Computing

R94/561

DL/SCI/TM103T

technical memorandum **Daresbury Laboratory**

DL/SCI/TM103T

SURVEY OF PARALLEL SOFTWARE PACKAGES OF POTENTIAL INTEREST IN SCIENTIFIC APPLICATIONS

by

R.J. ALLAN and P. LOCKEY, DRAL Daresbury Laboratory

September, 1994

94/779

DRAL
Daresbury Laboratory
Rutherford Appleton Laboratory

DRAL is part of the Engineering and Physical Sciences Research Council

Daresbury Laboratory

Daresbury
Warrington
Cheshire
WA4 4AD

ISSN 0144-5677

IMPORTANT

The EPSRC does not accept any responsibility for loss or damage arising from the use of information contained in any of its reports or in any communication about its tests or investigations.

Survey of Parallel Software Packages of potential interest in Scientific Applications

R.J.Allan and P.Lockey,
Daresbury Laboratory,
Daresbury, Warrington,
WA4 4AD, UK.

Email: {r.j.allan,p.lockey} @ daresbury.ac.uk

September 26, 1994

Abstract

Parallel software packages which may be of use in scientific and engineering applications of the type carried out on the parallel computing facilities at Daresbury Laboratory are surveyed. For each package, a brief description is given along with other useful information such as availability, contact addresses and systems supported.

keywords: parallel computing, software packages, scientific applications.

Contents

1 Introduction

This report provides a survey of software packages for parallel computers which may be of use in science and engineering applications. The aim is to enable scientific parallel computer users to identify whether there exists software for their particular needs without having to wade through the parallel computing literature. It is also hoped that the survey will help prevent the large amount of 're-inventing the wheel' which appears to take place in the field of parallel scientific computing.

1.1 Criteria for inclusion

Packages covered here are those which can be run on either true parallel machines, networked workstations, or on shared memory systems. In the main we restrict our attention to packages which are based on or can be used from Fortran 77, Fortran 90, C and C++, since these are the most widely used languages for applications and they are also available (or will eventually be available) on most machines.

The main criteria for inclusion is that a package should be of use in a scientific or engineering application. Most of the entries cover packages which are already in existence and available. However, packages which are under construction, or proposed software projects are included if they are thought to be of sufficient interest. Also included are packages offering 'novel' approaches to parallel programming which, although not widely used in these types of applications, could potentially be of use.

1.2 Package areas

To make the survey easier to digest, the software packages surveyed have been divided into nine main areas.

- **Parallel numerical libraries.** This covers software which provides parallel numerical algorithms.
- **Grid based tools.** Packages which provide facilities for construction and manipulation of grids.
- **Message Passing harnesses** which provide an environment in which parallel processes can interact and communicate with each other using message passing. Harnesses based around the MPI (Message Passing Interface) standard have their own section and are not included here.
- **MPI Implementations.** The MPI Standard is deemed important enough to warrant a separate section for message passing harnesses based on the standard.
- **Communication Skeletons** which provide higher-level communication facilities than the message passing model.
- **Languages and language extensions** for parallel programming which are particularly geared towards scientific applications. Language implementations based on the HPF standard have their own section and are not included here.

- **HPF Implementations.** The HPF standard, which extends Fortran 90 to cover data-parallel programming, is deemed to be of sufficient importance to warrant its own section.
- **Graphical tools.** This includes graphical interfaces to other parallel software, and also packages for graphical playback of parallel programs to aid debugging and performance tuning.
- **Alternative Parallel Models.** Packages which provide alternative models of parallel programming to the message-passing or data parallel models.

In practice the boundaries between these different areas are not always sharp. For example, some message passing communication harnesses provide extra facilities which overlap into other areas, such as numerical capabilities (eg. global numerical operations), alternative programming models (eg. shared memory 'monitors'), and higher level communication skeletons.

MPI (Message Passing Interface) and HPF (High Performance Fortran) are both (possibly) destined to be of great importance in the parallel scientific computing arena. For this reason they were both given their own sections.

1.3 Individual entries

The different fields for each package entry should be fairly self explanatory. Names and addresses given are simply somebody who can be contacted about the package; they are not meant to represent the entire cast responsible for the software. For full lists of the organisations and people involved the actual documentation should be consulted.

An earlier survey of parallel software was produced by Louis Turcotte (see section 11 for more details of this report). Packages covered here which also have an entry in the survey by Turcotte contain a cross-reference to that list giving the entry name, page number, and references in that document. For example, **(LT:P4,123)** reveals that a package has an entry in the survey by Turcotte, under the name P4, on page 123.

1.4 Intended Audience and Feedback

The survey is particularly geared towards users of the computing facilities at Daresbury Laboratory and for this reason also has a slight 'UK slant'. However, the information contained here should also be useful to a wider audience - simply ignore the comments relating to Daresbury!

It is our intention to keep this report as up to date as possible. To this end, we would be very keen to hear about any packages which are of interest in parallel scientific computing and are not currently included. Corrections and comments are also welcomed. We can be contacted at

- **P.Lockey @ Daresbury.ac.uk**
- **R.J.Allan @ Daresbury.ac.uk**

2 Parallel Numerical Libraries

This section covers Numerical Libraries for parallel machines which are of use in scientific applications. The main criteria for entry in this section is that a package provides parallel numerical algorithms/operations on some form of data.

The packages in this section encompass a wide range of functionality. At one end there are lower level routines such as parallel BLAS or parallel matrix multiply. At the other end are packages which provide a large number of higher-level numerical algorithms; These often make use of the lower level routines as 'building blocks'. There are three major efforts of this type covered here: The ScaLapack project at the University of Tennessee, the PETSc package from Argonne National Laboratory, and the Multicomputer Toolbox from Mississippi State University. These have individual entries later on in the section. However since these are important and innovative large scale projects we have included a brief discussion of them here, concentrating on the main concepts and philosophies behind the packages.

ScaLAPACK is an ongoing project aimed at creating a parallel distributed-memory version of the LAPACK (Linear Algebra Package) software. Part of the ScaLAPACK effort is involved with lower-level packages which provide building blocks for the higher-level numerical routines, but which are also useful packages in their own right. These include BLACS (Basic Linear Algebra Communication Subprograms) and PB-BLAS (Parallel Block Basic Linear Algebra Subprograms).

Routines which are currently provided in ScaLAPACK include:

- LU decomposition and solvers.
- QR factorisation and solvers.
- Cholesky factorisation and solvers.
- Condition estimation and iterative refinement for LU, QR and Cholesky.
- Reduction to upper hessenberg form.
- Reduction to tridiagonal or bidiagonal form.

Also, a number of major research projects have amalgamated their efforts under the ScaLapack umbrella. Packages in this group include ScaLapack itself, BLACS and PB-BLAS, and Lapack++, Arpack, PUMMA and CAPSS. All the packages are available from Netlib in directory scalapack. Most of the packages have their own specific contacts for feedback, queries etc, but there is also an overall contact at **scalapack @ cs.utk.edu**

PETSc - Portable Extensible Tools for Scientific Computation refers to a number of parallel software packages created at Argonne National Laboratory which have been brought together into a 'parallel toolkit'. The full set of tools is available in a single tar file by anonymous ftp from **info.mcs.anl.gov** in directory **pub/pdetools**. Included are:

- The Chameleon message passing system.
- The BlockComm package for communication of blocks of distributed arrays.

- The KSP iterative methods package.
- The SLES package which provides a higher level front-end and extensions to KSP.
- The SNEs package for solution of nonlinear systems.

Some of the key ideas behind the toolkit are covered in [75].

A basic design philosophy behind all the tools is 'data-structure-neutrality': Where possible the tools are written in terms of a number of key operations (for example matrix-vector multiply, vector product or vector norm). The user of the tool then provides their own routines for these operations. In this way, the user of the tool is not restricted to any particular data representation or storage scheme. This concept is described further in [71].

All the packages can still be obtained individually (see separate entries for Chameleon, BlockComm, KSP, SLES, and SNES).

The Multicomputer Toolbox is a project taking place at Mississippi State University to provide scalable, parallel libraries on a wide range of machines. The full set of tools are available by anonymous ftp from **aurora.cs.msstate.edu** in directory **pub/toolbox**. Included are

- CDASSL: Concurrent Differential-Algebraic Solver.
- Citer: Krylov Subspace iterative methods.
- Csparse: Sparse LU solvers.
- Cdense: Dense LU solvers.

These higher level libraries sit on top of lower level packages, such as concurrent BLAS, concurrent vector operations and transformations, and the Zipcode communications harness. The layered structure of the toolbox, and some of the concepts underlying the implementations are described further in [111] and [15].

Algorithms in the toolbox are formulated to be 'data-distribution-independent', allowing the higher level libraries to be flexible about the way the data upon which they operate is distributed. This removes the need for applications to redistribute data prior to calling library routines.

2.1 ARPACK

Name: ARPACK: Arnoldi's method Package

Description: ARPACK is a Fortran 77 package for solving large scale nonsymmetric, symmetric and generalised eigenvalue problems. The package is based on Arnoldi's method and can also compute a few eigenvalues and eigenvectors of a large, possibly sparse, matrix.

Systems: A parallel implementation is available for the Intel Touchstone Delta.

Contact: Danny Sorensen and Richard Lehoucq,
Department of Computational and Applied Mathematics,
Rice University, USA.

Email: sorensen @ rice.edu or lehoucq @ rice.edu

FTP: Available from netlib in directory **scalapack** (See section 11 for further information about netlib).

Comments:

References: [114]

2.2 Blocksolve

Name: Blocksolve

Description: The BlockSolve package contains iterative routines for solving large sparse symmetric systems of linear equations on massively parallel distributed memory systems. A number of different preconditioners are also provided.

Systems: Runs on top of the Chameleon message passing system (see separate entry).

Contact: Mark T. Jones and Paul E. Plassman,
Mathematics and Computer Science Division,
Argonne National Laboratory, USA.

Email: mjones @ mcs.anl.gov

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **pub/BlockSolve**

Comments:

References: [88]

2.3 CAPSS

Name: CAPSS: Cartesian Parallel Sparse Solver

Description: CAPSS is a fully parallel C package which solves a sparse linear system, where the matrix is symmetric positive definite, using Cholesky factorisation.

Systems: Intel iPSC/860 machines. The code uses message passing calls in PICL (see separate entry) and a few native iPSC/860 calls.

Contact: Padma Raghavan,
National Centre for Supercomputing Applications,
University of Illinois, USA.

Email: padma @ ncsa.uiuc.edu

FTP: Available from Netlib in directory **scalapack** (See section 11 for further information about netlib).

Comments:

References: [109]

2.4 DDL

Name: DDL: Distributed Data Library

Description: Package which provides facilities for creating, managing and operating on objects, such as matrices or vectors, which are physically distributed over a parallel machine, from a single threaded Fortran program. Can handle both dense and sparse objects.

Systems: Sits on top of PVM (see separate entry). A MPI version is currently being developed.

Contact: Cliff Addison,
Institute for Advanced Scientific Computation,
University of Liverpool, UK.

Email: cliff @ liverpool.ac.uk

FTP:

Comments: An example program which uses the sparse distributed objects available in DDL to solve a sparse system using the transpose-free QMR method is provided.
Available at Daresbury soon on Intel MPP and the workstation clusters.

References: [2, 87]

2.5 Distributed Iterative Linear System Solvers

Name:	Distributed Iterative Linear System Solvers
Description:	Package for solving linear systems of equations in parallel. Includes a number of different iterative methods and preconditioners. A number of different matrix storage schemes are available.
Systems:	Sits on top of PVM 3 (see separate entry). PICL version (see separate entry) in the pipeline.
Contact:	Victor Eijkhout, Department of Computer Science, University of Tennessee at Knoxville, USA
Email:	eijkhout @ cs.utk.edu
FTP:	Package and documentation available from **cs.utk.edu** in directory **/pub/eijkhout/code**
Comments:	An X-window interface is included. This requires tcl (embeddable scripting language), tk (X window toolkit and widgets), and wish (windowing shell), which are all available from **ftp.cs.berkeley.edu** in directory **/ucb/tcl**
References:	[58]

2.6 DSS libraries

Name:	Dakota Scientific Software libraries
Description:	Parallel and highly optimised versions of BLAS, LINPACK and LAPACK, tuned for the SPARC. Use exactly the same interfaces as the standard versions of these libraries, but run much faster.
Systems:	SunOS 4.1.3, Solaris 1.x, Solaris 2.x. Will run on either a single workstation or a network of workstations. Also makes use of multiple processors where they are available.
Contact:	Mike Boucher, Dakota Scientific Software, Inc. 501 east Saint Joseph Street Rapid City, SD 57701-3995, USA.
Email:	scisoft @ well.sf.ca.us or mboucher @ silver.sdsmt.edu or na.boucher @ na-net.ornl.gov
FTP:	
Comments:	Provides fault tolerance. If any of the workstations or CPUs develop software, hardware or network problems, then DSSLIB automatically reorganises the computation.
References:	(LT:DSSLIB,97)

2.7 GA toolkit

Name: Global Array toolkit

Description: Package provides tools for creating and manipulating matrices which are distributed over parallel processes. A process can asynchronously access bits of the matrix which are stored on other processes without them having to explicitly cooperate. Provides basic matrix operations such as fetch or store a patch of a distributed array, gather and scatter, and BLAS.

Systems: Currently runs on Intel MPP machines, IBM SP-1, KSR-2, and workstation clusters. Also available for multiple processor SUN, SGI, and IBM workstations using shared memory. Versions are planned for Cray T3D, CM-5 and SGI Challenge.

Contact: Jarek Nieplocha,
Molecular Science Research Center,
Pacific Northwest Laboratory, USA.

Email: j_nieplocha @ pnl.gov

FTP: Available by anonymous ftp from **ftp.pnl.gov** in directory **/pub/global**

Comments: Has been used by the quantum chemistry group at PNL for applications.

References:

2.8 KSP

Name: KSP: Krylov Space methods Package

Description: Package of iterative methods for the solution of linear systems of equations. Methods provided include conjugate-gradients, GMRES, BiCG-STAB, Transpose-free QMR, and others. New methods can easily be added to the package.

Systems: Package can run on single workstations, or in parallel (sitting on top of Chameleon - see separate entry) on the IBM SP1, Intel DELTA, CM-5, Cray C90, and Workstation clusters.

Contact: William Gropp,
Mathematics and Computer Science Division,
Argonne National Laboratory, USA.

Email: gropp @ mcs.anl.gov

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **pub/pdetools**

WWW: http://www.mcs.anl.gov/home/gropp/petsc.html

Comments: The methods are designed to be 'data-structure-neutral': They are based on a number of main operations which the package user provides (for example matrix-vector multiply or vector product), thus the user is not restricted to any particular data representation or storage scheme. Some of the more common storage schemes are provided with the package.
Part of the PETSc software toolkit for scientific computing. (See discussion at the beginning of this section for more information about PETSc).

References: [73, 71]

2.9 LAPACK++

Name: LAPACK++: Linear Algebra Package++

Description: LAPACK++ is an object-oriented version of LAPACK (Linear Algebra Package), written in C++. It contains a number of linear algebra methods, including solvers for linear systems of equations, linear least squares problems and eigenvalue problems. Future versions of the package will include 'distributed matrix' classes for use on parallel machines

Systems: Parallel version not implemented yet.
Built on top of LAPACK.

Contact: Jack J Dongarra,
Department of Computer Science,
University of Tennessee at Knoxville, USA.

Email: lapackpp @ cs.utk.edu

FTP: Available from Netlib in directory **c++/lapack++**.
(See section 11 for further information about netlib).

Comments:

References: [52][54]

2.10 Multicomputer Toolbox

Name: Multicomputer Toolbox

Description: Set of parallel libraries containing a large number of important numerical algorithms including ODE/DAE solver, Krylov Subspace iterative methods (GMRES, QMR, PCG, PTFQMR), Dense and Sparse LU solvers, and Concurrent BLAS. The toolbox is designed around the concepts of data-distribution-independence and uniform calling interfaces for different methods.

Systems: Sits on top of the Zipcode system (see separate entry).

Contact: Anthony Skjellum,
Computer Science Department & NSF Engineering Research Center,
Mississippi State University, USA.

Email: tony @ cs.msstate.edu

FTP: Available by anonymous ftp from **aurora.cs.msstate.edu** in directory **pub/toolbox**

Comments: A number of major applications have been developed using the toolbox including neutron transport, groundwater modelling, and process flow-sheeting codes.
See the discussion at the beginning of this section for more information about the Multicomputer Toolbox project.

References: [111, 15]

2.11 PARLANCE

Name:	PARLANCE: PARallel Library And Numerical Computing Environment
Description:	C++ object library which can be called from Fortran 77 or Fortran 90. Allows the composition and manipulation of 'global objects' made up of a number of component arrays on different parallel nodes. Arrays can be of any dimension and may overlap in the global array, allowing representation of halo data for cacheing.
Systems:	Currently uses PVM (see separate entry) for message passing, but MPI version is under construction.
Contact:	Robert Allan or Pete Lockey, Advanced Research Computing Group, Daresbury Laboratory, UK.
Email:	r.j.allan @ daresbury.ac.uk or p.lockey @ daresbury.ac.uk
FTP:	
Comments:	Currently internal alpha release. Previous version supported at Daresbury was written in Fortran 77. Available at Daresbury on Intel MPP and the workstation clusters as soon as possible.
References:	[5]

2.12 ParLib++

Name:	ParLib++, Parallel Programming Classes for C++
Description:	ParLib++ is a C++ class library which allows the user to declare 'parallel arrays', with a choice of different array types and parallel distributions. A number of different types of array operations such as mathematical operators $(+, *, -, /)$, logical operators $(<, >, ==, !=)$, reduction operators (min,max,sum), Fast Fourier Transforms etc. can then be applied to these parallel arrays.
Systems:	The package uses PVM 3.2 for communications (see separate entry for PVM). Makes use of the HDF file storage system from NCSA.
Contact:	David Greco, Parallel Computing Group, Centre for Advanced Studies, Research and Development, in Sardinia
Email:	David.Greco @ crs4.it
FTP:	
Comments:	A number of example programs are provided, including a seismic migration algorithm (using a spectral finite difference scheme) and solution of Laplace's equation. The HDF package is available by anonymous ftp from **ftp.ncsa.uiuc.edu** in directory **HDF**.
References:	[70]

2.13 PBBLAS

Name: PBBLAS: Parallel Block BLAS

Description: Fortran 77 package to provide parallel BLAS routines for block partitioned matrices. Package contains all 9 of the level 3 BLAS and 4 of the level 2 BLAS.

Systems: Uses BLACS or PICL (see separate entries) for communications. Also requires BLAS for computation.

Contact: Jack J Dongarra,
Department of Computer Science,
University of Tennessee at Knoxville, USA.

Email: scalapack @ cs.utk.edu

FTP: Available from Netlib in directory **scalapack** (See further information about netlib in section 11).

Comments: PBBLAS imposes some restrictions on the distributions of the matrices. Only one of the matrices can be distributed in both directions. The other two matrices must be split into either a column of blocks or a row of blocks. These restrictions allow optimisations to be made which would not be possible in the case of fully general distributions. Multiplication of matrices with more general distributions can be carried out using PUMMA (see separate entry).
Part of the ScaLAPACK project (see the discussion at the beginning of this section for further information about ScaLAPACK).

References: [44, 45]

2.14 PIM

Name: PIM: Parallel Iterative Methods

Description: Fortran 77 package containing a number of iterative methods for the solution of systems of linear equations. Methods provided include a number of variants of Conjugate Gradients (CG,Bi-CG,CGS,Bi-CGSTAB,CGNR), Generalised Minimum Residual (GMRES), Generalised Conjugate Residual (GCR) and Transpose-free Quasi Minimum Residual (TFQMR). The user may choose the particular stopping criteria required for the iterative method. The user can also supply a preconditioner to be used with the method.

Systems: The package has been tested on the Intel Paragon using the NX library, the Cray Y-MP2E/233, and networks of workstations using PVM 3.1, p4 1.2 and TCGMSG 4.02 (see separate entries for PVM, P4, and TCGMSG).

Contact: Rudnei Dias da Cunha and Tim Hopkins,
Computing Laboratory, University of Kent at Canterbury, UK

Email: rdd @ ukc.ac.uk or trh @ ukc.ac.uk

FTP: Available by anonymous ftp from **unix.hensa.ac.uk** in directory **/misc/netlib/pim**
User's guide is in file **/misc/ukc.reports/comp.sci/reports/2-94.Z**

Comments:

References: [51]

2.15 PRISM

Name: PRISM: Parallel Research on Invariant Subspace Methods

Description: PRISM is a research project aimed at producing portable parallel eigensolvers and also lower level kernels for use in building these solvers. The project makes use of algorithms based on invariant subspace decomposition approaches, which have a higher sequential complexity than other more commonly used algorithms but are significantly more scalable on a parallel machine.

Systems: Implementations have been tested on the Intel Delta, however currently only sequential versions of the solvers are available, for Sun and Cray machines.

Contact: Steve Lederman,
Supercomputing Research Center, USA

Email: lederman @ super.org

FTP: Implementations and technical reports are available for anonymous ftp at **ftp.super.org** in directory **pub/prism**

Comments:

References: [22, 11, 18, 12, 13, 118, 85, 21, 86, 19, 83, 20, 82, 84]

2.16 PUMMA

Name: PUMMA: Parallel Universal Matrix Multiply Algorithm

Description: Fortran 77 Routines to provide parallel matrix multiplication for block partitioned matrices.

Systems: Uses BLACS or PICL (see separate entries) for communications. Also requires BLAS for computation.

Contact: Jack J Dongarra,
Department of Computer Science,
University of Tennessee at Knoxville, USA.

Email: scalapack @ cs.utk.edu

FTP: Available from Netlib in directory **scalapack** (See further information about netlib in section 11).

Comments: Imposes fewer restrictions on the distributions of the matrices than PBBLAS (see separate entry).
Part of the ScaLAPACK project (see the discussion at the beginning of this section for further information about ScaLAPACK).

References: [47, 48]

2.17 ScaLAPACK

Name: ScaLAPACK: Scalable Linear Algebra Package

Description: ScaLAPACK is a distributed memory version of LAPACK (Linear Algebra Package), written in Fortran 77. It contains routines to carry out a number of Linear Algebra operations in parallel, including LU decomposition, Cholesky factorisation, QR factorisation, reduction to upper Hessenberg form, tridiagonal form and bidiagonal form etc.

Systems: Intel Gamma, Delta, and Paragon, Thinking Machines CM-5, and PVM 3.2 (see separate entry).

Contact: Jack J Dongarra,
Department of Computer Science,
University of Tennessee at Knoxville, USA.

Email: scalapack @ cs.utk.edu

FTP: Available from Netlib in directory **scalapack** (See further information about netlib in section 11).

WWW: http://www.netlib.org/scalapack/index.html

Comments: The ScaLAPACK package sits on top of BLAS, PBBLAS and BLACS (see separate entries for PBBLAS and BLACS).
Available at Daresbury soon on the Intel MPP and the workstation clusters.
See discussion at the beginning of this section for further information about ScaLAPACK.

References: [53, 46],(LT:DLAPACK,90)

2.18 SLES

Name: SLES: Simplified Linear Equation Solvers

Description: Package of methods for the solution of linear systems of equations. Both iterative and direct methods are provided. The package is designed to allow any matrix representation or storage scheme to be used. New methods can easily be added to the package without having to make changes to the code.

Systems: Package can run on single workstations, or in parallel (sitting on top of Chameleon - see separate entry) on the IBM SP1, Intel DELTA, CM-5, Cray C90, and Workstation clusters.

Contact: William Gropp,
Mathematics and Computer Science Division,
Argonne National Laboratory, USA.

Email: gropp @ mcs.anl.gov

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **pub/pdetools**

WWW: http://www.mcs.anl.gov/home/gropp/petsc.html

Comments: Uses the KSP package (see separate entry) for some of the underlying iterative methods.
Part of the PETSc software toolkit for scientific computing. (See discussion at the beginning of this section for more information about PETSc).

References: [72]

2.19 SNES

Name: SNES: Simplified Nonlinear Equation Solvers

Description: SNES provides a number of iterative methods for the solution of nonlinear systems of equations. The package allows the user a great deal of flexibility in the choice of data structures and solution methods used.

Systems: Package can run on single workstations, or in parallel (sitting on top of Chameleon - see separate entry) on the IBM SP1, Intel DELTA, CM-5, Cray C90, and Workstation clusters.

Contact: William Gropp,
Mathematics and Computer Science Division,
Argonne National Laboratory, USA.

Email: gropp @ mcs.anl.gov

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **pub/pdetools**

WWW: http://www.mcs.anl.gov/home/gropp/petsc.html

Comments: Uses the SLES package (see separate entry) for intermediate linear solves, although this can be replaced by the user if required.
Part of the PETSc software toolkit for scientific computing. (See discussion at the beginning of this section for more information about PETSc).

References:

3 Grid based tools

This section covers packages which provide facilities for the creation and manipulation of grids, both regular and irregular.

3.1 AMR++

Name: AMR++, Adaptive Mesh Refinement Class Library

Description: A C++ class library for building self-adaptive mesh refinement applications. Parallelisation and array handling are inherited from P++ (see entry for P++).

Systems: Built on P++ and M++ (see entry for P++) and the AT&T Standard component class library. Because AMR++ sits on top of P++, it should run on all systems that P++ runs on (see entry for P++). Has been tested on Intel iPSC/860 and SUN workstations.

Contact: Dan Quinlan

Email: dquinlan @ c3.lanl.gov

FTP:

Comments: AMR++ sits on top of P++, which has been intensively optimised. However there is still much scope for optimisation in AMR++ itself.

References: [95, 14]

3.2 GenMP

Name: GenMp: Generic MultiProcessor

Description: Package for carrying out grid based computation on a parallel machine. Allows dynamic partitioning of grids and coordination of separate partitions.

Systems: Intel iPSC/860 and Cray Y-MP

Contact: Scott R. Kohn,
University of California at San Diego, USA.

Email: skohn @ cs.ucsd.edu

FTP:

Comments: Predecessor to the LPAR-X package (see separate entry).

References: [8]

3.3 LPAR-X

Name: LPAR-X: Lattice Parallelism

Description: LPAR-X is a C++ package which provides the user with distributed arrays in a shared name space. The arrays can be dynamic and non-uniform, allowing adaptive mesh type applications to be used.

Systems: CM5, Intel Paragon, Intel iPSC/860, nCUBE/2, KSR, Networked workstations under PVM, Single workstations, Cray C-90 and T3D.

Contact: Scott R. Kohn,
Department of Computer Science and Engineering,
University of California at San Diego, USA.

Email: skohn @ cs.ucsd.edu

FTP: Available by anonymous ftp from **ftp.sdsc.edu** in directory **pub/sdsc/parallel/lparx**

Comments: Partly based on earlier work with the GenMP package (see separate entry).

References: [10, 9]

3.4 P++

Name: P++

Description: A C++ class library providing facilities for creating parallel codes for structured grid applications. Can handle 1D-4D block-structured grids. Hides most of the parallel bits, with the package user only needing to specify details of their grid application.

Systems: Currently compiled using the AT&T C++ C-Front precompiler 2.1. Uses the Intel NX-2 library for communications, or an 'EXPRESS-like' portable communication library from Caltech. Has been run on the Intel iPSC/860, Intel Simulator, SUN workstations, Cray 2, and IBM PC's.

Contact: Dan Quinlan

Email: dquinlan @ c3.lanl.gov

FTP:

Comments: Uses the M++ array library for sequential array operations. The P++ package also has exactly the same user interface as M++, thus sequential M++ programs can be run in parallel under P++ with no changes (and vice versa).
(M++ is a sequential C++ class library produced by **Dyad Software Corporation** which provides array syntax and features similar to Fortran-90).
P++ has been intensively optimised to give high performance.

References: [96, 94]

4 Graphical Tools

This section contains graphical tools for use in parallel programming, in particular those that provide graphical analysis and playback of the execution of a parallel program. Also included here are packages which produce trace data or log data for use in graphical packages.

4.1 ALOG

Name: ALOG

Description: ALOG produces log files from the execution of a parallel program, which can then be viewed using the UPSHOT package (see separate entry).

Systems: Supported machines include Sun, Next, IBM 3090, RS6000, Symmetry, Balance, Multimax, Butterfly 1 & 2, Intel iPSC 860 & delta, Titan, and SGI.

Contact: Ewing Lusk,
Argonne national Laboratory, Argonne, USA.

Email: lusk @ mcs.anl.gov

FTP: Included with the P4 distribution (see separate entry), which is available from Netlib in directory **p4** (see section 11 for further information about netlib).

Comments: Output format from ALOG is described in the User Manual for the UPSHOT package [80].

References:

4.2 ParaGraph

Name: ParaGraph

Description: ParaGraph allows the graphical playback of a parallel program based on execution trace data which was collected during the program execution. The trace information can be viewed in a number of different ways, for example processor utilisation against time, or communications traffic against time.

Systems: Requires X Window Xlib library and PICL (see separate entry).

Contact: Michael T. Heath,
University of Illinois, USA.

Jennifer E. Finger,
Engineering Physics and Mathematics Division,
Oak Ridge National Laboratory,
Oak Ridge, USA.

Email: heath @ ncsa.uiuc.edu
jenn @ msr.epm.ornl.gov

FTP: Available from Netlib in directory **paragraph** (see section 11 for further information about netlib).

Comments: ParaGraph operates on trace data produced by PICL (see separate entry), but will also work with any other message passing package as long as it produces trace data in the required format.
A package is available which generates trace files from PVM. See separate entry for PGPVM.

References: [79] (LT:ParaGraph,124)

4.3 PGPVM

Name: PGPVM

Description: PGPVM Produces trace files from PVM, which can then be viewed using the ParaGraph package (see separate entry).

Systems: Works with version 3.3 of PVM.

Contact: Brad Topol,
Georgia Institute of Technology, USA.

V.S.Sunderam
Department of Math and Computer Science,
Emory University, USA.

Email: topol @ cc.gatech.edu or topol @ mathcs.emory.edu or vss @ mathcs.emory.edu

FTP: Available by anonymous ftp from **mathcs.emory.edu** in directory **pub/topol**

Comments:

References: [117]

4.4 UPSHOT

Name: UPSHOT

Description: X-based tool which graphically displays information about the execution of a parallel program, using data collected in a log file. The required format of a log file is described in [80].

Systems: Uses the Athena Widget set and Xl toolkit for the X stuff.

Contact: Ewing Lusk,
Argonne national Laboratory, Argonne, USA.

Email: lusk @ mcs.anl.gov

FTP: Available from Netlib in directory **p4**

WWW: http://www.mcs.anl.gov/home/lusk/upshot/index.html

Comments: Log files in the required format can be generated using ALOG (see separate entry).
Versions of STRAND and PCN (see separate entries) can also produce log files in the required format.

References: [80] (LT:Upshot,147)

4.5 XMPI

Name: XMPI

Description: X/Motif Graphical user interface for running and debugging MPI programs. Allows the user to take a 'snapshot' of the parallel MPI program at any time during its execution. This provides information about the overall execution of the application, execution status of each process, and communication status of each process.

Systems: Works with LAM 5.2 (see separate entry). Runs on Sun 4, SGI IRIX, IBM RS/6000, DEC AXP, and HP 9000.

Contact: Jim Vaigl, The Trollius Project,
Research Computing,
The Ohio Supercomputer Center, USA.

Email: trollius @ tbag.osc.edu

FTP: Available by anonymous ftp from **tbag.osc.edu** in directory **pub/lam**

Comments: A Graphical trace visualisation tool is also under construction. The trace collection mechanisms are already built into LAM.

References:

4.6 XPVM

Name: XPVM

Description: Graphical interface for PVM, providing a graphical version of the PVM monitor and also animated viewing of PVM programs during execution.

Systems: Sits on top of PVM version 3.3.0 or greater.
Requires TCL 7.3, TK 3.6.1 or later.

Contact: Jim Kohl,
Engineering Physics and Mathematics Division,
Oak Ridge National Laboratory, Oak Ridge, USA.

Email: kohl @ msr.epm.ornl.gov

FTP: Available from Netlib in directory **pvm3/xpvm**

Comments: TCL is an 'embeddable scripting language'.
TK is based on TCL and provides an X11 toolkit and widgets.
Both are available from **ftp.cs.berkeley.edu** in directory **/ucb/tcl**

References:

5 Message Passing Harnesses (excluding MPI)

This section covers packages which provide implementations of the message-passing parallel programming model. Some of these packages also offer other facilities such as debugging and monitoring capability, and numerical operations such as global sum, global maximum etc. Some packages also support other programming models, for example P4 supports 'monitors' when used on a shared memory system.

5.1 Chameleon

Name: Chameleon

Description: Message-passing harness for distributed memory parallel machines. Designed to be portable, have a very low overhead, and help standardize operations such as parallel program startup and group operations which differ from one system to another.

Systems: Sits on top of P4, PVM, PICL, or native message passing systems such as Intel NX, IBM EUI and CM5.

Contact: William Gropp,
Mathematics and Computer Science Division,
Argonne National Laboratory, USA.

Email: gropp @ mcs.anl.gov

FTP: Package and documentation Available via anonymous ftp at **info.mcs.anl.gov** in directory **pub/pdetools**

Comments:

References: [74] (LT:Chameleon,82)

5.2 Express

Name: Express

Description: Parallel programming system based around the Express message passing harness. As well as the usual message passing capabilities this also incorporates higher level functions which are not included in the MPI standard. These include functions for data distribution, data exchange (combined send and receive), parallel I/O, parallel graphics output, and load balancing. Can be used from either Fortran or C.

Systems: nCube, Intel iPSC machines, Cray, IBM/3090, transputers, networked workstations.

Contact: ParaSoft Corporation, 2500 E. Foothill Blvd,
Pasadena, CA 91107, USA.

Email: info @ parasoft.com

FTP: Information about Express available via anonymous ftp from **ftp.parasoft.com** in directory **pub/express**

WWW: http://www.parasoft.com/express.html

Comments: As well as the message passing harness the system also includes an automatic paralleliser, algorithmic visualiser, parallel debugger, and performance analysis tools.

References: [61] (LT:Express,100)

5.3 FortNet

Name: FortNet

Description: Portable message passing harness developed for use in scientific applications. Integrated with a number of other parallel tools including a graphical execution monitoring package and a parallel program development tool.

Systems: Meiko Computing Surface, Intel iPSC machines, Cray X-MP or Y-MP, Alliant fx/8 or fx/2800, Kendall Square KSR1, Unix workstations using sockets or sitting on top of PVM, and various transputer machines and environments.

Contact: Robert Allan,
Advanced Research Computing Group,
Daresbury Laboratory, UK.

Email: r.j.allan @ daresbury.ac.uk

FTP:

Comments: Previously used for in-house application development in Computational Fluid Dynamics, Atomic Physics, and Parallel Software Development. Now largely superceded by PVM and MPI.
Also used in early versions of PARLANCE (see separate entry), and to test message passing extensions to the Argonne Schedule package.
Contains a profiler and interface to Paragraph.

References: [4]

5.4 NXLib

Name: NXLib

Description: A package which provides Intel Paragon native message passing calls for use on a cluster of workstations. Includes synchronous, asynchronous and interrupt driven Paragon communications. Allows Paragon programs to be developed on a workstation cluster or programs previously written for the Paragon to be run on a workstation cluster.

Systems: Current version 1.1 running on SGI, HP 9000, IBM RS6K, Dec Alpha and Sun workstations.

Created by: Stefan Lamberts, Georg Stellner, Thomas Ludwig, Arndt Bode,
SAB, Institut für Informatik,
Technische Universität München.

Email: nxlib @ informatik.tu-meunchen.de

FTP: Package and documentation available via anonymous ftp from **ftpbode.informatik.tu-muenchen.de** in **NXLIB** directory.

WWW: http://wwwbode.informatik.tu-muenchen.de/lamberts/NXLib/NXLib.html

Comments: Most of the calls on the Intel iPSC/860 are also covered by this package.
Available at Daresbury on the Sun front end to the Intel MPP (*tcsa*).

References: [115]

5.5 P4

Name: P4, Portable Programs for Parallel Processors

Description: A library for writing parallel programs in Fortran or C for shared-memory or message-passing machines. It will work on a network of workstations or 'real' parallel machines. Provides a number of different parallel models, including *monitors,message-passing*, and *clusters* for machines which support them.

Systems: Current version P4-1.3 available for practically everything, including Unix workstations, Intel, Cray, KSR, Thinking Machines, IBM.

Contact: Ralph Butler and Ewing Lusk,
Argonne national Laboratory, Argonne, USA.

Email: lusk @ mcs.anl.gov

FTP: Available from Netlib in directory **p4** (see section 11 for further information about netlib).

WWW: http://www.mcs.anl.gov/home/lusk/p4/index.html

Comments:

References: [36, 37],(LT:P4,123)

5.6 PARMACS

Name: PARMACS: PARallel MACros

Description: Portable message passing system for Fortran and C programs.

Systems: Available for Intel iPSC machines, nCUBE/2, Suprenum, Meiko CS-1, Cray Y-MP and C90, Parsytec GC, CM-5, and clusters of workstations based on PVM (see separate entry).

Contact: Rolf Hempel,
German National Research Centre for Computing Science (GMD),
Sankt Augustin, Germany.

Email: hempel @ gmd.de

FTP: Earlier version available from Netlib in directory **parmacs** (see section 11 for further information about netlib).

Comments: PARMACS has evolved through a number of versions. It was originally developed as basic macros for C programs at Argonne National Laboratory. Subsequently extra functionality and Fortran macros were added via a collaboration of ANL and GMD. The effort then split into further work on PARMACS at GMD, and evolution into the P4 package (see separate entry) at ANL. Currently research and development of PARMACS continues at GMD, and the package is commercially marketed and supported on a number of implementations by Pallas Software.
The list of systems above refers to PARMACS 5.1 which still made use of macros. PARMACS 6.0 now uses subroutine calls and implementations for the above systems are in progress. Utilities are available to convert programs using version 5.1 macros to version 6.0 subroutines.

References: [39]

5.7 PICL

Name: PICL: Portable Instrumented Communications Library

Description: Message passing library which is portable over a number of different machines. As well as the basic message passing operations it has a number of higher level operations such as global arithmetic and global boolean operations. Also provides execution tracing to assist debugging or performance tuning.

Systems: Runs on NX/2 (most Intel MIMD machines), VERTEX (nCUBE machines) and on top of PVM 3.3 (see separate entry).

Contact: P.H.Worley,
Mathematical Sciences Section,
Oak Ridge National Laboratory, USA.

Email: worleyph @ ornl.gov

FTP: Package and User Guide are available from Netlib in directory **picl** (see section 11 for further information about netlib).

Comments: Traces can be displayed graphically using **Paragraph** (see separate entry).

References: [67] (LT:PICL, 128)

5.8 PVM

Name: PVM: Parallel Virtual Machine

Description: A message passing harness which allows the user to write message passing programs in Fortran or C. These can then be run on a heterogeneous workstation cluster, parallel machine, single workstation, or collection of the above, with no change to the code.

Systems: Current version available for practically everything, including most Unix workstations, Thinking Machines, Cray, Intel, KSR.

Address: Engineering Physics and Mathematics Division,
Oak Ridge National Laboratory, Oak Ridge, USA.

Email: pvm @ msr.epm.ornl.gov

FTP: Available from Netlib in directory **pvm3** (see section 11 for further information about netlib).

WWW: http://www.eece.ksu.edu/pvm3/pvm3.html

Comments: Probably the most widely used message passing harness at the current time, hence it is now a 'de-facto standard'.
A PVM newsgroup exists, called **comp.parallel.pvm**
Available at Daresbury Laboratory on Intel MPP and the workstation clusters.
Will initially be one of the main message passing environment on the Cray T3D at Edinburgh.

References: [66, 116] (LT:PVM,134)

5.9 PVM-ATM

Name: PVM-ATM: PVM over ATM networks

Description: A version of PVM which runs over ATM (Asynchronous Transfer Mode) networks, allowing PVM programs to make use of the high bandwidth of the ATM network.

Systems: Current version compatible with PVM3.3.2. Requires Fore Systems ATM cards and network. In the process of being extended to support other ATM cards/networks.

Address: Distributed Multimedia Center and Computer Science Department, University of Minnesota, USA.

Email: pvm-atm @ cs.umn.edu

FTP: Available by anonymous ftp from **ftp.cs.umn.edu** in directory **/users/du/pvm-atm**

WWW: ftp://ftp.cs.umn.edu/users/du/pvm-atm/www.html

Comments: Currently being tested at Daresbury Laboratory on the workstation clusters.

References: [97]

5.10 TCGMSG

Name: TCGMSG: Theoretical Chemistry Group Message Passing System

Description: Message passing harness which is portable over a number of different machines. Provides message passing operations and also additional features including global operations, shared counters, event tracing, and communication statistics.

Systems: Runs on networks of workstations, and a number of parallel machines including Cray, KSR, Alliant, and Intel.

Contact: Robert Harrison,
Battelle Pacific Northwest laboratory, USA.

Email: rj_harrison @ pnl.gov

FTP: Available by anonymous ftp from **ftp.tcg.anl.gov** in directory **pub/tcgmsg**

Comments: Used by the Theoretical Chemistry Group at PNL in their production codes.
Comes with a set of example chemistry applications.

References: (LT:TCGMSG, 143)

5.11 ZIPCODE

Name: ZIPCODE

Description: Portable message passing system which includes a number of higher level features which are necessary for building libraries and large scale application software. These include process groups, communication contexts, collective operations, and virtual topologies.

Systems: Runs on Ncube/2, Intel MPP Machines, BBN TC2000, CM-5, Networked Sun and RS6000 networks, and on top of PVM (see separate entry).

Contact: A Skjellum,
Computer Science Department and NSF Engineering Research Center,
Mississippi State University, USA.

Email: tony @ cs.msstate.edu

FTP: Available by anonymous ftp from **aurora.cs.msstate.edu** in directory **pub/toolbox**

Comments: Many of the more innovative features of Zipcode have been included in the MPI standard.
Authors of Zipcode intend to use it as a base to explore further concepts which are not covered in MPI such as threads and active messages.

References: [113, 121]

6 MPI Implementations

This section contains packages which implement some or all of the MPI (Message Passing Interface) standard.

MPI is an attempt to provide a standard for writing message passing parallel programs. The development of the standard involved a large number of organisations, both academic and industrial. The main aims were to produce a standard for message passing which was efficient, reliable, portable, and flexible. In addition to point-to-point message passing MPI defines collective (reduction) operations, groups, contexts and communicators, including the best ideas from previous message passing implementations.

Reading the MPI FAQ is probably the best way to get started with MPI. It contains all of the information in this section and much more. It is available by anonymous ftp from **aurora.cs.msstate.edu** in directory **/pub/mpi/faq** in a number of formats, including ascii, postscript and Emacs info format.

The official MPI standard document is available from netlib in directory **mpi** in Postscript form (beware - it is almost 200 pages long).

The anonymous ftp site **aurora.cs.msstate.edu** contains a number of papers relating to MPI. A paper written by Tony Skjellum [112] describes how to make use of the special features of MPI when writing parallel library routines. This is in directory **pub/reports/SPLC93** in file **mpi_libraries.ps.Z**
Another paper by the same author describes a number of early attempts to use MPI in practical applications. This is in directory **pub/reports** in file **early_apps_mpi.ps.Z**
Also on the same site in directory **pub/mpi/papers** in file **MPI.bib** there is a bibliography of papers relating to MPI.

There are a couple of WWW pages devoted to MPI.
http://www.cs.msstate.edu/dist_computing/mpi.html contains a number of papers relating to MPI, the FAQ (Frequently Asked Questions) and a hypertext version of the MPI standard document.
http://www.mcs.anl.gov/mpi/index.html contains the FAQ, a postscript version of the MPI standard document, and an errata for the standard document which also lists a number of unresolved issues.

There exists a newsgroup covering MPI issues - **comp.parallel.mpi**

Finally, there are a number of implementations of MPI now available. Three of these are covered in this survey:

- **MPI Test Implementation** from Argonne National Laboratory and Mississippi State University.
- **LAM** from Ohio Supercomputer Centre.
- **CHIMP** from Edinburgh Parallel Computing Centre.

See separate entries for MPI, LAM and CHIMP for more details of these packages.

6.1 CHIMP/MPI

Name:	CHIMP: Common High Level Interface to Message Passing
Description:	Portable message passing system developed at Edinburgh Parallel Computing Centre for use in in-house parallel application work. Now supports most of the MPI interface through an MPI compatibility library which sits directly on top of CHIMP.
Systems:	Runs on a number of Unix workstations (Sun, Silicon Graphics, DEC Alpha, IBM RS/6000) and Meiko Computing Surface 1 and 2.
Contact:	Lyndon J Clarke, Edinburgh Parallel Computing Center, UK.
Email:	EPCC-Support @ ed.ac.uk
FTP:	Available by anonymous ftp from **ftp.epcc.ed.ac.uk** in directory **pub/chimp/release** User guide available in **pub/chimp/release/doc**
Comments:	See section 6 for further information about the MPI standard.
References:	[3] (LT:CHIMP, 83)

6.2 LAM/MPI

Name: LAM: The Local Area Multicomputer

Description: A message passing harness which provides a full implementation of the MPI standard for Fortran or C programs running on clusters of Unix workstations. Also provides PVM 3 compatiblity, parallel I/O, and debugging and monitoring facilities.

Systems: Currently runs on Sun 4, SGI, IBM RS6K, DEC AXP, HP 9000.

Address: Jim Vaigl, The Trollius Project,
Research Computing,
The Ohio Supercomputer Center, USA.

Email: trollius @ tbag.osc.edu

FTP: The package and full documentation are available via anonymous ftp from **tbag.osc.edu** in directory **pub/lam**

WWW: http://www.osc.edu/lam.html

Comments: An X/Motif graphical user interface is available for LAM. See separate entry for XMPI.

References: [32, 103, 104, 105, 34, 33]

6.3 MPI Model Implementation

Name: MPI: Message Passing Interface Model Implementation

Description: Implementation of the MPI standard, including both the Fortran and C bindings. Supports nearly all the MPI features.

Systems: Sits on top of the Chameleon message passing harness (see separate entry for Chameleon). Can also run directly on top of IBM EUI, Intel NX, P4 (see separate entry), CM5 and nCUBE.

Contact: Bill Gropp and Rusty Lusk,
Mathematics and Computer Science Division,
Argonne National Laboratory, Argonne, USA.

Tony Skjellum and Nathan Doss,
Mississippi State University, USA.

Email: gropp @ mcs.anl.gov or lusk @ mcs.anl.gov
Bug reports to mpi-bugs @ mcs.anl.gov

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **/pub/mpi**.

Comments: A major application code (nuclear-structure code) used in Argonne's Physics Division has been ported to MPI and run under the model implementation.
See section 6 for further information about the MPI standard.
Available at Daresbury on the Intel MPP machine.

References: (LT:MPI, 114)

6.4 UNIFY

Name: UNIFY

Description: Patches PVM to give a subset of the MPI standard from within the PVM environment. Designed to provide easy migration from PVM to MPI. Provides only static process creation, C-Interface and SPMD model at the moment.

Systems: Patches PVM versions 3.2, 3.2.6 or 3.3 (see separate entry for PVM). Currently works with the network versions of PVM only. Has been tested on SUN and SGI workstations.

Contact: Tony Skjellum, Paula Vaughan and Donna Reese,
NSF Engineering Research Center for Computational Field Simulation,
Mississippi State University.

Email: unify @ erc.msstate.edu

FTP: Available by anonymous ftp from **ftp.erc.msstate.edu** in directory **unify**

Comments: Allows applications written using PVM to make use of MPI libraries and calls.

References: [120]

7 Communication Skeletons

This section contains packages which provide higher-level communication facilities than those provided in a basic message passing environment. These are mainly geared towards moving pieces of matrices or grids around, without having to explicitly use message passing.

The GA toolkit (see entry in section 2) is also appropriate for this section as it provides high-level communication functionality including the ability to move sub-blocks of matrices around in an asynchronous manner. However it also provides higher level operations such as BLAS and eigensolvers and has therefore been placed in the numerical libraries section (section 2).

7.1 BLACS

Name: BLACS: Basic Linear Algebra Communication Subroutines

Description: Package of 'communication skeletons' for use in parallel linear algebra codes on message passing machines. Designed for efficient communication operations on 2D arrays and sub-arrays on a rectangular mesh of processors.

Systems: Runs on Intel MIMD machines, CM-5, and on top of PVM (see separate entry).

Contact: Jack J Dongarra,
Department of Computer Science,
University of Tennessee at Knoxville, USA.

Email: blacs @ cs.utk.edu

FTP: Available from Netlib in directory **scalapack** (see section 11 for further information about netlib).

WWW: ftp://cs.utk.edu/pub/rwhaley/HTML/Blacs.html

Comments: Created as part of the Scalapack project to provide portable communications utilities for the Scalapack routines.

References: [55]

7.2 BlockComm

Name: BlockComm

Description: A Package providing facilities for moving blocks of data around in a distributed memory parallel machine, without having to resort to low-level methods such as message passing. Blocks can be part of a regular structure (for example a regular grid, or a Fortran array), or a dynamic irregular structure (for example an irregular grid). Debugging facilities are also provided.

Systems: Available on workstation clusters (Sun, DEC, Silicon Graphics, IBM RS6000), IBM SP1, BBN TC-2000, Intel MPP machines, Convex, and Cray. Package is being ported to new parallel machines as they become available.

Contact: William Gropp,
Mathematics and Computer Science Division,
Argonne National Laboratory, USA.

Email: gropp @ mcs.anl.gov

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **pub/pdetools**

WWW: http://www.mcs.anl.gov/home/gropp/petsc.html

Comments: Part of the PETSc software toolkit for scientific computing. (See discussion at the beginning of section 2 for more information about PETSc).

References:

7.3 InterCom

Name: InterCom: Interprocessor Collective Communications Library

Description: Package provides high performance collective communications facilities, such as broadcast, scatter, gather, collect, and global combine.

Systems: Runs on Intel MPP machines, including iPSC/860, Delta, and Paragon.

Contact: Robert van de Geijn,
Department of Computer Sciences,
University of Texas at Austin, USA.

Email: intercom @ cs.utexas.edu

FTP: Available from Netlib in directory **intercom** (see section 11 for further information about netlib).

Comments: Package gives improved performance over many of the Intel NX routines, plus additional functionality.
A library is provided which automatically translates NX calls into InterCom calls.
Next release will include an MPI group collective communications interface.

References: [16]

8 Languages and Language Extensions (excluding HPF)

This section contains packages which provide non-HPF language extensions. These are mainly based on data-parallel programming models. Other language extensions which provide more 'novel' programming models can be found in section 10.

8.1 HPC

Name: HPC: High Performance C

Description: Data parallel extensions to C. Based on the High Performance Fortran standard (see section 9), but goes beyond HPF in a number of areas with features such as dynamic distributed data structures and dynamically allocatable irregular arrays.

Systems:

Contact: Vincent Van Dongen,
Centre de recherche informatique de Montréal (CRIM),
CANADA.

Email: vandonge @ crim.ca

FTP: Relevant papers available by anonymous ftp from **ftp.crim.ca** in directory **/apar/public**

WWW: http://www.crim.ca/Domaines_Services/APAR/eppp.html

Comments: HPC compiler is part of the EPPP (Environment for Portable Parallel Programming) project, which aims to develop the HPC language, compiler, simulator, and performance debugger.

References: [57, 56]

8.2 pC++

Name: pC++

Description: pC++ is an extension to C++ which gives a form of data parallelism. It contains many of the ideas found in HPF (High Performance Fortran). Compilers are available for a number of parallel machines, which take a pC++ program and convert it to a C++ program with machine specific calls.

Systems: Machines currently supported are CM5, paragon, KSR-1, Intel Paragon, IBM SP1, BBN TC2000, Sequent Symmetry and Networked workstations using PVM (see separate entry). Code can also be developed and tested in uniprocessor mode on most workstations.

Contact: Sage-team,
Center for Innovative Computer Applications,
Indiana University, USA.

Email: sage @ cica.indiana.edu

FTP: Available from **ftp.cs.indiana.edu** or **cica.cica.indiana.edu** in directory **/pub/sage**

WWW: http://www.cica.indiana.edu/sage

Comments: May be of interest to C++ programmers who are excited by HPF. Report [25] gives pC++ examples for a parallel tridiagonal matrix solver and a fast Poisson solver.

References: [25, 26, 98]

8.3 Split-C

Name:	Split-C
Description:	Parallel extensions to the C language. Aims to provide the most useful features from message passing, shared memory, and data-parallel programming models whilst making efficient use of the underlying hardware.
Systems:	Runs on the CM5. Requires Gnu CC, Gnu make, and CMOST 7.2 to install.
Contact:	The Split-C team, Computer Science Division, University of California at Berkeley.
Email:	split-c @ boing.cs.berkeley.edu for questions, split-c-bugs @ boing.cs.berkeley.edu for bug reports.
FTP:	Available from **ftp.cs.berkeley.edu** in directory **/ucb/CASTLE/Split-C**
WWW:	http://www.cs.berkeley.edu/public/parallel/split-c.html
Comments:	Sits on top of the CMAM active messages library (see separate entry).
References:	[50]

9 HPF Implementations

This section contains packages which implement some or all of the High Performance Fortran (HPF) standard. Also included are packages which provide data-parallel extensions to Fortran which are sufficiently similar to HPF.

The basic idea behind HPF is to provide a set of standard extensions to Fortran 90, allowing data-parallel programs to be written. HPF was developed by the High Performance Fortran Forum, a coalition of academic and industrial groups. The main aim was to produce a standard for data-parallel language extensions to Fortran which could achieve respectable performance on parallel machines.

The official HPF standard document is available from Netlib in the directory **hpf**

There is a World-Wide-Web page devoted to HPF. **http://www.erc.msstate.edu/hpff/home.html** this contains a postcript version of the HPF standard document, latest news on HPF implementations, papers related to HPF, and current research projects which are related to HPF.

In this survey, four HPF (or 'nearly HPF') compilers are included. These are:

- **xHPF** from Applied Parallel Research.
- **HPF Mapper** from NA Software.
- **ADAPTOR** from GMD.
- **PSI Compiler** from University of Missouri-Rolla.

See separate entries for more details of these packages.

9.1 ADAPTOR

Name: ADAPTOR: Automatic Data Parallelism Translator

Description: Adaptor transforms 'HPF like' data-parallel Fortran programs into Fortran with message passing calls. Includes array extensions, parallel loops, and layout directives.

Systems: A number of message passing harnesses are currently supported including versions of PVM greater than 3.0, P4, PARMACS 6.0, and the Argonne National Laboratory MPI test implementation (see separate entries for PVM, P4, PARMACS and the MPI test implementation).
The package can also run on a number of native message passing systems including CM-5, Intel Paragon, Intel iPSC, IBM SP1, KSR, SGI, Alliant, Meiko CS1 and CS2, and Parsytec GCel.

Contact: Thomas Brandes,
German National Research Center for Computer Science (GMD),
St. Augustin, Germany.

Email: brandes @ gmd.de

FTP: Available from **ftp.gmd.de** in directory **GMD/adaptor**
Documentation is in **GMD/adaptor/docs**

Comments: Comprehensive and high-quality documentation supplied, including installation guide, user guide, language reference manual, comparison of a number of HPF-like systems, and evaluation of HPF on real applications.
Available at Daresbury on the Intel MPP machine.

References: [31, 29, 30, 69, 28] (LT:Adaptor, 78)

9.2 HPF Mapper

Name: High Performance Fortran Mapper

Description: A pre-compiler which translates source code written using the HPF standard into Fortran-90 source with message passing calls.

Systems: Currently produces Fortran-90 code with PVM calls.

Contact: NA Software Limited, Roscoe House,
62 Roscoe Street, Liverpool, UK.

Email: n.a.software @ nasoftwr.demon.co.uk

FTP:

Comments: Currently internal alpha release.
Available at Daresbury on Intel MPP and the workstation clusters as soon as possible.

References:

9.3 PSI Compiler

Name: PSI Compiler: Portable, Scalable, Architecture Independent compiler

Description: A compiler which performs reductions on array expressions, based on the 'PSI Calculus' rules. Independent of any particular source or destination language. Current version can transform a subset of HPF into Fortran 90.

Systems: Networked SUN and SGI workstations using sockets, and CM5.

Contact: Lenore Mullin,
Department of Computer Science,
University of Missouri-Rolla, USA.

Email: psi @ cs.umr.edu

FTP: Available by anonymous ftp from **cs.umr.edu** in directory **pub**
Relevant Technical Reports in directory **pub/tech-reports**

WWW: http://www.cs.umr.edu/ psi/psi.html

Comments: Also available is a version which transforms MOAL (Mathematics of Arrays Language) into C.
A version which operates on Fortran 90 source is under development.

References: [100, 101, 102]

9.4 xHPF

Name: xHPF: High Performance Fortran

Description: Precompiler which takes an HPF subset program and produces parallel Fortran 77 code with calls to APR's parallel runtime library. Facilities are also provided for interactively reviewing parallelisation strategies, runtime parallel performance analysis, and automatic parallelisation of serial Fortran 77 codes into HPF.

Systems: APR's parallel runtime library acts as an interface to other common message passing libraries and runs on workstation clusters using PVM, Express or Linda, IBM SP1 using EUI, Intel Paragon using NT, Meiko using Parmacs, Cray T3D using PVM, nCUBE using native libraries, and CN-5 using native libraries.

Contact: Richard Friedman,
Applied Parallel Research,
550 Main Street, Suit 1,
Placerville, CA 95667, USA.

Email: rchrd @ netcom.com

FTP: Further information available by anonymous ftp from
ftp.netcom.com in directory **/pub/forge**

WWW: ftp://ftp.netcom.com/pub/forge/home.html

Comments: APR offer all their software via ftp for short term (15/30 day) evaluations.
The WWW and FTP servers contain detailed product information, HPF benchmarks, product user manuals, new product announcements, and much more.

References: [65],(LT:xHPF,150)

10 Alternative Programming Models

This section contains packages which provide alternatives to the message passing parallel programming model. These models may not have been widely used in scientific programming, but are felt to be of potential use. Indeed many of the packages here have been tested on example applications by their developers.

Examples of programming models which are appropriate for this section include active-messages, virtual shared memory, bulk synchronous programming, and object oriented approaches.

10.1 ADSMITH

Name: ADSMITH: A Distributed Shared Memory In Tsing Hua university

Description: A C package which provides a logical shared memory environment on a distributed memory machine.

Systems: Sits on top of PVM, version 3.3 or above.

Contact: William W.Y.Liang,
Parallel & Distributed System Lab,
Institute of Computer Science,
National Tsing Hua University, Hsin Chu, TW.

Email: wyliang @ solar.csie.ntu.edu.tw or apollo @ cs.nthu.edu.tw

FTP: Available from **140.114.78.191** in directory **pub**

Comments: Package includes a number of examples including a distributed matrix multiply.

References:

10.2 BEEBLEBROX

Name: BEEBLEBROX

Description: C++ class library for running divide-and-conquer (D&Q) type problems on a parallel machine. The parallel D&Q algorithms are all encapsulated in the class library, with the package user only needing to set up a class which specifies data flow between parent nodes and child nodes in the 'D&Q tree', and various other 'bits & pieces'.

Systems: Requires Cfront C++ compiler version 3.0.2 or greater. Requires LAM message passing harness version 5.1 or greater.

Contact: Andy Piper, Engineering Department,
Cambridge University, UK.

Email: ajp @ eng.cam.ac.uk

FTP: Available from **svr-ftp.eng.cam.ac.uk**
Package in directory **/pub/misc**
Documentation in files **/pub/reports/piper_***

Comments: Package includes a number of examples including matrix multiply, and a few sorting methods.
Zaphod Beeblebrox is a person with two heads [1].

References: [106]

10.3 Castle

Name: Castle

Description: Castle is a proposed parallel software project, taking place at the Computer Science Division of the University of California at Berkeley. The aim is to provide a parallel environment containing a number of facilities at different layers of abstraction (the 'floors' of the castle). Floor 1 - abstract computer incorporating 'active messages' and BLAS, floor 2 - physical multiprocessor containing shared memory and communication facilities, floor 3 - virtual multiprocessor containing distributed object and maths libraries, floor 4 containing high level languages including parallel C++ and HPF-like languages, top floor containing applications; a fluid dynamics code and global climate model code are under construction. Debugging, performance and optimising tools can access all levels ('Rapunzel's hair').

Systems:

Contact:

Email:

FTP: Currently implemented parts are available from **ftp.cs.berkeley.edu** in directory **/ucb/CASTLE**

WWW: http://www.cs.berkeley.edu/public/parallel/castle.html

Comments: Currently implemented are Active messages (see separate entry for CMAM) and a parallel extension to the C language (see separate entry for Split-C). See [81, 123, 49, 50, 122] for more details.

References: [81, 123, 49, 50, 122]

10.4 CHARM

Name: CHARM

Description: Machine independent programming system which provides high level mechanisms for developing (possibly complex) parallel applications. Consists of a small number of extensions to the C language.

Systems: Currently runs on Intel iPSC, NCUBE, Encore Multimax, Sequent Symmetry, Alliant FX/8, IBM SP-1, CM-5, Single processor UNIX workstations, and workstation clusters. It is currently being ported to Parsytec GCel, and Alliant FX/2800. Planned ports include KSR-1 and other parallel machines as they become available.

Address: Prof.L.V.Kale,
Parallel Programming Laboratory,
Department of Computer Science,
University of Illinois at Urbana-Champaign, USA.

Email: kale @ cs.uiuc.edu

FTP: Available from **a.cs.uiuc.edu**
Most recent version of package is in directory
pub/CHARM/CHARM.4.3
Brief overview of the system is given in **ABOUT_CHARM.ps**
Relevant papers in directory **pub/CHARM/papers**

WWW: http://charm.cs.uiuc.edu/

Comments: A number of related tools are available. **Charm++** is a C++ based version of CHARM. **Projections** is a performance visualisation tool, and **dagger** is a graphical tool which shows dependencies between messages and sub-computations.

References: [92, 59, 89, 110, 90, 91, 60]

10.5 CMAM

Name: CMAM: CM-5 Active Messages

Description: Provides active messages for the CM-5. Intended for use as a low-level layer in libraries and compilers.

Systems: Runs on the CM-5.

Contact: Thorsten von Eicken,
Computer Science Division,
University of California at Berkeley, USA.

Email: tve @ cs.berkeley.edu

FTP: Available from **ftp.cs.berkeley.edu** in directory **/ucb/CASTLE/Active_Messages**

WWW: http://www.cs.berkeley.edu/public/parallel/active_messages.html

Comments: The term active message refers to a message with an associated handler at the receiving end. When the message arrives, the computation on the node is interrupted and the handler is executed.

References: [122]

10.6 Concert

Name: Concert

Description: The Concert system is built around a fine-grained object-oriented language (Concurrent Aggregates language), which is based on the 'actor model'. Included in the distribution is a compiler, runtime system, libraries, emulator, debugger, and performance tools.

Systems: Distribution includes runtime systems for Sun Workstations and the CM5.

Contact: Concurrent Systems Architecture Group,
Department of Computer Science,
University of Illinois.

Email: concert @ red-herring.cs.uiuc.edu

FTP:

WWW: The Concert package, documentation and description of the Concurrent Aggregates Language are available from **http://www-csag.cs.uiuc.edu**

Comments: A number of application programs have been ported to the Concert system, including Molecular Dynamics, Computational Fluid Dynamics, Adaptive Mesh Refinement, Particle-in-Cell, and irregular N-body problem. Some of these are included in the release.

References: [40, 43, 42, 41]

10.7 Domain Objects

Name: Domain Objects

Description: A library written in C++ allowing the creation and use of 'domain objects' which represent an application domain. The underlying details of the mapping of the domain onto the parallel machine are hidden from the user.

Systems:

Contact: Ian G. Angus,
Northrop Research and Technology Center,
One Research Park,
Palos Verdes Peninsula, CA 90274, USA.

Email: iangus @ nrtc.northrop.com

FTP:

Comments:

References: [6]

10.8 DoPVM

Name: DoPVM: Distributed Object Parallel Virtual Machine

Description: A 'shared object toolkit' written in C++. Provides facilities for constructing objects which are shared across a distributed computing platform, and mechanisms for partitioning, scheduling, and synchronisation.

Systems: Sits on top of PVM.

Contact: Charles Hartley & V.S.Sunderam
Department of Math and Computer Science,
Emory University, USA.

Email: vss @ mathcs.emory.edu

FTP: Report describing the package is available from
emory.mathcs.emory.edu in directory **/pub/vss**

Comments:

References: [78] (LT:DoPVM, 94)

10.9 EPEE

Name: EPEE: Eiffel Parallel Execution Environment

Description: Data parallel and Control parallelism programming models encapsulated in classes of the Eiffel Object Oriented Language.

Systems: Prototype available for Intel iPSC/2, iPSC/860, Paragon XP/S, and Networked Workstations on top of TCP/IP.

Contact: Thierry Priol,
Institut de Recherche en Informatique et Systèmes Aléatoires,
Rennes, France.

Email: priol @ irisa.fr

FTP: Package and reports available by anonymous ftp from
ftp.irisa.fr in directory **local/KOAN**

Comments: Intel versions sit on top of the KOAN Virtual Shared Memory package (see separate entry).

References: [77]

10.10 KOAN/MYOAN

Name: KOAN/MYOAN

Description: Virtual Shared Memory system for Intel machines. Contains a number of features to improve performance including page broadcasting, page locking, and a mechanism to avoid double faults.

Systems: KOAN runs on the Intel iPSC2. MYOAN is an implementation of KOAN for the Intel Paragon.

Contact: Thierry Priol,
Institut de Recherche en Informatique et Systèmes Aléatoires,
Rennes, France.

Email: priol @ irisa.fr

FTP: Package and reports available by anonymous ftp from n
ftp.irisa.fr in directory **local/KOAN**

Comments:

References: [17, 93, 38, 76, 108, 107, 27]

10.11 NESL

Name: NESL: Nested Data-Parallel Language

Description: Interactive parallel programming environment based around the NESL language, which is a fine-grained, functional, nested data-parallel language. Also included are a graphics library, profiling and tracing facilities, and on-line documentation.

Systems: Connection Machine CM-2 or CM-5, Cray Y-MP, serial workstations.
Requires an ANSI C compiler, yacc, lex, Common Lisp, and an X11 library for the graphics routines.

Contact: Guy Blelloch,
School of Computer Science,
Carnegie Mellon University, USA.

Email: There is a NESL mailing list in operation. To join email nesl-request @ cs.cmu.edu

FTP: Available by anonymous ftp from **nesl.scandal.cs.cmu.edu** in directory **code/nesl**

WWW: http://www.cs.cmu.edu:8001/Web/Groups/scandal/www/home.html

Comments: The package comes with a large number of example programs including sorting algorithms, conjugate gradients, adaptive numerical integration,

References: [24, 23]

10.12 Oxford BSP Library

Name: Oxford Bulk Synchronous Parallel Library

Description: An implementation of the BSP model proposed by Valiant [119]. Provides subroutines to create processes, access remote data and bulk synchronisation.

Systems: Can be linked to Fortran, C and Pascal programs. Prototype version of the library has been implemented using the PVM message passing library. It has also been implemented directly on a transputer network and a Caplin i860 network. A version for shared memory systems is under development.

Contact: Joy Reed, Oxford Parallel, OUCL Wolfson House,
Parks Road, Oxford, OX1 3QD, UK.

Email: Joy.Reed @ comlab.ox.ac.uk

FTP: Available by anonymous ftp from **ftp.comlab.ox.ac.uk** in directory **/pub/Packages/BSP**

WWW: http://www.comlab.ox.ac.uk/oucl/oxpara/bsplib.html

Comments:

References: [99]

10.13 P4-LINDA

Name: P4-LINDA

Description: Package of C routines which provide some of the core functions of the Linda programming model as described in [68].

Systems: Sits on top of the P4 communications harness (see separate entry). Can make use of the message passing facilities of P4 on a distributed memory machine, or the monitor facilities of P4 on a shared memory machine.

Contact: Ralph Butler and Alan Leveton,
College of Comp. and Inf. Science,
University of North Florida, USA.

Ewing Lusk,
Mathematics and Computational Science Division,
Argonne National Laboratory, USA.

Email: rbutler @ sinkhole.unf.edu or lusk @ mcs.anl.gov

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **pub/p4**

Comments:

References: [35]

10.14 PCN

Name: PCN: Program Composition Notation

Description: PCN is a programming system based around the concept of 'program composition' operators which can be used to combine simple modules to produce (possibly complex) parallel programs.

Systems: Intel iPSC/860 and Delta, Silicon Graphics Iris, NeXT, IBM RS6000, and Sun 4.

Contact: Ian Foster, Robert Olson, Steven Tuecke,
Mathematics and Computer Science Division,
Argonne National Laboratory, USA.

Email:

FTP: Available by anonymous ftp from **info.mcs.anl.gov** in directory **pub/pcn**

Comments: As well as the PCN language itself, the system includes a symbolic debugger, execution profiler, standard UNIX libraries, and facilities to seamlessly link with existing Fortran or C code. It can also produce trace output which can be viewed by the UPSHOT package (see separate entry).
A number of substantial applications have been programmed using PCN, including a geophysical modelling code and a fluid dynamics code. These are described further in [62].
User Manual [64] is very comprehensive and well written.

References: [64, 62] (LT:PCN, 127)

10.15 Strand^{88}

Name:	Strand^{88}
Description:	A parallel development system centred around the Strand language. The language contains some of the key features of concurrent logic languages whilst avoiding unneccesarily complex features, in order to permit efficient and portable implementation.
Systems:	Runs on Sun workstations, Sequent Symmetry, Intel iPSC/2, Encore Multimax, MIPS RISComputer, Cogent XTM, and Transputer systems.
Contact:	Strand Software Technologies, Greycaine Road, Watford, Hertfordshire, WD2 4JP, UK.
Email:	strand88 @ ail.co.uk
FTP:	
Comments:	A 'foreign language' interface allows chunks of Fortran and C sequential code to be run in parallel, managed by a STRAND harness.
References:	[63, 7]

11 Other sources of information

11.1 Other Surveys

A very comprehensive source of information is the document '*A Survey of Software Environments for Exploiting Networked Computing Resources*' by Louis H. Turcotte. This surveys software packages and tools for use on workstation clusters, although many of the packages are also applicable to true parallel machines. Software packages described here which are also in the survey by Turcotte have been given extra reference information to allow cross referencing with this document, including name of main entry and entry in the bibliography.

The survey by Turcotte is available for anonymous ftp from **unix.hensa.ac.uk** in file **/pub/parallel/papers/surveys/soft-env-net-report.ps.gz**

11.2 Netlib

A number of the packages described in this document are available from Netlib. This is a large software repository which contains numerical software and tools, including a large amount of parallel software. Software on Netlib can be accessed in a number of different ways:

1. By anonymous ftp from **netlib2.cs.utk.edu**
   The entire contents of Netlib are mirrored in the **/netlib** directory on Hensa (**unix.hensa.ac.uk**); This may be more convenient to use from the UK.

2. By World-wide-web with url **http://netlib2.cs.utk.edu/index.html**
   The mirror of Netlib on Hensa can also be accessed via WWW gs with URL **ftp://www.hensa.ac.uk/ftp/pub/netlib/master/index.html.Z**

3. By email with a message of the form 'send index from pvm3' sent to netlib@ornl.gov . The general form is 'send *file* from *library*'. To get a list of all the libraries on netlib use 'send index'.

4. Using the Xnetlib package which allows interactive browsing of the Netlib directories. This can be obtained from Netlib via anonymous ftp in directory xnetlib, or via email with the message 'send xnetlib.shar from xnetlib' to netlib @ ornl.gov .

11.3 Hensa

The HENSA (Higher Education National Software Archives) ftp site in the UK holds the 'Transputer, occam and parallel computing archive' in directory /parallel. This archive contains a huge amount of information about almost every aspect of parallel computing. Information available includes:

- Archived newsgroup articles.

- Copies of the MPI and HPF standards.
- Details of conferences.
- Contents listings for Journals.
- Book lists and descriptions.
- Parallel research papers.
- Pointers to other WWW and FTP sites.

As previously mentioned, this site also has a mirror of the entire Netlib repository in directory /netlib.

The site can be accessed by anonymous ftp to **unix.hensa.ac.uk**

It can also be accessed via the World-Wide-Web with URL **ftp://www.hensa.ac.uk**

References

[1] D Adams. *The Hitchikers Guide to the Galaxy*. Pan Books Ltd., 1979. The 'Douglas Adams Worship Page' on the World-Wide-Web can be accessed at http://www.umd.umich.edu/ nhughes/dna/.

[2] C Addison, B Beattie, N Brown, R Cook, B Stevens, and D Watson. Distributed objects: sequential objects with parallel performance. In *Proceedings of the 6th SIAM Conference on Parallel Processing for Scientific Computing*, 1993.

[3] R Alasdair, A Bruce, J G Mills, and G Smith. Chimp/mpi user guide. Technical Report EPCC-KTP-CHIMP-V2-USER.1.2, Edinburgh Parallel Computing Centre, 1994.

[4] R J Allan. Fortnet v4.0: The parallel programming software. Technical report, Daresbury Laboratory, June 1992.

[5] R.J. Allan. *PARLANCE v2.0 User Manual: Numerical Algorithms in a Virtual Shared Data Space*. Daresbury Laboratory, Warrington, WA4 4AD, U.K., 1992.

[6] I G Angus. Domain objects: A new approach to parallel programming. Technical report, Northrop Research and Technology Center, undated.

[7] Artificial Intelligence Limited. *Harnessing Applications Software for Parallel Execution with STRAND*^{88}, January 1990.

[8] S B Baden and S Kohn. The reference guide to genmp - the generic multiprocessor. Technical Report CS92-243, University of California at San Diego, June 1992. Available by anonymous ftp from cs.ucsd.edu in directory pub/baden/genmp.

[9] S B Baden and S R Kohn. A robust parallel programming model for dynamic non-uniform scientific computations. Technical report, University of California at San Diego, 1994. Available by anonymous ftp from ftp.sdsc.edu in directory pub/sdsc/parallel/lparx.

[10] S B Baden and S R Kohn. Portable parallel programming under the lpar system. Technical report, University of California at San Diego, undated. Draft. Available by anonymous ftp from ftp.sdsc.edu in directory pub/sdsc/parallel/lparx.

[11] Z Bai. Error analysis of the lanczos algorithm for the nonsymmetric eigenvalue problem. *Math. Comp.*, April 1992. Available by anonymous ftp from ftp.super.org in file pub/prism/wn2.ps.

[12] Z Bai. Progress in the numerical solution of the nonsymmetric eigenvalue problem. *JNLAA*, October 1992. Available by anonymous ftp from ftp.super.org in file pub/prism/wn4.ps.

[13] Z Bai and J Demmel. Design of a parallel nonsymmetric eigenroutine toolbox (part i). Research Report 92-09, University of Kentucky, December 1992. Available by anonymous ftp from ftp.super.org in file pub/prism/wn5.ps.

[14] D Balsara, M Lemke, and D Quinlan. Amr++, a parallel adaptive mesh refinement object class library for fluid flow problems. In *Proceedings of Symposium on Adaptive, Multilevel and Hierarchical Strategies, ASME Winter Annual Meeting, Anaheim*, November 8-13 1992.

[15] P V Bangalore, A Skjellum, C Baldwin, and S G Smith. Dense and iterative concurrent linear algebra in the multicomputer toolbox. Technical report, Mississippi State University, 1993. Available by anonymous ftp from aurora.cs.msstate.edu in directory pub/reports/SPLC93.

[16] M Barnett, L Shuler, R van de Geijn, S Gupta, D G Payne, and J Watts. Interprocessor collective communication library (intercom). In *Proceedings of the Scalable High-Performance Computing Conference*, May 1994. Available by anonymous ftp from Netlib in directory intercom.

[17] R Berrendorf, M Gerndt, Z Lahjomri, and T Priol. A comparison of shared virtual memory and message passing programming techniques based on a finite element application. In *Proceedings of CONPAR'94-VAPP VI, Linz (Austria)*, September 1994. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[18] C Bischod and X Sun. A framework for symmetric band reduction and tridiagonalization. Preprint MCS P298-0392, Argonne National Laboratory, July 1992. Available by anonymous ftp from ftp.super.org in file pub/prism/wn3.ps.

[19] C Bischof, S Huss-Lederman, E Jacobson, X Sun, and A Tsao. On the impact of hpf data layout on the design of efficient and maintainable parallel linear algebra libraries, March 1993. (A description of virtual 2D torus wrap, submitted as comments to the HPF working group), Available by anonymous ftp from ftp.super.org in file pub/prism/wn10.ps.

[20] C Bischof, S Huss-Lederman, X Sun, and A Tsao. The prism project: Infrastructure and algorithms for parallel eigensolvers. In *Proceedings of the Scalable Parallel Libraries Conference*, October 1993. Available by anonymous ftp from ftp.super.org in file pub/prism/wn12.ps.

[21] C Bischof, C M Marques, and X Sun. Parallel bandreduction and tridiagonalization. In *Proceedings of the sixth SIAM conference on Parallel Processing for Scientific Computation*, March 1993. Available by anonymous ftp from ftp.super.org in file pub/prism/wn8.ps.

[22] C Bischof and X Sun. A divide-and-conquer method for computing complementary invariant subspaces of symmetric matrices. Technical Report MCS-P286-0192, Argonne National Laboratory, March 1992. Available by anonymous ftp from ftp.super.org in file pub/prism/wn1.ps.

[23] G E Blelloch. Nesl: A nested data-parallel language (version 3.0). Technical report, Carnegie Mellon University, 1994. Available by anonymous ftp from nesl.scandal.cs.cmu.edu in directory code/nesl/nesl/doc.

[24] G E Blelloch, J C Hardwick, and M Zagha. Nesl user's manual (for nesl version 3.0). Technical report, Carnegie Mellon University, 1994. Available by anonymous ftp from nesl.scandal.cs.cmu.edu in directory code/nesl/nesl/doc.

[25] F Bodin, P Beckman, D Gannon, S Narayana, and S X Yang. Distributed pc++: Basic ideas for an object parallel language. *Scientific Programming*, 2(3), Fall 1993. Available by anonymous ftp from ftp.cs.indiana.edu in directory /pub/sage.

[26] F Bodin, P Beckman, D Gannon, S Yang, S Kesavan, A Malony, and B Mohr. Implementing a parallel c++ runtime system for scalable parallel systems. In *Proceedings of the Supercomputing '93 Conference, Portland, Oregon*, November 1993. Available by anonymous ftp from ftp.cs.indiana.edu in directory /pub/sage.

[27] K Bouatouch, D Menard, and T Priol. Parallel radiosity using a shared virtual memory. In *Proceedings of the First Bilkent Computer Graphics Conference on Advanced Techniques in Animation, Rendering and Visualization, Ankara (Turkey)*, July 1993. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[28] T Brandes. Evaluation of high performance fortran on some real applications. Technical report, German National Research Center for Computer Science (GMD), 1993.

[29] T Brandes. Adaptor: Installation guide, version 2.0 (march 1994). Report Adaptor 1, German National Research Center for Computer Science (GMD), March 28, 1994.

[30] T Brandes. Adaptor: Language reference manual, version 2.0 (march 1994). Report Adaptor 3, German National Research Center for Computer Science (GMD), March 28, 1994.

[31] T Brandes. Adaptor: Users guide, version 2.0 (march 1994). Report Adaptor 2, German National Research Center for Computer Science (GMD), March 28, 1994.

[32] G Burns, R Daoud, and J Vaigl. Lam: An open cluster environment for mpi. Technical report, Ohio Supercomputer Center, undated.

[33] G Burns, V Radiya, R Daoud, and R Machiraju. All about trollius. *Occam User's Group Newsletter*, 1990.

[34] G D Burns. A local area multicomputer. In *Proceedings of the Fourth Conference on Hypercubes, Concurrent Computers and Applications*, 1989.

[35] R M Butler, A L Leveton, and E L Lusk. p4-linda: A portable implementation of linda. Technical report, Argonne National Laboratory, undated. Available by anonymous ftp from info.mcs.anl.gov in directory pub/p4.

[36] R M Butler and E L Lusk. User's guide to the p4 parallel programming system. Technical Report ANL-92/17, Argonne National Laboratory, October 1992. Available by anonymous ftp from Netlib in directory p4.

[37] R M Butler and E L Lusk. Monitors, messages, and clusters: the p4 parallel programming system. *Parallel Computing*, 20(4):547–564, 1994. Available by anonymous ftp from Netlib in directory p4.

[38] G Cabillic, T Priol, and I Puaut. Myoan: an implementation of the koan shared virtual memory on the intel paragon. Publication Interne 812, Institut de recherche en informatique et systèmes aléatoires, April 1994. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[39] R Calkin, R Hempel, H-C Hoppe, and P Wypior. Portable programming with the parmacs message passing system. *Parallel Computing*, 20(4):615–632, 1994.

[40] A A Chien. *Concurrent Aggregates: Supporting Modularity in Massively-Parallel Programs.* MIT Press, Cambridge, MA, 1993.

[41] A A Chien and V Karamcheti. Concert - efficient runtime support for concurrent object-oriented programming languages on stock hardware. In *Proceedings of SUPERCOMPUTING*, November 1993.

[42] A A Chien, V Karamcheti, and J Plevyak. The concert system - compiler and runtime support for efficient fine-grained concurrent object-oriented programs. Technical report UIUCDCS-R-93-1815, Department of Computer Science, University of Illinois, June 1993.

[43] A A Chien, V Karamcheti, J Plevyak, and X Zhang. Concurrent aggregates langauge report 2.0, September 1993.

[44] J Choi, J Dongarra, and D W Walker. Pb-blas: A set of parallel block basic linear algebra subprograms. Technical report ORNL/TM-12468, Oak Ridge National Laboratory, March 1994. Available by anonymous ftp from Netlib in directory scalapack.

[45] J Choi, J Dongarra, and D W Walker. Pb-blas reference manual (version 1.0beta (march 15, 1994)). Technical report ORNL/TM-12469, Oak Ridge National Laboratory, March 1994. Available by anonymous ftp from Netlib in directory scalapack.

[46] J Choi, J Dongarra, D W Walker, and R C Whaley. Scalapack reference manual: Parallel factorization routines (lu, qr, and cholesky) and parallel reduction routines (hrd, brd, and trd) (version 1.0beta (december 31, 1993)). Technical Report ORNL/TM-12470, Oak Ridge National Laboratory, April 1994. Available by anonymous ftp from Netlib in directory scalapack.

[47] J Choi, J J Dongarra, A Petitet, and D W Walker. Pumma reference manual (version 2.0). Technical report ORNL/TM-12494, Oak Ridge National Laboratory, March 1994. Available by anonymous ftp from Netlib in directory scalapack.

[48] J Choi, J J Dongarra, and D W Walker. Pumma: Parallel universal matrix multiplication algorithms on distributed memory concurrent computers. Technical report ORNL/TM-12252, Oak Ridge National Laboratory, August 1993. Available by anonymous ftp from Netlib in directory scalapack.

[49] W Crutchfield. Load balancing of an irregular algorithm. Technical report, Lawrence Livermore National Laboratory, June 1991.

[50] D E Culler, A Dusseau, S Goldstein, A Krishnamurthy, S Lumetta, T von Eicken, and K Yelick. Parallel programming in split-c. In *Proceedings Supercomputing*, 1993. Available by anonymous ftp from ftp.cs.berkeley.edu in directory /ucb/castle/Split-C.

[51] R D da Cunha and T Hopkins. Pim 1.0 the parallel iterative methods package for systems of linear equations users guide. Technical Report 2-94, Computing Laboratory, University of Kent at Canterbury, 1994. Available by anonymous ftp from unix.hensa.ac.uk in file /misc/ukc.reports/comp.sci/reports/2-94.Z.

[52] J Dongarra, E Greaser, R Pozo, and D Walker. Lapack++ v.0.9 release notes and installation manual. Technical report, The ScaLapack group, undated.

[53] J Dongarra, S A Moulton, L S Ostrouchov, A Petitet, and R C Whaley. Lapack working note ??: Installation guide for scalapack. Technical report, University of Tennessee, December 1993. Available by anonymous ftp from Netlib in directory scalapack.

[54] J Dongarra, R Pozo, and D Walker. Lapack++: A design overview of object-oriented extensions for high performance linear algebra. Technical report, The ScaLapack group, undated.

[55] J J Dongarra, R A van de Geijn, and R Clint Whaley. A users' guide to the blacs. Technical report, University of Tennessee, February 1993. Available by anonymous ftp from Netlib in directory pvm3.

[56] V Van Dongen, C Bonello, and C Freehill. Data parallelism with high performance c. In *Supercomputing Symposium 94, Canada's 8th annual High Performance Computing conference*, June 6-8 1994.

[57] V Van Dongen, C Bonello, and C Freehill. High performance c language specification. version 0.8.9. Rapport technique CRIM–EPPP-94/04-12, Centre de recherche informatique de Montréal, April 1994. Available by anonymous ftp from ftp.crim.ca in directory /apar/public.

[58] V Eijkhout. Distributed iterative linear system solvers using pvm: User manual. Technical report, University of Tennessee, July 1994. Available by anonymous ftp from cs.utk.edu in directory pub/eijkhout/code.

[59] W Fenton, B Ramkumar, V Saletore, A Sinha, and L V Kale. The chare kernel programming language manual. Parallel Programming Laboratory Report 89-17, University of Illinois, 1989.

[60] W Fenton, B Ramkumar, V Saletore, A B Sinha, and L V Kale. Supporting machine independent parallel programming on diverse architectures. In *Proceedings of the International Conference on Parallel processing*, August 1991.

[61] J Flower and A Kolawa. Express is not just a message passing system: Current and future directions in express. *Parallel Computing*, 20(4), 1994.

[62] I Foster, R Olson, and S Tuecke. Productive parallel programming: The pcn approach. *Scientific Programming*, 1:51–66, 1992. Available by anonymous ftp from info.mcs.anl.gov in directory pub/pcn.

[63] I Foster and S Taylor. *STRAND: New Concepts in Parallel Programming.* Prentice-Hall, 1989.

[64] I Foster and S Tuecke. Parallel programming with pcn. Technical report, Argonne National Laboratory, January 1993. Available by anonymous ftp from info.mcs.anl.gov in directory pub/pcn.

[65] R Friedman, R Sawdayi, D Goodrow, and J Levesque. *xHPF Version 1.0: User's Guide, Preliminary Edition.* Applied Parallel Research, December 1993. Available by anonymous ftp from ftp.netcom.com in directory /pub/forge.

[66] A Geist, A Beguelin, J Dongarra, W Jiang, R Manchek, and V Sunderam. Pvm 3 user's guide and reference manual. Technical Report ORNL/TM-12187, Oak Ridge National Laboratory, May 1994. Available by anonymous ftp from Netlib in directory pvm3.

[67] G A Geist, M T Heath, B W Peyton, and P H Worley. A users guide to picl: A portable instrumented communication library. Technical Report ORNL/TM-11616, Oak Ridge National Laboratory, October 1990. Available by anonymous ftp from Netlib in directory picl.

[68] D Gelernter. Generative communication in linda. *ACM Transactions on Programming Languages and Systems*, 7(1), 1985.

[69] V Getov, T Brandes, B Chapman, A Dunlop, T Hey, and D Pritchard. Comparison of hpf-like systems. PPPE Deliverable D4.3a, University of Southampton, GMD, University of Vienna, November 22, 1993.

[70] D Greco. Parlib++. Technical report, Parallel Computing Group, Centre for Advanced Studies and Development in Sardinia, 1993.

[71] W Gropp and B Smith. The design of data-structure-neutral libraries for the iterative solution of sparse linear systems. Technical Report PREPRINT MCS-P356-039, Argonne National Laboratory, March 1993. Available by anonymous ftp from info.mcs.anl.gov in directory pub/pdetools.

[72] W Gropp and B Smith. Simplified linear equation solvers users manual. Technical Report ANL-93/8-REV 1, Argonne National Laboratory, June 1993. Available by anonymous ftp from info.mcs.anl.gov in directory pub/pdetools.

[73] W Gropp and B Smith. Users manual for ksp: Data-structure-neutral codes implementing krylov space methods. Technical Report ANL-93/30, Argonne National Laboratory, August 1993. Available by anonymous ftp from info.mcs.anl.gov in directory pub/pdetools.

[74] W Gropp and B Smith. Users manual for the chameleon parallel programming tools. Technical Report ANL-93/23, Argonne National Laboratory, June 1993.

[75] W Gropp and B Smith. Scalable, extensible, and portable numerical libraries. Technical report, Argonne National Laboratory, undated.

[76] M Hahad, J Erhel, and T Priol. Scheduling strategies for sparse cholesky factorization on a shared virtual memory parallel computer. In *Proceedings of the 1994 International Conference on Parallel Processing, St Charles-Illinois*, 1994. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[77] F Hamelin, J M Jézéquel, and T Priol. A multi-paradigm object oriented parallel environment. In *Proceedings of IPPS'94, Cancun (Mexico)*, April 1994. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[78] C Hartley and V S Sunderam. Concurrent programming with shared objects in networked environments. Technical report, Department of Math & Computer Science, Emory University, 1993.

[79] M T Heath and J E Finger. *ParaGraph: A tool for visualizing performance of parallel programs.* University of Illinois, Oak Ridge National Laboratory, July 1994. Available by anonymous ftp from Netlib in directory paragraph.

[80] V Herrarte and E Lusk. Studying parallel program behaviour with upshot. Technical Report ANL–91/15, Argonne National Laboratory, 1991. Available by anonymous ftp from info.mcs.anl.gov in directory p4.

[81] P Hilfinger and P Colella. Fidil: a language for scientific programming. In *Frontiers in Applied Mathematics*, chapter 5, pages 97–138. SIAM, 1989.

[82] S Huss-Lederman, E Jacobson, and A Tsao. Comparison of scalable parallel matrix multiplication libraries. In *Proceedings of the Scalable Parallel Libraries Conference*, October 1993. Available by anonymous ftp from ftp.super.org in file pub/prism/wn13.ps.

[83] S Huss-Lederman, E Jacobson, A Tsao, and G Zhang. Matrix multiplication on the intel touchstone delta. Technical Report SRC-TR-093-101, Supercomputing Research Center, May 1993. Available by anonymous ftp from ftp.super.org in file pub/prism/wn11.ps.

[84] S Huss-Lederman, E Jacobson, A Tsao, and G Zhang. Matrix multiplication on the intel touchstone delta. Technical Report SRC-TR-093-101 (Revised), Supercomputing Research Center, Feb 1994. Expanded version of paper which supercedes that in file pub/prism/wn11.ps. Available by anonymous ftp from ftp.super.org in file pub/prism/wn14.ps.

[85] S E Huss-Lederman, S E Jacobson, A Tsao, and G Zhang. Matrix multiplication on the intel touchstone delta. In *Proceedings of the sixth SIAM conference on Parallel Processing for Scientific Computation*, March 1993. Available by anonymous ftp from ftp.super.org in file pub/prism/wn7.ps.

[86] S E Huss-Lederman, A Tsao, and G Zhang. A parallel implementation of the invariant subspace decomposition algorithm for dense symmetric matrices. In *Proceedings of the sixth SIAM conference on Parallel Processing for Scientific Computation*, March 1993. Available by anonymous ftp from ftp.super.org in file pub/prism/wn9.ps.

[87] Institute for Advanced Scientific Computation, University of Liverpool. *A users' guide to the DDL: Distributed Data Library*, May 1994.

[88] M T Jones and P E Plassmann. Blocksolve v1.1: Scalable library software for the parallel solution of sparse linear systems. ANL Report 92/46, Argonne National Laboratory, December 1992. Available by anonymous ftp from info.mcs.anl.gov in directory pub/BlockSolve.

[89] L V Kale. Parallel programming with the chare kernel: A simple case study. Parallel Programming Laboratory Report 90-12, University of Illinois, 1989.

[90] L V Kale. The chare kernel parallel programming language and system. In *Proceedings of the International Conference on Parallel processing*, pages 17–25, August 1990.

[91] L V Kale. A tutorial introduction to charm. Parallel Programming Laboratory Report 92-6, University of Illinois, 1992.

[92] L V Kale and W Shu. The chare kernel language for parallel programming: A perspective. Technical Report UIUCDCS-R-88-1451, University of Illinois, August 1988.

[93] Z Lahjomri and T Priol. Koan reference manual 2.1. Technical report, IRISA-INRIA Rennes, May 1993. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[94] M Lemke. *Multilevel Verfahren mit selbst-adaptiven Gitterverfeinerungen für Parallelrechner mit verteiltem Speicher.* PhD thesis, Ph.D. Dissertation, Heinrich Heine Universität, Düsseldorf, 1993.

[95] M Lemke, K Witsch, and D Quinlan. An object-oriented approach for parallel self adaptive mesh refinement on block structured grids. In *Proceedings of the 9th GAMM-Seminar Kiel, Notes on Numerical Fluid Mechanics*, 1993.

[96] M Lemke, K Witsch, and D Quinlan. An object-oriented approach for parallel self adaptive mesh refinement on block structured grids. In *Proceedings of the 9th GAMM-Seminar Kiel, Notes on Numerical Fluid Mechanics*, 1993.

[97] M Lin, J Hsieh, D H C Du, J P Thomas, and J A MacDonald. Distributed network computing over local atm networks. Technical Report TR-94-17, University of Minnesota, undated. Available by anonymous ftp from ftp.cs.umn.edu in directory /users/du/papers.

[98] A Malony, B Mohr, P Beckman, S Yang, and F Bodin. Performance analysis of pc++: A portable data-parallel programming system for scalable parallel computers. In *Proceedings of the 8th International Parallel Processing Symposium (IPPS), Cancun, Mexico*, April 1994. Available by anonymous ftp from ftp.cs.indiana.edu in directory /pub/sage.

[99] R Miller and J Reed. *The Oxford BSP Library: Users Guide.* Oxford Parallel, version 1.0 edition, 1993.

[100] L Mullin, C Chang, S Huang, M Mayer, N Nemer, and C Ramakrishna. Intermediate code generation for portable, scalable, compilers. architecture independent data parallelism: The preliminaries. Technical Report CSC-93-01, Department of Computer Science, University of Missouri-Rolla, 1993. Available by anonymous ftp from cs.umr.edu in file pub/tech-reports/93-01.ps.

[101] L Mullin and M Jenkins. Effective data parallel computation using the psi compiler. Technical Report CSC-94-02, Department of Computer Science, University of Missouri-Rolla, 1994. Available by anonymous ftp from cs.umr.edu in file pub/tech-reports/94-02.ps.

[102] L Mullin and S Thibault. A reduction semantics for array expressions: The psi compiler. Technical Report CSC-94-05, Department of Computer Science, University of Missouri-Rolla, 1994. Available by anonymous ftp from cs.umr.edu in file pub/tech-reports/94-05.ps.

[103] Ohio Supercomputer Center. *LAM for C Programmers*, May 1994.

[104] Ohio Supercomputer Center. *LAM for Fortran Programmers*, May 1994.

[105] Ohio Supercomputer Center. *LAM Installation Guide*, January 1994.

[106] A J Piper. Generalized parallel programming with divide-and-conquer: The beeblebrox system. Technical report CUED/F-INFENG/TR 132, Cambridge University Engineering Department, July 1993. Available by anonymous ftp from svr-ftp.eng.cam.ac.uk in file /pub/reports/piper_tr132.ps.

[107] T Priol and Z Lahjomri. Koan: a shared virtual memory for the ipsc/2 hypercube. Technical Report 597, Institut de recherche en informatique et systèmes aléatoires, July 1991. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[108] T Priol and Z Lahjomri. Trade-offs between shared virtual memory and message passing on an ipsc/2 hypercube. Technical Report 637, Institut de recherche en informatique et systèmes aléatoires, February 1992. Available by anonymous ftp from ftp.irisa.fr in directory local/KOAN.

[109] P Raghavan. Capss: A cartesian parallel sparse solver. Technical report, National Center for Supercomputing Applications, University of Illinois, November 1993. Available by anonymous ftp from Netlib in directory scalapack.

[110] W W Shu and L V Kale. Chare kernel - a runtime support system for parallel computations. *Journal of Parallel and Distributed Computing*, 11:198–211, 1990.

[111] A Skjellum. The multicomputer toolbox: Current and future directions. Technical report, Mississippi State University, 1993. Available by anonymous ftp from aurora.cs.msstate.edu in directory pub/reports/SPLC93.

[112] A Skjellum, N E Doss, and P V Bangalore. Writing libraries in mpi. In A Skjellum and D S Reese, editors, *Proceedings of the Scalable Parallel Libraries conference.* IEEE Computer Society Press, October 1993. Available by anonymous ftp from aurora.cs.msstate.edu in directory pub/reports/SPLC93.

[113] A Skjellum, S G Smith, N E Doss, and A P Leung a M Morari. The design and evolution of zipcode. *Parallel Computing*, 20(4):565–596, 1994. Available by anonymous ftp from ftp.sdsc.edu in directory pub/sdsc/parallel/lparx.

[114] D Sorensen and R Lehoucq. User's guide for arpack on delta touchstone. Technical report, Rice University, undated. Draft. Available by anonymous ftp from Netlib in directory scalapack.

[115] G Stellner, S Lamberts, and T Ludwig. *NXLib Users Guide Version1_1_2.* Technische Universitat Munchen, May 1994.

[116] V S Sunderam, G A Geist, J Dongarra, and R Manchek. The pvm concurrent computing system: Evolution, experiences, and trends. *Parallel Computing*, 20(4):531–545, 1994.

[117] B Topol and V Sunderam. Pgpvm: Performance visualization support for pvm. Technical report, Emory University, August 1994. Available by anonymous ftp from mathcs.emory.edu in directory pub/topol.

[118] A Tsao and T Turnbull. A comparison of algorithms for banded matrix multiplication. Technical Report SRC-TR-093-092, Supercomputing Research Center, 1993. Available by anonymous ftp from ftp.super.org in file pub/prism/wn6.ps.

[119] L Valiant. A bridging model for parallel computation. *Communications of the ACM*, 33(8), August 1990.

[120] P L Vaughan, A Skjellum, D S Reese, and F Cheng. Migrating from pvm to mpi, part i: The unify system. Technical report, Mississippi State University, July 1994. Available by anonymous ftp from aurora.cs.msstate.edu in directory pub/reports.

[121] K Viswanathan, N Doss, and A Skjellum. Zipcode: A portable, extensible communications layer, version 1.00. Technical report, Mississippi State University, July 1993. Available by anonymous ftp from aurora.cs.msstate.edu in directory pub/toolbox.

[122] T von Eicken, D Culler, S Goldstein, and K Schauser. Active messages: a mechanism for integrated communication and computation. In *Proceedings of the 19th International Symposium on Computer Architecture, Gold Coast, Australia*, May 1992. Available by anonymous ftp from ftp.cs.berkeley.edu in directory /ucb/CASTLE/Active_Messages.

[123] K Yelick. Programming models for irregular applications. In *Proceedings of the Workshop on Languages, Compilers and Run-Time Environments for Distributed Memory Multiprocessors, Boulder, CO*, October 1992. To appear in SIGPLAN notices.

Index of Package Names

Index of Package Functionality