Parallel Preconditioning Methods for Ill-Conditioned Problems

Kengo Nakajima
Information Technology Center
The University of Tokyo

3rd International Workshops on Advances in Computational Mechanics (IWACOM-III)
OW-4: High Performance Computing for Computational Engineering and Science
October 13-14, 2015, Tokyo, Japan

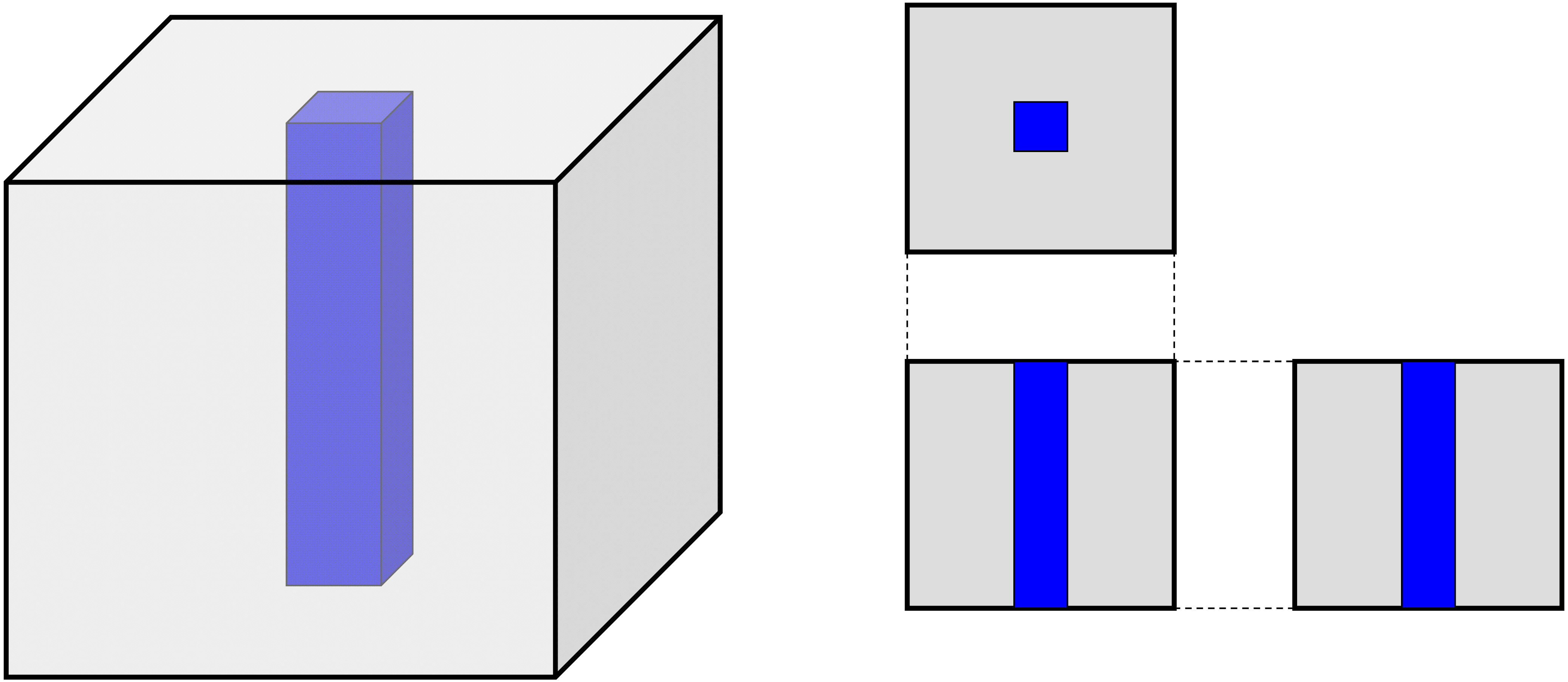
Preconditioning Methods (of Krylov Iterative Solvers) for Real-World Applications

- are the most critical issues in scientific computing
- are based on
  - Global Information: condition number, matrix properties etc.
  - Local Information: properties of elements (shape, size …)
- require knowledge of
  - background physics
  - applications

Technical Issues of “Parallel” Preconditioners in FEM

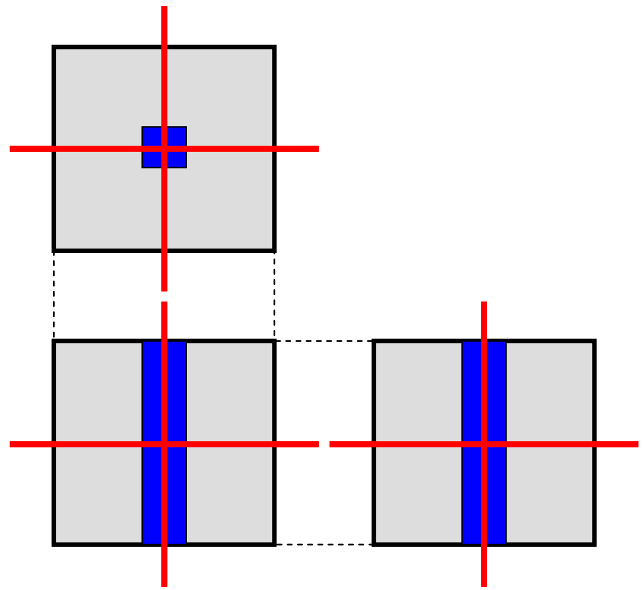
- Block Jacobi type Localized Preconditioners
- Simple problems can easily converge by simple preconditioners with excellent parallel efficiency.
- Difficult (ill-conditioned) prob’s cannot easily converge
  - Effect of domain decomposition on convergence is significant, especially for ill-conditioned problems.
    - Block Jacobi-type localized preconditioiners
    - More domains, more iterations
  - There are some remedies (e.g. deep fill-ins, deep overlapping)
  - ASDD does not work well for really ill-conditioned problems.

3D Linear Elastic Problem with 20^3 Tri-Linear Hexahedral Elements

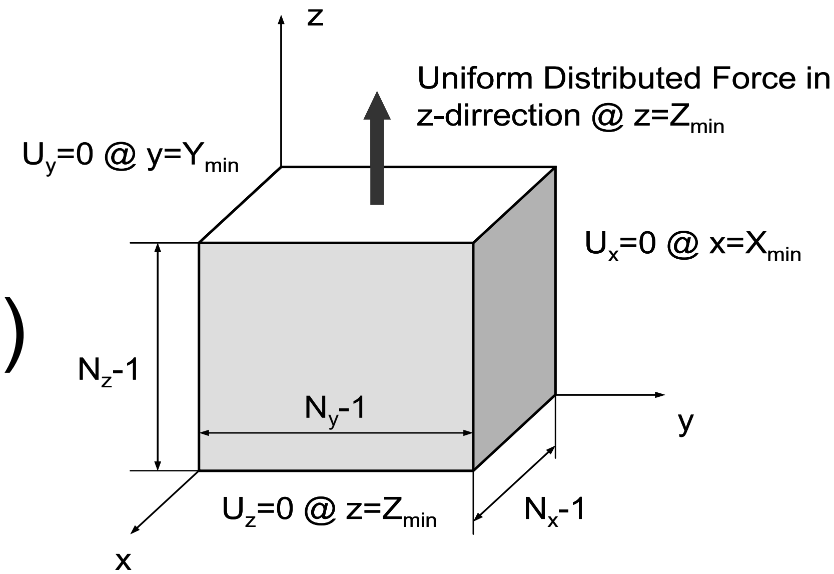


Iterations for Convergence BILU(0)-GPBiCG with 8 domains

- : ν= 0.25
- : E=1.00



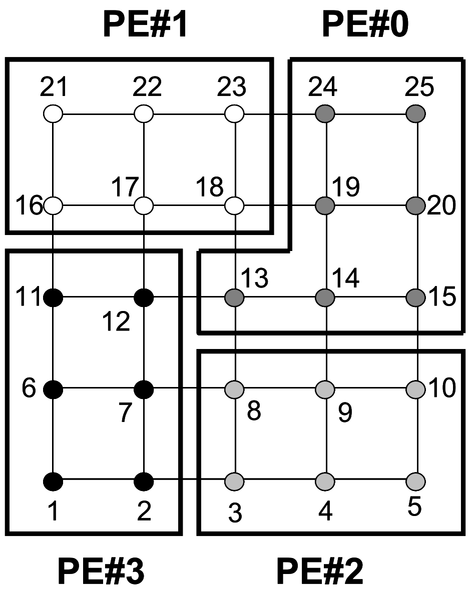
- 1-processor
  - : $E=10^0$, 31 iterations
  - : $E=10^{+3}$, 84 iterations
    - Harder, More ill-conditioned
- 8-processors (MPI, no-overlapping)
  - : $E=10^0$, 52 iter's（×1.68）
  - : $E=10^{+3}$, 158 iter's（×1.88）

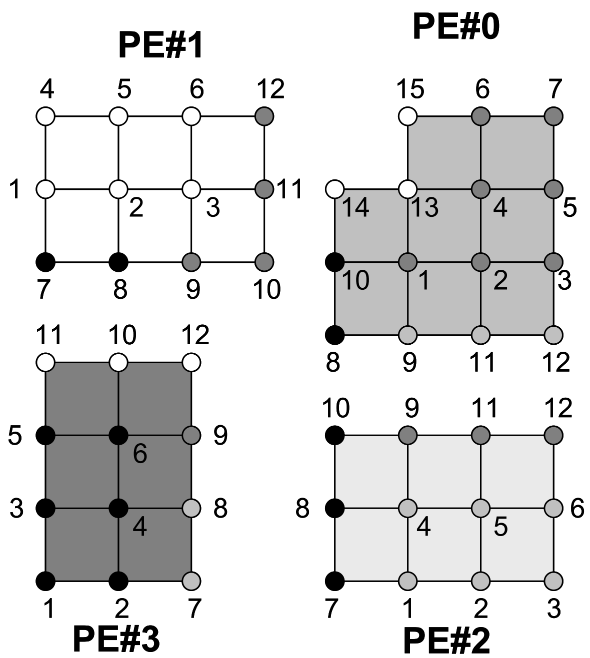


Remedies for Domain Decomposition

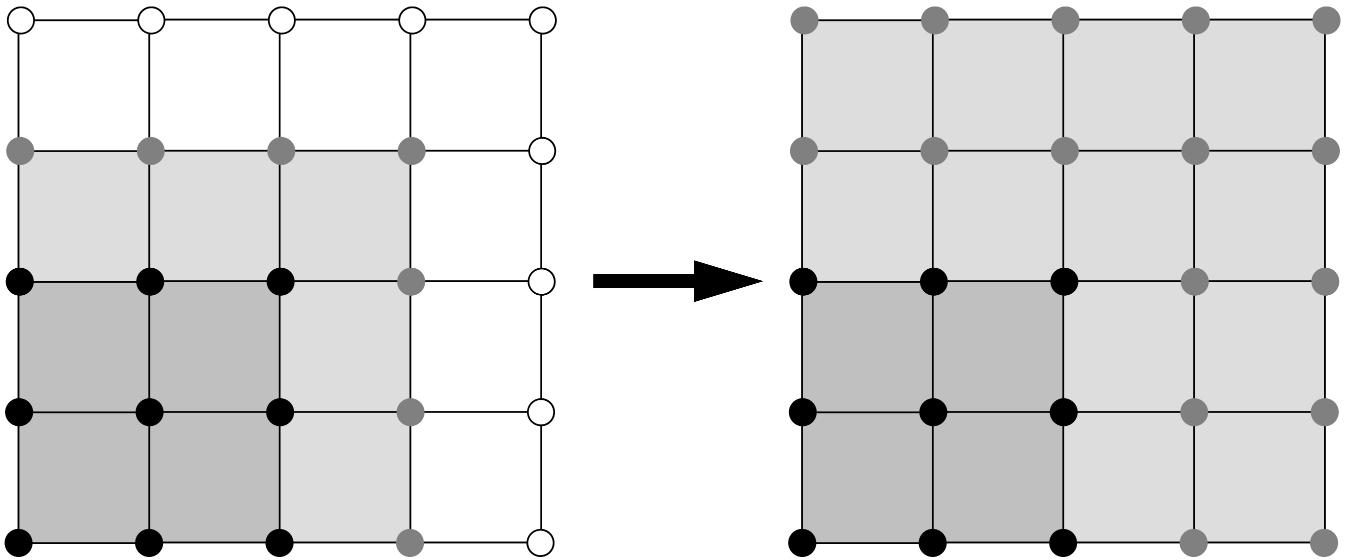
- Extended Depth of Overlapped Elements
  - Selective Fill-ins, Selective Overlapping [KN 2007]
    - adaptive preconditioning/domain decomposition methods which utilize features of FEM procedures
- PHIDAL/HID (Hierarchical Interface Decomposition) [Henon & Saad 2007]
- Extended HID [KN 2010]

Extension of Depth of Overlapping





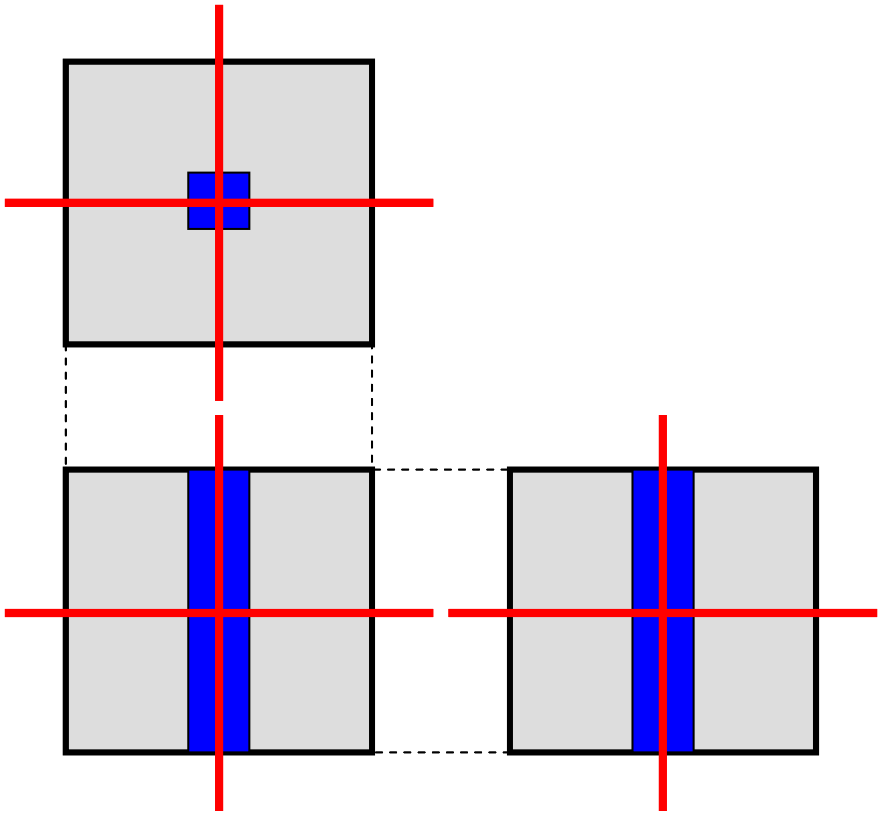
Cost for computation and communication may increase



● : Internal Nodes, ● : External Nodes
■ : Overlapped Elements

Number of Iterations for Convergence BILU(0)-GPBiCG, 8-domains (PE's)

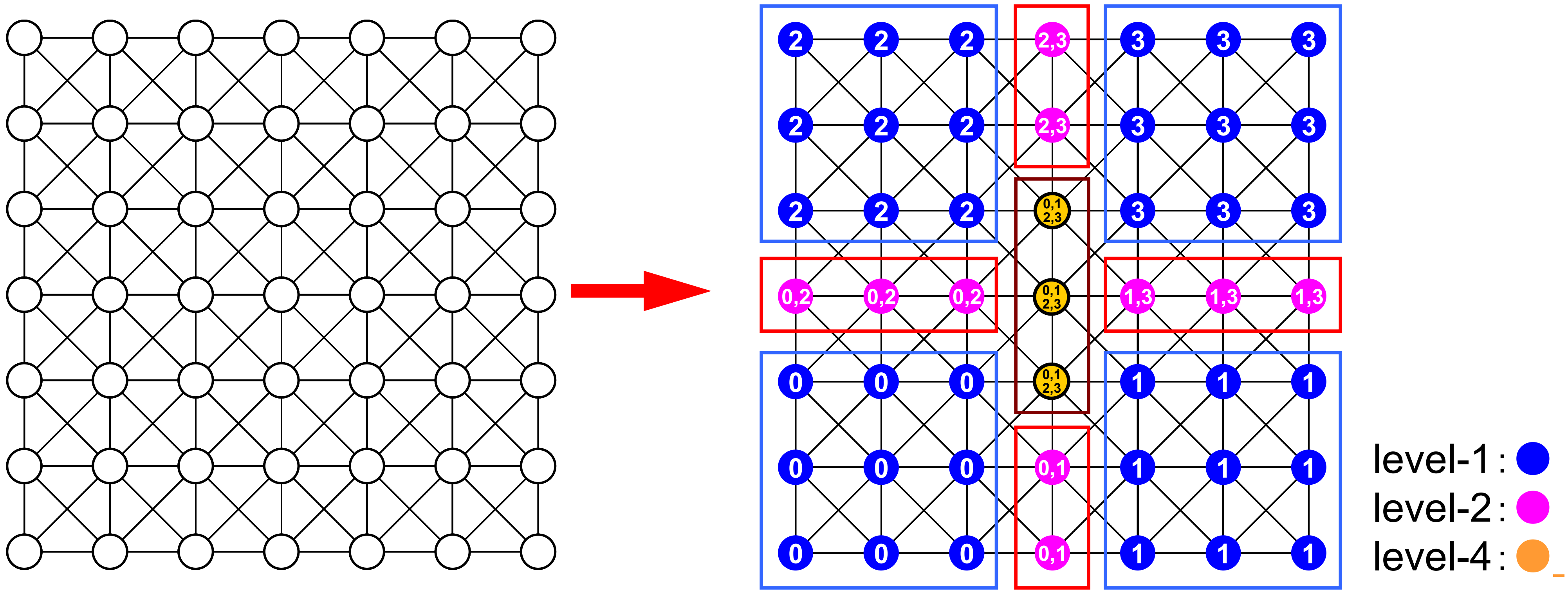
Effect of Extended Depth of Overlapping



Depth of Overlap	$E=10^0$	$E=10^3$
0	52	158
1	33	103
2	32	100
3	32	97
4	31	82
Single Domain	31	84

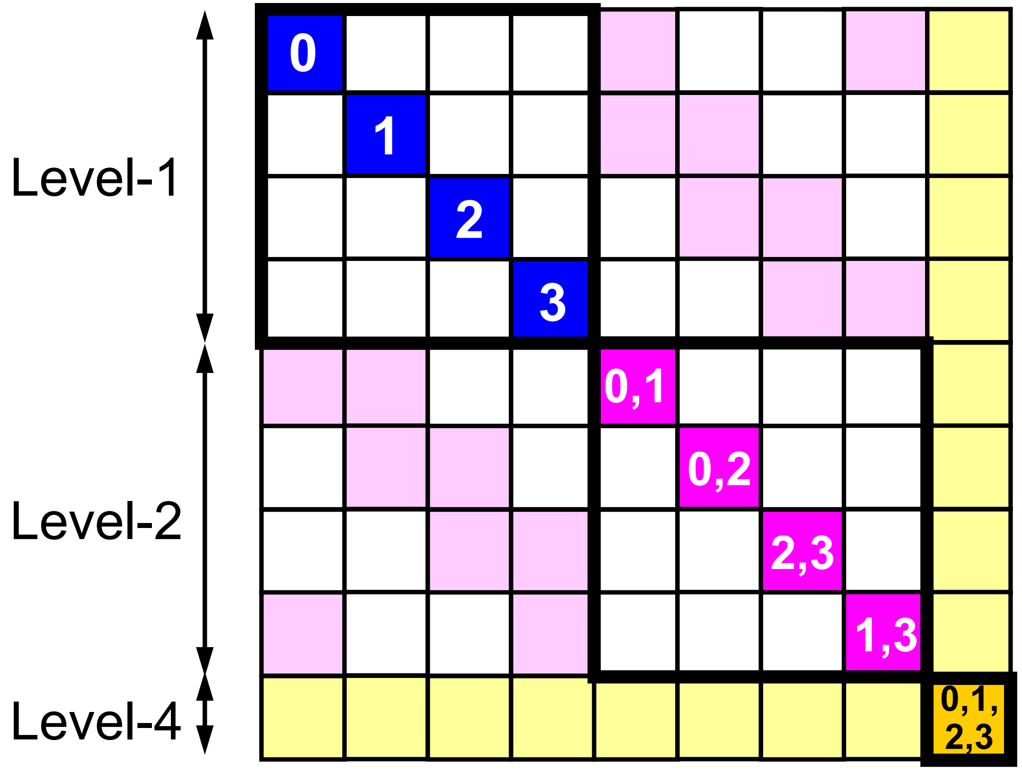
HID: Hierarchical Interface Decomposition [Henon & Saad 2007]

- Multilevel Domain Decomposition
  - Extension of Nested Dissection
- Non-overlapping at each level: Connectors, Separators
- Suitable for Parallel Preconditioning Method



Parallel ILU in HID for each Connector at each LEVEL

- The unknowns are reordered according to their **level** numbers, from the lowest to highest.
- The block structure of the reordered matrix leads to natural parallelism if ILU/IC decompositions or forward/backward substitution processes are applied.



Communications at Each Level Forward Substitutions

```
    do lev= 1, LEVELtot
      do i= LEVindex(lev-1)+1, LEVindex(lev)
        SW1= WW(3*i-2,R); SW2= WW(3*i-1,R); SW3= WW(3*i  ,R)
        isL= INL(i-1)+1; ieL= INL(i)
        do j= isL, ieL
          k= IAL(j)
          X1= WW(3*k-2,R); X2= WW(3*k-1,R); X3= WW(3*k  ,R)
          SW1= SW1 - AL(9*j-8)*X1 - AL(9*j-7)*X2 - AL(9*j-6)*X3
          SW2= SW2 - AL(9*j-5)*X1 - AL(9*j-4)*X2 - AL(9*j-3)*X3
          SW3= SW3 - AL(9*j-2)*X1 - AL(9*j-1)*X2 - AL(9*j  )*X3
        enddo
        X1= SW1; X2= SW2; X3= SW3
        X2= X2 - ALU(9*i-5)*X1
        X3= X3 - ALU(9*i-2)*X1 - ALU(9*i-1)*X2
        X3= ALU(9*i  )*  X3
        X2= ALU(9*i-4)*( X2 - ALU(9*i-3)*X3 )
        X1= ALU(9*i-8)*( X1 - ALU(9*i-6)*X3 - ALU(9*i-7)*X2)
        WW(3*i-2,R)= X1; WW(3*i-1,R)= X2; WW(3*i  ,R)= X3
      enddo

      call SOLVER_SEND_RECV_3_LEV(lev,…):

    enddo
```

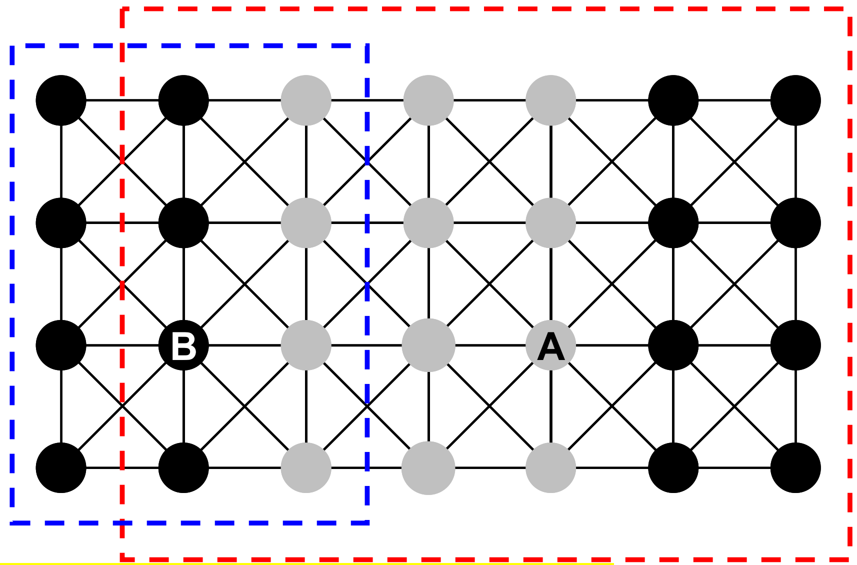
Communications using Hierarchical Comm. Tables.

Additional Comm.

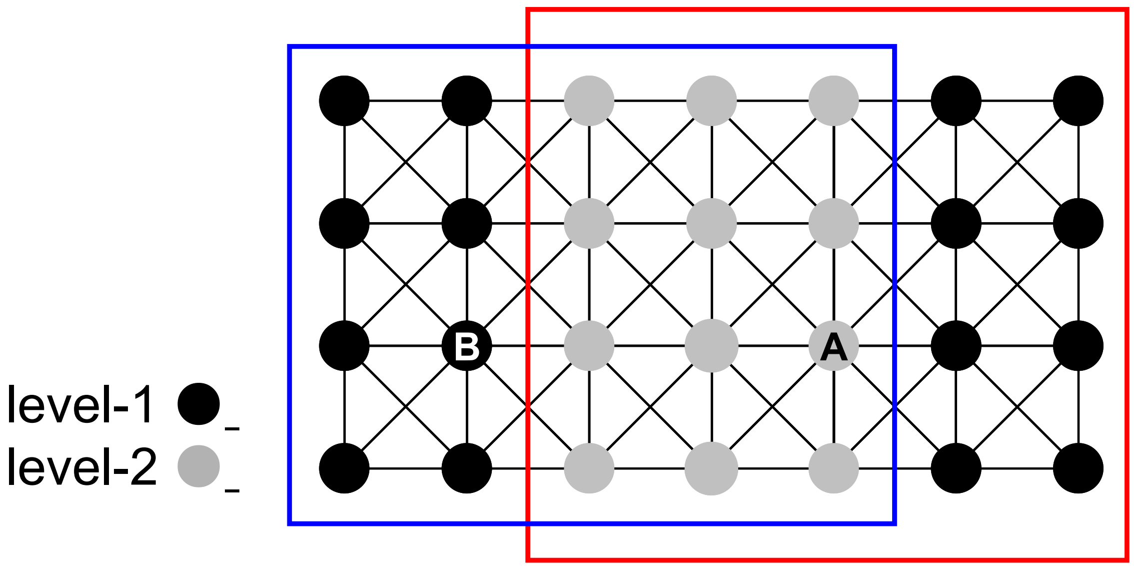
Extended HID [KN 2010] for Deeper Fill-in

- Thicker Separator
  - can consider the effects of fill-ins of higher order for external nodes at *same level*.
    - Effect of “A” can be considered for “B” in BILU(2)
  - **In global manner**
  - **seems to provide more robust convergence than Remedy 1.**
  - **difficulty for load-balancing**
- This option is not used in this study (no effects)

Distributed Local Data



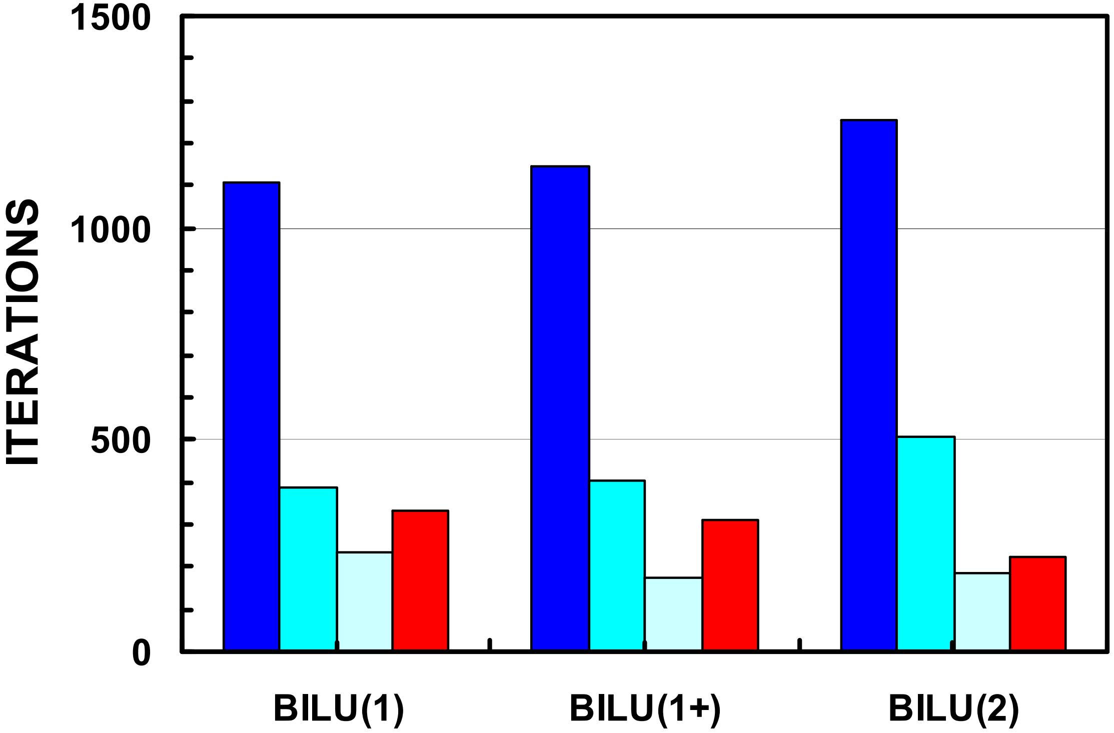
Range for “Global” Operations”

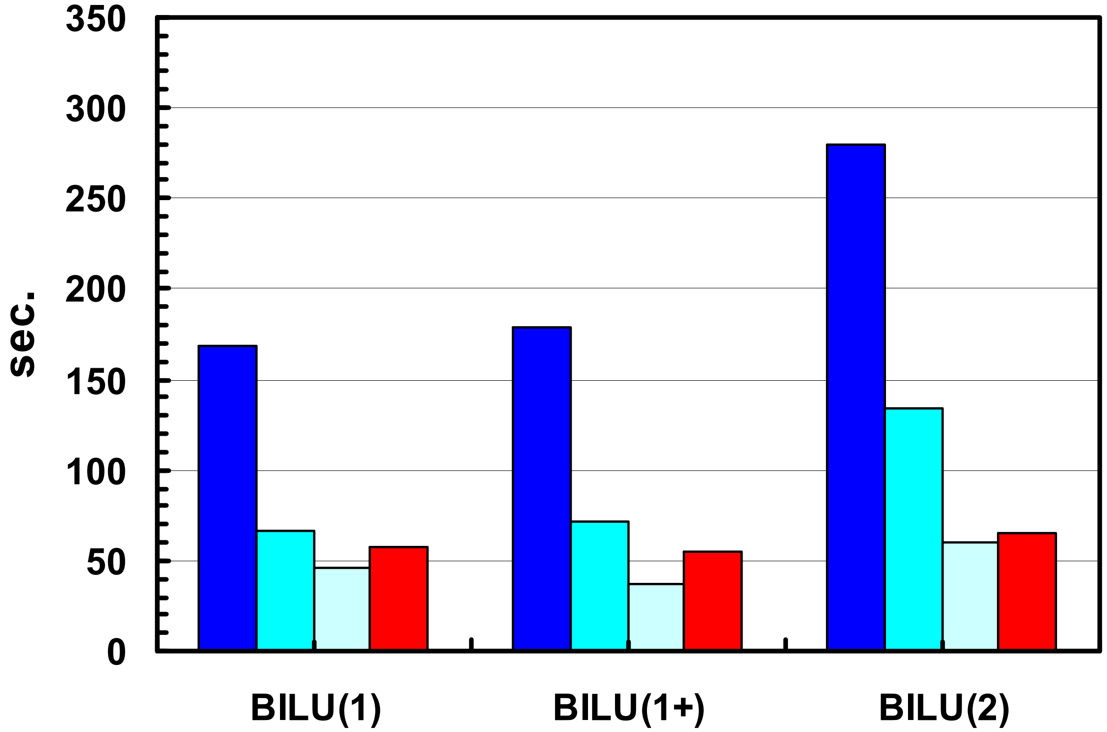


Results: 64 cores Contact Problems

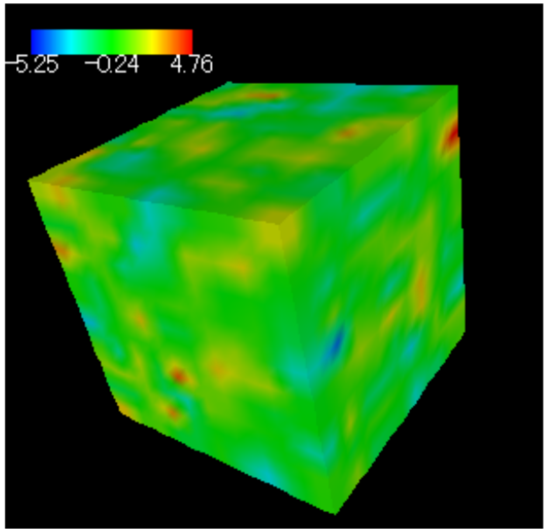
BILU(p)-(depth of overlapping) 3,090,903 DOF

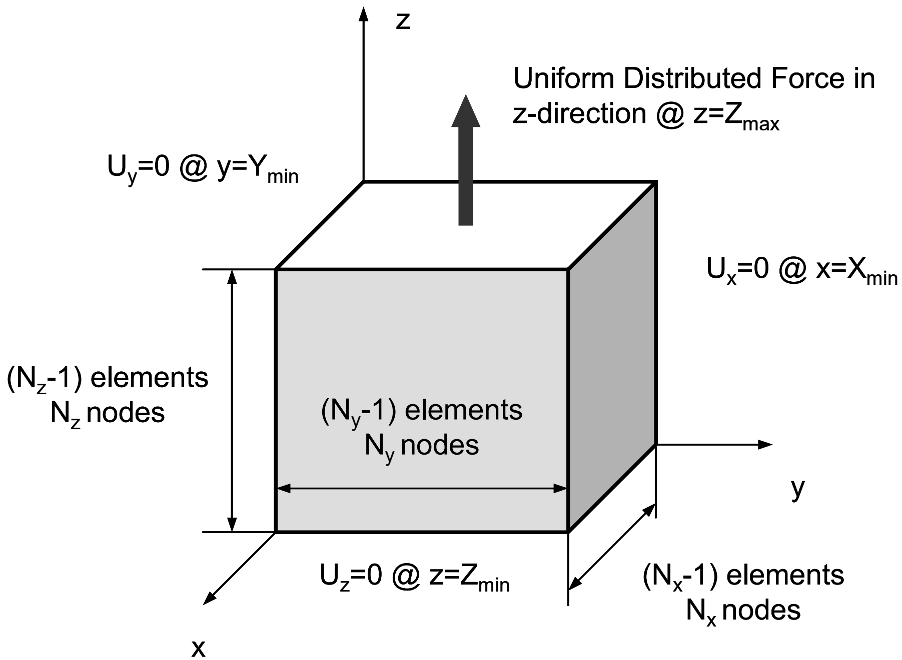
BILU(p)-(0): Block Jacobi
BILU(p)-(1)
BILU(p)-(1+)
BILU(p)-HID
GPBiCG





Hetero 3D (1/2)





- Parallel FEM Code (Flat MPI)
  - 3D linear elasticity problems in cube geometries with heterogeneity
  - SPD matrices
  - Young's modulus: 10^{-6}~10^{+6}
    - (E_{min}-E_{max}): controls condition number
- Preconditioned Iterative Solvers
  - GP-BiCG [Zhang 1997]
  - BILUT(*p*,*d*,*t*)
- Domain Decomposition
  - Localized Block-Jacobi with Extended Overlapping (LBJ)
  - HID/Extended HID

Hetero 3D (2/2)

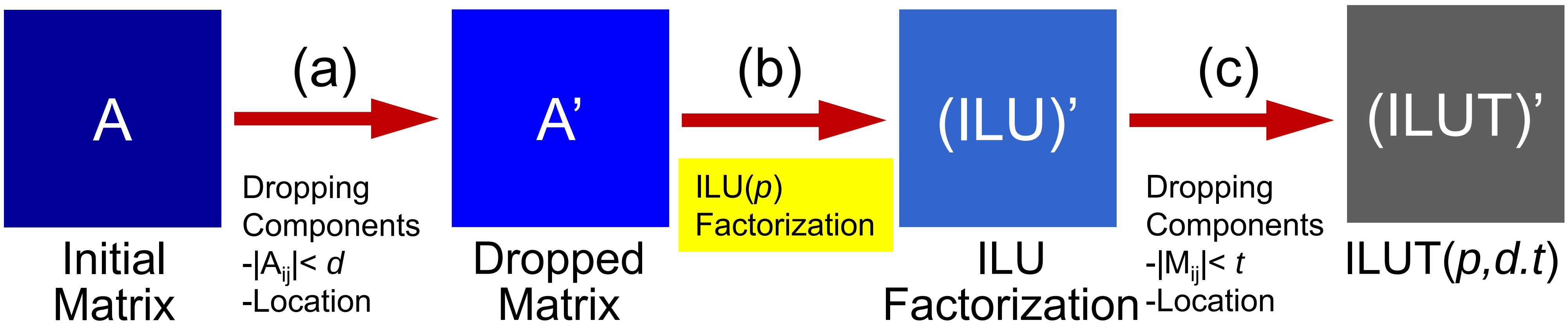
- based on the parallel FEM procedure of GeoFEM
  - Benchmark developed in **FP3C** project under Japan-France collaboration
    - Original Motivation: Reference implementation for evaluation of LRA by MUMPS
- Parallel Mesh Generation

FP3C

  - Fully parallel way
    - each process generates local mesh, and assembles local matrices.
  - Total number of vertices in each direction (N_x, N_y, N_z)
  - Number of partitions in each direction (P_x,P_y,P_z)
  - Number of total MPI processes is equal to $P_x \times P_y \times P_z$
  - Each MPI process has $(N_x/P_x) \times (N_y/P_y) \times (N_z/P_z)$ vertices.
  - Spatial distribution of Young's modulus is given by an external file, which includes information for heterogeneity for the field of 128^3 cube geometry.
    - If N_x (or N_y or N_z) is larger than 128, distribution of these 128^3 cubes is repeated periodically in each direction.

BILUT(*p*,*d*,*t*)

- Incomplete LU factorization with threshold (ILUT)
- ILUT(*p*,*d*,*t*) [KN 2010]
  - *p*: Maximum fill-level specified before factorization
  - *d*, *t*: Criteria for dropping tolerance before/after factorization
- The process (b) can be substituted by other factorization methods or more powerful direct linear solvers, such as *MUMPS*, *SuperLU* and etc.



Preliminary Results

- Hardware
  - 16-240 nodes (256-3,840 cores) of Fujitsu PRIMEHPC FX10 (Oakleaf-FX), University of Tokyo
- Problem Setting
  - $420 \times 320 \times 240$ vertices (3.194×10^7 elem's, 9.677×10^7 DOF)
  - Strong scaling
  - Effect of thickness of overlapped zones
    - BILUT(*p,d,t*)-LBJ-X (X=1,2,3)
  - RCM-Entire renumbering for LBJ

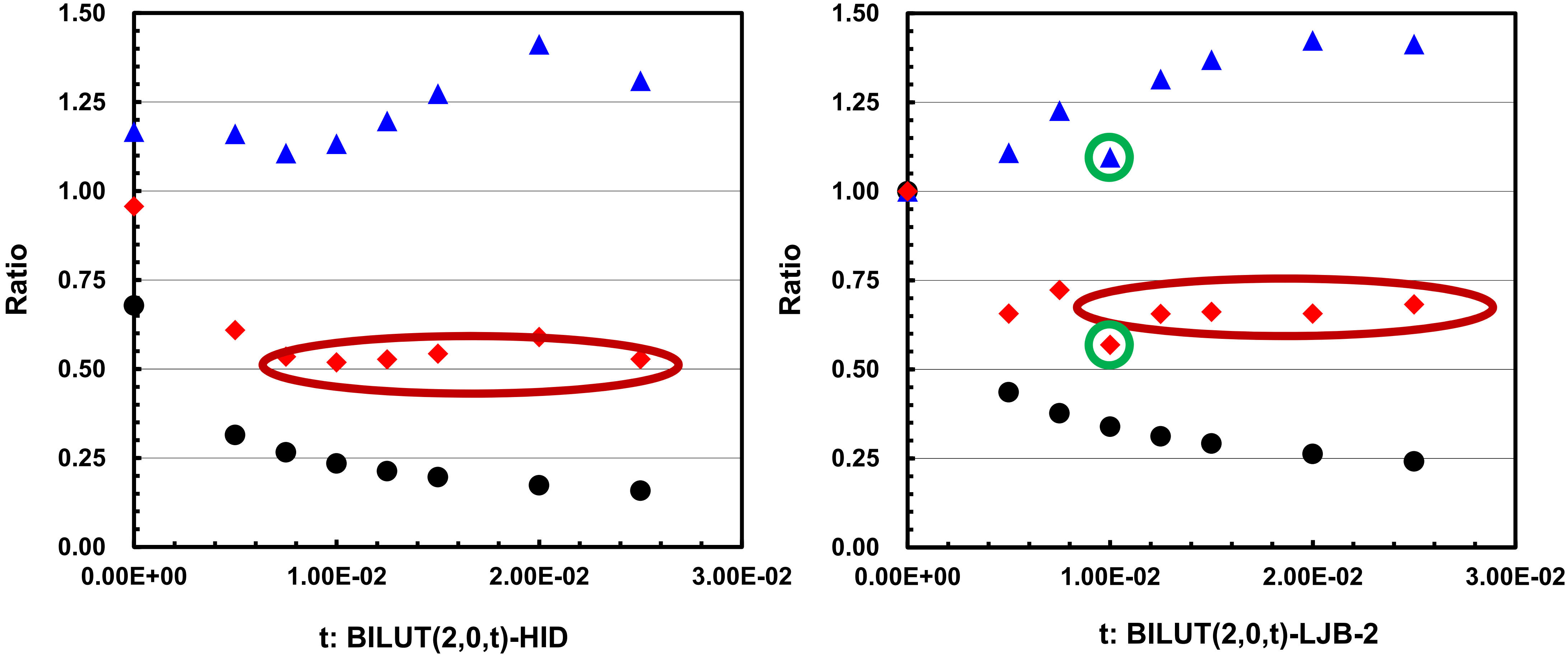
Effect of *t* on Performance

BILUT(2,0,*t*)-GPBi-CG with 240 nodes (3,840 cores)

$E_{max}=10^{-6}$, $E_{max}=10^{+6}$

Normalized by results of BILUT(2,0,0)-LBJ-2

●: [NNZ], ▲: Iterations, ◆: Solver Time



BILUT(p,0,0) at 3,840 cores NO dropping: Effect of Fill-in

Preconditioner	NNZ of [M]	Set-up (sec.)	Solver (sec.)	Total (sec.)	Iterations
BILUT(1,0,0)-LBJ-1	1.920×10^{10}	1.35	65.2	66.5	1916
BILUT(1,0,0)-LBJ-2	2.519×10^{10}	2.03	61.8	63.9	1288
BILUT(1,0,0)-LBJ-3	3.197×10^{10}	2.79	74.0	76.8	1367
BILUT(2,0,0)-LBJ-1	3.351×10^{10}	3.09	71.8	74.9	1339
BILUT(2,0,0)-LBJ-2	4.394×10^{10}	4.39	65.2	69.6	939
BILUT(2,0,0)-LBJ-3	5.631×10^{10}	5.95	83.6	89.6	1006
BILUT(3,0,0)-LBJ-1	6.468×10^{10}	9.34	105.2	114.6	1192
BILUT(3,0,0)-LBJ-2	8.523×10^{10}	12.7	98.4	111.1	823
BILUT(3,0,0)-LBJ-3	1.101×10^{11}	17.3	101.6	118.9	722
BILUT(1,0,0)-HID	1.636×10^{10}	2.24	60.7	62.9	1472
BILUT(2,0,0)-HID	2.980×10^{10}	5.04	66.2	71.7	1096

[NNZ] of [A]: 7.174×10^{9}, HID: Smaller number of NNZ

BILUT(*p*,0,0) at 3,840 cores NO dropping: Effect of Overlapping

Preconditioner	NNZ of [M]	Set-up (sec.)	Solver (sec.)	Total (sec.)	Iterations
BILUT(1,0,0)-LBJ-1	1.920×10^{10}	1.35	65.2	66.5	1916
BILUT(1,0,0)-LBJ-2	2.519×10^{10}	2.03	61.8	63.9	1288
BILUT(1,0,0)-LBJ-3	3.197×10^{10}	2.79	74.0	76.8	1367
BILUT(2,0,0)-LBJ-1	3.351×10^{10}	3.09	71.8	74.9	1339
BILUT(2,0,0)-LBJ-2	4.394×10^{10}	4.39	65.2	69.6	939
BILUT(2,0,0)-LBJ-3	5.631×10^{10}	5.95	83.6	89.6	1006
BILUT(3,0,0)-LBJ-1	6.468×10^{10}	9.34	105.2	114.6	1192
BILUT(3,0,0)-LBJ-2	8.523×10^{10}	12.7	98.4	111.1	823
BILUT(3,0,0)-LBJ-3	1.101×10^{11}	17.3	101.6	118.9	722
BILUT(1,0,0)-HID	1.636×10^{10}	2.24	60.7	62.9	1472
BILUT(2,0,0)-HID	2.980×10^{10}	5.04	66.2	71.7	1096

[NNZ] of [A]: 7.174×10^{9}

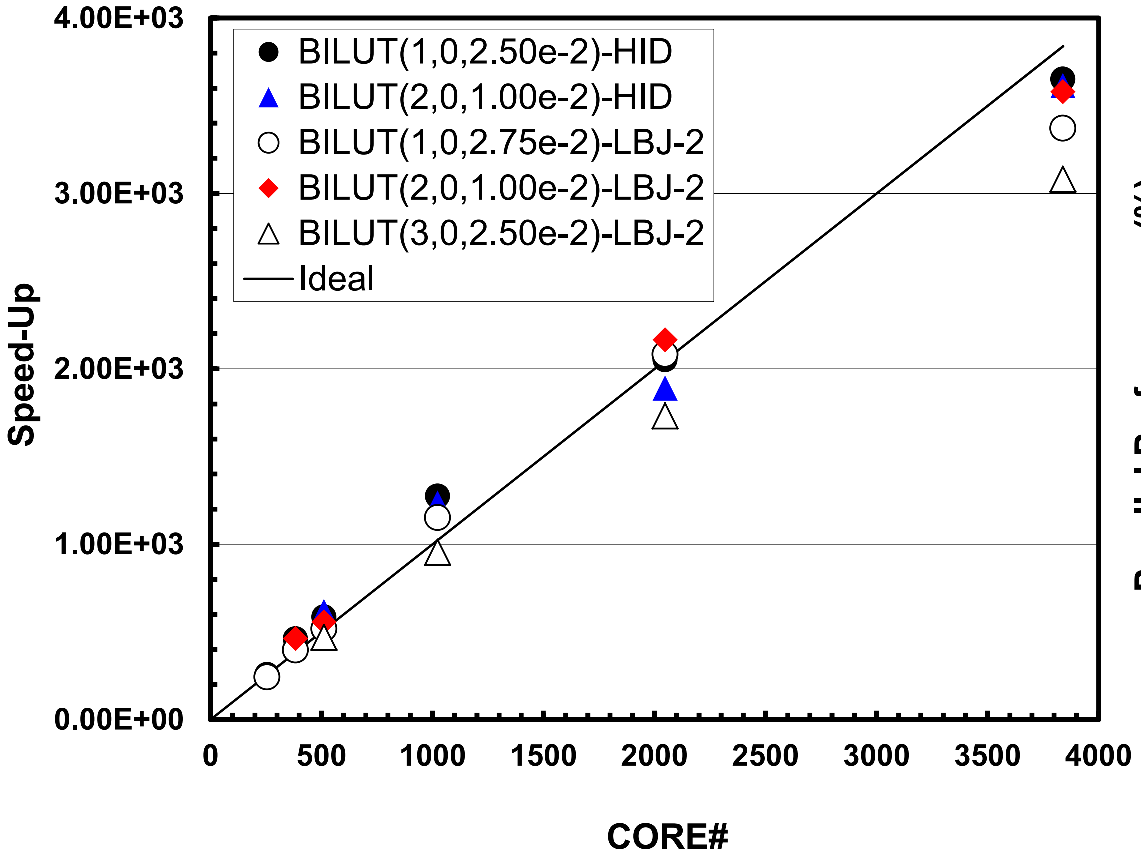
BILUT(p,0,t) at 3,840 cores Optimum Value of t

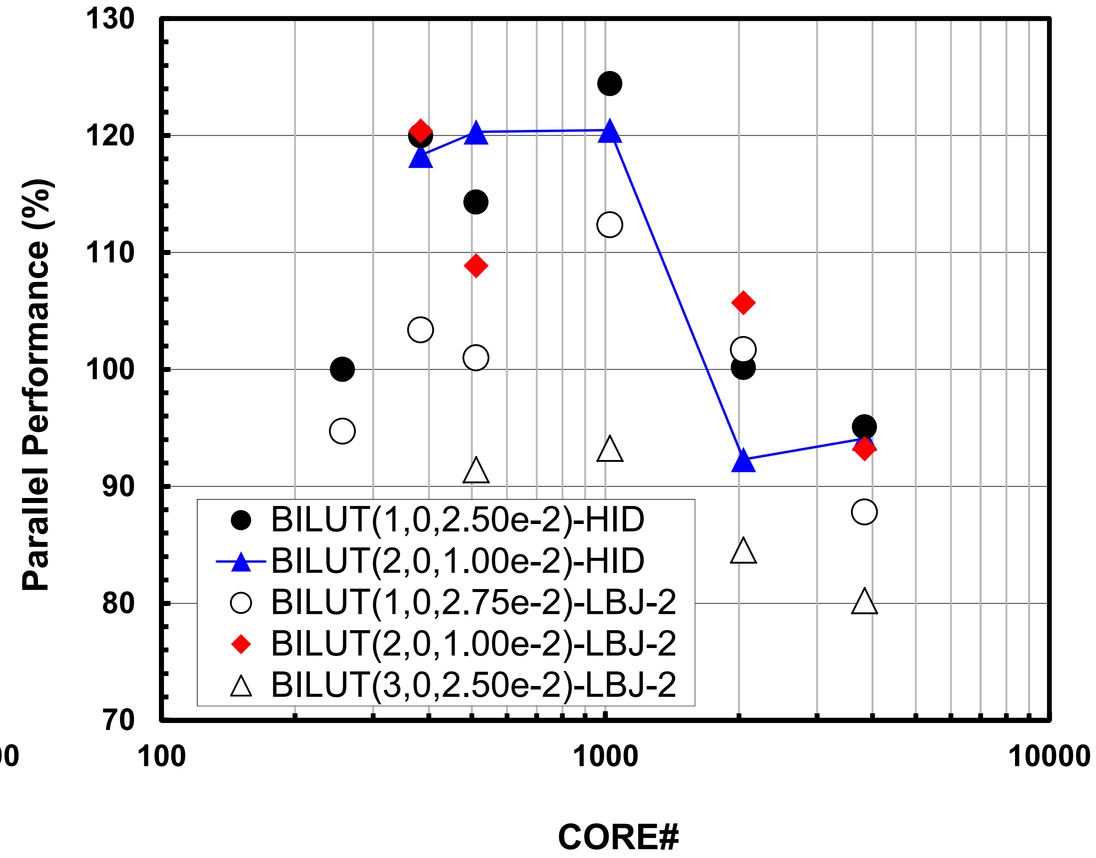
Preconditioner	NNZ of [M]	Set-up (sec.)	Solver (sec.)	Total (sec.)	Iterations
BILUT(1,0,2.75×10^{-2})-LBJ-1	7.755×10^{9}	1.36	45.0	46.3	1916
BILUT(1,0,2.75×10^{-2})-LBJ-2	1.019×10^{10}	2.05	42.0	44.1	1383
BILUT(1,0,2.75×10^{-2})-LBJ-3	1.285×10^{10}	2.81	54.2	57.0	1492
BILUT(2,0,1.00×10^{-2})-LBJ-1	1.118×10^{10}	3.11	39.1	42.2	1422
BILUT(2,0,1.00×10^{-2})-LBJ-2	1.487×10^{10}	4.41	37.1	41.5	1029
BILUT(2,0,1.00×10^{-2})-LBJ-3	1.893×10^{10}	5.99	37.1	43.1	915
BILUT(3,0,2.50×10^{-2})-LBJ-1	8.072×10^{9}	9.35	38.4	47.7	1526
BILUT(3,0,2.50×10^{-2})-LBJ-2	1.063×10^{10}	12.7	35.5	48.3	1149
BILUT(3,0,2.50×10^{-2})-LBJ-3	1.342×10^{10}	17.3	40.9	58.2	1180
BILUT(1,0,2.50×10^{-2})-HID	6.850×10^{9}	2.25	38.5	40.7	1313
BILUT(2,0,1.00×10^{-2})-HID	1.030×10^{10}	5.04	36.1	41.1	1064

[NNZ] of [A]: 7.174×10^{9}

Strong Scaling up to 3,840 cores

according to elapsed computation time (set-up+solver) for BILUT($1,0,2.5\times10^{-2}$)-HID with 256 cores





Related Work

- Selection of Threshold for ILUT (Single Processor)
  - Threshold, and max. number of components for each row
  - Y. Saad, ILUT: A dual threshold incomplete LU factorization., Numerical Linear Algebra with Applications (1994) 387-402
  - A. Gupta, and T. George, Adaptive Techniques for Improving the Performance of Incomplete Factorization Preconditioning, SIAM Journal on Scientific Computing, (2010) 84-100
    - Adaptive Approach
  - Jan Mayer; “Alternative Weighted Dropping Strategies for ILUTP,” SIAM Journal on Scientific Computing, vol. 27, no. 4, pp.1424-1437, (2006)
    - Weighting Dropping Strategy
  - Yong Zhang, Ting-Zhu Huang, Yan-Fei Jing and Liang Li, “Flexible incomplete Cholesky factorization with multi- parameters to control the number of nonzero elements in preconditioners”, Numerical Linear Algebra with Applications, vol. 19, Issue 3, pp.555-569, (2012)
    - Flexible Factorization
    - Number of non-zero components per row is controlled by heuristics

Related Work (cont.)

- Parallel Cases
  - Nakajima, K. and H.Okuda, Parallel Iterative Solvers for Simulations of Fault Zone Contact using Selective Blocking Reordering, Numerical Linear Algebra with Applications 11, 831-852 (2004)
  - Nakajima, K., Parallel Preconditioning Methods with Selective Fill-Ins and Selective Overlapping for Ill-Conditioned Problems in Finite-Element Methods, Lecture Notes in Computer Science 4489, 1085-1092. International Conference on Computational Science (ICCS 2007) (2007)
  - Nakajima, K., Strategies for Preconditioning Methods of Parallel Iterative Solvers in Finite-Element Applications on Geophysics, Advances in Geocomputing, Lecture Notes in Earth Science 119, 65-118 (2009)
  - Nakajima, K., Parallel Multistage Preconditioners by Extended Hierarchical Interface Decomposition for Ill-Conditioned Problems, Advances in Parallel ComputingVol.19 “From Multicores and GPU’s to Petascale”, IOS press, 99-106 (2010)
  - Hosoi, A., Washio, T., Okada, J., Kadooka, J., Nakajima, K., and Hisada, T., A Multi-Scale Heart Simulation on Massively Parallel Computers, ACM/IEEE Proceedings of SC10 (2010)

Summary

- Hetero 3D
- Generally speaking, HID is slightly more robust than LBJ with overlap extension
- BILUT(*p,d,t*)
  - effect of d is not significant
  - [NNZ] of [M] depends on *t* (not *p*)
  - BILU(3,0,t_0) > BILU(2,0,t_0) > BILU(1,0,t_0) for convergence, although cost of a single iteration is similar for each method
- Critical/optimum value of *t*
  - [NNZ] of [M] = [NNZ] of [A]
  - Further investigation needed.

Future Works

- Theoretical/numerical investigation of optimum *t*
  - Eigenvalue analysis etc.
  - Final Goal: Automatic selection BEFORE computation
  - Procedures of existing works for a single CPU
- Further investigation/development of LBJ & HID
- Comparison with other preconditioners/direct solvers
  - (Various types of) Low-Rank Approximation Methods
  - MUMPS…
- Extention of Hetero 3D
  - OpenMP/MPI Hybrid version
    - BILU(0) is already done, factorization is (was) the problem
  - Extension to Manycore/GPU clusters