Use of HPCC Software libraries in Industrial Applications

Kenneth A. Hawick
Northeast Parallel Architectures Center
Syracuse University, USA
&
David W. Walker
Mathematical Sciences Section
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Presented at:
SPAA'95 — PPoPP'95 — UIWOPPS
Santa Barbara, California, July 17 – 22, 1995

NPAC Technical Report SCCS 723

Overview:

- Acknowledgments
- HPCC Libraries
- Industrial Driving Forces
- Pre-requisites for Building Libraries
- Case Study: ScaLAPACK Library
- Industrial Application of ScaLAPACK
- Conclusions
- Online Internet Resources

Acknowledgments:

- ScaLAPACK Development Team (Univ Tennessee / ORNL): Jaeyoung Choi; Susan Ostrouchov; Antoine Petitet; Clint Whaley; Jack J. Dongarra; David W. Walker.
- Computational Electromagnetics Project Application Team (NPAC/Syracuse Research Corp): Gang Cheng; Ken Hawick; Xianneng Shen; Jay Mortensen; Jim Lauer; Debra Wilkes.
- Industrial Numerical and Simulations Group at the Edinburgh Parallel Computing Centre (EPCC): Ken Hawick; Brian Wylie; Simon Chapple; Evan Welsh; Mark Sawyer; Julian Parker; Hon Yau;.
- CHIMP/Parallel Utility Library (PUL) Group at EPCC: Lyndon Clarke; Shari Trwein; Bob Fletcher; Alasdair Bruce; James Mills.

HPCC Libraries:

- encapsulate expertise
- can be extensively tested independently
- can provide portability across different vendor platforms

Industrial Driving Forces:

- Software development for HPCC platforms often more expensive in terms of development and testing than the hardware, for industrial reliability requirements.
- Investment only makes sense if software reuse - as libraries - is possible.
- Libraries preferable to template/skeletons as greater encapsulation allows better testing.
- although trade-off of performance against reliability and reuse exists - high performance still highly desirable!

Industrial Examples (UKMO):

- UK Meteorological Office: Unified Model is 150k lines of Fortran
- Parallel coding effort easier if higher than raw message passing libraries exist for grid manipulations.
- Multiple algorithm paradigms (data parallel for dynamics; task parallel for precipitation model; scattered spatial decomposition for data assimilation) requires interoperable library components with standard library interfaces.
- Parallel Utility Library (PUL) set designed and built at Edinburgh as a result.

CHIMP/PUL Libraries:

- CHIMP (Common High Level Interface to Message Passing), predated MPI and was attempt to provide a message passing that would allow partitioning of message tag space for building software libraries on top of.
- PUL (Parallel Utility Library) is a collection of libraries and skeletal templates builton top CHIMP, and now MPI.
- PUL examples include: general grid decomposition (like BLACS in ScaLAPACK); Task farm paradigm; scattered spatial decomposition; generalised blocked distributed file I/O. (Clarke et al, Edinburgh Parallel Computing Center)

Industrial Examples (RR):

- Rolls Royce (Aerospace Engine Design)
- Turbofan Hypersonic CFD simulation code of circa 30k lines Fortran.
- Code required linear algebra library such as ScaLAPACK which was not then available in 1991.
- Prototype was built using customised solver, but not able to be introduced into production due to high degree of code maintainance that would have been required.
- Supported library module would have allowed use of parallel platform in production instead of vector machines only.

Industrial Examples (AEA):

- UK Atomic Energy Authority - nuclear reactor simulations codes
- Large codes, need to be **very** reliable, and require extensive recurrency testing of all software modules - test/verification suite often larger than simulation code itself.
- Software libraries allow testing effort to be reused, as well as design verification/validation against other codes using the same library or a different library if on a vector platform.
- Use of CHIMP message passing library and Parallel Utility library for block decompositions allowed introduction of parallel computing into an otherwise ‘vector’ environment.

Industrial Examples (BAe):

- British Aerospace - radar cross section analysis codes.
- customised codes using Occam and assembly language to exploit cheap parallel hardware. No reliable dense linear algebra library existed in 1990 for HPCC parallel platforms.
- ScaLAPACK would (now) allow improved portable implementation.

Pre-requisites for HPCC Libraries:

- library typically built on a reliable message passing system.
- message passing calls actually used must be reliable and widely available - either in a portable library or standard such as PVM, MPI or CHIMP, or in the proprietary package available on target platforms (eg Intel NX/2, IBM EUI,...)
- for ease of development of multiple library modules, message tag space needs to be sensibly partitioned - for example alphanumeric group tags plus numeric message ID allows each library module to restrict itself to its own tag-space and ensure non-interference of library modules.
- well defined purpose for library is important for user as well as software designer. (contrast with some proprietary libraries which are ad-hoc collection of software packages). Difficult to maintain with time, and hard for user to know what to expect.

Case Study: ScaLAPACK Library - Motivation

- On shared memory vector supercomputers large, optimized software libraries exist:
    - BLAS, EISPACK, LINPACK, LAPACK,...
    - NAG, IMSL, ESSL,...
- Little such software runs efficiently on current and emerging parallel architectures

  ⇒ "Software Gap"
- Development of high-quality, portable software libraries for concurrent computers as a **key enabling technology** essential to more widespread use of HPCC platforms by industry as well as by academia.

Case Study: ScaLAPACK Library - Objectives

- Goal:

  To develop a library of high-quality, portable software for performing linear algebra computations on NUMA supercomputers, specifically MIMD distributed memory concurrent computers.

- LAPACK has already done this for workstations and shared memory computers.
- ScaLAPACK extends the functionality of LAPACK to distributed memory machines.

ScaLAPACK=Scalable LAPACK

i.e., we want the performance/node to stay constant as the problem size scales with the number of nodes.

Case Study: ScaLAPACK Library - Basic Problems

- Basic problems addressed by ScaLAPACK include:
- Linear systems: $Ax = b$
- Least squares: $\min_x \|Ax - b\|_2$, $A = U\Sigma V^T$
- Eigenvalues and vectors: $Ax = \lambda x$, $Ax = \lambda Bx$
- ScaLAPACK and LAPACK use block-partitioned algorithms, so algorithm is expressed in terms of matrix-matrix operations performed using Level 3 BLAS. which maximizes data reuse in upper levels of memory, and reduces frequency of data movement between:
  - shared memory and cache for shared memory machines;
  - processors for distributed memory machines.

Case Study: ScaLAPACK Library - Building Blocks

- Basic Linear Algebra Communication Subprograms (BLACS) for communicating parts of a matrix. May be optimized for hardware.
- Parallel BLAS (PBLAS). Level 1, 2 and 3 BLAS routines for distributed matrices and vectors.
- Sequential BLAS. May be optimized for hardware.
- Matrix transpose routines.
- Data distribution transformation routines for dynamically changing data distribution.

Case Study: ScaLAPACK Library - BLACS

- Basic Linear Algebra Communication Subprograms communicate parts of: rectangular matrices; trapezoidal matrices.
- Processes are laid out on a 2D logical mesh
- Processes are referenced by location in topology
- Blocking point-to-point communication
- Collective communication over row, column or all of topology
  - broadcast
  - some reduction routines
- No message tags
- BLACS context is compatible with MPI communicator

Case Study: ScaLAPACK Library - PBLAS

- PBLAS perform Level 1, 2, and 3 BLAS operations on distributed matrices
- Matrices are global objects
- Matrices have a block cyclic data distribution
- PBLAS are a subset of the BLAS, but
  - no banded and packed storage schemes
  - no vector rotation routines
- Same calling sequence as BLAS except for each distributed matrix we have
  - global indices of start of matrix
  - descriptor array

Case Study: ScaLAPACK Library - Key Ideas

- Use block-partitioned algorithms to maximize data reuse in upper levels of memory
  - reduce cache misses
  - reduce frequency of communication
- Use Parallel BLAS (PBLAS) as main computational building blocks.
- Use Basic Linear Algebra Communication Subprograms (BLACS) to perform communication
- Hide parallelism within the PBLAS
- Fine-tune performance by adjusting data layout parameters

- **Important:** The PBLAS make use of the sequential BLAS for which assembly coded versions exist for many processors.

Case Study: ScaLAPACK Library - Data Decomposition

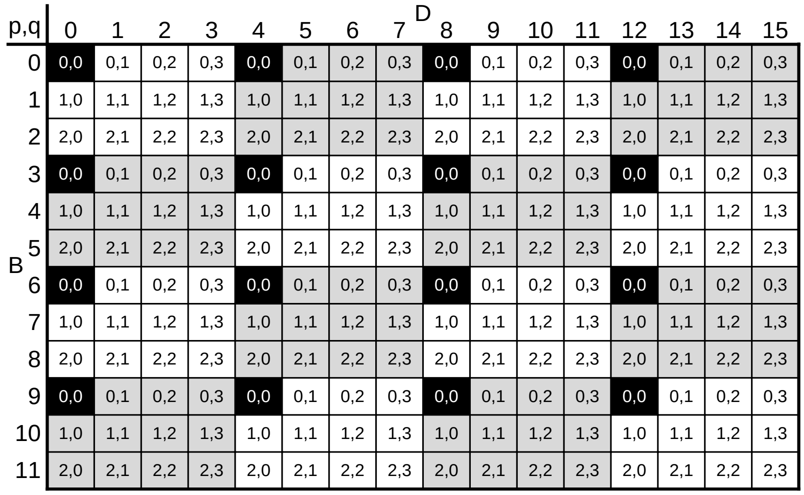
- We want a data decomposition scheme that:
- is practical,
- is general-purpose,
- gives good load balance,
- can reproduce all the most commonly-used data distributions.

⇒**Block-Cyclic Distribution**

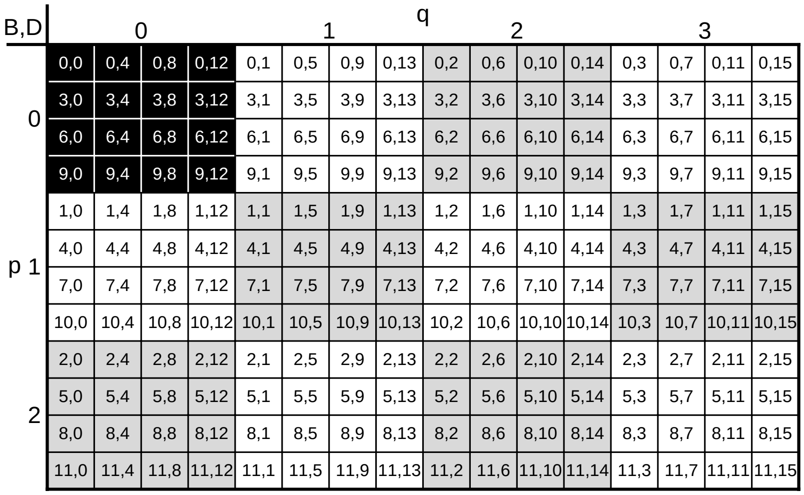
- Partition matrix into blocks of $r \times s$ elements.
- Can regard processors as being arranged as a 2-D mesh, or template.

$$(m,n) \mapsto (\,(p,q),(b,d),(i,j)\,)$$

Case Study: ScaLAPACK Library - Block Cyclic Example



p,q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3
3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
4	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
5	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3
6	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
7	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
8	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3
9	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3	0,0	0,1	0,2	0,3
10	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3	1,0	1,1	1,2	1,3
11	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3	2,0	2,1	2,2	2,3



B,D	0				1				2				3			
0	0,0	0,4	0,8	0,12	0,1	0,5	0,9	0,13	0,2	0,6	0,10	0,14	0,3	0,7	0,11	0,15
	3,0	3,4	3,8	3,12	3,1	3,5	3,9	3,13	3,2	3,6	3,10	3,14	3,3	3,7	3,11	3,15
	6,0	6,4	6,8	6,12	6,1	6,5	6,9	6,13	6,2	6,6	6,10	6,14	6,3	6,7	6,11	6,15
	9,0	9,4	9,8	9,12	9,1	9,5	9,9	9,13	9,2	9,6	9,10	9,14	9,3	9,7	9,11	9,15
1	1,0	1,4	1,8	1,12	1,1	1,5	1,9	1,13	1,2	1,6	1,10	1,14	1,3	1,7	1,11	1,15
	4,0	4,4	4,8	4,12	4,1	4,5	4,9	4,13	4,2	4,6	4,10	4,14	4,3	4,7	4,11	4,15
	7,0	7,4	7,8	7,12	7,1	7,5	7,9	7,13	7,2	7,6	7,10	7,14	7,3	7,7	7,11	7,15
	10,0	10,4	10,8	10,12	10,1	10,5	10,9	10,13	10,2	10,6	10,10	10,14	10,3	10,7	10,11	10,15
2	2,0	2,4	2,8	2,12	2,1	2,5	2,9	2,13	2,2	2,6	2,10	2,14	2,3	2,7	2,11	2,15
	5,0	5,4	5,8	5,12	5,1	5,5	5,9	5,13	5,2	5,6	5,10	5,14	5,3	5,7	5,11	5,15
	8,0	8,4	8,8	8,12	8,1	8,5	8,9	8,13	8,2	8,6	8,10	8,14	8,3	8,7	8,11	8,15
	11,0	11,4	11,8	11,12	11,1	11,5	11,9	11,13	11,2	11,6	11,10	11,14	11,3	11,7	11,11	11,15

Industrial Application of ScaLAPACK

- Large Scale industrial application employed by Syracuse Research Corporation (SRC) in defense simulations of radar cross sections for "flying objects"
- Serial code (used LINPACK) widely used by SRC's customers, but to allow simulation of new "flying objects" with a lot of mesh details necessary, HPCC was needed.
- Cost performance, portability across platforms was driving force. Code was sufficiently large that software investment effort porting to a single proprietary system was risky.
- Scalability also an issue for even larger problems in future.

SRC ParMoM Package

- Parametric Patch Method of Moments Code for radar cross section modeling of full airborne system.
- Problem can be summarised as assembly and solution of large dense matrix equation
- Matrix contains impedance coefficients
- RHS is (multiple) excitation vectors from different incoming radar signals
- solution vector is electric currents over surface of aircraft.

Design of Parallel ParaMoM

- Main component of the code is matrix L.U factorisation and solve (this is $O(N^3)$, where N is number of unknown or the points for this application.)
- although some proprietary systems have library for this (eg Thinking Machines' CMSSL, or Intel SSL) ScaLAPACK was only truly portable one.
- Matrix assembly is $O(N^2)$ and disassembly is $O(N)$ which are still significant for very large N.
- ScaLAPACK is conveniently implemented on the BLACS layer, which was an appropriate communications library for the matrix assembly code. The interoperability of these two layers allowed a truly portable application code.

Parallel ParaMoM

- Successful ports to Intel (ScaLAPACK BLACS on NX/2); CM5 (using CMMD); IBM SP2 using EUI-H; various workstation clusters (Sun, DEC, IBM,...) using PVM as underlying layer, including use of underlying ATM hardware.
- tunable blocking parameters in ScaLPACK library were valuable in tuning different application problem (mesh sizes) to different architectures - in a portable way.

Selected Timing comparisons for N = 4889 (in seconds)

Platform	N_p	Fill	LU
Alpha(FDDI)	8	1420	1120
IBM RS(Ether)	8	1501	1805
iPSC/860	64	526	281
CM-5	32	3295	171

Platform	N_p	Setup	RHS	Total
			+ Field	
Alpha(FDDI)	8	12.3	18.0	2570.2
IBM RS(Ether)	8	51.4	28.2	3385.0
iPSC/860	64	45.4	53.0	904.9
CM-5	32	21.1	4.3	3491.3

Portability and interoperability is greatest benefit.

Timing curves for various implementations:

Fig. 4 Fill Time vs. Processors(Matrix Size=988)

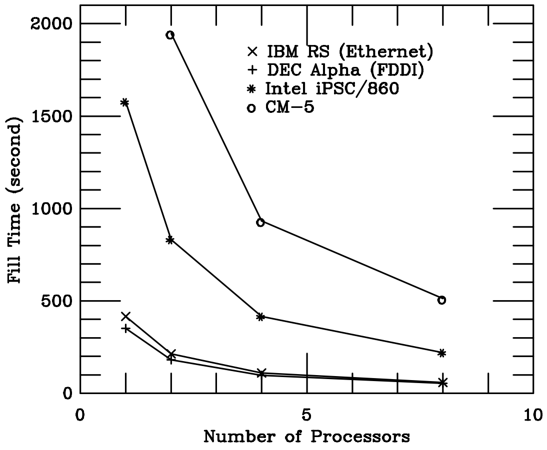
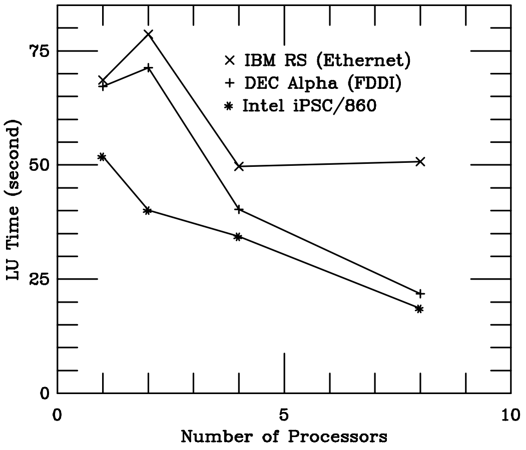


Fig. 5 LU Time vs. Processors(Matrix Size=988)



Conclusions/Summary:

- use of existing (tested) software always favored by industry
- cluster technology is viable for CEM applications of modest size
- use of portable (library based) software HPCC software allows straightforward move from application development on cluster to production run on MPP.
- good HPCC software libraries **can** be constructed - with careful design and high quality software engineering.
- - final thought - software libraries may form major component of the runtime libraries for high level parallel languages such as HPF.

Online Internet Resources:

- **http://www.npac.syr.edu/** The Northeast Parallel Architectures Center (NPAC) Main Server (containing documentation on CEM Application of ScaLAPACK)
- **http://www.netlib.org/nse/home.html** The National HPCC Software Exchange (containing ScaLAPACK software and documentation)
- Ken Hawick (hawick@npac.syr.edu); `http://www.npac.syr.edu/users/hawick/homepage`
- David Walker (walker@msr.epm.ornl.gov); `http://www.epm.ornl.gov/ walker`