DIMACS

Technical Report

Workshop

on

Interconnection Networks

and

Mapping and Scheduling

Parallel Computations

February 7-9, 1994

DIMACS

Rutgers University

Piscataway, New Jersey 08903
USA

Preface

The interconnection network is one of the most basic components of a massively parallel computer system. Each system consists of hundreds or thousands of processors, interconnected to work cooperatively on computations. One of the central problems in parallel computing is the task of mapping a collection of processes onto the processors and routing network of a parallel machine. Once processors are mapped onto the physical processors, it is critical to schedule computation within and communication among processors so that the necessary inputs for a process would be available where and when the process is scheduled to be computed.

In the context of the 1993-94 DIMACS special year on Massively Parallel Computation, funded jointly by the National Science Foundation and the New Jersey Commission for Science and Technology, a three day workshop on Interconnection Networks and Mapping and Scheduling Parallel Computations was held on February 7-9, 1994. The workshop focuses on the interconnection networks of parallel architectures of today and of the near future, examining the task of mapping and scheduling parallel computations on these parallel machines. Subject areas covered include network topologies, network properties, message routing, network embeddings, network emulation, mappings, and efficient scheduling.

The workshop was held at DIMACS at Rutgers University, Piscataway, New Jersey. DIMACS is the National Science Foundation science and technology center for discrete mathematics and computer science. It is a consortium of Rutgers and Princeton Universties, AT&T Bell Laboratories, and Bellcore.

There were forty-nine presentations in the twelve scheduled sessions of the Workshop. Over one hundred and fifteen scientists, mathematicians, and engineers from inside and outside DIMACS, and from fourteen countries attended the workshop. Participants included researchers from universities and research laboratories, as well as practitioners who are involved with the design, implementation, and application of massively parallel computer systems.

This report contains the set of titles and abstracts for the talks presented: the speaker of a co-authored paper is marked with an "$\star$". The report also contains the program and schedule of the Workshop. This report will be widely distributed to all researchers upon request. In particular, each participant of the Workshop will receive a copy of this DIMACS Technical Report. Complete and refereed papers based on the presentations will be published as a proceedings volume by the AMS in the DIMACS series. The Workshop was coorganized by:

D. Frank Hsu (Fordham Univ.; NYAS)
(hsu@murray.fordham.edu)

Arnold Rosenberg (U. Mass., Amherst)
(rsnbrg@cs.umass.edu)

Dominique Sotteau (LRI-CNRS)
(sotteau@lri.fr)

D. Frank Hsu

Arnold Rosenberg

Dominique Sotteau

Workshop Program

Tuesday, Feb. 8, 1994

Wednesday, Feb. 9, 1994

10:50	**William Y. C. Chen**, Los Alamos National Laboratories *Restricted Routing and Wide Diameter of the Cycle Prefix Network*	40
11:15	**Fred Annexstein**, University of Cincinnati *Hypercubic Hamiltonicity Revisited*	41
11:40	**Sotirios Ziavras**, New Jersey Institute of Technology *Investigation of Reduced Hypercube Interconnection Networks for Massively Parallel Computers*	42
12:05	**Siang Wun Song**, Universidade de São Paulo *Towards the Construction of Disjoint Cycles on the Hypercube*	43
12:30	Lunch	
2:00	**David Greenberg**, Sandia National Laboratories *Allowing the Machine to Fit the problem*	44
2:25	**Xing-de Jia**, Southwest Texas State University *Extremal Cayley Digraphs on Finite Cyclic Groups*	45
2:50	**R. Harbane**, Université de Paris-Sud *Fault Tolerant Kautz Network*	46
3:15	**Vittorio Scarano** , University of Massachusetts at Amherst *A Sublogarithmic Delay Execution of Programs PRAM on the Hypercube*	47
3:40	Break	
4:00	**Luisa Gargano**, Universita di Salerno *Fault-Tolerant Information Diffusion*	48
4:25	**Qiao Li**, University of Science and Technology of China *Restricted Connectivity and Restricted Fault Diameter of Interconnection Networks*	49
4:50	**Vipul Gupta**, Rutgers University *Efficient Mappings of Communication Patterns in an Iterconnection Cached Network for Multicomputers*	50
5:15	**R. Diekmann**, University of Paderborn *The HS-Heuristic: A Powerful New Approach to Graph Partitioning*	53
5:40	**Luděk Kučera**, Charles University *Harmonic Loops and Strings or How to Interconnect at Random*	55
6:05	**Workshop ends**	

Diameters, Eigenvalues and Routing

F. R. K. Chung

Bell Communications Research
Morristown, New Jersey 07960
USA

e-mail: `frkc@bellcore.com`

Abstract

We will discuss the relations of eigenvalues, diameters and routing in graphs with applications to communication networks and approximation algorithms.

Shuffled Tree Based
Hierarchical Interconnection Networks

Omar H. Karam ohkaram@eos.ncsu.edu

Dharma P. Agrawal* dpa@ncsu.edu

Department of Electrical and Computer Engineering
Box 7911
North Carolina State University
Raleigh, NC 27695-7911
USA

Tel: (919) 515-3984
Fax: (919)515-5523

Abstract

In this presentation, we introduce a class of Fully Shuffled Tree (FST) based Hierarchical Interconnection Networks (HIN's), wherein an r-ary tree is used to link disjoint clusters. The shuffle feature ensures that nodes within a cluster have no common parents, resulting in multiple intracluster disjoint paths and a fault tolerant environment. We illustrate the use of FST by clustering various schemes, including ring, hypercube, and full connection. We derive relations for the network parameters (diameter, degree of node, average distance, cost, fault tolerance, etc.) and compare them with existing networks. We also compute the execution complexity of Descend/Ascend algorithm, indicating the superiority of FST. A very low cost Incomplete Shuffled Tree (IST) configuration is also defined and analyzed by assigning 2 nodes of a cluster as gateway nodes.

Conflict-Free Access to Constant-Perimeter Rectangular Subarrays

Doreen L. Erickson[⋆,1]

Southern College of Technology
Marietta, GA 30060-2896
USA

e-mail: `erickson@sct.edu`

Charles J. Cobourn

University of Waterloo
Waterloo, Ontario
CANADA

Abstract

In a parallel computing environment, we consider conflict-free access to constant-perimeter rectangular subarrays, using a natural formulation in terms of latin squares. For parallel matrix computations, there are frequently portions of data, referred to as templates, that one wants stored and retrieved in so that one can operate on the data simultaneously with each processor working on one element of the template. If more than one processor attempts to retrieve data from a single memory module during the same memory cycle, a memory conflict arises. When the array is stored to allow the desired templates to be accessible from the memory modules without memory conflicts, the array is conflict-free for that set of templates. In this paper, we examine a set of structured templates where the number of template instances defined by a template grows with respect to the maximum size of a template. In particular, we examine the set of constant-perimeter rectangular subarrays.

A square is *perimeter rectangular conflict-free* p_{rcf} if it is conflict-free for all rectangular subarrays whose perimeter is less than or equal to $2p$. If p is even, the problem is to provide conflict-free access to all rectangular subarrays of size $(\frac{p}{2} - i) \times (\frac{p}{2} + i)$ for $-\frac{p}{2} \leq i \leq \frac{p}{2}$. We show that the necessary number of modules is $\frac{p^2}{2} - p + 1$, and there is a linear skewing scheme that realizes this bound. If p is odd, the problem is to provide conflict-free access to all rectangular subarrays of size $(\lfloor\frac{p}{2}\rfloor - i) \times (\lceil\frac{p}{2}\rceil + i)$ for $-\lfloor\frac{p}{2}\rfloor < i < \lceil\frac{p}{2}\rceil$. We show that the necessary number of memory modules is $\lfloor\frac{p}{2}\rfloor^2$, and there is a linear skewing scheme that realizes this bound. Furthermore, the linear results for the constant-perimeter rectangular subarrays hold when the subarrays are stretched by a factor of v, where v is relatively prime to the number of memory modules.

---

[1]research conducted as part of first author's Ph.D. at the University of Waterloo.

Becoming a Better Host Through Origami: A Mesh Can Be More Than Rows and Columns

Eric J. Schwabe[2]

Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208-3118
USA

e-mail: `schwabe@eecs.nwu.edu`

Abstract

Endowing a communication network with the ability to realize arbitrary communication patterns is an expensive proposition, both in hardware and in system software. One might instead ask whether a system can be built that performs well for a given application program. In this talk we look at the question of when a set of communication patterns is suitable for fast realization on a given network. In particular, we look at which patterns are realizable quickly on a mesh. Contrary to common wisdom, transpose is efficiently realizable on a mesh. However, some other important important patterns, such as perfect shuttle, are not.

[2]Joint work with David S. Greenberg and James K. Park

Some Practical Aspects of Designing Network Chips for Parallel Computers

Chris Jesshope

Department of Electronics and Electrical Engineering
University of Surrey
Guildford, Surrey
Gu2 5XH, UK

e-mail: `jesshope@ee.surrey.ac.uk`

Abstract

This talk will briefly review the MP1 network chip that has been designed and fabricated at the University of Surrey. This chip is synchronous packet router with support for a wide range of facilities, such as point-to-point and broadcast message delivery. This will include some of the limitations of this design in terms of both performance and application.

One of those limitations was that the chip was designed synchronously and with the current convergence between distributed and parallel computers this represents a major restriction in its use. We therefore made the decision in the design of a follow-up project to design an asynchronous router chip. This talk will focus on the design of this successor and the methodology used to ensure both that the design was correct and that performance was not compromised. We have made extensive use of a delay insensitive algebra in the specification of this design. This algebra will be introduced and the specification and implementation of a number of components from which we can design a range of routing devices will be presented. The resulting chips will have 1-2 orders of magnitude more performance when compared to the MP1 design, which is necessary in order to match the pace of microprocessor technology. We envisage applications of these chips in parallel computer architectures, distributed systems, such as high-speed LAN's and even as components of ATM based systems.

PVM and MPI:
Tools for Concurrent Computing

Jack Dongarra

Oak Ridge National Laboratory
University of Tennessee
Knoxville, Tennessee 37996-1301
USA

e-mail: `dongarra@cs.utk.edu`

Tel: (615) 974-8295
Fax: (615) 974-8296

Abstract

Wide area computer networks have become a basic part of today's computing infrastructure. These networks connect a variety of machines, presenting an enormous computing resources. In this talk we focus on developing methods and tools which allow a programmer to tap into this resource. In this talk we describe PVM and HeNCE, tools and methodology under development that assist a programmer in developing programs to execute on a parallel computer.

We will also describe a proposed standard message passing interface for MIMD distributed memory concurrent computers called MPI. The design of MPI has been a collective effort involving researchers in the United States and Europe from many organizations and institutions. MPI includes point-to-point and collective communication routines, as well as support for process groups, communication contexts, and application topologies. While making use of new ideas where appropriate, the MPI standard is based largely on current practice.

Systematic Synthesis of Knapsack Problems onto Fixed Sized Arrays with Lower Dimensions

Graham M. Megson

Department of Computing Science
University of Newcastle-Upon-Tyne
UK

e-mail: `Graham.Megson@newcastle.ac.uk`

Abstract

The production of regular computations using algorithmic engineering techniques is beginning to play an important role in the synthesis of massively parallel and VLSI processor arrays. In this paper we examine two issues in the mapping process. Firstly the problem of coping with so-called run-time dependencies and secondly the task of deriving fixed sized arrays of any given dimensions onto processors topologies with specified interconnection primitives. The techniques can be formally proved correct and we illustrate them by considering the complete automatic mapping of the well known Knapsack problem onto a 1-D array with a fixed number of processors.

Highly Adaptive Wormhole Routing Algorithms for n-Dimensional Torus

Jose Duato

Facultad de Informatica
Universidad Politecnica
P.O. Box 22012, 46071 Valencia
SPAIN

e-mail: `jduato@aii.upv.es`

Tel: (34) (63) 877573
Fax: (34) (63) 877209

Abstract

Deadlock avoidance is a key issue in wormhole networks. A first approach consists of removing the cyclic dependencies between channels. Many deterministic and adaptive routing algorithms have been proposed based on that approach. The absence of cyclic dependencies is a necessary and sufficient condition for deadlock-free deterministic routing. However, it can be relaxed for adaptive routing. A more powerful approach was proposed by us. It only requires the absence of cyclic dependencies on a connected channel subset. The remaining channels can be used in almost any way.

In this talk, we show that there exists a more relaxed condition for deadlock-free adaptive routing. This condition is the key for the design of maximally adaptive routing algorithms. In particular, we apply the former condition to the design of highly adaptive routing algorithms for n-dimensional torus.

Perpetual Gossiping

Arthur Liestman*
School of Computing Science
Simon Fraser University
Burnaby, B.C. V5A 1S6
CANADA
art@cs.sfu.ca

Dana Richards
National Science Foundation
4201 Wilson Blvd., Room 1145
Arlington, VA 22230
USA

Abstract

Gossiping is the information dissemination problem that exists when each member of a network knows a unique piece of information and must transmit it to every other member. Gossiping is accomplished by a sequence of calls made between pairs of individuals, such that during each call the two people involved exchange all of the information they know at that time; and such that at the end of the sequence of calls, everybody knows everything.

Broadcasting is a related problem in which one member of the network has a message which is to be transmitted to all of the other members by a series of calls over the network. Each of these calls requires one unit of time; any member of the network can participate in at most one call per time unit; and any member can only call an adjacent member. Gossiping can be viewed as simultaneous broadcasts from each member.

In previous work in this area, gossiping was assumed to be a one-shot problem, that is, a unique piece of information was known to each member and the goal was to produce a calling scheme to make all of the information known to all of the members. Perhaps a more realistic setting is to assume that new information arises at each member from time to time which must then be communicated to all of the other members. One way to accomplish this is to have the members gossip at fixed intervals. However, gossip schemes may require a large amount of information flow and this periodic interruption may significantly effect the performance of the overall network at these times. If the gossip scheme is spread out over time and then repeated, the problem of degrading the performance of the network may be alleviated but the result may be that the network members are not informed of new information in a timely fashion. We suggest an alternate approach, *perpetual gossiping*, where gossiping is considered to be an ongoing process which uses a limited amount of the network in each time unit but which ensures that information is disseminated in a timely fashion.

In this talk, I present some preliminary results on this new problem.

Throughput of Butterfly Networks Without Buffers

Burkhard Monien[3] bm@uni-paderborn.de

Martin Dietzfelbinger[4] dietzf@markov.informatik.uni-dortmund.de

Reinhard Lüling[*, 3] rl@uni-paderborn.de

Abstract

To realize a massively parallel architecture, one has to balance the computational performance of the processors and the routing efficiency of the communication network according to the needs of the targetted applications.

Routing efficiency is mostly measured using network throughput (maximal number of messages which can be delivered per time step) and network latency as a function of the message injection rate. These figures have been determined for a number of networks (hypercube-like networks, multistage-networks) and different routing strategies (store and forward, wormhole, cut-through) by a huge number of simulations. Theoretical work has mostly been done for models where every processor holds one message and sends it to a random destination.

In this talk we present results for the butterfly network without buffers at the routing switches. This is of highly practical relevance, since routing switches used today do not provide large buffer space, but can hold only one message per incoming channel.

Today's network flow algorithms (wormhole routing, cut-through routing) make it possible to achieve very low latency routing for underloaded processor networks. Using randomization techniques and local/global adaptive routing, there is a very small gap between injection rates holding this low latency routing times and injection rates leading to a nearly total blocking of the network (see the simulation results in [?]). On the other hand, routing latency is often dominated by message startup time. Therefore, at least for nowadays architectures, the network throughput is the more important parameter for the application performance.

In this talk we will present theoretical investigations of the network throughput using the steady-state model. An upper bound will be presented which is very close to the throughput determined by experiments.

References

[1] Funke, R., R. Lüling, B. Monien, F. Lücking, H. Blanke-Bohne, *An Optimized Reconfigurable Architecture for Transputer Networks*, **Proc. of the 25th Hawaii Int. Conf. on System Sciences (HICSS)**, 1992.

[3]Department of Mathematics and Computer Science, University of Paderborn, GERMANY.
[4]Department of Mathematics and Computer Science, University of Dormund, GERMANY.

[2] Monien, B., R. Lüling, F. Langhammer, *A Realizable Efficient Parallel Architecture, Proc. of 1st Int. Heinz Nixdorf Symposium 1992*, **Lecture Notes in Computer Science**, Vol. 678.

On Communication Schemes for Passive Optical Star Networks[5]

P. Berthomé and A. Ferreira*

CNRS-Laboratoire LIP, CNRS-URA 1398
ENS Lyon, 46, Allée d'Italie
69364 Lyon Cedex 07, FRANCE

e-mail: {berthome,ferreira}@lip.ens-lyon.fr

Abstract

For massively parallel architectures, the hardware complexity of the interconnection network is much higher than that of the processing units, employing most of the hardware involved. In many optical communication networks, Passive Optical Star (POS) technology using wavelength division multiplexing access (WDMA) offers optical multiple access channels that allow significant reduction in the complexity of the specialized routers connected to the processors, along with a substantial reduction in the intermediary latencies for one-to-many communications.

The WDMA technology allows us to share the enormous bandwidth of optical fibers among several messages, through the creation of multiple channels by the use of different wavelengths. This can be implemented by tunable feedback laser diodes for the transmitters, and wavelength tunable filters for the receivers. In this ongoing work, we start by surveying the several models of POS proposed in the literature and then show how to use the processors capability of freely tuning the wavelength they want to access at a given moment, in order to design efficient global communication schemes for some of the surveyed models of such an interprocessor network. Such schemes are commonly used in all areas of parallel computing, ranging from neural networks to linear algebra, and are as follows.

- One-to-all broadcasting a message from one node to all other nodes.
- One-to-all multiple broadcasting: broadcasting multiple messages from one node to all other nodes.
- All-to-all communication: broadcasting from each node to every other nodes (also known as gossiping).

We show how to implement these operations in such a way that there is a machine-dependable tradeoff between the system parameters for number of available wavelengths (λ) and the tuning time (D). For all of them, our algorithms come very close to the lower bound for communication costs, while keeping the tuning time as low as possible.

[5]Partially supported by ANM and C3, and by the French-Israel project on Optical Models for Parallel Computing.

Permutation Routing for Cayley Graphs

Gene Cooperman* **and Larry Finkelstein**

College of Computer Science
Northeastern University
USA

e-mail: `gene@ccs.neu.edu`

Abstract

Many important interconnection networks have the form of Cayley graphs or Cayley coset graphs. Previous methods using computational group theory are reviewed that allow optimal point-to-point routing information for such networks to be pre-computed and efficiently recovered at the cost of $\log_3(2)$ computer bits per node. Such techniques have been demonstrated on $PSL(2, p)$ (jointly with Namita Sarawagi), where a Cayley graph was used such that for all nodes, the subgraphs of radius 9 (and corresponding routing algorithms) is identical up to isomorphism, for $p > 37$ (graphs with more than 27,417 nodes). Such families of graph are also scalable in three space dimensions. The important case of permutation routing is discussed using similar techniques, but using a group representation in which the group to act on itself by right multiplication.

Interval Routing Schemes

Pierre Fraigniaud[⋆, 6,7] and **Cyril Gavoille**[7]
pfraign@lip.ens-lyon.fr gavoille@lip.ens-lyon.fr

LIP-CNRS
Ecole Normale Supérieure de Lyon
69364 Lyon Cedex 07
FRANCE

Tel: (33) (72) 72 84 33
Fax: (33) (72) 72 80 80

Abstract

Interval routing was introduced to reduce the size of the routing tables: a router finds the direction to forward a message by determining the interval that contains the destination address of the message, each interval being associated to one particular direction. This way of implementing routing function is quite attractive but very few was known about the topological properties that should satisfy a network to admit an interval routing function satisfying particular constraints (shortest paths, limited number of intervals associated to each direction, etc.). In this talk, we deeply investigate the study of the interval routing functions. In particular we characterize the networks that admit a *linear* or a *linear strict* interval routing function with only one interval per direction. We also derive practical tools to measure the efficiency of an interval routing function (number of intervals, length of the paths, etc.), and we describe large classes of networks that admit an *optimal* (linear) interval routing function. Finally, we derive the main properties that satisfy the popular networks used to interconnect the processors of a distributed memory parallel computer.

[6]The first author received the support of the Centre de Recerca Matemàtica, Institut d'Estudis Catalans, Bellaterra, Spain.
[7]Both authors are supported by the research programs ANM and C3.

Routing in Optical Networks: The Problem of Contention

Leslie Ann Goldberg

Algorithms and Discrete Mathematics Department
Sandia National Labs
P.O. Box 5800
Albuquerque, NM 87185-1110
USA

e-mail: `lagoldb@cs.sandia.gov`

Tel: (505) 845-7631
Fax: (505) 845-7442

Abstract

In massively parallel computers, processors communicate by exchanging messages over a network of communication links. Current technology restricts the network to be of low degree: each processor in the network can communicate directly with only a few others, and the remainder must be reached indirectly by routing messages along a sequence of links. The emerging technology of optical communication challenges the assumption that the network must be of low degree. In particular, the huge bandwidth of the optical medium can be divided so that each processor has its own channel for receiving messages and each processor can send on any channel. (It is already possible to build such an interconnection network for networks of size 50 and it is conceivable that such networks will be built with thousands of processors in the future.) Even though such an interconnection network is a complete graph, there remains the constraint that no processor can receive messages simultaneously from two other processors without corruption. That is, *contention* makes it difficult to route messages. The extent of this difficulty is partially captured by Goldberg, Jerrum, and MacKenzie, who have shown that *every* distributed randomized algorithm for routing h-relations on a completely connected optical network of size n requires $\Omega(h + \sqrt{\log\log n})$ communication steps. The fastest known distributed randomized algorithm for solving this problem (due to Goldberg, Jerrum, Leighton, and Rao) takes $O(h + \log\log n)$ communication steps.

Our goal is to understand the effect of contention on optical routing problems. We believe that every distributed randomized algorithm for routing h-relations on a completely connected optical network of size n requires $\Omega(h + \log\log n)$ communication steps. However, we have been unable to prove this conjecture. More is known for specific classes of optical routing algorithms. In particular, Goldberg, Jerrum, Leighton, and Rao considered a class of algorithms for routing h-relations that they called *direct* algorithms. These algorithms

can only send the original messages in the h-relation, and they can only send the messages directly to their final destinations, with no forwarding. They showed that direct randomized algorithms requires $\Omega(h + \log n)$ steps to route h-relations on completely connected optical networks of size n. In this talk, we consider a different class of algortihms which we call *token routing algorithms.* Like direct algorithms, these algorithms can only send the original messages in the communication problem. The messages can be forwarded arbitrarily. However, the messages are viewed as tokens and a processor cannot send a message unless it holds the token associated with that message. We consider a contention resolution problem which can be solved deterministically in $O(\log\log n)$ steps by token routing algorithms on a completely connected optical network of size n. We show that $\Omega(\log\log n)$ steps are required by all such algorithms. As a corollary, we show that routing h-relations on a completely connected optical network of size n requires $\Omega(h+\log\log n)$ communication steps for deterministic token routing algorithms.

Communications in Optically Interconnected Parallel Computer Systems

Mounir Hamdi

Department of Computer Science
Hong Kong University of Science & Technology
Clear Water Bay, Kowloon
HONG KONG

e-mail: `hamdi@cs.utk.hk`

Tel: (852) 358-6984
Fax: (852) 358-1477

Abstract

The unique properties of optical signal transmissions in waveguides, namely, unidirectional propagation and precisely predictable path delays, allow for the construction of optical buses on which many processors can write their messages simultaneously. Further, the high bandwidth of waveguides also facilitate the simultaneous transmission of multiple messages at different wavelengths. This has motivated the concept of Wavelength Division Multiplexed Pipelined Buses (WDMPB), which significantly diverges from conventional exclusive access buses and offers both possibilities and challenges for significantly improving the efficiency of communications in parallel computer systems. Different Star coupled organizations will be presented as the basis for WDMPB. Further, efficient multiaccess protocols will be presented for each organization and analyzed analytically and through simulation. Moreover, the embedding of a diverse well known interconnection networks, including binary trees, hypercubes, and pyramids will be presented and evaluated. Finally, the incorporation of these star coupled organizations into variants of k-ary n-cube networks in order to increase their communication bandwidth will be given along with the implementation of very fast algorithms on them.

Embedding $k - D$ Meshes into Optimum Hypercubes with Dilation $2k - 1$

Said Bettayeb[8], Zevi Miller[9], Tony Peng[10] and Hal Sudborough[⋆,11]

Abstract

The principal result of this paper is that, for every k, and for every k-dimensional mesh M, there exists a one-to-one embedding of M into the optimum size hypercube with dilation at most $2k - 1$.

Previous work has shown that every $2 - D$ mesh can be embedded into its optimum size hypercube with dilation at most 2 [C], every $3 - D$ mesh can be embedded into its optimum size hypercube with dilation at most 5 [C2], and, for all $k > 3$, every $k - D$ mesh can be embedded into its optimum size hypercube with dilation at most $4k + 1$ [C].This work in turn improved on earlier work showing, for example, that dilation 2 is possible for most $2 - D$ meshes [BMS]. Also, techniques described in [BMS] embedded, for $k > 2$, many $k - D$ meshes, but not all, into their optimum size hypercube with dilation k.

All of the work cited so far considers one-to-one embeddings of meshes into hypercubes. Many to one embeddings have also been considered. In [MS] it was shown, for every $m > 1$, that every $2 - D$ mesh can be embedded into a hypercube whose size is 2^m times optimum with dilation 1 by a $(2^m + 1)$-to-1 embedding. For example, every $2D$ mesh M can be embedded into the hypercube with 1/2 as many points as the optimum for M by a 3-to-1 embedding with dilation 1. This work also showed that (a) all $3 - D$ meshes M can be embedded into the hypercube with 1/2 as many points as the optimum for M by a 3-to-1 embedding with dilation 2 and (b) all $3 - D$ meshes M can be embedded into the hypercube with 1/4 as many points as the optimum for M by a 5-to-1 embedding with dilation 3.

As indicated in the abstract, our current results offer a significant improvement: For all $k > 1$, and for every $k - D$ mesh M, there is a dilation $2k - 1$ one-to-one embedding of M into its optimum hypercube. Thus, all $3 - D$ meshes can be embedded with dilation at most 5, and in general, for arbitrary $k - D$ meshes, the upper bound on dilation is improved substantially from $4k + 1$ to $2k - 1$.

References

[BMS] S. Bettayeb, Z. Miller, and I. H. Sudborough, "Embedding Grids into Hypercubes", **J. Computer and System Sci.** (1992).

[8]Department of Computer Science, Louisiana State University, Baton Rouge, Louisiana.
[9]Department of Mathematics and Statistics, Miami University, Oxford, Ohio.
[10]Mathematics and Computer Science, Creighton University, Omaha, Nebraska.
[11]Computer Science Program EC 3.1, University of Texas at Dallas, Richardson, Texas 75083-0688, hal@utdallas.edu.

[C] M. -Y. Chan, "Embedding of Grids into Optimal Hypercubes", **Siam J. Computing**, 20, 5(1991), pp. 834-864.

[C2] M.-Y. Chan, F. Chin, C.-N. Chu, and W. K. Mak, "Dilation 5 Embedding of 3-Dimensional Meshes into Hypercubes", to appear in **Proc. 1993 IEEE Symp. on Parallel and Distributed Computing**.

[L] F. T. Leighton, **Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes**.

[MS] Z, Miller and I. H. Sudborough, "Compressing Grids into Hypercubes", to appear in **Networks**.

Embedding by Exclusions:
An Approach of Emulation and Parallel Algorithms Implementation

Dominique Barth

LaBRI, URA 1304
Université Bordeaux I
351 Cours de la Liberation
33405 Talence, FRANCE

e-mail: `barth@labri.u-bordeaux.fr`

Tel: (33) (56) 84 60 47
Fax: (33) (56) 84 66 69

Abstract

We deal here with networks modelling parallel architectures. Emulating the execution of a parallel algorithm in an interconnection network G into another one H can be done by using the *embedding* of G into H: it is a couple (p, R), where p is an injective application of $V(G)$ into $V(H)$ (the sets of vertices), and where R associates to each edge (u, v) of G a chain $R(u, v)$ linking $p(u)$ and $p(v)$ in H. To evaluate the quality of an embedding, we use two parameters: the *dilation* equal to the greater length of a chain $R(u, v)$; the *congestion* equal to the maximal number of such chains a same edge in H. Then it is known that a step of communication in G, consisting for some vertices linked by an edge in exchanging data, can be emulated in at most dc steps in H, where d is the dilation of the embedding and c the congestion.

The vertices of H and G can represent different kind of nodes in the architecture, such as routers, processors, memories. Then if we use an embedding, we must take care of the nature of the vertex of G embedded in a vertex of H: for example, a router of H cannot emulate a processor of G. Moreover, if we want to emulate into H one (or a set of) given algorithm(s) running in G, we can optimize the embedding: for example, if two processors of G never do processing simultaneously, they can be emulated by a same processor of H.

We so propose a first approach of these problems, by taking care of the nature of each vertex of a network, and by giving a modelling of the execution of an algorithm in a network. We then define a new kind of embedding, called *embedding of exclusions* leading us to do more realistic emulations. We conclude by giving some examples and applications.

Efficient Dynamic Broadcasting Schemes-
Analysis, Simulations and Experiments

Larry Rudolph

Department of Computer Science
The Hebrew University
Jerusalem 91904
ISRAEL

e-mail: `rudolph@cs.huji.ac.il`

Abstract

Fast dissemination of data among subsets of processors is essential for any efficient large scale parallel computation. Standard broadcast mechanisms, such as a static broadcasting tree, are highly sensitive to the availability of individual processors. A processor that joins the broadcasting process late may cause large number of processors to waste much time waiting for the data.

We present dynamic broadcasting techniques for the two major types of parallel computers: message passing and shared memory machines. Our algorithms adapt to when individual processors become available. As a result, for any schedule of the processors, the algorithms perform close to optimal, both in the expected delay of processors and in the total amount of communication (or memory access requests) generated by the algorithm. We present theoretical analysis of the performance of the algorithms, together with extensive simulation results that validate the asymptotic analysis for systems of practical size. The algorithm is shown to improve the performance of the All-Pairs Shortest Paths application on a 16-node parallel processor.

Graph Embeddings from Hamming Bases

Dan Pritikin

Department of Mathematics & Statistics
Miami University
Oxford, Ohio 45056
USA
e-mail: `dpritikin@miavx1.acs.muohio.edu`

Abstract

An algorithm due to Aiello and Leighton dynamically embeds arbitrary binary trees into hypercubes, with small dilation and congestion and (with high probability) small load. The operation and performance level of the algorithm both depend upon first specifying a convenient choice of basis for each binary Hamming code. Making use of particular labellings of double-rooted complete binary trees, in this report we answer an open problem (posed in Leighton's text *Introduction to Parallel Algorithms and Architectures* by improving upon the previously best known choices of bases, and thereby reduce the upper bounds for the dilation and congestion associated with embeddings produced by the algorithm.

Fault-Tolerant Meshes with Small Degree

Jehoshua Bruck, Robert Cypher*, Ching-Tien Ho

IBM T. J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

e-mail: `cypher@watson.ibm.com`
e-mail: `ho@ibm.com`

Abstract

We present several constructions for fault-tolerant two-dimensional mesh architectures. The constructions are designed to tolerate k faults while maintaining a healthy n by n mesh as a subgraph. They utilize several novel techniques for obtaining trade-offs between the number of spare nodes and the degree of the fault-tolerant network.

We consider both worst-case and random fault distributions. In terms of worst-case faults, we give a construction that has constant degree and $O(k^3)$ spares. This is the first construction known in which the degree is constant and the number of spares is independent of n. In terms of random faults, we present several new degree-6 and degree-8 constructions and show (both analytically and through simulations) that they can tolerate large numbers of randomly placed faults.

On Consecutive-d Digraphs[12]

D. Z. Du[⋆,13] **D. F. Hsu**[14] **D. J. Kleitman**[15]

dzd@cs.umn.edu hsu@murray.fordham.edu

Abstract

The concept of a consecutive-d digraph was proposed by Du, Hsu and Hwang as a generalization of de Bruijn digraphs, Kautz digraphs, and their generalizations given by Imase and Itoh and Reddy, Pradhan and Kuhl. In this paper we determine the connectivity of consecutive-d digraphs.

[12]Supported in part by contracts AF OSR-86-0078 and NSF DMS-86-06225 and NSF CCR-9208913.

[13]Computer Science Department, University of Minnesota, Minneapolis, MN 55455, USA, and Institute of Applied Mathematics, Academia Sinica, Beijing.

[14]Department of Computer & Information Science, Fordham University, Bronx, New York 10458–5198.

[15]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139.

Embeddings and Emulations Involving Meshes of Trees

Louxin Zhang

Department of Computer Science
University of Waterloo
CANADA N2L 3G1

e-mail: `lzhang@neumann.uwaterloo.ca`

Abstract

Emulation and embedding problems for interconnection networks such as meshes and hypercubic networks have been extensively studied in recent years. The goal for the *emulation problem* is to emulate t_G steps of any computation on the guest network G in t_H steps on the host network H in a way such that the slowdown t_H/t_G is as small as possible. The slowdown of the emulation is at least as large as N_G/N_H, the ratio of the number of nodes of the networks G and H. It is said to be *work-preserving* if the slowdown is $O(N_G/N_H)$. Moreover, if the slowdown t_H/t_G is $O(1)$, we say the emulation is performed in *real-time*. An *embedding* of a network G into a network H is an assignment of the nodes of G to nodes of H and a mapping of edges of G to paths in H between the images of their endpoints. It is well known that a good embedding of the guest network into the host network often leads to an efficient emulation.

In this work, we will focus on the emulation and embedding involving the mesh of trees network. We prove that a N-node mesh can be embedded in real-time on a mesh of trees, even though any embedding of a N-node mesh into meshes of trees with constant load requires dilation $\Omega(\log N)$. On the other hand, we show that a mesh can perform a work-preserving emulation of a mesh of trees, but not in real-time.

The 2-dimensional meshes of trees can be one-to-one embedded into like-sized hypercubes and de Bruijn graphs with constant dilation and congestion. Since all bounded-degree hypercubic networks can emulate each other in real time, the butterfly network can efficiently emulate all computations on meshes of trees. In this work, we strengthen this fact by proving that a 2-dimensional mesh of trees can be one-to-one embedded into a butterfly with constant dilation and congestion. Such an embedding gives us an automatic way to compile and run algorithms designed for meshes of trees on the butterfly network without loss of efficiency.

A New Combinatorial Approach to Optimal Embeddings of Meshes into Meshes and Hypercubes

Rakesh M. Verma[16]

Computer Science Department
University of Houston
Houston, TX 77204, USA

e-mail: `rmverma@cs.uh.edu`

Tel: (713) 743-3348
Fax: (713) 743-3335

Abstract

Many important problems in computer science can be reduced to embedding rectangular grids (or meshes) into other graphs. We present a novel, general, combinatorial approach to (one-to-one) embedding meshes into their ideal meshes and optimal hypercubes. In contrast to earlier approaches of Aleliunas and Rosenberg, and Ellis, our approach is based on a special kind of doubly stochastic matrix. We prove that any rectangular grid can be embedded into its ideal rectangular grid with dilation equal to the ceiling of the compression ratio, which is both *optimal* up to a multiplicative constant and a substantial generalization of previous work. We also show that any rectangular grid can be embedded into its nearly ideal square grid with dilation at most 3. Finally, we show that any rectangular grid can be embedded into its *optimal* hypercube with *optimal* dilation 2, a result previously obtained, after much research, through an ad hoc approach. Our results also imply optimal simulations of 2-dimensional mesh-connected parallel machines by hypercubes and mesh-connected machines, where each processor in the guest machine is simulated by exactly one processor in the host.

[16]This is a joint work with H. Liu and Dr. S. Huang. The speaker's work was partially supported by NSF.

Synchronization Costs of Repetitive Tasks

Quentin F. Stout

Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122
USA

e-mail: `qstout@eecs.umich.edu`

Abstract

Many scientific and engineering codes consists of loops repeated many times. Typically within the loops there is significant parallelization possible, but often the task times have some random element, due to random aspects of the underlying simulation and/or random behavior of the parallel hardware or systems software (such as communication bottlenecks or random cache replacement). In such situations, overly synchronized code can have greatly reduced efficiency, though often the cause is attributed to the apparent load imbalance.

On many shared memory machines, the preferred programming style casues difficulties because it forces a "global synchronization" of all processors after each iteration. We develop a simple graph model allowing us to evaluate the improvements possible by using a more appropriate synchronization which has processors wait only for those other processors from which they need new results. This "local synchronization" is what has historically been utilized on distributed memory machines. We compare the local and global synchronization and show that the local rapidly converges to a significantly smaller time per iteration. This rapid convergence is important because often there are some global synchronizations, such as convergence checks, which still must occur after only a few iterations.

We also examine several of the sources of system randomization encountered on the shared memory KSR1 machine.

Routing n-Masks to Improve SHIFT and DISTRIBUTE Operations on the Butterfly

Petrisor Panaite[17]

LRI, CNRS URA 410
Bât. 490
Université de Paris-Sud
91405 Orsay Cedex
FRANCE

e-mail: `petr@lri.lri.fr`

Abstract

The wrapped butterfly graph BF_n is considered as the interconnection network of a multicomputer with a SIMD computation scheme. In spite of its constant degree, the algorithmical properties of wrapped butterfly are closed to those of the hypercube. We introduce a class of permutations (called n-masks) which can be routed in only n steps within all levels of BF_n simultaneously. Using the n-masks, we present two on-line algorithms which perform SHIFT operation in at most $\lfloor \frac{3n}{2} \rfloor + 1$ steps and DISTRIBUTE operation in $5n - 1$ steps.

[17]This work was done in Laboratoire de Recherche en Informatique (LRI), Université Paris-Sud, where the author was a BGF Fellow.
The permanent address of the author is: Racultatea de Informaticaā, Universitatea "AI. I. Cuza", Laqi, RO-6600, Romania

Scheduling with Communication: A Survey of Recent Results

Philippe Chrétienne

Université Pierre et Marie Curie
Institute Blaise Pascal – Laboratoire, LITP
FRANCE

e-mail: `phc@litp.ibp.fr`

Abstract

This talk is intended to give an overview of the most significant results concerning scheduling with communication and no task duplication. After having specified a basic scheduling model, we particularly point out the complexity aspects of these problems and their incidence on approximation algorithms. We conclude by some remarks on what should be future research directions in this field.

Building Scalable Expanders with Hierarchical Random Wiring

Frederic T. Chong*
ftchong@ai.mit.edu

Eric A. Brewer brewer@lcs.mit.edu
Thomas F. Knight, Jr. tk@ai.mit.edu
F. Thomson Leighton ftl@math.mit.edu

Artificial Intelligence Laboratory
Laboratory for Computer Science
545 Technology Square
and
Department of Mathematics

Massachusetts Institute of Technology
Cambridge MA 02139
USA

Abstract

Recent work has shown that randomly wired expander-based networks possess tremendous fault tolerance and very low congestion even under high loads. Unfortunately, the wiring complexity of such networks becomes physically problematic as they become large. We introduce *hierarchical random wiring*, a method which we have recently proven can scale small expanders to larger ones while maintaining practical physical construction. We also have empirical evidence that the resulting scalable expanders perform as well as randomly wired expanders.

This talk concentrates upon the design options enabled by the constrained randomness of heirarchical construction. Specifically, we discuss scalable designs that allow incremental system expansion and exploit physical locality. We discuss multiplexing to decrease cabling cost and efficiently use network bandwidth. We introduce *randomized multiplexing*, a method which goes beyond random wiring and provides both efficient bandwidth utilization and inexpensive implementation.

Acknowledgments: This work was supported in part by National Science Foundation grant CCR-8716884; by ARPA contracts N00014-91-J-1698 and N00014-92-J-1799; by Air Force Contract AFOSR F49620-92-J0125; by an Office of Naval Research Graduate Fellowship; by an equipment grant from Digital Equipment Corporation; and by grants from AT&T and IBM.

Scheduling as a Parallel Programming Tool

Paul Feautrier

Laboratoire PRiSM
Université de Versailles St. Quentin
45 Avenue de Etats Unis
78035 Versailles Cedex
FRANCE

e-mail: `Paul.Feautrier@masi.uvsq.fr`

Tel: (33) 1 39 25 40 66
Fax: (33) 1 39 25 40 57

Abstract

A schedule is a closed form function which gives the date at which each operation of a program may be executed under the hypotheses that there are as many (virtual) processors as necessary, and that each operation executes in unit time. From such a schedule, one may deduce an ordinary parallel program by multiplexing the virtual processors on the physical processors.

There are many kinds of schedules: in order of increasing complexity and power, wavefronts, affine schedules, piecewise affine schedules and multidimensional piecewise affine schedules. To be acceptable, a schedule has to satisfy the following causality constraint: the source of each dependence is scheduled at an earlier date than its destination. It is only for the class of multidimensional affine schedules that it is possible to prove that any sequential program has a schedule which satisfies the causality function.

To construct a schedule, one first select a prototype, i.e. a function in the selected class with indeterminate coefficients. One then deduces from the causality condition a set of constraints (hopefully, linear constraints) on the indeterminate coefficients. Lastly, one select a solution by a process akin to linear programming. There are several plausible objective functions which give schedules of increasing quality: minimum latency schedules, minimum delay schedules, earliest start time schedules. Multidimensional affine schedules may be found by an efficient algorithm somewhat similar to Allen and Kennedy's parallelization algorithm.

This parallelization method is very powerful. It can be proved that it subsumes all methods which are based on affine transformations of the iteration domain. On the other hand, its complexity is rather high, and more work on optimization and approximation techniques is needed before it can be used in practical compilers.

Scheduling Structured and Unstructured Computation [18]

Apostolos Gerasoulis[19]

Department of Computer Science
Rutgers University
New Brunswick, NJ 08903
USA
e-mail: `gerasoulis@cs.rutgers.edu`

Abstract

The scheduling of parallel computation has received considerable attention recently. Many new algorithms have been proposed, especially for static scheduling. Even though static scheduling is restrictive, it can utilize message-passing architecture better than dynamic scheduling when information is available statically. It is of interest to know for what kind of applications static scheduling is practical. In this talk we will briefly discuss a new scheduling system named PYRROS, that automatically schedules and produces code for parallel architectures such as nCUBE-2 and INTEL/i860. We will then discuss the performance of PYRROS on structured and unstructured computation. The applications range from dense matrix solvers, such as Gaussian Elimination, to sparse matrix solvers and particle simulations using hierarchical fast methods. For example, "almost" linear speed up has been obtained for the Greengard-Rokhlin 2D-Nbody algorithm on the nCUBE-2S hypercube for uniform distributions of particles. The size of the problems solved, ranging from few thousand to 1 million particles on 16Mbyte–128 processor machine, clearly demonstrating the usefulness and practicality of automatic scheduling systems.

Papers related to PYRROS scheduling algorithms and theory can be obtained via anonymous ftp from cs.rutgers.edu in the directory pub/gerasoulis/reports. PYRROS is also available as an experimental tool for the nCUBE-2S and INTEL/i860 machines.

---

[18] Joint work with:
Tao Yang, Department of Computer Science University of California at Santa Barbara.
e-mail: `tyang@cs.ucsb.edu`.
Jia Jiao, Department of Computer Science, Rutgers University.
e-mail: `jjiao@cs.rutgers.edu`.

[19] The work presented here was in part supported by ARPA contract DABT-63-93-C-0064, by the Office of Naval Research under grant N0001493101114, by a startup fund from the University of California at Santa Barbara and by NSF grant DMS-8706122. The content of the information herein does not necessarily reflect the position of the Government and official endorsement should not be inferred.

Sorting and Selection on Interconnection Networks

Sanguthevar Rajasekaran

Department of Computer & Information Science
University of Pennsylvania
Philadelphia, PA 19104
USA

e-mail: `raj@central.cis.upenn.edu`

Abstract

Sorting and Selection are well studied comparison problems. In this talk we identify the core ideas underlying the design of parallel sorting and selection algorithms over a variety of network models. Our talk will include both randomized and deterministic algorithms. We also introduce a new selection scheme and illustrate its applicability with examples.

Virtual Topologies: A New Concurrency Abstraction for High-Level Parallel Languages

James Philbin
NEC Research Institute
4 Independence Way, Princeton, NJ 08450
USA
e-mail: `philbin@research.nj.nec.com`

Tel: (609) 951 2749
Fax: (609) 951 2496

Abstract

In this talk we will introduce a new programming abstraction called Virtual Topologies that have three important advantages for concurrent programming (both parallel and distributed): (1) They allow the programmer to express an algorithm using its natural topology rather than the topology of the target hardware. (2) They allow parallel algorithms to be portable across different hardware architectures. And (3) They make parallel programs more efficient by promoting communication locality.

We have developed an operating system, called *Sting*, that supports first class threads and virtual processors. It is straightforward to build virtual topologies using these abstractions.

We have implemented several different virtual topologies (trees, meshes, butterflies, and hypercubes) and used them to run benchmark programs. Our results show that virtual topologies offer significant performance improvements because of improved communication locality.

Statement of Interests

Our research interests are in the areas of parallel languages, systems and compiler design. Our current research focuses on building a unified foundation for parallel and distributed computing.

We are particularly interested in investigating high-level language abstractions for expressing communication and synchronization properties among lightweight threads of control, and defining flexible runtime mechanisms to permit a variety of diverse parallel programming paradigms to be efficiently supported on a variety of architectures.

Currently, we are also in the process of implementing software and hardware mechanisms to effectively support a distributed shared virtual memory environment on disjoint memory architectures. Our research involves implementing novel memory management protocols, and network controllers to support such protocols efficiently.

A Simple and Efficient Parallel Sorting Algorithms on Star Graphs[20]

Hong Zhu

Department of Computer Science
Fudan University
Shanghai 200433
CHINA

e-mail: `hong.zhu@mhs.fudan.crn.cn`

Tel: (86) (21) 548 2082
(86) (21) 549 1069
Fax: (86) (21) 549 3232

Abstract

This paper gives a new explanation of extended Gray code. With this explanation we list the $n!$ permutations on n-star graph, by this list algorithms we introduce two modes: RM mode and CMj modes, $1 \leq j \leq n$. We design a simple and efficient sorting algorithm on n-star graph, while sorted in order of principle and the properties of n-shuffled or 2-shuffled of sequences. The complexity of this sort of algorithm is $O(n^3 \log n)$.

Key words: Sorting; Parallel algorithms; Complexity; Gray code; Star graphs.

[20]This research is partially supported by a Grant from Slovak Academy of Sciences No. 88, by EC Cooperative Action IC 1000.

Bus Interconnection Networks [21]

Jean-Claude Bermond*

I3S, CNRS, Université de Nice-Sophia Antipolis
bât 4, 250 rue A. Einstein, 0650 Valbonne, FRANCE

e-mail: `bermond@mimosa.unice.fr`

Fahir Ö. Ergincan
Department of Computing and Information Science
Queen's University at Kingston
K7L 3N6, CANADA

Abstract

We consider bus interconnection networks in which processors communicate through buses. In particular we are interested in constructing large (Δ, D, r)-bus networks in which each vertex has maximum degree Δ, each bus can interconnect at most r processors and the diameter is D. We give some new constructions in particular we propose a generalization of the classical de Bruijn networks and also Kautz networks based on the concept of directed hypergraphs. We also study some of the properties of these hypergraphs like connectivity.

[21] This work has been done within the group RUMEUR of the GDR-PRC PRS.

The Steiner Problem in Hypercubic Networks

Zevi Miller

Department of Mathematics & Statistics
Miami University
Oxford, Ohio 45056
USA

e-mail: `zmiller@miavx1.acs.muohio.edu`

Abstract

In this talk we survey some of our recent work on the graphical Steiner problem (GSP) restricted to the hypercube and related networks. Let X be a set of vertices in some connected graph G. We define $L(X, G)$ to be the minimum number of edges in any subtree of G which spans X. The GSP is the problem given G, X, and an integer bound k, determine if $L(X, G) \leq k$. Denote the n-dimensional hypercube and n-dimensional butterfly by $Q(n)$ and $B(n)$ respectively. We obtain the following results.

1. An exact formula for $L(X, Q(n))$ when $|X| \leq 5$.

2. Let $W(k, n)$ be the set of vertices in $Q(n)$ having weight k, where we normalize $k + 1 \leq \frac{n}{2}$. We apply a result of Frankl and Rödl on the generalized Turan problem for hypergraphs to show that

   $$L(W(k+1, n), Q(n)) \leq \binom{n}{k+1} + (1 + o(1))\frac{\log k}{k}\binom{n}{k}$$

   as $k \to \infty$.

   We also show that the second term on the right is within a factor of $\log k$ from optimal.

3. NP-completeness of the GSP for $Q(n)$ even when restricted to sets X consisting of the origin together with an arbitrary set of points of weight 2.

4. Bounds and exact values for $L(X, B(n))$ for certain X.

Fractiling: Combining Factoring and Tiling to Schedule Parallel Loops

Susan Flynn Hummel

Polytechnic University & IBM T.J. Watson Research Center
Six Metro Tech Center
Brooklyn, NY 11201
USA

e-mail: `hummel@mono.poly.edu`

Abstract

The fundamental tradeoff when scheduling parallel loops is processor load imbalances versus overhead. The overhead has two components: synchronization and communication. A scheduling scheme that reduces synchronization is *factoring*, when iterates are dynamically scheduled in decreasing *size* chunks. The sizes of the chunks are chosen so that there is high probability that they will be completed before the optimal finishing time. A technique for reducing communication, called *tiling*, statically partitions the iteration space into tiles whose *shape* is chosen to maximize data reuse and locality. When a data-space partition is specified, as in the HPF dialect of Fortran, then an "owner computers" rule is used to tile the iteration space.

Fractiling combines factoring and tiling: pre-processor tiles are statically created and linked together. Each processor initially executes its local tile in *fractiles* whose sizes are dynamically determined according to a global factoring rule. The simplicity of the factoring rule allows decreasing-size fractiles to be allocated according to common tiling rules using a self-similar numbering scheme. If a processor finishes its tile early, it obtains additional fractiles by following the tile links. Fractiling retains the load balancing properties of factoring and the data reuse properties of tiling. The incorporation of the HPF data-distribution directives into factoring scheduling to schedule nested parallel loops on shared- and disjoint-memory machines will be described.

Concurrent Processing of Linearly Ordered Data Structures On Hypercube Multicomputers[22]

Sajal K. Das*

Center for Research in Parallel and Distributed Computing
Department of Computer Science
University of North Texas, Denton, TX 76203-3886
USA

Joydeep Ghosh

Department of Electrical and Computer Engineering
University of Texas, Austin, TX 78712-1084
USA

Ajita John

Department of Computer Science
University of Texas, Austin, TX 78712
USA

Abstract

This talk presents a simple and effective method for the concurrent manipulation of linearly ordered data structures on hypercube multicomputers. The method is based on the existence of a *pruned binomial search tree* rooted at any arbitrary processor-node of the binary hypercube such that (i) every edge of the tree corresponds to a direct link between a pair of hypercube nodes, and (ii) the tree spans any arbitrary sequence of n consecutive nodes containing the root, using a fan-out of at most $\lceil \log_2 n \rceil$ and a depth of $\lceil \log_2 n \rceil + 1$. Search trees spanning non-overlapping processor lists are formed using only local information, and can be used concurrently without contention problems. Thus, they can be used for performing operations such as *broadcast* and *merge* simultaneously on sets with non-uniform sizes. Extensions of the tree to k-ary n-cubes and faulty hypercubes are presented. Applications of this concurrent data structure to low and intermediate-level image processing algorithms, and for dictionary operations involving multiple keys are outlined.

[22]This research is supported by an NSF Initiation Grant MIP 9011-787, and Texas Advanced Research and Technology Program grants under TARP Award No. 003594003 and TATP Award No. 0035658386. For comments or reprints, please contact ghosh@pine.ece.utexas.edu or das@cs.unt.edu.

Restricted Routing and Wide Diameter of the Cycle Prefix Network

William Y. C. Chen*, **Vance Faber and Emmanuel Knill**

C-3 Mail Stop B281
Los Alamos National Laboratories
Los Alamos, NM 87545
USA

e-mail: `vxf@c3.lanl.gov`

Tel: (506) 665-0897
Fax: (506) 662-0858

Abstract

The cycle prefix network is constructed as a Cayley coset digraph of the symmetric group, and it can also be represented in terms of sequences like the hypercube and the de Bruijn graph. This network has been shown to enjoy many remarkable communication properties. In this paper, we prove that the cycle prefix network has small wide diameter, a measurement which has caught a lot of recent attention. We also study the diameter change of the cycle prefix digraph when a few directions are avoided.

Hypercubic Hamiltonicity Revisited

Fred Annexstein

Department of Computer Science
University of Cincinnati
Cincinnati, OH 45221
USA

e-mail: annexste@san.uc.edu

Tel: (513) 566-1807

Abstract

The Hamiltonicity of hypercubic networks has been well studied, beginning a century ago. The recursive structure enjoyed by these networks,including the hypercube, butterfly, and the de Bruijn, has permitted the simple specification of Hamiltonian paths and cycles. In this talk, we review the historical development of these constructions, and we present new results concerning the determination of "ranking" algorithms that, in essence, embed these Hamiltonian paths and cycles using only local information. Ranking algorithms have applications to the efficient implementation of network emulations via SIMD-style parallel algorithms.

Investigation of Reduced Hypercube Interconnection Networks for Massively Parallel Computers

Sotirios G. Ziavras

Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ 07102
USA

e-mail: `ziavras@herta.njit.edu`

Tel: (201) 596-5651
Fax: (201) 596-5680

Abstract

The direct binary hypercube is one of the most frequently used topologies for the interconnection of processors in distributed-memory parallel computing systems. It offers low diameter and is so robust that it can very efficiently emulate a wide variety of other frequently used interconnection networks. However, the major drawback of the hypercube is the increase in the number of communication channels for each processor with an increase in the total number of processors in the system. This drawback has direct effect on the VISI complexity of the hypercube network and restricts inadvertently its adaptation for the construction of high-performance massively parallel systems. This work investigates the properties of a new topology which is produced from the hypercube by a uniform reduction in the number of edges for each node. This edge reduction technique produces interconnection networks with lower complexity than hypercubes, while it maintains to high extent the very powerful hypercube properties. An extensive comparison of the proposed reduced hypercube (RH) topology with the regular direct binary hypercube is presented. It is also shown that multiple copies of the popular cube-connected cycles interconnection network can be emulated simultaneously by an RH, with dilation equal to one. Finally, routing algorithms are presented and the efficiency of embedding the torus and ring topologies into RH's is also investigated.

Towards the Construction of Disjoint Cycles On the Hypercube

S. W. Song[23]

University of São Paulo – IME
Department of Computer Science
Rua do Matão 1010, São Paulo
SP– 05508-900 – BRAZIL

e-mail: `song@ime.usp.br`

Abstract

The problem of finding edge-disjoint cycles on a hypercube can be important in fault-tolerant distributed computing. For applications that require the usage of the nodes of the hypercube connected in a cycle, it is important to have simple algorithms to construct the edge-disjoint cycles. It is known that on a hypercube of dimension n, there exists $\lfloor n/2 \rfloor$ disjoint Hamiltonian cycles. Unfortunately, no simple algorithm is known to construct such cycles. We present an approach which might be interesting towards the obtention of algorithms for this purpose and present some preliminary results. Saad and Schultz have shown that hypercube or binary n-cube equivalent to the 4-ary $n/2$-cube which has $n/2$ dimensions and 4 nodes on each dimension. The construction method is based on the reflected Gray code. We first present a different method based on the "folding" of sub-cubes. This method has a nice property that makes it suitable to be used in divide-and-conquer algorithms. We then attempt to obtain cycles on the 4-ary $n/2$-cube. As a starting point, we consider $n = 4$, and we should find 2 Hamiltonian cycles. Then we attempt to construct cycles for $n = 6$, with the hope that we get some insight to get a general construction algorithm. For $n = 4$, Foregger gives an immediate construction of the desired cycles. For $n = 6$, we show the construction of 12 disjoint cycles of size 16 on a 4-ary 3-cube. It still remains to be shown the merging of these cycles to obtain the required Hamiltonian cycles. We show some kind of merging operations that may lead to the final result.

[23] Supported by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) - Proc. No. 93/0603-1 and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) - Proc. No. 306063/88-3 and PROTEM-CC/SP.

Allowing the Machine to Fit the Problem

David Greenberg

Sandia National Laboratories
Mail Stop 1423
P.O. Box 5800
Albuquerque, NM 87185-1423
USA

e-mail: `dsgree@cs.sandia.gov`

Tel: (505) 845-7601
Fax: (505) 845-7442

Abstract

Everyone would like a general purpose massively parallel computer. Unfortunately the cost of communications hardware seems to grow rapidly with the generality of the machine. If a program uses only one or two simple communications patterns then it is not difficult to build a machine which allows all communication to occur quickly. However, if the machine must be capable of servicing *any* communications patterns then it must have much more complex routing hardware and higher overall bandwidth.

In an attempt to reduce the cost of generality we suggest relaxing the requirement on the machine. Rather than asking that the parallel computer service any communication pattern *at any time* we ask only that it is possible (with a moderate time and work penalty) to reconfigure the machine to handle the specific communication patterns to be used by the next program using the machine. We discuss which patterns will then be easily configured and ways to allow the machine to be configured for quick changes between a few patterns.

Extremal Cayley Digraphs on Finite Cyclic Groups

Xing-De Jia[24]

Department of Mathematics
Southwest Texas State University
San Marcos, Texas 78666
USA
e-mail: jia@xingde.math.swt.edu

Tel: (512) 245-3445

Abstract

Let $Cay(m, A)$ denote the Cayley digraph on Z_m generated by A, where Z_m is the cyclic group of residues modulo m. Let $r(m, A)$ denote the average distance of $Cay(m, A)$. For any $r \geq 1$ and $k \geq 1$ define $m^\star(r, k)$ as the largest positive integer m such that the average distance of the Cayley digraph $Cay(m, A)$ is at most r for some set A with k elements.

In this paper, an asymptotic formula for $m^\star(r, 2)$ is proved and a lower bound for $m^\star(r, k)$ is also obtained for $k \geq 3$. Applications to the construction of optimal distributed loop networks is discussed in this paper. A lower bound of the average order of subsets for asymptotic bases in number theory is proved using the main theorem of this paper.

[24]This work was supported in part by a SWTU summer research grant.

Fault Tolerant Kautz Network

R. Harbane

Laboratoire de Recherche en Informatique
Bat. 490, Université de Paris-Sud Orsay 91405
FRANCE

e-mail: `harbane@lri.lri.fr`

Abstract

Given a graph $G_0 = (V_0, E_0)$ with $|V_0| = n$ and maximum degree Δ, we say that a graph $G_k = (V_k, E_k)$ on $n + k$ nodes is k-fault tolerant version of G_0, if we can embed G_0 in any n node induced subgraph of G_k.

Thus, if G_0 models an interconnection network, then G_k represents an interconnection network which is capable to sustain k-faults and still emulate G_0 without any performance degradation.

In this talk we will treat the case where the target graph G_0 is the Kautz graph $UK(d, D)$. For any integers d, D and k, we define a k-fault tolerant graph $UK_k(d, D)$ with maximum degree in $O(kd)$.

We also give a greedy procedure for designing k-fault tolerant graph G_k for any target graph G_0. This procedure constructs G_k in $O(|E_0|k^2)$ steps.

A Sublogarithmic Delay Execution of PRAM Programs on the Hypercube

Vittorio Scarano[25]

Department of Computer Science
University of Massachusetts at Amherst
Amherst, MA 01003, USA

e-mail: `vittorio@cs.umass.edu`

Abstract

In this paper, we study the problem of efficiently executing a CRCW PRAM program on a distributed memory parallel machine, where the interconnection network is the hypercube. A PRAM consists of N processors, each with its own segment of the "shared memory", communicating with one another through a complete network.

We show how to emulate a N-task CRCW PRAM program on a N-node hypercube with slowdown $O(\log\log N)$ with high probability, provided that the task i accesses variables in the memory of the task j with a relative frequency $f_{ij} = O\left(\frac{1}{N\log^2 N}\right)$.

[25] Joint work with Vincenzo Auletta and Alberto Negro, Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84081 Baronissi (SA) – ITALY

Fault-Tolerant Information Diffusion

Luisa Gargano

Università di Salerno
84081 Baronissi (SA)
ITALY

Abstract

In this talk I will focus on the determination of the values of the following functions:

$T_k(n) =$ minimum possible number of time units to broadcast a message in any network of n nodes having up to k non-functioning communication lines, and

$B_k(n) =$ minimum possible number of edges of a network of n nodes in which the broadcast can be performed in $T_k(n)$ time units even in presence of up to k non-functioning communication lines.

Restricted Connectivity and Restricted Fault Diameter of Some Interconnection Networks

Qiao Li

Department of Mathematics
University of Science and Technology of China
Hefei, 230026, CHINA

Abstract

When the interconnection networks are modeled by a graph, some graph parameters such as diameter, connectivity and fault diameter are critical to the performance evaluation of the network topology. Here we consider two not so well-known parameters. One is the restricted connectivity, which was first introduced by Esfahanian and Hakimi based on the assumption that all neighbors of any given vertex do not fail at the same time, and this parameter naturally leads to another parameter called restricted fault diameter. The last two parameters in conjunction with other well-known parameters may provide a more accurate measurement of network reliability properties.

In 1989 Esfahanian showed that the restricted connectivity of the ℓ-dimensional binary hypercube is $2\ell - 2$ and its $2\ell - 3$-restricted fault diameter is at most $\ell + 6$. In this paper, similar result for other good network families, such as the cube connected cycles, the undirected binary de Bruijn and Kautz networks, are derived.

Efficient Mappings of Communication Patterns in an Interconnection Cached Network for Multicomputers

Vipul Gupta*
vgupta@cs.rutgers.edu
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Eugen Schenfeld
eugen@research.nj.nec.com
NEC Research Institute, Inc.
4 Independence Way
Princeton, NJ 08540

Abstract

A fairly general approach to modelling parallel applications is to view them as a set of interacting processes and a collection of communication patterns[6]. In this model, an application proceeds in one or more phases each with a well defined communication structure. This communication structure can be represented by a communication graph in which vertices represent concurrently executing processes and edges denote interprocess communication. The communication graphs of many real applications display a phenomenon which we call *switching locality*, i.e. each process switches its communication between a small set of other processes. This is reflected in the sparse nature of communication graphs that arise frequently in parallel applications[3], e.g. 3-D meshes in computational fluid dynamics (CFD) applications and the binary tree pattern seen often in applications based on the "divide and conquer" algorithmic paradigm.

The Interconnection Cached Network (ICN) [5] is specifically designed to exploit switching locality in applications. Processing elements in the ICN are grouped into many small clusters. Each cluster has its own small (fast switching) crossbar which provides intracluster connectivity. Intercluster connectivity is provided by a single large (but slow switching) crossbar which connects between the smaller crossbars. The larger crossbar is used only to reconfigure the ICN according to the communication pattern of the application. It is switched infrequently. Message switching is accomplished by switching only the small crossbars. Using a clever partitioning of the communication graph, it is possible to map processes to PEs such that all communication paths can be established by switching at most two small crossbars. This not only keeps communication paths short but ensures that they can be set up rapidly. Both these factors contribute to overall efficiency of the network. When this happens, communication requests can be satisfied with small latency and we say that the switching neighborhood of each process has been *cached*. This is similar to the way in which memory caching allows memory references to be satisfied with small latency.

In order to gain all of the benefits described above, one needs to find a *bounded ℓ-contraction* of the communication graph. A *bounded ℓ-contraction* of a graph is a partition of its vertex set into subsets such that the following two constraints are satisfied:

- no subset contains more than ℓ vertices, and

- every subset has at least as many vertices as the number of other subsets it is connected to.

The problem of determining if a graph has a *bounded ℓ-contraction* is NP-complete for $\ell > 2$. In [1], Barak and Schenfeld present *bounded ℓ-contractions* for graphs representing classical communication topologies (such as rings, trees, grids, cube connected cycles and hypercubes) for small values of ℓ.

In this talk, we describe a heuristic algorithm based on simulated annealing for solving this partitioning problem. The algorithm takes an integer ℓ and a communication graph as input and produces a partitioning which approximates a *bounded ℓ-contraction.* Subsets that violate the second constraint mentioned above are called *bad* partitions. Communication involving nodes in these subsets, gets routed through longer paths (more than two hops). However, experiments indicate that for many communication graphs, the algorithm is able to find partitions in which the number and extent of such violations is very small[2].

As with any other implementation of simulated annealing, this one poses the challenge of choosing a 'good' mix of many algorithmic parameters. Our experiments point to interesting trade-offs available while making these choices. We also present the performance of this algorithm on different communication patterns. The results of Barak and Schenfeld[1] are used to evaluate the quality of partitions found by our algorithm. We conclude that our algorithm is an effective means for obtaining efficient mappings of communication patterns in the ICN. These mappings allow dynamic, low latency communication paths to be established quickly between communicating processes for better overall performance.

References

[1] Barak,A. and E. Schenfeld, *Embedding Classical Communication Topologies in the OPAM Architecture.* **Technical Report TR 90-12**, Department of Computer Science, The Hebrew University of Jerusalem, Israel, August 1990. (A shorter version of this report also appears in the **Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing**, 1991.

[2] Gupta, V. and E. Schenfeld, *A Heuristic Approach for Embedding Communication Patterns in an Interconnection Cached Parallel Processing Network.* **Proceedings of the 7th International Parallel Processing Symposium**, pp. 291-298, April 1993.

[3] Kung, H. T., *The Structure of Parallel Algorithms.* **Advances in Computers**. 19:65-112, 1980.

[4] Ramamurthy, B. and M. S. Krishnamoorthy, *Bounded p-contractibility in NP-complete.* **Technical Report 92-17**, Department of Computer Science, Renselaer Polytechnic Institute, Troy, New York, June 1992.

[5] Schenfeld, E., *The Optical Parallel Architecture Model (OPAM) Experiment – Project Plan and Motivations.* **Technical Report TR 88-17**, Department of Computer Science, The Hebrew University of Jerusalem, Israel. August 1988.

[6] Snyder, L., *Languages, Compilers and Run-Time Environments for Distributed Memory Machines*, Chapter Applications of the "Phase Abstractions" for Portable and Scalable Parallel Programming, Elsevier, 1992, pp. 79-102.

The HS-Heuristic: A Powerful Approach to Graph Partitioning

R. Diekmann*, B. Monien, R. Preis

Department of Mathematics and Computer Science
University of Paderborn
D-33095 Paderborn
GERMANY

e-mail: {diek,bm,rosy}@uni-paderborn.de

Abstract

The task of assigning large process networks or graph structured data items to massively parallel computing systems is one of the most important parts in the field of parallel programming. The related mapping problem is NP-complete and there exist almost no efficient sequential or parallel heuristics that solve this problem sufficiently.

With growing performance of interconnection networks and especially with the establishment of independent routing networks, it is appropriate to reduce the mapping problem to the task of partitioning the graph into a number p of equal sized (or weighted) clusters where p is the number of processors, and to minimize the number of edges crossing the partition boundaries. Because such partitioning problems are still hard to solve and there exist again only a very small number of sufficient heuristics, they are often handled by recursive bisection.

There exist a number of well known bisection algorithms. The most important are the Kernighan-Lin heuristic (KL) [6], Spectral Bisection (SP) [3,7], Simulated Annealing (SA) [2,5] and the geometric approach by Miller, Teng and Vavasis (see e.g. [1]).

As the mapping process is only a preprocessing to the parallel application that has to be executed, heuristics solving it have to be accurate, fast and as less space consuming as possible.

With the Helpful-Set (HS) we introduce a new method for improving graph partitions that is able to outperform most of the commonly used bisection heuristics not only in quality of the attained solutions but also in space and time requirements. Like the KL algorithm our heuristics is cased on local search: It tries repeatedly to improve a given bisection with the help of local rearrangements. We classify the nodes of the graph according to their number of external and internal edges (i.e. the ratio of incident edges crossing the cut). Based on this classification we search for k-helpful sets, i.e. sets of nodes that improve the cut size by k. If a k-helpful set is found on one side of the cut, we search for a balancing set of the same size in the other part of the graph. The balancing set should not increase the cut size by more than $k-1$ edges. Exchanging the two sets leads to an improvement of at least one in the cut size.

The idea to this heuristic is based on a technique used in [4] to show upper bounds on the bisection width of 4-regular graphs. In [4] it is shown that, provided some some conditions are fulfilled, whenever a 3-helpful set in a 4-regular graph is found, it is possible to find a balancing set of equal size that increases the cut size by at most 2 edges.

We compare our algorithm to various self-implemented versions of KL and SA [2] to results of the spectral method from Pothen et al. [7] and Henrickson and Leland [3] and to the results of the geometric approach from Miller et al. [1]. As test suite we use several self generated regular networks, especially de Bruijn networks, and a number of graphs from the Harwell-Boeing collection that are also used in [3] and [7].

The HS-Heuristic is also used for recursive bisection. As it is very fast we are able to perform a number of runs on different starting solutions at each level of recursion, thus improving the solution quality even more. It turns out that it is not necessary and often even not sensible to search for the very best cut sizes at the first levels of recursion. We start therefore with one run of the heuristic and increase the number of runs linear with increasing levels of recursion. Compared with the method of using equal numbers of runs at each level the linear increase saves running time but delivers the same solution quality.

References

[1] Bleloch, G. E., A. Feldmann, O. Ghattas, J. R. Gilbert, G. L. Miller, D. R. O'Hallaron, E. J. Schwabe, J. R. Shewchuk, S.-H. Teng, *Automated Parallel Solution of Unstructured PDE Problems.* Draft, CMU School of Computer Science.

[2] Diekmann, R., R. Lüling, J. Simon, *Problem Independent Distributed Simulated Annealing and Its Applications.* R. V. V. Vidal (ed.), *Applied Simulated Annealing*, **Lecture Notes in Economics and Mathematical Sciences**, Springer LNEMS 396 (1993), pp. 17-44.

[3] Henrickson, B., R. Leland, *A Multilevel Algorithm for Partitioning Graphs*, Technical Report SAND93-1301, Sandia National Laboratories, Oct. 1993.

[4] Hromkovič, J., B. Monien, *The Bisection Problem for Graphs of Degree 4 (Configuring Transputer Systems).* **Proc. of 16th Math. Foundations of Computer Science (MFCS'91)**, Springer LNCS 520, pp. 211-220.

[5] Johnson, D. S., C. R. Aragon, L. A. McGeoch, C. Schevon, *Optimization by Simulated Annealing: An Experimental Evaluation; Part 1, Graph Partitioning.* **Operations Research**, Vol. 37, No. 6, November- December 1989, pp. 865-893.

[6] Kernighan, B. W., S. Lin, *An Effective Heuristic Procedure for Partitioning Graphs.* **The Bell Systems Technical Journal**, Feb. 1970, pp. 291-308.

[7] Pothen, A., H. D. Simon, K. P. Liu, *Partitioning Sparse Matrices with Eigenvectors of Graphs*, **SIAM J. on Matrix Analysis and Applications 11**, 3 (1990), pp. 430-452.

Harmonic Loops and Strings or How to Interconnect at Random

Luděk Kučera[26]

Department of Applied Mathematics
Charles University
Prague, CZECH Republic

e-mail: `ludek@kam.ms.mff.cuni.cz`

Abstract

A new class of new interconnection networks for parallel computers is introduced. Networks are built using a randomized algorithm, and it is proved that, with large probability, both the diameter and the largest degree of a network is $O(\log n)$. Similarly, the congestion with respect to a routing problem given by an arbitrary permutation is $O(\log n)$ and there are strong indications that the same applies to the corresponding broadcast time.

Unlike most known connecting networks, our graphs can be obtain for arbitrary number of vertices, and the number of connections is not a function of the size of the network, but can vary in a wide range if an appropriate parameter of the generating algorithm is changed. The scalability and fault tolerance of networks are also excellent.

Results of computer simulation show that properties of this class of networks and the hypercube connecting pattern are very close.

[26]This is a preliminary report on ongoing research, which was supported by EC Cooperative Action IC–1000 (project ALTEC: *Algorithms for Future Technologies).*

List of Speakers (with Corresponding Page Number)