The Parallel Asynchronous Data Routing Environment PADRE ABSTRACT

Kei Davis and Daniel Quinlan

Computing, Information, and Communications, Scientific Computing Group CIC-19,
Los Alamos National Laboratory, Los Alamos, NM 87545, USA,
WWW home page: `http://www.c3.lanl.gov/cic19/teams/napc/`

1 Introduction

The increasing complexity of parallel scientific (numerical) software, just as in all other areas of software development, demands increasing modularity and reusability of software components. For parallel scientific applications, the management of data distribution and interprocess- and interprocessor communication is an issue of some complexity that is not typically a consideration in more mainstream areas of computing.

A natural evolution of solutions to the complexity of data distribution may be observed. Early parallel machines provided 'native' communication libraries, e.g. CMMD on the Thinking Machines CM-5. Next came attempts at standardization, e.g. PVM, MPI, and the currently emerging MPI-2, often built on top of native libraries. While providing facilities for communication, libraries such as these do not provide support for the management or organization of distributed data.

The lack of any general-purpose distributed-data management facilities motivated the development of libraries such as Multiblock PARTI [1], PGSLib [2, 3], and KeLP [4, 5]. At this level of abstraction (unlike at the PVM/MPI level) it appears unreasonable to hope for a single satisfactory solution because of widely differing distribution and communication requirements, sometimes even within a single application or parallel library.

The sophistication of distributed-data management libraries (or *distribution libraries*) is such that it behooves the parallel application or library designer to use existing libraries rather than build them from scratch. Unfortunately, their interfaces are of sufficient complexity that the use of more than one such library, or offering the user the choice of distribution library to be used, will almost certainly greatly complicate the parallel library code, particularly in the former case wherein the same data may be alternately or even simultaneously be 'managed' by more than one distribution library.

2 PADRE

PADRE seeks to provide a uniform distributed-data management interface in the presence of:

– a multiplicity of distribution libraries (Multilock PARTI, PGSLib, Domain Layout, KeLP, or perhaps none in the degenerate case);
– a choice of communication libraries (PVM, MPI, MPI-2, or perhaps none, again in the degenerate case);
– a choice of execution models (SPMD or threaded);
– varied computational/communication patterns, and a variety of architectures (particularly MPP, SMP and DSM, and NOW).

In addition to providing a uniform interface, by supporting the simultaneous use of more than one distribution library, with transparent conversion of data between them, PADRE makes possible higher application performance than could be obtained when restricted (by reason of software complexity) to the use of a single distribution library.

In its current initial version PADRE is heavily oriented toward the distribution of array or array-like data. This in part reflects the fact that PADRE is not an academic experiment but rather a piece of software for which there is an immediate need which influences the initial specification. Nonetheless, it is fully intended that PADRE facilitate the distribution of other kinds of data; in particular particle-like data as used in, for example, modelling the evolution of galactic morphology using AMR, a precedented approach [6].

3 Software availability

PADRE is freely distributed. The PADRE source and documentation may be obtained via `http://www.c3.lanl.gov/PADRE`.

References

1. A. Sussman and G. Agrawal. *A Manual for the Multiblock PARTI Runtime Primitives.* ftp://hpsl.cs.umd.edu/pub/block_parti_distribution/doc.ps.gz, 1994.
2. R.C. Ferrell, D.B. Kothe, and J.A. Turner. PGSLib: A Library for Portable, Parallel, Unstructured Mesh Simulations. In *Proceedings of 8th SIAM Conference on Parallel Processing for Scientific Computing.* 1997.
3. R.C. Ferrell. Cambridge Power Computing Associates, http://www.cpca.com/
4. S.J. Fink and S. Baden. Run-Time Support for Multi-tier Programming of Block-Structured Applications on SMP Clusters. In *International Scientific Computing in Object-Oriented Parallel Environments, (ISCOPE'97).* LNCS 1343, Springer-Verlag 1997.
5. S. Baden. *The KeLP Programming System.* http://www-cse.ucsd.edu/groups/hpcl/scg/kelp.html.
6. J.P. Ostriker and M.L. Norman. Cosmology of the Early Universe Viewed Through the New Infrastructure. *CACM 40:11*, 1997.