Advanced Numerical Techniques for Cluster Computing

Presented by

Piotr Luszczek

http://icl.cs.utk.edu/iter-ref/

SC06 POWERFUL BEYOND IMAGINATION

ICL UT INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY of TENNESSEE Computer Science Department

Presentation Outline

- Motivation – hardware
- Dense matrix calculations
- Sparse direct solvers
- Iterative methods
- Other applications

Floating-point Operations Per Cycle

Processor	Single			Double	
	x87	SSE	3DNow!	x87	SSE2
Pentium	1			1	
Pentium II	1			1	
Pentium III	1	4		1	
Pentium 4	1	4		1	2
Athlon	2		4	2	
Athlon4	2	4	4	2	
AthlonMP	2	4	4	2	

- AltiVec, VMX (G4, G5, POWER6)

Speed of Single versus Double Precision

Processor	BLAS	N	sPeak/ dPeak	SGEMM/ /DGEMM	SGETRF/ /DGETRF
IBM Cell 2.4 Ghz	custom	3000	~10	~10	~10
IBM Cell 3.2 Ghz	custom	3000	~10	~10	~10
Intel Pentium III Coppermine	Goto	3500	2	2.10	2.24
Intel Pentium III Katmai	Goto	3000	2	2.12	2.11
Intel Pentium IV Prescott	Goto	4000	2	2.00	1.86
Intel Pentium IV-M Northwood	Goto	4000	2	2.02	1.98
AMD Opteron	Goto	4000	2	1.98	1.93
Cray X1	SciLib	4000	2	1.68	1.54
IBM PowerPC G5 2.7 Ghz	VecLib	5000	2	2.29	2.05
Sun UltraSPARC IIe	Sunperf	3000	2	1.45	1.79
Compaq Alpha EV6	CXML	3000	1	0.99	1.08
IBM SP POWER3	ESSL	3000	1	1.03	1.13
SGI Octane	ATLAS	2000	1	1.08	1.13
Intel Itanium 2	Goto/ATLAS	1500	1	0.71	

Solving System of Linear Equations Ax=b

- Standard technique
- L U *P* = lu(A) $O(n^3)$
- x = U \ (L \ *P* b) $O(n^2)$
- r = b – Ax $O(n^2)$
- Hopefully || r || is small

Accumulate in extra precision hence the name: mixed-precision

- Iterative refinement (old)
- L U *P* = lu(A) $O(n^3)$
- x = U \ (L \ *P* b) $O(n^2)$
- r = b – Ax $O(n^2)$
- while || r || too big do $O(1)$
  - z = U \ (L \ *P* r) $O(n^2)$
  - x = x + z $O(n)$
  - r = b – Ax $O(n^2)$
- end

New Iterative Refinement: Focus on Speed (but keep Accuracy)

- L U *P* = lu(A) $O(n^3)$
- x = U \ (L \ *P* b) $O(n^2)$
- r = b – Ax $O(n^2)$
- while || r || too big do
  - z = U \ (L \ *P* r) $O(n^2)$
  - x = x + z $O(n)$
  - r = b – Ax $O(n^2)$
- Single precision in **black**
- Double precision in red

- Contributions
  - Speed is first priority
    - Accuracy is acceptable
  - Error analysis update
- Caveats
  - Condition number of A must be < 10^8
    - Helps convergence
  - Values of A cannot over/under-flow in single precision

Single versus Double Precision: Revisited

Processor	BLAS	N	sPeak/ dPeak	SGEMM/ /DGEMM	SGETRF/ /DGETRF	DGESV/ DSGESV	# Iter
IBM Cell 2.4 Ghz	custom	3000	~10	~10	~10	~8	4
IBM Cell 3.2 Ghz	custom	3000	~10	~10	~10	~8	4
Intel Pentium III Coppermine	Goto	3500	2	2.10	2.24	1.92	4
Intel Pentium III Katmai	Goto	3000	2	2.12	2.11	1.79	4
Intel Pentium IV Prescott	Goto	4000	2	2.00	1.86	1.57	5
Intel Pentium IV-M Northwood	Goto	4000	2	2.02	1.98	1.84	5
AMD Opteron	Goto	4000	2	1.98	1.93	1.53	5
Cray X1	SciLib	4000	2	1.68	1.54	1.38	7
IBM PowerPC G5 2.7 Ghz	VecLib	5000	2	2.29	2.05	1.24	5
Sun UltraSPARC IIe	Sunperf	3000	2	1.45	1.79	1.58	4
Compaq Alpha EV6	CXML	3000	1	0.99	1.08	1.01	4
IBM SP POWER3	ESSL	3000	1	1.03	1.13	1	3
SGI Octane	ATLAS	2000	1	1.08	1.13	0.91	4
Intel Itanium 2	Goto/ATLAS	1500	1	0.71			

Single vs. Double Precision in Parallel

- Cluster of dual AMD Opteron processors
- BLAS: Goto
- MPI: OpenMPI
- Interconnect: Myricom MX

- Notice very large N
  - Factorization dominates the time (N^3)
  - Iterative steps of negligible cost (N^2)

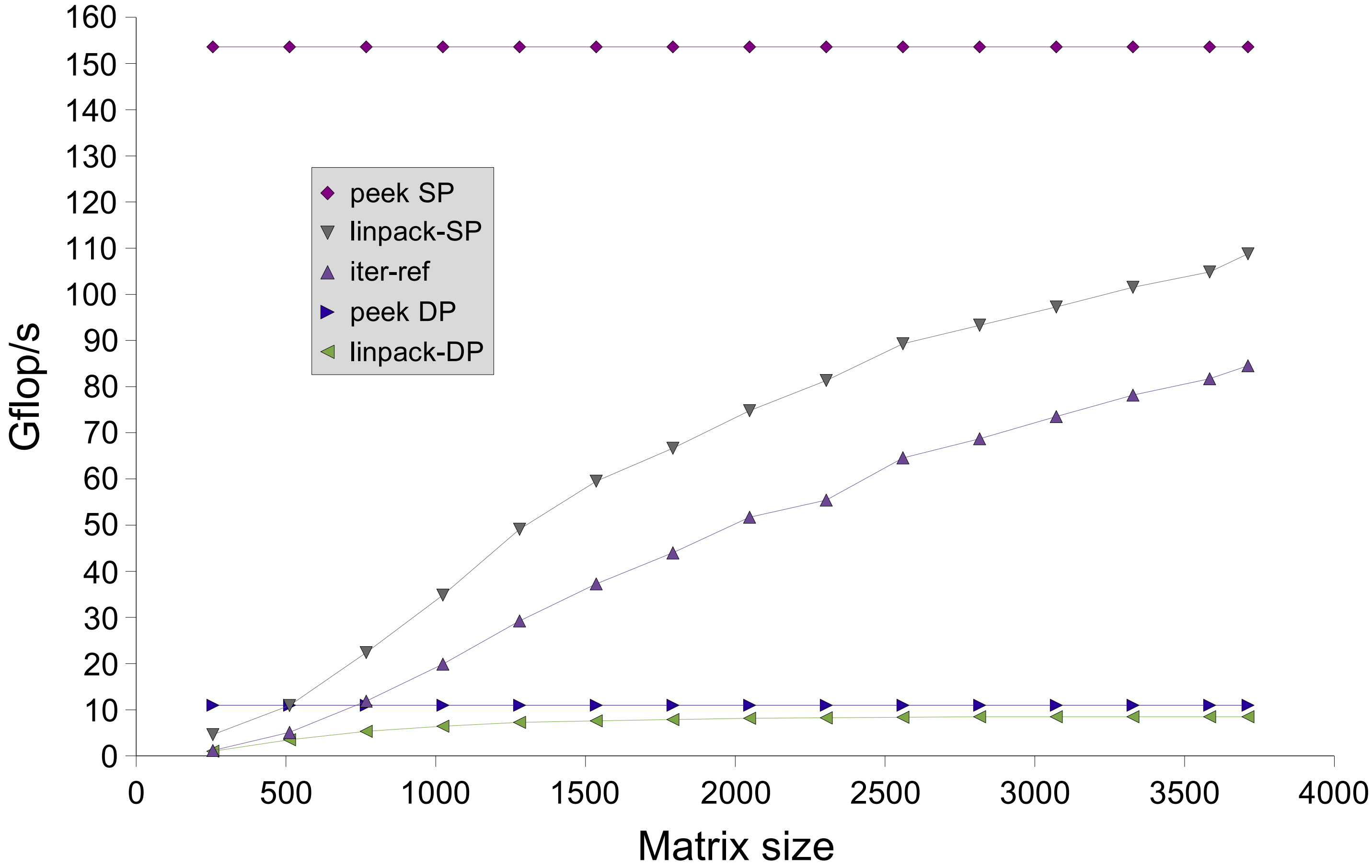
Proc	N	PDGETRF/ PSDGETRF	PDGESV/ PSDGESV	# Iter
32	22627	1.85	1.79	6
64	32000	1.9	1.83	6

Using Quad-Precision: Single=64 bits, Double=128 bits

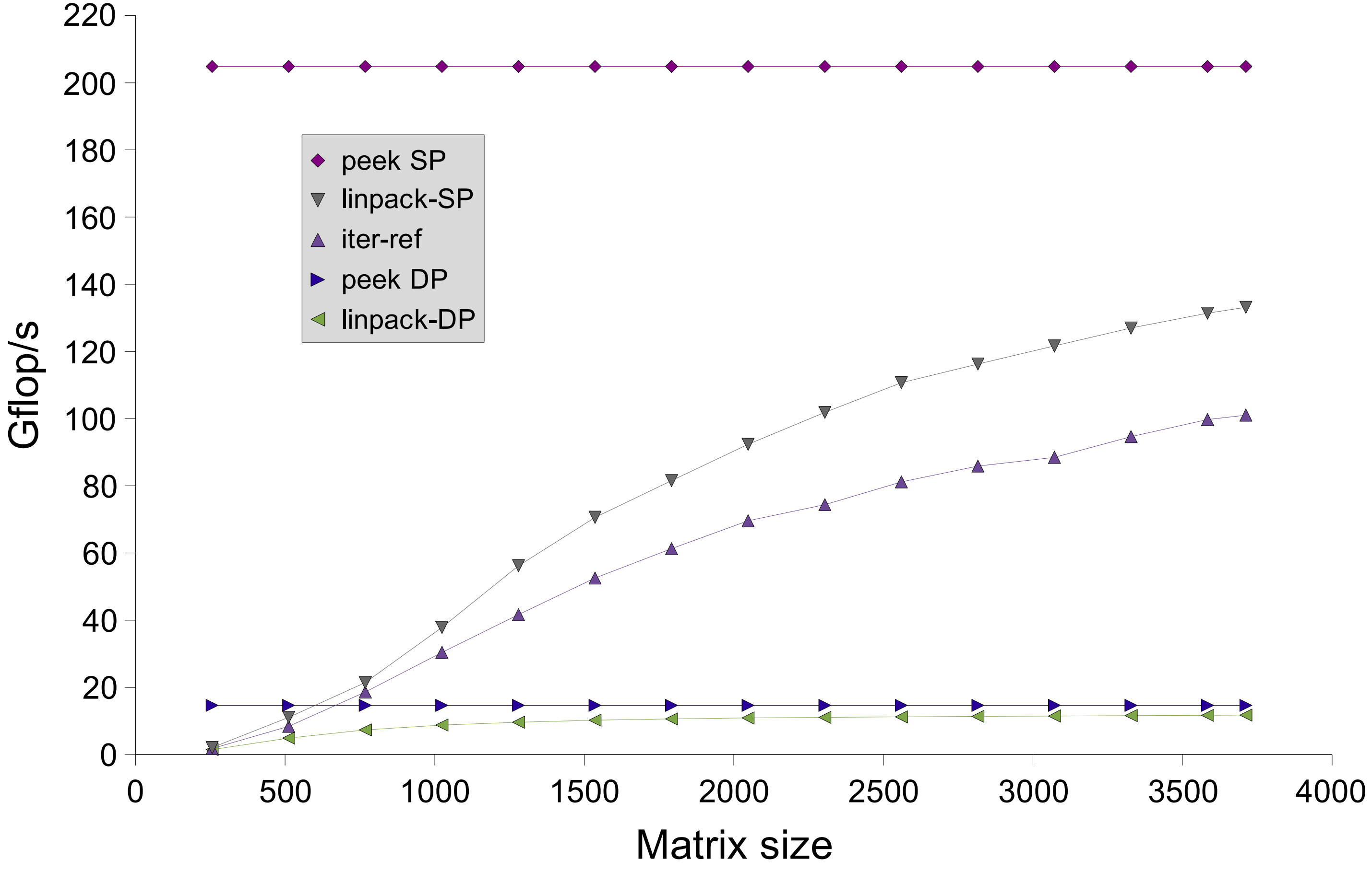
	Time (s)		
N	QGESV	QDGESV	Speedup
100	0.29	0.03	9.67
200	2.27	0.10	22.70
300	7.61	0.24	31.71
400	17.81	0.44	40.48
500	34.71	0.69	50.30
600	60.11	1.01	59.51
700	94.95	1.38	68.80
800	141.75	1.83	77.46
900	201.81	2.33	86.61
1000	276.94	2.92	94.84

- Hardware/software:
  - Intel Xeon 3.2 GHz
  - Intel Fortran compiler
  - Reference BLAS
- Expected accuracy: 32 decimal digits
- Up to 3 iterations needed

Running on IBM Cell Blade @ 2.4 GHz



Running on IBM Cell Blade @ 3.2 GHz



Linpack Report 11/4/2006, Page 76

Computer (Full Precision)	Processors or Cores	Rmax Gflop/s	Nmax Order	N1/2 Order	Rpeak Gflop/s
Sun Fire 6900 (UltraSPARC IV, 1.2 Ghz)	48	98.26	96116	8300	115.2
IBM Cell BE (3.2 Ghz) *****	9	98.05	4096	1536	14.6 (64 bit) 204.8 (32 bit)
SGI Altix 3000, 900 Mhz	32	97.67	82079	82079	115

- Linpack matrices are uniformly random centered around 0:
  - Distribution: U(0,1)
- Condition number ~ $\sqrt{N}$
- Mixed-precision approach works
- Mixed-precision not legal for TOP500 and HPCC submissions
  - Distorts historical data

Applying Mixed-Precision to Sparse Direct Solvers

- 41 real-world matrices
  - Tim Davis' collection
- Varying
  - Disciplines
  - Structure
  - Conditioning ($10\text{-}10^{16}$)
- Solvers
  - MUMPS
    - Multifrontal (mat-mat)
  - SuperLU (sequential)
    - Supernodal (mat-vec)
- Machines
  - AMD Opteron 246
  - IBM PowerPC
  - Intel Core Duo, Pentium III, Xeon, Woodcrest
  - Sun UltraSPARC IIe
- Speedup
  - MUMPS: 7 - 150%
  - SuperLU: 0 - 30%
- Accuracy (2 failures)
  - Same as full double*

* Can do better if slightly less accurate

Application To Iterative Methods

- Original Idea
  - Preconditioned Richardson
  - $x_{k+1} = (I - A_S \backslash A) x_k + A_S \backslash b$
  - Lower precision factorization is an ideal preconditioner
- Doesn't work for real-world sparse matrices
  - Large condition number
  - Forming lower precision factors is too expensive
- Basic idea:
  - Use single precision (mostly) in preconditioner
  - Use inner-outer iteration to embed an iterative method as a lower precision solver
- PCG-PCG
  - Inner PCG carries the computational load
- GMRES-FGMRES
  - Need to accommodate non-const preconditioner

Mixed-Precision Iterative Methods' Results

- Model problem
  - Adaptive discretization of a 3D linear elasticity problem on a tetrahedral mesh with piecewise linear elements

N	NNZ	Log10(Cond)
11,142	442,225	3
25,980	1,061,542	4
79,275	3,374,736	4
230,793	9,991,028	5
602,091	26,411,323	5

- Speedup
  - Standard method
    - 50-120%
  - Mixed-precision PCG
    - 40-100%
  - Mixed-precision GMRES
    - 40-90%

Mixed-Precision Refinement Technique

- Linear Systems
  - LU (dense and sparse)
  - Cholesky, Symmetric
  - QR
- Eigenvalue
  - Symmetric/hermitian
  - SVD
  - Reduce to special form in lower precision improve in higher (N^2 per value/vector)
- Iterative methods
  - Relaxed GMRES
  - Inner/outer iteration

Collaborators

- Alfredo Buttari
- Jack Dongarra
- Jakub Kurzak
- Julie Langou
- Julien Langou
- Stanimire Tomov

http://icl.cs.utk.edu/iter-ref/