AD-A278 377

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

(2)

...ed to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, ... reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this ... burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, ... and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED
FINAL/01 NOV 90 TO 31 OCT 93

4. TITLE AND SUBTITLE
PROGRAM COMPOSITION (U)

5. FUNDING NUMBERS
2304/A2
AFOSR-91-0070

6. AUTHOR(S)
Professor K. Chandy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
California Institute of Technology
Computer Science 256-80
Pasadena, CA 91125

8. PERFORMING ORGANIZATION REPORT NUMBER
AFOSR-TR- 94 0168

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 DUNCAN AVE, SUITE B115
BOLLING AFB DC 20332-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFOSR-91-0070

DTIC ELECTE APR 2 1 1994 S B D

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

12b. DISTRIBUTION CODE
UL

13. ABSTRACT *(Maximum 200 words)*

Work done under this grant falls under four main areas: (1) Theory of Concurrent Systems - has led to much simpler ways of demonstrating the correctness of concurrent programs, (2) Use of Ada in Parallel Programming - exploring simple extensions of Ada that would make it suitable for shared-memory multiprocessor machines, (3) Parallel Pradigm Integration - researchers developed software and methods that help in developing large parallel applications incorporoating both functional and data parallelism, (4) Language Development - this grant initiated work on PCN (Program Composition Notation), a very simple language for composing programs in Fortran and C;PCN has been ftp'd at over 300 sites, including Air Force Laboratories, and is being used at several universities for teaching parallel programming.

DTIC QUALITY INSPECTED 3

14. SUBJECT TERMS

15. NUMBER OF PAGES
5

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR(SAME AS REPORT)

94 4 20 095

FEB 07 1994

03010

AFOSR-TR- 94 0168

Approved for public release;
distribution unlimited.

Final Technical Report
AFOSR 91-0070
Program Composition

K. Mani Chandy
California Institute of Technology

31 January 1994

94-12050

1 Work on this Grant

The earlier work done under this grant falls under four main areas:

Theory of Concurrent Systems This research has led to much simpler ways of demonstrating the correctness of concurrent programs. This work shows that a small set of rules, similar to those in existing theorem provers, can be used to reason about parallel programs.

Papers partially supported by the grant include:

1. K.M. Chandy, "Properties of Concurrent Programs," to appear in *Formal Aspects of Computing Science*, 1994.
2. K.M.Chandy and B.A.Sanders, "Conjunctive Predicate Transformers for Reasoning about Concurrent Computation," Technical Report 197, ETH Departement Informatik, 1993, submitted *Science of Computer Programming*.
3. K.M.Chandy and B.A.Sanders, "Compositional Reasoning about Safety and Progress Properties for Message Passing Systems," submitted Conference on Formal Methods, 1994.
4. K.M.Chandy and B.A.Sanders, "Compositional Proofs of Concurrent Systems," in preparation.
5. U.Binau, "Distributed Diners: From UNITY Specification to CC++ Implementation," Caltech Computer Science Tech. Report, CS-TR-93-20.

Use of Ada in Parallel Programming Our attempts at integrating scientific numeric computation within larger embedded reactive systems, led us to explore the use of Ada. Our reasons for choosing Ada as a language for exploration are:

1. Ada is the language of choice for many DoD applications.
2. Ada is a standardized concurrent programming language.
3. Many reactive applications have been written in Ada.

Our emphasis was on exploring simple extensions of Ada that would make the language suitable for shared-memory multiprocessor machines.

Availability Codes
Dist | Avail and/or Special
A-1

We developed a simple extension, implemented a simple (though somewhat inefficient) compiler for the extension, and developed and proved several programs using the extension. The focus of the Ada applications was on a functional template that allowed sequential programs to be transformed into parallel programs in a simple systematic way.

Papers on this topic include:

1. J. Thornley, "Parallel Programming with Declarative Ada," Caltech Computer Science Tech. Report, CS-TR-93-03.
2. J. Thornley, "The Programming Language *Declarative Ada* Reference Manual," Caltech Computer Science Tech. Report, CS-TR-93-04.
3. J. Thornley, "A Collection of Declarative Ada Example Programs," Caltech Computer Science Tech. Report, CS-TR-93-05.
4. J. Thornley, "Functional Parallel Programming with Ada," to be submitted to TriAda, 94.

Parallel Paradigm Integration. We developed software and methods that help in developing large parallel applications that incorporate both functional and data parallelism. A particular emphasis was on applications with a task-parallel framework where units within this framework were data-parallel. Such applications arise in digital signal processing and many other areas.

We developed a method of integrating Fortran with PCN to obtain an integrated task and data parallel programming system, and developed a few applications using the system. The applications fit an overall template in which the program structure is data-parallel and the nodes within the program are task-parallel.

We also did research on integrating functional programming with imperative parallel programming. Functional programming has the advantage of having clean semantics, but a very small percentage of air force personnel use functional programming languages. We developed and then implemented systematic transformations from functional programs to parallel programs in C++, thus having the advantages of functional specifications and implementations in languages that are more familiar to DoD.

Papers in this area include:

1. B. Massingill, "Integrating Task and Data Parallelism," Caltech Computer Science Tech. Report, CS-TR-93-01.
2. J. Thornley, "Integrating Functional and Parallel Programming: CC++ Solutions to the Salishan Problems," submitted to IPPS, 1993.

Development of Templates We developed three templates for mesh computations, spectral computations and divide and conquer. We have parallelized a large airshed quality model that predicts smog (over 30 separate pollutants) in the L.A. basin; this parallelization was done using the spectral tempate. We are parallelizing a fluid dynamics application using the mesh template, and we have parallelized several combinatorics applications using the divide and conquer template. At this point, we are working with people in other departments (specifically Applied Mathematics) in completing performance measurements prior to publishing the results.

Language Development This grant initiated the work on PCN, Program Composition Notation. The last release of PCN was ftp'd at over 300 sites, including airforce laboratories. Several universities, including Caltech, use PCN for teaching parallel programming. The book, *Introduction to Parallel Programming*, specifically mentions grants from AFOSR for supporting the work on PCN.

The focus of PCN was to develop a very simple language that could be used to compose programs in Fortran and C. We wanted to implement a language with just four statements: parallel composition, sequential composition, choice and recursion. The use of PCN in several institutions is an indication of the simplicity of the language.

Another goal of PCN was to develop applications transportable across a wide range of architectures. It is now executed on networks of workstations, multicomputers and shared-memory multiprocessor systems.

PCN is a success in the sense that it is used quite widely for teaching and research, and it is a simple language with clean semantics. We are now using ideas from PCN to develop parallel extensions of Fortran, C++ and Ada; we expect these implementations to be much more widely used because of the prevalence of the base languages.

K.M.Chandy, and S.Taylor, *An Introduction to Parallel Programming*, Jones and Bartlett, Publishers, 1992.

2 Technology Transfer

PCN PCN has been used in a variety of sites including airforce laboratories, Argonne National Laboratory and the Aerospace Corp. It continues to be available from Argonne by ftp, and remains in wide use.

Collaboration with Phillips Laboratories Ms. Berna Massingill, a member of the group at Caltech, spent part of her summer at Phillips Air Force Labs in Albuquerque. She worked on archetypes for electromagnetics codes, and has been parallelizing one particular code of interest to Phillips.

Collaboration with Rome Laboratories Ms. Milissa Benincasa and Carla Burns have been working with Mani Chandy on software engineering methods for parallel program development. We expect to publish a paper in early 1994 describing our results.

Satellite Design Ms. Massingill is working with JPL on a multidisciplinary integration of CAD tools for satellite design. We expect to use this effort to help us develop archetypes for multidisciplinary CAD applications.

Environmental Engineering The spectral archetype has been used to parallize smog model code. This code was developed by Prof. Seinfeld, and Donald Dabdub at Caltech, and our group showed how templates could be used to simplify development. The code runs on the IBM SP1, and was demonstrated at Supercomputing 94.

Computational Fluid Dynamics We are now working with Prof. Dan Meiron on using templates for fluid dynamics. We expect to have a complete application with graphics and animation by the end of January.