Parallel Sparse Linear Solvers For Process Engineering Problems on a Cluster of Workstations

Youdong Lin[1], Jennifer A. Scott[2], and Mark A. Stadtherr[1]

[1]Department of Chemical Engineering
University of Notre Dame
Notre Dame, IN, USA

[2]Computational Science and Engineering Department
Rutherford Appleton Laboratory
Oxon, England

AIChE Annual Meeting, Reno, NV
November 2001

Objective

Test performance of new parallel multiple front solver (HSL routine MP43) on process engineering matrices using a cluster of workstations.

Outline

- Overview
  - Process Engineering Problems
  - Frontal Method
  - Parallel Multiple Front Method
- Parallel Multiple Front Solver (MP43)
- Results and Discussion
- Concluding Remarks

Process Engineering Problems

- Realistically complex process simulation and optimization problems typically require large-scale computation.
- When an equation-based problem formulation is used, a key computational bottleneck is often the solution of large, sparse linear equation systems (may be as much as 80-90% of total simulation time).
- Properties of process engineering matrices:
  - Very sparse
  - Very unsymmetric (structurally)
  - Numerically indefinite
  - Not diagonally dominant
  - May be ill-conditioned

Process Engineering Problems (cont'd)

- Solve A$\mathbf{x}$ = $\mathbf{b}$, where A is large, sparse and has highly asymmetric structure.
- General-purpose direct solvers (e.g., the HSL routine MA48 of Duff and Reid, 1993) typically used
  - Factor: $PAQ = LU$ (P and Q represent row and column permutations)
  - Solve: $L\mathbf{y} = P\mathbf{b}$

    $U\mathbf{z} = \mathbf{y}$

    $\mathbf{x} = Q\mathbf{z}$

  using row- or column-oriented Gaussian elimination with threshold pivoting to obtain LU factors.
- Frontal elimination is an attractive alternative for a wide range of modern computer architectures.

Frontal Method (Irons, 1970; Hood, 1976; Duff, 1979, 1980)

- Basic idea: Restrict computations to a relatively small *front* (or *frontal matrix*).
  - Allows use of high-level BLAS optimized for machine architecture
  - If desired, can be implemented to use only small amount of main memory
- Applied to process engineering problems on vector/parallel machines by Vegeais and Stadtherr (1985,1990).
- FAMP code (Zitney and Stadtherr, 1993) was used in CRAY versions of commercial process simulation codes (e.g. SPEEDUP, ASPEN PLUS).
- Today, the mathematical subroutine library HSL provides MA42 (Duff and Scott, 1992), a general-purpose frontal solver for elements or assembled problems.

Frontal Method (cont'd)

- Frontal matrix sizes, and thus overall computational performance, depend strongly on row ordering.
- Recent advances in row ordering techniques have greatly improved the performance of frontal solvers on general highly unsymmetric problems.
  - RMCD algorithm (Camarda and Stadtherr, 1998)
  - MSRO algorithm (Scott, 1999, 2000) is highly effective on most problems
- MRSO is implemented in HSL routine MC62, which is used here with MA42 and MP43.

Parallel Multiple Front Solver

- Use singly-bordered block-diagonal (BoBD) form

$$A = \begin{bmatrix} A_{11} & & & & C_1 \\ & A_{22} & & & C_2 \\ & & \cdots & & \cdots \\ & & & A_{NN} & C_N \end{bmatrix}$$

A_{ii} are $m_i \times n_i$

C_i are $m_i \times k$

- Variations based on using bottom border or double border are also possible.
- Recent advances in ordering to BoBD form
  - GPA-SUM (Camarda and Stadtherr, 1999)
  - MONET (Hu et al., 2000); available in HSL 2002 as routine MC66

Parallel Multiple Front Solver (cont'd)

- Frontal elimination can be applied to each (A_{ii} C_i) independently and in parallel.
- After parallel frontal elimination, contributions from each of the N subproblems are assembled into a $k \times k$ "interface matrix" F.
- After factorization of F, do block forward elimination and back substitution (in parallel) to complete solution.

MP43 (Scott, 2001 – to appear in Comput. Chem. Eng.)

Software for implementing parallel multiple front method.

- Fortran 90
- MPI for message passing
- One processor designated as the “host”
- Highly portable: shared or distributed memory
- Use for general unsymmetric sparse linear systems that have been put into BoBD form
- Available in HSL 2000

Outline of MP43 algorithm (Scott, 2001)

- Initialize (serial): Host processor distributes each subproblem to one of P processors.
- Analyze (parallel, optional): Generate row ordering (MC62) for subproblems and send approximate FLOP count to host. Host may redistribute workload for better balance.
- Factor
  - (parallel) Perform frontal elimination (MA42) with partial pivoting, storing computed columns of L and rows of U and sending contribution to F to host.
  - (serial) Host performs frontal elimination on interface matrix F (MA42).
- Solve
  - (parallel) Forward elimination on subproblems.
  - (serial) Forward elimination and back substitution on interface problem.
  - (parallel) Back substitution on subproblems.

Numerical Experiments

- Hardware: Cluster of Sun Ultra 2/2400 nodes (2 CPUs/node) connected by 100 Mbps switched Ethernet. Experiments done using both 1 CPU/node and 2 CPUs/node.
- Test problems: Several process simulation matrices from various sources. Results given here for 5 typical problems.
- Software:
  - MA48: default control parameters
  - MA42: default control parameters; MC62 row ordering; min. pivot block size = 8
  - MP43: default control parameters (min. pivot block size = 8); MC62 row ordering; MONET ordering to BoBD form with 8 blocks (not included in timings)
- All timings are wallclock times in seconds for complete Analyze, Factor and Solve (A/F/S).
- Similar studies done by Scott (2001) on SGI Origin 2000, Cray T3E-1200E, Compaq DS20.

Results: Problem *bayer01*

- n = 57,735 nz = 159,082
- Source: Bayer AG
- MONET ordering: N = 8; k = 295 (0.51%); imbalance = 3.80%
- Timing results (1 CPU/node)

		MP43			
MA48	MA42	P=1	2	4	8
6.97	5.96	4.18	2.53	1.86	1.53

- Parallel speedup

P=	2	4	8
1 CPU/node	1.65	2.25	2.73
2 CPU/node	1.77	2.39	2.90
SGI Origin 2000 (Scott, 2001)	1.77	2.69	4.43

Results: Problem *lhr71c*

- n = 70,304 nz = 1,528,092
- Source: SEQUEL simulation
- MONET ordering: N = 8; k = 1251 (1.78%); imbalance = 2.09%
- Timing results (1 CPU/node)

		MP43			
MA48	MA42	P=1	2	4	8
62.73	29.68	29.48	17.51	11.94	9.80

- Parallel speedup

P=	2	4	8
1 CPU/node	1.68	2.47	3.01
2 CPU/node	1.80	2.69	3.12
SGI Origin 2000 (Scott, 2001)	1.83	3.23	4.94

Results: Problem *10cols*

- n = 29,496   nz = 109,588
- Source: ASCEND simulation
- MONET ordering: N = 8; k = 315 (1.07%); imbalance = 1.03%
- Timing results (1 CPU/node)

		MP43			
MA48	MA42	P=1	2	4	8
8.22	1.59	2.02	1.20	0.86	1.35

- Parallel speedup

P=	2	4	8
1 CPU/node	1.68	2.35	1.50
2 CPU/node	1.82	2.49	2.92
SGI Origin 2000 (Scott, 2001)	1.72	3.03	4.54

Results: Problem *bayer04*

- n = 20,545 nz = 159,082
- Source: Bayer AG
- MONET ordering: N = 8; k = 439 (2.14%); imbalance = 15.49%
- Timing results (1 CPU/node)

		MP43			
MA48	MA42	P=1	2	4	8
3.32	2.38	2.33	1.61	1.14	1.57

- Parallel speedup

P=	2	4	8
1 CPU/node	1.44	2.04	1.38
2 CPU/node	1.62	2.20	2.56
SGI Origin 2000 (Scott, 2001)	1.74	2.86	4.56

Results: Problem *ethylene-1*

- n = 10,673 nz = 80,904
- Source: NOVA simulation
- MONET ordering: N = 8; k =162 (1.52%); imbalance = 13.93%
- Timing results (1 CPU/node)

		MP43			
MA48	MA42	P=1	2	4	8
0.80	6.88	1.11	0.73	0.57	0.45

- Parallel speedup

P=	2	4	8
1 CPU/node	1.52	1.95	2.47
2 CPU/node	1.68	2.22	2.84
SGI Origin 2000 (Scott, 2001)	1.65	2.70	4.60

Discussion

- With good row ordering (MC62), frontal solvers (MA42, MP43) are competitive with MA48 for A/F/S on one processor.
- Speedups limited by
  - Solution of interface problem (serial)
  - Any workload imbalance
  - Communication overhead
- MONET is remarkably effective in producing a very small interface problem.
- Lower communication speed in workstation cluster (relative to Origin 2000) leads to lower speedups.
- For problems of this size, up to 4 processors can be used effectively on this cluster, and up to 4-8 processors on the Origin 2000.
- MONET reordering time must be amortized over several factorizations.

Discussion (cont'd)

- Timings for solve only (with previously computed L and U factors) were also obtained.
- For single processor, MA48 is almost always fastest for solve only (due to sparser L and U factors compared to frontal factorization).
- For this cluster, speedups for solve only with MP43 are typically poor. Ratio of communication to computation is much higher than in A/F/S.
- For Origin 2000, useful speedups for solve only with MP43 are observed (Scott, 2001), but are less than for A/F/S.

Concluding Remarks

- With a good row ordering, frontal solvers can provide a powerful and competitive alternative to general-purpose sparse solvers for chemical process applications.
- MP43 is a general-purpose, highly portable, multiple front code for solving large sparse unsymmetric linear systems in parallel.
- For problems of the size tested, MP43 provides good speedup on up to 4 processors for the cluster used, and up to 8 processors for an Origin 2000.
- With increasingly affordable gigabit connectivity (1000 Mbps) for clusters, improved speedups for workstation clusters is likely.

Concluding Remarks (cont'd)

- Information on obtaining HSL routines is available from http://www.cse.clrc.ac.uk/Activity/HSL
- A copy of these slides will be available after the conference at http://www.nd.edu/~markst/presentations.html