# DARPA/ISTO QUARTERLY REPORT

| | |
|---|---|
| **CONTRACT NUMBER:** | N00014-88-K-0458 |
| **CONTRACTOR:** | Duke University |
| **ARPA ORDER NO:** | 4331714-01/4-6-88 |
| **PROGRAM NAME:** | N00014 -88-K-0458 |
| **CONTRACT AMOUNT:** | 696,899.00 |
| **EFFECTIVE DATE OF CONTRACT:** | July 1, 1988 |
| **EXPIRATION DATE OF CONTRACT:** | June 30, 1991 |
| **PRINCIPAL INVESTIGATOR:** | John H. Reif |
| **TELEPHONE NUMBER:** | (919) 660-6568 |
| **SHORT TITLE OF WORK:** | Parallel Algorithms Derivation |
| **REPORTING PERIOD:** | January 1, 1989 - March 31, 1989 |

## DESCRIPTION OF PROGRESS--

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

1. Derivation of various parallel algorithms, parallel graph connectivity and parallel list ranking (with student, Doreen Yen),
2. Automatic Parallel Compilation from segmented straight line programs (with student, Lorrie Tomek),
3. Derivation of pipelined algorithms on small networks (with student, Steve Tate),
4. Programming Languages: Common Prototype Language (CPL), developed by student Lars Nyland and Professors Jan Prins, Robert Wagner and John Reif. CPL uses UNITY Primitives.
5. Duke Algorithm Derivation Seminar: participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; four visiting guest speakers in attendance were Professors Guy Blelloch, Gary Miller, Yijie Han and Victor Pan.
6. Planning a textbook on "Synthesis of Parallel Algorithms" (to be edited by Reif)--See participating authors/chapter titles list (enclosed).

# DARPA/ISTO QUARTERLY REPORT

| | |
|---|---|
| **CONTRACT NUMBER:** | N00014-88-K-0458 |
| **CONTRACTOR:** | Duke University |
| **ARPA ORDER NO:** | 4331714-01/4-6-88 |
| **PROGRAM NAME:** | N00014 |
| **CONTRACT AMOUNT:** | 696,899.00 |
| **EFFECTIVE DATE OF CONTRACT:** | July 1, 1988 |
| **EXPIRATION DATE OF CONTRACT:** | June 30, 1991 |
| **PRINCIPAL INVESTIGATOR:** | John H. Reif |
| **TELEPHONE NUMBER:** | (919) 660-6568 |
| **SHORT TITLE OF WORK:** | Parallel Algorithms Derivation |
| **REPORTING PERIOD:** | April 1, 1989 - June 30, 1989 |

## DESCRIPTION OF PROGRESS--

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

1. Derivation of various parallel algorithms, parallel graph connectivity and parallel list ranking (with student, Doreen Yen); parallel list ranking is now complete, now working on parallel graph connectivity;
2. Automatic Parallel Compilation from segmented straight line programs (with student, Lorrie Tomek),
3. Derivation of pipelined algorithms on small networks (with student, Steve Tate and Prof. Robert Wagner),
4. Programming Language: Common Prototype Language (CPL), developed by student Lars Nyland and Professors Jan Prins, Robert Wagner and John Reif. CPL uses UNITY Primitives; now collaborating with Kestrel Institute with DARPA Proposal in CPL;
5. Duke Algorithm Derivation Seminar: participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; four visiting guest speakers in attendance were Professors Guy Blelloch, Gary Miller, Yijie Han and Victor Pan;

6. Proceeding on a textbook on "Synthesis of Parallel Algorithms" (to be edited by Reif) participating authors are to visit Duke Universtiy in Fall 1989, many of whom are collaborating with Reif on their chapters.

DARPA/ONR QUARTERLY REPORT

CONTRACT NUMBER: N00014-88-K-0458

CONTRACTOR: Duke University

ARPA ORDER NO: 4331714-01/4-6-88

PROGRAM NAME: N00014

CONTRACT AMOUNT: 696,899.00

EFFECTIVE DATE OF CONTRACT: July 1, 1988

EXPIRATION DATE OF CONTRACT: June 30, 1991

PRINCIPAL INVESTIGATOR: John H. Reif

TELEPHONE NUMBER: (919) 660-6568

SHORT TITLE OF WORK: Parallel Algorithms Derivation

REPORTING PERIOD: July 1, 1989 - December 31, 1989

## DESCRIPTION OF PROGRESS--

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

1. Derivation of various parallel algorithms, parallel graph connectivity and parallel list
ranking (with student, Doreen Yen); parallel list ranking is now complete, now working on parallel graph connectivity;
2. Automatic Parallel Compilation from segmented straight line programs (with student,
Lorrie Tomek),
3. Derivation of pipelined algorithms on small networks (with student, Steve Tate and Prof. Robert Wagner),
4. Programming Language: Common Prototype Language (CPL), developed by student Lars
Nyland and Professors Jan Prins, Robert Wagner and John Reif. CPL uses UNITY Primitives;
now collaborating with Kestrel Institute with DARPA Proposal in CPL;

5. Duke Algorithm Derivation Seminar: participants--Professors Robert Wagner, Donald
Loveland, Gopalan Nadathur and John Reif; four visiting guest speakers in attendance were
Professors Guy Blelloch, Gary Miller, Yijie Han and Victor Pan;
6. Proceeding on a textbook on "Synthesis of Parallel Algorithms" (to be edited by Reif)
participating authors are to visit Duke Universtiy in Fall 1989, many of whom are collaborating with Reif on their chapters.

# DARPA/ONR QUARTERLY REPORT

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 January 1990 — 31 March 1990 |

## DESCRIPTION OF PROGRESS--

Special note: We're involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2); and Kestral Institute (see item # 7).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

1. Doreen Yen (graduate student) with John Reif:

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction", is now written.

2. Sri Krishnan (graduate student) with John Reif:
Krishnan is working on derivations of interior algorithms for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Kachinian Ellipsoid Algorithm. This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and certain features that distinguish it from integer programming. Then we examine the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. We have derived the general method in a methodical manner from very simple starting point.

3. Mike Landis (graduate student) with Robert Wagner and John Reif:

We concluded that Moldovan's method for mapping algorithms into VLSI arrays is not general enough to accommodate algorithms with complex data movement like many dynamic programming algorithms. Furthermore, we have discovered a number of fundamental flaws that are inherent in the method. Therefore, we decided that we should pursue other methods of solving this problem.

Our current work is now related to Guerra's work on synthesizing non-uniform systolic designs. Moreno and Lang also have a method that is similar to the one we are developing.

4. Sandeep Sen (graduate student then post doc) with John Reif.

We have extended the techniques used for obtaining efficient parallel algorithms for Computational Geometry in PRAM models to more practical computing models, namely the inter-connection network models. We have been able to derive optimal algorithms for 2-D convex hulls, triangulation, visibility all of which are very fundamental problems. In the process, we were also able to solve several basic problems like binary search and processor allocation which should be of use in deriving other combinatorial algorithms.

5. Subhrajit Bhattacharya with John Reif:

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

6. Hillel Gazit with John Reif

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time isT = O(log(n)) with P = O((m+n)/log(n))) processors, where $m$ is the number of edges and $n$ is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires O(m + n) space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes O(\log(n)) time with $P = O(n^{1.5}\sqrt{\log(n)})$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs 2 log(m) log(n) +O(log(n)) random bits. The number of random bits can be decreased to O(log(n)) by increasing the processors number to $n^{3/2+\varepsilon}$. This algorithm significantly improves the previous results of $n^4$ processors.

6. Reif has invited a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting.

At the present writing, some 21 researches have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and

presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

**Participating Authors and Topics**

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Ierardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah | Deterministic Parallel Computational Geometry |
| Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Sanguthevar Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Competeness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |

7. Programming Language: Common Prototype Language (CPL), developed by student Lars Nyland and Professors Jan Prins, Robert Wagner and John Reif.

DARPA/ISTO has challenged the community to design a prototyping language with ambitious capabilities. We proposed a Common Prototyping Language (CPL) that meets this challenge. This work was supported by this contract until 1 April 1990 (in collaboration with Kestral, another group supported by DARPA/ONR), so is reported here. It is separately funded as of 1 April 1990.

We are developing a language to be used for prototyping. The goal is to facilitate the initial prototyping of algorithms as executable programs. The Common Prototype Language (CPL) is our proposed answer for this goal. Our view of prototyping is the ability to write programs primarily for the purpose of experimentation and validation of ideas in a quick and easy fashion. Prototype programs do not necessarily have complete specifications.

We adopt UNITY, a programming paradigm for developing algorithms as the fundamental model of control in CPL. UNITY is supported by a programming logic, which allows

formal reasoning about the correctness, safety and progress of computations. The UNITY style supports formal program development from abstract specifications to concrete implementations. We provide CPL with a rich data model incorporating set-theoretic type constructors such as sets and relations, objects, user-defined data abstractions, and a rich, flexible notion of type. We augment the control structures of UNITY to support block structure, subroutines, sequencing, iteration, AI programming methods, and modules. These extensions will be consistent with the underlying semantics and will provide a surface syntax similar to conventional languages. We adopt REFINE's language extension capabilities, which support the definition of grammars, parsing and printing, a standard representation of abstract syntax within the data model, and a powerful pattern and transformation capability.

Any scheme for derivation of algorithms depends on an initial description of the algorithm to be solved; various methods can then be applied to transform such an initial description into a real program. However, the construction of initial descriptions, or specifications, seems itself to be non-trivial. The availability of a prototyping language, such as CPL, will vastly improve our ability to construct such initial statements of what an algorithm is supposed to be able to accomplish. In turn this should have significant impact on the forms of the transformations we need to consider within the context of derivation of algorithms. The availability of CPL should result in a common language which might well replace English as a means for describing describing algorithm intent. Future work in algorithm derivation may be able to assume a CPL version, as an initial description, and concentrate on derivation of algorithms targeted to specific architectures, or with certain performance goals, from that description.

Lars Nyland (graduate student) on CPL team:

Nyland has been working on the development of the Common Prototyping Language (CPL) with Drs. Reif, Wagner, Prins, Goldberg, and Jullig. We are in the process of developing a preliminary language design document. At Duke, we are concentrating on the control model, while the researchers at the Kestrel Institute are focusing their attention on the data model.

I visited the Kestrel Institute early in April in an effort to share the results of our efforts. We had a two-day meeting where each group presented their preliminary work. We had lengthy discussions, and set up plans to meet again at near the end of May in conjunction with the CPL meeting scheduled by DARPA.

8. Duke Algorithm Derivation Seminar: participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; four visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

9. Researchers supported (other than PI):

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student
Sandeep Sen, graduate student then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

10. Degrees awarded:

Sandeep Sen received his Ph.D. from Duke and is now here as a post-doc.

11. Papers

J. Reif and H. Gazit. A Parallel Planar Graph Isomorphism Algorithm. Accepted by Proc. of ACM Aegean Workshop On Computing, 1990.

J. Reif and V. Pan. On the Bit-Complexity of Discrete Solutions of PDEs: Compact Multigrid. Accepted by Proc. of ICALP '90. Also presented at Copper Mountain Conference on Interative Methods, Copper Mountain, Colorado, 1990.

H. Gazit. An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph (extended version). Accepted to SIAM J. of Computing, 1990.

H. Gazit. An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph. Submitted for Publication, 1990.

R. Wagner and M. Landis. Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement. Submitted for Publication, 1990.

J. Reif and S. Bhattacharya. Derivation of Efficient Parallel Sorting Algorithm. Draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and S. Sen. Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms. ACM 2nd Annual Symposium on Parallel Algorithms and Architectures, July 1990

J. Reif and D. Tygar. The Computational Complexity of Optical Beam Tracing. To appear in 1990.

J. Reif, J. Conny, and A. Page. An Exact Algorithm for Kinodynamic Planning in the Plane. Submitted for publication, 1990.

J. Reif and V. Pan. Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families. March, 1990.

# DARPA/ONR QUARTERLY REPORT

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 April 1990 — 30 June 1990 |

## DESCRIPTION OF PROGRESS--

Special note: We're involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

1. Doreen Yen (graduate student) with John Reif:

This paper extends the *Proofs as Programs paradigm* of Pfenning [Pfenning 88] to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential recursive functional algorithm is extracted from a constructive inductive proof and retains quantifiers and set notation. The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, ALGORITHM $C_u$, for computing the connected component which contains the vertex $u$.

An initial parallel connected components algorithm is derived from ALGORITHM $C_u$. The parallelism arises naturally by using a higher order function, *map*, which applies ALGORITHM $C_u$ to each element of the vertex set. This initial parallel algorithm is improved by algebraic transformation using a derived disjoint set union algorithm to obtain the parallel connected components algorithm CONNECT, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm CONNECT can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call *stream compaction*, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by $O(\log n)$.

A proof is given to show the $O(\log n)$ speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is $O(\log n)$.

2. Sri Krishnan (graduate student) with John Reif:

J. Reif and S. Bhattacharya. Derivation of Efficient Parallel Sorting Algorithm. Draft, 1990.

The work that Krishnan has done ( in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a small well chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming. In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

3. Mike Landis (graduate student) with Robert Wagner and John Reif:

A New Method for Mapping Algorithms into Parallel Architectures

Robert Wagner and Mike Landis

We are working on a new method for mapping algorithms into parallel architectures. An instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analagous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and build a new one that has a bounded in-degree and incorporates a limitation in the number of copies of available operands (bounded out-degree). To construct the new DAG, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build an operand cardinality matrix, R(i,j) = the number of operands that are a subset of C(i) that are actually used in the computation of C(j).

3) Build a time-availability matrix t(i,j) and vector T(j) where each t(i,j) is filled with the time that each computation point C(i) is available for use in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

We then use a method of histogram analysis to determine T(j) from the columns of t.

We also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the matrix t(i,j).

In order for a computation C(i) to complete at time T(j), there must be adequate processor availability. The number of processors required is equal to the cardinality of C(i), which is the total number of intermediate results in an instance of an algorithm that map to computation point C(i).

To demonstrate the usefulness of our method, we show how to derive various parallel implementations of algorithms to perform CFG recognition, dynammic programming, matrix multiplication, transitive closure, and vector outer product. We contrast our

method with that of direct functional derivation using Kosaraju's CFG recognition problem as an example.

4. Sandeep Sen (graduate student then post doc) with John Reif.

We have extended the techniques used for obtaining efficient parallel algorithms for Computational Geometry in PRAM models to more practical computing models, namely the inter-connection network models. We have been able to derive optimal algorithms for 2-D convex hulls, triangulation, visibility all of which are very fundamental problems. In the process, we were also able to solve several basic problems like binary search and processor allocation which should be of use in deriving other combinatorial algorithms.

5. Subhrajit Bhattacharya with John Reif:

6. Hillel Gazit with John Reif

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with $P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes $O(\log(n))$ time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs $2 \log(m) \log(n) + O(\log(n))$ random bits. The number of random bits can be decreased to $O(\log(n))$ by increasing the processors number to $n^{3/2+\varepsilon}$. This algorithm significantly improves the previous results of $n^4$ processors.

6. Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting.

At the present writing, some 21 researches have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

**Participating Authors and Topics**

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Margaret Reid-Miller | Parallel Tree Contraction and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Zlerardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah | Deterministic Parallel Computational Geometry |
| Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Sanguthevar Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Compeleteness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |

8. Duke Algorithm Derivation Seminar: participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; four visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankaj Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

9. Researchers supported (other than PI):

Subhrajit Bhattacharya, graduate student

Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student
Sandeep Sen, graduate student then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

10. Degrees awarded:

Sandeep Sen received his Ph.D. from Duke and is now here as a post-doc.

11. Papers

J. Reif and H. Gazit. A Parallel Planar Graph Isomorphism Algorithm. Accepted by Proc. of ACM Aegean Workshop On Computing, 1990.

J. Reif and V. Pan. On the Bit-Complexity of Discrete Solutions of PDEs: Compact Multigrid. Accepted by Proc. of ICALP '90. Also presented at Copper Mountain Conference on Interative Methods, Copper Mountain, Colorado, 1990.

H. Gazit. An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph (extended version). Accepted to SIAM J. of Computing, 1990.

H. Gazit. An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph. Submitted for Publication, 1990.

R. Wagner and M. Landis. Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement. Submitted for Publication, 1990.

J. Reif and S. Bhattacharya. Derivation of Efficient Parallel Sorting Algorithm. Draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and S. Sen. Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms. ACM 2nd Annual Symposium on Parallel Algorithms and Architectures, July 1990

J. Reif and D. Tygar. The Computational Complexity of Optical Beam Tracing. To appear in 1990.

J. Reif, J. Conny, and A. Page. An Exact Algorithm for Kinodynamic Planning in the Plane. Submitted for publication, 1990.

J. Reif and V. Pan. Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families. March, 1990.

DARPA/ONR MONTHLY REPORT

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 July 1990 — 30 September 1990 |

Surface mailed to:
Director, ARPA
1400 Wilson Street
Arlington, VA 22209

Ralph Wachter
Scientific Officer
Computer Science Division, Code 1133
Office of Naval Research
300 North Quincy Street
Arlington, VA 22217-5000

Marta L. Morris
The Ohio State University Research Center
1314 Kinnear Road
Columbus, OH 43212-1194

email: njacobs@vax.darpa.mil

DESCRIPTION OF PROGRESS--

Special note: We're involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

**1. Doreen Yen (graduate student) with John Reif:**

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction", is now written.

This paper extends the *Proofs as Programs paradigm* of Pfenning [Pfenning 88] to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential recursive functional algorithm is extracted from a constructive inductive proof and retains quantifiers and set notation. The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, ALGORITHM $C_u$, for computing the connected component which contains the vertex $u$.

An initial parallel connected components algorithm is derived from ALGORITHM $C_u$. The parallelism arises naturally by using a higher order function, *map*, which applies ALGORITHM $C_u$ to each element of the vertex set. This initial parallel algorithm is improved by algebraic transformation using a derived disjoint set union algorithm to obtain the parallel connected components algorithm CONNECT, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm CONNECT can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call *stream compaction*, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by $O(\log n)$.

A proof is given to show the $O(\log n)$ speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is $O(\log n)$.

**2. Sri Krishnan (graduate student) with John Reif:**

J. Reif and S. Bhattacharya. Derivation of Efficient Parallel Sorting Algorithm. Draft, 1990.

Krishnan is working on derivations of interior algorithms for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Kachinian Ellipsoid Algorithm. This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and certain features that distinguish it from integer programming. Then we examine the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. We have derived the general method in a methodical manner from very simple starting point.

The work that Krishnan has done ( in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a small well chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming. In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

**3. Mike Landis (graduate student) with Robert Wagner and John Reif:**

A New Method for Mapping Algorithms into Parallel Architectures

Robert Wagner and Mike Landis

We are working on a new method for mapping algorithms into parallel architectures. An instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analagous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and build a new one that has a bounded in-degree and incorporates a limitation in the number of copies of available operands (bounded out-degree). To construct the new DAG, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build an operand cardinality matrix, R(i,j) = the number of operands that are a subset of C(i) that are actually used in the computation of C(j).

3) Build a time-availability matrix t(i,j) and vector T(j) where each t(i,j) is filled with the time that each computation point C(i) is available for use in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

We then use a method of histogram analysis to determine T(j) from the columns of t.

We also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the matrix t(i,j).

In order for a computation C(i) to complete at time T(j), there must be adequate processor availability. The number of processors required is equal to the cardinality of C(i), which is the total number of intermediate results in an instance of an algorithm that map to computation point C(i).

To demonstrate the usefulness of our method, we show how to derive various parallel implementations of algorithms to perform CFG recognition, dynammic programming, matrix multiplication, transitive closure, and vector outer product. We contrast our method with that of direct functional derivation using Kosaraju's CFG recognition problem as an example.

We concluded that Moldovan's method for mapping algorithms into VLSI arrays is not general enough to accommodate algorithms with complex data movement like many dynamic programming algorithms. Furthermore, we have discovered a number of fundamental flaws that are inherent in the method. Therefore, we decided that we should pursue other methods of solving this problem.

Our current work is now related to Guerra's work on synthesizing non-uniform systolic designs. Moreno and Lang also have a method that is similar to the one we are developing.

**4. Subhrajit Bhattacharya with John Reif:**

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

**5. Hillel Gazit with John Reif**

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with $P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes O(log(n)) time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs 2 log(n) log(n) +O(log(n)) random bits. The number of random bits can be decreased to O(log(n)) by increasing the processors number to $n^{3/2+\varepsilon}$. This algorithm significantly improves the previous results of $n^4$ processors.

6. Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting.

At the present writing, some 21 researches have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

**Participating Authors and Topics**

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Margaret Reid-Miller | Parallel Tree Contraction and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |

| | |
|---|---|
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Ierardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah | Deterministic Parallel Computational Geometry |
| Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Sanguthevar Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Competeness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |

**7. Duke Algorithm Derivation Seminar:**

participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; four visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

**8. Researchers supported (other than PI):**

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

## 9. Papers

H. Gazit, "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph", (extended version), accepted to SIAM J. of Computing, 1990.

H. Gazit, "An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph", submitted for publication, 1990.

R. Wagner and M. Landis, "Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement", submitted for publication, 1990.

J. Reif and S. Bhattacharya, "Derivation of Efficient Parallel Sorting Algorithm", draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and V. Pan, "On the Bit Complexity of Discrete Approximations to PDEs", *International Colloquium on Automata, Languages, and Programming*, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990.

J. Reif and J. Storer, "A Parallel Architecture for High Speed Data Compression", *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, Maryland, October, 1990.

J. Reif and A. Yoshida, "Optical Expanders", manuscript, August 1989.

J. Reif, "Efficient Algorithms for Optical Computing with the DFT Primitive", *The 10th Conference on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

J. Reif, J. Canny, and A. Page, "An Exact Algorithm for Kinodynamic Planning in the Plane", *Symposium on Computational Geometry*, San Francisco, June, 1990.

J. Reif and S. Sen, "Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July,1990.

J. Reif, "Efficient Parallel Algorithms: Theory and Practice", *SIAM 35th Anniversary Meeting*, Denver, CO, Oct. 1987. Also *XI World Computer Congress*, IFIP 89, San Francisco, CA , 1989.

J. Reif and H. Djidjev, "An Efficient Algorithm for the Genus Problem", draft, April 1989.

J. Reif and H. Gazit, "A Parallel Planar Graph Isomorphism Algorithm", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July, 1990.

J. Reif and V. Pan, "Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families", submitted for publication, 1990.

J. Reif and S. Sen, "Randomized Parallel Algorithms", workshop on Capabilities and Limitations of Parallel Computing, IBM, San Jose, CA, Dec. 1988. Also in *Information Processing* 89, G. Ritter (ed) Elsevier Science Publishers, North Hollond, IFIP, 1989, pp. 455-458 , and as Randomization in Parallel Algorithms and its Impact on Computational Geometry, in Optimal Algorithms, H. Djidjev editor, Springer-Verlag Lecture Notes in Computer Science 401, 1989, 1-8. .

J. Reif, R. Paturi, and S. Rajasekaran, "The light bulb problem", presented at Workshop on Computational Learning Theory as Efficient and Robust Learning Using Statistical Bootstrap, Morgan Kaufmann Pub., Santa Cruz, CA, Aug. 1989. Submitted for journal publication.

J. Reif, D. Tygar, and A. Yoshida, "The Computation Complexity of Optical Beam Tracing", *31th IEEE Symposium on Foundations of Computer Science*,, Saint Louis, Missouri, October, 1990. .

J. Reif and A. Tyagi, "Energy Complexity of Optical-Computations", to appear in, *The 2nd IEEE Symposium on Parallel and Distributed Processing*. December, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Hector, "BLITZEN: a highly integrated, massively parallel machine", *2nd Symposium on Frontiers of Massively Parallel Computation*, Fairfax, VA, Oct. 1988. Also in *Journal of Parallel and Distributed Computing*, Feb. 1990.

J. Reif and V. Ramachandran, "An optimal parallel algorithm for planarity (with V. Ramachandran). 30th Annual Symposium on Foundations of Computer Science, Durham, NC, Oct, 1989, pp. 282-287. Also University of Texas at Austin Technical Report TR-90-15, June 1990. Also invited to special issue of Journal of Algorithms,1990.

J. Reif and V. Pan, "Fast and efficient parallel solution of dense linear systems", *Computers and Mathematics with Applications* vol. 17, no. 11, pp. 1481-1491, 1989.

J. Reif and P. Gacs, "A simple three-dimensional real-time reliable cellular array", *17th Annual Symposium on Theory of Computing*, 388-395, Providence, RI, 1985.Accepted for publication by *Journal of Computer and System Sciences*, vol. 36, no. 2, p. 125-147, April 1990.

DARPA/ONR Quarterly REPORT

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 Oct 1990 — 30 December 1990 |

DESCRIPTION OF PROGRESS--

Special note: We're involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

**1. Doreen Yen (graduate student) with John Reif:**

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction", is now written.

This paper extends the *Proofs as Programs paradigm* of Pfenning [Pfenning 88] to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential recursive functional algorithm is extracted from a constructive inductive proof and retains quantifiers and set notation. The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, ALGORITHM $C_u$, for computing the connected component which contains the vertex $u$.

An initial parallel connected components algorithm is derived from ALGORITHM $C_u$. The parallelism arises naturally by using a higher order function, *map*, which applies ALGORITHM $C_u$ to each element of the vertex set. This initial parallel algorithm is improved by algebraic transformation using a derived disjoint set union algorithm to obtain the parallel connected components algorithm CONNECT, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm CONNECT can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call *stream compaction*, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by $O(\log n)$.

A proof is given to show the $O(\log n)$ speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is $O(\log n)$.

**2. Sri Krishnan (graduate student) with John Reif:**

J. Reif and S. Bhattacharya. Derivation of Efficient Parallel Sorting Algorithm. Draft, 1990.

Krishnan is working on derivations of interior algorithms for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Kachinian Ellipsoid Algorithm. This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and certain features that distinguish it from integer programming. Then we examine the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. We have derived the general method in a methodical manner from very simple starting point.

The work that Krishnan has done ( in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a small well chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming. In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

**3. Mike Landis (graduate student) with Robert Wagner and John Reif:**

A New Method for Mapping Algorithms into Parallel Architectures

Robert Wagner and Mike Landis

We are working on a new method for mapping algorithms into parallel architectures. An instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analagous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and build a new one that has a bounded in-degree and incorporates a limitation in the number of copies of available operands (bounded out-degree). To construct the new DAG, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build an operand cardinality matrix, R(i,j) = the number of operands that are a subset of C(i) that are actually used in the computation of C(j).

3) Build a time-availability matrix t(i,j) and vector T(j) where each t(i,j) is filled with the time that each computation point C(i) is available for use in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

We then use a method of histogram analysis to determine T(j) from the columns of t.

We also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the matrix t(i,j).

In order for a computation C(i) to complete at time T(j), there must be adequate processor availability. The number of processors required is equal to the cardinality of C(i), which is the total number of intermediate results in an instance of an algorithm that map to computation point C(i).

To demonstrate the usefulness of our method, we show how to derive various parallel implementations of algorithms to perform CFG recognition, dynammic programming, matrix multiplication, transitive closure, and vector outer product. We contrast our method with that of direct functional derivation using Kosaraju's CFG recognition problem as an example.

We concluded that Moldovan's method for mapping algorithms into VLSI arrays is not general enough to accommodate algorithms with complex data movement like many dynamic programming algorithms. Furthermore, we have discovered a number of fundamental flaws that are inherent in the method. Therefore, we decided that we should pursue other methods of solving this problem.

Our current work is now related to Guerra's work on synthesizing non-uniform systolic designs. Moreno and Lang also have a method that is similar to the one we are developing.

**4. Sandeep Sen (graduate student then post doc) with John Reif.**

We have extended the techniques used for obtaining efficient parallel algorithms for Computational Geometry in PRAM models to more practical computing models, namely the inter-connection network models. We have been able to derive optimal algorithms for 2-D convex hulls, triangulation, visibility all of which are very fundamental problems. In the process, we were also able to solve several basic problems like binary search and processor allocation which should be of use in deriving other combinatorial algorithms.

**5. Subhrajit Bhattacharya with John Reif:**

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

**6. Hillel Gazit with John Reif**

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with $P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes $O(\log(n))$ time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs $2 \log(m) \log(n) + O(\log(n))$ random bits. The number of random bits can be decreased to $O(\log(n))$ by increasing the processors number to $n^{3/2+\varepsilon}$. This algorithm significantly improves the previous results of $n^4$ processors.

7. Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting.

At the present writing, some 21 researches have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

**Participating Authors and Topics**

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Margaret Reid-Miller | Parallel Tree Contraction and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with |

| | |
|---|---|
| | Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Ierardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah | Deterministic Parallel Computational Geometry |
| Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Sanguthevar Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Competeness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |

**8. Programming Language: Common Prototype Language (CPL)**

This work was initially funded entirely by this DARPA/ISTO contract but later was augmented by an additional contract. It is an important example of how fundamental work (in derivation) can also lead to more applied work (in this case in programming support for derivation and prototyping).

DARPA/ISTO has challenged the community to design a prototyping language with ambitious capabilities. We proposed a Common Prototyping Language (CPL) that meets this challenge. This work was supported by this contract until 1 April 1990 (in collaboration with Kestral, another group supported by DARPA/ONR), so is reported here. It is separately funded as of 1 April 1990.

We are developing a language to be used for prototyping. The goal is to facilitate the initial prototyping of algorithms as executable programs. The Common Prototype Language (CPL) is our proposed answer for this goal. Our view of prototyping is the ability to write programs primarily for the purpose of experimentation and validation of ideas in a quick and easy fashion. Prototype programs do not necessarily have complete specifications.

We adopt UNITY, a programming paradigm for developing algorithms as the fundamental model of control in CPL. UNITY is supported by a programming logic, which allows formal reasoning about the correctness, safety and progress of computations. The UNITY style supports formal program development from abstract specifications to concrete implementations. We provide CPL with a rich data model incorporating set-theoretic type constructors such as sets and relations, objects, user-defined data abstractions, and a rich, flexible notion of type. We augment the control structures of UNITY to support block structure, subroutines, sequencing, iteration, AI programming methods, and modules. These extensions will be consistent with the underlying semantics and will provide a surface syntax similar to conventional languages. We adopt REFINE's language

extension capabilities, which support the definition of grammars, parsing and printing, a standard representation of abstract syntax within the data model, and a powerful pattern and transformation capability.

Any scheme for derivation of algorithms depends on an initial description of the algorithm to be solved; various methods can then be applied to transform such an initial description into a real program. However, the construction of initial descriptions, or specifications, seems itself to be non-trivial. The availability of a prototyping language, such as CPL, will vastly improve our ability to construct such initial statements of what an algorithm is supposed to be able to accomplish. In turn this should have significant impact on the forms of the transformations we need to consider within the context of derivation of algorithms. The availability of CPL should result in a common language which might well replace English as a means for describing describing algorithm intent. Future work in algorithm derivation may be able to assume a CPL version, as an initial description, and concentrate on derivation of algorithms targeted to specific architectures, or with certain performance goals, from that description.

Lars Nyland (graduate student) on CPL team:

Nyland has been working on the development of the Common Prototyping Language (CPL) with Drs. Reif, Wagner, Prins, Goldberg, and Jullig. We are in the process of developing a preliminary language design document. At Duke, we are concentrating on the control model, while the researchers at the Kestrel Institute are focusing their attention on the data model.

I visited the Kestrel Institute early in April in an effort to share the results of our efforts. We had a two-day meeting where each group presented their preliminary work. We had lengthy discussions, and set up plans to meet again at near the end of May in conjunction with the CPL meeting scheduled by DARPA.

**9. Duke Algorithm Derivation Seminar:**

participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; four visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

**10. Researchers supported (other than PI):**

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student
Sandeep Sen, graduate student then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

**11. Degrees awarded:**

Sandeep Sen received his Ph.D. from Duke and was here as a post-doc.

## 12. Papers

H. Gazit, "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph", (extended version), accepted to SIAM J. of Computing, 1990.

H. Gazit, "An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph", submitted for publication, 1990.

R. Wagner and M. Landis, "Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement", submitted for publication, 1990.

J. Reif and S. Bhattacharya, "Derivation of Efficient Parallel Sorting Algorithm", draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and V. Pan, "On the Bit Complexity of Discrete Approximations to PDEs", *International Colloquium on Automata, Languages, and Programming*, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990.

J. Reif and J. Storer, "A Parallel Architecture for High Speed Data Compression", *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, Maryland, October, 1990.

J. Reif and A. Yoshida, "Optical Expanders", manuscript, August 1989.

J. Reif, "Efficient Algorithms for Optical Computing with the DFT Primitive", *The 10th Conference on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

J. Reif, J. Canny, and A. Page, "An Exact Algorithm for Kinodynamic Planning in the Plane", *Symposium on Computational Geometry*, San Francisco, June, 1990.

J. Reif and S. Sen, "Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July,1990.

J. Reif, "Efficient Parallel Algorithms: Theory and Practice", *SIAM 35th Anniversary Meeting*, Denver, CO, Oct. 1997. Also *XI World Computer Congress*, IFIP 89, San Francisco, CA , 1989.

J. Reif and H. Djidjev, "An Efficient Algorithm for the Genus Problem", draft, April 1989.

J. Reif and H. Gazit, "A Parallel Planar Graph Isomorphism Algorithm", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July, 1990.

J. Reif and V. Pan, "Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families", submitted for publication, 1990.

J. Reif and S. Sen, "Randomized Parallel Algorithms", workshop on Capabilities and Limitations of Parallel Computing, IBM, San Jose, CA, Dec. 1988. Also in *Information Processing* 89, G. Ritter (ed) Elsevier Science Publishers, North Hollond, IFIP, 1989, pp. 455-458 , and as Randomization in Parallel Algorithms and its Impact on Computational Geometry, in Optimal Algorithms, H. Djidjev editor, Springer-Verlag Lecture Notes in Computer Science 401, 1989, 1-8. .

J. Reif, R. Paturi, and S. Rajasekaran, "The light bulb problem", presented at Workshop on Computational Learning Theory as Efficient and Robust Learning Using Statistical Bootstrap, Morgan Kaufmann Pub., Santa Cruz, CA, Aug. 1989. Submitted for journal publication.

J. Reif, D. Tygar, and A. Yoshida, "The Computation Complexity of Optical Beam Tracing", *31th IEEE Symposium on Foundations of Computer Science*,, Saint Louis, Missouri, October, 1990. .

J. Reif and A. Tyagi, "Energy Complexity of Optical-Computations", to appear in, *The 2nd IEEE Symposium on Parallel and Distributed Processing*. December, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Hector, "BLITZEN: a highly integrated, massively parallel machine", *2nd Symposium on Frontiers of Massively Parallel Computation*, Fairfax, VA, Oct. 1988. Also in *Journal of Parallel and Distributed Computing*, Feb. 1990.

J. Reif and V. Ramachandran, "An optimal parallel algorithm for planarity (with V. Ramachandran). 30th Annual Symposium on Foundations of Computer Science, Durham, NC, Oct, 1989, pp. 282-287. Also University of Texas at Austin Technical Report TR-90-15, June 1990. Also invited to special issue of Journal of Algorithms,1990.

J. Reif and V. Pan, "Fast and efficient parallel solution of dense linear systems", *Computers and Mathematics with Applications* vol. 17, no. 11, pp. 1481-1491, 1989.

J. Reif and P. Gacs, "A simple three-dimensional real-time reliable cellular array", *17th Annual Symposium on Theory of Computing*, 388-395, Providence, RI, 1985.Accepted for publication by *Journal of Computer and System Sciences*, vol. 36, no. 2, p. 125-147, April 1990.

# DARPA/ONR Quarterly Report

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 Jan 1991 — 31 Mar 1991 |

## DESCRIPTION OF PROGRESS--

Special note: We are involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

### 1. Mike Landis (A.B.D.) with Robert Wagner and John Reif:

Summary:

We have completed our work in the investigation of how to map context-free grammar recognition onto systolic arrays. We are currently in the final phases of the preparation of a technical report which will document this work. Our current research efforts are to extend our method to other algorithms. We are just beginning this investigation. We plan to submit our work for publication in a journal after doing this extension, as described below.

Details:
We have developed a new method for mapping algorithms into parallel architectures. This new method works very well for a class of dynamic programming problems, including CFG recognition.

In our work, we define an instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analogous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture. To ease this task, we have proven that a restricted form of recurrence equation, called a quasi-uniform recurrence equation (QURE), is computationally equivalent to a class of systolic array architectures. The class of QUREs is a broad class, the main restriction being that the recurrence must exhibit a finite set of dependency vectors over all computations. Expressing an algorithm as a recurrence equation, the problem of mapping the algorithm onto a systolic array becomes the problem of translating the recurrence equation into QURE form. It is also very easy to translate between the DAG representation and the recurrence relation.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms, especially dynamic programming algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and label its arcs and nodes to meet two constraints: The number of copies of each computation that can exist in each timestep, and the number of data that each computation can receive in each timestep. To implement these constraints, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build a time-availability table t (i,j) and vector T (j) where each t(i,j) is filled with the time that each computation point C(i) is available for use in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

In building this table, we also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the table t(i,j).

3) We then augment this table with information about the source through which each datum may be passed.

4) We then invert this table to determine what data is passed on which computation points at each time.

Current investigation on the extension of our method mostly involves step 1. Deriving the computational indexing scheme is analagous to solving a specific graph layout problem. In order to achieve a QURE as the result of our method, the nodes in the DAG must be labeled in a multi-dimensional coordinate system such that the computation of each node only depends upon nodes that are a fixed distance away. We are currently investigating a shortest-path method to perform this indexing, which promises to work for a broad class of algorithms -- not just dynamic programming.

**2. Sri Krishnan (graduate student) with John Reif:**

M. Karr, S. Krishnan and J. Reif "Derivation of the Ellipsoid Algorithm" Draft, 1990.

Krishnan is working on derivations of the ellipsoid algorithm for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Khachian Ellipsoid Algorithm.This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. The work that Krishnan has done ( in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a small well-chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming.In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

**3. Doreen Yen (graduate student) with John Reif:**

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction," is now written.

This paper uses a denotational definition derivation to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential recursive functional algorithm retains quantifiers and set notation. It is extracted from a denotational definition of a monotonic continuous functional whose least fixed point is a function consistent with the problem specification.

The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, Algorithm Cu, for computing the connected component which contains the vertex u.

An initial parallel connected components algorithm is derived from Algorithm Cu. The parallelism arises naturally by using a higher order function, map, which applies Algorithm Cu to each element of the vertex set. A more efficient parallel algorithm, fcc, is also obtained by denotational definition derivation. Its transition to an iterative algorithm is by discovery of lemmas showing the number of iterations needed to reach the fixed point and choice of efficient data structures to implement disjoint set union. An arbitrary constraint upon the data structure is imposed to obtain the parallel connected components algorithm connect, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm connect can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call stream compaction, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by O(log n).

A proof is given to show the O(log n) speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is O(log n).

**4. Subhrajit Bhattacharya with John Reif:**

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

**5. Hillel Gazit with John Reif**

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with $P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes O(log(n)) time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs $2 \log(m) \log(n) + O(\log(n))$ random bits. The number of random bits can be decreased to O(log(n)) by increasing the processors number to $n^{3/2+e}$. This algorithm significantly improves the previous results of $n^4$ processors.

## 7. Various Authors--*Synthesis of Parallel Algorithms* (edited by Reif)

Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting. *Synthesis of Parallel Algorithms* will be published by Morgan Kaufmann in Summer 1991.

At the present writing, some 27 researchers have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

### Participating Authors and Topics

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Margaret Reid-Miller | Parallel Tree Contraction and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |

**Participating Authors and Topics (continued)**

| | |
|---|---|
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Ierardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah and Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Sanguthevar Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Competeness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |
| Faith Fich | The Parallel Random Access Machine |

**8. Duke Algorithm Derivation Seminar:**

Participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

**9. Researchers supported (other than PI):**

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student, then post-doc
Sandeep Sen, graduate student, then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

**10. Degrees awarded:**

Sandeep Sen received his Ph.D. from Duke and was here as a post-doc. Steve Tate and Lars Nyland received their Ph.D.s in January 1991 under Reif and are both remaining at Duke as post-docs.

## 11. Papers

H. Gazit, "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph," (extended version), accepted to SIAM J. of Computing, 1990.

H. Gazit, "An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph," submitted for publication, 1990.

H. Gazit, "Finding the Diameter of a Directed Graph," submitted for publication, 1990.

H. Gazit, "Parallel Algorithms for Connectivity, Ear Decomposition and st-Numbering of Planar Graph," accepted to the Fifth International Parallel Processing Symposium, 1990.

Nyland, L., "The Design of a Prototyping Programming Language for Parallel and Sequential Algorithms," Ph.D. Thesis, Duke University, 1990.

Tate, Steve, "Arithmetic Circuit Complexity and Motion Planning," Ph.D. Thesis, Duke University, 1990.

R. Wagner and M. Landis, "Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement", submitted for publication, 1990.

J. Reif and S. Bhattacharya, "Derivation of Efficient Parallel Sorting Algorithm", draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and V. Pan, "On the Bit Complexity of Discrete Approximations to PDEs", *International Colloquium on Automata, Languages, and Programming*, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990.

J. Reif and J. Storer, "A Parallel Architecture for High Speed Data Compression", *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, Maryland, October, 1990.

J. Reif and A. Yoshida, "Optical Expanders", manuscript, August 1989.

J. Reif, "Efficient Algorithms for Optical Computing with the DFT Primitive", *The 10th Conference on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

J. Reif, J. Canny, and A. Page, "An Exact Algorithm for Kinodynamic Planning in the Plane", *Symposium on Computational Geometry*, San Francisco, June, 1990.

J. Reif and S. Sen, "Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July,1990.

J. Reif, "Efficient Parallel Algorithms: Theory and Practice", *SIAM 35th Anniversary Meeting*. Denver, CO, Oct. 1987. Also *XI World Computer Congress*, IFIP 89, San Francisco, CA , 1989.

J. Reif and H. Djidjev, "An Efficient Algorithm for the Genus Problem", draft, April 1989.

J. Reif and H. Gazit, "A Parallel Planar Graph Isomorphism Algorithm", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July, 1990.

J. Reif and V. Pan, "Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families", submitted for publication, 1990.

J. Reif and S. Sen, "Randomized Parallel Algorithms", workshop on Capabilities and Limitations of Parallel Computing, IBM, San Jose, CA, Dec. 1988. Also in *Information Processing* 89, G. Ritter (ed) Elsevier Science Publishers, North Hollond, IFIP, 1989, pp. 455-458 , and as Randomization in Parallel Algorithms and its Impact on Computational Geometry, in Optimal Algorithms, H. Djidjev editor, Springer-Verlag Lecture Notes in Computer Science 401, 1989, 1-8.

J. Reif, R. Paturi, and S. Rajasekaran, "The light bulb problem", presented at Workshop on Computational Learning Theory as Efficient and Robust Learning Using Statistical Bootstrap, Morgan Kaufmann Pub., Santa Cruz, CA, Aug. 1989. Submitted for journal publication.

J. Reif, D. Tygar, and A. Yoshida, "The Computation Complexity of Optical Beam Tracing", *31th IEEE Symposium on Foundations of Computer Science*,, Saint Louis, Missouri, October, 1990.

J. Reif and A. Tyagi, "Energy Complexity of Optical-Computations", to appear in, *The 2nd IEEE Symposium on Parallel and Distributed Processing*. December, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Hector, "BLITZEN: a highly integrated, massively parallel machine", *2nd Symposium on Frontiers of Massively Parallel Computation*, Fairfax, VA, Oct. 1988. Also in *Journal of Parallel and Distributed Computing*, Feb. 1990.

J. Reif and V. Ramachandran, "An optimal parallel algorithm for planarity (with V. Ramachandran). 30th Annual Symposium on Foundations of Computer Science, Durham, NC, Oct, 1989, pp. 282-287. Also University of Texas at Austin Technical Report TR-90-15, June 1990. Also invited to special issue of Journal of Algorithms,1990.

J. Reif and V. Pan, "Fast and efficient parallel solution of dense linear systems", *Computers and Mathematics with Applications* vol. 17, no. 11, pp. 1481-1491, 1989.

J. Reif and P. Gacs, "A simple three-dimensional real-time reliable cellular array", *17th Annual Symposium on Theory of Computing*, 388-395, Providence, RI, 1985.Accepted for publication by *Journal of Computer and System Sciences*, vol. 36, no. 2, p. 125-147, April 1990.

# DARPA/ONR Quarterly Report

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 April 1991 — 30 June 1991 |

## DESCRIPTION OF PROGRESS--

Special note: We are involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

### 1. Mike Landis (A.B.D.) with Robert Wagner and John Reif:

Summary:

We have completed our work in the investigation of how to map context-free grammar recognition onto systolic arrays. We are currently in the final phases of the preparation of a technical report which will document this work. Our current research efforts are to extend our method to other algorithms. We are just beginning this investigation. We plan to submit our work for publication in a journal after doing this extension, as described below.

Details:
We have developed a new method for mapping algorithms into parallel architectures. This new method works very well for a class of dynamic programming problems, including CFG recognition.

In our work, we define an instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analogous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture. To ease this task, we have proven that a restricted form of recurrence equation, called a quasi-uniform recurrence equation (QURE), is computationally equivalent to a class of systolic array architectures. The class of QUREs is a broad class, the main restriction being that the recurrence must exhibit a finite set of dependency vectors over all computations. Expressing an algorithm as a recurrence equation, the problem of mapping the algorithm onto a systolic array becomes the problem of translating the recurrence equation into QURE form. It is also very easy to translate between the DAG representation and the recurrence relation.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms, especially dynamic programming algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and label its arcs and nodes to meet two constraints: The number of copies of each computation that can exist in each timestep, and the number of data that each computation can receive in each timestep. To implement these constraints, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build a time-availability table t (i,j) and vector T (j) where each t(i,j) is filled with the time that each computation point C(i) is available for use in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

In building this table, we also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the table t(i,j).

3) We then augment this table with information about the source through which each datum may be passed.

4) We then invert this table to determine what data is passed on which computation points at each time.

Current investigation on the extension of our method mostly involves step 1. Deriving the computational indexing scheme is analagous to solving a specific graph layout problem. In order to achieve a QURE as the result of our method, the nodes in the DAG must be labeled in a multi-dimensional coordinate system such that the computation of each node only depends upon nodes that are a fixed distance away. We are currently investigating a shortest-path method to perform this indexing, which promises to work for a broad class of algorithms -- not just dynamic programming.

**2. Sri Krishnan (graduate student) with John Reif:**

M. Karr, S. Krishnan and J. Reif "Derivation of the Ellipsoid Algorithm" Draft, 1990.

Krishnan is working on derivations of the ellipsoid algorithm for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Khachian Ellipsoid Algorithm.This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. The work that Krishnan has done ( in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a small well-chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming.In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

**3. Doreen Yen (graduate student) with John Reif:**

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction," is now written.

This paper uses a denotational definition derivation to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential recursive functional algorithm retains quantifiers and set notation. It is extracted from a denotational definition of a monotonic continuous functional whose least fixed point is a function consistent with the problem specification.

The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, Algorithm Cu, for computing the connected component which contains the vertex u.

An initial parallel connected components algorithm is derived from Algorithm Cu. The parallelism arises naturally by using a higher order function, map, which applies Algorithm Cu to each element of the vertex set. A more efficient parallel algorithm, fcc, is also obtained by denotational definition derivation. Its transition to an iterative algorithm is by discovery of lemmas showing the number of iterations needed to reach the fixed point and choice of efficient data structures to implement disjoint set union. An arbitrary constraint upon the data structure is imposed to obtain the parallel connected components algorithm connect, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm connect can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call stream compaction, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by O(log n).

A proof is given to show the O(log n) speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is O(log n).

### 4. Subhrajit Bhattacharya with John Reif:

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

### 5. Hillel Gazit with John Reif

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with $P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes O(log(n)) time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs $2 \log(m) \log(n) + O(\log(n))$ random bits. The number of random bits can be decreased to O(log(n)) by increasing the processors number to $n^{3/2+e}$. This algorithm significantly improves the previous results of $n^4$ processors.

## 7. Various Authors--*Synthesis of Parallel Algorithms* (edited by Reif)

Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting. *Synthesis of Parallel Algorithms* will be published by Morgan Kaufmann in Summer 1991.

At the present writing, some 27 researchers have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

### Participating Authors and Topics

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Margaret Reid-Miller | Parallel Tree Contraction and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |

**Participating Authors and Topics (continued)**

| | |
|---|---|
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Ierardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah and Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Sanguthevar Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Competeness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |
| Faith Fich | The Parallel Random Access Machine |

**8. Duke Algorithm Derivation Seminar:**

Participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

**9. Researchers supported (other than PI):**

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student, then post-doc
Sandeep Sen, graduate student, then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

**10. Degrees awarded:**

Sandeep Sen received his Ph.D. from Duke and was here as a post-doc. Steve Tate and Lars Nyland received their Ph.D.s in January 1991 under Reif and are both remaining at Duke as post-docs.

## 11. Papers

H. Gazit, "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph," (extended version), accepted to SIAM J. of Computing, 1990.

H. Gazit, "An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph," submitted for publication, 1990.

H. Gazit, "Finding the Diameter of a Directed Graph," submitted for publication, 1990.

H. Gazit, "Parallel Algorithms for Connectivity, Ear Decomposition and st-Numbering of Planar Graph," accepted to the Fifth International Parallel Processing Symposium, 1990.

Nyland, L., "The Design of a Prototyping Programming Language for Parallel and Sequential Algorithms," Ph.D. Thesis, Duke University, 1990.

Tate, Steve, "Arithmetic Circuit Complexity and Motion Planning," Ph.D. Thesis, Duke University, 1990.

R. Wagner and M. Landis, "Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement", submitted for publication, 1990.

J. Reif and S. Bhattacharya, "Derivation of Efficient Parallel Sorting Algorithm", draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and V. Pan, "On the Bit Complexity of Discrete Approximations to PDEs", *International Colloquium on Automata, Languages, and Programming*, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990.

J. Reif and J. Storer, "A Parallel Architecture for High Speed Data Compression", *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, Maryland, October, 1990.

J. Reif and A. Yoshida, "Optical Expanders", manuscript, August 1989.

J. Reif, "Efficient Algorithms for Optical Computing with the DFT Primitive", *The 10th Conference on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

J. Reif, J. Canny, and A. Page, "An Exact Algorithm for Kinodynamic Planning in the Plane", *Symposium on Computational Geometry*, San Francisco, June, 1990.

J. Reif and S. Sen, "Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July,1990.

J. Reif, "Efficient Parallel Algorithms: Theory and Practice", *SIAM 35th Anniversary Meeting*, Denver, CO, Oct. 1987. Also *XI World Computer Congress*, IFIP 89, San Francisco, CA , 1989.

J. Reif and H. Djidjev, "An Efficient Algorithm for the Genus Problem", draft, April 1989.

J. Reif and H. Gazit, "A Parallel Planar Graph Isomorphism Algorithm", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July, 1990.

J. Reif and V. Pan, "Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families", submitted for publication, 1990.

J. Reif and S. Sen, "Randomized Parallel Algorithms", workshop on Capabilities and Limitations of Parallel Computing, IBM, San Jose, CA, Dec. 1988. Also in *Information Processing* 89, G. Ritter (ed) Elsevier Science Publishers, North Hollond, IFIP, 1989, pp. 455-458 , and as Randomization in Parallel Algorithms and its Impact on Computational Geometry, in Optimal Algorithms, H. Djidjev editor, Springer-Verlag Lecture Notes in Computer Science 401, 1989, 1-8.

J. Reif, R. Paturi, and S. Rajasekaran, "The light bulb problem", presented at Workshop on Computational Learning Theory as Efficient and Robust Learning Using Statistical Bootstrap, Morgan Kaufmann Pub., Santa Cruz, CA, Aug. 1989. Submitted for journal publication.

J. Reif, D. Tygar, and A. Yoshida, "The Computation Complexity of Optical Beam Tracing", *31th IEEE Symposium on Foundations of Computer Science*,, Saint Louis, Missouri, October, 1990.

J. Reif and A. Tyagi, "Energy Complexity of Optical-Computations", to appear in, *The 2nd IEEE Symposium on Parallel and Distributed Processing*. December, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Hecter, "BLITZEN: a highly integrated, massively parallel machine", *2nd Symposium on Frontiers of Massively Parallel Computation*, Fairfax, VA, Oct. 1988. Also in *Journal of Parallel and Distributed Computing*, Feb. 1990.

J. Reif and V. Ramachandran, "An optimal parallel algorithm for planarity (with V. Ramachandran). 30th Annual Symposium on Foundations of Computer Science, Durham, NC, Oct, 1989, pp. 282-287. Also University of Texas at Austin Technical Report TR-90-15, June 1990. Also invited to special issue of Journal of Algorithms,1990.

J. Reif and V. Pan, "Fast and efficient parallel solution of dense linear systems", *Computers and Mathematics with Applications* vol. 17, no. 11, pp. 1481-1491, 1989.

J. Reif and P. Gacs, "A simple three-dimensional real-time reliable cellular array", *17th Annual Symposium on Theory of Computing*, 388-395, Providence, RI, 1985.Accepted for publication by *Journal of Computer and System Sciences*, vol. 36, no. 2, p. 125-147, April 1990.

# DARPA/ONR Quarterly Report

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 July 1991 — 30 Sept 1991 |

## DESCRIPTION OF PROGRESS--

Special note: We are involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

### 1. Mike Landis (A.B.D.) with Robert Wagner and John Reif:

Summary:

We have completed our work in the investigation of how to map context-free grammar recognition onto systolic arrays. We are currently in the final phases of the preparation of a technical report which will document this work. Our current research efforts are to extend our method to other algorithms. We are just beginning this investigation. We plan to submit our work for publication in a journal after doing this extension, as described below.

Details:
We have developed a new method for mapping algorithms into parallel architectures. This new method works very well for a class of dynamic programming problems, including CFG recognition.

In our work, we define an instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analogous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture. To ease this task, we have proven that a restricted form of recurrence equation, called a quasi-uniform recurrence equation (QURE), is computationally equivalent to a class of systolic array architectures. The class of QUREs is a broad class, the main restriction being that the recurrence must exhibit a finite set of dependency vectors over all computations. Expressing an algorithm as a recurrence equation, the problem of mapping the algorithm onto a systolic array becomes the problem of translating the recurrence equation into QURE form. It is also very easy to translate between the DAG representation and the recurrence relation.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms, especially dynamic programming algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and label its arcs and nodes to meet two constraints: The number of copies of each computation that can exist in each timestep, and the number of data that each computation can receive in each timestep. To implement these constraints, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build a time-availability table t (i,j) and vector T (j) where each t(i,j) is filled with the time that each computation point C(i) is available for use

in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

In building this table, we also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the table t(i,j).

3) We then augment this table with information about the source through which each datum may be passed.

4) We then invert this table to determine what data is passed on which computation points at each time.

Current investigation on the extension of our method mostly involves step 1. Deriving the computational indexing scheme is analagous to solving a specific graph layout problem. In order to achieve a QURE as the result of our method, the nodes in the DAG must be labeled in a multi-dimensional coordinate system such that the computation of each node only depends upon nodes that are a fixed distance away. We are currently investigating a shortest-path method to perform this indexing, which promises to work for a broad class of algorithms -- not just dynamic programming.

**2. Sri Krishnan (graduate student) with John Reif:**

M. Karr, S. Krishnan and J. Reif "Derivation of the Ellipsoid Algorithm" Draft, 1990.

Krishnan is working on derivations of the ellipsoid algorithm for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Khachian Ellipsoid Algorithm.This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. The work that Krishnan has done ( in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a small well-chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming.In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

**3. Doreen Yen (graduate student) with John Reif:**

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction," is now written.

This paper uses a denotational definition deri- ation to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential rec. ive

functional algorithm retains quantifiers and set notation. It is extracted from a denotational definition of a monotonic continuous functional whose least fixed point is a function consistent with the problem specification.

The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, Algorithm Cu, for computing the connected component which contains the vertex u.

An initial parallel connected components algorithm is derived from Algorithm Cu. The parallelism arises naturally by using a higher order function, map, which applies Algorithm Cu to each element of the vertex set. A more efficient parallel algorithm, fcc, is also obtained by denotational definition derivation. Its transition to an iterative algorithm is by discovery of lemmas showing the number of iterations needed to reach the fixed point and choice of efficient data structures to implement disjoint set union. An arbitrary constraint upon the data structure is imposed to obtain the parallel connected components algorithm connect, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm connect can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call stream compaction, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by O(log n).

A proof is given to show the O(log n) speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is O(log n).

### 4. Subhrajit Bhattacharya with John Reif:

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

### 5. Hillel Gazit with John Reif

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with
$P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes O(log(n))

time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs
$2 \log(m) \log(n) + O(\log(n))$ random bits. The number of random bits can be decreased to $O(\log(n))$ by increasing the processors number to $n^{3/2+e}$. This algorithm significantly improves the previous results of $n^4$ processors.

### 7. Various Authors--*Synthesis of Parallel Algorithms* (edited by Reif)

Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting. *Synthesis of Parallel Algorithms* will be published by Morgan Kaufmann in Summer 1991.

At the present writing, some 27 researchers have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

#### Participating Authors and Topics

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Margaret Reid-Miller | Parallel Tree Contraction and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |

**Participating Authors and Topics (continued)**

| | |
|---|---|
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Ierardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah and Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design Sanguthevar |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Competeness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |
| Faith Fich | The Parallel Random Access Machine |

**8. Duke Algorithm Derivation Seminar:**

Participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

**9. Researchers supported (other than PI):**

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student, then post-doc
Sandeep Sen, graduate student, then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

**10. Degrees awarded:**

Sandeep Sen received his Ph.D. from Duke and was here as a post-doc. Steve Tate and Lars Nyland received their Ph.D.s in January 1991 under Reif and are both remaining at Duke as post-docs.

## 11. Papers

H. Gazit, "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph," (extended version), accepted to SIAM J. of Computing, 1990.

H. Gazit, "An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph," submitted for publication, 1990.

H. Gazit, "Finding the Diameter of a Directed Graph," submitted for publication, 1990.

H. Gazit, "Parallel Algorithms for Connectivity, Ear Decomposition and st-Numbering of Planar Graph," accepted to the Fifth International Parallel Processing Symposium, 1990.

Nyland, L., "The Design of a Prototyping Programming Language for Parallel and Sequential Algorithms," Ph.D. Thesis, Duke University, 1990.

Tate, Steve, "Arithmetic Circuit Complexity and Motion Planning," Ph.D. Thesis, Duke University, 1990.

R. Wagner and M. Landis, "Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement", submitted for publication, 1990.

J. Reif and S. Bhattacharya, "Derivation of Efficient Parallel Sorting Algorithm", draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and V. Pan, "On the Bit Complexity of Discrete Approximations to PDEs", *International Colloquium on Automata, Languages, and Programming*, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990.

J. Reif and J. Storer, "A Parallel Architecture for High Speed Data Compression", *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, Maryland, October, 1990.

J. Reif and A. Yoshida, "Optical Expanders", manuscript, August 1989.

J. Reif, "Efficient Algorithms for Optical Computing with the DFT Primitive", *The 10th Conference on Foundations of Software Technology and Theoretical Computer Science* , Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

J. Reif, J. Canny, and A. Page, "An Exact Algorithm for Kinodynamic Planning in the Plane", *Symposium on Computational Geometry*, San Francisco, June, 1990.

J. Reif and S. Sen, "Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July,1990.

J. Reif, "Efficient Parallel Algorithms: Theory and Practice", *SIAM 35th Anniversary Meeting*, Denver, CO, Oct. 1987. Also *XI World Computer Congress*, IFIP 89, San Francisco, CA , 1989.

J. Reif and H. Djidjev, "An Efficient Algorithm for the Genus Problem", draft, April 1989.

J. Reif and H. Gazit, "A Parallel Planar Graph Isomorphism Algorithm", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July, 1990.

J. Reif and V. Pan, "Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families", submitted for publication, 1990.

J. Reif and S. Sen, "Randomized Parallel Algorithms", workshop on Capabilities and Limitations of Parallel Computing, IBM, San Jose, CA, Dec. 1988. Also in *Information Processing* 89, G. Ritter (ed) Elsevier Science Publishers, North Hollond, IFIP, 1989, pp. 455-458 , and as Randomization in Parallel Algorithms and its Impact on Computational Geometry, in Optimal Algorithms, H. Djidjev editor, Springer-Verlag Lecture Notes in Computer Science 401, 1989, 1-8.

J. Reif, R. Paturi, and S. Rajasekaran, "The light bulb problem", presented at Workshop on Computational Learning Theory as Efficient and Robust Learning Using Statistical Bootstrap, Morgan Kaufmann Pub., Santa Cruz, CA, Aug. 1989. Submitted for journal publication.

J. Reif, D. Tygar, and A. Yoshida, "The Computation Complexity of Optical Beam Tracing", *31th IEEE Symposium on Foundations of Computer Science*,, Saint Louis, Missouri, October, 1990.

J. Reif and A. Tyagi, "Energy Complexity of Optical-Computations", to appear in, *The 2nd IEEE Symposium on Parallel and Distributed Processing*. December, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Hector, "BLITZEN: a highly integrated, massively parallel machine", *2nd Symposium on Frontiers of Massively Parallel Computation*, Fairfax, VA, Oct. 1988. Also in *Journal of Parallel and Distributed Computing*, Feb. 1990.

J. Reif and V. Ramachandran, "An optimal parallel algorithm for planarity (with V. Ramachandran). 30th Annual Symposium on Foundations of Computer Science, Durham, NC, Oct, 1989, pp. 282-287. Also University of Texas at Austin Technical Report TR-90-15, June 1990. Also invited to special issue of Journal of Algorithms,1990.

J. Reif and V. Pan, "Fast and efficient parallel solution of dense linear systems", *Computers and Mathematics with Applications* vol. 17, no. 11, pp. 1481-1491, 1989.

J. Reif and P. Gacs, "A simple three-dimensional real-time reliable cellular array", *17th Annual Symposium on Theory of Computing*, 388-395, Providence, RI, 1985.Accepted for publication by *Journal of Computer and System Sciences*, vol. 36, no. 2, p. 125-147, April 1990.

# DARPA/ONR Quarterly Report

| | |
|---|---|
| CONTRACT NUMBER: | N00014-88-K-0458 |
| CONTRACTOR: | Duke University |
| ARPA ORDER NO: | 4331714-01/4-6-88 |
| PROGRAM NAME: | N00014 |
| CONTRACT AMOUNT: | 696,899.00 |
| EFFECTIVE DATE OF CONTRACT: | July 1, 1988 |
| EXPIRATION DATE OF CONTRACT: | June 30, 1991 |
| PRINCIPAL INVESTIGATOR: | John H. Reif |
| TELEPHONE NUMBER: | (919) 660-6568 |
| SHORT TITLE OF WORK: | Parallel Algorithms Derivation |
| REPORTING PERIOD: | 1 Oct. 1991 — 30 Dec. 1991 |

## DESCRIPTION OF PROGRESS--

Special note: We are involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

### 1. Mike Landis (A.B.D.) with Robert Wagner and John Reif:

Summary:

We have completed our work in the investigation of how to map context-free grammar recognition onto systolic arrays. We are currently in the final phases of the preparation of a technical report which will document this work. Our current research efforts are to extend our method to other algorithms. We are just beginning this investigation. We plan to submit our work for publication in a journal after doing this extension, as described below.

Details:
We have developed a new method for mapping algorithms into parallel architectures. This new method works very well for a class of dynamic programming problems, including CFG recognition

In our work, we define an instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analogous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture. To ease this task, we have proven that a restricted form of recurrence equation, called a quasi-uniform recurrence equation (QURE), is computationally equivalent to a class of systolic array architectures. The class of QUREs is a broad class, the main restriction being that the recurrence must exhibit a finite set of dependency vectors over all computations. Expressing an algorithm as a recurrence equation, the problem of mapping the algorithm onto a systolic array becomes the problem of translating the recurrence equation into QURE form. It is also very easy to translate between the DAG representation and the recurrence relation.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms, especially dynamic programming algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and label its arcs and nodes to meet two constraints: The number of copies of each computation that can exist in each timestep, and the number of data that each computation can receive in each timestep. To implement these constraints, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build a time-availability table t (i,j) and vector T (j) where each t(i,j) is filled with the time that each computation point C(i) is available for use

in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

In building this table, we also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the table t(i,j).

3) We then augment this table with information about the source through which each datum may be passed.

4) We then invert this table to determine what data is passed on which computation points at each time.

Current investigation on the extension of our method mostly involves step 1. Deriving the computational indexing scheme is analagous to solving a specific graph layout problem. In order to achieve a QURE as the result of our method, the nodes in the DAG must be labeled in a multi-dimensional coordinate system such that the computation of each node only depends upon nodes that are a fixed distance away. We are currently investigating a shortest-path method to perform this indexing, which promises to work for a broad class of algorithms -- not just dynamic programming.

**2. Sri Krishnan (graduate student) with John Reif:**

M. Karr, S. Krishnan and J. Reif "Derivation of the Ellipsoid Algorithm" Draft, 1990.

Krishnan is working on derivations of the ellipsoid algorithm for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Khachian Ellipsoid Algorithm.This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. The work that Krishnan has done ( in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a smal! well-chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming.In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

**3. Doreen Yen (graduate student) with John Reif:**

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction," is now written.

This paper uses a denotational definition derivation to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential recursive

functional algorithm retains quantifiers and set notation. It is extracted from a denotational definition of a monotonic continuous functional whose least fixed point is a function consistent with the problem specification.

The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, Algorithm Cu, for computing the connected component which contains the vertex u.

An initial parallel connected components algorithm is derived from Algorithm Cu. The parallelism arises naturally by using a higher order function, map, which applies Algorithm Cu to each element of the vertex set. A more efficient parallel algorithm, fcc, is also obtained by denotational definition derivation. Its transition to an iterative algorithm is by discovery of lemmas showing the number of iterations needed to reach the fixed point and choice of efficient data structures to implement disjoint set union. An arbitrary constraint upon the data structure is imposed to obtain the parallel connected components algorithm connect, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm connect can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call stream compaction, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by O(log n).

A proof is given to show the O(log n) speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is O(log n).

### 4. Subhrajit Bhattacharya with John Reif:

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

### 5. Hillel Gazit with John Reif

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with
$P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes O(log(n))

time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs
$2 \log(m) \log(n) + O(\log(n))$ random bits. The number of random bits can be decreased to $O(\log(n))$ by increasing the processors number to $n^{3/2+e}$. This algorithm significantly improves the previous results of $n^4$ processors.

### 7. Various Authors--*Synthesis of Parallel Algorithms* (edited by Reif)

Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting. *Synthesis of Parallel Algorithms* will be published by Morgan Kaufmann in Summer 1991.

At the present writing, some 27 researchers have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

#### Participating Authors and Topics

| | |
|---|---|
| Guy Blelloch | Prefix Sums and Applications |
| Margaret Reid-Miller | Parallel Tree Contraction and Applications |
| Sara Baase | Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques |
| Uzi Vishkin | Advanced Parallel Prefix-sums, List Ranking and Connectivity |
| Hillel Gazit | Randomized Parallel Connectivity |
| Vijaya Ramachandran | Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity |
| Vijay Vazirani | Parallel Graph Matching |
| Erich Kaltofen | Dynamic Parallel Evaluation of Computation DAGs |
| Jeffrey Ullman | Parallel Evaluation of Logic Queries |
| Philip Klein | Parallel Algorithms for Chordal Graphs |

**Participating Authors and Topics (continued)**

| | |
|---|---|
| Victor Pan | Parallel Solution of Sparse Linear and Path Systems |
| Andrew Goldberg | Parallel Algorithms for Network Flow Problems |
| Stephen Tate | Newton Iteration and Integer Division |
| Joachim von zur Gathen | Parallel Linear Algebra |
| Dexter Kozen and Doug Ierardi | Parallel Resultant Computation |
| Richard Cole | Parallel Merge Sort |
| Mikhail Atallah and Michael Goodrich | Deterministic Parallel Computational Geometry |
| Sandeep Sen and Rajasekaran | Random Sampling Techniques and Parallel Algorithms Design Sanguthevar |
| Philip Gibbons | Asynchronous PRAM Algorithms |
| Raymond Greenlaw | Polynomial Competeness and Parallel Computation |
| Baruch Schieber | Parallel Lowest Common Ancestor Computation |
| Faith Fich | The Parallel Random Access Machine |

**8. Duke Algorithm Derivation Seminar:**

Participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

**9. Researchers supported (other than PI):**

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student, then post-doc
Sandeep Sen, graduate student, then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

**10. Degrees awarded:**

Sandeep Sen received his Ph.D. from Duke and was here as a post-doc. Steve Tate and Lars Nyland received their Ph.D.s in January 1991 under Reif and are both remaining at Duke as post-docs.

## 11. Papers

H. Gazit, "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph," (extended version), accepted to SIAM J. of Computing, 1990.

H. Gazit, "An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph," submitted for publication, 1990.

H. Gazit, "Finding the Diameter of a Directed Graph," submitted for publication, 1990.

H. Gazit, "Parallel Algorithms for Connectivity, Ear Decomposition and st-Numbering of Planar Graph," accepted to the Fifth International Parallel Processing Symposium, 1990.

Nyland, L., "The Design of a Prototyping Programming Language for Parallel and Sequential Algorithms," Ph.D. Thesis, Duke University, 1990.

Tate, Steve, "Arithmetic Circuit Complexity and Motion Planning," Ph.D. Thesis, Duke University, 1990.

R. Wagner and M. Landis, "Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement", submitted for publication, 1990.

J. Reif and S. Bhattacharya, "Derivation of Efficient Parallel Sorting Algorithm", draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and V. Pan, "On the Bit Complexity of Discrete Approximations to PDEs", *International Colloquium on Automata, Languages, and Programming*, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990.

J. Reif and J. Storer, "A Parallel Architecture for High Speed Data Compression", *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, Maryland, October, 1990.

J. Reif and A. Yoshida, "Optical Expanders", manuscript, August 1989.

J. Reif, "Efficient Algorithms for Optical Computing with the DFT Primitive", *The 10th Conference on Foundations of Software Technology and Theoretical Computer Science* , Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

J. Reif, J. Canny, and A. Page, "An Exact Algorithm for Kinodynamic Planning in the Plane", *Symposium on Computational Geometry*, San Francisco, June, 1990.

J. Reif and S. Sen, "Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July,1990.

J. Reif, "Efficient Parallel Algorithms: Theory and Practice", *SIAM 35th Anniversary Meeting*, Denver, CO, Oct. 1987. Also *XI World Computer Congress*, IFIP 89, San Francisco, CA , 1989.

J. Reif and H. Djidjev, "An Efficient Algorithm for the Genus Problem", draft, April 1989.

J. Reif and H. Gazit, "A Parallel Planar Graph Isomorphism Algorithm", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July, 1990.

J. Reif and V. Pan, "Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families", submitted for publication, 1990.

J. Reif and S. Sen, "Randomized Parallel Algorithms", workshop on Capabilities and Limitations of Parallel Computing, IBM, San Jose, CA, Dec. 1988. Also in *Information Processing* 89, G. Ritter (ed) Elsevier Science Publishers, North Hollond, IFIP, 1989, pp. 455-458 , and as Randomization in Parallel Algorithms and its Impact on Computational Geometry, in Optimal Algorithms, H. Djidjev editor, Springer-Verlag Lecture Notes in Computer Science 401, 1989, 1-8.

J. Reif, R. Paturi, and S. Rajasekaran, "The light bulb problem", presented at Workshop on Computational Learning Theory as Efficient and Robust Learning Using Statistical Bootstrap, Morgan Kaufmann Pub., Santa Cruz, CA, Aug. 1989. Submitted for journal publication.

J. Reif, D. Tygar, and A. Yoshida, "The Computation Complexity of Optical Beam Tracing", *31th IEEE Symposium on Foundations of Computer Science*,, Saint Louis, Missouri, October, 1990.

J. Reif and A. Tyagi, "Energy Complexity of Optical-Computations", to appear in, *The 2nd IEEE Symposium on Parallel and Distributed Processing*. December, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Hector, "BLITZEN: a highly integrated, massively parallel machine", *2nd Symposium on Frontiers of Massively Parallel Computation*, Fairfax, VA, Oct. 1988. Also in *Journal of Parallel and Distributed Computing*, Feb. 1990.

J. Reif and V. Ramachandran, "An optimal parallel algorithm for planarity (with V. Ramachandran). 30th Annual Symposium on Foundations of Computer Science, Durham, NC, Oct, 1989, pp. 282-287. Also University of Texas at Austin Technical Report TR-90-15, June 1990. Also invited to special issue of Journal of Algorithms,1990.

J. Reif and V. Pan, "Fast and efficient parallel solution of dense linear systems", *Computers and Mathematics with Applications* vol. 17, no. 11, pp. 1481-1491, 1989.

J. Reif and P. Gacs, "A simple three-dimensional real-time reliable cellular array", *17th Annual Symposium on Theory of Computing*, 388-395, Providence, RI, 1985.Accepted for publication by *Journal of Computer and System Sciences*, vol. 36, no. 2, p. 125-147, April 1990.

# ONR Annual Report


| | |
|---|---|
| Principal Investigator: | John H. Reif |
| PI Institution: | Duke University |
| PI Phone Number: | (919) 660-6568 |
| PI E-mail Address: | reif@cs.duke.edu |



| | |
|---|---|
| Grant or Contract Title: | Parallel Algorithms Derivation |
| Grant or Contract Number: | N00014-88-K-0458 |
| Reporting Period: | 1 Oct 90 - 30 Sep 91 |

ONR Annual Report

Principal Investigator(s) Name(s): John H. Reif
PI Institution: Duke University
PI Phone Number: (919) 660-6568
PI E-mail Address: reif@cs.duke.edu

Grant or Contract Title: Parallel Algorithms Derivation
Grant or Contract Number: N00014-88-K-0458
Reporting Period: 1 Oct 90 - 30 Sep 91

**1. Productivity measures:**

Refereed papers submitted but not yet published: 7

Refereed papers published: 3

Unrefereed reports and articles: none

Books or parts thereof submitted but not yet published:

*Synthesis of Parallel Algorithms*, Morgan Kaufmann Publishers

Books or parts thereof published: none

Patents filed but not yet granted: none

Patents granted (include software copyrights): none

Invited presentations: 11

Contributed presentations: 9

Honors received: none

Prizes or awards received: none

Promotions obtained: none (Reif is a full Professor at Duke)

Graduate students supported >= 25% of full time: 6

(Salman Azhar, Michael Landis, Tassos Markas, Steve Tate, Doreen Yen, Akitoshi Yoshida)

Post-docs supported >= 25% of full time: none

Minorities supported: none

ONR Annual Report

Principal Investigator(s) Name(s): John H. Reif
PI Institution: Duke University
PI Phone Number: (919) 660-6568
PI E-mail Address: reif@cs.duke.edu

Grant or Contract Title: Parallel Algorithms Derivation
Grant or Contract Number: N00014-88-K-0458
Reporting Period: 1 Oct 90 - 30 Sep 91

**2. Detailed summary of technical progress:**

OBJECTIVE

Our objective is the systematic development of rules and techniques for derivation of various classes of parallel algorithm. We are stressing the development of fundamental derivation techniques that can be utilized in as wide a class of parallel algorithms as possible. We intend that the derivation techniques developed in this project will be implemented within the various other DARPA sponsored derivation systems, such as at Kestrel.

APPROACH

Our approach is a derivational and synthesis approach: to make the derivation and synthesis of a family of related algorithms (rather than a single algorithm) from fundamental algorithmic techniques. We developed rules and techniques for derivation of parallel algorithms. The specific parallel algorithms derived were carefully chosen to be as fundamental as possible. We stressed the development of fundamental derivation techniques that can be utilized in as wide a class of parallel algorithms as possible. This work yielded derivations of algorithms in the following areas:

-Systolic algorithms for various fixed connection networks
-Randomized parallel algorithms
-Parallel algorithms for tree and graph problems
-Parallel algorithms for algebraic problems

This included graph connectivity, tree contraction, CFL parsing, searching and sorting. We concentrated our work on certain fundamental derivatives which we deem most important. This allowed us to leverage our efforts and thus do a more extensive and in-depth development of derivation techniques for these classes of fundamental problems. Various experts in parallel algorithms worked collaboratively with us on derivations of parallel algorithms in their area of expertise. The derivation techniques developed in this project are beginning to be implemented within the various other DARPA sponsored derivation systems, such as (in proposed collaboration with D. Smith) at Kestrel institute.

PROGRESS

We have completed many of the parallel algorithm derivations in the areas which which we had proposed in the contract and have moreover completed (with collaboration from a large part of the parallel algorithm community) a major text in parallel algorithm synthesis. We are now progressing to a further stage of our research; namely the topic of deriving parallel algorithm implementations on actual parallel machines. This seems a logical consequence of our previous work, and though more practical than our previous research, should draw upon the theoretical techniques we have developed to date.

RECENT ACCOMPLISHMENTS

Special note: We are involved in a number of collaborations with other DARPA/ONR contractors: Mike Karr, of Software Options, Inc. (see item #2 of recent accomplishments).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

1. Mike Landis (graduate student) and Robert Wagner (Duke Associate Professor)

We developed a new method for mapping algorithms into parallel architectures. This new method works very well for a class of dynamic programming problems, including CFG recognition. In particular, we considered the task of mapping context-free grammar recognition onto systolic arrays. This work is described in a paper by Landis and Wagner, "Derivation of Systolic Algorithms", to be published

2. Sri Krishnan (graduate student) with John Reif:

Krishnan is working on derivations of the ellipsoid algorithm for linear programming, and has recently coauthored a paper on this subject (M. Karr, S. Krishnan and J. Reif "Derivation of the Ellipsoid Algorithm" Duke Technical Report C-1991-17) with Mike Kerr, another DARPA/ONR contractor that formally derives the Khachian Ellipsoid Algorithm. This paper examines some important aspects of linear programming.

3. Doreen Yen (graduate student) with John Reif:

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A completed paper, "Derivation of Parallel Graph Algorithms via Stream Compaction," is now written and has been submitted for publication.

4. Subhrajit Bhattacharya (graduate student) with John Reif:

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. This work derived these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

5. Hillel Gazit (postdoc at Duke) with John Reif

In the paper "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph" we present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is logarithmic with optimal processor bound. In the paper "A Randomized Parallel Algorithm for Planar Graph Isomorphism" we derive an efficient parallel randomized algorithm for finding if two planar graphs are isomorphic using a derivation from separator algorithms.

6 Sandeep Sen (recent Ph.D. under Reif) with Reif

"Randomized Parallel Algorithm in Computational Geometry". We developed efficient parallel algorithms for load balancing and searching in fixed connection networks. This work was reported in a paper in SPAA90.

6. Ming Kao (assistant Professor, travel partially supported by Reif)

Kao developed parallel algorithms for k-connectivity and a new parallel search method related to parallel breadth first search, which is much more processor efficient.

## TWO SIGNIFICANT EVENTS

(1) In addition to our own work in parallel algorithm derivation in the last three years, we have organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a te `book on parallel algorithms synthesis. The text is now completed and is titled *Synthesis of Parallel Algorithms* (edited by Reif). It is being published by Morgan Kaufmann. This text draws together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting as well have practical applications. The text contains 22 chapters and some 27 researchers have written chapters for this text (note: a number of chapters are coauthored with Reif or other researchers). Reif collaborated with several of these researchers, and has invited them to visit Duke. Each chapter begins with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution.

(2) Robert Paige (Courant Institute, NYU) and John Reif (Duke Univ) organized a workshop from Aug. 31 - Sep. 2, 1991, "On Parallel Algorithm Derivation and Program Transformation."

Description: Transformational programming and parallel computation are two emerging fields that may ultimately depend on each other for success. Perhaps, because ad hoc programming on sequential machines is so straightforward, sequential programming methodology and transformational methodology has had impact mostly only within the academic community. However, because ad hoc programming for parallel machines is so hard, and because progress in because progress in software construction has lagged behind architectural advances for such machines, there is a much greater need to develop parallel programming and transformational methodologies. This workshop stimulated investigation of formal ways to overcome problems of parallel computation - with respect to both software development and algorithm design. We invited 40 distinguished computer scientists from two different communities - transformational programming and parallel algorithm design - to discuss programming, transformational and compiler methodologies for parallel architectures, and algorithmic paradigms, techniques, and tools for parallel machine models.

## CONCURRENT and RELATED WORK

We are continuing our research under support of a further DARPA/ISTO contract, "Derivation and Analysis Tools for the Synthesis and Implementation of Parallel Algorithms". The existence of efficient parallel algorithms is potentially a major asset to the DARPA community, but few are targeted to existing machines, and many are far too complex to be of practical use. Therefore, there is a substantial gap between the existence of theoretical parallel algorithms and their implementation on actual parallel machines. The goal of this proposal is close the gap between theory and practice by the tasks listed below.

Our approach is a derivational and synthesis approach: to make the derivation and synthesis of a family of related algorithms (rather than a single algorithm) from fundamental algorithmic techniques, and then to refine their implementations onto a family of parallel machines (rather than targeting the entire effort to a single machine). This process will be substantially aided by the techniques and tools summarized below.

1. Techniques and Tools for semi-automatic implementations of parallel algorithms: Process to Processor Mapping. A key task is of mapping from virtual processes to actual processors. We will develop techniques for semi-automatic implementations of parallel algorithms onto various networks, and will develop prototype implementations of these techniques.

2. Techniques and Tools for timing analysis of parallel algorithms and their Implementations. We are working to improve the techniques for time and space analysis of parallel algorithms to the point that it is feasible to use semi-automatic methods.

## 3. A COMMON IMPLEMENTATION PLATFORM

An important task we are working on is the construction of a common implementation platform, so that programs can run on multiple types of parallel machines (such at the MIMD parallel BUTTERFLY, WARP, and CRAY YMP, as well as the SIMD parallel CONNECTION machine (hypercube connected) and MASSPAR machine (x-grid connected)). (Note that this is not a high level programing language task, but instead the implementation of certain important control primitives on various machines.). This will clearly have impact on the DARPA efforts in Software and HPC.

4. Concrete Implementations on Actual Parallel Machines. We are validifying our approach by some prototype implementations of parallel algorithms for two representative classes of problems.

ONR Annual Report

Principal Investigator(s) Name(s): John H. Reif
PI Institution: Duke University
PI Phone Number: (919) 660-6568
PI E-mail Address: reif@cs.duke.edu
Grant or Contract Title: Parallel Algorithms Derivation
Grant or Contract Number: N00014-88-K-0458
Reporting Period: 1 Oct 90 - 30 Sep 91

**3. Lists of publications, presentations and reports.**

**Refereed papers submitted but not yet published:**

(1) Expected parallel time and sequential space complexity of graph and problems (with P. Spirakis). Accepted for publication in special issue of Algorithmica, 1991.

(2) Derivation of Parallel Connectivity Algorithms via Stream Contraction (with D. Yen), submitted for publication 1990, Duke University Technical Report, 1989.

(3) Derivation of Linear Programming Algorithms (with Mike Kerr and Sri Krishnan), submitted for publication, 1990.

Error Resilient One-Way Dynamic Communication, (with J. A. Storer), November 1990, submitted for publication, October 1990.

(4) Continuous Alternation, (with S. Tate), invited for special issue of Journal of Algorithmica, edited by B. Donald, 1991.

(5) Towards Randomized Strongly Polynomial Algorithms for Linear Programming (with S. Krishnan), submitted for publication, 1990. Duke University Technical Report CS-1991-18.

(6) Fast Computations of Vector Quantization Algorithms (with T. Markas), NASA Technical Report TR-91-58, 1991.

(7) On the Bit Complexity of Discrete Approximations to PDEs (with V. Pan). International Colloquium on Automata, Languages, and Programming, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990. Appeared in Computers and Mathematics with Applications, Vol 20, no 2, 9-16, 1990. Also additional results in "Compact Multigrid", in *SIAM Journal of Scientific and Statistical Computing*, accepted and to appear 1991.

**Refereed papers published:**

(1) Optimal Size Integer Division Circuits (with S. Tate). 21st Annual ACM Symposium on Theory of Computing, pp. 264-273, Seattle, WA, May 1989. Presented at meeting on Efficient Algorithms, Mathematisches Forschungsinstitut, W. Germany, September 1989. Also published in SIAM Journal of Computing, Vol 19, No 5, 912-924, October 1990.

(5) An Exact Algorithm for Kinodynamic Planning in the Plane (with J. Canny and A. Page). Symposium on Computational Geometry, San Francisco, CA, June 1990. Also, journal publication in Discrete and Computational Geometry, vol. 6, 461-484, 1991.

Parallel tree contraction and its application (with G. Miller). 26th IEEE Symposium on Foundations of Computer Science, 478-489, Portland, OR, October 1985. Part I in Randomness and Computation, Vol 5., S. Micoli, ed., JAI Press, 1989, pp. 47-72.Parallel Tree Contraction Part II: Further Applications, *SIAM Journal of Computing*, Vol 20, no 6, December 1991.

**Unrefereed reports and articles:**

none

**Books or parts thereof submitted but not yet published:**

*Synthesis of Parallel Algorithms*, to be published by Morgan Kaufmann

**Books or parts thereof published:**

none

**Invited Presentations:**

(1) Panel at *3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, MD, October 1990.

(2) Member of Program Committee, *Symposium on Frontiers of Massively Parallel Computing*, Fairfax, Virginia, October 1990.

(3) *31st IEEE Symposium on Foundations of Computer Science*, St. Louis, MO, October 1990.

(4) *The 2nd IEEE Symposium on Parallel and Distributed Processing*, Dallas, TX, December 1990.

(5) *10th Conference on Foundations of Software Technology and Theoretical Computer Science*, Bangalor, India, December 1990.

(6) *Advanced Research in VLSI Conference*, MIT Press, March 1991, Santa Cruz, CA.

(7) Program co-Chairman, *First IEEE Conference on Parallel Algorithms*, Snowbird, UT, April, 1991.

(8) *23rd Annual Symposium on Theory of Computing*, New Orleans, LA, May 1991, pp 337-345.

(9) *4th SIAM Conference in Applied Linear Algebra*, September 1991, Minneapolis, MN

(10) Member of Program Committee, *32nd IEEE Symposium on Foundations of Computer Science*, October 1991.

(11) *3rd IEEE Symposium on Parallel and Distributed Processing*, Dallas, TX, December 1991.

**Contributed Presentations:**

(1) A Parallel Architecture for High Speed Data Compression (with J.A. Storer), *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, MD, October 1990.

(2) The Computation Complexity of Optical Beam Tracing (with D. Tygar and A. Yoshida). *31st IEEE Symposium on Foundations of Computer Science,* St. Louis, MO, October 1990. Also submitted for Journal publication as "The Computation Complexity of Ray Tracing".

(3) Energy Complexity of Optical Computations (with A. Tyagi), appeared in *The 2nd IEEE Symposium on Parallel and Distributed Processing,* Dallas, TX, December 1990, pp. 14-21.

(4) Efficient Algorithms for Optical Computing with the DFT Primitive (with A. Tyagi) *10th Conference on Foundations of Software Technology and Theoretical Computer Science* , Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

(5) An Optical Delay Line Memory Model with Efficient Algorithms, (with A. Tyagi), *Advanced Research in VLSI Conference,* MIT Press, March 1991, Santa Cruz, CA.

(6) Image Compression Methods with Distortion Controlled Capabilities (with T. Markas), Proceedings of Data Compression Conference (DCC 91), Snowbird, UT, April 1991, pp. 93-102.

(7) An Efficient Algorithm for the Genus Problem with Explicit Construction of Forbidden Subgraphs (with H. Djidjev). *23rd Annual Symposium on Theory of Computing*, New Orleans, LA, May 1991, pp 337-345.

(8) Decreasing the Precision of Linear Algebra Computations by Using Compact Multigrid and Backward Interval Analysis (with V. Pan), *4th SIAM Conference in Applied Linear Algebra,* Minneapolis, MN, September 1991.

(9) Prototyping Parallel and Distributed Programs in Proteus, (with P. H. Mills, L.S.Nyland, J.F. Prins, R.A. Wagner), *3rd IEEE Symposium on Parallel and Distributed Processing,* Dallas, TX, December 1991

ONR Annual Report

Principal Investigator(s) Name(s): John H. Reif
PI Institution: Duke University
PI Phone Number: (919) 660-6568
PI E-mail Address: reif@cs.duke.edu
Grant or Contract Title: Parallel Algorithms Derivation
Grant or Contract Number: N00014-88-K-0458
Reporting Period: 1 Oct 90 - 30 Sep 91

## 4. Transitions and DoD interactions:

Reif attended a DARPA software meeting in fall, 1991 where he gave a invited talk on the work he has done on prototyping languages; he also particularly described a language for prototyping parallel and distributed computations.

Special note: We are involved in a number of collaborations with other ONR contractors:

(a) Mike Karr, of Software Options, Inc. (see item #2 of recent accomplishments).
(b) Doug Smith, Kestrel Institute, investigating derivation of parallel dynamic programming algorithms

ONR Annual Report

Principal Investigator(s) Name(s): John H. Reif
PI Institution: Duke University
PI Phone Number: (919) 660-6568
PI E-mail Address: reif@cs.duke.edu

Grant or Contract Title: Parallel Algorithms Derivation
Grant or Contract Number: N00014-88-K-0458
Reporting Period: 1 Oct 90 - 30 Sep 91

**5. Software and hardware prototypes:**

none