BLUE WATERS

SUSTAINED PETASCALE COMPUTING

# Performance Test of Parallel Linear Equation Solvers on Blue Waters – Cray XE6/XK7 system

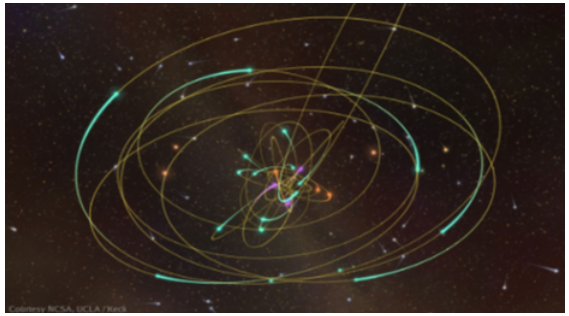
JaeHyuk Kwack, Gregory H Bauer, and Seid Koric
National Center for Supercomputing Applications

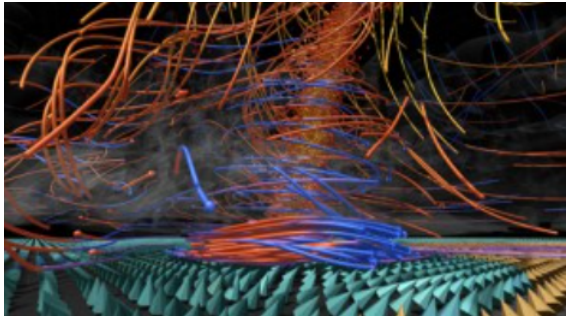
NCSA

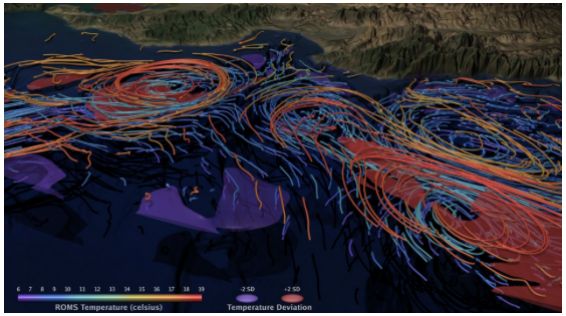

GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION

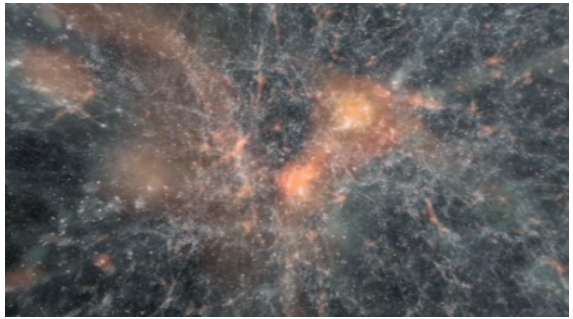
CRAY

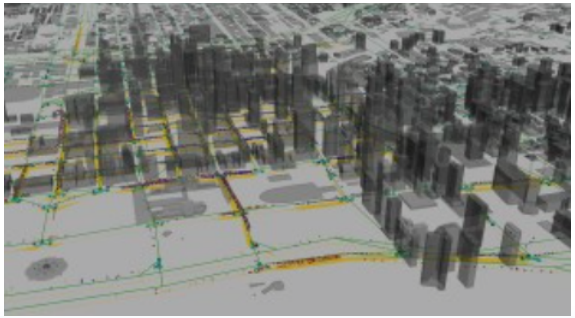
# The best linear equation solver for your simulations?

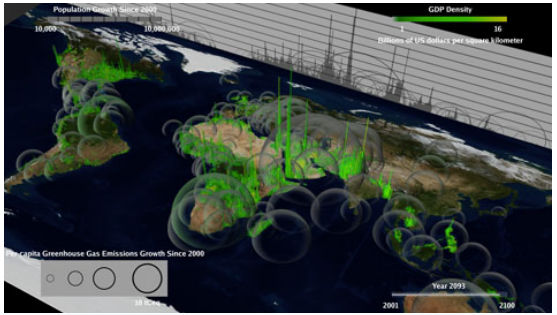








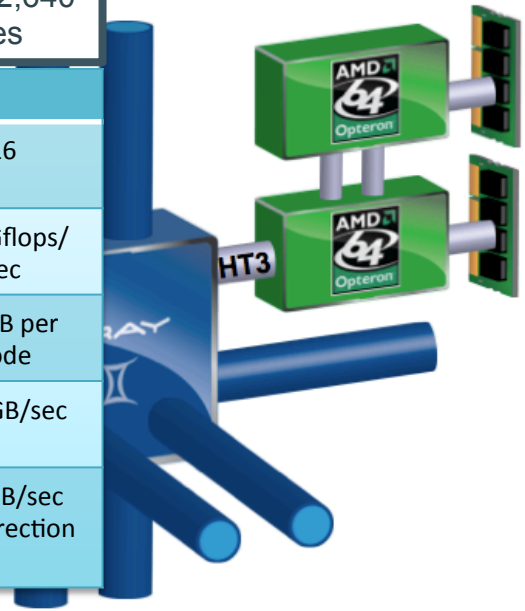



Source: NCSA's Advanced Visualization Laboratory (https://avi.ncsa.illinois.edu)

# Blue Waters XE6/XK7 node

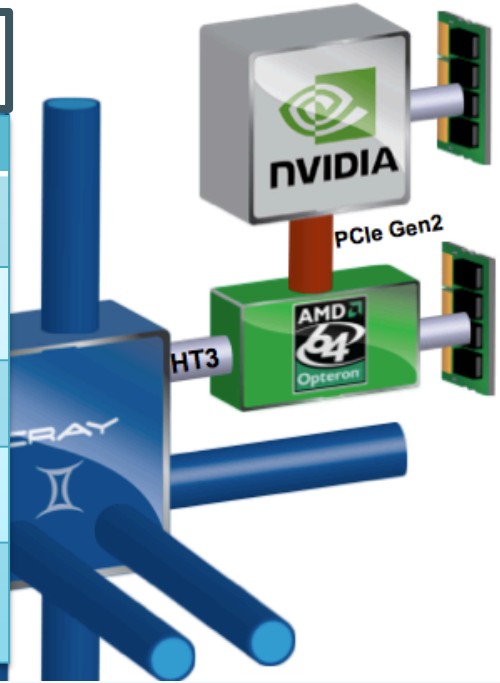
Blue Water contains 22,640 XE6 compute nodes

| Node Characteristics | |
|---|---|
| Number of Core Modules* | 16 |
| Peak Performance | 313 Gflops/ sec |
| Memory Size | 64 GB per node |
| Memory Bandwidth (Peak) | 102 GB/sec |
| Interconnect Injection Bandwidth (Peak) | 9.6 GB/sec per direction |




Blue Waters contains 4,224 NVIDIA Kepler (GK110) GPUs

| XK7 Compute Node Characteristics | |
|---|---|
| Host Processor | AMD Series 6200 (Interlagos) |
| Host Processor Performance | 156.8 Gflops |
| Kepler Peak (DP floating point) | 1.32 Tflops |
| Host Memory | 32GB 51 GB/sec |
| Kepler Memory | 6GB GDDR5 capacity > 180 GB/sec |



# Non-symmetric matrices from a CFD simulation

## Incompressible Navier-Stokes equations

$$\rho \boldsymbol{v}_{,t} + \rho \boldsymbol{v}\cdot\nabla\boldsymbol{v} - \nabla\cdot\boldsymbol{\sigma}_v(\boldsymbol{v}) + \nabla p = \rho \boldsymbol{f} \qquad \text{in } \Omega \times ]0,T[$$

$$\nabla\cdot\boldsymbol{v} = 0 \qquad \text{in } \Omega \times ]0,T[$$

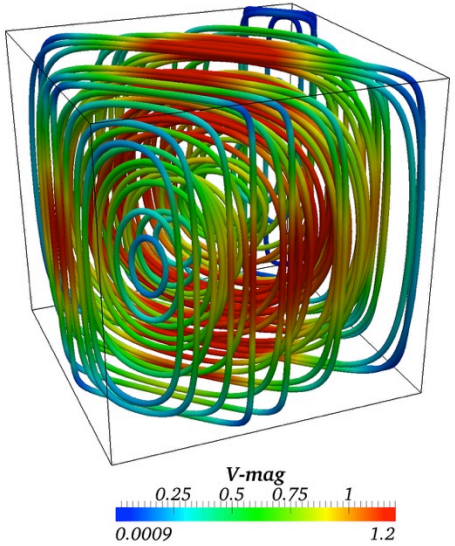
$$\boldsymbol{v} = \boldsymbol{g} \qquad \text{on } \Gamma_g \times ]0,T[$$

$$\boldsymbol{\sigma}\cdot\boldsymbol{n} = (\boldsymbol{\sigma}_v(\boldsymbol{v}) - p\boldsymbol{I})\cdot\boldsymbol{n} = \boldsymbol{h} \qquad \text{on } \Gamma_h \times ]0,T[$$

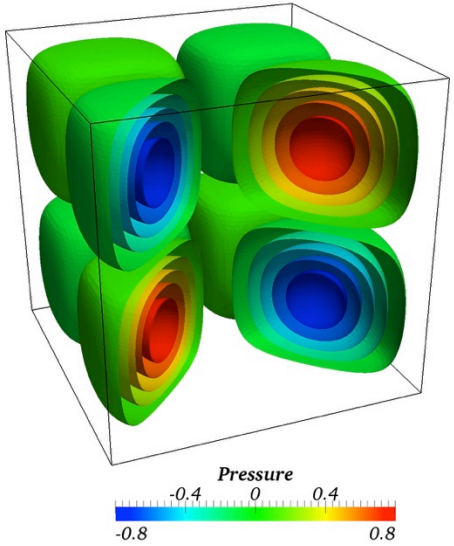
$$\boldsymbol{v}(\boldsymbol{x},0) = \boldsymbol{v}_0 \qquad \text{on } \Omega \times \{0\}$$

- $\boldsymbol{v}$ := velocity vector
- $p$ := pressure
- $\boldsymbol{\varepsilon}(\boldsymbol{v}) := (\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^{\mathrm{T}})/2$ := the rate-of-deformation tensor
- $\boldsymbol{\sigma}_v = 2\eta\boldsymbol{\varepsilon}(\boldsymbol{v})$ := viscous stress tensor

## Velocity streamlines



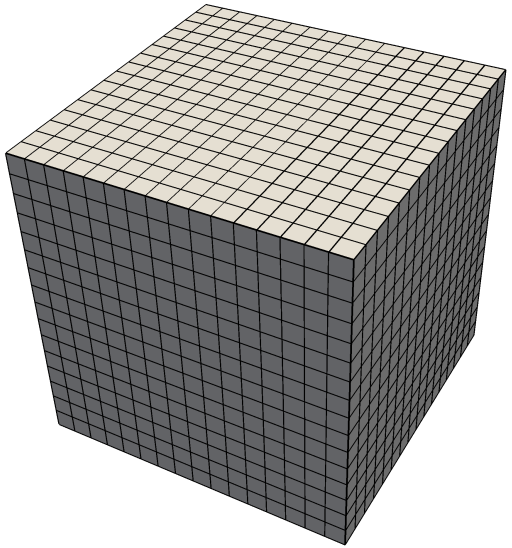

## Pressure contours



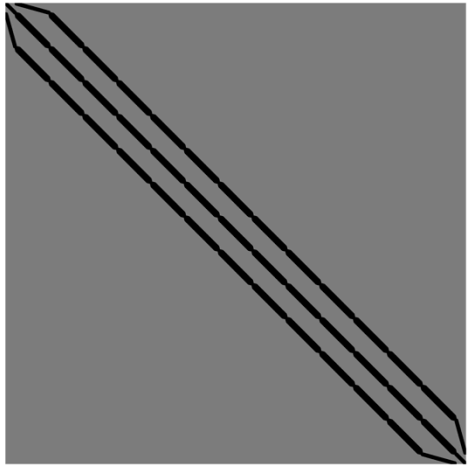
# Non-symmetric matrices from a CFD simulation

| Matrix | Number of finite elements | Number of equations | Number of non-zero components | Sparsity (Density) (%) | Condition number* | File size in CSR format |
|---|---|---|---|---|---|---|
| LHM14 | $14^3$ | 9,965 | 891,083 | 0.8974 | 1.66E+04 | 32 MB |
| LHM20 | $20^3$ | 29,837 | 2,835,299 | 0.3185 | 6.33E+04 | 102 MB |
| LHM24 | $24^3$ | 52,125 | 5,066,723 | 0.18648 | 1.25E+05 | 181 MB |
| LHM28 | $28^3$ | 83,437 | 8,239,907 | 0.11836 | 2.25E+05 | 294 MB |
| LHM30 | $30^3$ | 102,957 | 10,231,499 | 0.09652 | 2.94E+05 | 365 MB |
| LHM32 | $32^3$ | 125,309 | 12,520,739 | 0.07974 | 3.75E+05 | 447 MB |
| LHM36 | $36^3$ | 179,277 | 18,075,107 | 0.05624 | 5.91E+05 | 645 MB |
| LHM46 | $46^3$ | 377,197 | 38,621,387 | 0.02715 | 1.54E+06 | 1.4 GB |
| LHM56 | $56^3$ | 684,317 | 70,756,067 | 0.015109 | 3.33E+06 | 2.5 GB |
| LHM60 | $60^3$ | 843,117 | 87,435,299 | 0.012300 | 4.38E+06 | 3.1 GB |
| LHM62 | $62^3$ | 930,989 | 96,677,579 | 0.011154 | 4.99E+06 | 3.4 GB |
| LHM64 | $64^3$ | 1,024,756 | 106,549,283 | 0.010146 | 5.65E+06 | 3.8 GB |
| LHM66 | $66^3$ | 1,124,637 | 117,071,147 | 0.009256 | 6.39E+06 | 4.1 GB |
| LHM68 | $68^3$ | 1,230,797 | 128,263,907 | 0.008467 | 7.19E+06 | 4.5 GB |
| LHM70 | $70^3$ | 1,343,437 | 140,148,299 | 0.007765 | 8.07E+06 | 4.9 GB |
| LHM80 | $80^3$ | 2,010,557 | 210,670,499 | 0.005212 | 1.37E+07 | 7.4 GB |
| LHM112 | $112^3$ | 5,545,789 | 586,210,979 | 0.0019060 | 5.27E+07 | 21 GB |

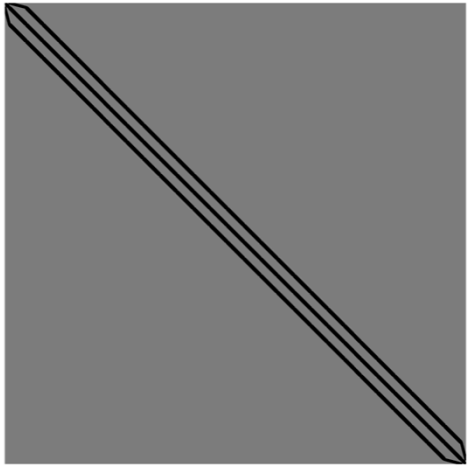
* Condition number is estimated via the error analysis routine with full statics of MUMPS (i.e., ICNTL(11) is set to 1)



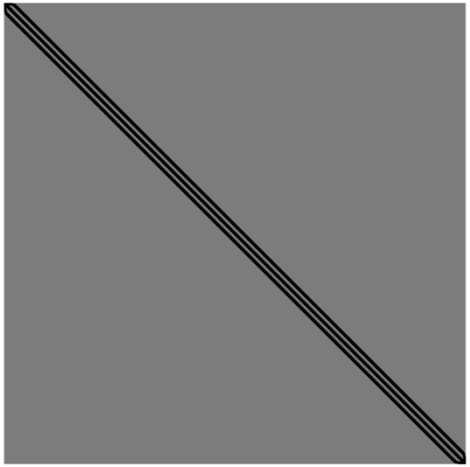
# Sparsity pattern of non-symmetric matrices



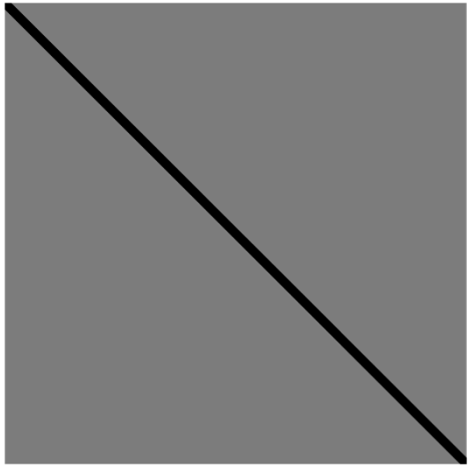
LHM14 – 10K eqns



LHM28 – 83K eqns



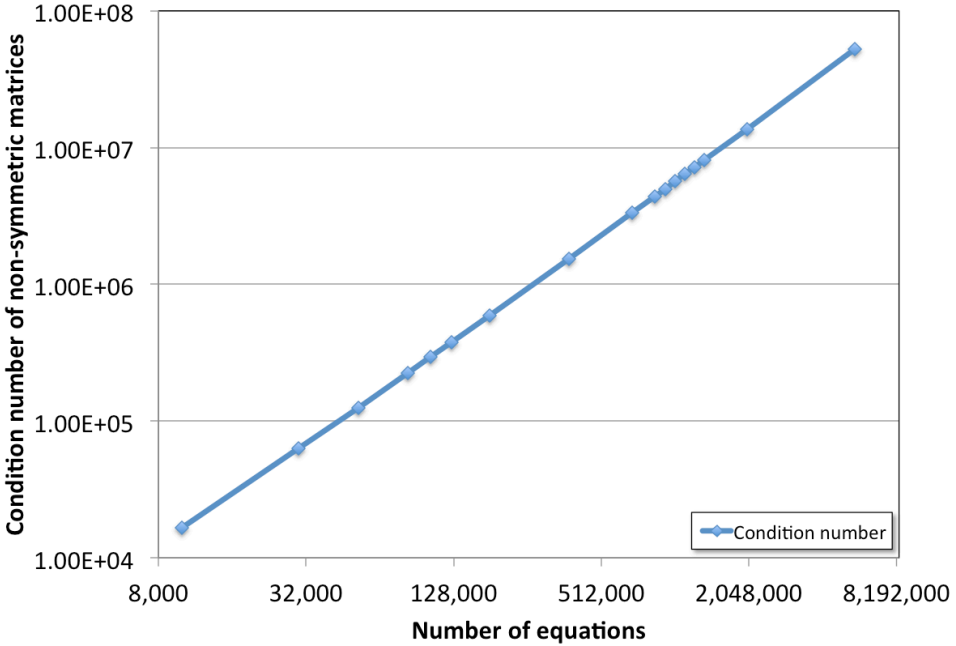
LHM56 – 684K eqns



LHM112 – 5.5M eqns

# Condition numbers of non-symmetric matrices

- Condition number increase along the mesh-refinement
- 16,600 for LHM14
- 52.7 M for LHM112
- Around 3,000 times increments



# Employed Linear Equation Solver Libraries

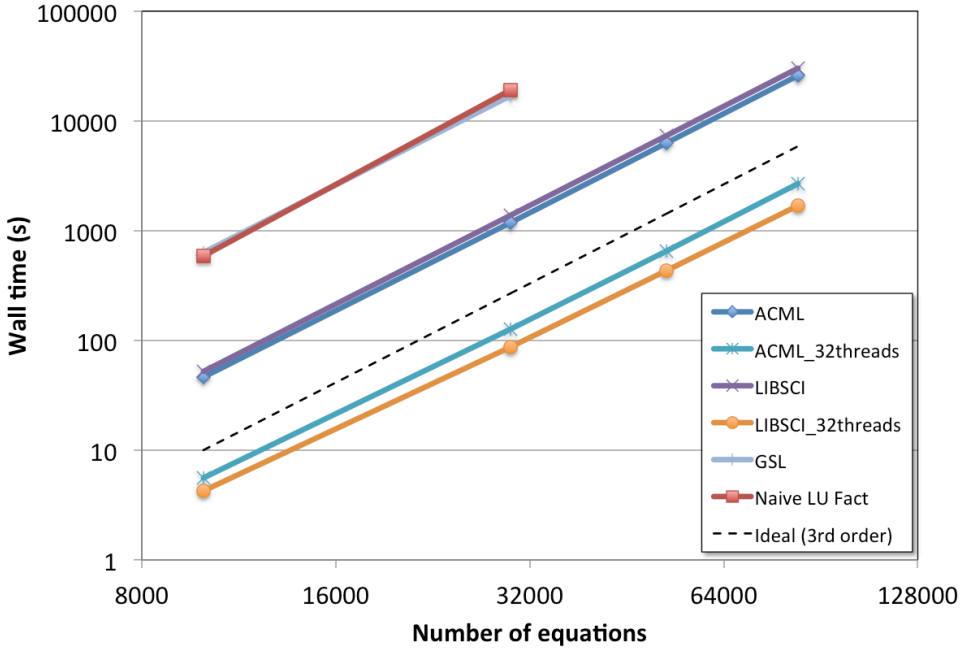
- Cray modules
  - Cray LibSci version 13.3.0 for an optimized LAPACK with SMP performance on Cray XE nodes,
  - Cray PETSc version 3.6.1.0 for KSP Linear Equations Solvers on Cray XE nodes,
  - Cray TPSL version 1.5.2 for MUMPS, SuperLU_DIST and ParMetis on Cray XE nodes,
  - Intel Composer XE version 15.0.3.187 for Intel MKL PARDISO solver on Cray XE nodes,
  - ACML version 5.1.1 for an optimized LAPACK with SMP performance on Cray XE nodes,
  - GSL version 1.16-2015-04 for LAPACK on Cray XE nodes,
  - NVIDIA cudatoolkit version 7.0.28-1.0502.10742.5.1 for cuSolver on Cray XK nodes
- User-built libraries
  - SuperLU_DIST version 4.3 for MPI on Cray XE nodes,
  - MUMPS version 5.0.0 for MPI on Cray XE nodes,
  - IBM WSMP version 16.01.10 on Cray XE nodes,
  - NVIDIA AmgX version 1.2.1-build112 and 1.2.0-build108 on Cray XK nodes.

# SINGLE-NODE TEST RESULTS FOR THE BEST SMP PERFORMANCE

# Dense matrix direct solvers on an XE node

| Matrix | Number of equations | Cray LibSci | | | ACML | GSL | Naïve LU |
|---|---|---|---|---|---|---|---|
| | | 1 thread | 32 threads* | Speedup | | | |
| LHM14 | 9,965 | 52.13 s | 4.23 s | 12.3 | | | |
| LHM20 | 29,837 | 1373.7 s | 86.27 s | 15.9 | | | |
| LHM24 | 52,125 | 7325.4 s | 429.7 s | 17.0 | | | |
| LHM28 | 83,437 | 30017 s | 1689.2 s | 17.8 | | | |

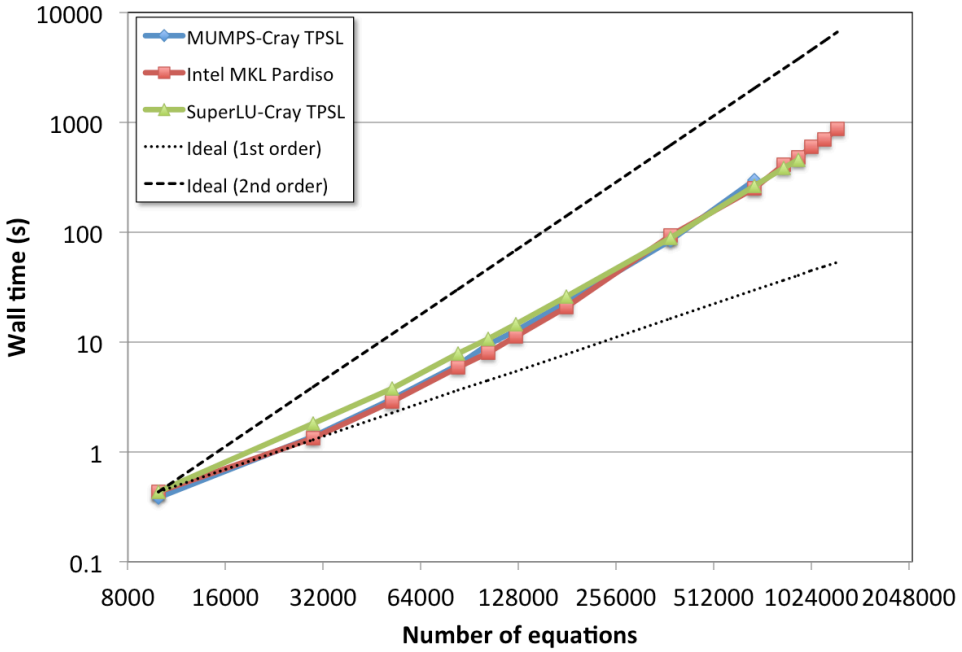
* one thread per one integer core (two threads per one Bulldozer core)



# Sparse matrix direct solvers on an XE node

| Matrix | Number of equations | MUMPS (Cray TPSL) | Intel MKL PARDISO | SuperLU-DIST (Cray TPSL) |
|---|---|---|---|---|
| LHM14 | 9,965 | 0.3808 s | 0.4317 s | 0.4348 s |
| LHM20 | 29,837 | 1.3864 s | 1.333 s | 1.7960 s |
| LHM24 | 52,125 | 3.027 s | 2.860 s | 3.776 s |
| LHM28 | 83,437 | 6.170 s | 5.889 s | 7.790 s |
| LHM30 | 102,957 | 9.485 s | 8.051 s | 10.689 s |
| LHM32 | 125,309 | 12.650 s | 11.142 s | 14.564 s |
| LHM36 | 179,277 | 22.85 s | 20.79 s | 25.79 s |
| LHM46 | 377,197 | 83.91 s | 92.52 s | 88.41 s |
| LHM56 | 684,317 | 295.2 s | 248.3 s | 264.9 s |
| LHM60 | 843,117 | OOM | 408.1 s | 385.8 s |
| LHM62 | 930,989 | OOM | 477.8 s | 455.2 s |
| LHM64 | 1,024,756 | OOM | 601.5 s | OOM |
| LHM66 | 1,124,637 | OOM | 696.9 s | OOM |
| LHM68 | 1,230,797 | OOM | 872.2 s | OOM |

OOM: Out of memory during the factorization process

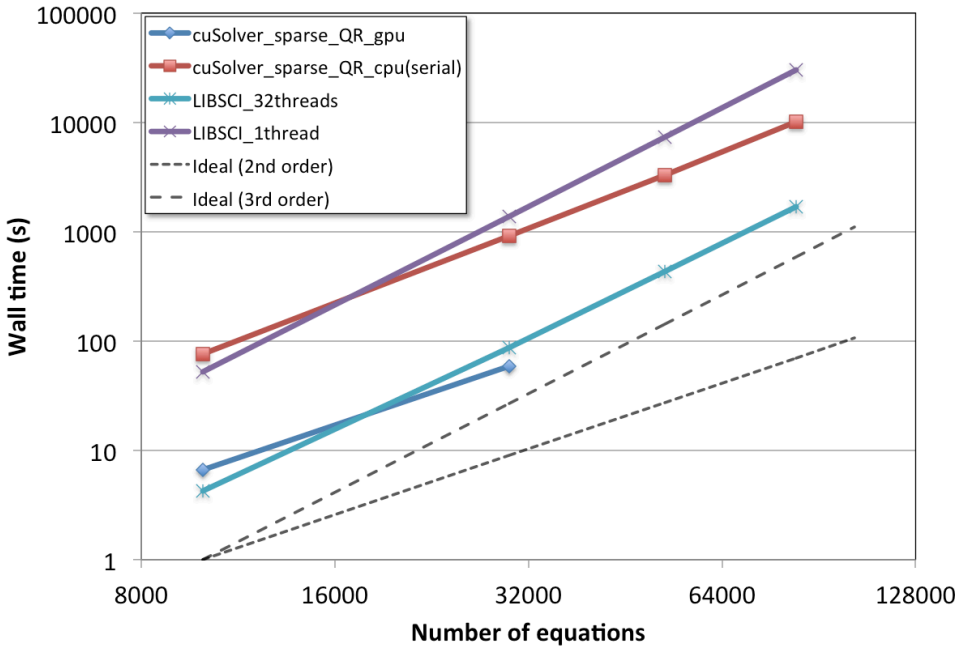


# Direct solvers on an XK node - cuSOLVER

| cuSolver_dense_QR on GPU |
|---|
| step 1: copy matrices (i.e., A) and residual vectors (i.e., B) to device (i.e., GPU) |
| step 2: solve A*x=B on GPU (call cusolverDnDgeqrf, cusolverDnDormqr, and cu |
| step 3: copy solution vectors (i.e., x) to host (i.e., CPU) |
| **cusolver_sparse_QR on GPU** |
| step 1: copy matrices (i.e., A) and residual vectors (i.e., B) to device (i.e., GPU) |
| step 2: solve A*x = b on GPU (i.e., call cusolverSpDcsrlsvqr subroutine on device |
| step 3: copy solution vectors (i.e., x) to host (i.e., CPU) |
| **cusolver_sparse_QR on CPU** |
| step 1: solve A*x = b on CPU (i.e., call cusolveSpDcsrlsvqrHost subroutine on ho |

| Matrix | Number of equations | Dense QR on GPU | Sparse QR on GPU | Sparse QR or CPU (1 thread |
|---|---|---|---|---|
| LHM14 | 9,965 | 7.04 s | 6.63 s | 76.1 s |
| LHM20 | 29,837 | OOM | 58.5 s | 921.2 s |
| LHM24 | 52,125 | OOM | OOM | 3303.2 s |
| LHM28 | 83,437 | OOM | OOM | 10165.6 s |

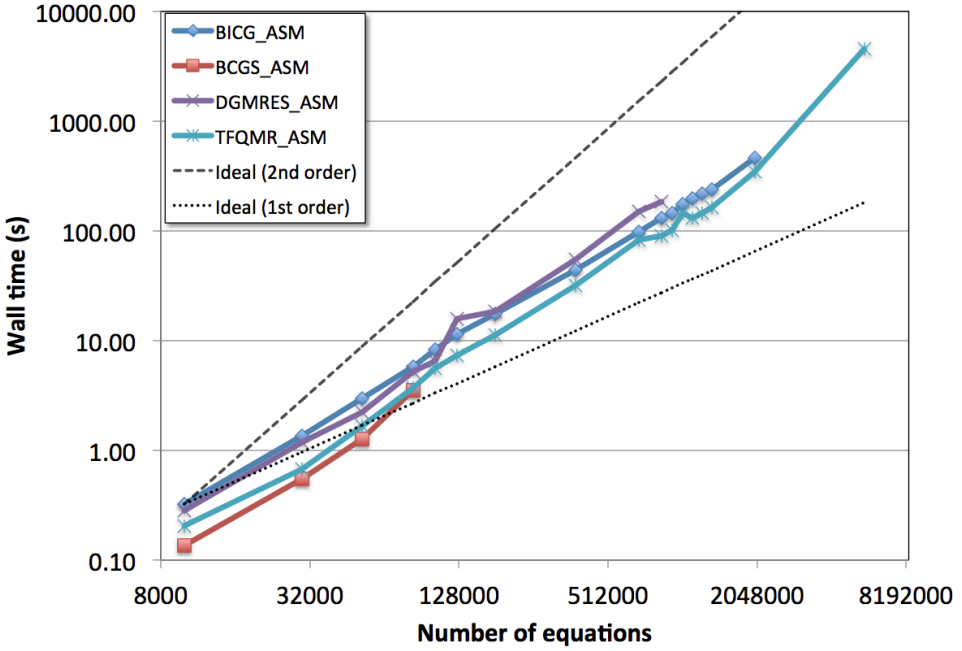
OOM: Out of memory during the factorization process



# Sparse matrix iterative solvers on an XE node - PETSc

| Matrix | Number of equations | bicg with asm | bcgs with asm | dgmres with asm | tfqmr with asm |
|---|---|---|---|---|---|
| LHM14 | 9,965 | 0.32 s | 0.14 s | 0.28 s | 0.21 s |
| LHM20 | 29,837 | 1.36 s | 0.55 s | 1.18 s | 0.67 s |
| LHM24 | 52,125 | 2.95 s | 1.26 s | 2.22 s | 1.65 s |
| LHM28 | 83,437 | 5.82 s | 3.51 s | 5.18 s | 3.73 s |
| LHM30 | 102,957 | 8.30 s | NC | 6.53 s | 5.57 s |
| LHM32 | 125,309 | 11.50 s | NC | 15.72 s | 7.29 s |
| LHM36 | 179,277 | 17.46 s | NC | 18.51 s | 11.24 s |
| LHM46 | 377,197 | 44.04 s | NC | 54.51 s | 31.92 s |
| LHM56 | 684,317 | 98.67 s | NC | 149.09 s | 82.30 s |
| LHM60 | 843,117 | 130.06 s | NC | 182.68 s | 89.53 s |
| LHM62 | 930,989 | 144.49 s | NC | NC | 100.61 s |
| LHM64 | 1,024,756 | 174.46 s | NC | NC | 145.06 s |
| LHM66 | 1,124,637 | 197.02 s | NC | NC | 130.24 s |
| LHM68 | 1,230,797 | 217.62 s | NC | NC | 145.19 s |
| LHM70 | 1,343,437 | 236.11 s | NC | NC | 164.11 s |
| LHM80 | 2,010,557 | 463.37 s | NC | NC | 346.59 s |
| LHM112 | 5,545,789 | NC | NC | NC | 4538.61 s |

NC: not converged (it means total number of iteration is equal to maxits).



# Sparse matrix iterative solvers on an XK node – NVIDIA AmgX

- Mode parameter for AmgX: dDDI
- CUTOTOOLKIT version 7.0.28-1.0502.10742.5.1
- Maximum number of iterations = 10,000
- Relative residual tolerance = $10^{-12}$

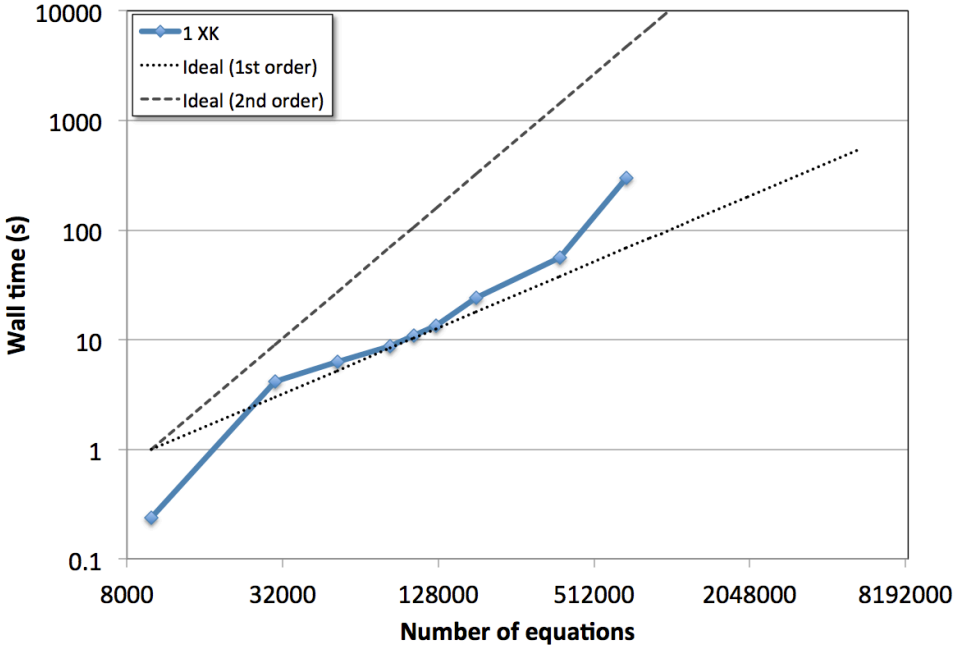
## Configuration for AmgX solver

```
{  "config_version": 2,
   "solver": {"print_grid_stats": 1,
              "store_res_history": 1,
              "solver": "FGMRES",
              "print_solve_stats": 1,
              "obtain_timings": 1,
              "preconditioner": {
                  "interpolator": "D2",
                  "print_grid_stats": 1,
                  "aggressive_levels": 1,
                  "solver": "AMG",
                  "smoother": {
                      "relaxation_factor": 1,
                      "scope": "jacobi",
                      "solver": "JACOBI_L1"},
                  "presweeps": 2,
                  "selector": "PMIS",
                  "coarsest_sweeps": 1,
                  "coarse_solver": "DENSE_LU_SOLVER",
                  "max_iters": 1,
                  "max_row_sum": 0.9,
                  "strength_threshold": 0.25,
                  "min_coarse_rows": 2,
                  "scope": "amg_solver",
                  "max_levels": 24,
                  "cycle": "V",
                  "postsweeps": 2},
              "max_iters": 10000,
              "monitor_residual": 1,
              "gmres_n_restart": 500,
              "convergence": "RELATIVE_INI_CORE",
              "tolerance": 1e-12,
              "norm": "L2"}}
```

# Sparse matrix iterative solvers on an XK node – NVIDIA AmgX

| Matrix | Number of equations | Wall time | Number of iterations | Total reduction in residual |
|---|---|---|---|---|
| LHM14 | 9,965 | 0.24 s | 1 | 6.42E-15 |
| LHM20 | 29,837 | 4.18 s | 357 | 9.96E-13 |
| LHM24 | 52,125 | 6.25 s | 420 | 9.29E-13 |
| LHM28 | 83,437 | 8.79 s | 483 | 9.95E-13 |
| LHM30 | 102,957 | 10.98 s | 529 | 9.81E-13 |
| LHM32 | 125,309 | 13.44 s | 581 | 9.85E-13 |
| LHM36 | 179,277 | 24.28 s | 867 | 9.52E-13 |
| LHM46 | 377,197 | 55.79 s | 1175 | 9.94E-13 |
| LHM56 | 684,317 | 298.25 s | 2922 | 9.70E-13 |
| LHM60 | 843,117 | NC | 10000 | 5.07E-11 |
| LHM62 | 930,989 | NC | 10000 | 7.54E-07 |
| LHM64 | 1,024,756 | NC | 10000 | 9.78E-08 |
| LHM66 | 1,124,637 | NC | 10000 | 6.22E-08 |
| LHM68 | 1,230,797 | NC | 10000 | 1.67E-07 |
| LHM70 | 1,343,437 | NC | 10000 | 1.32E-07 |
| LHM80 | 2,010,557 | NC | 10000 | 5.73E-08 |
| LHM112 | 5,545,789 | OOM | | |

NC: not converged (it means total number of iteration is equal to maxits).
OOM: out of memory



# MULTIPLE-NODE TEST RESULT FOR THE BEST INTERCONNECT PERFORMANCE

# Sparse direct solvers on XE nodes – MUMPS, WSMP and SuperLU

## LMH56 – 684K equations

| Nodes | Cores | MUMPS | | | | | WSMP | | | | | SuperLU_DIST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PreFac | Fac | Solve | Total | Tflops | PreFac | Fac | Solve | Total | Tflops | PreFac | Fac | Solve | Total | Tflops |
| 1 | 16 | 32 | 284 | 9 | 325 | 0.085 | | | | | | 44 | 393 | 1 | 438 | 0.06 |
| 2 | 32 | 33 | 182 | 7 | 222 | 0.134 | 74 | 201 | 0.6 | 276 | 0.123 | 42 | 193 | 0.7 | 236 | 0.127 |
| 4 | 64 | 34 | 100 | 5 | 139 | 0.244 | 73 | 95 | 0.6 | 169 | 0.258 | 40 | 96 | 0.5 | 137 | 0.25 |
| 8 | 128 | 38 | 75 | 5 | 118 | 0.325 | 71 | 70 | 0.4 | 141 | 0.348 | 39 | 49 | 0.4 | 88 | 0.49 |
| 16 | 256 | 42 | 56 | 5 | 103 | 0.435 | 70 | 39 | 0.4 | 109 | 0.622 | 39 | 28 | 0.4 | 67 | 0.88 |
| 32 | 512 | 42 | 51 | 5 | 98 | 0.478 | 71 | 20 | 0.3 | 91 | 1.23 | 39 | 18 | 0.4 | 57 | 1.37 |
| 64 | 1024 | | | | | | 70 | 11 | 0.3 | 81 | 2.12 | 39 | 14 | 0.4 | 53 | 1.79 |

Nodes := number of XE nodes
Cores := number of MPI-rank * number of threads. (4 threads/MPI-rank for WSMP, pure MPI for MUMPS and SuperLU_DIST)
PreFac := elapsed time for pre-factorization process including reordering and symbolic factorization
Fac := elapsed time for factorization process
Solve := elapsed time for forward and backward substitution process
Total := PreFac + Fac + Solve
Tflops := Tera Flops for the factorization process

# Sparse direct solvers on XE nodes – MUMPS, WSMP and SuperLU

## LMH80 – 2M equations

| Nodes | Cores | MUMPS | | | | | WSMP | | | | | SuperLU_DIST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PreFac | Fac | Solve | Total | Tflops | PreFac | Fac | Solve | Total | Tflops | PreFac | Fac | Solve | Total | Tflops |
| 2 | 32 | | | | | | | | | | | 140 | 1345 | 2 | 1487 | 0.87 |
| 4 | 64 | 108 | 780 | 20 | 908 | 0.27 | 183 | 1137 | 1.7 | 1322 | 0.19 | 140 | 664 | 2 | 806 | 0.33 |
| 8 | 128 | 122 | 537 | 20 | 679 | 0.4 | 183 | 615 | 1.5 | 800 | 0.35 | 139 | 342 | 2 | 483 | 0.61 |
| 16 | 256 | 138 | 374 | 18 | 530 | 0.57 | 184 | 323 | 1 | 508 | 0.66 | 140 | 182 | 2 | 324 | 1.1 |
| 32 | 512 | 181 | 386 | 18 | 585 | 0.55 | 195 | 163 | 0.7 | 359 | 1.31 | 138 | 111 | 2 | 251 | 1.9 |
| 64 | 1024 | 234 | 414 | 18 | 666 | 0.51 | 194 | 86 | 0.7 | 281 | 2.52 | 139 | 98 | 2 | 239 | 2.1 |
| 128 | 2048 | | | | | | 190 | 44 | 0.7 | 235 | 4.93 | | | | | |
| 512 | 8192 | | | | | | 194 | 21 | 0.6 | 216 | 10.1 | | | | | |

4 threads/MPI-rank for WSMP; pure MPI for MUMPS and SuperLU_DIST

# Sparse direct solvers on XE nodes – MUMPS, WSMP and SuperLU

## LMH112 – 5.5M equations

| Nodes | Cores | MUMPS | | | | | WSMP | | | | | SuperLU_DIST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PreFac | Fac | Solve | Total | Tflops | PreFac | Fac | Solve | Total | Tflops | PreFac | Fac | Solve | Total | Tflops |
| 32 | 256 | 371 | 2241 | 60 | 2672 | 0.75 | 676 | 2501 | 2.8 | 3180 | 0.7 | 80 | 1663 | 11 | 1754 | 1.93 |
| 64 | 512 | 464 | 1785 | 60 | 2309 | 0.93 | 683 | 1326 | 2.6 | 2012 | 1.3 | 178 | 788 | 9 | 975 | 3.46 |
| 128 | 1024 | 714 | 1611 | 61 | 2386 | 1.05 | 676 | 1021 | 2.5 | 1697 | 1.6 | 334 | 591 | 9 | 934 | 4.78 |
| 256 | 2048 | 1317 | 1608 | 68 | 2993 | 1.04 | 676 | 531 | 1.8 | 1209 | 3.2 | 566 | 1798 | 17 | 2381 | 1.43 |
| 512 | 4096 | | | | | | 685 | 267 | 1.7 | 951 | 6.4 | | | | | |
| 8 threads/MPI-rank for WSMP; pure MPI for MUMPS and SuperLU_DIST | | | | | | | | | | | | | | | | |

# Sparse direct solvers on XE nodes – MUMPS, WSMP and SuperLU

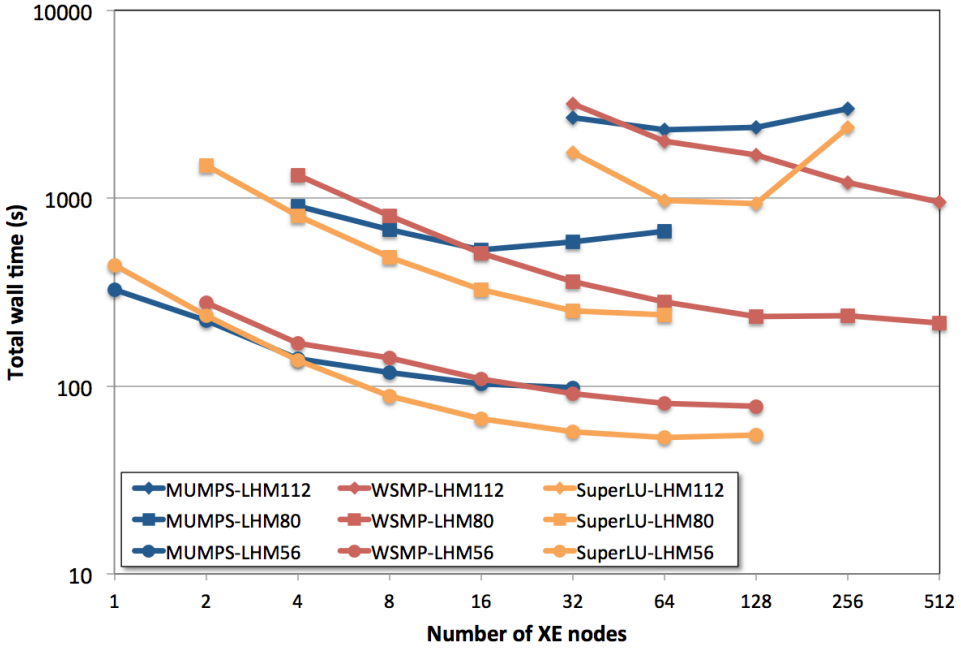
**Pre-factorization:**
reordering and symbolic factorization

↓

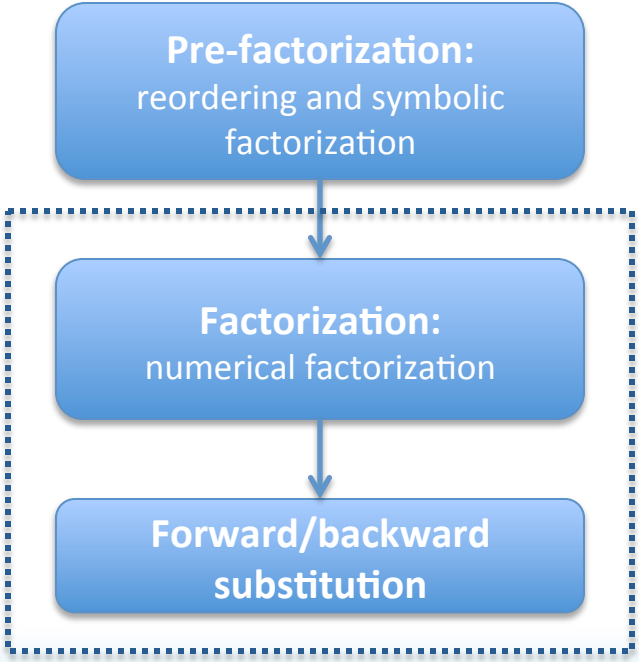
**Factorization:**
numerical factorization

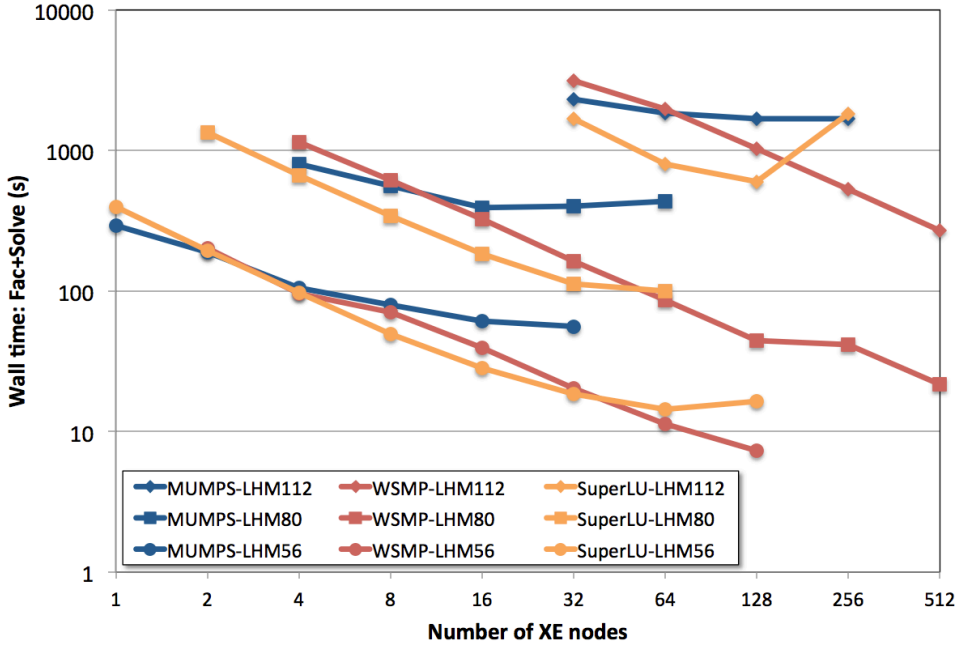
↓

**Forward/backward substitution**

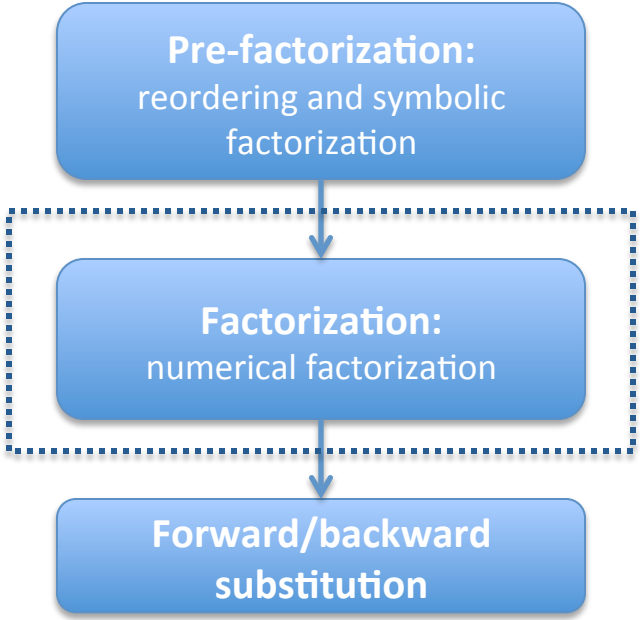


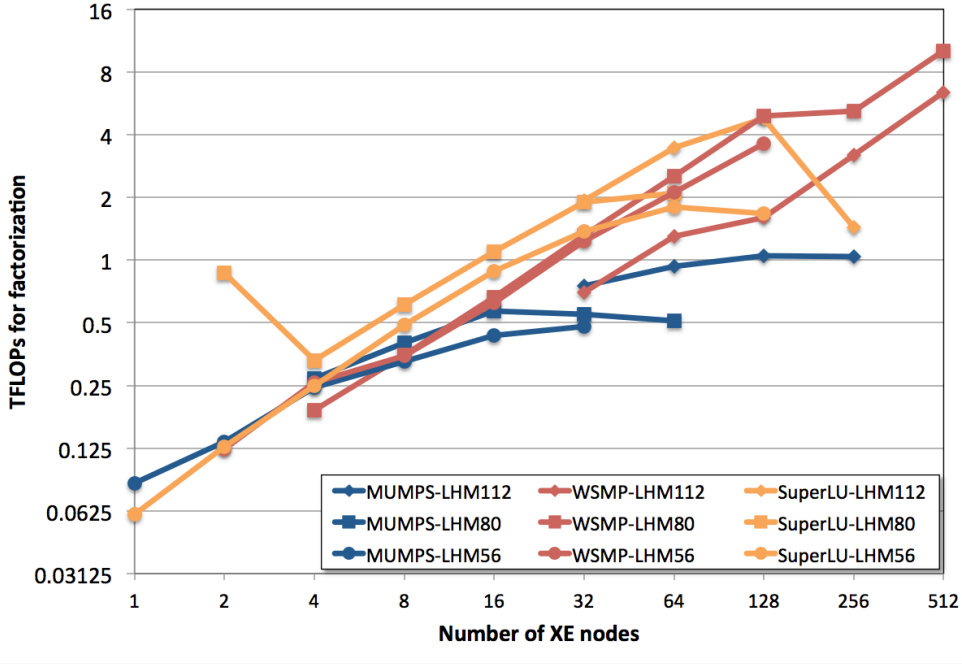
# Sparse direct solvers on XE nodes – MUMPS, WSMP and SuperLU



# Sparse direct solvers on XE nodes – MUMPS, WSMP and SuperLU

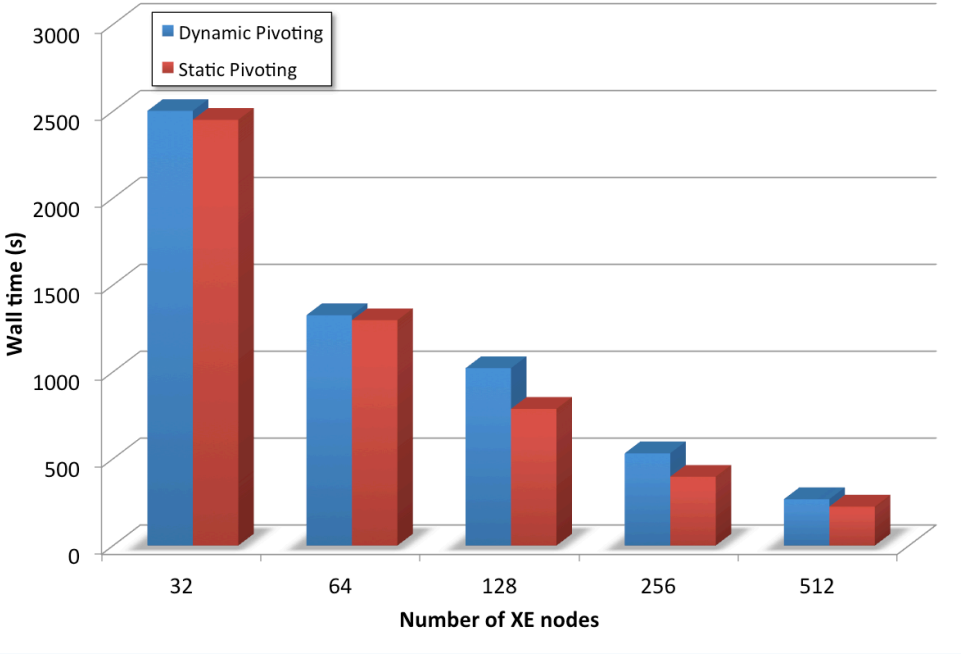


# Static pivoting vs. dynamic pivoting of WSMP

- Dynamic pivoting is more stable.
- Static pivoting is faster in factorization

| XE nodes | Dynamic pivoting | Static pivoting | Dynamic/ Static |
|---|---|---|---|
| 32 | 2501 s | 2450 s | 102 % |
| 64 | 1326 s | 1297 s | 102 % |
| 128 | 1021 s | 787 s | 130 % |
| 256 | 531 s | 396 s | 134 % |
| 512 | 267 s | 224 s | 119 % |

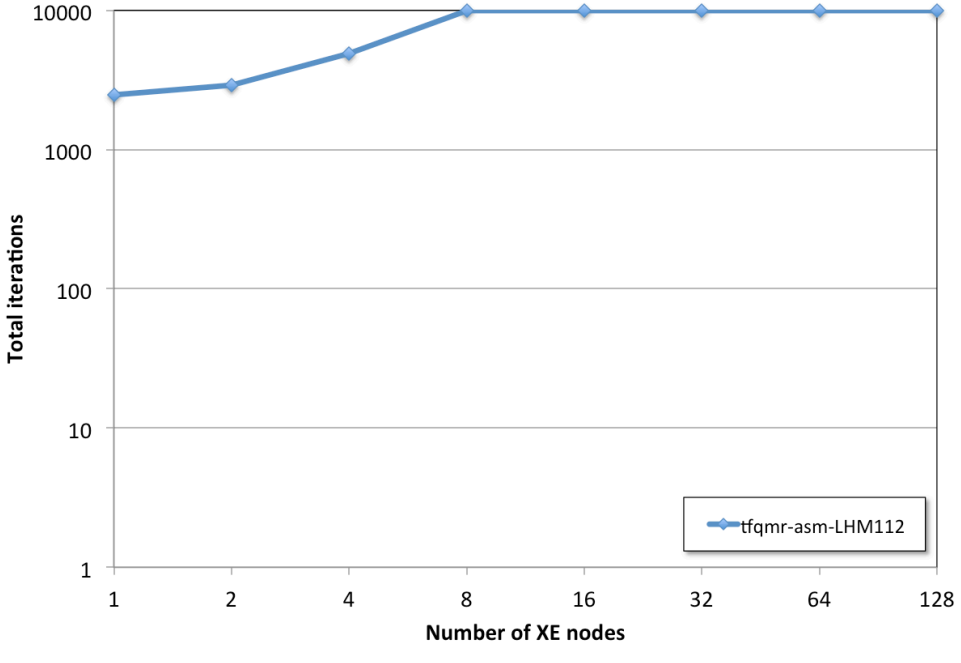


# Sparse iterative solvers on XE nodes – PETSc

TFQMR (KSP) with ASM (PC) on multiple XEs

| Number of XE nodes | Cores* | Wall time | Number of iterations |
|---|---|---|---|
| 1 | 16 | 4538.61 s | 2469 |
| 2 | 32 | 2100.78 s | 2904 |
| 4 | 64 | 1653.03 s | 4859 |
| 8 | 128 | NC | 10000 |
| 16 | 256 | NC | 10000 |
| 32 | 512 | NC | 10000 |
| 64 | 1024 | NC | 10000 |
| 128 | 2048 | NC | 10000 |

* Used 16 Bulldozer cores/node
NC: not converged



# Sparse iterative solvers on XE nodes – PETSc

Numerical trials to find out the optimal combination of KSP and PC on 8 XE nodes (LHM112)

| pc_types | ksp types | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | bicg | gmres | fgmres | dgmres | gcr | bcgs | cgs | tfqmr | tcqmr | cr | lsqr |
| -pc_type jacobi | D | NC | NC | D | NC | D | D | D | NC | NC | NC |
| -pc_type bjacobi | D | NC | NC | NC | NC | D | D | NC | NC | NC | NC |
| -pc_type asm | D | NC | NC | NC | NC | D | D | NC | NC | NC | NC |
| -pc_type gamg -pc_gamg_type agg * | D | NC | NC | NC | NC | D | D | NC | D | D | D |
| -pc_type gamg -pc_gamg_type classical | D | NC | NC | NC | NC | D | D | NC | NC | D | NC |
| -pc_type hypre -pc_hypre_set_type ams | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM |
| -pc_type hypre -pc_hypre_set_type ads | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM |
| -pc_type asm -pc_asm_type none | D | NC | NC | NC | NC | D | D | NC | NC | NC | NC |
| -pc_type asm -pc_asm_type interpolate | D | NC | NC | NC | NC | D | D | NC | NC | NC | NC |
| -pc_type asm -pc_asm_type basic | D | NC | NC | NC | NC | D | D | C | NC | NC | NC |

* with -pc_gamg_agg_nsmooths 0 for non-symmetric matrices
D for 'diverged'/ NC for 'not converged'/ OOM for 'out of memory'/ C for 'converged'

# Sparse iterative solvers on XE nodes – PETSc

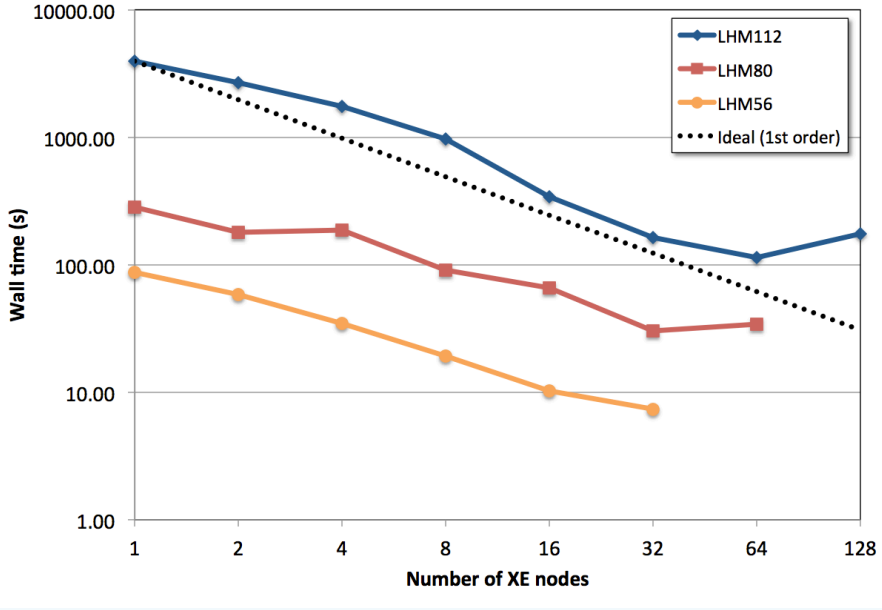
## TFQMR (KSP) with ASM_BASIC (PC) on multiple XEs

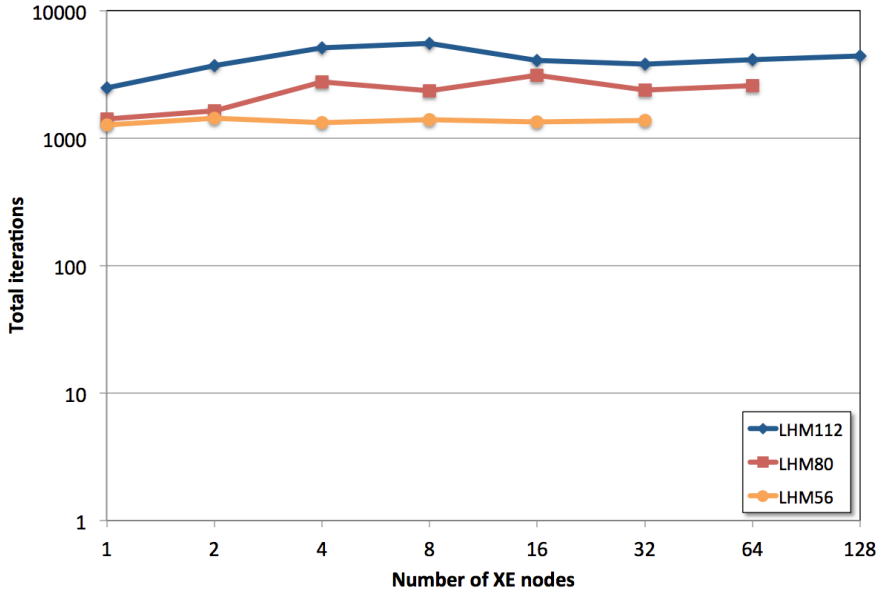
| Number of XE nodes | Cores* | LHM112 | | | LHM80 | | | LHM56 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Wall time | Number of iterations | Speedup | Wall time | Number of iterations | Speedup | Wall time | Number of iterations | Speedup |
| 1 | 16 | 3931.61 s | 2467 | 1.0 | 282.83 s | 1406 | 1.0 | 87.48 s | 1258 | 1.0 |
| 2 | 32 | 2690.40 s | 3710 | 1.5 | 179.68 s | 1625 | 1.6 | 58.25 s | 1431 | 1.5 |
| 4 | 64 | 1752.43 s | 5107 | 2.2 | 185.97 s | 2753 | 1.5 | 34.60 s | 1326 | 2.5 |
| 8 | 128 | 966.55 s | 5512 | 4.1 | 90.13 s | 2338 | 3.1 | 19.13 s | 1397 | 4.6 |
| 16 | 256 | 339.41 s | 4053 | 11.6 | 66.19 s | 3120 | 4.3 | 10.23 s | 1334 | 8.6 |
| 32 | 512 | 163.34 s | 3772 | 24.1 | 30.08 s | 2366 | 9.4 | 7.37 s | 1380 | 11.9 |
| 64 | 1024 | 113.73 s | 4131 | 34.6 | 34.17 s | 2563 | 8.3 | | | |
| 128 | 2048 | 174.65 s | 4374 | 22.5 | | | | | | |

* Used 16 Bulldozer cores/node

# Sparse iterative solvers on XE nodes – PETSc

TFQMR (KSP) with ASM_BASIC (PC) on multiple XEs



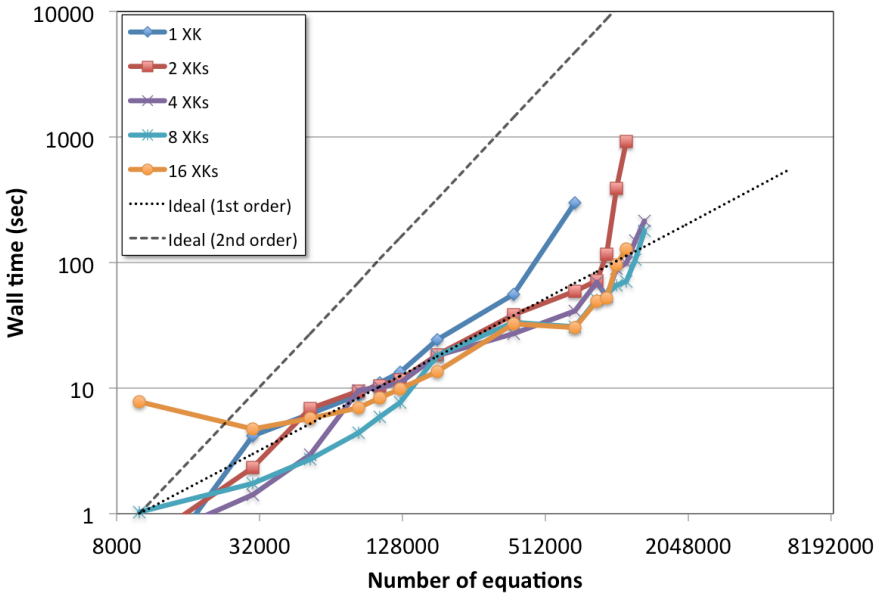



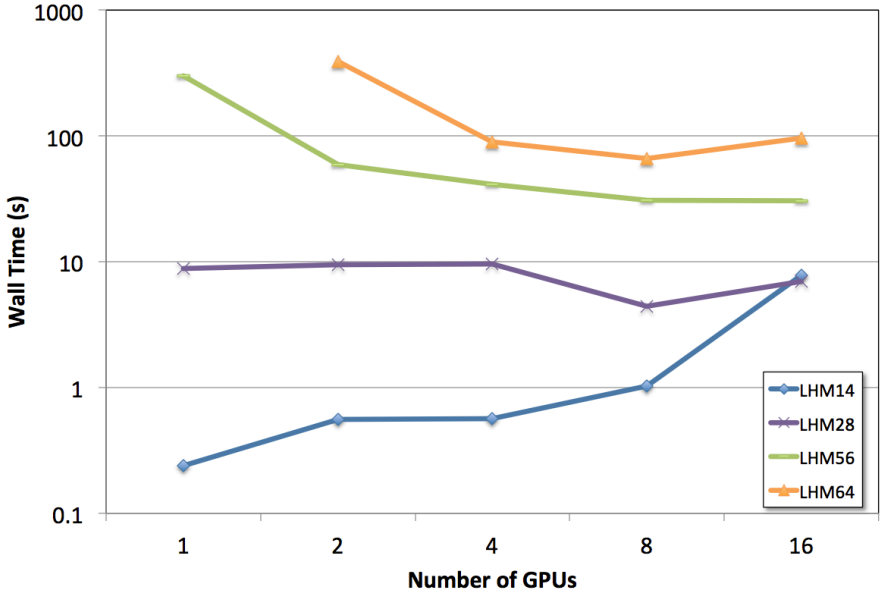
# Sparse iterative solvers on XK nodes – AmgX

| Matrix | Number of equations | 2 XK nodes | | | 4 XE nodes | | | 8 XK nodes | | | 16 XK nodes | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Wtime | Niter | Reduc | Wtime | Niter | Reduc | Wtime | Niter | Reduc | Wtime | Niter | Reduc |
| LHM14 | 9,965 | 0.56 | 57 | 6.89E-13 | 0.57 | 79 | 7.81E-13 | 1.03 | 140 | 9.38E-13 | 7.83 | 447 | 8.66E-13 |
| LHM20 | 29,837 | 2.33 | 65 | 7.83E-13 | 1.41 | 87 | 9.52E-13 | 1.75 | 157 | 7.98E-13 | 4.73 | 318 | 9.58E-13 |
| LHM24 | 52,125 | 6.92 | 427 | 9.20E-13 | 2.94 | 93 | 7.70E-13 | 2.71 | 163 | 8.39E-13 | 5.71 | 332 | 8.47E-13 |
| LHM28 | 83,437 | 9.49 | 491 | 9.74E-13 | 9.56 | 506 | 9.99E-13 | 4.42 | 170 | 7.99E-13 | 7.02 | 343 | 8.51E-13 |
| LHM30 | 102,957 | 10.44 | 545 | 9.96E-13 | 10.09 | 565 | 9.81E-13 | 5.90 | 174 | 8.71E-13 | 8.39 | 357 | 9.36E-13 |
| LHM32 | 125,309 | 11.60 | 602 | 9.99E-13 | 10.97 | 624 | 9.79E-13 | 7.72 | 179 | 9.91E-13 | 9.85 | 365 | 9.03E-13 |
| LHM36 | 179,277 | 18.42 | 873 | 9.33E-13 | 18.14 | 914 | 9.93E-13 | 18.23 | 941 | 9.39E-13 | 13.55 | 369 | 8.83E-13 |
| LHM46 | 377,197 | 38.61 | 1172 | 9.65E-13 | 27.36 | 1172 | 9.91E-13 | 33.51 | 1493 | 9.93E-13 | 32.43 | 1491 | 9.97E-13 |
| LHM56 | 684,317 | 59.14 | 1239 | 9.99E-13 | 41.25 | 1248 | 1.00E-12 | 30.88 | 1268 | 9.99E-13 | 30.49 | 1351 | 1.00E-12 |
| LHM60 | 843,117 | 71.45 | 1283 | 9.80E-13 | 70.15 | 1845 | 9.68E-13 | 50.46 | 1841 | 9.74E-13 | 49.10 | 2028 | 9.97E-13 |
| LHM62 | 930,989 | 116.35 | 1859 | 9.94E-13 | 53.31 | 1337 | 9.88E-13 | 55.91 | 1874 | 9.79E-13 | 52.37 | 2168 | 9.94E-13 |
| LHM64 | 1,024,756 | 390.57 | 4467 | 9.75E-13 | 90.02 | 2052 | 9.89E-13 | 65.84 | 2033 | 9.96E-13 | 95.82 | 3852 | 9.99E-13 |
| LHM66 | 1,124,637 | 925.48 | 9973 | 9.97E-13 | 99.07 | 2170 | 9.96E-13 | 70.40 | 2129 | 9.82E-13 | 128.03 | 5000 | 9.92E-13 |
| LHM68 | 1,230,797 | NC | 10000 | 5.38E-11 | 150.12 | 3012 | 9.98E-13 | 106.31 | 3033 | 9.75E-13 | NC | 10000 | 1.82E-12 |
| LHM70 | 1,343,437 | NC | 10000 | 1.12E-07 | 215.35 | 4156 | 1.00E-12 | 179.33 | 4946 | 9.90E-13 | NC | 10000 | 7.68E-09 |
| LHM80 | 2,010,557 | NC | 10000 | 8.93E-08 | OOM | 410 | 3.60E-06 | NC | 10000 | 5.39E-10 | NC | 10000 | 6.35E-09 |
| LHM112 | 5,545,789 | OOM | | | OOM | | | OOM | 234 | 2.74E-03 | OOM | 460 | 1.48E-05 |

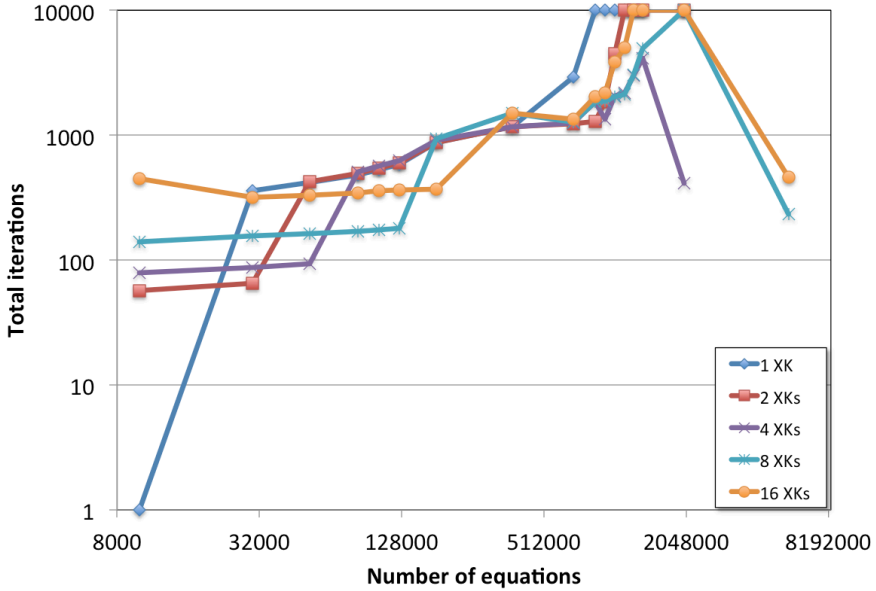
Wtime: elapsed wall time by AmgX; Niter: number of iterations; Reduc: total reduction in residual,
NC for 'not converged'/ OOM for 'out of memory'

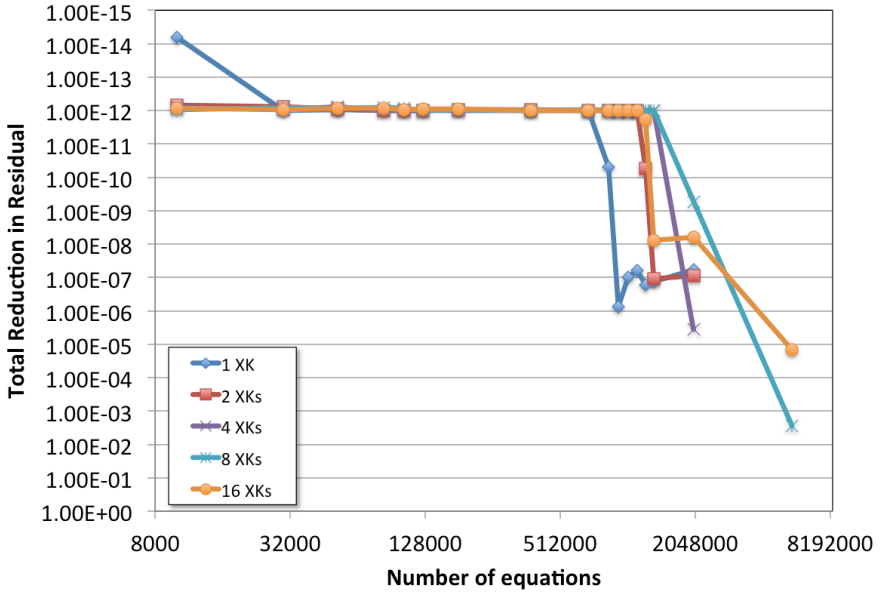
# Sparse iterative solvers on XK nodes – AmgX



# Sparse iterative solvers on XK nodes – AmgX



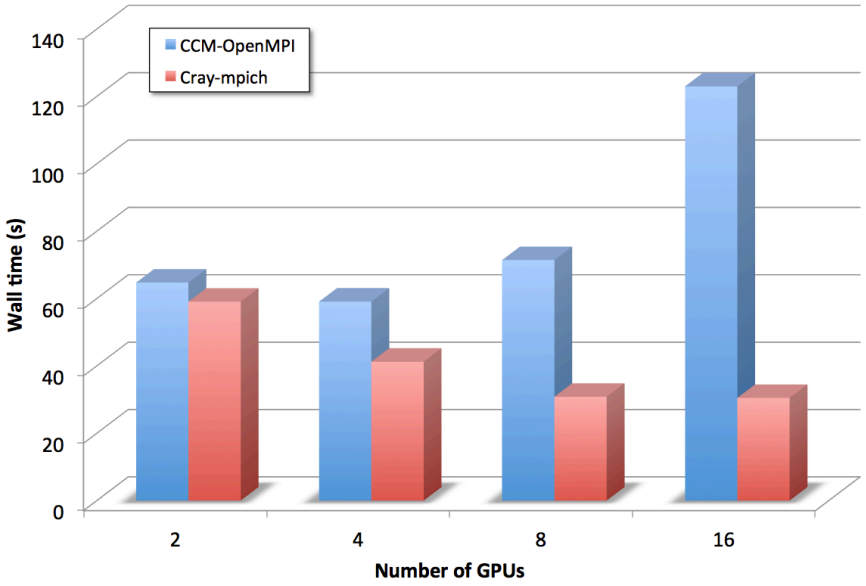
# AmgX with Cray-mpich and OpenMPI

| | | | 2 XK nodes | | 4 XK nodes | | 8 XK nodes | | 16 XK nodes | |
|---|---|---|---|---|---|---|---|---|---|---|
| Matrix | Number of equations | MPI types | Wtime | Niter | Wtime | Niter | Wtime | Niter | Wtime | Niter |
| LHM56 | 684,317 | CCM | 64.70 s | 1239 | 59.05 s | 1248 | 71.38 s | 1258 | 122.96 s | 1351 |
| | | Cray-mpich | 59.05 s | 1239 | 41.21 s | 1248 | 30.80 s | 1268 | 30.45 s | 1351 |
| Wall time ratio, CCM/Cray (%) | | | 110 % | | 143 % | | 232 % | | 404 % | |
| LHM64 | 1,024,765 | CCM | 145.09 s | 1959 | 152.29 s | 2052 | 273.69 s | 2033 | 946.73 s | 3852 |
| | | Cray-mpich | 390.91 s | 4467 | 89.96 s | 2052 | 65.80 s | 2033 | 95.65 s | 3852 |
| Wall time ratio, CCM/Cray (%) | | | 37 %* | | 169 % | | 416 % | | 990 % | |

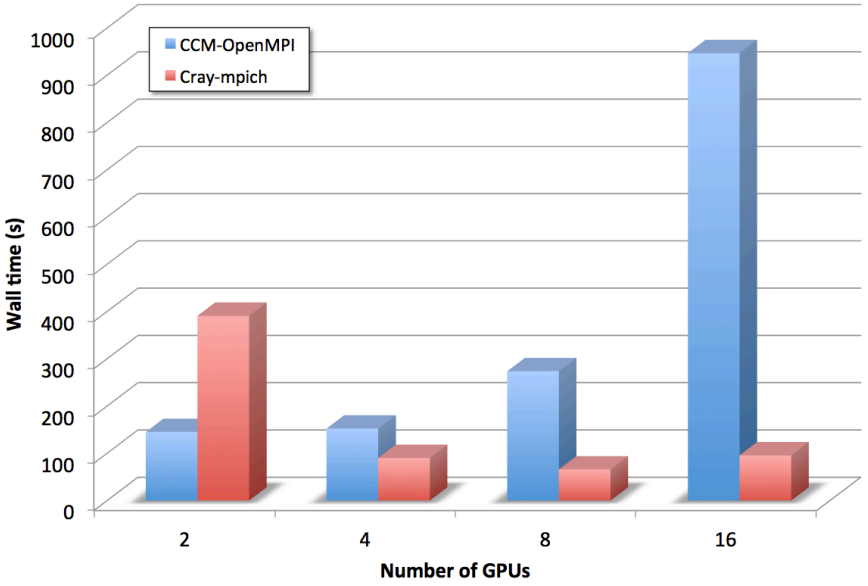
* Cray-compatible AmgX got the following warning message, and it restarted from the initial residual at the 500th iteration:
WARNING: Cannot allocate next Krylov vector, out of memory. Falling back to DQGMRES

# AmgX with Cray-mpich and OpenMPI

LHM56



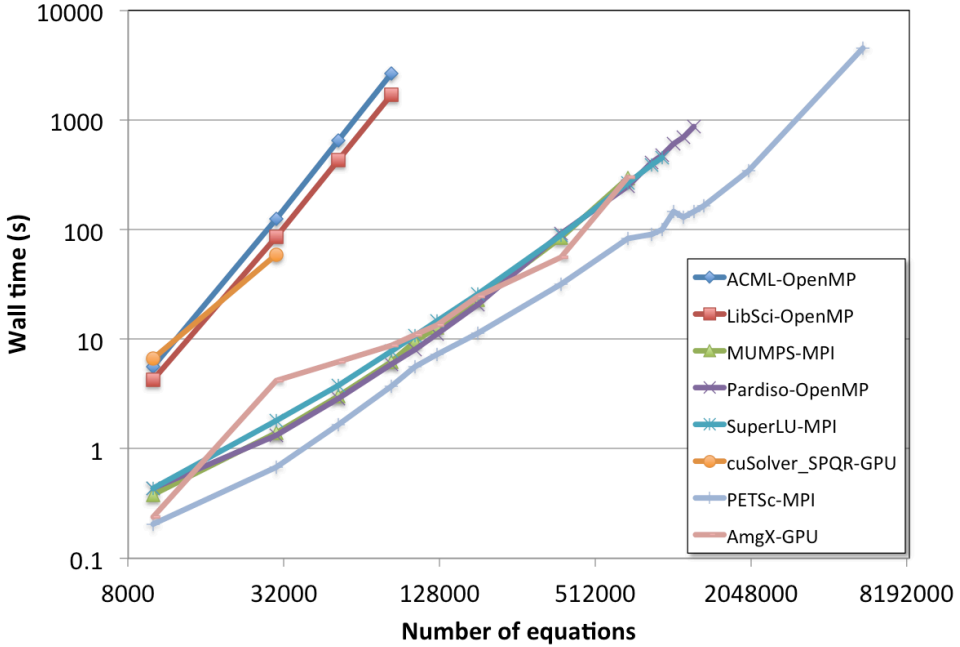

LHM64



# CONCLUDING REMARKS

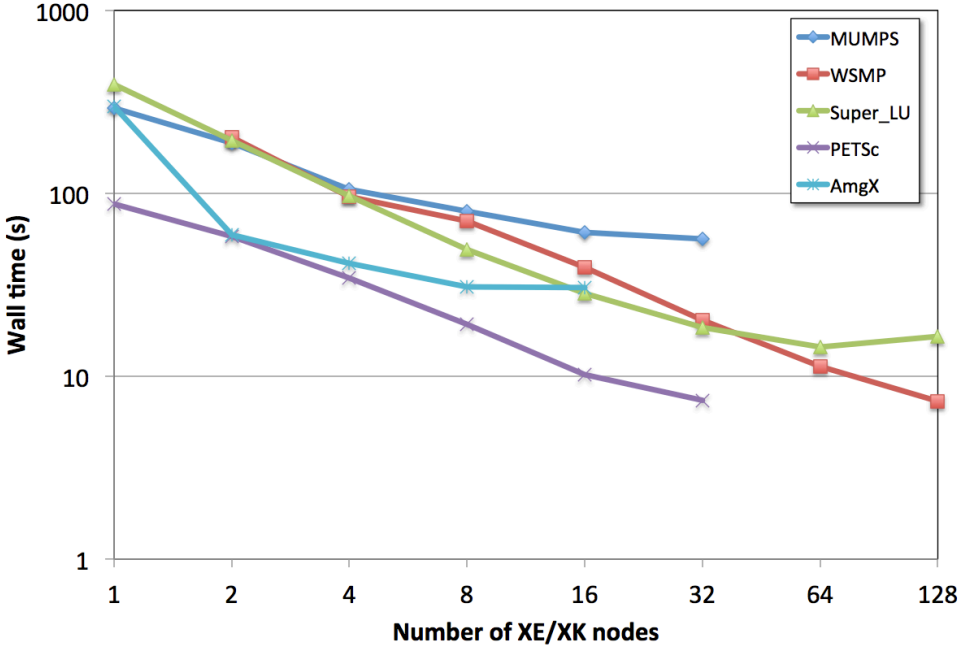
# LHM14 to LHM112 on a single XE/XK node

- Dense direct solvers
  - 3rd order time-complexity
  - Easy interface for users
  - The most expensive
- Sparse direct solvers
  - Better memory usage
  - Faster than dense direct solvers
  - Outstanding numerical stability
- Sparse iterative solvers
  - The most cost-effective
  - It may be difficult to find an optimal combination of iterative methods and preconditioners for ill-posed problems



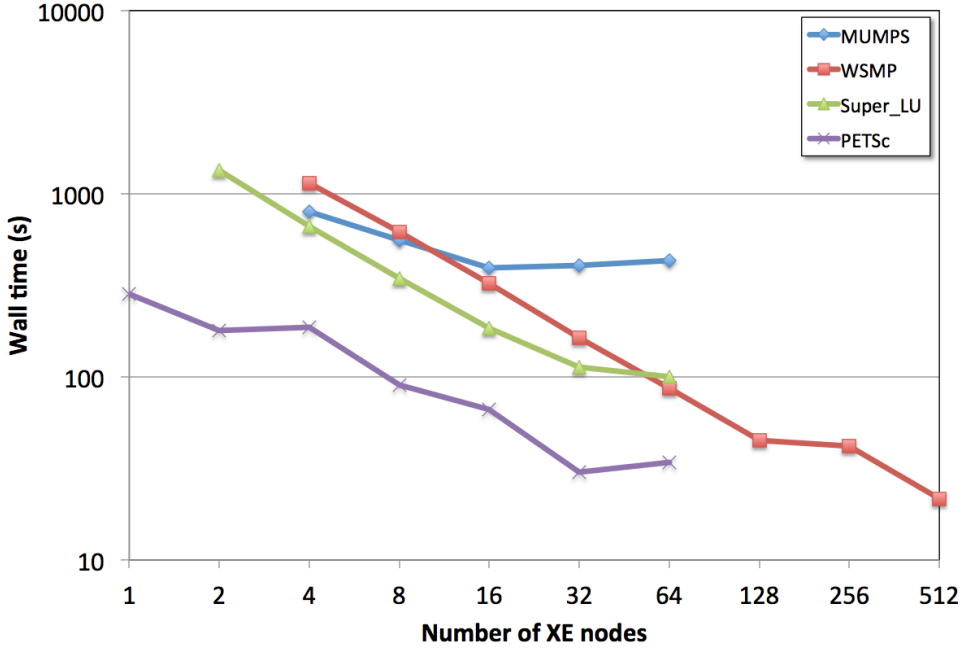
# LHM56 (684K equations) on multiple XE/XK nodes

- PETSc and AmgX solvers are more economical than direct solvers
- MUMPS, SuperLU and WSMP show excellent numerical stability for the ill-posed problem with condition number, 3.3 millions
- SuperLU and MUMPS are faster than WSMP with small numbers of nodes
- WSMP shows an excellent scalability with large numbers of nodes.



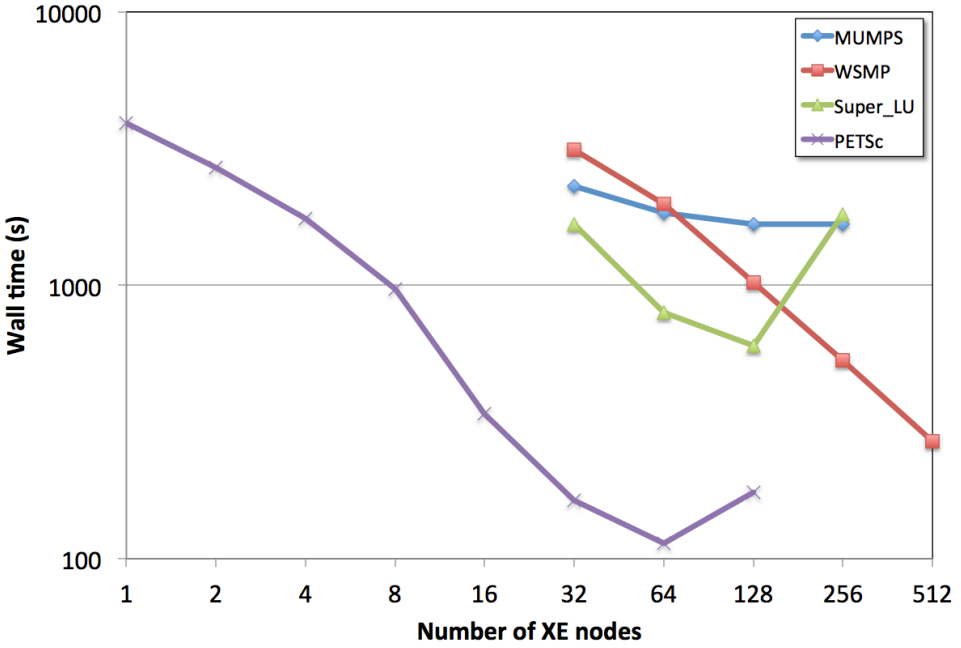
# LHM80 (2M equations) on multiple XE/XK nodes

- PETSc is the most cost-effective solver among employed solvers
- MUMPS, SuperLU and WSMP show excellent numerical stability for the ill-posed problem with condition number, 13.7 millions
- SuperLU and MUMPS are faster than WSMP with small numbers of nodes
- WSMP shows an excellent scalability with large numbers of nodes.



# LHM112 (5.5M equations) on multiple XE/XK nodes

- PETSc is the most cost-effective solver among employed solvers
- MUMPS, SuperLU and WSMP show excellent numerical stability for the ill-posed problem with condition number, 52.7 millions
- SuperLU and MUMPS are faster than WSMP with small numbers of nodes
- WSMP shows an excellent scalability with large numbers of nodes.



## Acknowledgment

This study is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.



# References


- B. Bode, M. Butler, T. Dunning, W. Gropp, T. Hoe-fler, W. Hwu, and W. Kramer (alphabetical). The Blue Waters Super-System for Super-Science. *Contemporary HPC Architectures*, Jeffery Vetter editor. Sitka Publications, November 2012.Edited by Jeffrey S . Vetter, Chapman and Hall/CRC 2013, Print ISBN: 978-1-4665-6834-1, eBook ISBN: 978-1-4665-6835-8.
- W. Kramer, M. Butler, G. Bauer, K. Chadalavada, C. Mendes. Blue Waters Parallel I/O Storage Sub-system, *High Performance Parallel I/O*, Prabhat and Quincey Koziol editors, CRC Publications, Taylor and Francis Group, Boca Raton FL, 2015, Hardback Print ISBN 13:978-1-4665-8234-7.
- S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page. *https://www.mcs.anl.gov/petsc*, 2016.
- P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2): 136-156, 2006.
- X. S. Li and J. W. Demmel. A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems. *ACM Trans. Mathematical Software*, 29(2):110-140, 2003.
- A. Gupta. WSMP: Watson Sparse matrix package (Part-II: direct solution of general systems). *Technical Report RC 21888* (*98472*). Yorkton Heights, NY: IBM T.J. Watson Research Center; 2016.
- NVIDIA AmgX webpage. *https://developer.nvidia.com/amgx*, 2016.
- J. Kwack and A. Masud. A stabilized mixed finite element method for shear-rate dependent non-Newtonian fluids: 3D benchmark problems and application to blood flow in bifurcating arteries. *Computational Mechanics*, 53:751-776, 2014.
- S. Koric, Q. Lu and E. Guleryuz. Evaluation of massively parallel linear sparse solvers on unstructured finite element meshes. *Computers and Structures*, 141:19-25, 2014.
- M. G. Venkata, R.L. Grahma, N.T. Hjelm, S.K. Gutierrez. Open MPI for Cray XE/XK systems. *CUG-2012*. https://www.open-mpi.org/papers/cug-2012/cug_2012_open_mpi_for_cray_xe_xk.pdf.