PINEAPL: A European Project to Develop a Parallel Numerical Library for Industrial Applications

Daniela di Serafino[1], Lucia Maddalena[1] and Almerico Murli[1]

Center for Research on Parallel Computing and Supercomputers (CPS) - CNR
Via Cintia, Monte S. Angelo, 80126 Naples, Italy

Abstract. PINEAPL (Parallel Industrial NumErical Applications and Portable Libraries) is an ESPRIT project, in the area of High Performance Computing and Networking, with partners from industral, academic and research organizations. The aim of the PINEAPL project is to produce a library of parallel numerical software that is relevant to a wide range of industrial applications. Therefore, the choice of the areas of numerical software is driven by the applications provided by the industrial partners. In this paper we describe the structure and the strategies of the project and report on progress so far. Moreover, as an example, we present an application provided by an industrial partner and the parallel routine that has been developed taking into account the requirements of the application.

1 Introduction

PINEAPL (Parallel Industrial NumErical Applications and Portable Libraries) is a three years IV Framework ESPRIT project in the area of High Performance Computing and Networking, which started on January 1996. The main goal of the project is *to produce a general purpose library of parallel numerical software suitable for a wide range of computationally intensive industrial applications and portable and efficient across a wide range of high performance machines.*

Numerical software libraries supply *building blocks* for the solution of mathematical problems common to different applications. The availability of reliable, accurate and efficient numerical software allows users to solve application problems without dealing with details related to numerical algorithms and their implementation. This enables non-expert users to exploit the experience of numerical analysts and mathematical software developers, thus reducing their efforts and improving the quality of the results. Presently, there are more than 3000 industrial and academic organisations using general purpose numerical libraries in Europe alone. On the other hand, the widespread and effective use of High Performance Computing resources is inhibited by the lack of software libraries that are suitable for such advanced environments. The demand for these libraries is increasing as more end-users take advantage of networks and clusters of workstations and distributed memory computers. In this context, the PINEAPL project aims at reducing this gap, enabling users to take advantage of the increasing computing power and memory offered by multiple processors.

The project relies on a close collaboration between *academic, research* and *industrial organizations*, which are the partners of the PINEAPL Consortium. The industrial organizations, that make use of numerically intensive application codes for design and simulation, are the *end-users* of the project. They take advantage of the opportunity of introducing new mathematical models and techniques into existing application codes and of enhancing their performance by the incorporation of parallel library routines. The end-users' application codes have been chosen to represent a varied cross-section of industrial problems and the PINEAPL library coverage has been primarily determined by the requirements of these applications. The research and academic organizations, with well-known experience in the development of sequential and parallel mathematical software, act in the project as *numerical software suppliers* and *application parallelization experts*. Their expertise is used for the development of parallel numerical software of interest for the industrial applications. Each industrial partner has one or more associated parallelization experts, as it is shown in Table 1.

Table 1. PINEAPL Consortium

Project Coordinator: Numerical Algorithms Group (NAG), UK	
Industrial Partners	**Related Parallelization Experts**
British Aerospace (BAe), UK	University of Manchester (ManU), UK
Piaggio Veicoli Europei, Italy	IBM SEMEA, Italy; CPS, Italy
Thomson-CSF, France	CERFACS, France; CPS, Italy
Danish Hydraulic Institute (DHI), Denmark	Math-Tech, Denmark

Contacts with external collaborators and other related EC funded projects (EUROPORT) and user groups, such as the NAG User's Association, give further input to the software design and specification, to ensure that the needs of a larger industrial audience are met.

More details on the project can be found in the PINEAPL Web page located at the URL `http://www.nag.co.uk/projects/PINEAPL.html`.

2 Project Description and Progress

The choice of the areas of numerical software for the PINEAPL library has been driven by the applications provided by the industrial partners, summarized in Table 2. A first stage of the project has been devoted to a careful analysis of these applications and has led to the identification of the main areas to be covered by the numerical library. Some more areas have been added, that are likely

Table 2. PINEAPL applications

Applications	End-users
Wing design: maximum lift/minimum drag	BAe
Electro-magnetic wave reflection of ducts	
Estuarine and coastal hydraulics and oceanography, and environmental simulation	DHI
Oil reservoir simulation	Math-Tech
Engine simulation: chemically reactive flows with sprays	Piaggio
Design of nanometric recording devices	Thomson-CSF
Simulation of beam propagation in rod lasers	

to be of use in a wide range of computationally intensive applications outside the Consortium. The parallel library includes also dynamic load balancing and support/utility routines, that aid the development and the use of parallel software. All these areas are summarized in Table 3.

Table 3. Areas covered by the PINEAPL library

Areas	Routines	SW suppliers
Dense Linear Algebra	Banded solvers Condition number estimators	NAG
	Eigensolvers	NAG, CERFACS
Sparse Linear Algebra	Krylov iterative solvers Preconditioners	NAG
Non-linear Optimization	Constrained Unconstrained	ManU
Discrete Fourier Transform	1-D, 2-D, 3-D FFT	CPS
Partial Differential Equations	2-D, 3-D Fast Poisson Solvers 3-D Multigrid for Helmholtz equation	CPS
	Mesh partitioning	NAG
Dynamic Load Balancing		ManU
Support/Utility	Basic sparse matrix operations Input/output	NAG
	Data distribution	NAG, ManU

The library software is written in Fortran 77 and uses popular *de facto* standard message-passing libraries, mainly the BLACS [3], and MPI [6] and PVM [4] where the BLACS do not have the required functionality. The library is targeted primarily at distributed memory machines, ranging from low cost networks of workstations to high-end distributed memory parallel computers, but can be used also on shared memory machines provided that implementations of MPI or PVM are available. The driving criteria in the development of the library routines are ease of use, portability, performance, flexibility and reliability [5].

Specification documents have been produced for all the parallel library routines. Most of these routines are currently under development, and some of them have been already completed in the areas of Dense and Sparse Linear Algebra, Discrete Fourier Transforms, Optimization and Support/Utility.

Since the application codes provided by the end-users act as a testbed for most of the library routines, initial working implementations of these codes on the target architectures have been produced, taking into account the library software specifications. Moreover, benchmark problems have been prepared for future testing of the parallel versions of the codes that will be based on the library routines.

To test the quality of numerical algorithms and software developed within the PINEAPL project, the tool PRECISE (PRecision Estimation and Control in Scientific and Engineering computing) is being used [2]. PRECISE is a set of tools to perform numerical experiments exploring the accuracy and stability of computational schemes, and consists of two modules, one performing statistical backward error analysis and the other performing sensitivity analysis. At the beginning of the PINEAPL project it was in a Matlab prototype form; part of the work in the project is devoted to bring it to a mature and commercial quality. Most of the computational routines required for a PRECISE analysis have been converted to Fortran 77; the core PRECISE routines have been completed, and further work is devoted to allowing the user or developer of parallel algorithms to easily call PRECISE routines from the parallel applications. Moreover, PRECISE has been applied to numerical problems obtained from DHI, Piaggio and Thomson-CSF applications.

The project progress can also be found in the first project Annual Report, available on the WEB page.

3 Example: a 2-D FFT Routine for Simulation of Beam Propagation in Rod Lasers

In this section we give a short presentation of a pair application / library routine, where CPS is involved both as application parallelization expert and numerical software supplier.

3.1 Application and Numerical Solution Method

The application presented here is the modelling of the propagation of beams in diodes pumped rod lasers (coupled with thermal effects), provided by Thomson-CSF. The modelling is mainly devoted to describe the evolution of an optical beam inside the rod laser, in order to estimate the deformation of the plane wave, the resulting quality and amplification of the beam, and thus the energy efficiency of the whole component.

Basically, the beam propagation is modelled by a Helmholtz equation:

$$\varDelta \varPhi + k_0^2 n^2 \varPhi = 0,$$

where $\varPhi = \varPhi(x, y, z)$ is the (polarized) electromagnetic field, n is the index of the medium, and k_0 is the wave number $2\pi/\lambda$, where λ is the wavelength. $\varPhi$ and n have complex values. The separation of the fast variation of the field along the direction of propagation, say z, leads to an equation of the form:

$$\varDelta \varPsi + 2i\, k_z \frac{\partial \varPsi}{\partial z} + k_0^2 (n^2 - n_z^2) \varPsi = 0. \tag{1}$$

The solution of (1) is performed using a method in the class of *Beam Propagation Methods* (BPMs) [8], that are methods widely used in integrated optics. The version of BPM used here is based on a splitting of (1) into two equations, modelling the propagation and the "lens" diffraction, which are solved in turn, step by step, along the z-direction:

$$\varDelta_t \varPsi + \frac{\partial^2 \varPsi}{\partial z^2} + 2i\, k_z \frac{\partial \varPsi}{\partial z} = 0\,, \quad 2i\, k_z \frac{\partial \varPsi}{\partial z} + k_0^2 (n^2 - n_z^2) \varPsi = 0.$$

($\varDelta_t$ is the Laplace operator in the plane of the transverse field, i.e. in the (x, y)-plane.) The propagation from z to $z + dz$ is computed by a transformation from the (x, y)-plane to the Fourier space, where the solution is given analytically. This transformation, and its inverse, are performed as 2-D Discrete Fourier Transforms (DFTs) on complex data obtained with a uniform grid discretisation with periodic boundary conditions. The DFTs are carried out using a 2-D Fast Fourier Transform (FFT) algorithm.

At each step along the z-direction, the most time consuming part of the algorithm is the execution of direct/inverse FFTs, which requires at least 70% of the total computation time. Therefore, a significant speedup can be achieved using a parallel 2-D FFT routine.

3.2 The 2-D FFT Routine

The 2-D FFT routine developed for the PINEAPL library computes the 2-D DFT of a bivariate sequence of complex data values $Z = (z_{j_1 j_2})$:

$$\hat{z}_{k_1 k_2} = \frac{1}{\sqrt{mn}} \sum_{j_1=0}^{m-1} \sum_{j_2=0}^{n-1} z_{j_1 j_2} \times e^{\pm 2\pi i \left(\frac{j_1 k_1}{m} + \frac{j_2 k_2}{n}\right)}, \quad \begin{array}{l} j_1, k_1 = 0, \ldots, m-1 \\ j_2, k_2 = 0, \ldots, n-1 \end{array}.$$

The 2-D DFT is obtained performing m 1-D transforms of length n, on the rows of the matrix Z, and n 1-D transforms of length m, on the rows of the transposed resulting matrix, i.e. on the columns of the resulting matrix. The FFT algorithm used to perform the 1-D transforms is the *Stockam self-sorting algorithm*, described in [7].

As all the routines of the PINEAPL Library, the FFT routine assumes that a 2-D grid of p logical processors is avalable. The parallel algorithm is based on a *row-block distribution* of the matrix Z, that is $M_b \simeq m/p$ consecutive rows of Z are allocated to processors on the 2-D grid row by row (i.e. in row major ordering of the grid), starting from the $\{0,0\}$ processor. Each processor performs M_b 1-D transforms of length n, on the rows assigned to it, contributes to the transposition of the global resulting matrix, performs $N_b \simeq n/p$ 1-D transforms of length m, and finally, contributes to the transposition of the global final matrix. All the communication is performed in the transposition of the distributed matrices. More details are given in [1].

We note that the routine allows to avoid the second transposition of data, leading to a significant reduction of the execution time. This choice is recommended, for example, when a forward DFT followed by a backward DFT has to be computed, since the second DFT can be computed directly from a matrix in transposed order. Further time saving is allowed when several DFTs of the same length must be computed, since the exponential coefficients required by the FFT algorithm can be stored and reused.

Preliminary experiments have been carried out to test the parallel performance of the 2-D FFT routine, on an Intel iPSC/860 at CPS, with 8 nodes and 16 MB of memory per node. Each node has a peak performance of 40 Mflops in double precision; the peak bi-directional bandwidth among two nodes is 2.8 MB/s and the latency is 6.5 ms. The routine has been executed in double precision, using the NX (Intel native node operating system) version of the BLACS.

Figure 1 shows the parallel efficiency of the routine, with and without the second global transpose, for $p = 2, 4, 8$ and different values of $m = n$. Such values are radix-r, with $r = 2, 3, 5$ ($256 = 2^8$, $512 = 2^9$, $243 = 3^5$, $625 = 5^3$), and mixed-radix, also including large prime factors ($450 = 2 \times 3^2 \times 5^2$, $600 = 2^3 \times 3 \times 5^2$, $249 = 3 \times 83$, and $544 = 2^5 \times 17$). On one processor the (sequential) NAG Fortran Library routine which implements the Stockam self-sorting algorithm has been executed. As it was expected, avoiding the second global transposition of data improves the efficiency of the routine. Among the radix-r DFTs, the case $r = 3$ corresponds to the lowest efficiency; this can be explained since the sequential radix-3 FFT algorithm has a relatively lower operation count. The most efficient parallel executions are those corresponding to the mixed-radix case with large prime factors, since their sequential counterparts require a larger computation time. These results can be considered satisfactory, since communication has a significant weigth in parallel FFT algorithms, and typically accounts for a large part of the computational effort.

Acknowledgments. The authors wish to thank Thomson-CFS LCR for providing the description of their application and NAG Ltd for their helpful comments.

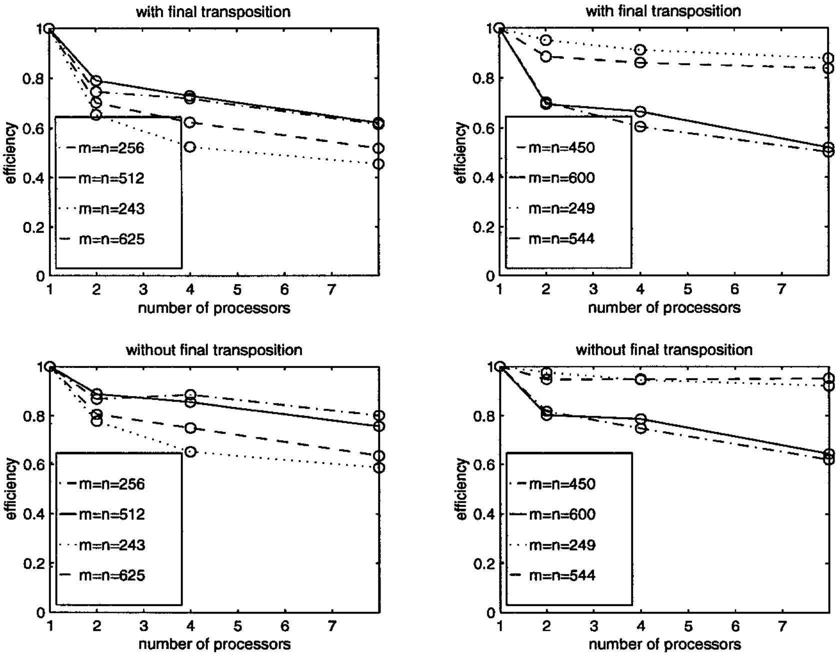

Fig. 1. Parallel efficiency of the 2-D FFT routine with and without the final global transposition, on 2, 4 and 8 processors

References

1. Carracciuolo, L., De Bono, I., De Cesare, M.L., di Serafino, D., Perla, F.: Development of a Parallel Two-dimensional Mixed-radix FFT Routine. Tech. Rep. TR-97-8, Center for Research on Parallel Computing and Supercomputers (CPS) - CNR, Naples, Italy (1997)
2. Chaitin-Chatelin, F., Fraysse, V.: Lectures on Finite Precision Computations. SIAM, Philadelphia (1996)
3. Dongarra, J.J., Whaley, R.C.: A Users' Guide to the BLACS v 1.0. LAPACK Working Note No. 94, Technical Report CS-95-281, Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN (1995)
4. Geist, A., Beguelin, A., Dongarra, J.J., Jiang, W., Manchek, R., Sunderam, V.: PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing. MIT Press, Cambridge, MA (1994)
5. Krommer, A., Derakhshan, M., Hammarling, S.: Solving PDE Problems On Parallel and Distributed Computer Systems Using the NAG Parallel Library. Communication at the Conference HPCN'97, April (1997)
6. Snir, M., Otto, S.W., Huss-Lederman, S., Walker, D.W., Dongarra, J.J.: MPI: The Complete Reference. MIT Press, Cambridge, MA (1996)
7. Temperton, C.: Self-sorting Mixed-radix Fast Fourier Transforms. J. Comp. Phys. **52** (1983) 1–23
8. Van Roey, J., Van der Donk, J., Lagasse, P.E.: J. Opt. Soc. Am. **71** (1981) 803