Efficient translation of High Performance Fortran for distributed memory architectures

Case for support

Purpose

The recently published High Performance Fortran (HPF) language standard is expected to have a major impact on the future of parallel computing, especially on distributed-memory MIMD machines. It is based on a simple and familiar programming language, offers wide portability, a simple migration method for existing Fortran codes, and the promise of high performance. There is every chance, therefore, that it will rapidly become the preferred language for scientific and engineering applications of parallel computers.

Until now, the lack of a programming language with these desirable features has been a major factor behind the slow take-up of distributed memory architectures. Despite their high performance and cost effectiveness—demonstrated across a range of important applications—the development cost of application software targetting specialised architectures too often outweighs any inherent price-performance advantage of the hardware.

Looking ahead some years, there is no obvious reason why HPF compilers should not produce code competitive with hand-crafted message-passing code. It has been argued that an HPF compiler can actually do better because, for example, it can target low-level machine instructions, and statically allocate message buffers (rather than going through message-passing portability layers). However, early HPF compilation systems are unlikely to generate such efficient code. HPF-like systems are in their infancy, and there has been relatively little study of optimisation issues. Research is urgently needed into efficient implementation of the complex HPF semantics.

The aim of this project is therefore to investigate HPF compiler optimisation, particularly in the following areas:

- To develop methods to determine optimal placement of sub-expression calculation, to supersede the widely-used, very crude, "owner-computes" heuristic.
- To investigate the relative merits of various mooted schemes for organising data communications in HPF-like languages.
- To implement the fruits of this research within a Subset HPF compilation system currently under development by our group, and evaluate the performance gains from these optimisations.
- To compare our Subset HPF system with the Distributed Memory Paralleliser and HPF compiler in the 'Forge 90' system, widely considered to represent the state-of-the-art in Fortran parallelisation for distributed-memory systems.

Our Subset HPF compilation system will generate Fortran 90 output with calls to a communication library. We plan to make this freely available to the research community.

NAG Ltd, who market a successful Fortran 90 compiler, will act as "heavy uncle" for this project. They will allow us to use relevant parts of their Fortran 90 compiler and run-time library source code, and provide other assistance. Subject to the negotiation of suitable terms, they will distribute our system together with their Fortran 90 compiler as a complete Subset HPF compilation system.