Development of a Public Domain *High Performance Fortran* Translation System

1 Summary

High Performance Fortran (HPF) is a newly defined programming language, designed for development of "scientific" programs to run on distributed memory supercomputers. It consists of a set of extensions to Fortran 90. We propose to develop a portable HPF translation system which would be made freely available to higher education establishments.

Research on HPF and precursor systems has been pursued for several years in the University's Department of Electronics and Computer Science (ECS). Building upon expertise and existing software from ECS we can credibly produce a prototype "Subset HPF" system—to a standard usable within HEIs—in the academic year 93/94. Follow-on funding would allow the prototype to be developed to near product standard, and thereafter improved and maintained.

The outputs of the project will consist of a translator from HPF to Fortran 90, a run-time library which encapsulates all platform dependencies of the translation system, and associated documentation.

The resources required for the project are funds to employ one experienced programmer (recruited from ECS).

2 Background

2.1 Distributed Memory Supercomputers

During the past few years advocates of distributed memory supercomputers have been able to demonstrate impressive cost and performance figures for their machines, across a broad range of applications. But these machines have so far failed to make an impact on mainstream scientific computing commensurate with these successes. The reason, of course, is the large investment required to develop quality software for specialised architectures. HPF is a promising new approach to tackling this problem—its claim is that it will enable efficient *portable* (architecture-independent) parallel programming.

2.2 *High Performance Fortran*

In the last year the High Performance Fortran Forum (HPFF), a coalition of many leading industrial and academic groups in the field of parallel processing, has established a language standard called High Performance Fortran. This consists of a set of extensions to Fortran 90, facilitating data-parallel computations on multi-processor architectures. In particular, HPF includes directives

for specifying (at various levels of abstraction) how program data should be distributed over the memories of virtual or physical processors.

Apart from the data-mapping directives, an HPF program looks essentially like a normal Fortran program. HPF is *not* an explicit message-passing model. All communications are handled transparently by the compilation system. Thus an HPF program can be written without detailed knowledge of the target architecture or low-level communication primitives—the burdens which currently discourage programmers of distributed memory (or even shared-memory MIMD) computers.

The standard is supported by a majority of vendors of parallel hardware, including Cray, DEC, Fujitsu, HP, IBM, Intel, Maspar, Meiko, nCube, Sun and Thinking Machines. Several vendors have already announced their intention to develop HPF implementations, including hardware vendors DEC, Intel, Maspar, Meiko, Thinking Machines, and compiler vendors ACE, APR, KAI, Lahey, NA Software, Portland and PSR.

2.3 The Southampton connection

Since 1989, the Concurrent Computation Group (CCG) at Southampton has been involved in research and development on a system called ADAPT ("Array Distribution Automatic Parallelisation Tool") for automatically transforming Fortran 90, supplemented by data distribution declarations, to code for distributed memory message-passing architectures. The input language for the ADAPT system, Distributed Fortran 90, may be regarded (alongside Fortran D and Vienna Fortran) as a precursor of HPF.

Southampton had a significant influence on the development of HPF, through an active involvement in the HPFF process. Southampton's contributions include the `PURE` procedure mechanism, which relaxes some of the restrictions of an otherwise predominantly SIMD programming model. The CCG can therefore claim an established track-record in the field of HPF-like languages and their implementation.

In the last year researchers in the CCG have been working on the design of an HPF translation system. The translator would be based on existing ADAPT software, extended to handle HPF syntax, and the more general distributed data model of HPF. This is the system we propose to develop with the requested funding.

3 Objectives

The HPF Forum have defined an official subset of HPF to facilitate early implementations. It is expected that the first true "Subset HPF" compilers, implemented as translators to standard Fortran plus message-passing, will appear

at the end of 1993 or start of 1994. We propose to develop a public domain version of such a system.

Our prototype translation system will contain two components. A translator, developed from the ADAPT software, will take a program written in HPF and output a Fortran 90 program containing calls to a run-time library called ADLIB. ADLIB is a high-level library for managing distributed data arrays. It will contain, for example, routines for creating distributed arrays, routines for "remapping" such arrays (ie, changing their distribution format), and generalised "gather" and "scatter" operations on such data. All dependence on the underlying message-passing platform will be hidden in the implementation of ADLIB.

This software architecture provides for maximal portability. Because the target language is a standard high-level language, no changes to the translator are necessary in passing between hardware or software platforms. Only the run-time library, ADLIB, has to be changed. Even if the target machine is changed from (say) a distributed to a shared memory machine, the translator need not be modified.

Evidently there is some performance penalty in this modular design. "Production" HPF systems may need closer tailoring of the compiler to the target hardware. For this reason we describe the proposed system as a "prototype". It will allow users to gain experience in parallel programming in HPF with modest (perhaps negative) improvement in performance, on their existing platforms. If the HPF idea succeeds, later software and hardware developments should allow HPF programs to compete with "hand-crafted" message-passing programs in performance.

3.1 Fortran 90

Fortran 90 has been chosen as the target language for our translator. Other obvious choices would be Fortran 77 or C. Fortran has some clear advantages over C, since the source language is also a dialect of Fortran. ADAPT originally targetted Fortran 77, but complications arise, for example due to the lack of dynamic memory allocation. Fortran 90 should prove a more flexible target language. The data-abstraction features of Fortran 90 will facilitate a clean implementation of distributed arrays. There is every reason to expect that within the time scale of this project Fortran 90 will become widely available in academic institutions.

3.2 ADLIB

The initial implementations of ADLIB will target PVM, for portability. PVM is a public domain message-passing system with implementations on many platforms. In particular, it is available for machines connected by local area (or

even wide area) networks. This will allow HPF programs to run on networks of workstations as well as "conventional" parallel computers.

We will also provide a tuned implementation of ADLIB for the Meiko CS-2 (there is an installation of this hardware available at Southamton). The low-level primitives for that machine should allow a particularly efficient implementation.

Although ADLIB is designed specifically to support HPF, it is anticipated that it will be a useful tool in its own right for explicit Single-Program-Multiple-Data (SPMD) programming.

4 Collaborations and Related Projects

The project will involve close collaboration between SUCS and ECS. We have made contact with NAG, the vendors of one of the first successful Fortran 90 compilers. They are taking a close interest in the Southampton HPF work, and may provide software assistance. Within ECS, SERC funding is being sort for further research into HPF compiler optimisation issuces. The results of that research could eventually be integrated into the system to be developed under this proposal.

Discussions with NAG concerning full productisation and support are taking place.

5 Contribution to the Aims of the *New Technology Initiative*

The prototype HPF system proposed here seems to fit closely the requirements outlined in section 5 of the "invitation to bid". The level of support we are requesting is consistent with the "Small scale higher risk" category described in section 6b.

With reference to section 12:

a. There is a widespread feeling that the HPF model will prevail as an efficient, portable programming style for parallel scientific applications over the next few years.

b. Our software will be made freely available to students and researchers, for example, through a local FTP site or HENSA. We hope to distribute sources, as well as binaries. Certainly the sources of ADLIB will be made public, to facilitate porting efforts.

c. The educational potential of an HPF implementation is clear, given our hopes for the widespread adoption of the language. It will facilitate and encourage uptake of high performance computing in institutions of higher education.

d. We are not aware of similar academic projects within the UK.

e. The system will allow HE establishments to gain experience with HPF, free of charge.

f. See section 4.

g. We have identified programmers within the University of Southampton competent to engage in the project described.

6 Outline Work Plan and Deliverables

We seek funding for one experienced programmer to work on this project. His tasks will be

1. Complete the design of the HPF translation system.
2. Make the necessary extensions to the ADAPT translator, to support the HPF language.
3. Implement ADLIB for PVM and the Meiko CS-2.

In practice we hope that item 1 would be well advanced by the time the project started. We may choose to schedule item 3 before item 2, because as mentioned above ADLIB is a useful deliverable in its own right, even if for some reason the translator couldn't be completed within the time scale of the project.

Accordingly, the deliverables are:

1. A specification document describing the overall structure of the translation system. Most importantly, this will include a detailed definition of the ADLIB interface.
2. The implementations of ADLIB indicated above.
3. The HPF to Fortran 90 translator.
4. User Guide and Manual.
5. Establish mechanism for distribution of software.

7 Resources Required

We seek funding for one programmer for one year (in the first instance).

Resource	
Staff	
1	£25,000
Total Cost	£25,000