Parallel Library Development at NAG

Stefano Salvini *

October 7, 1994

Abstract

This paper introduces the work being carried at NAG Ltd towards the development of a Fortran 77 Library aimed at parallel, distributed memory machines, including networks and clusters of workstations.

1 Introduction

The advent of parallel, distributed memory systems has been a real challenge for the producers of numerical software libraries such as NAG. NAG has a long and strong tradition in the production quality numerical software, particularly our Fortran 77 and our new Fortran 90 Libraries. The term 'quality' embraces a number of features essential to software developers using our Libraries, such as efficiency, reliablity, accuracy, usability and, of course, portability.

Naturally, one of our principal aims is to shield users from the rapid changes and diversity in architecture. Just as software modules like the BLAS form a portability base for the current NAG Libraries, so the libraries form the portability base for users and their application packages and programs.

To date, our main activities have been concentrated on shared memory machines, on which our main Libraries are available. As progress on the development of new distributed memory systems proceeds apace, it is clear that this class of systems will aquire increasing importance for general scientific computing in the future. Hence we have resolved to start building a portable Parallel Library for such systems.

New parallel algorithms are required for such a Library. However, these will not be discussed in this paper, where we will rather concentrate on the underlying strategy of our approach.

2 Distributed Memory Systems and Portability Base

In the context of this paper, the term 'distributed memory' encompasses a variety of classes of systems, such as

- Message Passing Systems, for example, the Intel Paragon or Meiko CS-2.
- Virtual Shared Memory Systems, for example, the Kendall Square Research KSR-1 or the Cray Research T3D.
- Clusters of Workstations, for example, IBM SP-1 and SP-2, DEC-α-farms.

*Numerical Algorithms Group Ltd, Wilkinson House, Jordan Hill Road, Oxford OX2 8DR, UK, email: stef@nag.co.uk

- Networks of Workstations.

In order to guarantee portability across as wide a range as possible of such systems, a number of choices were made at the onset:

Programming Language Fortran 77 is being used throughout. Until Fortran 90 compilers reach a more mature stage, Fortran 77 remains the main language for general scientific and numerical computation. However, subsequent upgrade to the use of Fortran 90, and beyond that, to High Performance Fortran (HPF) [?] is also envisaged.

Programming Model The SPMD (Single Program Multiple Data) model is assumed throughout, using message passing. This is much less restrictive than it may seem, though it does not allow, for example, the use of multi-gridding.

Underlying Topology A 2-dimensional logical grid of processors or tasks is assumed. Mapping the logical grid on the actual topology of the system is a task of the operating system and communication mechanism.

Communication Mechanism BLACS (Basic Linear Algebra Communication Subprograms) [?] are used for all communications. While native BLACS are available on a range of systems, an implementation of these based on PVM (Parallel Virtual Machine) [?] exist for almost all systems. If BLACS are based on PVM, task management is carried out using PVM facilities; otherwise BLACS interface directly to the native communication and task management mechanisms. Where the functionality of BLACS was not sufficient, PVM subprograms were used, in a limited, encapsulated context to allow for easy portability to other underlying coommunication mechanisms.

Other Primitives BLACS also provide some primitives for global sums, max, etc.

3 A Library for Distributed Memory Machines

A numerical Library must consist of a number of components:

- Numerical software
- Documentation
- Test software, used to port the Library across a range of systems.
- Example programs: these are fully functional, albeit simple, programs which illustrate the use of the routines of the Library and can act as templates for the user.

Standards have been defined for all these components. Also, the Library has a very close relationship with our Fortran 77 Library, to allow users to move from one to the other with minimum induction.

Particular attention have been dedicated to internal Library mechanisms, such as error reporting and task management mechanisms. An error mechanism is non-trivial for distributed memory systems: it is imperative that all tasks, irrespective of the actual computations carried out, behave homogeneously when an error is detected in any task, i.e. all must report correctly any error condition raised.

Mark 1 of the Parallel Library has now been completed (end of September) and is being checked thoroughly and implemented on a number of target systems. We must stress that it consists of a 'pilot library', and we expect that by Mark 2 its functionality will have been extended considerably. The Library contains a number of routines:

Dense Linear Algebra Routines are provided for the solution of simultaneous systems of linear equations, QR factorization and relative solvers, reduction to tridiagonal and Hessenberg forms, computation of the eigenvalues of symmetric matrices, etc. Most of these routines were derived from ScaLAPACK [**?**].

Iterative Solvers These include the preconditioned Conjugate Gradient method, for sparse symmetric systems of simultaneous linear equations; the preconditioned RGMRES (Restarted Generalized Minimal Residual), Bi-CGSTAB (Bi-Conjugate Gradient Stabilized), TFQMR (Transpose Free Quasi-Minimal Residual) methods, for sparse unsymmetric systems. These routines were adapted from the PIM (Parallel Iterative Methods) package [**?**].

Quadrature A comprehensive set of routines for 1-, 2-, and multi-dimensional quadrature.

Optimization A routine for non-linear least squares minimization.

Random Numbers Routines for generating in parallel statistically independent sequences of random numbers (Wichmann-Hill).

Input/Output A set of routines to read/write matrices from/to disk and distribute them over the tasks.

Data Distribution Some routines to distribute matrices over the processors.

Task Management Routines to spawn tasks and initialize the BLACS, terminating BLACS and tasks.

Auxiliary Routines Routines such as BLAS, PBBLAS, and any other required auxiliary routines, including those dedicated to error handling.

The use of the SPMD model is not always straightforward, for example in our quadrature algorithms a farming model is more natural. However, SPMD is the HPF model of programming, and it does not really place any undue restriction on the user.

A number of utility routines further insulate the user and ourselves from the underlying message passing system. Before calling the library the user calls an initialization routine, and calls an exit routine at the end, along the lines of the following example.

```
*     ..  call the NAG initialisation routine ..
          call nagspawn( ...  )
          :
          call d02hafp( ...  )
          :
*     ..  call the NAG clean-up routine ..
          call nagexit( ...  )
```

Initialising PVM, when used, and the BLACS (or any other future communication system that we adopt) is done within `nagspawn`. From the user's perspective the error handling mechanism and interface is the same as for the Fortran 77 Library, with some additional error checks such as ensuring that a global variable has the same value on all processors.

4 Future Developments

Our strategy for the future development of parallel libraries is illustrated graphically in Figure **??**. Eventually, we expect that our Fortran 90 and Parallel Libraries will merge into an HPF

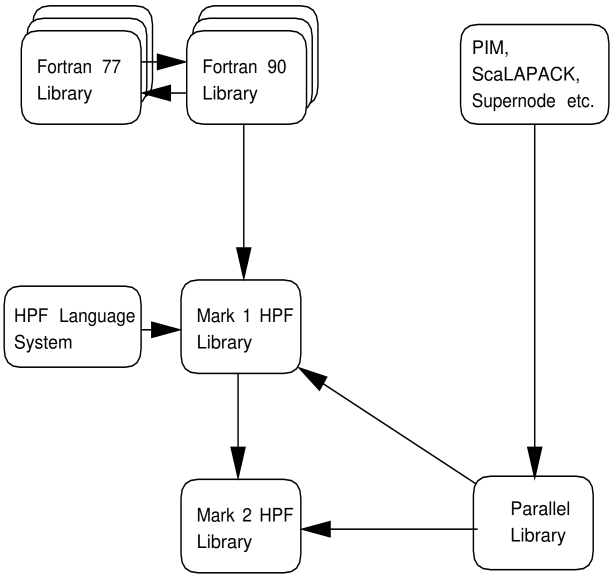

Figure 1: Strategy for Parallel Library Development

development activity. We hope that in future, HPF will allow us to maintain a single source library for both shared and distributed memory systems. It is not clear, though, whether we can can embrace networks of workstations as well, or whether they will continue to require a library explicitly based upon message passing. Also, we will take into account developments in MPI [**?**], with the aim of using it into the Library in the future.

5 Acknowledgements

This work represents the concerted effort of a group of people, including myself, at NAG Ltd, in particular M.Derakhshan, S.Hammarling, K.McDonald, V.Fernando, J.Du Croz. Also, the contributions of a number of people outside NAG Ltd are gratefully aknowledged.

References

[1] J. Choi, J. J. Dongarra, R. Pozo, and D. W. Walker. ScaLAPACK: A scalable linear algebra library for distributed memory concurrent computers. LAPACK Working Note No.55. Technical Report CS-92-181, Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA, 1992.

[2] R. D. da Cunha and T. Hopkins. PIM 1.1: The parallel iterative methods package for systems of linear equations – users' guide (Fortran 77 version). Technical report, Computing Laboratory, University of Kent at Canterbury, Kent CT2 7NZ, UK, 1994.

[3] J. J. Dongarra, R. A. van de Geijn, and R. C. Whaley. Users' guide to the BLACS. Technical report, Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA, 1993.

[4] Message Passing Interface Forum. MPI: A message-passing interface standard. Technical report, Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA, 1994.

[5] G. A. Geist, A. Beguilin, J. J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. PVM 3 user's guide and reference manual. Technical Memorandum ORNL/TM-12187, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA, 1993.

[6] C. H. Koelbel, D. B. Loveman, R. S. Schreiber, G. L. Steele Jr., and M. E. Zosel. *The High Performance Fortran Handbook*. The MIT Press, Cambridge, MA, USA, 1994.