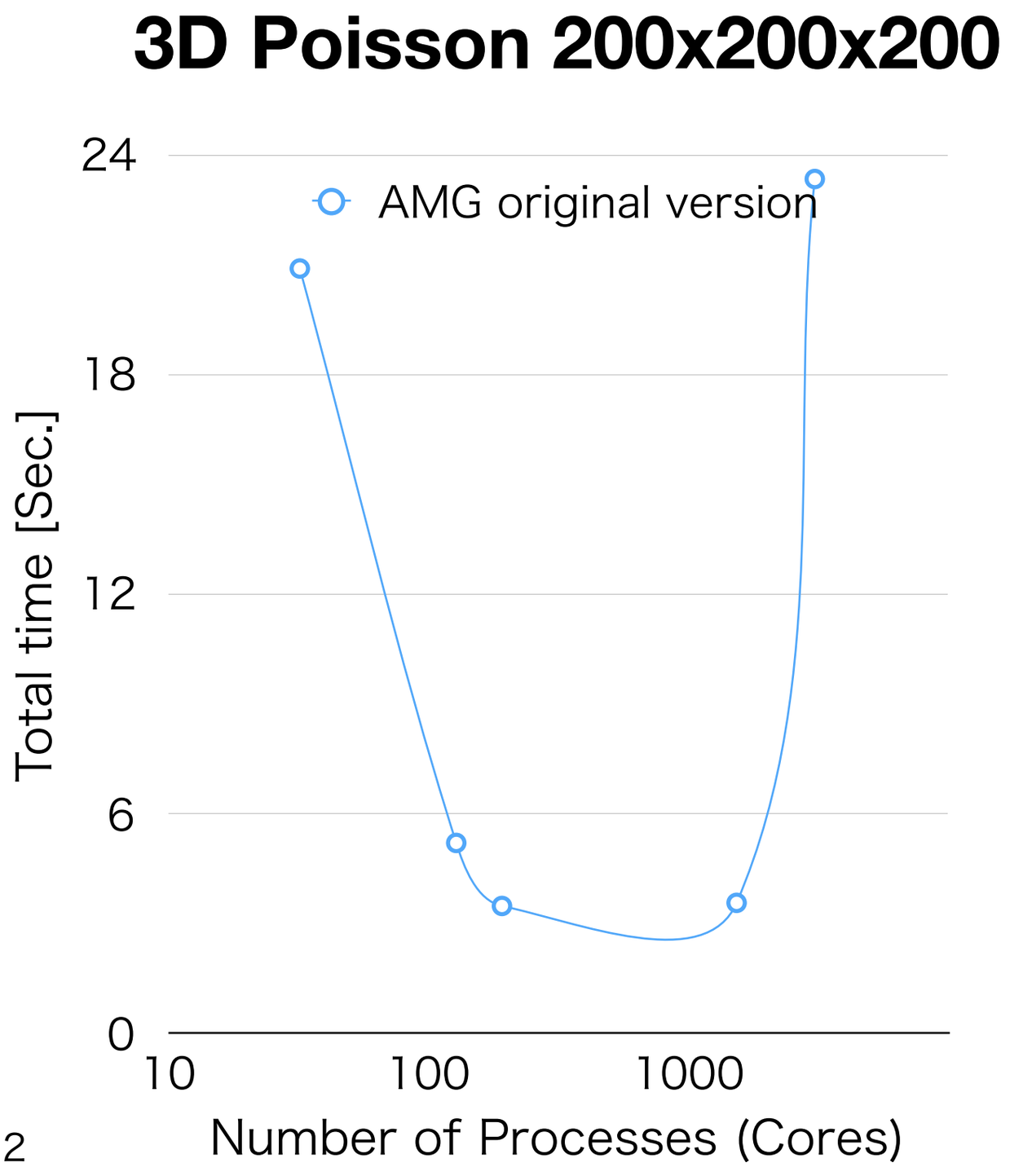
Algebraic Multigrid Solver using Coarse Grid Aggregation with Independent Aggregation

Takuya Nomura, Akihiro Fujii, Teruo Tanaka [Kogakuin Univ.],
Osni Marques[LBNL]
Kengo Nakajima [Tokyo Univ.]

SA-AMG solver

- Multigrid linear solver for
  Ax=b
- It has a “reduction” structure
  - Coarse grids are much smaller than the fine grids.



Outline

- SA-AMG
  - Aggregation strategies
  - Related works
  - Our method
  - Implementation
  - Numerical tests

SA-AMG Method

Setup part

1. aggregation of unknowns
2. Calculate interpolation matrix P
3. $A_c = P^t A P$

Iterative part

1. Gauss Seidel smoother is used alternately between Fine level A and Coarse level Ac

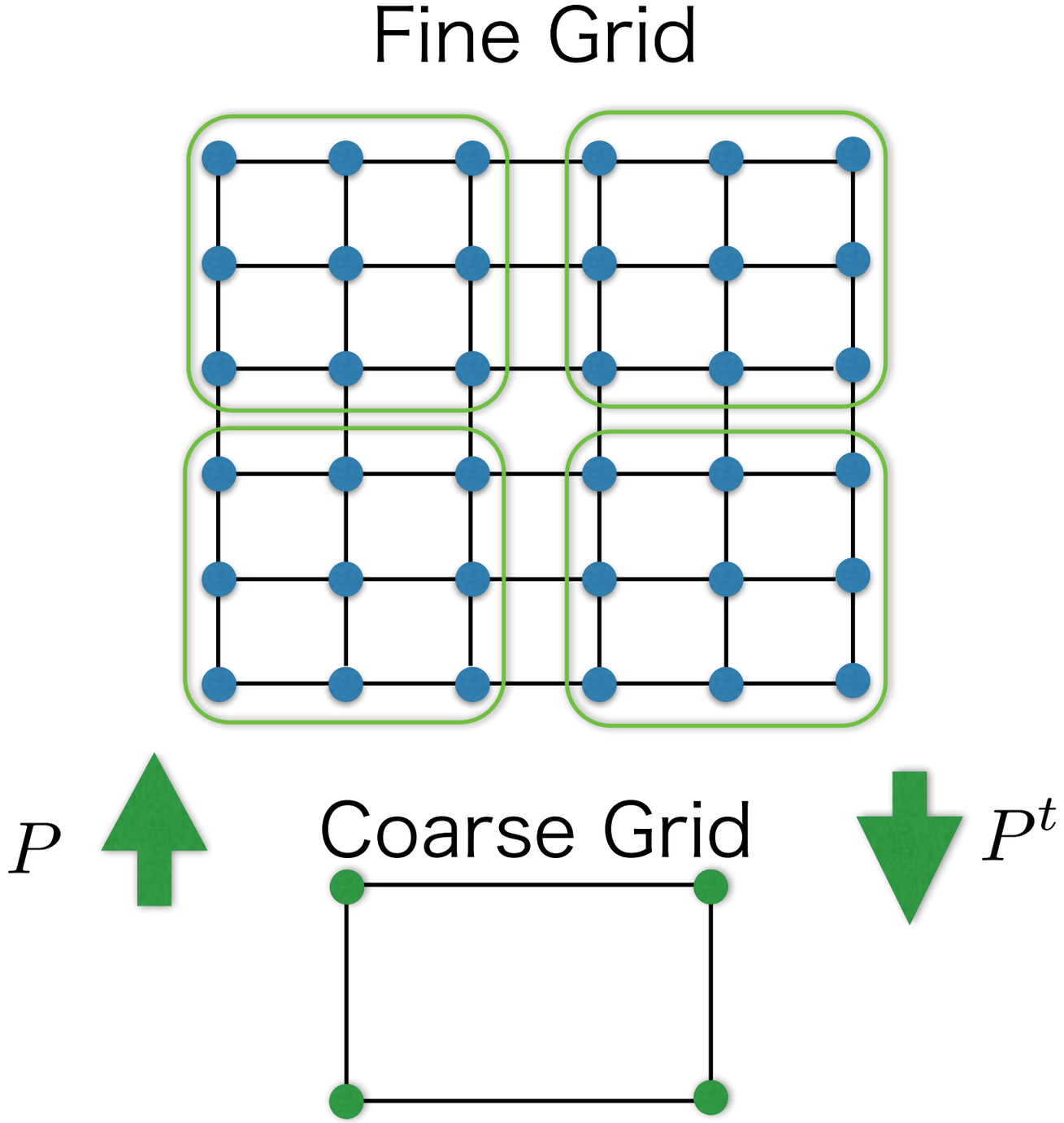
SA-AMG Method

- Fine Grid is Original problem
  - node：Unknown variable
  - edge：non-zero elements of problem matrix

- Aggregation
  - Groups the unknowns

- Interpolation matrix P is calculated from aggregates
  - In this case, the matrix P prolong 4 unknowns to 36 unkowns

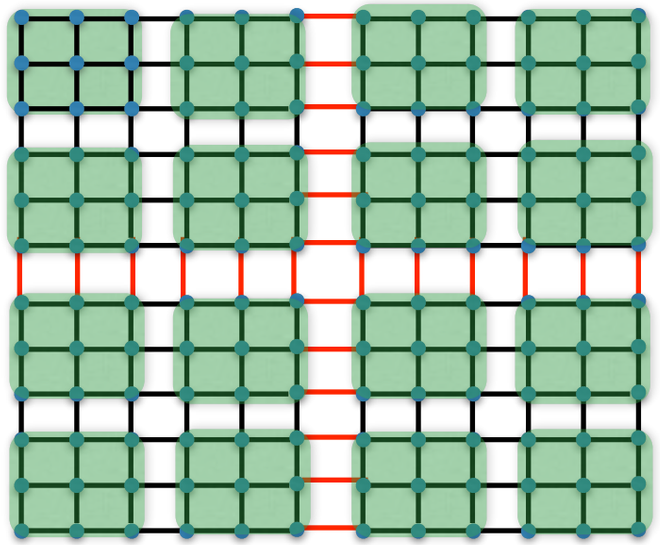
- Generation of coarse matrix
  - $A_c = P^t A P$



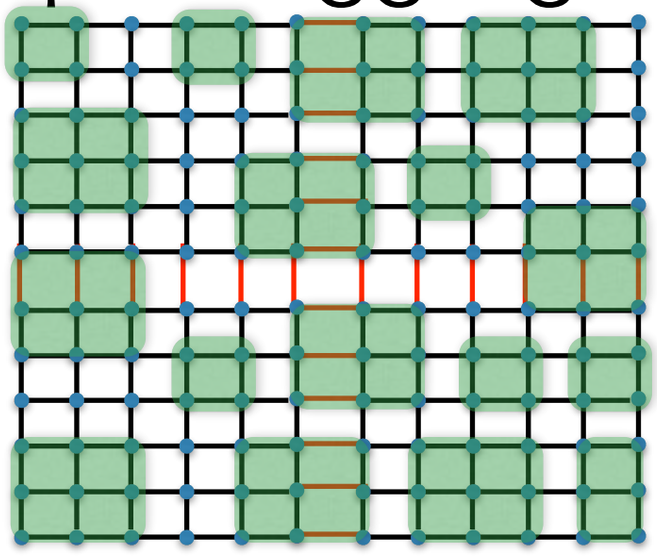
Aggregation strategy

- Independent aggregation
  - Aggregates do not go beyond borders. Each process domain has at least one aggregate.
  - It does not work for anisotropic problems. Process domain border hinders the aggregates
- Coupled aggregation
  Aggregation on domain borders is done at first.
  - It can create aggregates irrelevant of process domain borders
  - It works for anisotropic problems

Independent aggregation



Coupled aggregation



Related works on coarse grid solvers

- Sunder et. al. proposed to use geometric multigrid as a coarse grid solver.
  Sunder et. al. : ”Parallel geometric-algebraic multigrid on unstructured forests of octrees”, SC12

- Williams et. al. proposed to use CABICGSTAB as a coarse grid solver
  Williamsら: “s-step krylov subspace methods as bottom solvers for geometric multigrid" IPDPS 2014

- Coarse Grid re-districution
  Adams:"A parallel maximal independent set algorithm”, UCB Tech. Rep. 1998
  Nakajima: ”Openmp/mpi hygrid parallel multigrid method on fujitsu FX10 supercomputer system”, Cluster Computing Workshops, 2012

- Coarse Grid re-distribution and independent aggregation
  P. T. Lin, “Improving multigrid performance for unstructured mesh drift–diffusion simulations on 147,000 cores,” International Journal for Numerical Methods in Engineering, Aug. 2012.

idea is similar with Lin’s study. Our method didnot re-distribute the coarse grids

Ordinary implementation matrix repartition

1. Independent aggregation
2. Interpolation mat. P
3. $A_c = P^t A P$
4. Matrix repartition

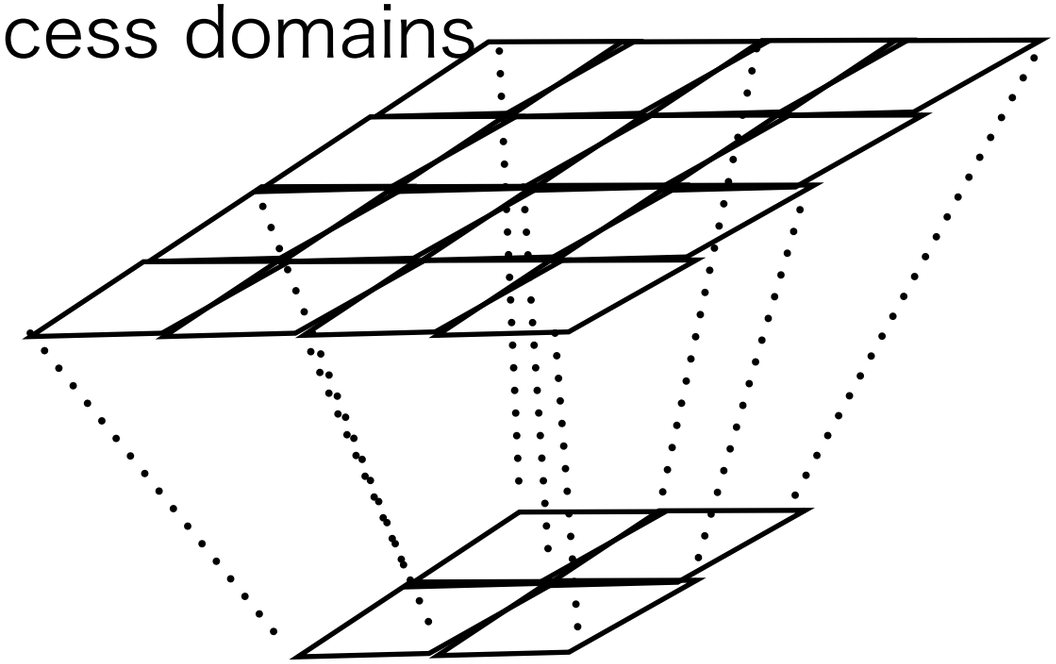
Our method

1. Independent aggregation
   1. Create graph representing the processes with aggregates of unknowns
   2. ParMETIS couples the process domains
   3. Change aggregate number sequential according to process coupling
2. Interpolation mat. P
3. $A_c = P^t A P$

Our method

- It uses independent aggregation with Coarse Grid Aggregation based on process domain coupling
  - ParMETIS library is used for process domain coupling
- Features
  - Parallelism adjusted Coarse Grid is created at once
  - It works for anisotropic problems

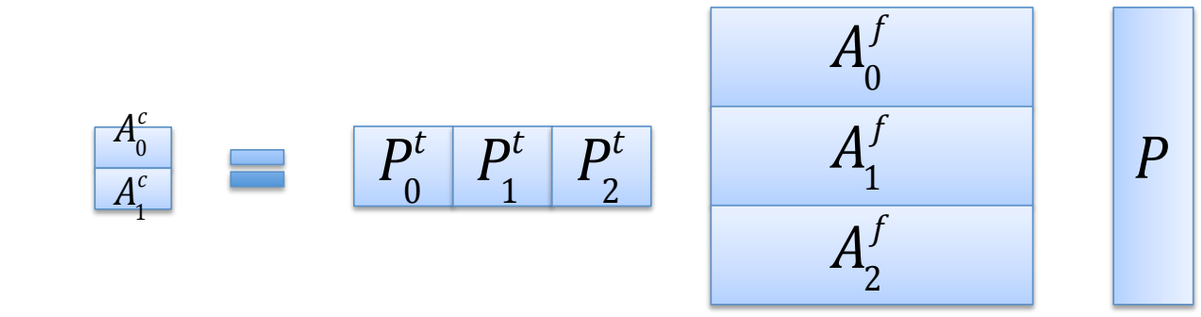
Fine Grid
16 process domains

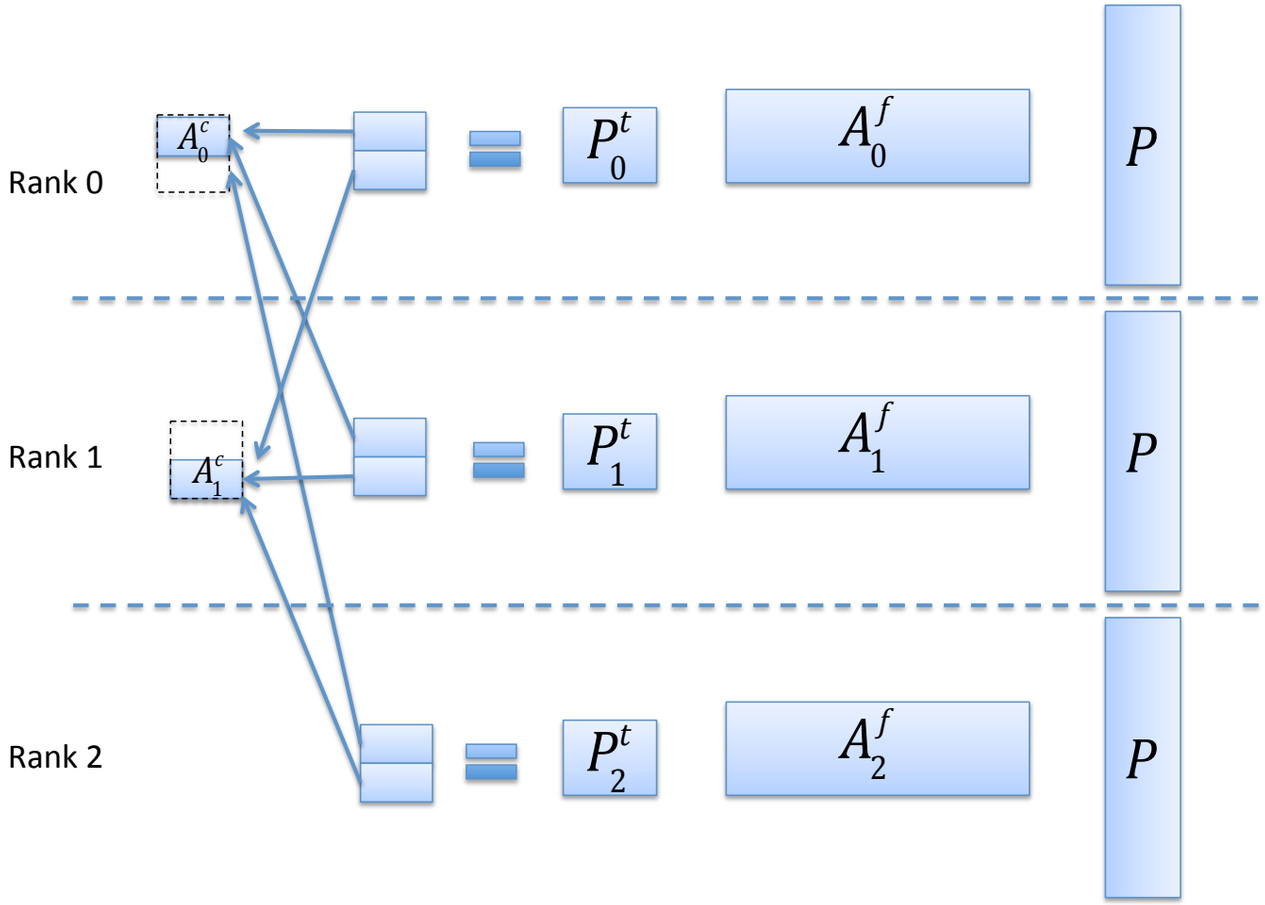


Coarse Grid
4 process domains

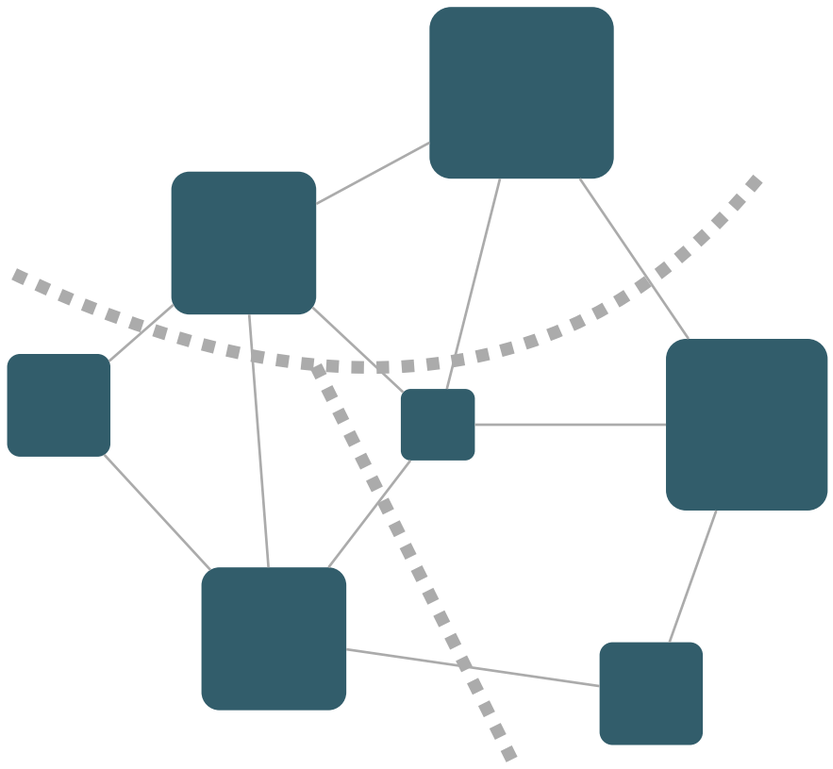
Coarse Grid Aggregation

- For unstructured matrices, there are few studies on CGA.
- SA-AMG can adjust the coarse level parallelism without matrix redistribution
- Block row widths of coarser level mat. can be adjusted when the coarser level is calculated.
- coarser lev. mat. can be reordered by the column ordering of mat. P





Coarse grid aggregation



1. Determine the degree of parallelism on the coarser level
2. Make a graph showing process domains
3. Coupling processes is done with graph partitioner (ParMETIS)
4. Adjust the block row widths and column vector order of P based on aggregated domains
5. 3-matrix product $P^t AP$

Nodes: aggregates of unknowns in each process' domain

Edges: finer level's halos connect domains

Numerical test

- Anisotropic problems
  Poisson problem with 1/100 times anisotropy in z axis direcsion on 3 dimensional cubic domain

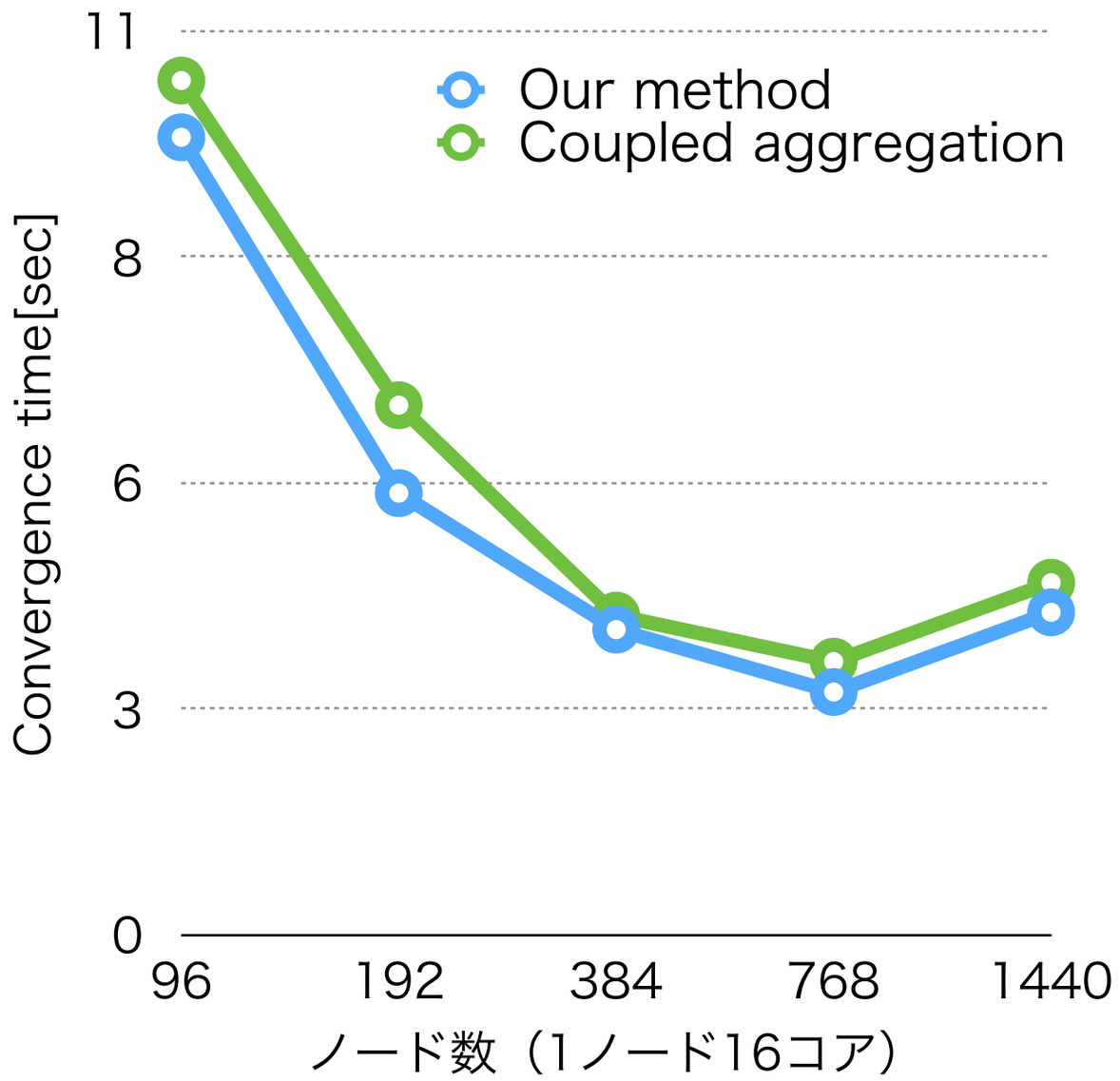
- Poisson problems appearing in Darcy flow problems
  Diffusion coefficient heterogeneously changes from 10^{-5} to 10^{5}

- FX10 Supercomputer at Tokyo University is used.
  1 node with 16 cores. We executed the program with 1 process 16 threads on a node.

- Our method aggregates the process domains so that each domain has at least 1000 unknowns

Anisotropic problem

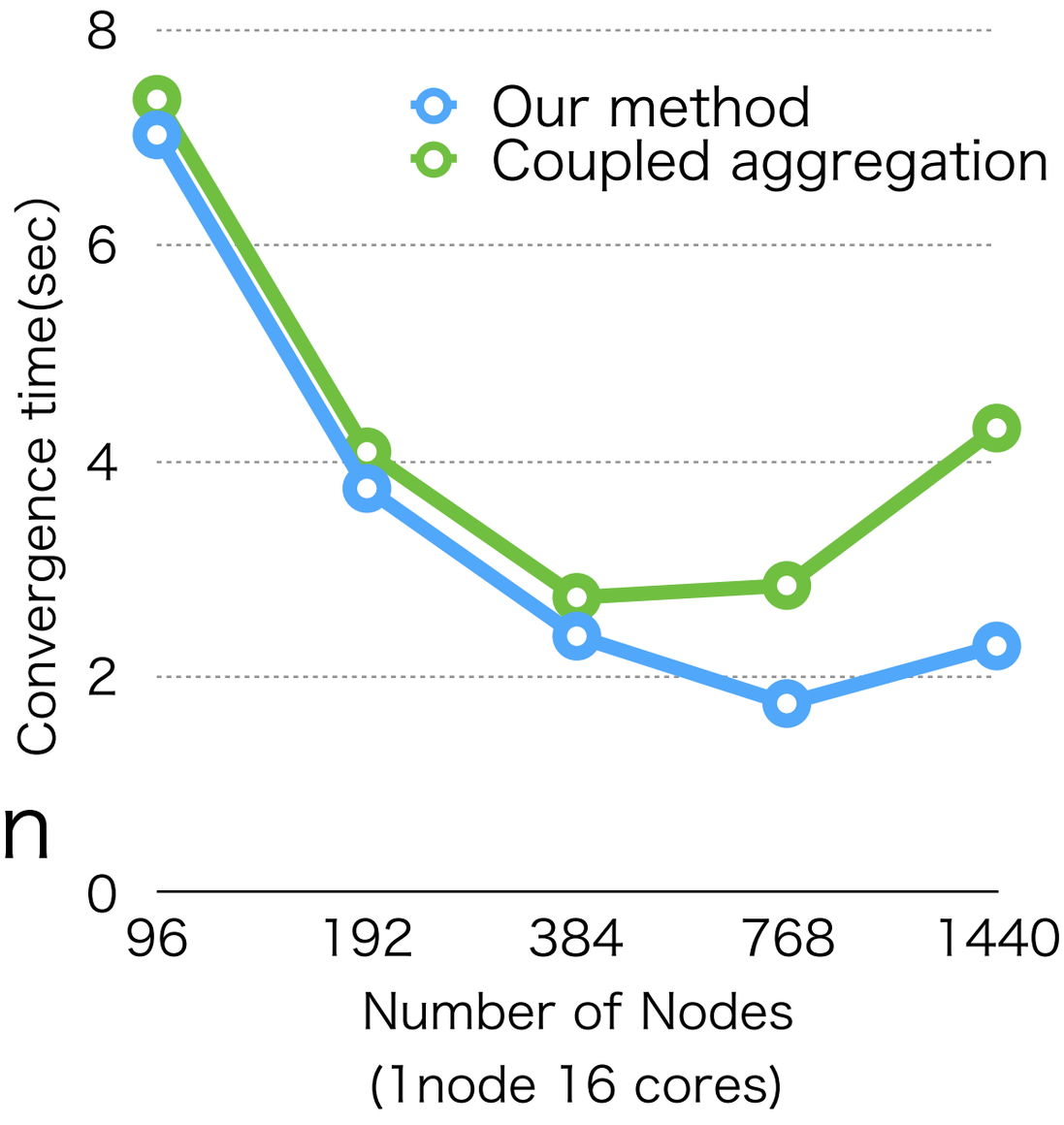
- Problem domain size is fixed at 300x300x300
- Independent aggregation doesnot converge within 500 iterations with 768 nodes
- Our method using independent aggregation works as well as Coupled aggregation



Darcy flow problem
Strong scaling test

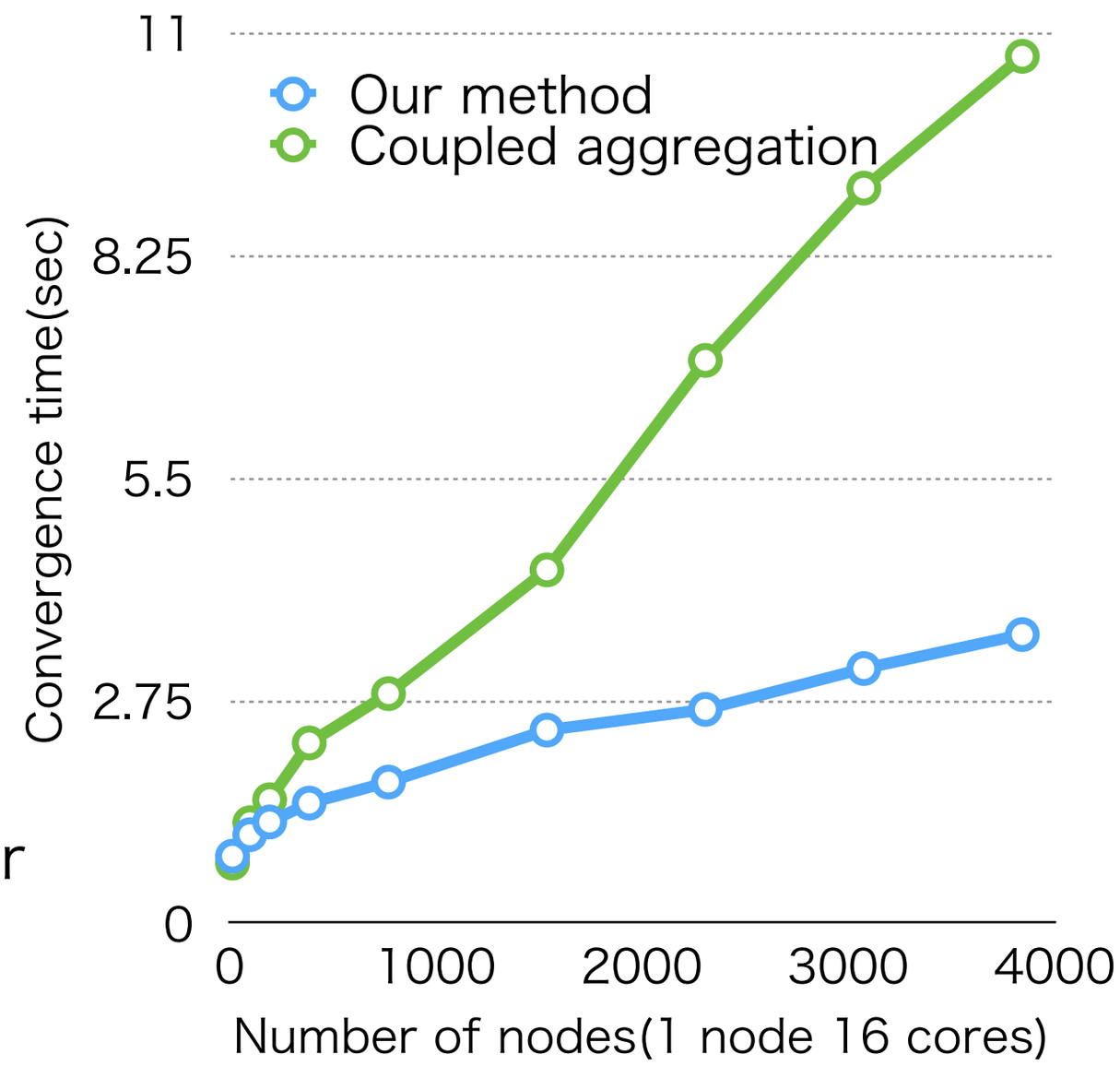
- Problem domain size is fixed at 300x300x300. Strong scaling setup

- Our method is 1.6 times faster than coupled aggregation with 768 nodes



Darcy flow problem
Weak scaling test

- Problem domain is fixed as 32x32x32 per process. Weak scaling setup
- Although Iteration number for convergence is the same level, Coupled aggregation much slower than our method
- Our method can reduce the time for communication at the coarser levels, and it becomes faster



Darcy flow problem

Weak scaling test

	Coupled aggregation			Our method		
Nodes	Iter	Setup	Itertive	Iter	Setup	Iterative
12	12	0.454	0.29	16	0.453	0.377
96	15	0.612	0.63	14	0.592	0.503
192	17	0.738	0.79	15	0.692	0.561
384	20	1.04	1.19	16	0.820	0.666
768	16	1.51	1.33	16	1.05	0.741
1536	17	2.63	1.74	18	1.44	0.949
2304	22	4.27	2.69	17	1.70	0.945
3072	24	5.99	3.10	19	2.06	1.09
3840	22	7.57	3.15	19	2.46	1.11

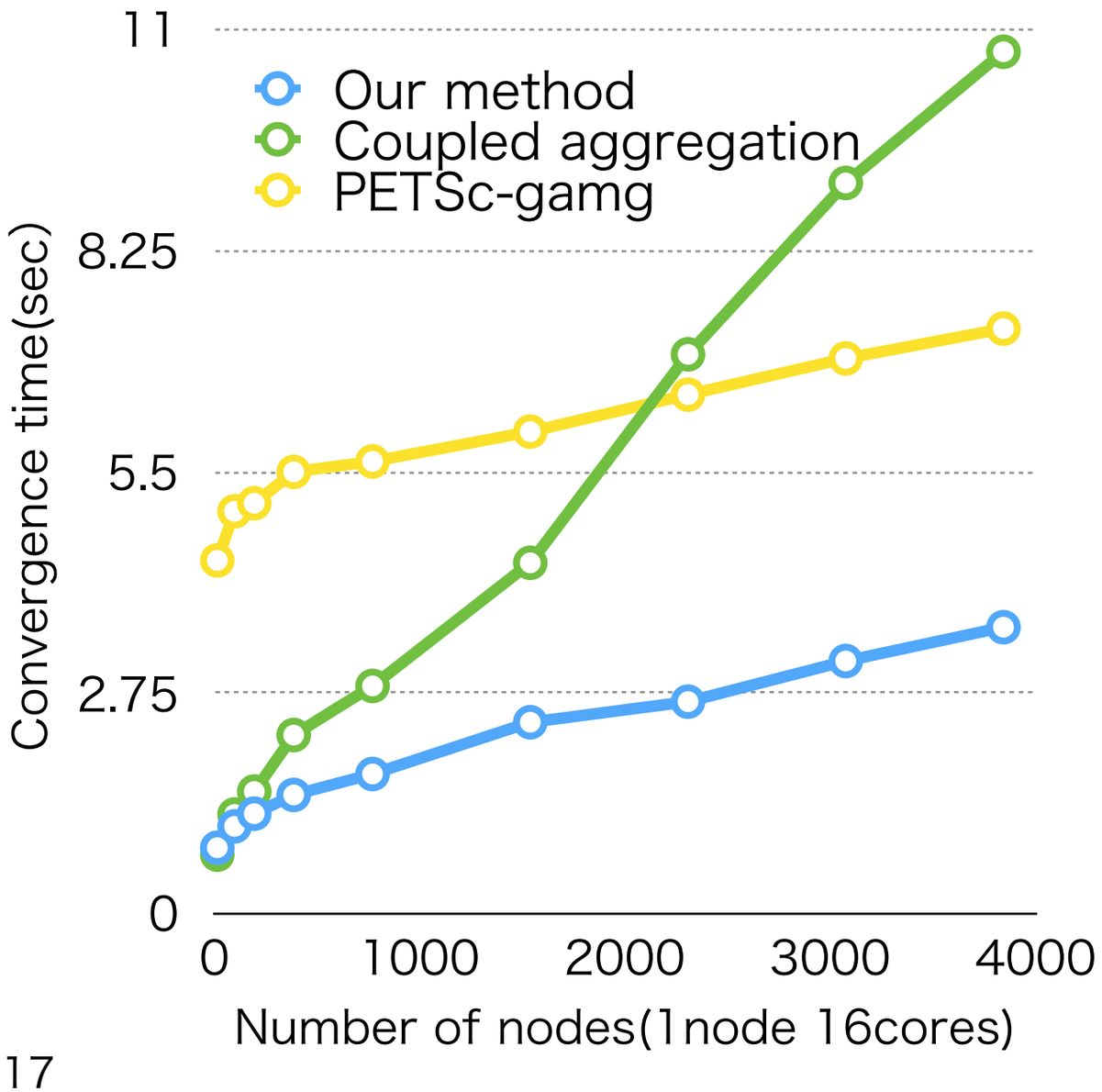
Unit : times, sec, sec

Comparison with PETSc-gamg

- Darcy flow problem:
  Weak Scaling

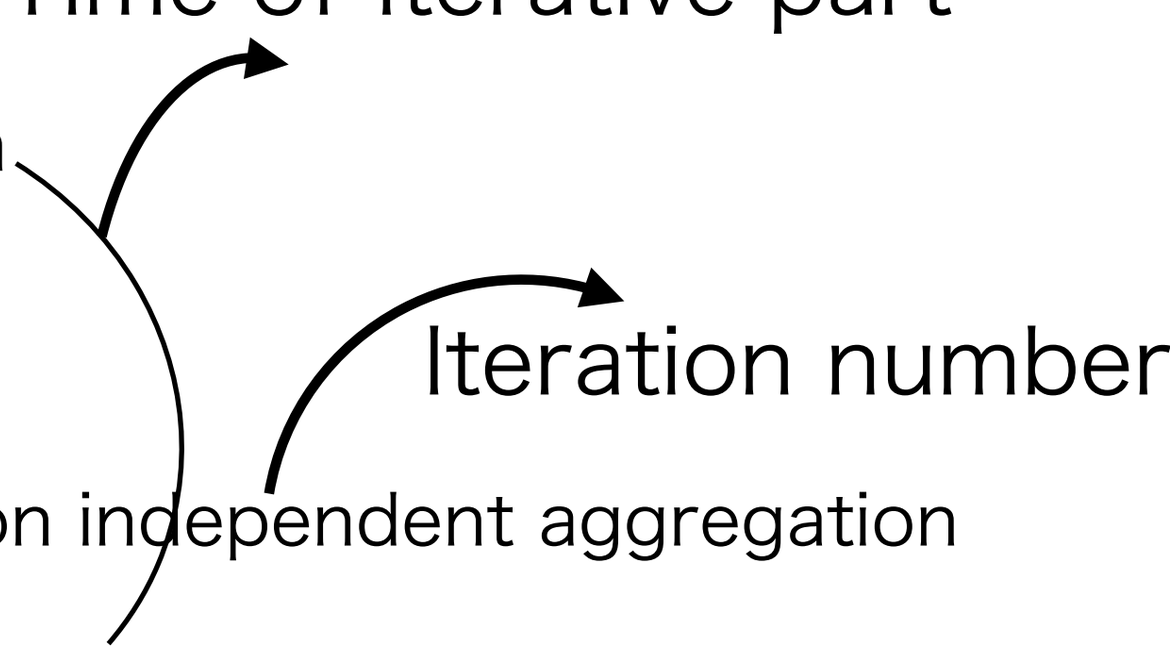
- PETSc is 1node 1process flat MPI model.
  - Multiple processes on one node execution becomes slower then 1 process per node.

- PETSc-gamg with default parameter setting shows the performance baseline.

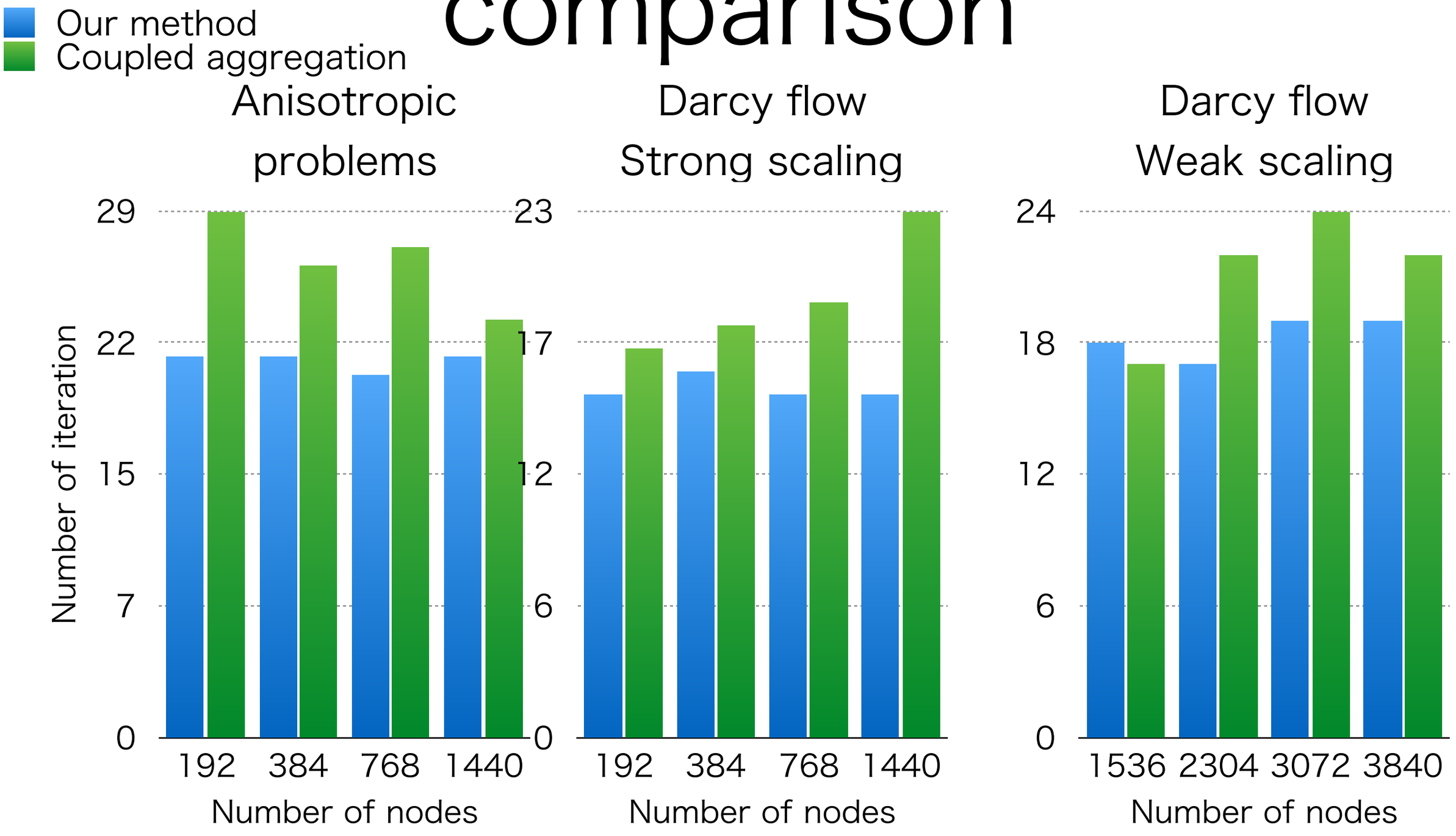


Cost and merit of our method

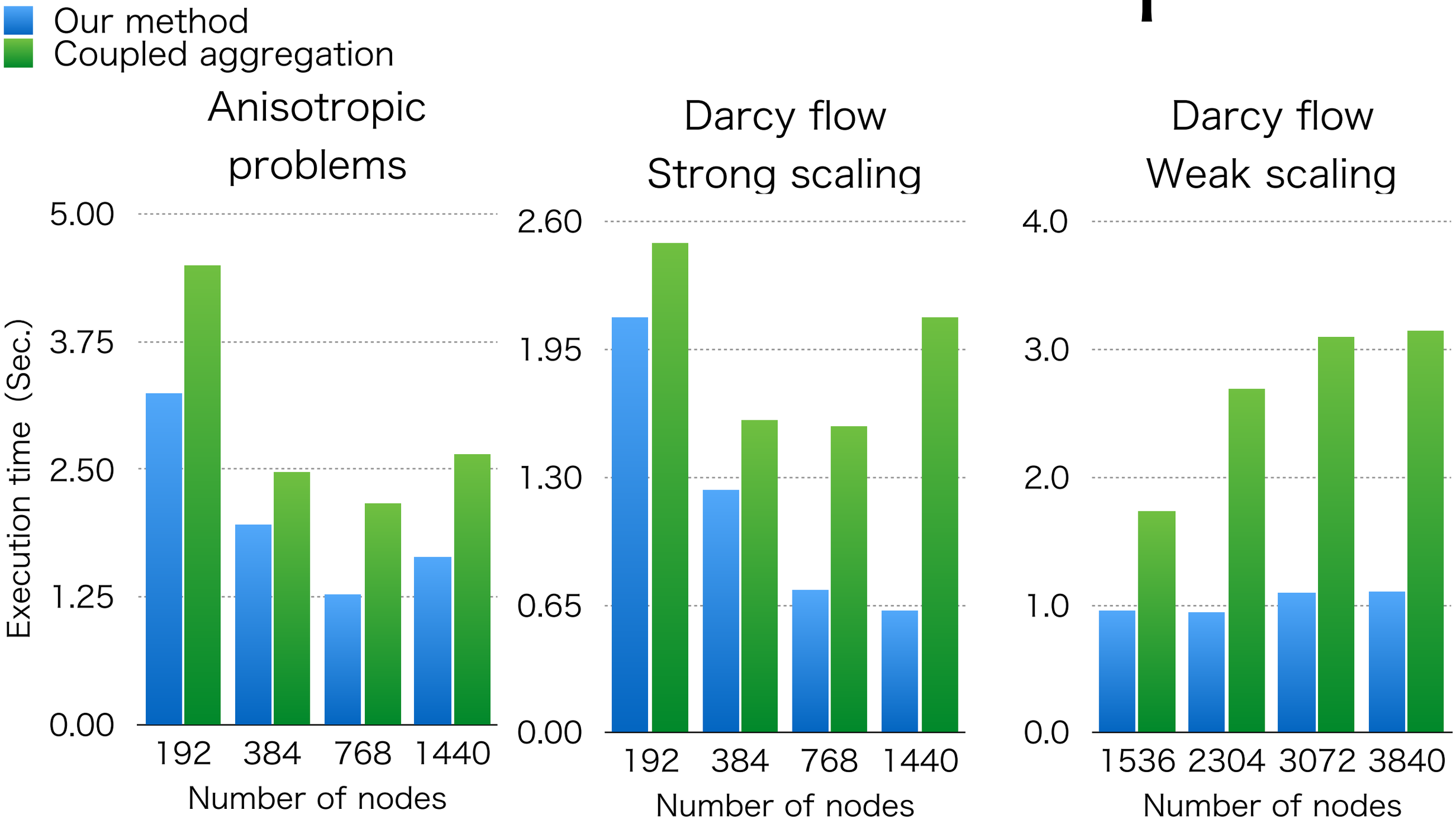
- Cost
  - Calculation cost for aggregation of process domains : setup part
    - Lowered degree of parallelism
- Merit
  - Improve convergence based on independent aggregation
  - Reduce the number of too small process domains and communication cost.



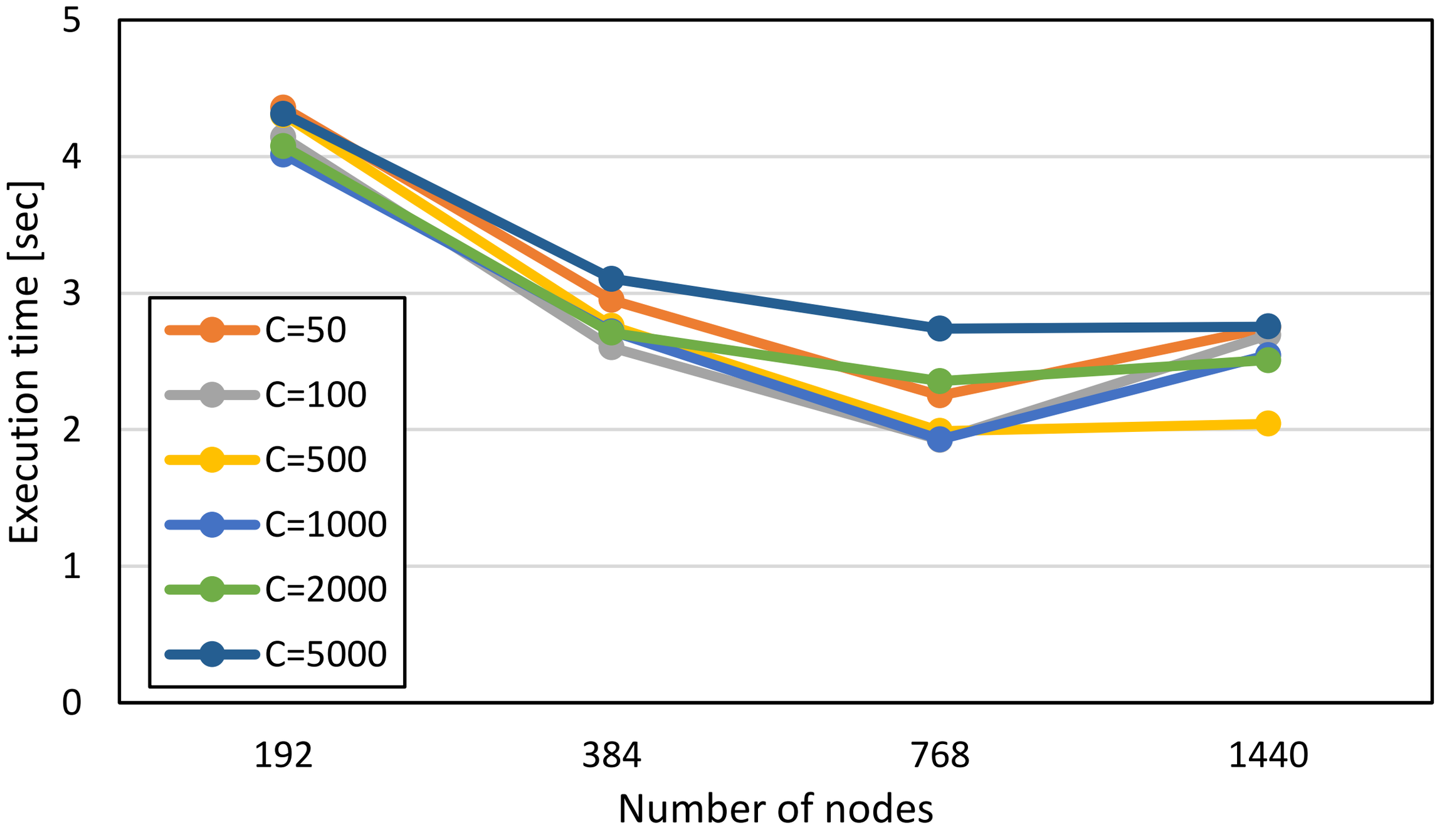
Iteration number comparison



Time of iterative part



Darcy flow,Strong scaling
CGA parameter C is specified by minimum size of a process domain



Conclusion and Future works

- Our method calculates the parallelism adjusted coarser level matrix at once
  - It uses independent aggregation and couples process domains
- We evaluated the method with anisotropic problems and Darcy flow problems in comparison with coupled aggregation
  - Our method works well with all problems
  - Execution time was accelerated from coupled aggregation more than 3 times at most
- CGA parameter strongly influences the SA-AMG performance.
  We will study the tuning method of the parameter