

**PARALLEL  
AND DISTRIBUTED  
PROCESSING**

TP301.6/B789

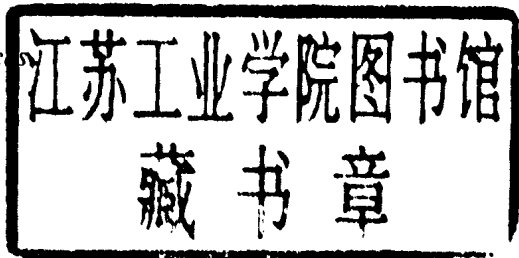
# PARALLEL AND DISTRIBUTED PROCESSING

Proceedings of the Second Workshop on  
Parallel and Distributed Processing (WP&DP '90)  
Sofia, Bulgaria, 27–29 March, 1990

*edited by*

KIRIL BOYANOV

*Bulgarian Academy of Sciences  
Sofia, Bulgaria*



1991

NORTH-HOLLAND  
AMSTERDAM • NEW YORK • OXFORD • TOKYO

**ELSEVIER SCIENCE PUBLISHERS B.V.**  
Sara Burgerhartstraat 25  
P.O. Box 211, 1000 AE Amsterdam, The Netherlands

Distributors for the United States and Canada:  
**ELSEVIER SCIENCE PUBLISHING COMPANY, INC.**  
655 Avenue of the Americas  
New York, N.Y. 10010, U.S.A.

**Library of Congress Cataloging-in-Publication Data**

**Workshop on Parallel and Distributed Processing (2nd : 1990 : Sofia, Bulgaria)**

Parallel and distributed processing : proceedings of the Second Workshop on Parallel and Distributed Processing (WP&DP '90), Sofia, Bulgaria, 27-29 March 1990 / edited by Kiril Boyanov.

p. cm.

Includes bibliographical references and index.

ISBN 0-444-88868-3

1. Parallel processing (Electronic computers)--Congresses.
  2. Electronic data processing--Distributed processing--Congresses.
- I. Boyanov, Kiril L. II. Title.

QA76.58.W67 1990

004'.35--dc20

90-48106  
CIP

ISBN: 0 444 88868 3

© ELSEVIER SCIENCE PUBLISHERS B.V., 1991

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form of by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher, Elsevier Science Publishers B.V./Physical Sciences and Engineering Division, P.O. Box 103, 1000 AC Amsterdam, The Netherlands.

Special regulations for readers in the U.S.A. - This publication has been registered with the Copyright Clearance Center Inc. (CCC), Salem, Massachusetts. Information can be obtained from the CCC about conditions under which photocopies of parts of this publication may be made in the U.S.A. All other copyright questions, including photocopying outside of the U.S.A., should be referred to the copyright owner, Elsevier Science Publishers B.V., unless otherwise specified.

No responsibility is assumed by the publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

pp. 41-58, 169-184, 315-334: copyright not transferred.

Printed in The Netherlands

## PREFACE

In 1989 a workshop on Parallel and Distributed Processing was organised at the Bulgarian Academy of Sciences in Sofia on the initiative of the Laboratory for Distributed Systems and Computer Networks, and the Laboratory for High Performance Systems and Parallel Algorithms.

The participants at this first meeting were scientists from the United Kingdom and Bulgaria. The United Kingdom was represented by researchers from Manchester University, Imperial College and the University of Southampton.

The success of the workshop and the great interest shown in it, encouraged the organisers to go further with the second workshop in 1990. There were participants from the United Kingdom, Belgium, Germany, Bulgaria, Greece, and France.

The aim of the workshop was to provide discussions on the main problems in the theory of parallel and distributed architectures. A second main goal was to expand the contact between researchers of different countries. At this time of worldwide political change and increased and related understanding, the free exchange of opinions among scientists is of great significance. Because of this, mutual confidence will be raised and problems will be looked at from different viewpoints, thus establishing the merits of respective partners. Researchers from different countries will probably set up working groups to look into common projects. At the same time decisions to solve problems on parallel data processing will be looked for by research teams in laboratories and universities.

The WP&DP'90 was held in Sofia between March 27-29, and covered a wide range of basic topics concerned with the above mentioned problems. Most of the papers presented are theoretical but there were some presentations covering application areas with the possibility of direct implementation.

I hope that WP&DP'90 provided a step toward solving some of these problems. What, however, is more important is that it should become a step to establishing a tradition of annual meetings, where research results can be presented.

In this volume selected papers have been published which follow the main topics of the workshop. I would like to express my gratitude to all authors and to my associates M. Iliev and D. Djendov for their kind and selfless assistance in the preparation of these proceedings.

I strongly encourage readers to participate in future workshops, since this will greatly increase the numbers of people interested in parallel processing.

Professor K. Boyanov

## COMMITTEES

### WORKSHOP CHAIRMEN

Kiril Boyanov  
*Head of the Laboratory for  
 Distributed Systems and Computer Networks  
 Centre for Informatics and Computer Technology  
 Bulgarian Academy of Sciences*

Stoyan Markov  
*Head of the Laboratory for  
 High Performance Systems and Parallel Algorithms  
 Centre for Informatics and Computer Technology  
 Bulgarian Academy of Sciences*

### PROGRAMME COMMITTEE

K. Boyanov (BG)	Ch. Hankin (UK)
P. Capon (UK)	Ch. Jesshope (UK)
M. Cosnard (FRA)	A. Knowles (UK)
H. Glaser (UK)	Vi. Lasarov (BG)
J. Gurd (UK)	M. Reeve (UK)

### ORGANISING COMMITTEE

K. Yanev (BG)  
 Vi. Lasarov (BG)  
 M. Iliev (BG)  
 Kr. Arabadjiiski (BG)  
 D. Djendov (BG)

## **SPONSORS**

Institute for Microprocessor Systems and Instruments  
Central Institute for Computer Technics and Technologies

## **ORGANISERS**

Bulgarian Academy of Sciences  
Under the Patronage of the President: Acad. Bl. Sendov

## TABLE OF CONTENTS

Preface	v
Committees	xi
Sponsors and Organisers	xiii
Introduction K. Boyanov	1
<b>PARALLEL AND DISTRIBUTED ARCHITECTURES</b>	
High Performance Communications Architectures Ch. Jesshope	7
A Message Passing System for a Network of Transputers A.E. Knowles, N.N. Avramov	27
Formal Specification of a Distributed Router for nD Hypercube M. Krapp, V. Jossifov	41
Parallel Processing in Frequency Dependent Neural Network L. Hadjyisky	59
Dynamic Process Support in a Parallel Multicomputer System Vl. Lazarov, R. Iliev, D. Djendov	75
The Influence of Technology on the Choice of a Multiprocessor Interconnection Network M. Engels, R. Lauwereins, J.A. Peperstraete	91
<b>PARALLEL PROGRAMMING</b>	
C.NET: A High Level Programming Environment for Supernode Multiprocessor J.M. Adamo, J. Bonneville, C. Bonello	113
Distributed Operating Environment: DIOGEN I.M. Nedelchev, At.N. Parashkevov	131
Graphical Software Environments for Transputer Based Systems P.C. Capon, A.J. West	141

System Software for Multitransputer Systems-Analysis and Comparison between the Basic Approaches L.K. Manasiev, A.T. Vassilev, R.D. Ushatov	157
A Pragmatic Approach to the Analysis and Compilation of Functional Languages H. Glaser, P. Hartel, J. Wild	169
Parallelism in Parsing Algorithms of Specification Languages LXIC and SYNT I.V. Bojanova, V.T. Dimitrov	185
A Paragon Description of Parallel Supercombinator Graph Rewriting D. Bolton, Ch. Hankin	199

## PARALLEL ALGORITHMS

Implementation of Parallel Algorithms on a Programme Environment for Symbolic and Numeric Computations K. Boyanov, D. Djendov, M. Iliev	215
Experiences on the Monte Carlo Parallel Solutions V. Sabev, P. Nicolov	233
On the Analysis of Polynomials Roots-Finding Parallel Algorithms M. Cosnard, P. Fraigniaud	243
A Parallel Tessellation Based on Transputers A.M. Velichkov	263
PARCLIB: A Parallel Library Based on a Replication Paradigm I. Nitcheva, O. Tchipev	277
A Systolic Linear Array for the Knapsack Problem V. Aleksandrov, R. Andonov	285

## PERFORMANCE EVALUATION

Comparison of Dynamic Load Balancing Strategies H. Kuchen, A. Wagener	303
A New Method for Round-Off Error Estimation V.V. Voevodin, P.Y. Yalamov	315
Performance of Computing Environments in DSP R.L. Petrov, A.S. Ognianov, O.D. Gorchakov, Iv.D. Terziev	335



**Influences on the Processor Utilisation in Parallel Computers  
with Point-to-Point Interconnection Networks**

**R. Lauwereins, M. Engels, J.A. Peperstraete**

361

**Monitoring of Transputer CPU/FPU Utilisation in Run-Time**

**K. Yanev, J. Giurov**

361

**Presentation and Comparison of Three Self Adaptive Multiple  
Access Protocols for Local Area Networks**

**S.A. Koubias**

377

**Author Index**

385

## INTRODUCTION

Kiril Boyanov

Laboratory for Distributed Systems and Computer Networks  
Center for Informatics and Computer Technology  
Bulgarian Academy of Sciences  
Sofia, Bulgaria

Over the recent years the amount of data processing has considerably augmented. The speed, reliability and cost for information unity requirements have also increased. The demands originated also into connection to a number of new problems in science, industry and the social environment.

In order to increase the speed of computer systems, researches were directed towards two basic trends: the creation of high-speed circuits and the exploration of new methods for parallel processing. The first trend almost reached its limits, and the second one is being intensely developed.

Parallel processing is an effective instrument for information processing, which uses mostly concurrent events during computation. Parallelism may be looked at the following levels: (i) at the software level; (ii) at the hardware level. The computer system's performance is increased through the concurrent execution of many programmes in the computer.

Several levels of parallel processing may be considered in relation to software. The highest level is the use of the multiprogramme mode. The execution of parallel algorithms depends on the appropriate distribution of the limited hardware and software resources between the lot of programmes. The next level of parallel processing is executed through the use of appropriate procedures in the same programme. This includes the decomposition of the programme in many tasks. At the lowest level, instruction concurrency is achieved, the data is being analyzed to discover possibilities for the parallel processing of instructions. It is also possible to look for internal parallelism at the instruction level, which is executed directly by the hardware. The part of the firmware increases from the higher to the lower levels. On the other side a software implementation of parallelism is sought after from the lower to the higher levels. The compromise between the hardware and the software approach for solving the problem has always been a matter for discussion. As the cost of hardware goes down, the software cost goes up. Hardware methods replace more and more the conventional software approach. This tendency is kept up by the increasing demand for a faster real time computing environment,

distributed resources and fault tolerance. On account of the price reduction of VLSI circuits and the increasing facilities for technological implementations, software ideas, created at the programming level, are often realized in silicon compilers.

Parallel information processing is closely bound up with distributed processing. The creation of a distributed environment, in which separate elements of the programme may be concurrently processed, increases the performance of the computing system. Therefore the final effect, which is anticipated from the introduction of parallel architectures and distributed systems is the enhancement of the overall performance.

As the progress of communication technology heads on, the difference between parallel and distributed processing almost disappears. In this broad sense distributed processing may be considered as a form of parallel processing in a specific environment.

Naturally the approaches to parallelism and distributed processing are different, as they complement one another. Besides it is only natural to look for a complex solution, so that the overall effect could be optimal. But one must always bear in mind the necessity to take into consideration the existing software.

The putting into operation of new systems with parallel architecture or distributed processing will be completed on stages, until users turn to a new way of interaction with the computing systems.

According to Hwang and Briggs, in *Computer Architectures and Parallel Processing* - McGraw-Hill 1985, from an application point of view, "the mainstream usage of computers may be considered like a trend of four ascending levels of sophistication: data processing, information processing, knowledge processing, and intelligence processing. There is a relationship between data, information, knowledge, and intelligence processing. The data space is the largest, including numeric numbers in various formats, character symbols, and multidimensional measures. Data objects are considered mutually unrelated in the space. An information item is a collection of data objects that are related by some syntactic structure or relation. Hence information items form a subspace of the dataspace. Knowledge consists of information items plus some semantic meaning. Thus knowledge items form a subspace of the information space. Finally, intelligence is derived from a collection of knowledge items, so the intelligence space is the last subspace formed in the knowledge space".

Recently, the quick accumulation of knowledge bases increased the use of computers for knowledge processing. I.g., different expert systems helped solving problems in certain fields. It is surmised this will be the main trend in computer application in the 90-ties. Today's computers have a high throughput and a large memory, which may be utilized for data, information, and knowledge processing. However, the existing computers may not be accepted as systems, having minimal intellectual qualities.

The creation of new microprocessor elements gives the possibility to offer new architectural solutions for parallel processing. The new architectures of parallel machines, built up from universal or custom microprocessor elements, presuppose the clearing of a number of questions, related to the information communication. Naturally a question

arises: which information is to be processed: intelligence, knowledge or data, i.e. are there architectures, more effective for intelligence than for data processing, so that better performance is to be achieved? And if the next generation machines process knowledge bases and intelligence with a sufficiently high speed, will they cover the requirements for processing in the data space? Or in other words: if a high speed machine is implemented, showing a sufficiently effective processing of the whole data space, will it mean that it is an suitable device for intelligence processing? And vice versa: if an architecture for intelligence processing is created, will it be effective for unstructured data processing? A possible question is whether an inadequate architecture must be related to each subspace to achieve a maximum performance and efficiency, and is it possible to achieve this with a reconfigurable architecture?

To answer these questions it is necessary to proceed with research in the following directions:

- design of new parallel and distributed architectures;
- development of new concepts for parallel programming;
- creating effective parallel algorithms;
- implementation of new methods and approaches for performance evaluation.

Other directions under development could be pointed. Part of the them are worked upon intensely, and results have been obtained. A quick answer to all questions may not be expected, as during the scientific research additional difficulties will arise.

The workshop discussions as well as the published papers were aimed at the search of means and approaches for the solutions of the above mentioned problems. Exchange of opinions and ideas was the main objective of the workshop.



# **PARALLEL AND DISTRIBUTED ARCHITECTURES**



## HIGH PERFORMANCE COMMUNICATIONS ARCHITECTURES

Dr Chris Jesshope

Dept of Electronics and Computer Science  
The University  
Southampton  
SO9 5NH  
UK

This paper presents simulation results of the mad postman routing strategy and implementation details if a mad postman routing chip. It is shown that latencies of a few microseconds can be achieved for global permutations in a 2-D 1024 node network built using these chips. It is also shown that similar latencies can be maintained for the routing of randomly addressed data in the same network. The success of this scheme begs a more efficient interface to the processor and the paper considers this future direction.

### 1. INTRODUCTION

This paper presents further results concerning the implementation and simulated performance of a chip implementing the the mad-postman packet routing strategy and the method of virtual networks for deadlock avoidance in such networks[1-6]. The chip MP1 is currently in the final stages of design and will be submitted for fabrication via ES2's 1.5 micron CMOS cell-based design route, by the summer of 1990.

The packet-routing node is implemented to act as a co-processor to a variety of 32 bit microprocessors, such as the INMOS transputer or the Acorn ARM chip. It will provide a network with up to 1024 nodes directly addressable, driven synchronously by a single clock signal. It will be shown that larger networks or different temporal regimes are supported using nested addressing and gateway techniques.

Each communication node will be realised with one or more MP1 chips, where each chip provides two routing planes ("virtual" planes). A one chip solution requires software to provide class- climbing or multiplexing solutions to the deadlock problem. Normally nodes will be constructed with pairs of chips to provide multiples of four "virtual" planes. One pair of chips will together provide a message routing bandwidth in the region of 100 Mbyte/sec bits per second per node, giving a system bandwidth of 100 Gbyte/sec for a complete network. Processor intervention will only be required when injecting or retrieving messages from the network, which will be achieved by block transfer.



Details of the architecture together with estimates of the performance are presented here for the MP1 chip implementation. The chip will support a number of features which greatly facilitate heterogeneous systems support. It includes a generic 32 bit bus-interface, direction configuration, broadcast and point-to-point communication within the same network (at the same time) and nested packet addressing for deterministic routing to facilitate fault avoidance, domain partitioning etc.

In this paper we also look at the motivation for this design and conclude that in some respects the mad-postman strategy is too successful, in that we have achieved message delivery latencies in large networks of the order of a few microseconds, even for randomly addressed packets. We show that for current microprocessor interfaces the delay in presenting data to the MP1 chip is likely to exceed these network delays.

We believe however that in order to provide extremely fine grain parallel processing systems in the future, it will be necessary to design a dedicated microprocessor interface, which should support packet assembly and disassembly and direct access to memory. This in turn however will require additional forms of synchronisation implemented in hardware between the locally shared memory exposed within such an architecture (i.e. between processor and network interface).

## 2. FINE GRAIN PARALLEL PROCESSING

Fine grain parallel processing is necessary if we wish to be able to scale the performance of a parallel computer by the addition of further processors; to date however, this has been an elusive goal. In most practical systems for example, as the degree of parallelism is increased the efficiency of computation decreases, and in many reported cases adding processors can actually decrease performance rather than increasing it.

The computer characteristics that determine the granularity that can be supported on a given parallel processor include the processing and communications performance, but may also be limited by memory. For example, if communications bandwidth compared to processing bandwidth is very poor, then more memory will be required per processor in order to support the large grains of computation that this ratio requires. If on the other hand communications bandwidth is large compared to processing bandwidth then less memory per processor can be tolerated due to the smaller grain size of computation that can now be accommodated.

Whatever this ratio however, the requirements for communication (and synchronisation) may be divided into two separate regimes, both of which are equally important and must be addressed in any network design. These two regimes may be called the *saturated* and *tail-effect* regime and correspond to the following situations:

**Saturated Regime:** in this regime we find the maximal concurrency in a computation and the overheads for communication and synchronisation will be hidden by overlapping them with computation, provided of course the computation time exceeds the communication time for a given code and architecture.

For example, if one process wishes to communicate and synchronise with another process on another processor, then its communication, which must be blocking, will deschedule that process. However, provided that we are in the saturated regime there will exist *parallel slackness*, i.e. internal concurrency which ensures that other processes that may use the processor's computation resource and hence hide the communication.