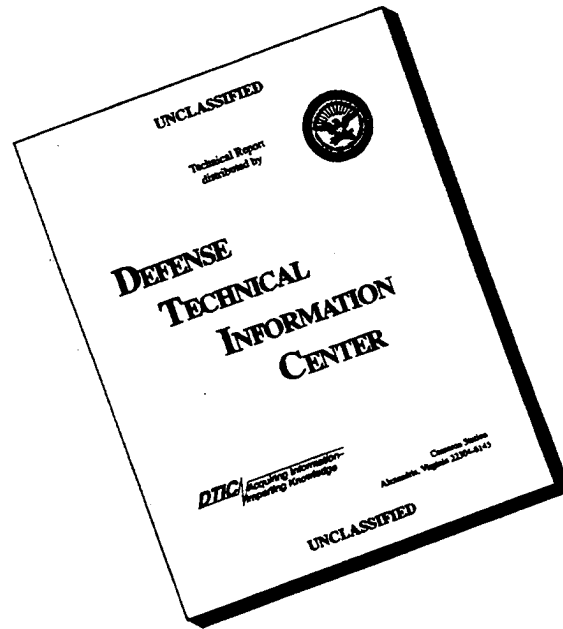# REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, 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 10/31/95	3. REPORT TYPE AND DATES COVERED Final Report / 6/1/92 - 5/31/95

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Final Report on AFOSR Grant # F49620-92-J-0240 "Graphics Interfaces & Sensor Actuator Extensions for the Process Trellis Software"	AFOSR # F49620-92-J-0240

6. AUTHOR(S)

David Gelernter et al.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Yale University
Grant and Contract Administration
12 Prospect Street
New Haven, CT.   06511 3516

8. PERFORMING ORGANIZATION REPORT NUMBER

AFOSR-TR-96
0094

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR
ATTN: PKA
110 Duncan Ave., Suite B115
Bolling AFB, DC 20332-0001

10. AGENCY REPORT NUMBER

19960320 059

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

We tested the new LAN-capable sensor/actuator package by developing a monitor controller for the Piranha adaptive parallelism environment. Piranha is a system that allows processes of a parallel application to be created dynamically (for example on newly idle LAN nodes) and removed dynamically (for example, when an owner resumes work at his node) while the computation as a whole continues without interruption. Overall research results of the AFOSR Grant # F49620-92-J-0240 support include development of the underlying Pirahna system for adaptive parallelism to manage nodes of a dedicated multiprocessor efficiently, specific Network Piranha and Multiprocessor (CM-5) applications and implementations, and the (ongoing) development of a practical software application called "Lifestreams," discussed in the attached final report.

14. SUBJECT TERMS
Adaptive parallelism, Piranha, Network Piranha, Multiprocessor Piranha, Lifestreams.

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

DTIC QUALITY INSPECTED 1

DISCLAIMER NOTICE

UNCLASSIFIED

Technical Report
distributed by

DEFENSE
TECHNICAL
INFORMATION
CENTER

DTIC Acquiring Information-
Imparting Knowledge

Cameron Station
Alexandria, Virginia 22304-6145

UNCLASSIFIED

THIS DOCUMENT IS BEST
QUALITY AVAILABLE. THE
COPY FURNISHED TO DTIC
CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO
NOT REPRODUCE LEGIBLY.

GRANT NUMBER: _____

FORM A2-2

AUGMENTATION AWARDS FOR SCIENCE & ENGINEERING RESEARCH TRAINING (AASERT) REPORTING FORM

The Department of Defense (DOD) requires certain information to evaluate the effectiveness of the AASERT program. By accepting this Grant Modification, which bestows the AASERT funds, the Grantee agrees to provide the information requested below to the Government's technical point of contact by each annual anniversary of the AASERT award date.

1. Grantee identification data: (R & T and Grant numbers found on Page 1 of Grant)

    a.   <u>Yale University</u>
        University Name

    b.   <u>F49620-92-J-0240</u>   c.   <u>FQ8671-920109-3484/S5</u>
        Grant Number   R & T Number

    d.   <u>David Gelernter</u>   e. From: 6/1/94 To: 5/31/95
        P.I. Name   AASERT Reporting Period

NOTE: Grant to which AASERT award is attached is referred to hereafter as "Parent Agreement."

2. Total funding of the Parent Agreement and the number of full-time equivalent graduate students (FTEGS) supported by the Parent Agreement during the 12-month period **prior to** the AASERT award date.

    a. Funding: $ <u>98,676</u>

    b. Number FTEGS: <u>-0-</u>

3. Total funding of the Parent Agreement and the number of FTEGS supported by the Parent Agreement during the current 12-month reporting period.

    a. Funding: $ <u>-0-</u>

    b. Number FTEGS: <u>-0-</u>

4. Total AASERT funding and the number of FTEGS and undergraduate students (UGS) supported by AASERT funds during the current 12-month reporting period.

    a. Funding: $ <u>243,075</u>

    b. Number FTEGS: <u>3</u>

    c. Number UGS: <u>-0-</u>

VERIFICATION STATEMENT: I hereby verify that all students supported by the AASERT award are U.S. citizens.

_____   <u>10/30/95</u>
Principal Investigator   Date

Final Report on AFOSR Grant F49620-92-J-0240

Abstract and Project Overview

Abstract: We tested the new LAN-capable sensor/actuator package by developing a monitor controller for the Piranha adaptive parallelism environment. Piranha is a system that allows processes of a parallel application to be created dynamically (for example on newly idle LAN nodes) and removed dynamically (for example, when an owner resumes work at his node) while the computation as a whole continues without interruption. Overall research results of the AFOSR Grant # F49620-92-J-0240 support include development of the underlying Pirahna system for adaptive parallelism to manage nodes of a dedicated multiprocessor efficiently, specific Network Piranha and Multiprocessor (CM-5) applications and implementations, and the (ongoing) development of a practical software application called "Lifestreams."

Project Overview: The Piranha systems poses a number of monitoring and control problems: it requires that the current idle/busy and "idleness criteria" status of all nodes be maintained, that predictions be developed with respect to likely future idleness patterns of each node in the pool, that the status of all Piranha applications and their behaviors be maintained and that Piranha jobs be assigned to particular idle nodes, among other issues. The problem was a good test bed for the sensor/actuator LAN trellis because it required actuator and not just sensor capability, it was inherently distributed and required LAN capacity, and it was inherently a significant, interesting problem. The Piranha-trellis we developed worked sufficiently well to suggest that the entire Piranha system (and not jusst monitor-control functions) might be structured as a trellis. A trellis-structured wide area Piranha system being developed.

The use of a system like Trellis to monitor and control a wide-area Piranha network poses one obvious difficulty. Scheduling decisions are centralized in the Trellis; we need to insure that propagation delays between available network nodes and the Trellis's scheduling modules are not such as to make any decisions of a centralized scheduler obsolete by the time they are made, and still more obsolete when they are put into effect.

The hierarchical design of the Trellis gives us a framework for addressing these problems. Our intention is to insure that relatively more time-critical decisions are made relatively closer to the nodes to which they are relevant.

Scheduling decisions that might entail moving processes from domain M to domain N are made in general by the lowest-ranking Trellis module that spans N and M; such modules will "as close as possible" to the domains they monitor. The fewer domains a scheduling module monitors, the greater the likelihood that the module can be placed "relatively close" to all of them—assuming that we use network geography as a guide when we group domains together. The higher-ranking a scheduling module—the more domains it covers, and the farther it is from its most-distant domain—the more conservative the scheduling module must be in moving processes around the network.

1

Suppose that a scheduling module is responsible for deciding whether processes should be moved from the "main campus" to the "medical school," and assume that those domains are far apart in terms of network geography. It must be willing to guess that any idle nodes serving as targets for new processes will be idle at least as long as the delay involved in learning about their statuses and moving processes to them—otherwise it has no reason to expect that they will still be idle when the processes arrive. But of course in practice, it must be more conservative than that, in order to justify the overhead incurred by moving processes from one domain to a distant other domain. Schedulers at increasingly higher, more global levels must have increasingly large estimates for likely node idle times in hand in order to justify moving processes.

In sum, the Trellis's inherently hierarchical structure provides what seems like exactly the right framework for dealing with increasingly global Piranha scheduling decisions.

Research Results / AFOSR Grant # F49620-92-J-0240

1) Adaptive Parallelism and Piranha

Adaptive parallelism refers to parallel computations on a dynamically changing set of processors: processors may join or withdraw from the computation as it proceeds. Networks of fast workstations are the most important setting for adaptive parallelism at present. Workstations at most sites are typically idle for significant fractions of the day, and those idle cycles may constitute in the aggregate a powerful computing resource. For this reason and others, we believe that adaptive parallelism is assured of playing an increasingly prominent role in parallel applications development over the next decade.

Under adaptive parallelism, the set of processors executing a parallel program may grow or shrink as the program runs. Potential gains include the capacity to run a parallel program on the idle workstations in a conventional LAN—processors join the computation when they become idle, and withdraw when their owners need them—and to manage the nodes of a dedicated multiprocessor efficiently. Experience to date with our Piranha system for adaptive parallelism suggests that these possibilities can be achieved in practice on real applications at comparatively modest costs.

2) Network Piranha and Multiprocessor (CM-5) Piranha

Piranha is an adaptive version of master-worker parallelism. Programmers specify in effect a single general purpose (but application specific) *worker function* called piranha(). They do not create processes and their applications do not rely on any particular number of active processes. When a processor becomes available, a new process executing the piranha() function is created there; when a processor withdraws, a special retreat() function (which is also provided by the application programmer) is invoked, and then the local piranha process is destroyed. Thus, there are no "create process" operations in the user's program, and the number of participating *processes* (and not merely processors) varies dynamically.

2

The Multiprocessor (CM-5) Piranha research explores Piranha on the Connection Machine DM-5 multiprocessor. Our preliminary results suggest that adaptive parallelism provides not only substantial computing power for parallel programs but also an attractive alternative to traditional methods for sharing multiprocessors among users.

3) **"Lifestreams": A Piranha application being developed from research under AFOSR Grant # F49620-92-J-0240**

Today, just managing one's own electronic world can be a frustrating task for most computer users, requiring too many separate applications, too many file transfers and format translations, and the construction of organizational hierarchies that too quickly become obsolete. What is needed is a metaphor and system for organizing the electronic "bits of paper" we all so easily collect, whether they come to us in the form of electronic mail, downloaded images, pages gathered from the Web, or scheduling reminders.

Lifestreams is such a system. It uses a simple organizational metaphor, a time-ordered stream of documents, to replace conventional files and directories. *Stream filters* and software agents are used to organize, locate, summarize and monitor incoming information. Lifestreams subsumes many separate desktop applications to accomplish the most common communication, scheduling, and search and retrieval tasks; yet its machine-independent, client-server architecture is open so that users can continue to use the document types and viewers/editors they are accustomed to. Currently we are running a research prototype of the system in house, and hope eventually to make the software publically available.

Publications

- "Adaptive Parallelism and Piranha," Nicholas Carriero, Eric Freeman, David Gelernter, and David Kaminsky. *Computer*, January 1995.

- "Adaptive Parallelism on Multiprocessors: Preliminary Experience with Piranha on the CM-5," Nicholas Carriero, Eric Freeman, and David Gelernter. *Languages and Compilers for Parallel Computing*, Lecture Notes in Computer Science # 768,Springer-Verlag, 1994.

- "Piranha Scheduling: Strategies and their Implementation," YALEU/DCS/RR-983, September 1993, revised November 1994.

- "Finding and Reminding Reconsidered," Scott Fertig, Eric Freeman and David Gelernter. YALEU/DCS/RR-1090. Submitted to SIGCHI Bulletin, October 1995.

- "Lifestreams: Organizing Your Electronic Life," YALEU/DCS/RR-1083, AAAI Fall Symposiums: AI Applications in Knowledge Navigation and Retrieval, Cambridge, MA, November 1995.

3

- "Lifestreams for the Newton," Eric Freeman. PDA Developer, vol 3.4, July/Aug 1995.

- "The Lifestreams Approach to Reorganizing the Information World," YALEU/DCS/RR-1070, April 1995. Submitted to Technology Review.

4