NPAC Technical Report SCCS 641

An Applications Perspective on High Performance Fortran

G.C.Fox, K.A.Hawick **Northeast Parallel Architectures Center**,
Syracuse, NY 13244-4100

August 1994

Abstract

We describe our planned work to investigate the applicability of High Performance Fortran in real applications. As a working 'networked notebook' we are assembling a package on the World Wide Web to collect codes, algorithms, information and commentary on the High Performance Fortran language and data parallel algorithms that allow its efficient use.

Contents

1 Introduction

High Performance Fortran (HPF) was designed from parallel application experience with both message passing application codes and use of early data parallel languages such as CMFortran, DAP and Maspar Fortran, FortranD and Vienna Fortran. The HPF Forum refined and improved this definition. It was understood that the first HPF specification was incomplete and the current HPF2 forum is now studying areas where the language should be changed. One important set of changes will allow the language to express a broader set of applications.

We believe that the current version of HPF is capable of expressing efficiently and elegantly some reasonable fraction of scientific and engineering simulations - perhaps around 50estimate is inherently inaccurate - as one will obtain different answers depending on the metric used. For instance, using a count of codes will give a different answer from counting machine time used. Quantum Chromodynamics(QCD) simulations are certainly suitable for the current HPF and represent one of the largest single use of supercomputer cycles. Typically, QCD codes do not exercise all the features of a programming language as they are usually rather easy to implement for parallel systems.

Further new codes are using improved algorithms with more irregular adaptive features (such as adaptive unstructured grids for partial differential equations). These new algorithms are harder to express in the current HPF than the older approaches which tend to be static and regular. The research community (especially the work of Saltz at Maryland and his collaborators including Rice and Syracuse) has shown how to improve the HPF language so that it can express a broad set of irregular adaptive problems. We believe that over 90engineering computations can be expressed by these improved constructs developed in research compilers. Other plans for a task parallel mechanism to be part of HPF2, following the work of Foster and others on Fortran-M, is likely to enable nearly all scientific and engineering work currently done in Fortran 77 to benefit and be implementable in some way on HPCC systems.

However this is rather wishful thinking as currently (August 1994) we do not have mature commercial compilers and we have therefore no HPF experience from a **real** user base. Thus we have decided that now is a good time to evaluate the actual language we have in the HPF definition and further the possible improvements for HPF2. This therefore revisits some of the work in the original design but in a more focused and applied fashion.

It has become clear recently that despite the introduction of Fortran 90 and High Performance Fortran and the interest generated by the related compiler projects, very little applications level code has been written in those languages. It is not surprising that this is the case, since stable compilation systems have only begun to appear. However, NPAC is in the fortuitous position of having access to near production level compilation systems, as well as a body of experience that has resulted from using the pre-HPF HPCC systems such as the Thinking Machines and MasPar systems.

It is proposed that a meta-project called **HPFA** (High Performance Fortran Applications) for want of a better acronym, be initiated to overlap with existing funded projects and thus cross co-ordinate the development of HPF application codes and experience.

Our objectives for the immediate future are:

1. to coordinate the large range of HPF applications efforts in our own organisation (NPAC);
2. to ensure HPF applications work at other institutions is integrated (especially CRPC).
3. give us a set of applications which include particular features of the language and so can be used to test the performance of compilers on these language features;
4. to give the user community a set of exemplars which will help them learn to use HPF effectively.
5. to arrive at some quantitative evaluation of F90/HPF principally for expressiveness and general appropriateness for building high performance application codes. This only **indirectly** involves evaluating the specific performance of existing implementations.
6. to help clarify omissions in the HPF language;

We believe that to explore the Fortran 90 and HPF languages it is necessary to actually use them at an **applications** level rather than as an interesting curiosity of computer language science. There are essentially three levels that we need to address: language familiarity; simple algorithm expression; full application potential (or compound algorithm expression).

Language familiarity is essentially the ability to write programs without the constant use of a manual. Programmers nearly always quickly learn the main concepts of a language by studying existing source code examples. Unfortunately there is very little F90/HPF source code in existence yet. Furthermore, the language text books currently available are also short on examples.

Evaluating the languages for their expressive capabilities requires trying to solve particular application problems, and there are two levels to this. Simple algorithms like: solving a partial differential equation on a grid; n-body particle dynamics; cluster labeling; and Fourier transforms etc can each be expressed in a relatively short code and can make encapsulated **kernel** codes. Generally these kernels will have one or at most two principle data distributions corresponding to the one or two core algorithms involved. Full applications however, may have several possibly incompatible data distributions required in the same code, each called for by the various sub-algorithms.

2 Proposed Plan

To enable the HPFA meta-project, a Technology Integration Server (TIS) is under construction. This will be a Web/Mosaic oriented repository for codes and documents (conventional and hypertextual) related to this effort. The TIS will integrate Fortran 77, Fortran 90, HPF and HPF-subset languages through online language user guides, summaries and glossaries. The applications codes will be divided into the three main sections:

1. Language feature **examples** - at the level of a worked example of each of the the mainstream features of the languages to illustrate their use. Typically less than 100 lines of source and written specially for the TIS.
2. **Kernel** Applications - at the level of relatively short kernel applications programs that are complete worked demonstrations of how F90/HPF can be used to solve a generic type of problem. Typically less than 1000 lines of source, and thus short enough to be frozen and incorporated directly into the TIS. The NAS benchmark kernels are examples of this.
3. Full **Applications** Level collection - whereby we identify a set of small to medium sized applications that can be expressed in F90/HPF. These may be many thousands of lines of source and will be associated with other ongoing projects eg NASA, MADIC,...

3 HPF Applications Web Server Package

The features of the NPAC HPFA (High Performance Fortran Application) World Wide Web resource are:

- a list of Available Compilers;
- a list of Industrial and Academic application areas with an indication of appropriate software including suitability of High Performance Fortran.
- banana list of generic (benchmark) applications with discussion of issues of relevance to HPF and HPF+. This list also points to exemplar codes which are not complete but include implementations in F77, F90, HPF and message passing. We also point to relevant large scale codes at NPAC and elsewhere
- a list of papers and books on HPF, Fortran90 and associated parallel computing issues;
- an on-line HPF Tutorial;
- answers to commonly asked questions on HPF;
- talks and lectures on HPF.

The following Uniform Resource Locators (URL) provide Hypertext transfer protocol (HTTP) links to information assets relevant to this project:

http://www.npac.syr.edu/hpfa/ The HPF Applications World Wide Web package.

http://www.npac.syr.edu/ The Northeast Parallel Architectures Center (NPAC) Main Server.

http://www.netlib.org/nse/home.html The National HPCC Software Exchange.

4 Acknowledgments

It is a pleasure to thank Bhaven Avalani, Ed Bogucz, Alok Choudhary, Paul Coddington, Ammar Degani, Kivanc Dincer, Tom Haupt, Miloje Makivic, Nancy McCracken, Sanjay Ranka, Guy Robinson and Hon Yau for their ongoing contributions towards this project.

References

[1] Adams, J., Brainerd, W., Martin, J., Smith, B., and Wagener, J., *Fortran 90 Handbook: Complete ANSI/ISO Reference*, McGraw-Hill, 1991.

[2] High Performance Fortran Forum, "High-performance Fortran language specification," *Scientific Programming*, vol.2 no.1, July 1993. (Also available by anonymous ftp from ftp.npac.syr.edu (/HPFF).)

[3] Hiranandani, S., Kennedy, K., and Tseng, C.W., "Compiler support for machine-independent parallel programming in FortranD," in *Compiler and Runtime Software for Scalable Multiprocessors*, 1991.

[4] Hiranandani, S., Kennedy, K., and Tseng, C.W., "Preliminary experiences with the FortranD compiler," *Proceedings of Supercomputing '93,* Portland, OR, 1993, p. 338.

[5] Fox, G.C., Hiranadani, S., Kennedy, K., Koelbel, C., Kremmer, U., Tseng, C.W., and Wu, M., "Fortran D language specification," Technical Report, Rice and Syracuse Universities, 1992.

[6] Bozkus, Z., Choudhary, A., Fox, G., Haupt, T., and Ranka, S., "Fortran 90D/HPF compiler for distributed-memory MIMD computers: design, implementation, and performance results," *Proceedings of Supercomputing '93*, Portland, OR, 1993, p.351.

[7] Chapman, B., Mehrotra, P., Mortisch, H., and Zima, H., "Dynamic data distributions in Vienna Fortran," *Proceedings of Supercomputing '93*, Portland, OR, 1993, p.284.

[8] PARKBENCH Committee, "Public international benchmarks for parallel computers," to submitted to *Scientific Programming*, 1994. (Also available by anonymous ftp from ecs.soton.ac.uk (pub/benchmarks).)

[9] Bailey, D., Barton, J., Lasinski, T. and Simon, H., Editors, "The NAS Parallel Benchmarks", NASA Ames, NASA Technical Memorandum 103863, July 1993.

[10] Koelbel, C.H., Loveman, D.B., Schreiber, R.S., Steele, G.L., Zosel, M.E., "The High Performance Fortran Handbook", MIT Press 1994.

[11] Bogucz, E.A., Fox, G.C., Haupt, T., Hawick, K.A., Ranka, S. "Preliminary Evaluation of High Performance Fortran as a Language for Computational Fluid Dynamics", 25 AIAA Fluid Dynamics Conference, June 20-23, 1994, Colorado Springs.

[12] Hawick. K.A., Fox, G.C., "Exploiting High Performance Fortran for Computational Fluid Dynamics", Proc. "High Performance Computing and Networks ", Int. Conf., Milan, 1995.

[13] Hawick, K.A., Dincer, K., Robinson, G., Fox, G.C., "Conjugate Gradient Algorithms in Fortran 90 and High Performance Fortran" NPAC Technical Report SCCS 691, February 1995.

[14] Robinson, G., Hawick, K.A., Fox, G.C., "Fortran 90 and High Performance Fortran for Dense Matrix-Formulated Applications" NPAC Technical Report SCCS-709, May 1995.

[15] Hawick, K.A., et al, "High Performance Fortran for Dense Linear Algebra", NPAC Technical Report. Work in progress, 1995.

[16] Yau, H.W., Robinson, G., Hawick, K.A., et al, "High Performance Fortran for Spectral Transform Methods", Work in progress, 1995.