweig, 20.–22. 3. 1974

Herausgegeben im Auftrag der GI und der NTG von H.-O. Leilich

Springer-Verlag
Berlin · Heidelberg · New York 1974

Editorial Board: D. Gries · P. Brinch Hansen
C. Moler · G. Seegmüller · N. Wirth

Prof. Dr.-Ing. H.-O. Leilich
(wissenschaftlicher Tagungsleiter)
Institut für Datenverarbeitungsanlagen
Technische Universität Braunschweig
33 Braunschweig
Schleinitzstraße 23

Programmausschuß

H. Billing
Max-Planck-Institut für Astrophysik, München

A. Endres
IBM-Labor Böblingen

H. O. Leilich
Technische Universität Braunschweig

W. E. Proebster
IBM Stuttgart

B. Schlender
Universität Kiel

W. Urich
Technische Universität Braunschweig

H. R. Wiehle
Leibniz-Rechenzentrum München

AMS Subject Classifications (1970): 00A10, 68-02, 68A05, 68A10, 68A30, 68A35

ISBN 3-540-06677-2 Springer-Verlag Berlin · Heidelberg · New York
ISBN 0-387-06677-2 Springer-Verlag New York · Heidelberg · Berlin

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks.

Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to the publisher, the amount of the fee to be determined by agreement with the publisher.

© by Springer-Verlag Berlin · Heidelberg 1974. Library of Congress Catalog Card Number 74-233. Printed in Germany.

Offsetdruck: Julius Beltz, Hemsbach/Bergstr.

V O R W O R T

Die Fachausschüsse "Rechnerorganisation" und "Betriebssysteme" der Gesellschaft für Informatik und "Technische Informatik" der Nachrichtentechnischen Gesellschaft im VDE haben bereits zwei gemeinsame Tagungen über ihre weitgehend überlappenden Themenkreise veranstaltet - im Oktober 197o in Erlangen und im April 1972 in Darmstadt. Bei der Vorbereitung der Fachtagung in Braunschweig, deren Vorträge in diesem Band vorliegen, erschien es nach wie vor zweckmäßig, diese Gebiete, die zwischen eigentlicher hardware - der digitalen Schaltungstechnik - und software - der Programmierungs- und Compilertechnik - liegen, gemeinsam zu behandeln. Es ist geplant, diese Fachtagungsserie fortzusetzen.

Die Beiträge aus Universitäten und Industrielabors wurden ausgesucht, um die Gesichtspunkte und Prinzipien der Effizienz von Rechnersystemen vorzustellen, nicht um spezielle Anlagen und Betriebssysteme anzupreisen. Der Rahmen der Fachtagung (ohne Parallelsitzungen) gebot es, eine ganze Reihe von guten Vortragsanmeldungen über andere Teilgebiete der Rechnerstrukturen und Betriebssysteme zurückzustellen, wie z.B. über Entwurfssysteme, Testung und Zuverlässigkeit, die es verdienten, bei anderer Gelegenheit schwerpunktsmäßig behandelt zu werden.

Bei der Tagung werden für jede Themengruppe ausführlichere Übersichts- und Einleitungsreferate von den Sitzungsleitern gegeben, die hier jedoch nicht abgedruckt wurden. Sie sollen den Teilnehmern, die in den einzelnen Gebieten nicht ganz "zuhause" sind, einen aktuellen Überblick geben, so daß sie die Einzelreferate besser einordnen können und so einen größeren Nutzen von der ganzen Tagung haben. Außerdem konnten wir einige renomierte ausländische Gäste gewinnen, zu diesen Themen beizutragen.

Die Themenkreise der vier Sitzungen, die bei dem Entwurf des "Call for Papers" festgelegt wurden, erscheinen nun nach der Auswahl der Beiträge und dem Vorliegen der vollen Referate etwas willkürlich, da sich die Inhalte doch vielfach überlappen. Wir haben dennoch die Leitworte beibehalten, weil wir es für nützlich halten, wenn dem Hörer und dem Leser die inneren Zusammenhänge von hardware-Strukturen und betriebsmäßigen Strategien von verschiedenen Gesichtspunkten aus vor Augen geführt werden.

In der ersten Sitzung ("Rechnerstrukturen") werden Mikroprogrammkonzepte - als Übergang von Schaltkreistechnik zum Strukturentwurf - vorgetragen, dann Vorschläge zum modularen Systemaufbau und für Strukturen von Multiprozessoren, die im Zeitalter der integrierten Halbleiterschaltung sehr aktuell werden. Die drei letzten Vorträge behandeln den Assoziativen Speicher als Systembaustein und neue Aspekte für seine Verwendung.

Der zweite Themenkreis ("Schnittstellen und Leistungskriterien") behandelt die verschiedenen Entwurfsebenen von der hardware bis zur Betriebssoftware. Hier wird der Einfluß von Pufferspeichern wie von Organisationsmethoden auf die Effizienz von Rechenanlagen diskutiert.

In der dritten Sitzung ("Auftragslast, Leistung und Messung") wird versucht, den Begriff der Arbeitsleistung eines Rechensystems zu objektivieren; ferner werden Jobmixe und Benchmark-Tests behandelt.

Das letzte Thema ("Betriebsmittelvergabe") behandelt Modelle und Strategien zur Optimierung des Durchsatzes, ein schwieriges und wichtiges Gebiet der Betriebssystem-Forschung.

Zur Erleichterung der Kontaktaufnahme sind am Ende dieses Bandes die Adressen der Autoren zusammengestellt.

Allen, die am Zustandekommen und Gelingen dieser Tagung und dieses Bandes mitgewirkt haben bzw. noch beitragen werden, sei herzlich gedankt, insbesondere den Autoren, die diesmal alle rechtzeitig ihre Beiträge eingereicht haben, sowie dem Springer-Verlag, der sich Mühe gibt, die Tagungsbände vor Beginn der Sitzungen zur Verfügung zu stellen.

Braunschweig, im Januar 1974

H.O. Leilich

此为试读，需要完整PDF请访问：www.ertongbook.com

INHALTSVERZEICHNIS

BETRIEBSMITTELVERGABE

H A U P T V O R T R A G

PERSPECTIVES ON COMPUTERS: WHAT THEY ARE AND WHAT THEY DO

Robert R. Johnson
Burroughs Corporation
Detroit, Michigan

A Introduction

For the computer industry this year, 1974, is beginning with issues and trends that characterize our state of understanding of computers and their use:

1. Security and Privacy
2. Microprogramming
3. Language-Defined Architecture
4. Performance Measurement
5. Parallel Processing
6. Cost of Programming

These themes thread the papers in this conference, and they have been interwoven in the papers of many conferences this winter (2) (3). It is interesting to note that issues of arithmetic speeds, instruction timing, circuit speeds and detailed logic designs are not in the forefront of the discussions at computer conferences today. Even time-sharing is a bit passe.

Our industry is attaining a certain adolescent maturity. We are beginning to look at the consequences of our past designs. We have had our fling at begetting clever things; we are now settling down, evaluating the good and the bad, and analyzing needs. Simple computers are cheap and plentiful. On one $20 chip today, we can have the equivalent arithmetic power of the LGP-30 computer of twenty years ago; on a few chips, the equivalent of the IBM 1401 computer of 14 years ago. Today, computational horsepower is cheap and available.

As a result, the computer industry has made the transition from a capital-intensive industry to a people-intensive industry; it now costs more to use computers than to buy them. The people around the computers now loom of greater economic importance than do the machines.

At this transition point in our industry, it thus seems appropriate to re-examine what it is that computers do for people. If we are now to use computers to maximize the effectiveness of people, the architectural balances and designs of the past should be re-examined and subjected to our current values. The themes and issues listed initially are all the consequences of this new balance wherein hardware-costs are cheap and people-costs are expensive.

Having asserted that this shift in values has occurred, the outline of this talk is to discuss what it is that computers do for people, and then to show how each of that initial set of current issues is changing what computers are.

B What Do Computers Do?

Most of us would probably say that computers process information. Fine, but what is information? Shoveling "numbers" from one place to another might be "communication", but we all know that "data processing" also involves masticating "numbers" by some preconceived recipes.

The task of building a theory to answer this question, "What do computers do?" is the area of my personal research and it is also the area of prime interest for two of the other invited speakers here, Drs. Hellerman and Muntz. Each of us has his view on how to answer this question, and this conference should prove to be an interesting forum for debate on these approaches.

My approach to answer this question of what computers do was to examine the problems that machines solve for people; and to find out what "information" is, I went back to the classical definition of information (5). Shannon's work in 1948 established the science of information theory and provided a way to represent and measure information. The representation of an information system that Shannon used was a directed graph, and the measure he used was a weighted measure of the probabilities of being at each branch in that graph. The unit of information that he assigned to the normalized, weighted sum of these probabilities was the bit, and the measure he used was the simple sum:

$$H = -\sum_{i} p_i \log p_i \quad \text{where} \quad \sum_{i} p_i = 1 \qquad (1)$$

where H is measured in bits.

The bridge between Shannon's work and computers appeared to me with the information system representation provided by Dr. C. A. Petri at the University of Bonn. This representation, Petri Nets, provides a methodology for depicting most of the important properties of information systems. By applying Shannon's measure to Petri Nets we are able to represent the important aspects of information systems and measure them.

此为试读,需要完整PDF请访问: www.ertongbook.com

Of particular significance is the fact that the information system being represented and measured can be a problem, or a program, or a machine. The representation used to describe the problem that is to be solved is the same representation used to describe the program that solves the problem, and it is also the same representation that can be used to describe the machine that executes that program.

With Shannon's measure applied to the Petri Net descriptions of different programs solving the same problem, it is now possible to apply a numeric method to measure the "information-gain" of these different programming approaches. Similarly, it is now possible to numerically evaluate the "information-gain" of different machine designs for the same program.

The initial assertion of this theory is that it has identified that important parameter which we can identify and measure in information systems. That parameter is the branch in a graph, and the important dimensions of the branch are the number of branches leaving a node, and the transition-probabilities of each branch. If these branch-points are called "decisions", one can say that the one important thing about programs, and the machines that execute programs, is their ability to make these "decisions". It is these "decisions" which produce the "information" which people want as the problem solution.

Thus decision-making is what computers do. The effect of a computer's information-processing can be evaluated by the change in probabilities expected for the output states of the problems that that computer is to solve.

Other parameters of interest are the resources (CPU time, memory space, channel capacity, etc.) used in solving a problem, as well as the response time and execution time required to effect a solution. These other parameters of interest are not yet incorporated into this theory of information-system-performance.

Thus in answer to the question: "What do computers do?" I assert they make "decisions" which give people information. Information is measured by probabilities and the tasks which people give computers are to determine the probable outcomes of those problems that they want analyzed. The information-gain of a computer can now be computed just as the information loss of a communication channel was computed.

This probabilistic answer is not yet proven experimentally nor generally accepted. You will hear other ideas this week on how to represent information systems and how to evaluate what computers do. The exciting thing is that today we have debate and real effort on this problem. The industry has matured to the point where serious theoretical and practical efforts are being directed to characterize that which a computer does, and to make new machines and programs that will do "it" more efficiently.

In addition to the ideas on system representation that you will hear about in this conference, I would like to mention the work of J. B. Johnston (6) on Contour Models; that of C. W. Bachman and J. Bouvard (1) on machine processable function definition algorithms; and that of Kotov (7) on graph-directed program design.

These are other efforts aimed at extracting the structural content of a problem and using that structure as the basis for new systems design.

C New Things Computers are Expected to Do

The topics of contemporary interest listed at the start of this talk can be rephrased in terms of the problems that people need solved. The hardware and software solutions to these problems is what computers will become.

1. <u>Security and Privacy</u>

People want the same information as before, but in addition, they now want to control its accessibility and dissemination.

2. <u>Microprogramming</u>

The application problems that people want to use computers to solve have so many different structures that no single set of machine instructions has been found optimal for mapping all problem solutions into one machine. People want a better means to match problem structure with machine structure. Microprogramming is a means to give people the ability to create different virtual machines whose structures are intended to be better matches to their applicational problems.

3. <u>Language-Defined Architecture</u>

Compiler languages have achieved widespread use for describing problem solutions. Programs written in these languages utilize structures characteristic of each such language. Thus a machine's architecture can be specialized to handle the information structures typically encountered when using a particular compiler language. The people-problem thus is to provide a machine better able to handle the languages that people use in describing problem solutions.

4. Performance Measurement

The basic problem people have with computers is predicting the time and resources needed for any computer to solve any given problem. Subsets of this basic problem are:

a) Tuning an existing machine solution to obtain better performance.

b) Measuring computer performance and analyzing application structure to determine what parameters are important when characterizing performance.

c) Developing theories that depict computer performance.

d) Designing new computers based on these theories that will be better performers and designing computers whose performance will be more predictable.

5. Parallel Processing

The people-problems here are, first, to devise practical means to solve big problems whose structures are somehow made up of parallel elements which offer the prospect of concurrent processing; and second, to devise means to utilize all the resources in a system "in parallel" so that optimal time utilization is made of all those resources.

6. Cost of Programming

The issue of greatest importance to people trying to solve problems using computers to assist them in those solutions is the difficulty of mapping a solution to the problem onto a real computer. This difficulty is measured in:

a) The economic costs of paying people to effect this mapping, i.e., "to program" the computer.

b) The time it takes to get a correct program.

c) The errors and misinterpretations that occur due to the opaqueness of programs which obscures their purpose and obscures the structures intended to achieve that purpose.

d) The problems that aren't solved because of these "costs" of programming.

D The Computers That Are Evolving To Solve These Problems

1. Privacy and Security

The two principle aspects here are, first, legal-social: what do people want to keep private; and second, technical: how to provide the security controls to assure that privacy. Only the second, technical, aspect will be discussed here.

The principle factors comprising security are:

a) User Identification.

b) System access control stratagems as a function of user identity.

c) System access-device control.

d) Audit records of accesses exercised.

e) Maintenance-access controls and audit records.

f) Certification of access control and audit recording systems.

g) Access security under hardware and software fault conditions.

h) Disposal procedures for obsolete records.

i) Encryption techniques for records and intra-and inter-system communications.

Computers of the future will provide their managers, users, and maintenance people specific means to exercise whatever level of security control desired.

Today, there are several levels of user identification employed (passwords and secure identification cards) and two types of system access control stratagems (hardware access control registers that provide "rings of control" and software stratagems that provide file and language barriers). Multics is an example of the hardware rings of control approach and Burroughs B6700 is an example of the language barrier system.

Weaknesses in all present systems are that the console operators and the maintenance people have access to all information and can therefore violate system security, and there is so far no good means to insure that a user with the correct password is who he says he is.

此为试读，需要完整PDF请访问：www.ertongbook.com