ExaDG — Generic algorithms for exa-scale high order discontinuous Galerkin methods

G. Kanschat, K. Kormann, M. Kronbichler, W. Wall

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

SPPEXA Annual Plenary Meeting 2018

IWR

Overall goals of ExaDG

Development of generic exa-scale discontinuous Galerkin algorithms: From efficient data re-use through tensor-aware multigrid solvers and preconditioners to challenging benchmark problems from engineering

1. Message passing geometric multigrid on adaptively refined meshes
2. Node level performance
   - High-thoughput local operator application
   - Fast, parallelizable smoothers
3. Application to incompressible flows

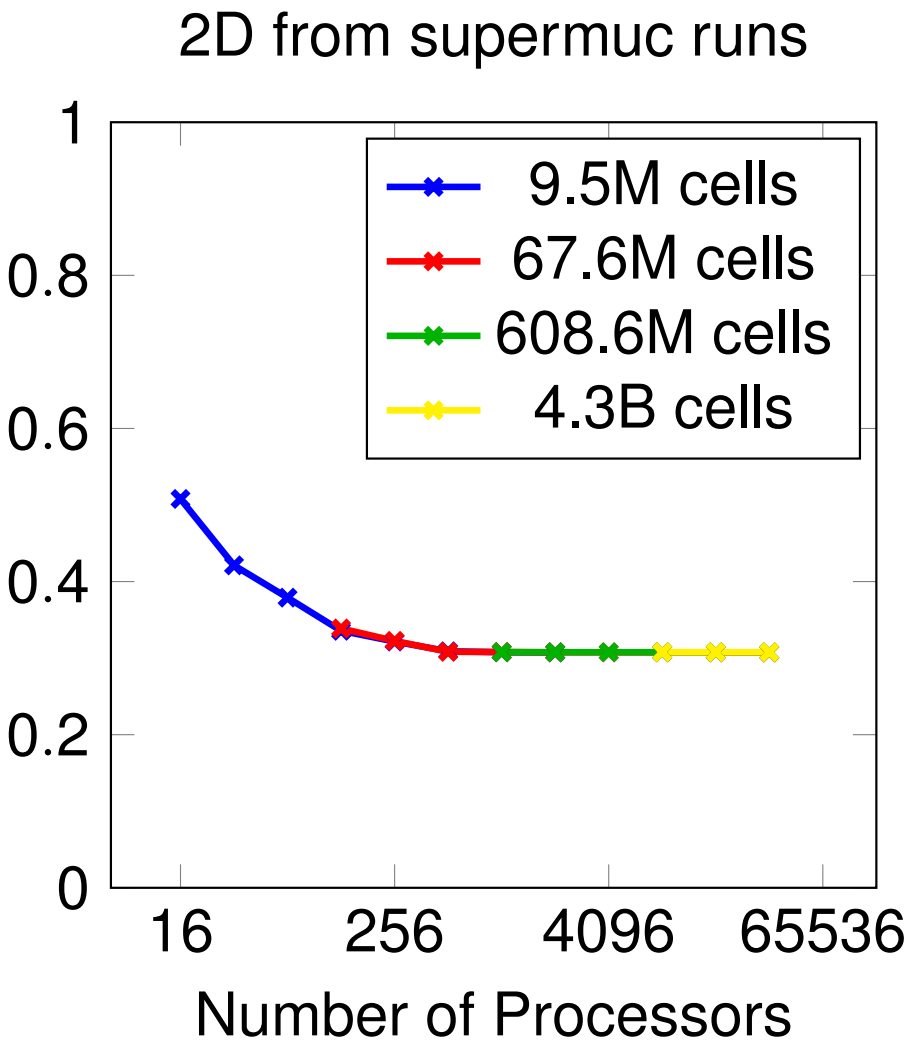
IWR

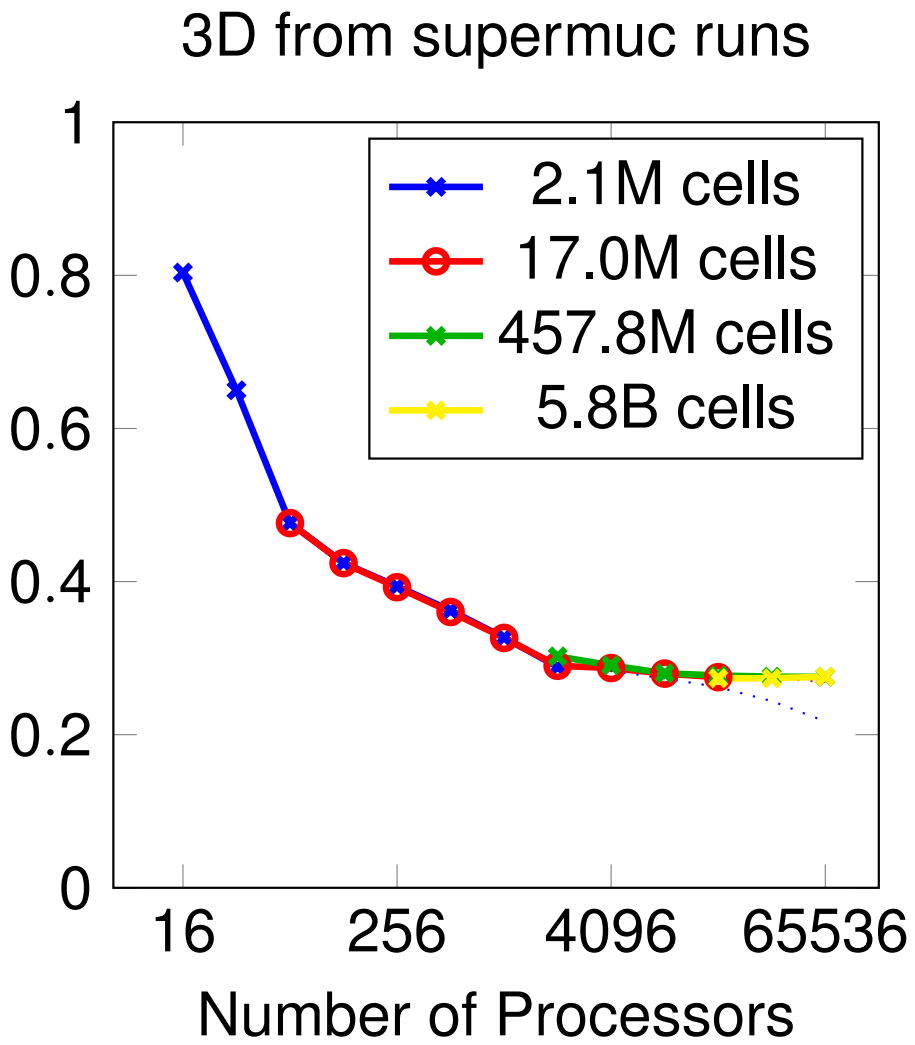
1. **Parallel geometric multigrid**
2. Node level performance
3. Incompressible Flow

Parallel geometric multigrid

- Distribute leaf mesh using Z-curve (p4est)
- Distribute coarser cells according to first child
  - Minimize communication for grid transfer
  - Neglect balance on coarser meshes

Partitioning efficiency

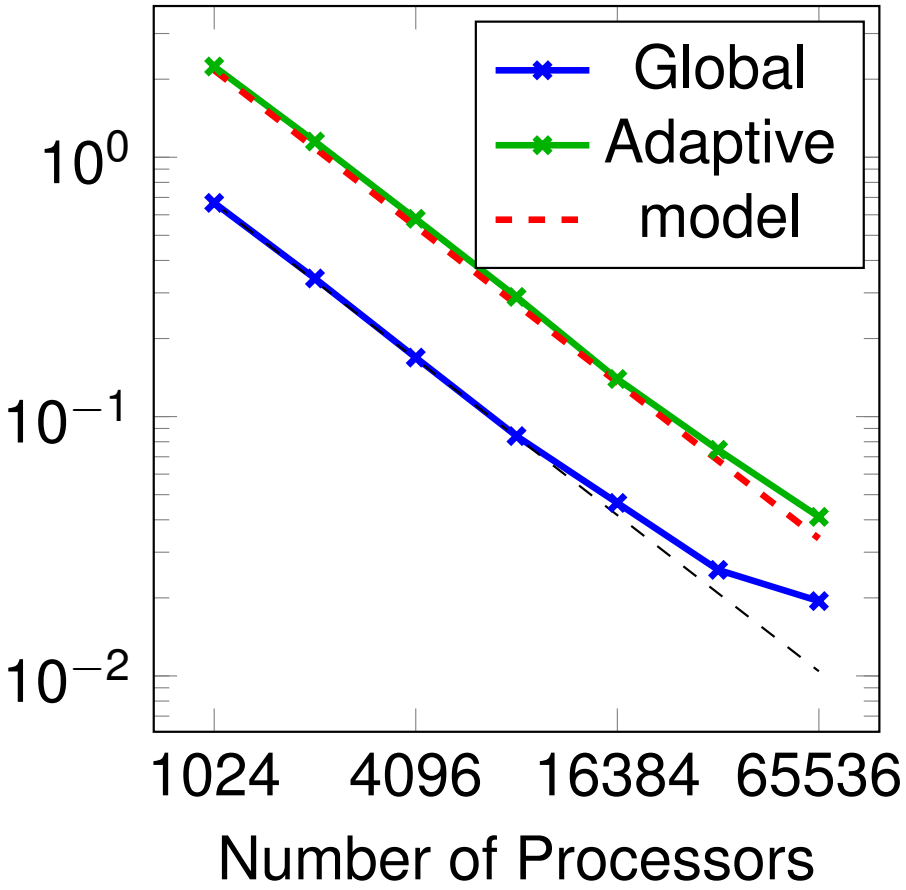




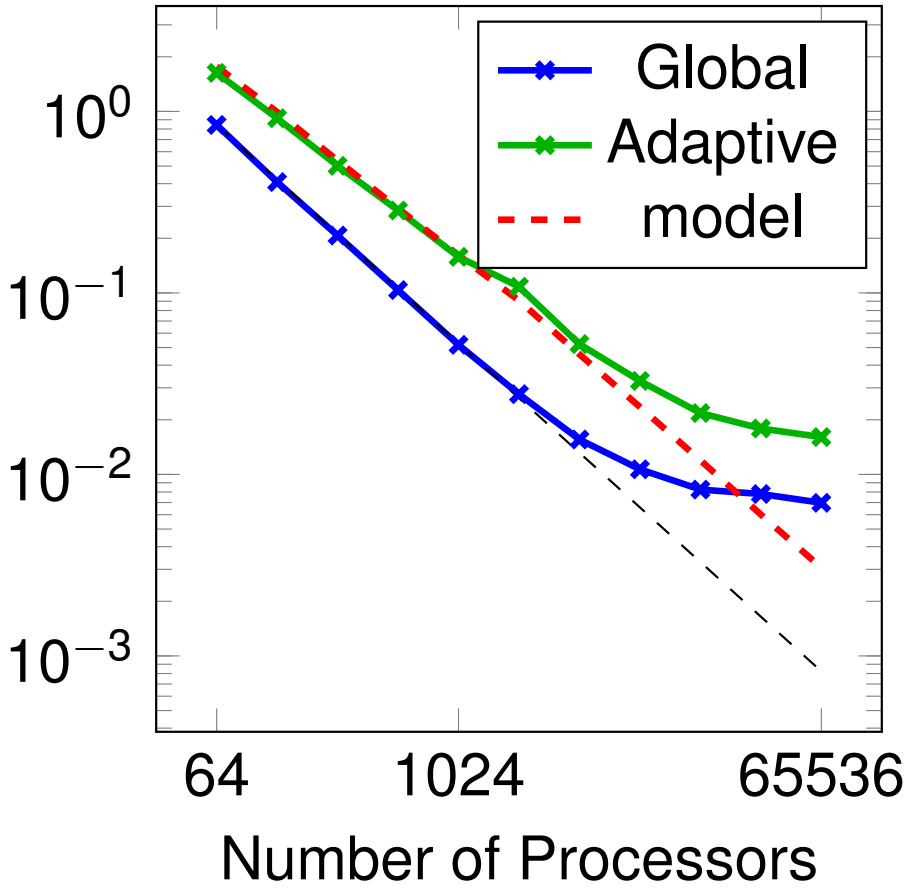
Leaf mesh equidistributed

Strong scaling

≈606.3M cells/2425.4M dofs



≈16.9M cells/137.4M dofs

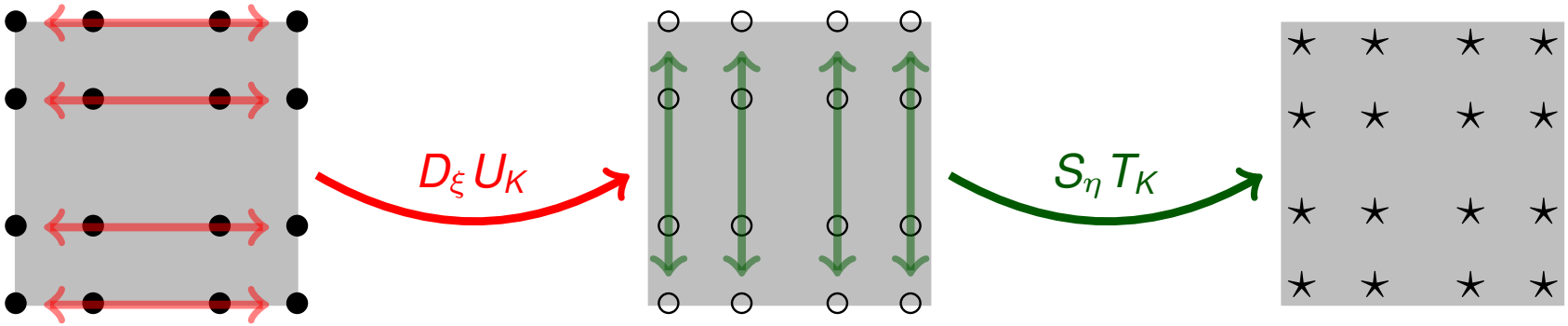


1. Parallel geometric multigrid

2. **Node level performance**
   - Local operator application
   - Smoothers

3. Incompressible Flow

Evaluation of interpolation and integration kernels



Vector values u_K on nodes

$\frac{\partial}{\partial \xi} u^h$ on quadrature points

$$\frac{\partial}{\partial \xi} u^h(\xi_q, \eta_q)\big|_{\text{q\_points}} = (D_\xi \otimes S_\eta)\mathbf{u}_K$$

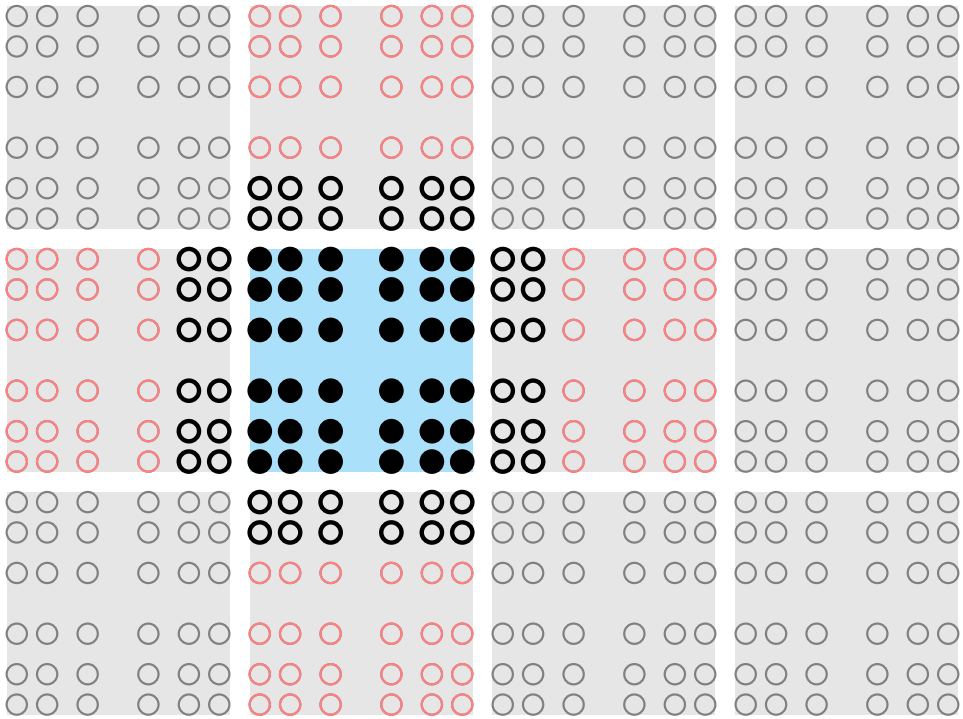
Dense matrix-matrix multiply D_ξ $\mathbf{U}_K$ S_η^T

Evaluation cost: $\mathcal{O}(dk^4)$ per element (degree $k-1$, in 3D)

Naive evaluation: $\mathcal{O}(k^6)$

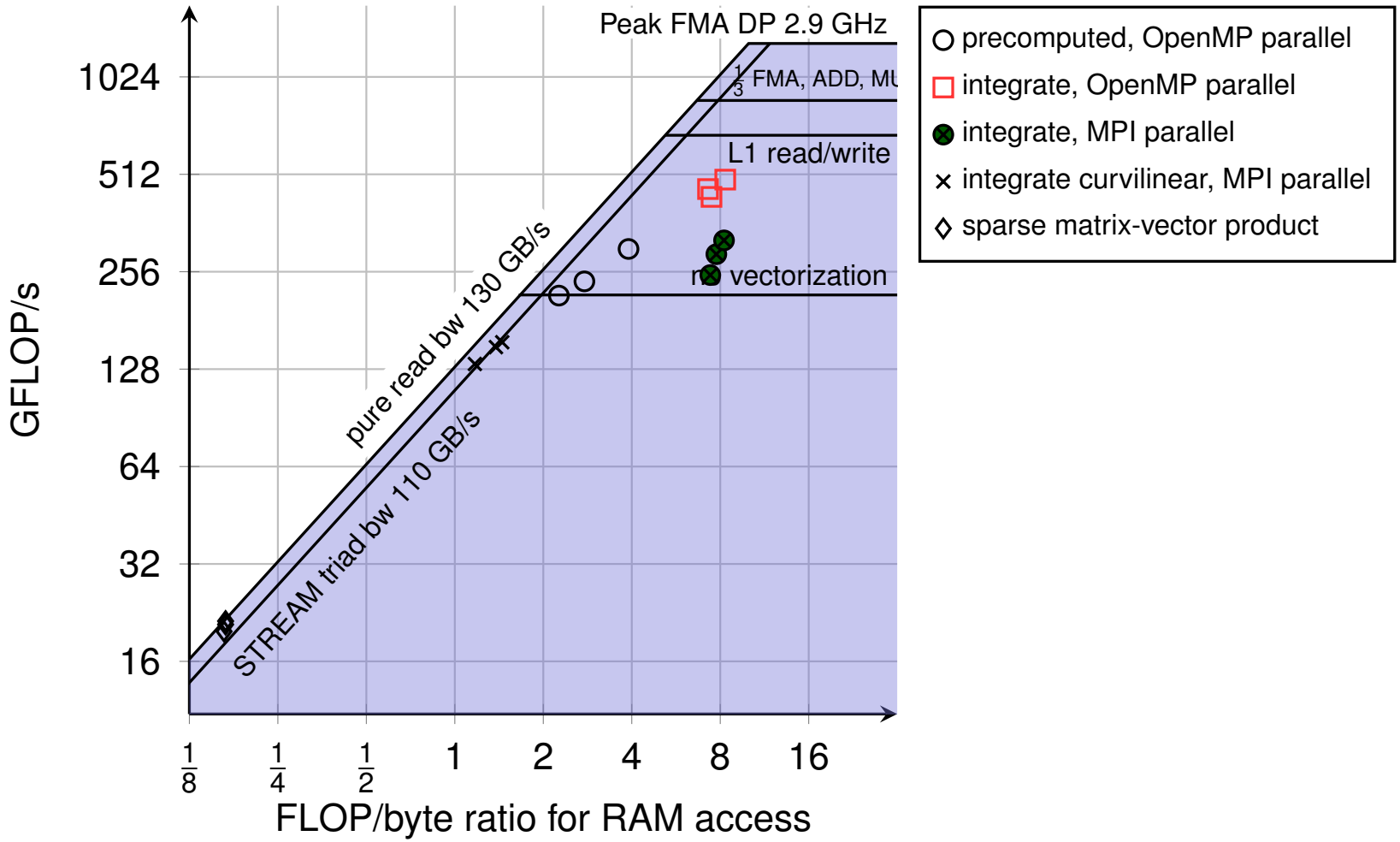
Improved polynomial basis

- Hermite interpolation reduces functions on interfaces
  - Complexity standard: k^d
  - Reduced to: $2(d-1)k^{d-1}$

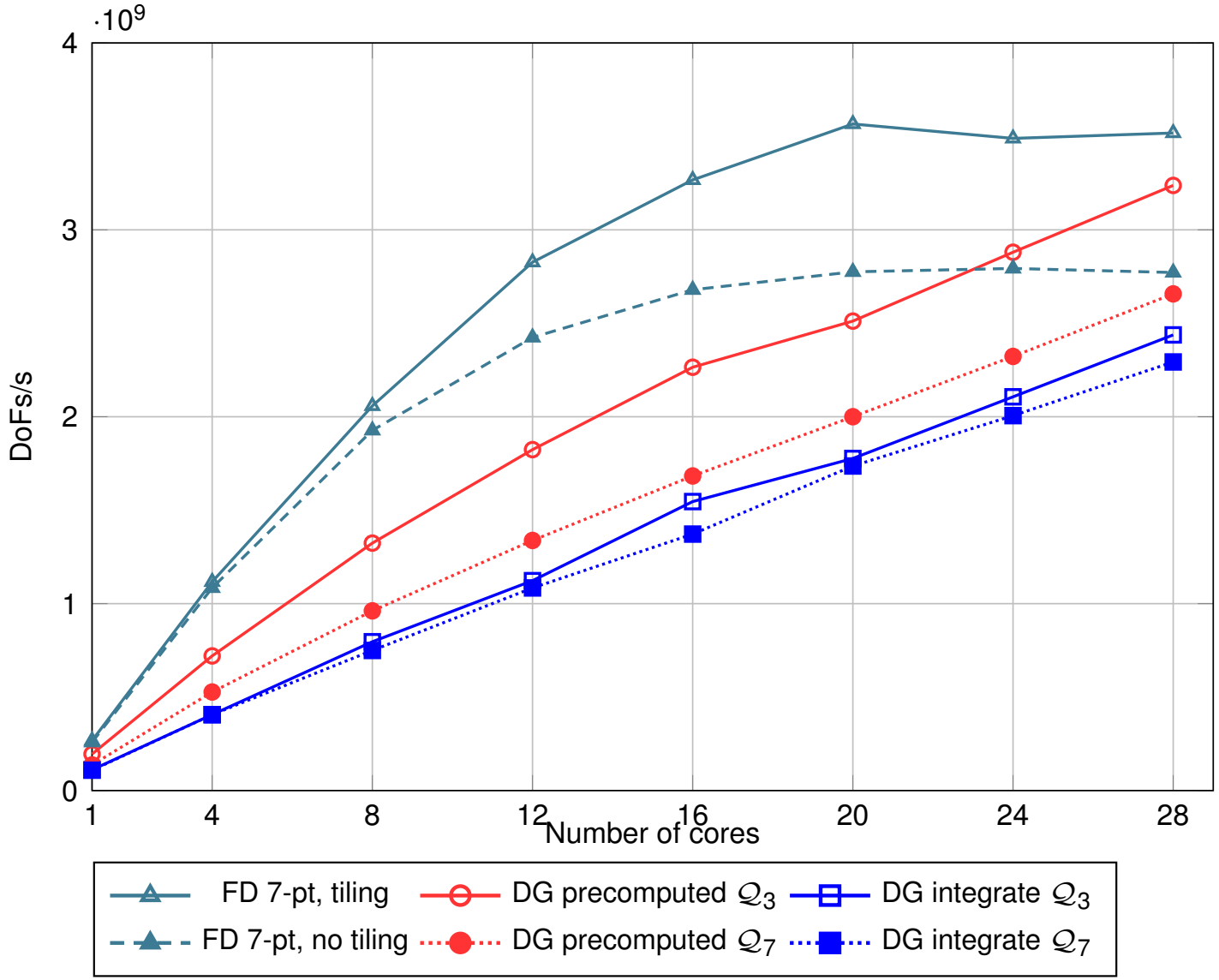


Throughput results

- 1.5 Gdofs/sec on Cartesian meshes
- 0.9 Gdofs/sec on general meshes
- Compare 4.1 Gdofs/sec just read/write



Compared to finite differences (Wichmann/Wall)



1. Parallel geometric multigrid

2. **Node level performance**
   - Local operator application
   - Smoothers

3. Incompressible Flow

Cell smoothers

- Robust for high order
- Simple structure
- “minimal”

Vertex-patch smoothers

- Robust for high order
- Works for incompressibility

Cell smoother

Levels	Order				#col
2D	3	4	7	10	
5	7.5	7.8	10.4	11.4	2
6	7.5	7.7	10.3	11.3	2
7	7.5	7.6	10.3	11.2	2
8	7.4	7.5	10.2	10.8	2
9	7.3	7.4	9.9	10.6	2
10	7.2	7.3	9.8	10.5	2
3D	3	4	7	10	
2	8.1	8.4	11.8	13.5	2
3	8.3	8.5	11.9	13.5	2
4	8.4	8.8	11.9	13.3	2
5	8.4	8.9	11.8	13.3	2
6	8.4	8.8	11.7	12.8	2

Vertex-patch smoother

Levels	Order				#col
2D	3	4	7	10	
5	2.0	2.2	2.0	2.0	6
6	2.0	2.2	2.0	2.0	6
7	2.0	2.2	2.0	2.0	6
8	2.0	2.1	2.0	2.0	6
9	2.0	2.1	2.0	2.0	6
10	2.0	2.1	2.0	2.0	6
3D	3	4	7	10	
2	2.0	2.0	2.0	2.0	1
3	2.0	2.1	2.0	2.0	14
4	2.0	2.1	2.0	2.0	16
5	2.0	2.1	2.0	2.0	16
6	2.0	2.1	2.0	2.0	16

- Multiplicative smoother
- Parallelized by coloring (#col)
- Normalized iteration counts for preconditioned GMRES
- relative tolerance 10^{-8}

IWR

Fast Diagonalization Method (FDM)

For separable operators

$$A = L_1 \otimes M_2 + M_1 \otimes L_2$$

$$A^{-1} = Q_1 \otimes Q_2 \left[\Lambda_1 \otimes I_2 + I_1 \otimes \Lambda_2\right]^{-1} Q_1^T \otimes Q_2^T$$

- For cell and block matrices

Complexity

	standard	*tensor-product*
inverting	$\mathcal{O}(k^{3d})$	$\mathcal{O}(dk^3)$
storage	$\mathcal{O}(k^{2d})$	$\mathcal{O}(dk^2)$
vmult	$\mathcal{O}(k^{2d})$	$\mathcal{O}(dk^{d+1})$
		polynomial space

k is dimension of 1D

[1] Lynch, Rice, and Thomas 1964

1 Parallel geometric multigrid

2 Node level performance

3 **Incompressible Flow**

Benchmark Taylor-Green vortex problem