Summary Report on ScaLAPACK

Anirudh Modi

May 4, 1998

1 Introduction

ScaLAPACK is an acronym for Scalable Linear Algebra Package or Scalable LAPACK. It is a library of high-performance linear algebra routines for distributed memory message-passing MIMD computers and networks of workstations supporting parallel virtual machine (PVM) and/or message passing interface (MPI). It is a continuation of the LAPACK project, which designed and produced analogous software for workstations, vector supercomputers, and shared memory parallel computers. Both libraries contain routines for solving systems of linear equations, least squares problems, and eigenvalue problems. The goals of both projects are efficiency, scalability, reliability, portability, flexibility, and ease of use(??!!).

2 How is ScaLAPACK used ?

Four basic steps are required to call a ScaLAPACK routine.

1. *Initialize the process grid*: A call to the ScaLAPACK TOOLS routine SL_INIT initializes the process grid. This routine initializes a $P_{row} \times P_{col}$ process grid by using a row-major ordering of the processes, and obtains a default system context.

2. *Distribute the matrix on the process grid*: All global matrices must be distributed on the process grid prior to the invocation of a ScaLAPACK routine. It is the user's responsibility to perform this data distribution. Each global matrix that is to be distributed across the process grid must be assigned an array descriptor. array descriptor is initialized

with a call to a ScaLAPACK TOOLS routine called DESCINIT and must be set prior to the invocation of a ScaLAPACK routine.

3. *Call ScaLAPACK routine*: All ScaLAPACK routines assume that the data has been distributed on the process grid prior to the invocation of the routine. There naming convention is similar to the LAPACK library routines.

4. *Release the process grid*: After the desired computation on a process grid has been completed, it is advisable to release the process grid via a call to BLACS_GRIDEXIT.

3 Problems with ScaLAPACK

ScaLAPACK promises a lot in functionality, but unlike its serial counterpart (LAPACK), the documentation lacks in clarity and is almost non-understandable. The example source code that is used in the documentation is horribly written with few comments and uses a fixed size matrix of size 9×9. It seems very cumbersome to even extend the example to solve for a 10×10 matrix, let alone understand how to use ScaLAPACK! To top it all, the choice of variable names used throughout the documentation are meaningless and add to the chaos. Needless to say, the name of the routines themselves do not make sense and a lookup table is required at every step of the programming in order to write a working code. This is really bad as its been over five years now since the advent of FORTRAN 90 which supports longer than 6 character names. As a conclusion, programming today in ScaLAPACK for a new user is nothing but frustrating!