REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
November 1990

3. REPORT TYPE AND DATES COVERED
final report 30Aug88-28Feb90

4. TITLE AND SUBTITLE
QLISP for Parallel Processors

5. FUNDING NUMBERS
C: N00039-84-C-0211
T: 19, 23

6. AUTHOR(S)
John McCarthy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Science Department
Stanford University
Stanford, CA 94305

8. PERFORMING ORGANIZATION REPORT NUMBER

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

sponsoring agency:
SPAWAR 3241C2
Space & Naval Warfare Systems Command
Washington, D.C. 20363-5100

monitoring agency:
ONR Resident Representative
Mr. Paul Biddle
Stanford Univ., 202 McCullough
Stanford, CA 94305

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

See attached report.

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
UL	UL	UL	UL

Sponsored by

Defense Advanced Research Projects Agency (DoD)
1400 Wilson Boulevard
Arlington, Virginia 22209-2389

"QLISP for Parallel Processors"

ARPA Order No. 5826

Issued by Space and Naval Warfare Systems Command

Under Contract No. N00039-84-C-0211, Tasks 19 and 23

Qlisp Final Report

Stanford University has completed the initial phase of the Qlisp project for research on parallel Lisp programming. The research made use of the Qlisp implementation on the Alliant FX/8 delivered by Lucid, Inc. under subcontract to Stanford University. The results we have obtained confirm, in general, our predictions that a shared-memory multiprocessor is an effective tool for executing symbolic programs in Lisp.

The major highlights of Qlisp have been:

- Demonstrating that Qlisp could be implemented as originally specified, and incorporating additional features that we found to be necessary or desirable.
- Discovering the importance of dynamic control of parallelism.
- Implementing the basic parts of several major Lisp applications, and finding that Qlisp was a good vehicle for expressing the parallelism in these programs.

The implementation of Qlisp proceeded with few difficulties. Several major enhancements to Lucid Common Lisp, that had been envisioned as necessary, were performed. These include deep binding (for special variables); support for multiple control stacks; and the removal or synchronization of many parts of the system that otherwise would not run correctly in parallel.

The control of parallelism at runtime was known from the start to be an important part of programming in Qlisp. Our belief in the best way to accomplish this, however, has changed. Experiments with control by cutting off parallelism based on the height or depth of a program's computation tree showed that this method was hard to deal with and not as flexible as we had hoped. Cutting off parallelism based on the runtime state of the machine, however, performed better than we expected and is now our preferred method. Along the way we discovered what kinds of programs benefit the most from this method (those whose computation trees are fairly balanced), and found theoretical justification for the efficacy of dynamic control, which had not been done before.

The applications with which Qlisp has been tested include small and medium-sized programs in the areas of:

- polynomial arithmetic and symbolic computation
- theorem proving
- functional programming and semantics
- production systems
- backtracking search

Several members of the Qlisp project attended the US/Japan Parallel Lisp Workshop in June 1989. Dan Pehousheck attended the Oregon Workshop on Parallel Lisp held in June 1990. Lucid gave Qlisp presentations to the Computer Science Department and Coordinated Science Lab of the University of Illinois, Los Almos National Labs and Sandia Labs.

The work on Qlisp is reported in publications and reports listed at the end of this report. Qlisp has been a powerful tool, and fairly easy to use, in most of these areas. Our best results have been in "and-parallel" computations, as found in polynomial manipulations, functional programs, and many parts of theorem proving. Potential available parallelism is easy to achieve in the context of only 8 processors. The next phase of research should include both more substantial applications and experiments with systems of 20-30 processors, to verify that the initial results scale. In addition, we still need more experience, and Qlisp needs more tuning, to handle well the "or-parallel" computations found in many search algorithms.

Qlisp users

The following is a list of people who have used Qlisp.

Qlisp people at Stanford:

Ian Mason
John McCarthy
Dan Pehoushek
Igor Rivin
Kelly Roach
Dan Scales
Carolyn Talcott
Joe Weening

Qlisp people at Lucid:

Dick Gabriel
Ron Goldman
Richard Mlynarik
Peter Benson
Carol Sexton
Harlan Sexton
Claire Wolfe

Other Stanford researchers and visitors:

Charles Buckley — computer algebra and geometric reasoning
Anoop Gupta — production systems
Hiroshi Okuno — production systems

Berkeley researchers:

Jim Larus — parallel compilation techniques
Carl Ponder — computer algebra

Qlisp Publications

[1] Richard P. Gabriel, Ron Goldman, and Carol Sexton. Qlisp manual, 1990.

[2] Richard P. Gabriel and John McCarthy. Queue-based multiprocessing Lisp. Technical Report STAN-CS-84-1007, Stanford University, Stanford, California, June 1984.

[3] Richard P. Gabriel and John McCarthy. Queue-based multiprocessing Lisp. In *Conference Record of the 1984 ACM Symposium on Lisp and Functional Programming*, pages 25–44, Austin, Texas, August 1984.

[4] Richard P. Gabriel and John McCarthy. Qlisp. In Janusz S. Kowalik, editor, *Parallel Computation and Computers for Artificial Intelligence*, chapter 4, pages 63–89. Kluwer Academic Publishers, Boston, Massachusetts, 1988.

[5] Ron Goldman and Richard P. Gabriel. Preliminary results with the initial implementation of Qlisp. In *Conference Record of the 1988 ACM Symposium on Lisp and Functional Programming*, pages 143–152, July 1988.

[6] Ron Goldman and Richard P. Gabriel. Qlisp: Experience and new directions. In *Proceedings of ACM/SIGPLAN Parallel Programming Experience, Applications, Languages, and Systems*, July 1988.

[7] Ron Goldman and Richard P. Gabriel. Qlisp: Parallel processing in Lisp. In *Proceedings of HICSS-22, Hawaii International Conference on System Sciences*, January 1989.

[8] Ron Goldman, Richard P. Gabriel, and Carol Sexton. Qlisp: Parallel processing in Lisp. In *Proceedings of the US/Japan Workshop on Parallel Lisp*, June 1989. To be published by Springer-Verlag, 1990.

[9] Ian A. Mason, Joseph D. Pehoushek, Carolyn L. Talcott, and Joseph S. Weening. Parallel list mapping operations, 1990.

[10] Ian A. Mason, Joseph D. Pehoushek, Carolyn L. Talcott, and Joseph S. Weening. Programming in Qlisp. Technical report, Stanford University, 1990.

[11] H. G. Okuno and A. Gupta. Parallel execution of OPS5 in Qlisp. Technical Report STAN-KSL-87-43, Stanford University, Knowledge Systems Laboratory, 1987.

[12] Joseph D. Pehoushek and Joseph S. Weening. An analysis of dynamic partitioning. In *Proceedings of the US/Japan Workshop on Parallel Lisp*, June 1989. To be published by Springer-Verlag, 1990.

[13] Joseph D. Pehoushek and Joseph S. Weening. A process spawning strategy for parallel Lisp, 1989.

[14] Arkady Rabinov and Igor Rivin. Programming in Qlisp: A case study. Technical Report STAN-CS-89-1247, Stanford University, February 1989.

[15] Daniel Scales. Parallelizing the OPS5 matching algorithm in Qlisp. Technical report, Stanford University, 1990. in preparation.

[16] Joseph S. Weening. A parallel Lisp simulator. Technical Report STAN-CS-88-1206, Stanford University, Stanford, California, May 1988.

[17] Joseph S. Weening. *Parallel Execution of Lisp Programs.* PhD thesis, Stanford University, Stanford, California, June 1989.

[18] Joseph S. Weening. Qlisp manual, 1989.