Quarterly R&D Status Report No. 12
Common Runtime Support for High Performance Languages

Period Covered
From: 1 April 1997
To: 30 June 1997

July 10, 1997

Prepared By

Geoffrey C. Fox
Northeast Parallel Architectures Center
at Syracuse University
111 College Place
Syracuse, NY 13244-4100

Prepared For

ESC/ENS
5 Eglin Street, Building 1704
Hanscom AFB, MA 01731-2116
Attn.: Mr. Harry Koch

Sponsored By

Advanced Research Projects Office/CSTO
Common Runtime Support for High Performance Parallel Languages
ARPA Order No. B526
Issued by ESC/ENS under contract
F19628-94-C-0057

Program Technical Status

A. Description of progress during reporting period, supported by reasons for any changes reported previously

Following what was described in our last quarterly report, identified work items have been carried out smoothly in this reporting period. In particular,

- Reflecting the cut in budget, we have discussed the remaining issues in the project with all other subcontractors, and evolved the draft revised SOW and work plan as appended in the last quarterly report to the revised SOW and work plan appended in this report. This appendix will be our basic guidance for how the mission of the project to be completed.
- The PCRC runtime kernel (to be released in August, 1997) has been completed with documentation.
- We have completed the ghost region support (the first item in the work plan) in both PCRC runtime and related HPF compiler. This accomplishment enable our compiler to produce higher performance executable code than commercial HPF compiler (pghpf 2.0).
- The Java unparser of our front-end system is finished.
- We have partially documented what we have been doing and reorganized our web site as in http://www.npac.syr.edu/projects/pcrc/demo1997. This site provides a clear picture on various activities under the project.

B. Description of any special programs developed or major items of experiments or special equipment purchased or constructed during reporting period

Along with various Java activities connected with this project, a prototype Fortran-to-Java converter has been developed, and downloadable from http://www.npac.syr.edu/projects/pcrc/f2j.html. This converter is built from our front-end system, thus also serves as a concrete example for the usefulness of the infrastructure.

C. Notifications of any changes in key personnel associated with the contract during the reporting period

No change during this reporting period.

D. Summary of substantive information derived from noteworthy trips, meetings, and special conferences held in connection with the contract during the reporting period

- Promoted by PCRC, the ACM Workshop on Java for Science and Engineering Computation was held in Las Vegas, NE, June 21, 1997, as one of distinct components of ACM SIGPLAN annual activities. 105 people from academia, industry, and national labs participated.
- Poster presentation of "PCRC Activities in NPAC" in the CRPC annual meeting held in Rice University, May 20-22.
- Presentation of "PCRC Activities" in the DARPA System Environments PI meeting held in June 8-10.

E. Summary of any problems or areas of concern on which procuring activity or guidance is required

We provide an appendix for revised SOW and work plan, as a result of discussion among subcontractor PIs and our intention for a no-cost-extension of this project as requested previously.

F. Statement relative to the apparent sufficiency of contractually prescribed effort to achieve the objectives of the contract

The contractually prescribed effort is sufficient to achieve the objectives of this contract.

G. Related accomplishments: i.e., scientific papers published, etc.

A special issue of *Concurrency — Practice and experience* is published in June, 1997, which collects all the papers presented in the Java for Scientific and Engineering Computation Workshop held last December in Syracuse.

The following is a partial list of related reports/papers completed during this reporting period.

1. Bryan Carpenter, Geoffrey Fox, Xiaoming Li, et al, "PCRC-based HPF Compilation," has been accepted by LCPC'97.

2. Bryan Carpenter, Yuh-Jye Chang, Geoffrey Fox, et al, "Java as a Language for Scientific Parallel Programming," has been accepted by LCPC'97.

3. Geoffrey Fox, Xiaoming Li, Zheng Qiang, et al, "A Prototype of Fortran-to-Java Converter" was presented in ACM Workshop on Java for Science and Engineering Computation on June 21, 1997.

4. Dennis Gannon, "Programming Environments, Compiler Technology and Runtime Systems for Object Oriented Parallel Processing," DARPA PI meeting, June 9, 1997.

5. Jim Browne, "An Environment for End-to-End Performance Design of Large Scale Parallel Adaptive Computer/Communications Systems," DARPA PI meeting, June 9, 1997.

6. Joel Saltz, "Performance Prediction Using Application Emulators," DARPA PI meeting, June 9, 1997.

H. Planned activities and milestones for the next reporting period

Besides the overall work plan as described in the appendix, the most significant work for the next reporting period will be the release of PCRC runtime kernel to the public domain. For that to be accomplished, the following items have to be conducted.

- Documentation — the code release will be accompanied by proper document.
- Installation procedure — portability issue will be addressed here.
- Fortran interface. The kernel itself is in C++. A Fortran interface will be provided as an exampler for how other languages can access this runtime.
- Testing. A set of test programs will be included in the release for both functionality and performance of the runtime kernel.

I. List of all equipment purchased during reporting period

None.

Program Financial Status

The table below provides information required by contract.

WORK BREAKDOWN STRUCTURE	CUMULATIVE TO DATE			AT COMPLETION		
ELEMENT	Planned Expended	Actual Expended	Percent Completed	BAC	LRE	Remarks
CLIN 0001						
Syracuse University	3,218,954	2,882,344				
& subcontractors						
TOTAL	3,218,954	2,882,344	89.50%			

A. Is current funding sufficient for the current FY ? Yes.

B. What is the next Fiscal Year's funding requirement at current anticipated levels ? No new funding required to complete remaining work as described in this report by June 30, 1998.

C. Have you included in the report any explanation of the above data and are they cross-referenced ? Yes.

Distribution:

ARPA/CSTO (2 org)
3701 N. Fairfax Drive
Arlington, VA 22203-1714

ESC/ENK (ltr only)
5 Eglin Street
Hanscom AFB, MA 01731-2116
Attn.: Mr. Dale V. Hawley

Appendix: Revised SOW and Work Plan

As mentioned in the QR for the first quarter of 1997, we are requesting a no cost extension of this project until June 30, 1998. We divide the period from April 1, 1997 to June 30, 1998 in two stages. Stage 1 is from April 1, 1997 to August 31, 1997. Stage 2 is from September 1, 1997 to June 30, 1998. In this appendix, a revised statement of work is provided corresponding to the new situation, then a detailed work plan is described.

Revised statement of work

We will continue the activities towards the deliverables stated in project proposal. In particular, besides various research activities surrounding the central theme of PCRC, the following items are identified as distinct software packages deliverable to the public domain.

- PCRC runtime system. A partial release in the summer of 1997, and a full release at the new requested end date.
- An experimental HPF compiler that uses the PCRC runtime.
- A compiler/interpreter research and development infrastructure, including multiple front-ends (HPF, C++, and Java) and tools.

The following institutions will be involved in the project for the remaining period.

- Syracuse University — lead, responsible for majority of the implementation work.
- Cooperating Systems — help in software quality assurance and documentation.
- University of Maryland — responsible for interoperability among different runtimes.
- University of Rochester — help in the feasibility study of a Java compiler using the PCRC runtime, and partial implementation.
- University of Florida — implement novel combining-scatter functions in PCRC runtime, studies of new algorithms for PCRC functions.
- Indiana University — help in the feasibility study of pC++ and Java compilers using the PCRC runtime, and implementation of pC++.
- Texas — DAGH as part of PCRC runtime.
- Rice — modest involvement, especially with Java compiler requirements.

A two-stage work plan including the No-Cost Extension period

Stage 1 (April 1, 1997 to August 31, 1997): aiming at accomplishing basic PCRC requirement and having a preliminary PCRC runtime release to the public domain.

We identify the following working items.

1. A quality public domain runtime (kernel) in C++.

   This runtime will be considered as PCRC proper. It will be based on and extended from Adlib. It will be rich in contents, fulfilling most of the claims in PCRC project proposal, in particular, task 1, 2, 3, 4.

   To have a quality public release, the following work has to be completed.

   a. sorting functions
   b. new combining scatter functions compatible with PCRC runtime.
   c. Adlib documentation
   d. PCRC documentation
   e. C++ interface
   f. Fortran interface
   g. Java interface
   h. experiment on interoperability between Syracuse runtime and Maryland runtime.

   Syracuse will be the main force in the implementation work. Cooperating Systems will help in testing and documentation. Florida will be responsible for item `b`. Maryland will implement item `h`. Rochester is involved in `g`; Rice in `f` and `g`. Moreover, Texas will provide a preliminary release of DAGH, which is an effort to produce an application specific parallel library. We will, in stage 2, investigate how this application specific library can work in concert with a general library such as the PCRC kernel.

2. An operational compiler for a subset of HPF using the PCRC runtime.

   The motivation for this distinct working item may be seen as the following

   - It is required (task 8 and 9) in the original project contract that compilers are to be developed to demonstrate the usability of the runtime.

     Thus, this compiler will serve as a primary exampler for this purpose.

- It is our hope to provide a 'playable' data parallel language system for distributed memory machines in public domain. By 'playable', we mean something other people can use to do various experiments in the context of a complete system. It has been observed that a few high quality public domain parallel language system exist for shared memory machines (such as SUIF). These systems have been helping research community and industry to a great extent. But non for distributed memory machines exists (some announced, such as shpf, are not really playable due to poor implementation). We are making effort to fill this gap. The compiler is one of the three components in such a system. (the other two components are runtime and an infrastructure)

  Technically, capability of the compiler will be "FORTRAN 77 + array syntax + FORALL + directives". Emphasis will be on quality implementation, so that not only some real application can be compiled through this compiler, but also allow the compiler itself to be modified and extended by others.

To achieve this goal, the following work has to be done in the stage 1:

a. documentation

b. demos and tests

c. full FORALL statement

d. procedure interface

e. front-end, testsuite and tests

Syracuse will be the main force in the implementation work. Cooperating Systems will help in testing and documentation.

3. Preparation for stage 2.

Successful completion of the above two items fulfills many of the basic PCRC requirements. We will investigate the applicability of PCRC runtime, as well as further improve its quality. We do this in the context of constructing a quality compiler/interpreter development infrastructure. In particular, we will build a system which incorporate HPF/Java/C++ front-ends and related tools, together with some studies/experiments on compilation of Java/C++ based on PCRC runtime. Robustness of the system will be emphasized.

To achieve this goal, the following work has to be done:

a. Java compilation study

b. Integration of pC++ (and C++) front-end with HPF/Java front-end
c. pC++ compilation study
d. Xsageop documentation
e. System documentation
f. Benchmark of PCRC and related runtimes
g. Comparison of COCOLIB and metachaos
h. demos and tests

Syracuse will be the main force in the implementation work. Cooperating Systems will help in testing and documentation. Indiana will help for item `b` and `c`. Florida will help in benchmarking collective data movement routines.

Stage 2 (September 1, 1997 to June 30, 1998) : aiming to fulfill all PCRC requirement. The following items are identified.

1. Enhancement of PCRC runtime.

   a. a fully operational connection between NPAC library and Maryland library via meta-chaos or other means
   b. provide support for irregularly distributed arrays in PCRC runtime (HPF-2-style irregular block distribution formats and/or Adlib-style "non-Cartesian arrays").
   c. additional runtime routines
   d. a full release of DAGH
   e. more benchmarks and tests
   f. more documents

   Syracuse will be the main force in the implementation work. Cooperating Systems will help in testing and documentation. Maryland will help for item `b`. Florida will help for item `c`, and Texas for `d`.

2. A quality public domain compiler/interpreter construction infrastructure.

   a. further polish our multi front-end (HPF, pC++, Java) system
   b. a selected set of tools built around our multi front-end, including WebFlow and web-based program steering and debugging utilities.

Syracuse will be the main force in the implementation work. Cooperating Systems will help in testing and documentation.

3. A releasable HPF compiler.

   a. support for general FORALL constructs
   b. support for INDEPENDENT directive
   c. some support for HPF 2.0 features
   d. test and demo suite with real applications.

   Syracuse will be the main force in the implementation work. Cooperating Systems will help in testing and documentation.

4. An infrastructure supporting high level SPMD programming practice.

   This is to be a valuable by-product of this project that we have been seeing its emergence from our work, namely our runtime may also be effectively used as an infrastructure to support SPMD programming. In particular, application programmers will not have to use `send()` and `receive()` in their code. Instead, they use high level facilities provided by the runtime. We intend to use Java as a vehicle for demonstrating this SPMD programming model. More specifically, user will be able to write Java SPMD programs that call PCRC runtime functions as native methods.

   Particular work needed for this product includes optimization, extensive testing on both functionality and portability, and documentation. Syracuse will be mainly responsible for it.