Linear Algebra Session - CRPC May 1997 Annual Meeting

Jack Dongarra
University of Tennessee
and
Oak Ridge National Laboratory
http://www.netlib.org/utk/people/JackDongarra

1

Outline

- Jack Dongarra - Overview/Update
- Dan Sorensen - ARPACK Update
- Rich Lehoucq - Really Large Chemistry Eigenvalue Problems

2

Posters: Linear Algebra Session

- Chico Gomez - ARPACK++
- Marielba Rojas - Regularization of Discrete Ill-Posed problems
- Clint Whaley - ScaLAPACK in HPF
- Chao Yang - Crystal growth, or Truncated RQ or Polynomial spectral transformations using P_ARPACK

3

Panel: Linear Algebra Session

- Leader: Dan Sorensen
- Panelists:
  - Todd Arbogast
  - Dennis Gannon
  - Richard Tapia

4

ScaLAPACK Update

- Release 1.5 just out
- First portable, message passing/heterogeneous library for numerical computations.
- Uses latency-tolerant design and works well for systems with deep memory hierarchy.
- ScaLAPACK Users' Guide finished
  - Users Guide available 7/97 from SIAM
  - CD Rom of code, pre-built, working notes, & testing
- Sparse direct solver
- Out-of-Core methods
- HPF Interface

5

ScaLAPACK - What's Included

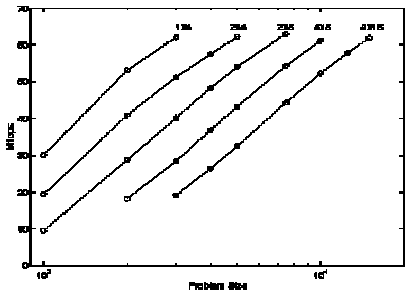
Problem type Ax = b	SDrv	EDrv	Factor	Solve	Inv	Cond Est	Iter Refin
Triangular				X	X	X	X
SPD	X	X	X	X	X	X	X
SPD Banded	X		X	X			
SPD Tridiagonal	X		X	X			
General	X	X	X	X	X	X	X
General Banded	X		X	X			
General Tridiagonal	X		X	X			
Least squares	X		X	X			
GQR			X				
GRQ			X				

Ax = λx or Ax = λBx	SDrv	Edrv	Reduct	Solution
Symmetric	X	X	X	X
General	X		X	X
Generalized BSPD			X	X
SVD	X	X	X	X

- Timing and Testing routines for almost all
- This is a large component of the package
- Prebuilt libraries available for SP, PGON, HPPA, DEC, Sun, RS6K

ScaLAPACK Performance

- SLUG provides discussion of performance and scalibility.
- MPP as well as Clusters.
  - T3E, SP2, ASCI Red, Paragon, Berkeley NOW, Cluster of Sun's
- Data layouts and blocksize



7

Performance of ScaLAPACK

	Process Grid	Block Size	Values of N				
			2000	5000	7500	10000	15000
Cray T3E[a]	1 × 4	32	702	884	932		
PSGESV	2 × 8	32	1608	2680	3218	3356	3602
	4 × 16	32	2419	6912	9028	10299	12547
IBM SP2[b]	1 × 4	50	421	603			
PDGESV	2 × 8	50	722	1543	1903	2149	
	4 × 16	50	924	3017	4295	5596	7057
Intel XP/S MP Paragon[c]	1 × 4	32	212	282			
PDGESV	2 × 8	32	460	865	1010		
	4 × 16	32	721	2084	2837	3344	3963
Intel ASCI Red Supercomputer[d]	1 × 4	32	566				
PDGESV	2 × 8	32	1379	2497			
	4 × 8	32	1912	4030			
	4 × 16	32	2389	6329		9731	
Cluster of Sun Ultra. 2s[e]	2 × 2	64	203	340	380		
PDGESV	2 × 4	64	196	499	618	689	
	2 × 6	64	212	608	816	934	
Berkeley NOW[f]	1 × 4	32	350				
PDGESV	1 × 8	32	560	744			
	2 × 8	32	811	1310	1472	1547	
	4 × 8	32	948	1913	2149	2324	2436
	4 × 16	32	1171	3091	3937	4263	4560

Parallelism in ScaLAPACK

- Level 3 BLAS block operations
  - All the reduction routines
- Pipelining
  - QR Algorithm, Trianaular Solvers, classic factorizations
- Redundant computations
  - Condition estimators
- Static work assignment
  - Bisection
- Task parallelism
  - Sign function eigenvalue computations
- Divide and Conquer
  - Tridiagonal and band solvers, symmetric eigenvalue problem and Sign function
- Cyclic reduction
  - Reduced system in the band solver

9

HPF Interface for ScaLAPACK

- Interface provided for a subset of routines
- Linear solvers (general, pos. def., least squares) & matrix multiply
- Given these declarations:

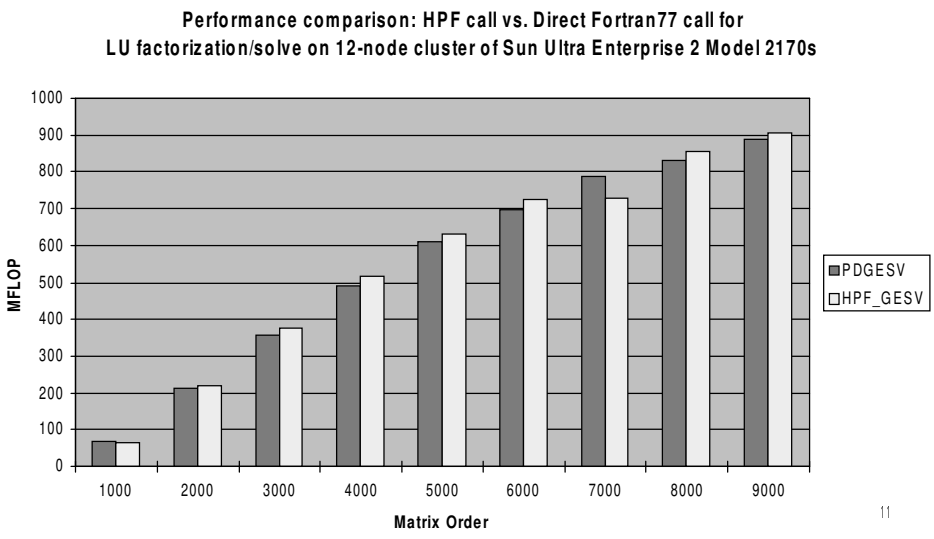
```
USE  HPF_LAPACK
REAL, DIMENSION(1000,1000) :: A, B, C
!HPF$ DISTRIBUTE (CYCLIC(64), CYCLIC(64)) :: A, B, C
```

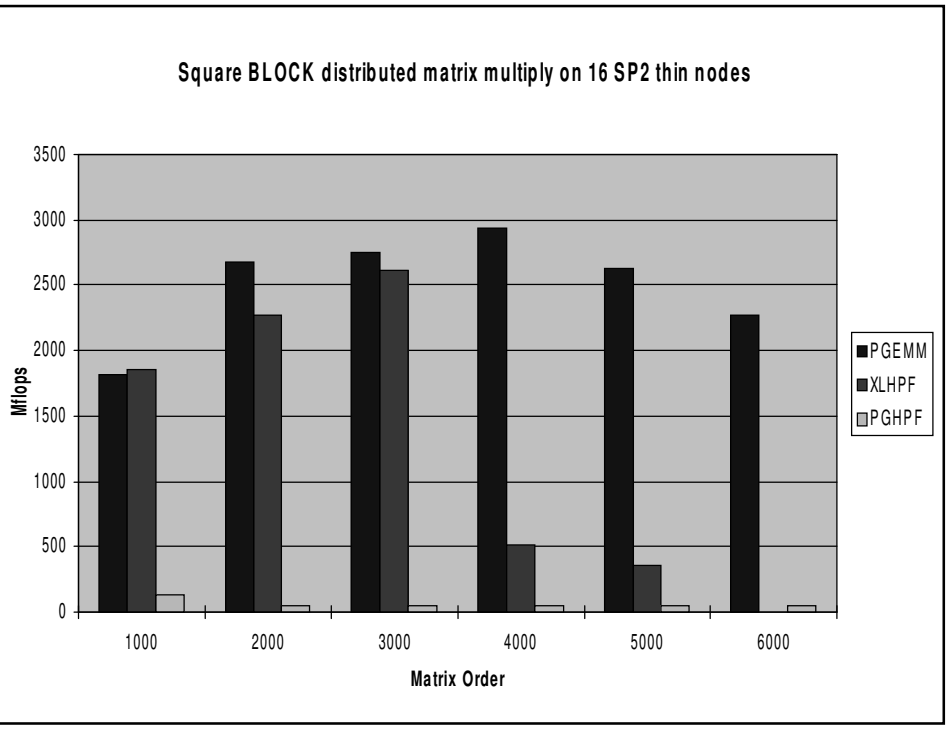
- Calls could be made as simple as (same as F90):

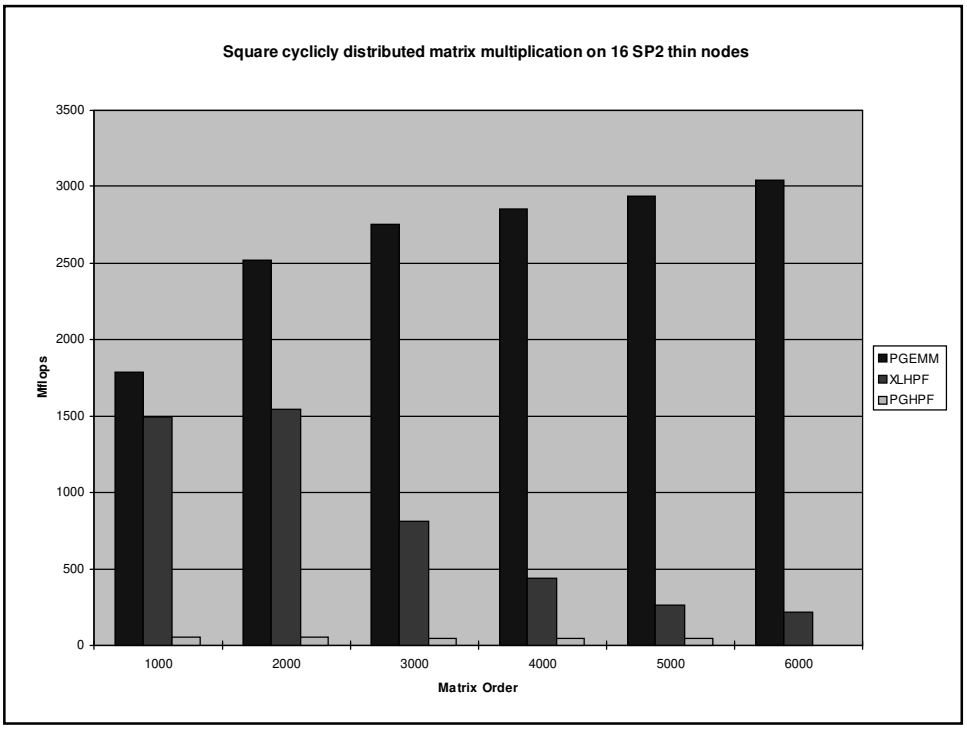
```
CALL LA_GESV(A, B)
CALL LA_POSV(A, B)
CALL LA_GELS(A, B)
CALL LA_GEMM(A, B, C)
```

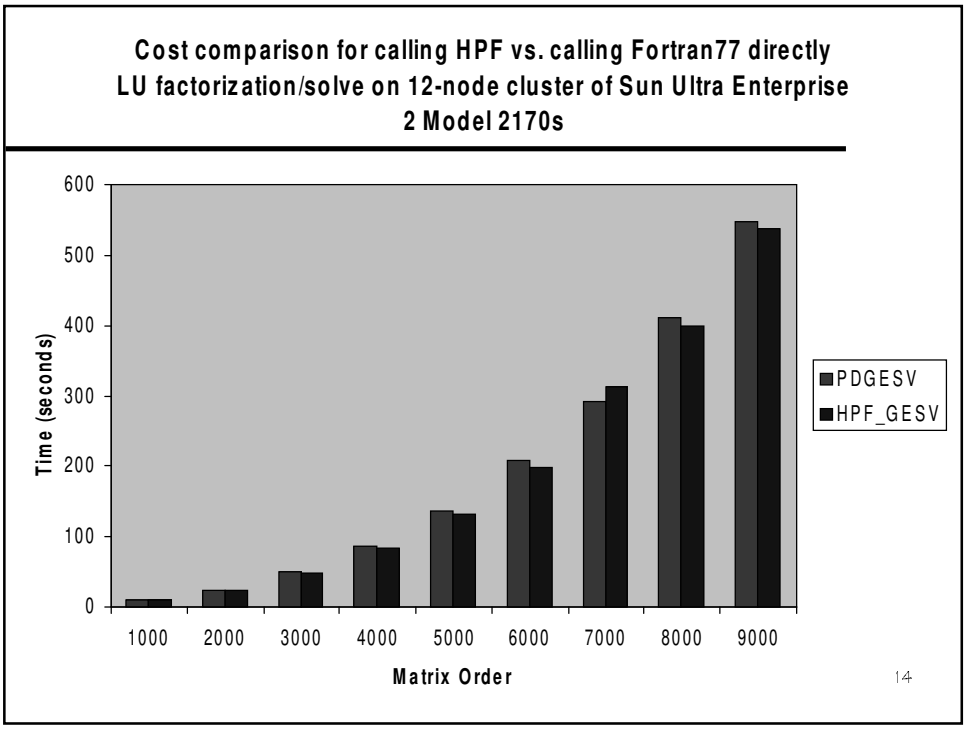
10

ScaLAPACK HPF Comparison






Square cyclicly distributed matrix multiplication on 16 SP2 thin nodes
Mflops
3500
3000
2500
2000
1500
1000
500
0
1000
2000
3000
4000
5000
6000
Matrix Order
PGEMM
XLHPF
PGHPF

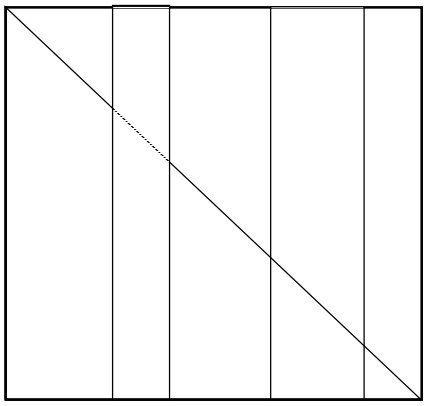

Cost comparison for calling HPF vs. calling Fortran77 directly
LU factorization/solve on 12-node cluster of Sun Ultra Enterprise 2 Model 2170s
Time (seconds)
600
500
400
300
200
100
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Matrix Order
PDGESV
HPF_GESV
14

Out of Core Approach

- High-level I/O Interface
- ScaLAPACK uses a 'Right-looking' variant for LU, QR and Cholesky factorizations.
- A 'Left-looking' variant is used for Out-of-core factorization to reduce I/O traffic.
- Requires two in-core column panels.
- Imposes another level in the memory hierarchy.

15

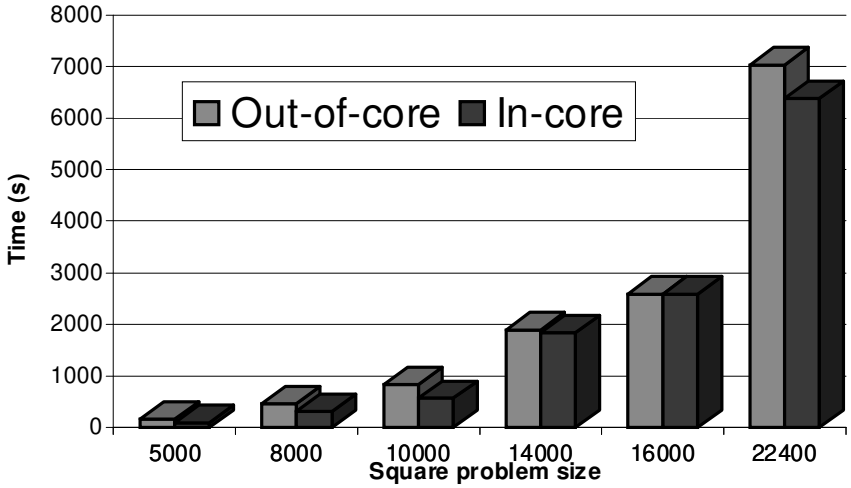
Out of Core Algorithm



- Hybrid approach
- Algorithm is "Left-Looking" in nature, but uses "Right-Looking" (ScaLAPACK) on Panel B
- Latency Tolerant
- Model for deep memory hierarchy algorithms

16

QR Factorization on 64 processors Intel Paragon



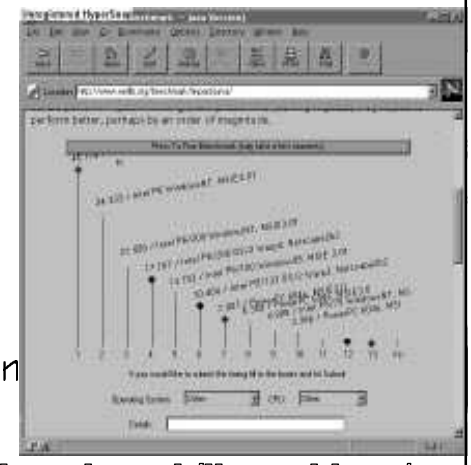
Java

- Java likely to be a dominant language.
- Provides for machine independent code.
- C++ like language
- No pointers or memory deallocation
- Portability achieved via abstract machine
- Java is a convenient user interface builder which allows one to develop quickly customized interfaces.

18

Java for Numerical Computations?

- Few months ago 1 Mflop/s on a 600 Mflop/s processor.
- Top performer today is 46 Mflop/s for a P6 using MS Explorer 3.0 JIT (62 Mflop/s in Fortran).



http://www.netlib.org/benchmark/linpackjava/

19

LAPACK to JAVA

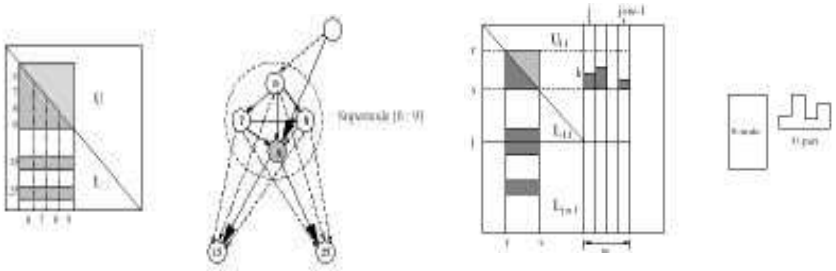
- Working on translator to go from LAPACK to Java Byte Code
  - f2j: formal compiler of a subset of f77 sufficient for BLAS & LAPACK
- Plan to enable all of LAPACK
  - Compiler provides quick, reliable translation.
- Allows Java programmers to access to BLAS/LAPACK routines.
- Focus on LAPACK Fortran
  - Simple - no COMMON, EQUIVALANCE, SAVE, I/O

20

Super-LU

- Algorithm: PA = LU , A sparse and unsymmetric.
- Exploit dense submatrices in the L & U factors of PA = LU .

- Supernodes:
  - Permit use of higher level BLAS.
  - Reduce inefficient indirect addressing.
  - Reduce symbolic time by traversing supernodal graph.



Sparse Gaussian Elimination

- Super LU, supernodal SMP version
- Designed to exploit memory hierarchy
  - Serial Supernode-panel organization permits "BLAS 2.5" performance
  - Up to 40% of machine peak on large sparse matrices on IBM RS6000/590, MIPS R8000, 25% on Alpha 21164

22

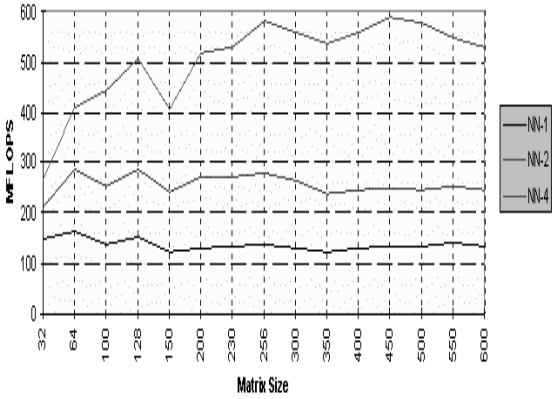
Super LU

- Several data structures redefined,
- Redesigned for parallel access Barrier-less, nondeterministic implementation
- Task queue for load balance (over 95% on most large problems)
- For n=16614 matrix from 3D flow calculation, 66 nonzeros/row

Machine	Nprocs	Speedup	Gflop/s	% Peak
Alpha 8400	8	7	0.78	17
Cray J90	8	6.5	0.83	25
Cray C90	8	6.6	2.58	25

Intel Pentium Pro Cluster

- DGEMM on a 4 processor Pentium Pro 200 MHz



- Cluster of PCs
- Cost for 20 way ~ $70K
- 5x4 way - 256 MB each
- 100 Mb switched
- + Myrinet
  - High BW/Low Latency

24

BLAS Technical Forum

- BLAS Functionality
- Sparse BLAS
- BLAS Lite
- Parallel BLAS
- Language binding issues
  - C Interface to the BLAS
- 32 people at the last meeting
- Next meeting in Portland August 14-15, 1997
- HP-Convex, IBM, Intel, NEC, SGI-CRI, NAG, VN-IMSL plus various Labs and Universities

25

ScaLAPACK - Ongoing Work

- Increasing flexibility and usability
  - Algorithm redistribution in PBLAS
  - Removal of alignment restrictions
  - Algorithmic blocking
- Increased functionality
  - Divide and conquer for SEP and SVD
- Sparse Direct Solver in parallel
- C++ and Java interface
- Fault tolerant aspects

26

ScaLAPACK Summary

- Portability across wide range of architectures
  - Will work on a heterogeneous platform
- ScaLAPACK already in use by SGI/Cray, IBM, HP-Convex, Fujitsu, NEC, Hitachi, NAG, &Visual Numerics (IMSL)
  - Tailor performance & provide support
- Software, examples, prebuilt available in netlib