

2

AD-A257 946



Principal Investigator Name: Jeffrey D. Ullman
PI Institution: Stanford University
PI Phone Number: (415) 723-9745
PI Email Address: ullman@cs.stanford.edu
Contract Title: Design and Implementation of Parallel Algorithms
Contract Number: N00014-88-K-0166
Reporting Period: 10/90-9/91

1. NUMERICAL PRODUCTIVITY MEASURES

Refereed papers, submitted, not published: 11
Refereed papers published: 4
Unrefereed reports and articles: 3
Books or parts submitted, not published: 0
Books or parts published: 2
Patents filed, not granted: 1
Patents granted: 0
Invited presentations: 6
Contributed presentations: 6
Honors received: 2
Prizes or awards received: 0
Promotions obtained: 0
Graduate students supported: 3
Postdocs supported: 2
Minorities supported: 0

DTIC
ELECTE
NOV 19 1992
S E D

DISTRIBUTION STATEMENT

Approved for public release

Distribution Unlimited

338 550

92-29841



80

92 11 18 288

DTIC QUALITY INSPECTED

Principal Investigator Name: Jeffrey D. Ullman
 PI Institution: Stanford University
 PI Phone Number: (415) 723-9745
 PI Email Address: ullman@cs.stanford.edu
 Contract Title: Design and Implementation of Parallel Algorithms
 Contract Number: N00014-88-K-0166
 Reporting Period: 10/90-9/91

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	<i>per ltr</i>
By _____	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
<i>A-1</i>	

2. DETAILED SUMMARY OF TECHNICAL RESULTS

Coupled Execute/Control Processor Architecture

Andy Freeman has designed and simulated a new processor architecture for use in multiprocessors. The basic problem is designing processors that can cope with long memory latency. His approach is to couple two processors by two-way queues; neither processor has the capability associated with conventional processors. One executes only "straight-line code," i.e., computations, and the second executes only tests and branches. Through simulation he has shown the superiority of this architecture on applications that are similar to "vectorizable" computations, including those on which vector architectures perform well and some that are (slightly) too sequential for vector processors. No reports on this work have yet been produced.

Timestamping in Networks

Orli Waarts and Cynthia Dwork of IBM have looked at the question of assigning timestamps in an asynchronous network of processors (Dwork and Waarts [1991]). The model is that of shared memory, where the processors communicate through a memory accessible to all; the memory, like the processors, may be distributed. The problem is to issue tickets in a first-come-first served order when there are unlimited numbers of requests for service that may arrive asynchronously. Their solution uses time proportional to the number of processors involved, while previous solutions were superlinear.

A key problem to solve is avoiding a bottleneck at one memory location. Obvious approaches have the processors queue up for access to a counter that assigns timestamps. The solution of Dwork and Waarts uses instead a bounded timestamp, which is in the form of a vector of values, one for each processor. Each processor controls one component of the vector. In particular, we must be able to recycle timestamps so we do not confuse a new value with an old, identical value. We avoid this problem by allowing each processor to designate which of its values comes first; the others follow in a fixed, cyclic order.

Linearizable Counting

A related problem, called "linearizable counting," assigns consecutive integers to processors (timestamping assigns tokens, which might be integers, to processes) in a first-come-first served order. The algorithm of Herlihy, Shavit, and Waarts [1991] performs this assignment in time proportional to the number of processors, and does so in a way that does not produce a bottleneck at a shared memory location.

Processor Assignment

Yossi Azar and Joseph Naor, with R. Rom of IBM, looked at the problem of on-line assignment of tasks to processors (Azar, Naor, and Rom [1991]). That is, a sequence of tasks enter the system asynchronously, and each can be performed by any of a subset of the n processors. We must assign tasks to processors as they enter, and the object is to balance the load on the processors. They offer an algorithm that can be no worse than a factor $\log_2 n$, compared with a "clairvoyant" algorithm that can predict future demand for the various processors. They also show this ratio is best possible. Further, they consider the use of a randomized algorithm for the same assignment problem, and show that a ratio of $\log_e n$ is sufficient and best-possible.

Derandomization of Algorithms

Many of the best known parallel algorithms are probabilistic, in the sense that they perform well with high probability, but there is no guarantee they will finish in any particular amount of time. In many cases, it is possible to replace a probabilistic, parallel algorithm by a deterministic parallel algorithm by discovering a small set of bit strings that can represent possible sequences of "coin flips" (the steps that introduce the randomization). It is necessary that these small sets have certain properties of pseudo-randomness, in the sense that on any input there is at least one among them on which the probabilistic algorithm will terminate relatively quickly.

Azar, Motwani, and Naor [1991] address the problem of approximating arbitrary joint distributions of random variables. They construct, for any $\epsilon > 0$, a set of strings that is of size polynomial in both the number of variables and $1/\epsilon$, whose joint distribution approximates that of the given distribution, in the sense that none of the coefficients of the Fourier transform of the two distributions differ by more than ϵ . This result can be used in two ways. One, it is possible to run the original random algorithm using each of the strings of "coin flips" in the set, in parallel, terminating as soon as the first among them terminates. Second, it allows one to replace a large number of coin flips by a smaller number (the logarithm of the size of the set) that is "almost as random" as the original.

Bipartite Matching

Andy Goldberg and Serge Plotkin looked at the open question of sublinear (NC) algorithms for parallel bipartite matching (Fisher, Goldberg, and Plotkin [1991]). This problem, in addition to being one of the most important of the classical combinatorial problems, has special significance for parallel computation because a number of other important problems, such as depth-first search, are known to have NC algorithms if bipartite matching does. They show that there is an NC algorithm for a strong form of approximate matching, where, with different versions of the algorithm, they can guarantee a matching that comes within $1 - \epsilon$ of the maximum match for any $\epsilon > 0$.

Earlier work on using interior point methods in parallel algorithms has appeared in a journal (Goldberg, Plotkin, Shmoys, and Tardos [1991]).

Network Flow Problems

A survey of parallel algorithms for network flow was published (Goldberg, Tardos, and Tarjan [1990]). Also, earlier work on reducing the number of processors needed to solve network flow problems in parallel appeared in a journal (Goldberg [1991]).

Journal Publication of Old Results

Several other works, reported last year or earlier, have now appeared or been accepted for journals. These include Azar [1991a, b], Dolev et al. [1991], and Ullman and Yannakakis [1991].

Principal Investigator Name: Jeffrey D. Ullman
PI Institution: Stanford University
PI Phone Number: (415) 723-9745
PI Email Address: ullman@cs.stanford.edu
Contract Title: Design and Implementation of Parallel Algorithms
Contract Number: N00014-88-K-0166
Reporting Period: 10/90-9/91

3. REFERENCES

- Alon, N., J. Bruck, J. Naor, M. Naor, and R. Roth [1991]. "Constructing asymptotically good low-rate error-correcting codes through pseudo-random graphs," to appear in *IEEE Trans. Inf. Th.*
- Azar, Y. [1991a]. "Parallel merging of many ordered lists," to appear in *Theor. CS.*
- Azar, Y. [1991b]. "Parallel computation of threshold and symmetric functions," to appear in *SICOMP.*
- Azar, Y., R. Motwani, and J. Naor [1991]. "Approximating distributions using small sample spaces," unpublished manuscript, submitted to *Combinatorica.*
- Azar, Y., J. Naor, and R. Rom [1991]. "The competitiveness of on-line assignments," unpublished manuscript, submitted to 1992 *SODA.*
- Dolev, D., C. Dwork, O. Waarts, and M. Yung [1991]. "Perfectly secure message transmission," to appear in *JACM.* Also in 1990 *FOCS.*
- Dwork, C. and O. Waarts [1991]. "Simple and efficient bounded concurrent timestamp system," unpublished manuscript, IBM Almaden Research Center.
- Fischer, T., A. V. Goldberg, and S. A. Plotkin [1991]. "Approximating matchings in parallel," STAN-CS-91-1369, Dept. of CS, Stanford Univ, June, 1991.
- Goldberg, A. V. [1991]. "Processor-efficient implementation of a network flow algorithm," to appear in *Inf. Proc. Letters.*
- Goldberg, A. V., S. A. Plotkin, D. Shmoys, and E. Tardos [1991]. "Interior point methods in parallel computation," *SIAM J. Computing* 16:2, pp. 351-381.
- Goldberg, A. V., E. Tardos, and R. E. Tarjan [1990]. "Network flow algorithms," in *Flows, Paths, and VLSI Layout* (B. Korte, L. Lovasz, H. J. Promel, and A. Schrijver, eds.), Springer-Verlag, pp. 101-164.
- Herlihy, M., N. Shavit, and O. Waarts [1991]. "Low contention linearizable counting," to appear in 1991 *FOCS.*
- Hochbaum, D. and J. Naor [1991]. "Simple and fast algorithms for linear and integer programs with two variables per inequality," unpublished manuscript, submitted to *JACM.*

Klein, P., S. Khuller, and J. Naor [1991]. "The lattice structure of planar flow," unpublished manuscript, submitted to *SIAM J. Discrete Math.*

Klein, P., S. A. Plotkin, C. Stein, and E. Tardos [1991]. "Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts," TR-961 OR/IE, Cornell Univ.

Leighton, T., F. Makedon, S. A. Plotkin, C. Stein, S. Tragoudas, and E. Tardos [1991]. "Fast approximation algorithms for multicommodity flow problems," *Proc. Twenty-third Annual ACM Symposium on the Theory of Computing*, pp. 101-111.

Plotkin, S. A., D. Shmoys, and E. Tardos [1991]. "Fast approximation algorithms for fractional packing and covering problems," to appear in 1991 *FOCS*.

Ullman, J. D. and M. Yannakakis [1991]. "High-probability parallel transitive closure algorithms," *SIAM J. Computing* 20:1, pp. 100-125.

Principal Investigator Name: Jeffrey D. Ullman
PI Institution: Stanford University
PI Phone Number: (415) 723-9745
PI Email Address: ullman@cs.stanford.edu
Contract Title: Design and Implementation of Parallel Algorithms
Contract Number: N00014-88-K-0166
Reporting Period: 10/90-9/91

4. TRANSITIONS AND DOD INTERACTIONS

We note the survey By Goldberg, Tardos, and Tarjan [1990]. Also, articles by Gibbons and by Ullman, for the text on parallel algorithms edited by John Reif, which we reported last year, is expected to reach the bookstores by the end of the year.

Andy Freeman has discussed his architectural proposals with representatives of IDA.

Principal Investigator Name: Jeffrey D. Ullman
PI Institution: Stanford University
PI Phone Number: (415) 723-9745
PI Email Address: ullman@cs.stanford.edu
Contract Title: Design and Implementation of Parallel Algorithms
Contract Number: N00014-88-K-0166
Reporting Period: 10/90-9/91

5. SOFTWARE AND HARDWARE PROTOTYPES

None at a distributable stage. Andy Freeman has an extensive simulator for microprocessors, and Andy Goldberg has been working on code for parallel flow on a connection machine, in connection with a competition for algorithms of this type.