Multilevel domain decomposition at extreme scales

S. Badia, A. Martin, J. Principe

Universitat Politècnica de Catalunya & CIMNE

Jeju, July 7th, 2015

Outline

Outline

Current trends of supercomputing

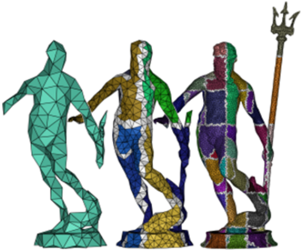
- Transition from **today's 10 Petaflop/s** supercomputers (SCs)
- ... to exascale systems w/ **1 Exaflop/s expected in 2020**
- $\times$ 100 performance based on concurrency (not higher freq)
- Future: Multi-Million-*core* (in broad sense) SCs

Current trends of supercomputing

- Transition from **today's 10 Petaflop/s** supercomputers (SCs)
- ... to exascale systems w/ **1 Exaflop/s expected in 2020**
- $\times$ 100 performance based on concurrency (not higher freq)
- Future: Multi-Million-*core* (in broad sense) SCs

Weakly scalable solvers

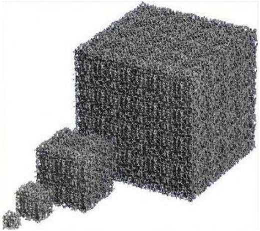
- This talk: One challenge, **weakly scalable algorithms**



Source: Dey et al, 2010

Weak scalability

If we increase X times the number of processors, we can solve an X times larger problem



Source: parFE project

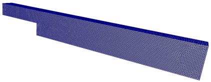
- Key property to face more complex problems / increase accuracy

Scalable linear solvers (AMG)

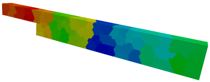
- Most scalable solvers for CSE are **parallel AMG** (Trilinos [Lin, Shadid, Tuminaro, ...], Hypre [Falgout, Yang,...],...)
- Hard to scale up to largest SCs today (one million cores, < 10 PFs)
- Problems: large communication/computation ratios at coarser levels, densification coarser problems,...

Multilevel framework

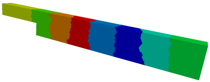
- Propose a highly scalable implementation of Multilevel DD methods (MLBDDC [Mandel et al'08])
- MLDD based on a hierarchy of meshes/functional spaces
- It involves local subdomain problems at all levels (L1, L2, ...)



FE mesh



Subdomains (L1)



Subdomains (L2)

Outline

1. Motivation I: Develop a multilevel framework suitable for extremely scalable implementations

2. Motivation II: Apply the multilevel framework for scalable linear algebra (MLBDDC)

Outline

1. Motivation I: Develop a multilevel framework suitable for extremely scalable implementations

2. Motivation II: Apply the multilevel framework for scalable linear algebra (MLBDDC)

Outline

1. Motivation I: Develop a multilevel framework suitable for extremely scalable implementations
2. Motivation II: Apply the multilevel framework for scalable linear algebra (MLBDDC)

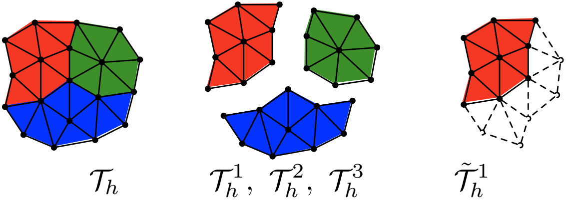
All implementations in FEMPAR (in-house code) to be distributed as open-source SW soon*

* Funded by Proof of Concept Grant 640957 - FEXFEM: On a free open source extreme scale finite element software

Outline

Premilinaries

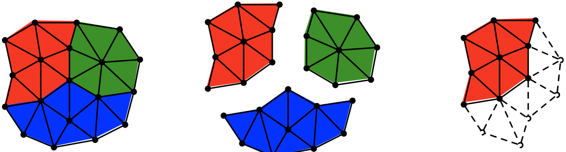
- Element-based (non-overlapping DD) distribution (+ limited ghost info)



- Gluing info based on objects
  - **Object:** Maximum set of interface nodes that belong to the same set of subdomains

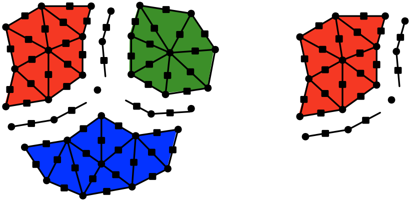
Premilinaries

- Element-based (non-overlapping DD) distribution (+ limited ghost info)



$\mathcal{T}_h$ $\qquad$ $\mathcal{T}_h^1,\ \mathcal{T}_h^2,\ \mathcal{T}_h^3$ $\qquad$ $\tilde{\mathcal{T}}_h^1$

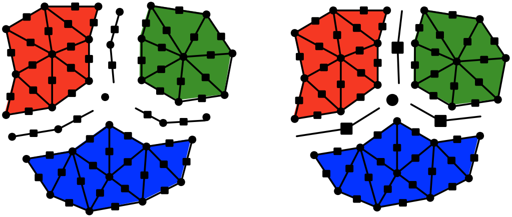
- Gluing info based on objects
  - **Object:** Maximum set of interface nodes that belong to the same set of subdomains



Automatic hierarchical mesh generator

Classification of objects (vef's at the next level) in 3D

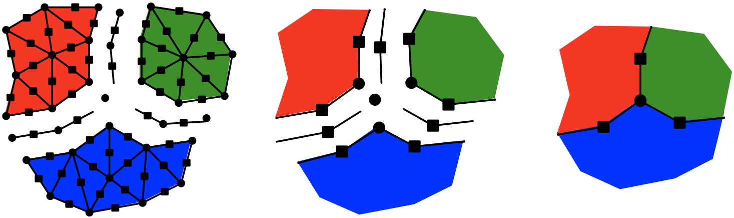
- **Faces:** Objects that belong to 2 subdomains
- **Edges:** Objects that belong to more than 2 subdomains
- **Corners:** Edges and faces with cardinality 1



Coarser triangulation

- Similar to FE triangulation object but wo/ reference element
- Instead, aggregation info

object level 1 = aggregation (vef's level 0)



Coarser FE space

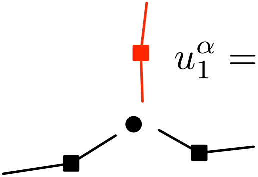
- On top of coarser triangulation, we create a FE-like functional space
- DOFs on geometrical objects at the coarser level (as in FEs)
- Aggregation info for DOFs ($u_1^\alpha = \mathcal{F}_\alpha(u_1)$)

Coarser FE space

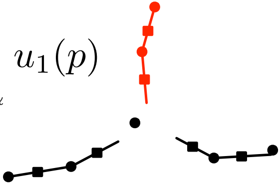
- On top of coarser triangulation, we create a FE-like functional space
- DOFs on geometrical objects at the coarser level (as in FEs)
- Aggregation info for DOFs ($u_1^\alpha = \mathcal{F}_\alpha(u_1)$)

Coarser FE space

- On top of coarser triangulation, we create a FE-like functional space
- DOFs on geometrical objects at the coarser level (as in FEs)
- Aggregation info for DOFs ($u_1^\alpha = \mathcal{F}_\alpha(u_1)$)

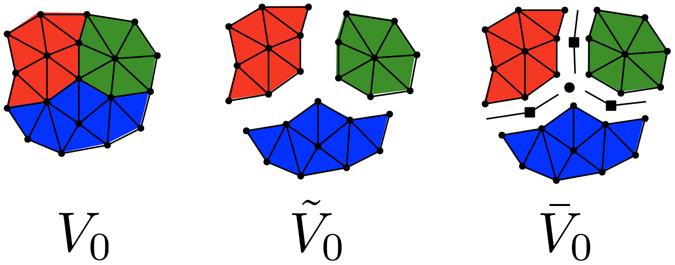


$$u_1^\alpha = \frac{1}{\#(p)} \sum_{p \in \mathcal{E}^\alpha} u_1(p)$$



Hierarchical FE spaces

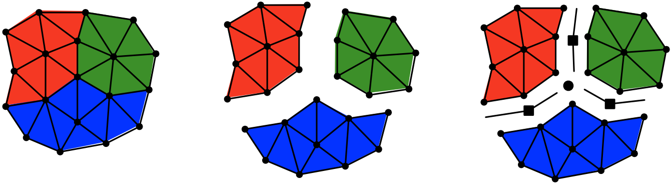
- The under-assembled space $\bar{V}_0 = \{v \in \tilde{V}_0 | \text{ continuous } \mathcal{F}_1(v)\}$
- $\bar{V}_0$ is a multiscale space



- Compute sol'on in V_0 using $\bar{V}_0$ correction as preconditioner (multilevel precond)
- BDDC DD preconditioner is a particular realization of $\bar{V}_0$ (corners/edges/faces)

Hierarchical FE spaces

- The under-assembled space $\bar{V}_0 = \{v \in \tilde{V}_0 | \text{ continuous } \mathcal{F}_1(v)\}$
- $\bar{V}_0$ is a multiscale space

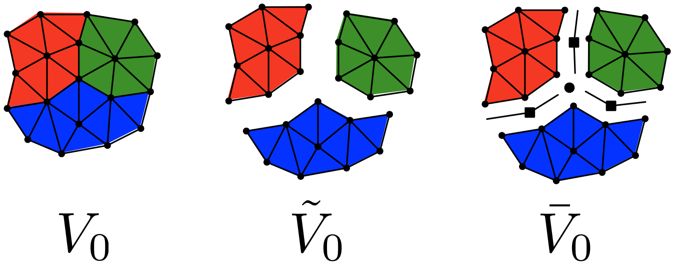


V_0 $\tilde{V}_0$ $\bar{V}_0$

- Compute sol'on in V_0 using $\bar{V}_0$ correction as preconditioner (multilevel precond)
- BDDC DD preconditioner is a particular realization of $\bar{V}_0$ (corners/edges/faces)

Hierarchical FE spaces

- The under-assembled space $\bar{V}_0 = \{v \in \tilde{V}_0 | \text{ continuous } \mathcal{F}_1(v)\}$
- $\bar{V}_0$ is a multiscale space

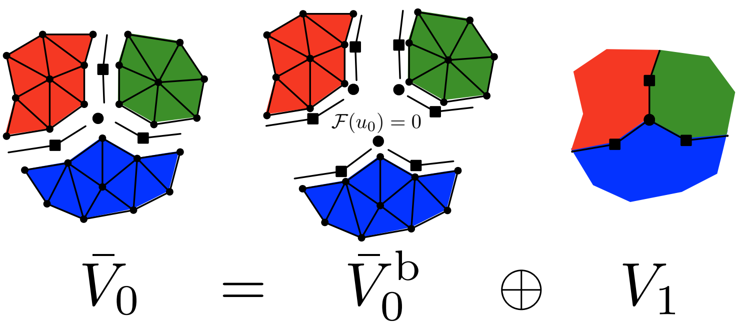


- Compute sol'on in V_0 using $\bar{V}_0$ correction as preconditioner (multilevel precond)
- BDDC DD preconditioner is a particular realization of $\bar{V}_0$ (corners/edges/faces)

Hierarchical FE spaces

The under-assembled space $\bar{V}_0$ can be decomposed as [Dohrmann'03]:

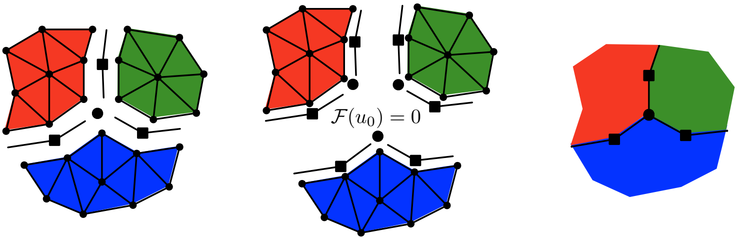
- Its bubble space $\bar{V}_0^{\mathrm{b}} = \{v \in \bar{V}_0 | \mathcal{F}(v) = 0\}$
- The coarser FE space $V_1 = \{v \in \bar{V}_0 | v \perp_{\tilde{\mathcal{A}}} \bar{V}_0^{\mathrm{b}}\}$



Hierarchical FE spaces

The under-assembled space $\bar{V}_0$ can be decomposed as [Dohrmann'03]:

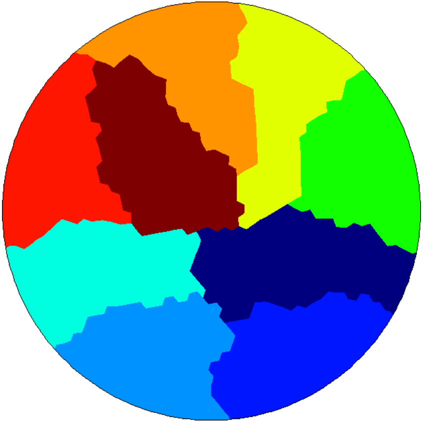
- Its bubble space $\bar{V}_0^{\mathrm{b}} = \{v \in \bar{V}_0 | \mathcal{F}(v) = 0\}$
- The coarser FE space $V_1 = \{v \in \bar{V}_0 | v \perp_{\tilde{\mathcal{A}}} \bar{V}_0^{\mathrm{b}}\}$



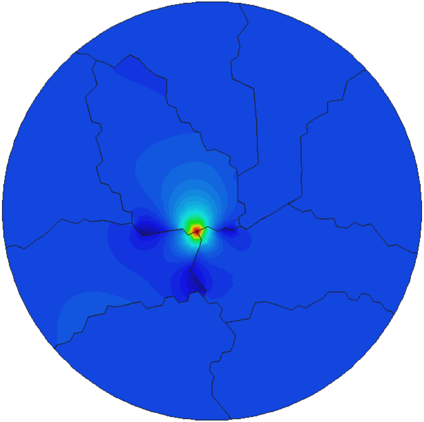
$$\bar{V}_0 \quad = \quad \bar{V}_0^{\mathrm{b}} \quad \oplus \quad V_1$$

Coarse corner function

- Compute via local problems a basis for $V_1 = \{\Phi_1, \ldots, \Phi_{n_c}\}$
- Every Φ is a **coarse shape function related to a coarse DoF**



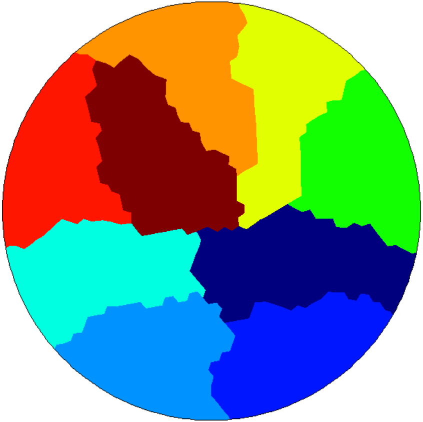
Circle domain partitioned into 9 subdomains



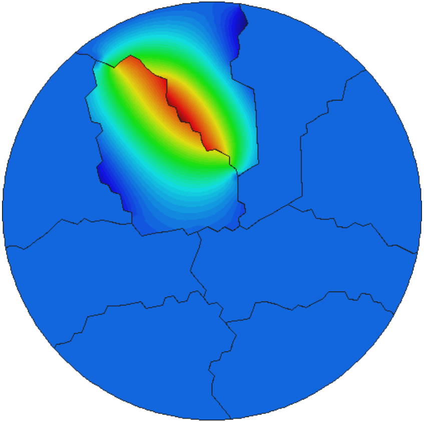
V_1 corner basis function

Coarse edge function

- Compute via local problems a basis for $V_1 = \{\Phi_1, \ldots, \Phi_{n_c}\}$
- Every Φ is a **coarse shape function related to a coarse DoF**



Circle domain partitioned into 9 subdomains



V_1 edge basis function

Multilevel/scale concurrency

The problem in $\bar{V}_0 = V_1 \oplus V_0^{\mathrm{b}}$:

$$\bar{u}_0 \in \bar{V}_0 \ : a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

can be decomposed as $\bar{u}_0 = \bar{u}_0^b + u_1$ (orthogonality $V_1 \perp_{\tilde{\mathcal{A}}} \bar{V}_0^{\mathrm{b}}$)

$$\begin{aligned} u_0^{\mathrm{b}} \in \bar{V}_0^{\mathrm{b}} \ &: a(u_0^{\mathrm{b}}, v_0^{\mathrm{b}}) = (f_0, v_0^{\mathrm{b}}) \ \forall v_0 \in \bar{V}_0^{\mathrm{b}} \\ u_1 \in V_1 \ &: a(u_1, v_1) = (f_1, v_1) \ \forall v_1 \in V_1 \end{aligned}$$

- Bubble component is local to every subdomain (parallel)
- Coarse global problem

Multilevel/scale concurrency

The problem in $\bar{V}_0 = V_1 \oplus V_0^{\mathrm{b}}$:

$$\bar{u}_0 \in \bar{V}_0 \;: a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

can be decomposed as $\bar{u}_0 = \bar{u}_0^b + u_1$ (orthogonality $V_1 \perp_{\tilde{\mathcal{A}}} \bar{V}_0^{\mathrm{b}}$)

$$\begin{aligned} u_0^{\mathrm{b}} \in \bar{V}_0^{\mathrm{b}} &\;: a(u_0^{\mathrm{b}}, v_0^{\mathrm{b}}) = (f_0, v_0^{\mathrm{b}}) \;\forall v_0 \in \bar{V}_0^{\mathrm{b}} \\ u_1 \in V_1 &\;: a(u_1, v_1) = (f_1, v_1) \;\forall v_1 \in V_1 \end{aligned}$$

- Bubble component is local to every subdomain (parallel)
- Coarse global problem

Multilevel/scale concurrency

The problem in $\bar{V}_0 = V_1 \oplus V_0^{\mathrm{b}}$:

$$\bar{u}_0 \in \bar{V}_0 \ : a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

can be decomposed as $\bar{u}_0 = \bar{u}_0^b + u_1$ (orthogonality $V_1 \perp_{\tilde{\mathcal{A}}} \bar{V}_0^{\mathrm{b}}$)

$$\begin{aligned} u_0^{\mathrm{b}} \in \bar{V}_0^{\mathrm{b}} \ &: a(u_0^{\mathrm{b}}, v_0^{\mathrm{b}}) = (f_0, v_0^{\mathrm{b}}) \ \forall v_0 \in \bar{V}_0^{\mathrm{b}} \\ u_1 \in V_1 \ &: a(u_1, v_1) = (f_1, v_1) \ \forall v_1 \in V_1 \end{aligned}$$

- Bubble component is local to every subdomain (parallel)
- Coarse global problem

Multilevel/scale concurrency

The problem in $\bar{V}_0 = V_1 \oplus V_0^{\mathrm{b}}$:

$$\bar{u}_0 \in \bar{V}_0 \ : a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

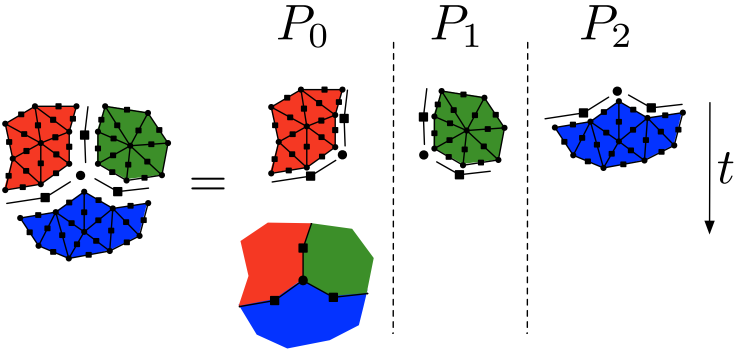
can be decomposed as $\bar{u}_0 = \bar{u}_0^b + u_1$ (orthogonality $V_1 \perp_{\tilde{\mathcal{A}}} \bar{V}_0^{\mathrm{b}}$)

$$\begin{aligned} u_0^{\mathrm{b}} \in \bar{V}_0^{\mathrm{b}} \ &: a(u_0^{\mathrm{b}}, v_0^{\mathrm{b}}) = (f_0, v_0^{\mathrm{b}}) \ \forall v_0 \in \bar{V}_0^{\mathrm{b}} \\ u_1 \in V_1 \ &: a(u_1, v_1) = (f_1, v_1) \ \forall v_1 \in V_1 \end{aligned}$$

- Bubble component is local to every subdomain (parallel)
- Coarse global problem

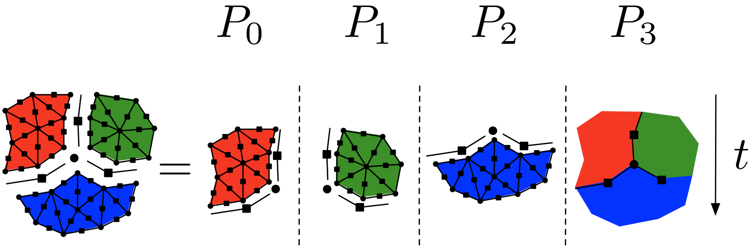
Multilevel concurrency is BASIC for extreme scalability implementations

Multilevel concurrency



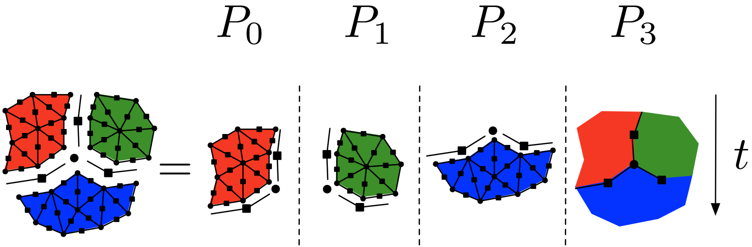
- L1 duties are fully parallel
- L2 duties destroy scalability because
  - # L1 proc's $\sim \times$ 1000 # L2 proc's
  - L2 problem size increases w/ number of proc's

Multilevel concurrency



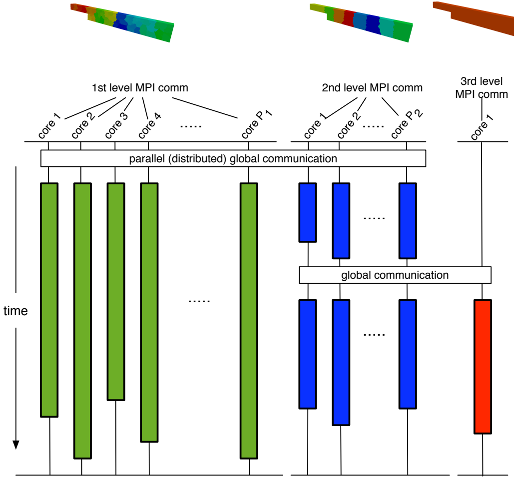
- Every processor has one level/scale duties
- Idling dramatically reduced (energy-aware solvers)
- Overlapped communications / computations among levels

Multilevel concurrency



Inter-level overlapped bulk asynchronous (MPMD) implementation in FEMPAR

Multilevel extension straightforward (starting the alg'thm with V_1 and level-1 mesh)



FEMPAR implementation

Multilevel extension straightforward
(starting the alg'thm with V_1 and level-1 mesh)

Extremely scalable implementation in FEMPAR:

- Recursive (extensible to arbitrary # of levels)
- Inter-level overlapped (bulk asynchronous)

Outline

BDDC preconditioning

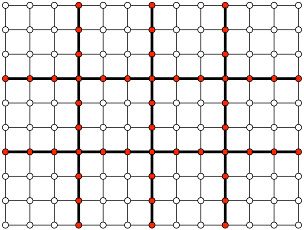
BDDC preconditioner [Dohrmann'03, ...]

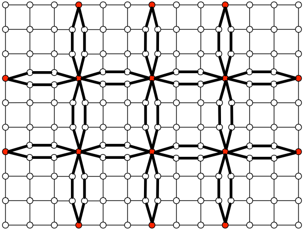
- Replace V_0 by $\bar{V}_0$ (reduced continuity)
- Define the injection $I : \bar{V}_0 \longrightarrow V_0$ (weight, comm and add)
- Find $\bar{u}_0 \in \bar{V}_0$ such that:

$$\bar{u}_0 \in \bar{V}_0 \ : a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

and obtain $u = M_{BDDC} r = \mathcal{E} I \bar{u}_0$, where $\mathcal{E}$ is the harmonic extension operator (correct in the interior of subdomains)

V_0





$\bar{V}_0$

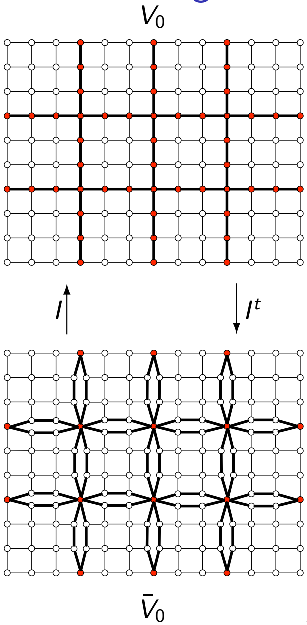
BDDC preconditioning

BDDC preconditioner [Dohrmann'03, ...]

- Replace V_0 by $\bar{V}_0$ (reduced continuity)
- Define the injection $I : \bar{V}_0 \longrightarrow V_0$ (weight, comm and add)
- Find $\bar{u}_0 \in \bar{V}_0$ such that:

$$\bar{u}_0 \in \bar{V}_0 \ : a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

and obtain $u = M_{BDDC} r = \mathcal{E} I \bar{u}_0$, where $\mathcal{E}$ is the harmonic extension operator (correct in the interior of subdomains)



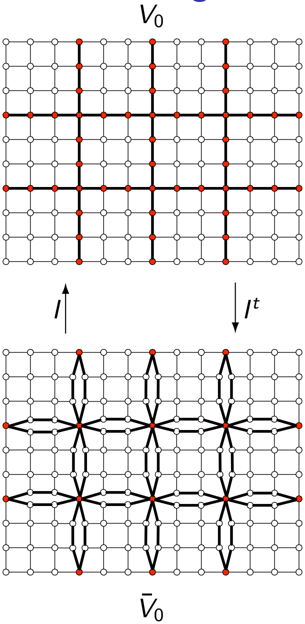
BDDC preconditioning

BDDC preconditioner [Dohrmann'03, ...]

- Replace V_0 by $\bar{V}_0$ (reduced continuity)
- Define the injection $I : \bar{V}_0 \longrightarrow V_0$ (weight, comm and add)
- Find $\bar{u}_0 \in \bar{V}_0$ such that:

$$\bar{u}_0 \in \bar{V}_0 \ : a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

and obtain $u = M_{BDDC} r = \mathcal{E} I \bar{u}_0$, where $\mathcal{E}$ is the harmonic extension operator (correct in the interior of subdomains)



BDDC preconditioning

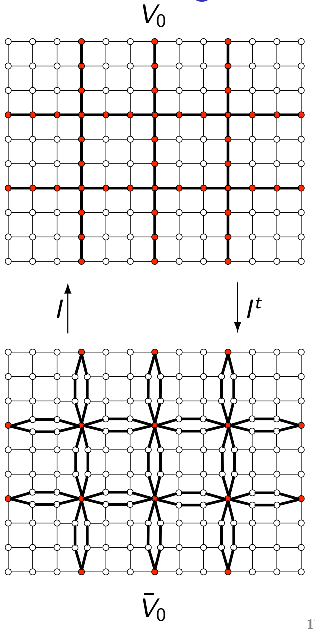
BDDC preconditioner [Dohrmann'03, ...]

- Replace V_0 by $\bar{V}_0$ (reduced continuity)
- Define the injection $I : \bar{V}_0 \longrightarrow V_0$ (weight, comm and add)
- Find $\bar{u}_0 \in \bar{V}_0$ such that:

$$\bar{u}_0 \in \bar{V}_0 \;\; : a(\bar{u}_0, \bar{v}_0) = (f, \bar{v}_0) \quad \forall \bar{v}_0 \in \bar{V}_0$$

and obtain $u = M_{BDDC} r = \mathcal{E} I \bar{u}_0$, where $\mathcal{E}$ is the harmonic extension operator (correct in the interior of subdomains)

The application of $M_{BDDC}(\cdot)$ can be implemented using the multilevel framework above



Overlapping regions

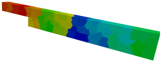
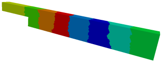
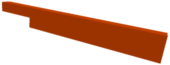
- Classify duties among levels
- 3 overlapping regions (!)

Solve $Ax = b$ w/ BDDC-PCG

Precond'er set-up (M_{BDDC})
call PCG(A,M_{BDDC},b,x_0)

PCG

$r_0 := b - Ax_0$
$z_0 := M_{\mathrm{BDDC}}^{-1} r_0$
$p_0 := z_0$
for $j = 0, \ldots,$ till CONV **do**
$s_{j+1} = Ap_j$
$\ldots$
$z_{j+1} := M_{\mathrm{BDDC}}^{-1} r_{j+1}$
$\ldots$
end for

PCG-BDDC tasks

L_1 MPI tasks	L_2 MPI tasks	L_3 MPI task
Identify local coarse DoFs		
Gather coarse-grid DoFs		
Algorithm 1 ($k \equiv i_{L_1}$) Algorithm 2 ($k \equiv i_{L_1}$)	Build $G_{A_{\mathrm{C}}^{(j_{L_2})}}$ Identify local coarse DoFs	
Compute $\Phi_{i_{L_1}}$	Gather coarse-grid DoFs	
$A_{\mathrm{C}}^{(i_{L_1})} \leftarrow \Phi^t_{i_{L_1}}(-C^T_{i_{L_1}}\Lambda_{i_{L_1}})$	Algorithm 1 ($k \equiv i_{L_2}$)	Build $G_{A_{\mathrm{C}}}$ Re+Sy fact($G_{A_{\mathrm{C}}}$)
Gather $A_{\mathrm{C}}^{(i_{L_1})}$		
Algorithm 3 ($k \equiv i_{L_1}$) Algorithm 4 ($k \equiv i_{L_1}$)	$A_{\mathrm{C}}^{(j_{L_2})} := \text{assemb}(A_{\mathrm{C}}^{(i_{L_1})})$ Algorithm 2 ($k \equiv i_{L_2}$) Compute $\Phi_{i_{L_2}}$ $A_{\mathrm{C}}^{(i_{L_2})} \leftarrow \Phi^t_{i_{L_2}}(-C^T_{i_{L_2}}\Lambda_{i_{L_2}})$	
	Gather $A_{\mathrm{C}}^{(i_{L_2})}$	
	Algorithm 3 ($k \equiv i_{L_2}$)	$A_{\mathrm{C}} := \text{assemb}(A_{\mathrm{C}}^{(i_{L_2})})$ Num fact(A_{C})
Gather $r_{\mathrm{C}}^{(i_{L_1})}$		
Algorithm 5 ($k \equiv i_{L_1}$)	$r_{\mathrm{C}}^{(j_{L_2})} := \text{assemb}(r_{\mathrm{C}}^{(i_{L_1})})$ Algorithm 4 ($k \equiv i_{L_2}$)	
	Gather $r_1^{(i_{L_2})}$	
	Algorithm 5 ($k \equiv i_{L_2}$)	$r_{\mathrm{C}} := \text{assemb}(r_{\mathrm{C}}^{(i_{L_2})})$ Solve $A_{\mathrm{C}} z_{\mathrm{C}} = r_{\mathrm{C}}$
	Scatter z_{C} into $z_{\mathrm{C}}^{(i_{L_2})}$	
	Algorithm 6 ($k \equiv i_{L_2}$)	
Scatter $z_{\mathrm{C}}^{(j_{L_2})}$ into $z_{\mathrm{C}}^{(i_{L_1})}$		
Algorithm 6 ($k \equiv i_{L_1}$)		

Algorithm 1

Re+Sy fact($G_{A_F^{(k)}}$)

Re+Sy fact($G_{A_{II}^{(k)}}$)

Algorithm 2

Num fact($(A_0^{\mathrm{b}})^{(k)}$)

Algorithm 3

Num fact($A_{II}^{(k)}$)

Algorithm 4

$\delta_I^{(k)} \leftarrow (A_{II}^{(k)})^{-1} r_I^{(k)}$

$r_\Gamma^{(k)} \leftarrow r_\Gamma^{(k)} - A_{\Gamma I}^{(k)} \delta_I^{(k)}$

$r^{(k)} \leftarrow I_k^t r$

Algorithm 5

Solve

$$(A_0^{\mathrm{b}})^{(k)} \begin{bmatrix} t \\ s_{\mathrm{F}}^{(k)} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ r^{(k)} \\ 0 \end{bmatrix}$$

Algorithm 6

$s_{\mathrm{C}}^{(k)} \leftarrow \Phi_i z_{\mathrm{C}}^{(k)}$

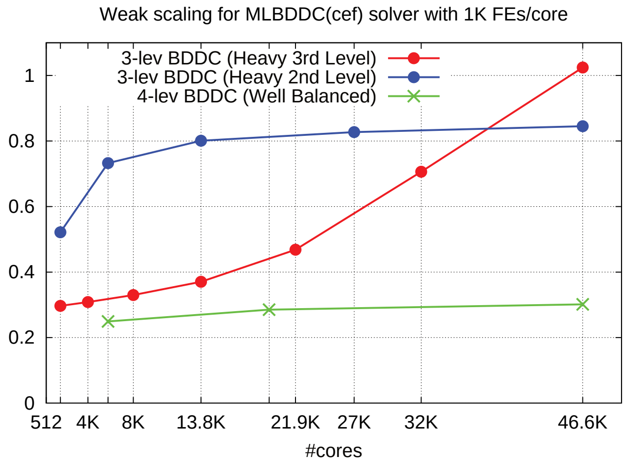
$z^{(k)} \leftarrow I_i(s_{\mathrm{F}}^{(k)} + s_{\mathrm{C}}^{(k)})$

$z_I^{(k)} \leftarrow -(A_{II}^{(k)})^{-1} A_{I\Gamma}^{(k)} z_\Gamma^{(k)}$

$z_I^{(k)} \leftarrow z_I^{(k)} + \delta_I^{(k)}$

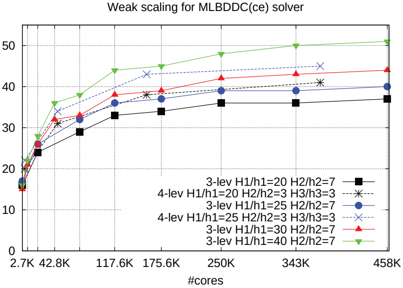
Interlevel load balance

Goal: strike a balance such that blue/red areas are kept below green ones!

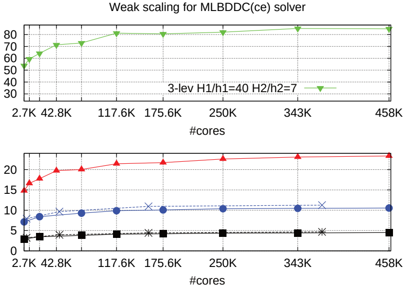


Weak scaling 3-lev BDDC(ce) solver

3D Laplacian problem on IBM BG/Q (JUQUEEN@JSC)
16 MPI tasks/compute node, 1 OpenMP thread/MPI task



#PCG iterations

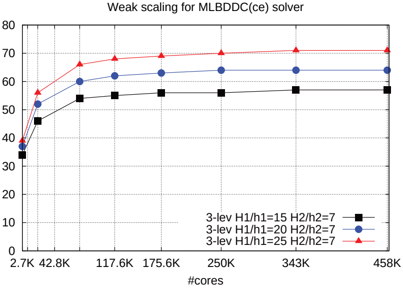


Total time (secs.)

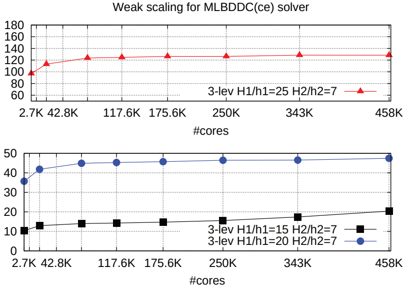
Experiment set-up								
Lev.	# MPI tasks							FEs/core
1st	42.8K	74.1K	117.6K	175.6K	250K	343K	456.5K	$20^3/25^3/30^3/40^3$
2nd	125	216	343	512	729	1000	1331	7^3
3rd	1	1	1	1	1	1	1	n/a

Weak scaling 3-lev BDDC(ce) solver

3D Linear Elasticity problem on IBM BG/Q (JUQUEEN@JSC)
16 MPI tasks/compute node, 1 OpenMP thread/MPI task



#PCG iterations



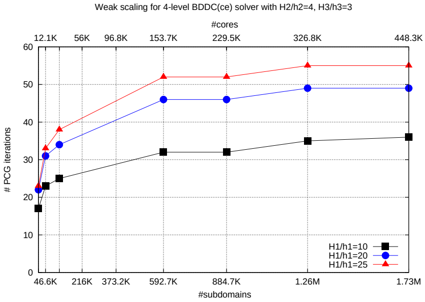
Total time (secs.)

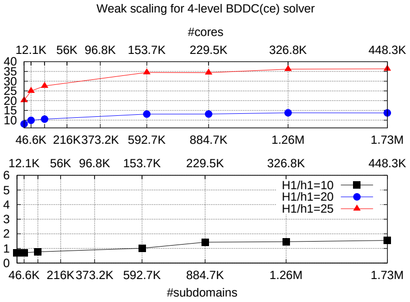
Experiment set-up								
Lev.	# MPI tasks							FEs/core
1st	42.8K	74.1K	117.6K	175.6K	250K	343K	456.5K	$15^3/20^3/25^3$
2nd	125	216	343	512	729	1000	1331	7^3
3rd	1	1	1	1	1	1	1	n/a

Weak scaling 4-lev BDDC(ce)

3D Laplacian problem on IBM BG/Q (JUQUEEN@JSC)

64 MPI tasks/compute node, 1 OpenMP thread/MPI task





Total time (secs.)

Lev.	# MPI tasks							FEs/core
1st	46.7K	110.6K	216K	373.2K	592.7K	884.7K	1.26M	**10^3/20^3/25^3**
2nd	729	1.73K	3.38K	5.83K	9.26K	13.8K	19.7K	4^3
3rd	27	64	125	216	343	512	729	3^3
4th	1	1	1	1	1	1	1	n/a

Outline

Conclusion

- Extremely scalable implementation (MLBDDC)
  - Fully-distributed and communicator-aware
  - Interlevel-overlapped (bulk-asynchronous)
  - Recursive (extensible to arbitrary # levels)

- Remarkable scalability
  - 3D Laplacian and Linear Elasticity PDEs
  - 3/4 levels are sufficient to (efficiently) scale till full JUQUEEN
  - More levels probably needed in the future

Conclusion

- Extremely scalable implementation (MLBDDC)
  - Fully-distributed and communicator-aware
  - Interlevel-overlapped (bulk-asynchronous)
  - Recursive (extensible to arbitrary # levels)

- Remarkable scalability
  - 3D Laplacian and Linear Elasticity PDEs
  - 3/4 levels are sufficient to (efficiently) scale till full JUQUEEN
  - More levels probably needed in the future

Conclusion

- Extremely scalable implementation (MLBDDC)
    - Fully-distributed and communicator-aware
    - Interlevel-overlapped (bulk-asynchronous)
    - Recursive (extensible to arbitrary # levels)

- Remarkable scalability
    - 3D Laplacian and Linear Elasticity PDEs
    - 3/4 levels are sufficient to (efficiently) scale till full JUQUEEN
    - More levels probably needed in the future

Future work:

- Unstructured mesh weak scalability analyses (technical aspects, mesh generation)
- Include adaptive selection of coarse DOFs (not so important in hydrodynamics)

Farewell

Thank you!

SB, A. F. Martín and J. Principe. Multilevel Balancing Domain Decomposition at Extreme Scales. Submitted, 2015.

Work funded by the European Research Council under:

- Starting Grant 258443 - COMFUS: Computational Methods for Fusion Technology
- Proof of Concept Grant 640957 - FEXFEM: On a free open source extreme scale finite element software