A Proposal for Parallel Sparse BLAS

Fabio Cerioni and Michele Colajanni * and Salvatore Filippone ** and Stefano Maiolatesi

1 IBM Semea
P.le G. Pastore 6
Roma, Italy
2 Dipartimento di Informatica,
Sistemi e Produzione
Università degli Studi di
Roma - Tor Vergata, Roma, Italy

Abstract. In this paper we propose a set of parallel interfaces that extends the sparse BLAS presented in [8] to distributed memory parallel machines with message passing programming interfaces. Our main target is the implementation of iterative methods with domain-decomposition based preconditioners in an effective and structured way, while keeping the benefits of the serial sparse BLAS. The structure of our framework has also been influenced by the dense PBLAS proposal [5].

1 Introduction

Our aim in this work is to provide an interface for the convenient implementation of iterative methods for sparse linear systems on distributed memory computers; therefore the kind of applications we have in mind closely follows those forming the framework of the serial Sparse BLAS (SpBLAS) proposal [8].

The current proposal should be viewed as complementary to the “templates” concept provided in [2], addressing convenient implementation of those templates.

One of the key points of this proposal is to maintain the benefits of the SpBLAS proposal by focussing on those issues arising from the distributed memory nature of our target computational engines, while keeping the local parts of the computations strictly adherent to the SpBLAS interface.

In handling data distribution and communications this proposal has also been influenced by the PBLAS proposal developed in the context of ScaLAPACK [4]; however the specific nature of sparse linear systems has forced a number of changes in the semantics of the operations.

Our main contribution is the definition of an adequate set of parallel operations that are able to capture the computational requirements of common sparse computations, while guaranteeing convenient packaging of data structures, isolation from low-level message passing details and preservation of the advantages of

* colajanni@utovrm.it
** filippon@vnet.ibm.com
the serial SpBLAS software. This is achieved by providing a dual Fortran 90 and FORTRAN 77 interface; to allow maximal reuse of existing serial software [8] the internal implementation is based on the FORTRAN 77 interface.

Our approach assumes that the application is parallelized through an index space decomposition; this approach is convenient and consistent with common usage for most applications we are aware of, with the user keeping control over the choice of the data decomposition.

We have been influenced by ideas expressed at the BLAS Technical Workshop held at the University of Tennessee in November 1995 [6]; a previous version of this paper has appeared in [3].

2 General Overview

The main operations included in our library, called "Parallel Sparse BLAS" (PSBLAS), are:

- Sparse matrix by dense matrix product;
- Sparse triangular systems solution;
- Vector and matrix norm;
- Dense matrix sums;
- Dot products;
- Data exchange and update;
- Data structure preprocessing and initialization.

A more detailed description of these operations will be given in Sec. 4. Some of the operations do not appear in the serial SpBLAS because they are trivially implemented either through open code or calls to the dense BLAS; they do appear here, however, and are different from the similar PBLAS because of the peculiarity of the adopted data distribution. The domain decomposition features of the underlying data distribution also prompts the need for specialized data exchange routines; we have chosen the BLACS [7] to perform the underlying communication operations in a portable and efficient way.

2.1 Data Decomposition

In any distributed memory application the data structure used to partition the problem is essential to the viability of the entire approach; the criteria guiding the decomposition choice are:

1. Load balancing;
2. Communication costs reduction;
3. Efficiency of the serial parts of the computation;

In the case of dense linear algebra algorithms [4] a block-cyclic data distribution of the index space is sufficiently general and powerful to achieve a good compromise between the different factors affecting performance; this distribution also