jh190022-NAJ

鈴木 厚（大阪大学 サイバーメディアセンター）

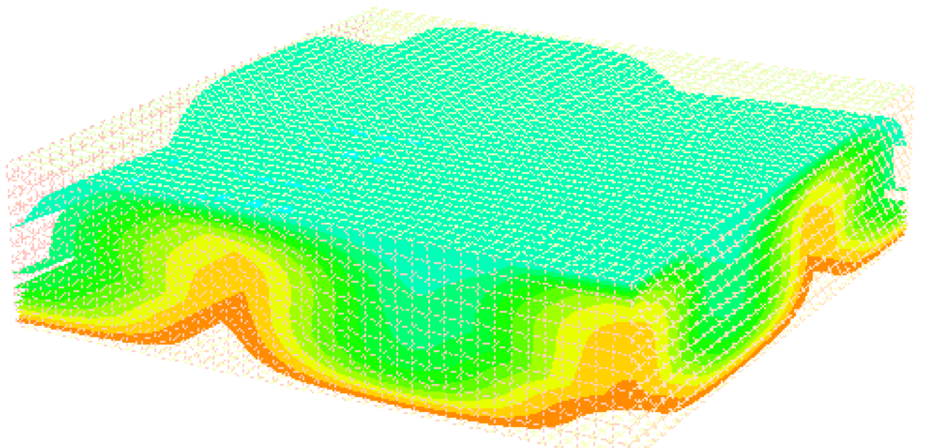
High performance simulations using FreeFem++ on mixed distributed-plus shared-memory architecture

abstract

We are focused on the finite element method to discretize partial differential equations (PDE). FreeFem++ developed by F. Hecht in LJLL, UPMC is a powerful tool to describe weak formulation of the PDE and it can connect to modern domain decomposition solver, “HPDDM” (by P. Jolivet) for distributed parallelism. The aim of this project is to combine efficient direct solver “Dissection” (by A. Suzuki), inside one node through shared-memory parallelism and HPDDM with “GenEO” preconditioner for numerical robustness of mixed direct- and iterative solver.

target problem

Domain specific language FreeFem++ is designed to solve nonlinear partial differential equations for industrial problem, e.g., fluid-structure interaction, fluid-thermal coupling, semi-conductor, electro-magnetic problems.



Rayleigh-Benard thermal convection

computing resource, and strategy for linear solver

Time dependent nonlinear system needs to be solved with internal linear solver for large sparse matrix with high condition number.

Linear system consists of 10 to 100 million DOF, which is arising from adaptive mesh refinement technique to 100^3-based mesh for 5 or 7 physical components.

DOF	1 million	10 million	100 million
solver	direct solver Pardiso/MUMPS/Dissection	Krylov method + preconditioner additive Schwarz	iterative substructuring method BDD/GenEO
hardware	16core/64GB	160core/640GB	512core/2TB

member of the project

Atsushi Suzuki : Cybermedia Center, Osaka University
Pierre Jolivet : CNRS, ENSEEIHT, France
Daisuke Furihata : Cybermedia Center, Osaka University

target system

octopus (Intel Xeon Gold 6126), NEC SX-ACE
@ Cybermedia Center, Osaka University

Numerical simulation by FreeFem++

```
load "Dissection"
defaulttoDissection();
real Pr = 7.0;  real Ra = 1500.0;
macro d11(u1)     dx(u1) //
macro d12(u1,u2)  (dy(u1) + dx(u2))/2.0 //
macro ugrad(u1,u2,u3,v) (u1*dx(v) + u2*dy(v) + u3*dz(v)) //
macro Ugrad(u1,u2,u3,v1,v2,v3) [ugrad(u1,u2,u3,v1),
                ugrad(u1,u2,u3,v2), ugrad(u1,u2,u3,v3)] //
mesh3 Th=readmesh3("mesh.data");
fespace Wh(Th,[P2,P2,P2,P1,P2]);
varf vDRB ([u1,u2,u3,p,th],[v1,v2,v3,q,psi]) =
  int3d(Th)( 2.0*(d11(u1)*d11(v1)+
                  d22(u2)*d22(v2)+
                  d33(u3)*d33(v3)+
                  2.0*d12(u1,u2)*d12(v1,v2)+
                  2.0*d13(u1,u3)*d13(v1,v3)+
                  2.0*d23(u2,u3)*d23(v2,v3))
    - p * div(v1, v2, v3) - q * div(u1, u2, u3)
    + (Ugrad(u1,u2,u3,up1,up2,up3)'*[v1,v2,v3] // '
    + Ugrad(up1,up2,up3,u1,u2,u3)'*[v1,v2,v3]) / Pr) //'
  - int3d(Th)( Ra * th * v3 )
  + int3d(Th)( dx(th)*dx(psi)+dy(th)*dy(psi)+dz(th)*dz(psi) )
  + int3d(Th)( ugrad(up1,up2,up3,th)*psi
             + ugrad(u1,u2,u3,thp)*psi )
  + on(1,3,u2=1.0) + on(2,4,u1=1.0) + on(5,6,u3=1.0)
  + on(5,6,th=1.0);
matrix K=vDRB(Wh,Wh,solver=sparsesolver);
```

definition of a stiffness matrix by weak formulation in Newton iteration for Rayleigh-Benard thermal convection equation (stationary Navier-Stokes eqs. + heat equation)

iterative substructuring-type domain decomposition method

Neumann-Neumann preconditioner

$$A^{(p)} = \begin{bmatrix} A^{(p)}_{II} & A^{(p)}_{IB} \\ A^{(p)}_{BI} & A^{(p)}_{BB} \end{bmatrix}$$ local stiffness matrix with Neumann b.c.

$$S^{(p)} = A^{(p)}_{BB} - A^{(p)}_{BI} A^{(p)-1}_{II} A^{(p)}_{IB}$$ local Schur complement

$S^{(p)\dagger}$ inverse on $\mathrm{Im}(S^{(p)})$ when subdomain is floating

$$S = \sum_{1\leq p\leq M} R^{(p)T} S^{(p)} R^{(p)}$$ global Schur complement

$A\,x = f \;\Rightarrow\; S\,y = g$ interface problem

$$Q_{\mathrm{NN}} = \sum_{1\leq p\leq M} R^{(p)T} D^{(p)} S^{(p)\dagger} D^{(p)} R^{(p)}$$

Balancing Neumann-Neumann preconditioner

kernel of the local Schur complement consists of zero energy mode (e.g. rigid body modes for elasticity problem)

$$\mathrm{Ker}S^{(p)} = R^{(p)}_B \mathrm{Ker}A^{(p)}$$

coarse space is used to accelerate convergence of the interface probem

$$v = \sum_{1\leq p\leq M} R^{(p)T} D^{(p)} v^{(p)} \qquad v^{(p)} \in \mathrm{Ker}S^{(p)}$$

$$Q_{\mathrm{BNN}} S = (I-P) Q_{\mathrm{NN}} (I-P^T) S + P$$

no necessary to form Schur complement nor BNN preconditioner

GenEO preconditioner

coarse space is enriched by spectral information with eigenvalue and egienvector $(\lambda^{(p)}, v^{(p)})$ of a generalized eigenvalue problem

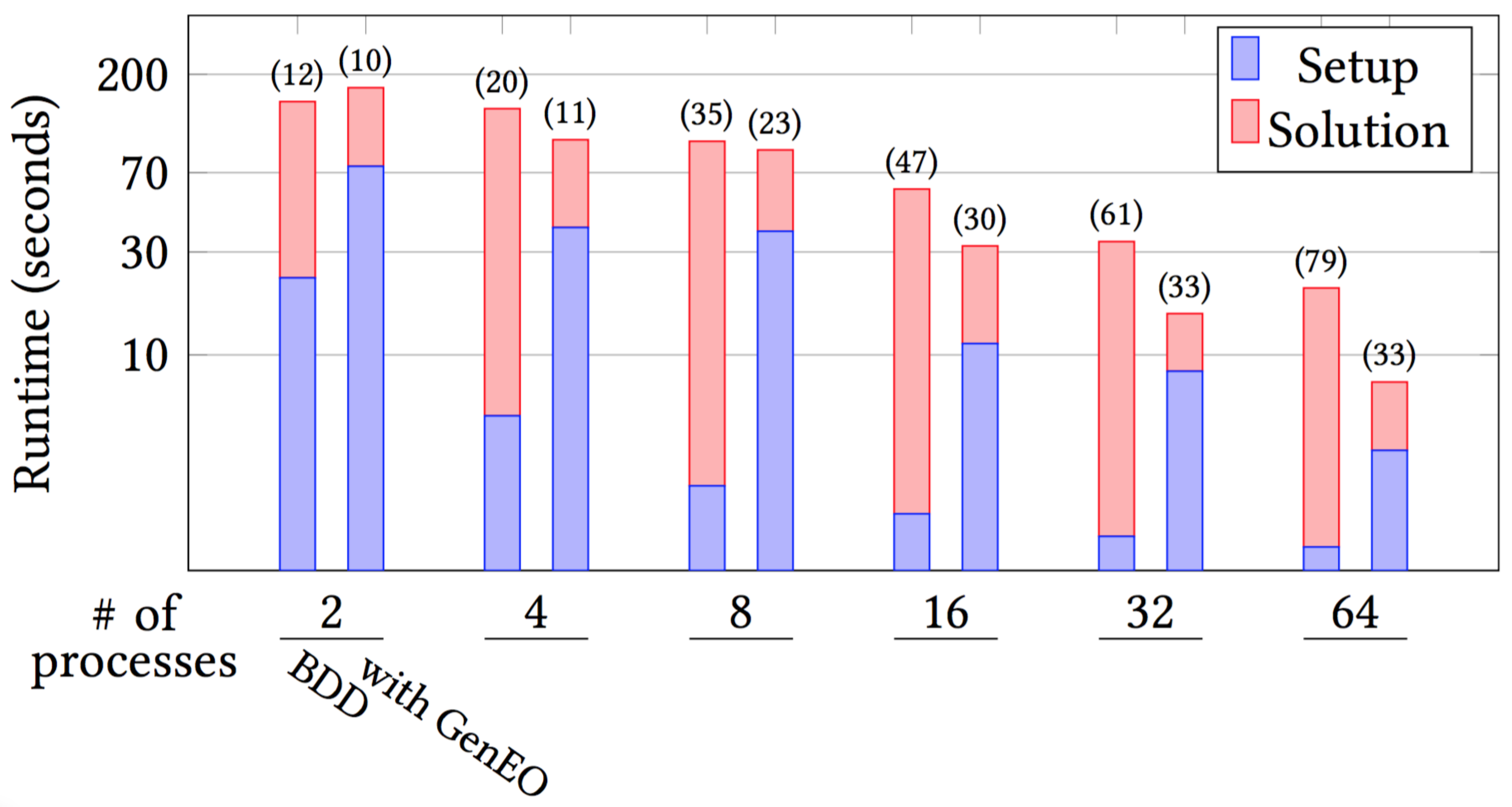
$$S^{(p)} v^{(p)} = \lambda^{(p)} D^{(p)} R^{(p)} S R^{(p)T} D^{(p)} v^{(p)}$$

ARPACK is used to compute the eigenvalue problem with dense local Schur compelemt.

local Schur complement needs to be formed explicitly for eigenvalue analysis

zero eigenvalue corresponds to zero energy mode $\lambda^{(p)} = 0$

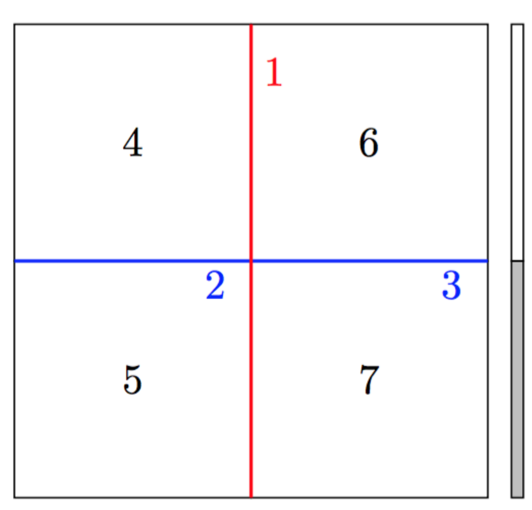
improvement of sparse direct solver for subdomain problem



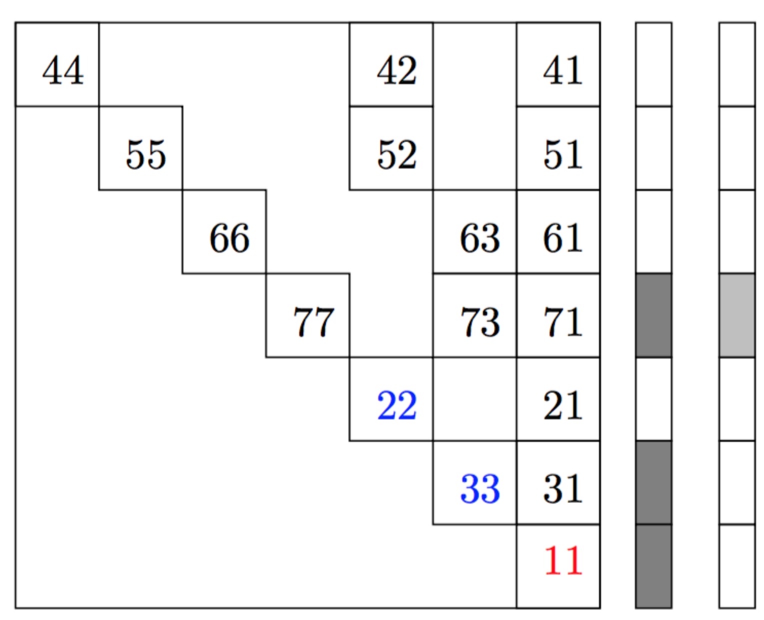
GenEO with enriched coarse space converges faster than standard BDD with coarse space of zero energy mode, but construction of Schur complement consumes more time than solution phase.

local Schur complement matrix is explicitly computed with two forward substitutions of sparse multiple right-hand sides

$$A^{(p)}_{BI} A^{(p)-1}_{II} A^{(p)}_{IB} = (U^{(p)-T}_{II} A^{(p)T}_{BI})^T D^{(p)-1}_{II} (L^{(p)-1}_{II} A^{(p)}_{IB})$$



entries of sparse RHS locate on the interface of the local subdomain.



forward substitution contributes only on half of the bisection tree.