

HYPERCUBE AND DISTRIBUTED COMPUTERS



HYPERCUBE AND DISTRIBUTED COMPUTERS

Proceedings of the First European Workshop on
Hypercube and Distributed Computers
held in Rennes, France, 4-6 October, 1989
Organised by the Institut National de Recherche en
Informatique et en Automatique
— Unité de Recherche Rennes — IRISA

edited by

F. ANDRE
INRIA-IRISA
Rennes, France

J. P. VERJUS
IMAG
Grenoble, France



Institut National de Recherche en
Informatique et en Automatique



1989

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, NY 10010, U.S.A

ISBN: 0 444 88086 0

Elsevier Science Publishers B.V., 1989

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher, Elsevier Science Publishers B.V./Physical Sciences & 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, 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.

Printed in The Netherlands

FOREWORD

The idea of organizing a European Workshop on Hypercube and Distributed Architectures arose about one year ago when it appeared necessary to gather the growing number of researchers and company members, interested in distributed parallel computers together.

A meeting organized in Brussels, in October 1988, showed clearly that such a conference would draw a lot of interest. It was then decided that the First European Workshop would take place in Rennes, France.

A European Program Committee of 21 members had the responsibility to select 23 papers and 22 posters out of around 65 propositions. Each paper has been reviewed by at least 3 referees. The final scientific program was established during the Program Committee Meeting held in Paris on April 10th, 1989.

We would like to thank all the organizations which have generously co-sponsored this workshop:

- University of Rennes I
- Société de Mathématiques Appliquées et Industrielles (S.M.A.I.)
- Office National d'Etudes et de Recherches Aéronautiques (O.N.E.R.A.)
- Centre National de la Recherche Scientifique (C.N.R.S. Section 08)
- French C.N.R.S. Coordinated Research Program C3
(Communication - Cooperation - Concurrence)

It is interesting to underline that thanks to this National Research Program quite a lot of research projects on parallelism have been carried out by various French teams working together. We may assume that some contributions proposed to this workshop were, in a way, the result of such a cooperation.

We would like to thank all the persons who have contributed to the success of this workshop. We acknowledge the hard work which the Program Committee members and the reviewers have carried out.

This workshop has been organized by the INRIA - Rennes - IRISA Research Center and we would like to particularly thank all the staff both from the headquarters

in Rocquencourt and in Rennes for all the work they have accomplished during these past twelve months.

Again our sincere thanks to all who have contributed to the success of the First European Workshop on Hypercube and Distributed Computers.

Françoise André
Workshop Chairperson

Jean Pierre Verjus
Program Committee Chairperson

SPONSORS, PRESIDENT AND CHAIRPERSONS OF THE CONFERENCE

Sponsors

Université de Rennes I
 CNRS-Section 08 (Centre National de la Recherche Scientifique)
 C3 (Coopération, Concurrence, Communication (GRECO CNRS))
 GAMNI/SMIAI (Société de Mathématiques Appliquées et Industrielles)
 ONERA (Office National d'Etudes et de Recherches Aéronautiques)

President / Chairpersons

J.P. Verjus, IMAG -- Grenoble, France

C. Addison, CMI -- Fantoft, Norway
 F. André, INRIA/IRISA -- Rennes, France
 L. Beernaert, Katholieke Universiteit Leuven, Belgium
 T. Bemmerl, Technische Universität München, F.R.G.
 M. Bertocchi, University of Bergamo, Italy
 R. Chamberlain, INTEL -- Swindon, G.B.
 M. Cosnard, ENS -- Lyon, France
 R. Hempel, GMD -- St Augustin, F.R.G.
 B. Kagström, University of Umeå, Sweden
 P. Leca, ONERA -- Chatillon, France
 H. Leroy, INRIA/IRISA -- Rennes, France
 A. Lichnewsky, INRIA -- Rocquencourt, France
 B. Philippe, INRIA/IRISA -- Rennes, France
 P. Quinton, IRISA -- Rennes, France
 M. Raynal, IRISA -- Rennes, France
 Y. Robert, ENS -- Lyon, France
 D. Roose, Katholieke Universiteit Leuven, Belgium
 K. Solchenbach, SUPRENUM -- Bonn, F.R.G.
 U. Trottenberg, SUPRENUM -- Bonn, F.R.G.
 M. Valerno, UPC -- Barcelona, Spain

TABLE OF CONTENTS

Foreword	v
Sponsors, president, and chairpersons	vii
Invited Conference	
An assessment of second-generation multicomputers C.L. Seitz	1
NON NUMERICAL ALGORITHMS	
Component labelling on a distributed memory multiprocessor H. Embrechts, D. Roose, P. Wambacq	5
Dynamic programming on a ring of processors S. Miguet, Y. Robert	19
PARTIAL DIFFERENTIAL EQUATIONS	
Parallel multigrid solver for 3D anisotropic elliptic problems U. Gärtel	37
Parallel solution of tridiagonal linear systems A. Krechel, H.-J. Plum, K. Stüben	49
A comparison of parallel multigrid strategies S. Vandewalle, R. Piessens	65
TOOLS AND ENVIRONMENT	
LOTOP — A load generator for parallel computers T. Bemmert, P. Hofstetter, T. Ludwig	83

Communication benchmarks for the iPSC/2 L. Bomans, D. Roose	93
--	----

A control replay scheme for distributed systems J.-L. Pazat	105
--	-----

BASIC ALGORITHMS IN LINEAR ALGEBRA

Updating and downdating the inverse Cholesky factor on a hypercube multiprocessor E. Chu, A. George	117
---	-----

Parallel triangularization of sparse matrices on distributed memory multiprocessors J. Duato, J. Pons	133
---	-----

Computing a rank-identifying QR factorization on a distributed-memory multiprocessor C.H. Bischof	147
---	-----

LANGUAGES AND OPERATING SYSTEMS

S: the CHEOPS operating system J.M. Lepine, F. Rubi	161
--	-----

Mapping programs on regular distributed architectures G.-R. Perrin, P. Clauss, S. Damy	175
---	-----

A language for the design of regular parallel algorithms P. Gachet, C. Mauras, P. Quinton, Y. Saouter	189
--	-----

NEURAL NETWORKS

Parallel implementations of neural network simulations A. Petrowski, L. Personnaz, G. Dreyfus, C. Girault	205
--	-----

Parallelization of a neural learning algorithm on a hypercube J. Bourrelly	219
---	-----

LINEAR ALGEBRA AND SCIENTIFIC COMPUTING

Distributed block algorithms for the triangular Sylvester equation with condition estimator B. Kågström, P. Poromaa	233
Scientific parallel computing with transputers networks A. Cosnau	249
Embedding a honeycomb topology onto a hypercube D. Deserale, Y. Dupont	263

INTERCONNECTION NETWORKS

De Bruijn and Kautz networks: a competitor for the hypercube? J.-C. Bermond, C. Peyrat	279
Efficient communication in polynode systems K.T. Malowanec, H. Scheidig, M.F. Schneider, R. Spurr	295
Performance analysis of broadcasting in hypercubes P. Fraigniaud	311
How to use an incomplete binary hypercube for fault tolerance M. Sultan Alam, R.G. Melhem	329

POSTERS

Computation of the 2-terminal network reliability on a iPSC2 by means of Monte-Carlo method M. El Kahadiri, R. Marie, G. Rubino	345
Study of a parallel preconditioner for a large-scale finite-element problem on a distributed memory machine (hypercube) Y.H. De Roeck	347
LU-decomposition on the iPSC/2 hypercube C.A. Addison	349
Data broadcast on a local memory, message passing computer L.R. Hagen, C.A. Addison, J.M. Cook	351

Parallelization of iterative solution methods and preconditioners for non-diagonally dominant, block-tridiagonal systems of equations S. Holmgren, K. Otto, L. Frändén	353
Adaptive block techniques for matrix factorization on distributed-memory computers H. Strauss, W. Rönsch	355
MODULOR: a modular reconfigurable highly parallel architecture D. Comte, Ch. Fraboul, J.-Y. Rousselot, P. Siron	357
Implementing sorting on a hypercube A.G. Ferreira, M. Gastaldo	359
Localisation — the next imperative after vectorisation O. Buneman	361
Parallelizing computer algebra: first explorations on an NCube/4+ machine A. Fekken, D. Hilhorst, E.J.H. Kerckhoffs	363
Analysis of basic numerical routines on the FPS T20 hypercube J.-Y. Blanc, P. Michallon, B. Tourancheau, D. Trystram, F. Vincent	365
Multicomputer simulator J. Duato, A. González	367
A comparison of stochastic search techniques on the Intel iPSC/2 multiprocessor J.E. Brandenburg	369
Fast matrix solver on the iPSC/2 with concurrent I/O facility E. Kushner, R. Young	371
Orthogonalization and singular value decomposition on a hypercube B. Philippe, B. Vital	373
A parallel development environment for the iPSC hypercube T. Bemmerl, R. Gebhart, P. Ginzinger, T. Ludwig	375
Distributed computers in the design and analysis of distributed algorithms (position paper) ADP Research Team	377

Cube-sharing in the hypercube Ch. Duff, G. Saucier	379
Partition shapes for explicit difference methods: slices vs rectangles M. Thune	383
Actor languages and distributed algorithms L. Litzler, M. Trehel	385
DAIRE – distributed artificial intelligence research environment P.J. Busalacchi, J.R. Slagle	387
An hypercube architecture for concurrent object-oriented computing J.-C. Heudin	389

AN ASSESSMENT OF SECOND-GENERATION MULTICOMPUTERS

Charles L. Seitz

Department of Computer Science
California Institute of Technology
Pasadena, California 91125, USA

Invited Presentation

Abstract

By 1990, new single-chip processors will allow medium-grain multicomputers to reach performance levels of tens of MIPS and tens of Mflops per node. Low-dimension cut-through networks of custom-VLSI routing chips will achieve communication bandwidths of 100 MBytes/s between nodes. Although message latency at the handler level of the node operating system will be reduced to a few microseconds, context switching and other inevitable multiprogramming overhead will prevent the reduction of message latencies at the user-program level to below about 200 instruction times. The low-level, portable programming systems will support internal file systems, heterogeneous configurations, and numerous new features, such as global operations across sets of cohort processes; however, performance and portability considerations will assure that the node operation systems will continue to be relatively small and streamlined. The substantial advantages that can be obtained with high-level compiler-based programming systems, both for programming productivity and for performance, will continue to provide a great opportunity for advancing the utility of distributed-memory concurrent computers.

The research described in this presentation was sponsored in part by the Defense Advanced Research Projects Agency, DARPA Order number 6202, and monitored by the Office of Naval Research under contract number N00014-87-K-0745; and in part by grants from Intel Scientific Computers and Symult Systems, Inc.

NON NUMERICAL ALGORITHMS

COMPONENT LABELLING ON A DISTRIBUTED MEMORY MULTIPROCESSOR

Hugo EMBRECHTS[†], Dirk ROOSE[†] and Patrick WAMBACQ[‡]

Department of Computer Science
 Katholieke Universiteit Leuven
 Celestijnenlaan 200A.
 B-3030 Heverlee, Belgium.

We present a parallel component labelling algorithm for binary pictures. It is based on a "divide-and-conquer" strategy, which makes it suitable for coarse-grained parallel machines. The algorithm was tested on an Intel iPSC[®]/2 hypercube. We give timings and efficiency results for artificial pictures, for which a perfect load balance can easily be obtained. Furthermore we study the load imbalance that occurs in some realistic pictures. We present upper bounds for the load imbalance and we indicate how one can achieve load balance.

1. INTRODUCTION

Binary pictures are two-dimensional arrays whose elements are called pixels and can have two values: one or zero (foreground or background).

The component labelling problem consists of assigning a label to each foreground pixel of a picture, so that two foreground pixels are assigned the same label, if and only if, these two pixels are connected by a path of neighbouring foreground pixels. We assume 4-connectivity. This means that horizontal and vertical neighbours are considered to be neighbouring and diagonal neighbours are not.

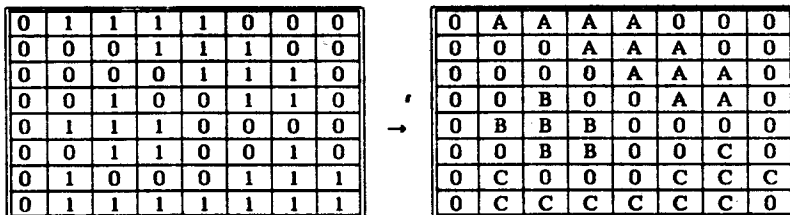


Figure 1. Component labelling of a binary picture.

[†] Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3030 Heverlee, Belgium.

[‡] ESAT/M12, Department of Electrical Engineering, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, B-3030 Heverlee, Belgium.

The set of all pixels that has been assigned a certain label is called a component. In Fig. 1 we present an example with three components, labelled with the letters A, B and C.

We present a component labelling algorithm that has been written for a coarse-grained parallel machine. This means that the number of atomic units of the processed data (i.e. : the number of pixels in the picture) is much larger than the number of processors of the machine. The algorithm has been coded and tested on the Intel iPSC/2 hypercube, a distributed memory machine. For an algorithm suited for a fine-grained parallel machine we refer to Ibrahim [6] and Cypher [2].

2. THE SEQUENTIAL LABELLING ALGORITHM

The main idea is to initially assign a label to every foreground pixel of the picture, such that pixels of different components are assigned a different label and pixels of the same component are assigned, as much as possible, the same label. The labels are then members of equivalence classes with equivalence relation "belongs to the same component as". While assigning labels to the pixels a so called equivalence table is constructed that indicates which labels belong to the same equivalence class. Afterwards this table is used to obtain a consistent labelling.

The pixels are traversed one by one from top to bottom and from left to right. A foreground pixel gets a label that hasn't been used yet, if its left and upper neighbour are background pixels. If one or both of those neighbours are foreground pixels, then their label is taken over. If this leads to a contradiction, i.e. if the left neighbour has label l_1 and the upper has label l_2 and $l_1 \neq l_2$, then the smallest of the two is taken over, and in order to inscribe the equivalence of l_1 and l_2 in the equivalence table, a process known as Union-Find is applied [9]. The generated labels are natural numbers starting from 1.

After the first traverse the equivalence table has to be reorganised, such that each label is associated with a representative label of the equivalence class to which it belongs. Then the picture is traversed once again to replace each label by this representative.

3. THE PARALLEL ALGORITHM

3.1 Introduction

Parallelism is introduced by dividing the picture into a number of non-overlapping subregions. In a first phase the sequential algorithm is executed in each subregion. At the end of this phase only the boundary pixels are relabelled using the equivalence table. The inner pixels are only relabelled at the end of the algorithm (phase IV), when the labels in the equivalence table have been replaced by global labels.

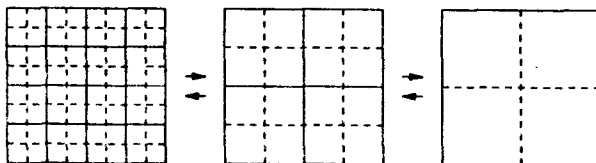


Figure 2. An example of the partitioning of a picture and the joining of its subregions in different steps. The common boundaries of the subregions that are joined are drawn with dashed lines.