Lecture Notes in Computer Science 774
Edited by G. Goos and J. Hartmanis

Michel Banâtre Peter A. Lee (Ed.)

Hardware and Software Architectures for Fault Tolerance

Experiences and Perspectives

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 69 80
Vincenz-Priessnitz-Straße 1
D-76131 Karlsruhe, Germany

Juris Hartmanis
Cornell University
Department of Computer Science
4130 Upson Hall
Ithaca, NY 14853, USA

Volume Editors

Michel Banâtre
INRIA-IRISA
Campus de Beaulieu, F-35042 Rennes Cedex, France

Peter A. Lee
Department of Computing Science, University of Newcastle upon Tyne
NE1 7RU Newcastle upon Tyne, United Kingdom

CR Subject Classification (1991): C.3, D.4.5, H.2.7, E.5, B.4.5, D.4.7

ISBN 3-540-57767-X Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57767-X Springer-Verlag New York Berlin Heidelberg

CIP data applied for

Typesetting: Camera-ready by author
SPIN: 10131918 45/3140-543210 - Printed on acid-free paper

Preface

For many years now, fault tolerance has been a very active research area, and there are many conferences and workshops at which fault tolerance research papers and results are presented. However, in 1992 we felt that the time was ripe for a different, yet complementary, workshop on fault tolerance where a small number of key researchers and practitioners in the area could get together for more "intimate" discussions and presentations. The kinds of issues we wished such a workshop to consider included:

- There seem to have been relatively few major advances in fault tolerant architectures over the last few years; is this true, and if so is this because the subject is becoming mature or stale?
- Or is it the case that fault tolerance has become a mature topic for some applications (e.g. transaction processing) but still requires research to be undertaken in other application areas (e.g. safety-critical or real-time)?
- What are the present-day causes of system failures which require fault tolerance in our systems? Are hardware faults as relevant today as the amount of literature on hardware fault tolerance techniques might suggest?
- What operating system work is relevant to fault tolerant architectures?
- What is the application experience - that is, what do applications like transaction processing and database systems or real-time systems now require in terms of support from the hardware and operating systems, to enable them to provide an efficient and reliable service?
- What can we learn from the experiences so far, and hence what will be the hot topics for research in the coming years?

With questions such as these in mind, we approached a number of experts in the field, and formed a Program Committee consisting of Michel Banâtre, Pete Lee, Ken Birman, W. Kent Fuchs, Farnam Jahanian, David Powell and Jack Stiffler. The committee decided upon the technical topics to be covered in the workshop and selected a final set of approximately 25 speakers to present invited papers. The committee were also responsible for selecting the attendees of the workshop, and we were very happy with the enthusiastic responses of speakers and participants to our personal invitations. Organisation of the workshop was undertaken by INRIA-IRISA, Rennes and the University of Newcastle upon Tyne, and was held in June 1993, at Le Mont Saint Michel in France. Approximately 40 experts eventually attended the workshop, and by all accounts, the workshop was a great success, well received and commended by those who attended.

This volume of Lecture Notes contains the papers presented at the workshop, but revised after the workshop to take account of some of the issues that the

workshop raised. We have organised the papers into five sections: Field Experiences with Fault Tolerant Systems; Hardware Architectures for Fault Tolerance; Software Architectures for Fault Tolerance; Embedded and Real-Time Systems; and finally Data and Databases. Within each section there is a mixture of papers from academia and from industry, with technical presentations as well as position papers addressing some of the issues mentioned earlier. Indeed, it is these position papers, presenting the views of people in the computer industry on fault tolerance and its future, which make this book unique. (The papers are discussed below with respect to the author who presented the paper at the workshop, although some of the papers have multiple authors.)

The section on "Field Experiences with Fault Tolerant Systems" contains three papers from authors working in industry, discussing different aspects of their experiences of real fault tolerant systems. There are papers from Ram Chillarege of IBM TJ Watson Research Center, Ytzhak Levendel of AT&T and Doug Locke from IBM Federal Systems. Chillarege's paper presents his view of the top challenges facing the practice of fault tolerance. Reliability experiences from AT&T's Electronic Switching Systems (ESS) are the subject of Levendel's paper, which also discusses the fault tolerance approaches that future systems might take. Locke's paper, derived from many years experience of specifying and constructing high reliability applications, addresses the fault tolerance requirements from the perspective of applications, noting the increasing dependence upon software for fault tolerant behaviour and hence the growing importance of software-fault tolerance techniques.

The section on "Hardware Architectures for Fault Tolerance" contains papers presented by Michel Banâtre (IRISA-INRIA), W. Kent Fuchs (University of Illinois), Barry Gleeson (Unisys Corporation), Jeremy Jones (Trinity College Dublin), David Liddell (IMP Ltd.), Dhiraj Pradham (Texas A&M University) and Jack Stiffler (Sequoia Systems Inc.). The papers are a mixture of technical topics and position papers on the way ahead. Banâtre's paper addresses some on-going research into the provision of fault tolerant scalable shared memory multiprocessor architectures. The paper by Fuchs and colleagues introduces a novel compiler-assisted roll-back scheme for repairing the erroneous effects of speculative instruction executions in parallel architectures. Gleeson's paper discusses the issue of why fault tolerant systems are not in as widespread use as might be expected, the cost/benefit trade-offs for fault tolerance and the market and technology challenges that lie ahead.

As CPU cycles become more freely and cheaply available, the I/O performance in a system is rapidly becoming the bottleneck constraining overall system performance. Thus the paper by Jones, describing the Stable Disk which is a high-performance RAID disk system with additional fault tolerance characteristics, covers an important issue in fault tolerant systems. Liddell's paper presents another industrial viewpoint on the problems faced by fault tolerant system designers, and overviews the design approach taken in IMP's fault tolerant computer systems. The paper by Pradhan studies schemes which can avoid the roll-back of a task's state in the face of transient failures through the use

of replication. In the final paper in this section, Stiffler argues that the need for fault tolerant systems continues to grow and he surveys some of the existing commercial approaches. He presents his view on the problems that lie ahead and need to be solved in the next generation of fault tolerant systems.

The third section of papers on "Software Architectures for Fault Tolerance" has contributions from Yair Amir (Hebrew University of Jerusalem), Ken Birman (Cornell University), Elmootazbellah Elnozahy (Carnegie Mellon University), Yennun Huang (AT&T Bell Laboratories), Pete Lee (University of Newcastle upon Tyne), Jim Lipkis (Chorus Systemes), Gilles Muller (IRISA-INRIA), David Powell (LAAS-CNRS) and Santosh Shrivastava (University of Newcastle upon Tyne). Amir's paper describes the construction of a reliable distributed application build on top of the Transis environment. Birman argues that a new class of distributed application is emerging due to the manner in which organisations are using computing systems, and such applications will require advances in the techniques used to build reliable distributed software systems. It is clear that networks of workstations will be a feature of many distributed systems, and the paper by Elnozahy describes the Manetho system which provides transparent fault tolerance for parallel applications running on such systems.

A conclusion which comes out of many of the papers in the conference is that software faults remain a key problem area. The paper by Huang describes two techniques that have been successfully applied to enhance the reliability of software systems without requiring the provision of redundancy in the form of design diversity. Lee's paper is also concerned with the problems of software faults. He discusses the nature of software faults, summarises the techniques which have been proposed for tolerating such faults, and describes an overall architecture for fault tolerant software systems.

Micro-kernels are an increasingly important facet of present-day operating systems. The papers by Lipkis and by Muller address some of the issues of micro-kernel support for fault tolerance, in the Chorus and Mach systems respectively. Powell's paper presents some of the lessons learned from the Delta-4 project which was investigating fault tolerance in distributed systems. Finally in this section, Shrivastava presents some case studies in building reliable distributed applications in computing environments which consist of off-the-shelf hardware and software components with no special provisions for fault tolerance.

The section of the book entitled "Embedded and Real-Time Systems" contains papers by Rod Bark (Hewlett-Packard Laboratories), Hermann Kopetz (Technical University of Vienna), Jay Lala (Charles Stark Draper Laboratory), and Farnam Jahanian (University of Michigan). Bark's paper is concerned with the telecommunications markets, their fault tolerance requirements, and the hardware and software issues that arise. Critical, real-time systems are the focus of the paper by Kopetz, who presents some of the key design approaches for the construction of large real-time systems. Real-time systems are also addressed by Lala and Jahanian. Lala's paper discusses strategies for dealing with common-mode failures in such systems, while Jahanian argues that the traditional approaches to fault tolerance need to be re-examined for time-critical applications.

The final section of the book, on "Data and Databases", has papers by Andrea Borr (Hewlett-Packard) and Jehan-Francois Pâris (University of Houston). Borr's paper argues that special hardware provisions for fault tolerance are not a necessary part of an environment which is required to provide a high level of availability to data, and she discusses a prototype client-server system based on standard hardware and software components. Last, but by no means least, the paper by Pâris concentrates on the problems of managing replicated data. The paper provides a summary of the state-of-the-art in this area, and presents some of the unresolved issues which are likely to require attention in the years ahead.

Conclusions

Many of the conclusions arising from this workshop are covered in the papers outlined above, since the authors have had a chance to revise their workshop presentations for this book. However, in this section we provide a selection of key fault tolerance issues that arose during the course of the 3 days.

Hardware

- There is a strong move away from proprietary hardware solutions
- Fault tolerance measures are still needed for the hardware components in a computing system
- Hence, off-the-shelf hardware components must begin to provide support for fault tolerance mechanisms. This should be the next use of spare silicon, and the mechanisms required are well understood
- Cost is still a major issue. Customers want reliable behaviour but are not willing to pay other than a very small premium for fault tolerance. (One participant drew a parallel between selling fault tolerance and insurance!)
- Unreliable communications hardware is causing enormous software problems.

Software

- There is a growing emphasis on software architectures as the basis for reliable distributed applications
- A set of different architectural models is emerging: group communications, atomic actions/transactions, transparent fault tolerance
- The Unix model makes life difficult for the fault tolerance implementer
- Software-fault tolerance remains a key issue
- Seemingly ad-hoc (but low-cost) techniques do seem to work in practice

December 1993

Michel Banâtre
Peter A. Lee

Program Committee

Chair

Michel Banâtre	IRISA/INRIA, Rennes (F)
Pete Lee	University of Newcastle upon Tyne (UK)

Members

Ken Birman	University of Cornell (USA)
W. Kent Fuchs	University of Illinois (USA)
Farnam Jahanian	IBM, R.J Watson (USA)
David Powell	LAAS, Toulouse (F)
Jack Stiffler	Sequoia, Massachusetts (USA)

Organizing Committee

Michel Banâtre	IRISA/INRIA, Rennes (F)
Elisabeth Lebret	IRISA/INRIA, Rennes (F)
Pete Lee	University of Newcastle upon Tyne (UK)
Christine Morin	IRISA/INRIA, Rennes (F)

A number of people contributed to the success of the workshop. We offer our sincere thanks to all of them. We are particulary grateful to Maryse Auffray, Marie-Noëlle Georgeault, Philippe Lecler, Evelyne Livache and Gilles Muller.

Sponsors

- Bull
- CEC (Esprit)
- France telecom

Organization

IRISA/INRIA, Rennes
University of Newcastle upon Tyne

Table of Contents

I Field Experiences with Fault Tolerant Systems

II Hardware Architectures for Fault Tolerance

III Software Architectures for Fault Tolerance

IV Embedded and Real-Time Systems

V Data and Databases