Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

77

Gregor v. Bochmann

Architecture of Distributed Computer Systems

Springer-Verlag
Berlin Heidelberg New York 1979

Editorial Board

W. Brauer P. Brinch Hansen D. Gries C. Moler G. Seegmüller
J. Stoer N. Wirth

Gregor v. Bochmann
Université de Montréal
Dépt. d'Informatique et de
Recherche Opérationelle (I.R.O.)
Case Postale 6128
Succursale "1"
Montréal, P.R.H3C 3J7
Canada

AMS Subject Classifications (1970): 68A05, 68B20, 94Axx
CR Subject Classifications (1974): 3.8, 4.3, 6.0

ISBN 3-540-09723-6 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-09723-6 Springer-Verlag New York Heidelberg Berlin

Library of Congress Cataloging in Publication Data
Bochmann, Gregor v. 1941-
Architecture of distributed computer systems.
(Lecture notes in computer science; 77)
Bibliography: p.
Includes index.
1. Electronic data processing--Distributed processing. 2. Computer architecture.
I. Title. II. Series.
QA76.9.D5B63 001.6'4 79-24491
ISBN 0-387-09723-6

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 1979
Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2145/3140-543210

PREFACE

This text is written for computer programmers, analysts and scientists, as well as computer science students, as an introduction to the architecture of distributed computer systems. The emphasis is placed on a clear understanding of the principles, rather than on details; and the reader will learn about the structure of distributed systems, their problems, and approaches to their design and development. The reader should have a basic knowledge of computer systems and be familiar with modular design principles for software development. He should also be aware of present-day remote-access and distributed computer applications.

The first part of the text serves as an introduction to the concept of "distributed system". We give examples, try to define terms, and discuss the problems that arise in the context of parallel and distributed processing. The second part deals with the description of parallelism, making abstraction from the physical distribution of the different system components. We discuss formalized methods that may be used to specify, and analyse the behaviour of, parallelism in local operating systems or distributed computer systems. In the third part, we explain the architecture of distributed systems and the role of the different communication protocols used. This includes the discussion of data transmission networks, as well as so-called higher level protocols used in computer networks for communication between different application programs, data bases, and terminals.

This text does not give the description of any particular distributed system, nor does it discuss the advantages and disadvantages of distributed computer applications, such as for banking transactions or distributed data bases. We have given extensive references to more detailed descriptions of the topics discussed, to complementary articles, and to explanations of certain prerequisite concepts, most readers will be familiar with.

This text was written when the author was a visiting professor at the Ecole Polytechnique Fédérale de Lausanne, Switzerland. It represents the lecture notes of a one-semester course ("troisième cycle") given in the Département de Mathématiques in 1977-78. I would like to thank the Département de Mathématiques, and in particular Professor G. Coray, for my pleasant stay in Lausanne. For the preparation of this text, I have profited from many discussions, in particular with J. Gecsei (Montreal), D. Gurtner and F. Vittoz. I thank S. Waddell for suggesting many improvements of the original manuscript, and Ch. Luyet and D. Salconi (Montreal) for the careful typing. Last, but not least, I thank my wife, Elise, for her patience and moral support.

TABLE OF CONTENTS

Part I: Introduction

1. Distributed systems: examples and definition 1

1.1. Distribution of control and data in existing systems 1

1.1.1. Systems distributed over long distance 1

1.1.1.1. Remote access 1

1.1.1.2. Computer networks 6

1.1.1.3. Systems for distributed processing 9

1.1.2. Locally distributed systems 13

1.1.3. Multi-processor systems 18

1.1.4. Virtual distribution 18

1.2. Classification of distributed systems 20

1.2.1. Degree of coupling 20

1.2.2. Interconnection structure 21

1.2.3. Interdependence of components 24

1.2.4. Synchronization between components 24

1.3. Definition of "distributed system" 25

2. Parallelism 31

2.1. Parallel processes and applications 31

2.2. Constraints on independence 33

2.3. Modular system structure and abstraction 37

3. Common problems 42

3.1. Cooperation 42

3.1.1. Compatibility 42

3.1.2. Synchronization 43

3.2. Distributed resource sharing 43

3.3. Naming and addressing 44

3.3.1. Search strategies for link editors 44

3.3.2. Naming of I/O flows 44

3.3.3. Addressing scheme of telephone networks 45

3.3.4. Process addressing by ports 45

3.4. Protection 48

3.4.1. Protection in distributed systems 49

3.5. Error recovery 50

3.5.1. Error detection 50

3.5.2. Recovery by retry 51

3.5.3. Redundant hardware 51

3.5.4. The design principle of recovery blocks 52

3.6. Real time considerations 53
3.6.1. Time-outs 54
3.6.2. Clock synchronization in distributed systems 54

Part II: Logical description of parallelism

4. A general formalism for the description of systems 57
4.1. The basic model 57
4.1.1. Transition systems 57
4.1.2. Operations 58
4.1.3. Transition and relations between states 58
4.1.4. Abstraction 61
4.1.5. Parallelism and functionality 63
4.2. Reachability and execution sequences 65
4.2.1. Possible operation sequences 65
4.2.2. Liveness 67
4.2.3. Equivalence between systems 68
4.3. Synchronization mechanisms 68
4.4. Non-instantaneous operations 72
4.4.1. Mutual exclusion 72
4.4.2. Queueing considerations and scheduling 74
4.5 Processes 75
4.5.1. The concept 76
4.5.2. Cooperation 77
4.5.3. Mutual exclusion 82
4.6. The induction principle 84
4.7. Distinction between "control structure" and "interpretation" 85
4.7.1. Notation for the case of a finite control structure 88
4.8. Assertions 89
4.9. Formalized specification methods for systems with parallelism 92

Part III: Architecture and communication protocols for distributed systems

5. Architecture of distributed systems 96
5.1. Layered hierarchial system structure and physical distribution 97
5.2. Typical structure of a distributed system 108
5.2.1. Communication over a dedicated circuit 109
5.2.2. Communication through a network 112
5.2.3. A uniform transport service 113
5.2.4. Higher level protocols 115

5.3. Compatibility and interworking issues 118

5.3.1. Requirements for compatibility 119

5.3.2. Network interconnection 120

5.3.3. System interworking and adaptation 124

5.4. Specification, verification and implementation of protocols 125

5.4.1. Specification techniques 126

5.4.2. Protocol verification 127

5.4.3. Protocol implementation 129

6. Message transport requirements and data transmission networks 133

6.1. Message transport requirements 133

6.2. Data transmission services 135

6.2.1. Dedicated circuits 135

6.2.2. Switching 136

6.2.3. Circuit and packet switching 136

6.3. The transport protocol 137

7. Line protocols 140

7.1. Transmission of bits 141

7.1.1. Interface procedures 141

7.1.2. Physical media 142

7.1.3. Bit synchronization 144

7.1.4. Typical performances 147

7.2. Transparency and framing 148

7.2.1. Bit-oriented method 149

7.2.2. Character-oriented method 150

7.2.3. Method based on envelope transmission 151

7.3. Transmission error detection and correction 151

7.3.1. Principles 151

7.3.2. Error detecting codes 155

7.3.3. Error correcting codes 158

7.4. Retransmission protocols 158

7.4.1. Principles 158

7.4.2. The "alternating bit" protocol 161

7.4.3. The HDLC classes of procedures 164

7.4.4. Multiplexing 170

8. Technological developments and standards 176

References 179

Annex: An example of a protocol implementation based on a formalized specification. 192

A.1. A unified model for the specification and verification of protocols 194

A.2. Combining assertions and states for the validation of process communication 201

A.3. Defining a layer service 206

A.4. A formalized specification of HDLC classes of procedures 210

A.5. Development and structure of an X.25 implementation 222