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

INRIA
Institut National de Recherche en
Informatique et en Automatique

N·H
C P C

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

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 Aérospatiales (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 tc 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/SMAI (Société de Mathématiques Appliquées et Indùstrielles)
ONERA (Office National d'Etudes et de Recherches Aérospatiales)

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 Umea, Sweden
P. Leca, ONERA – Chatillon, France
H. Leroy, INRIA/IRISA – Renñes, 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

Invited Conference

NON NUMERICAL ALGORITHMS

PARTIAL DIFFERENTIAL EQUATIONS

TOOLS AND ENVIRONMENT

BASIC ALGORITHMS IN LINEAR ALGEBRA

LANGUAGES AND OPERATING SYSTEMS

NEURAL NETWORKS

LINEAR ALGEBRA AND SCIENTIFIC COMPUTING

INTERCONNECTION NETWORKS

POSTERS

Hypercube and Distributed Computers
F. André and J.P. Verjus (Editors)
Elsevier Science Publishers B.V. (North-Holland), 1989

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

Hypercube and Distributed Computers
F. André and J.P. Verjus (Editors)
Elsevier Science Publishers B.V. (North-Holland), 1989

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.

0	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	0
0	0	1	0	0	1	1	0
0	1	1	1	0	0	0	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	1	1	1	1	1	1

→

0	A	A	A	A	0	0	0
0	0	0	A	A	A	0	0
0	0	0	0	A	A	A	0
0	0	B	0	0	A	A	0
0	B	B	B	0	0	0	0
0	0	B	B	0	0	C	0
0	C	0	0	0	C	C	C
0	C	C	C	C	C	C	0

Figure 1. Component labelling of a binary picture.

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

[‡] ESAT/MI2, 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 'eft 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.

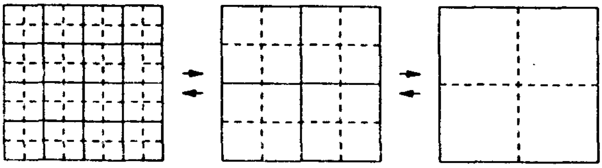

Figure 2. An example of the partitioning of a picture and the joining of its subregions in different steps. The common boundaries of the subregions that are joined are drawn with dashed lines.