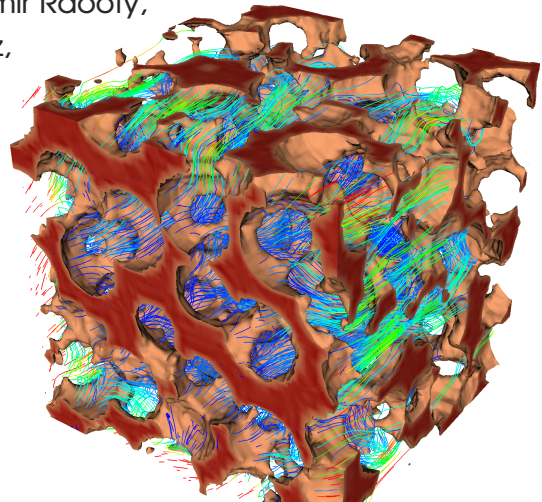
A Parallel Arbitrary-Order Accurate AMR Algorithm for the Advection-Diffusion Equation

Arash Bakhtiari, Dhairya Malhotra, Amir Raoofy,
Miriam Mehl, Hans-Joachim Bungartz,
George Biros



SC16
Salt Lake City, Utah | hpc matters.

Motivating Applications

Porous media flow

Complex fluids

Features

- Unconditionally stable Semi-Lagrangian
- Volume integral formulation
- Arbitrary high-order in space
- Dynamic Adaptive Mesh Refinement (AMR)
- Different velocity / concentration octrees
- Distributed/shared memory + vectorization

Highlights

- **1.4 billion unknowns on 16k CPU cores**
  40s per time step
- **High-order vs Low-order:**
  15x fewer unknowns
  40x faster
- **Dynamic AMR vs Uniform:**
  10x faster
- **Novel Partitioning Scheme:**
  up to 20x less communication
  3x faster

Outline

Time Discretization

- Scalar advection-diffusion problem

$$\frac{\partial c(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v}(\boldsymbol{x}, t) \cdot \boldsymbol{\nabla} c(\boldsymbol{x}, t) - \mathcal{D} \Delta c(\boldsymbol{x}, t) = 0$$

- 2nd order implicit in time

$$\frac{\frac{3}{2} c(\boldsymbol{x}, t_{k+1}) - 2c(\boldsymbol{X}_k, t_k) + \frac{1}{2} c(\boldsymbol{X}_{k-1}, t_{k-1})}{\delta t} - \mathcal{D} \Delta c(\boldsymbol{x}, t_{k+1}) = 0$$

Semi-Lagrangian TODO

- Solving the characteristics backward in time (second-order Runge-Kutta)
- Interpolation at the departure points

Volume Integral Formulation

$$\left(\frac{3}{2} - \delta t D \Delta\right) c_{k+1} = 2c_k - \frac{1}{2} c_{k-1},$$

- Modified Laplace formulation

$$\alpha c - \Delta c = f, \qquad c(x) = \frac{1}{4\pi} \int \frac{e^{-\sqrt{\alpha}|x-y|}}{|x-y|} f(y) dy,$$

- PVFMM library: Fast Multipole Method based volume integral solver (Parallel, Scalable, Optimized)

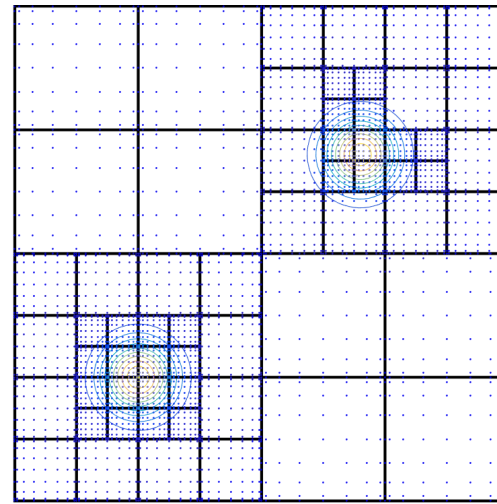
Spatial Discretization

- Partition the domain with an octree
- Approximate the field values with piecewise Chebyshev polynomials of degree q

$$\hat{c}(x, y, z) = \sum_{i+j+k \leq q} \alpha^{\mathcal{B}}_{ijk} T_i(x) T_j(y) T_k(z)$$

- Estimate of error per octant

$$\epsilon = \sum_{i,j,k \geq 0}^{i+j+k=n} |\alpha_{i,j,k}|$$



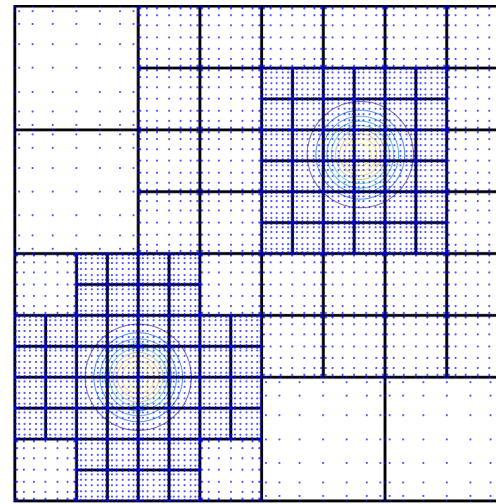
Spatial Discretization

- Partition the domain with an octree
- Approximate the field values with piecewise Chebyshev polynomials of degree q

$$\hat{c}(x,y,z) = \sum_{i+j+k \leq q} \alpha^{\mathcal{B}}_{ijk} T_i(x) T_j(y) T_k(z)$$

- Estimate of error per octant

$$\epsilon = \sum_{i,j,k \geq 0}^{i+j+k=n} |\alpha_{i,j,k}|$$



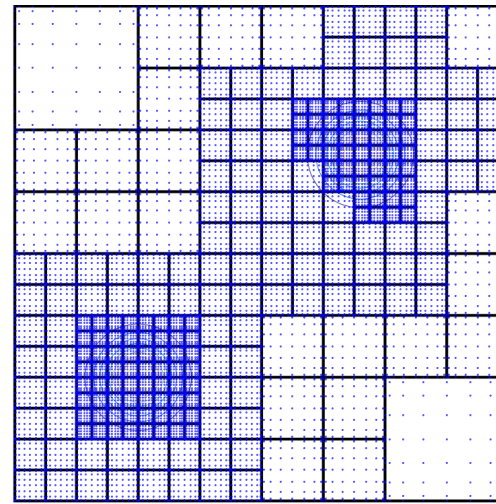
Spatial Discretization

- Partition the domain with an octree
- Approximate the field values with piecewise Chebyshev polynomials of degree q

$$\hat{c}(x,y,z) = \sum_{i+j+k \le q} \alpha^{\mathcal{B}}_{ijk} T_i(x) T_j(y) T_k(z)$$

- Estimate of error per octant

$$\epsilon = \sum_{i,j,k \ge 0}^{i+j+k=n} |\alpha_{i,j,k}|$$



Distributed-Memory Parallelization

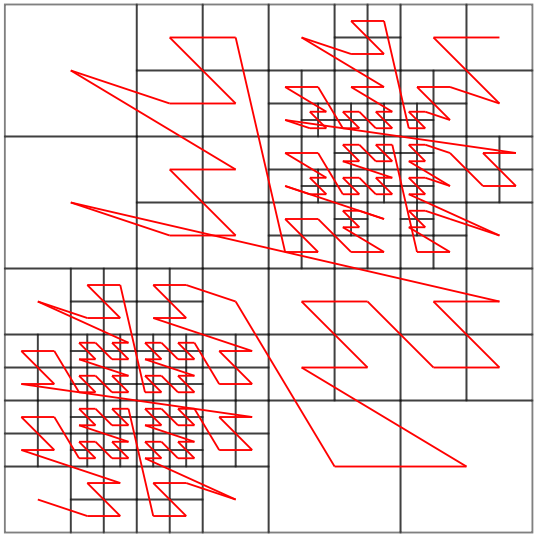
Elliptic solver with PVFMM:

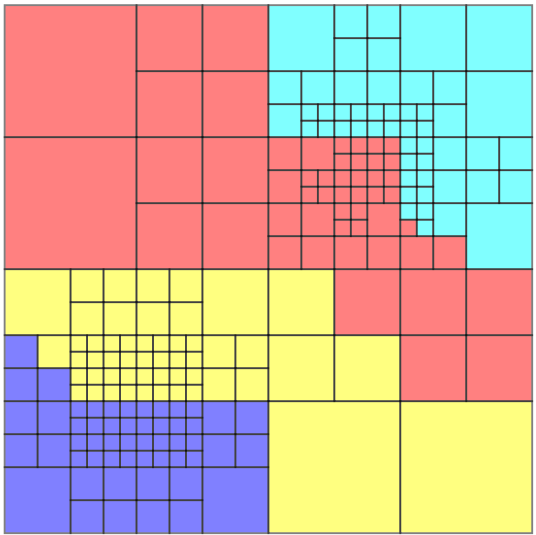
- Point-2-Point and All-to-all communication
- More structured communication

Semi-Lagrangian:

- Near and far communication
- Different trees for velocity and concentration

Space Filling Curves

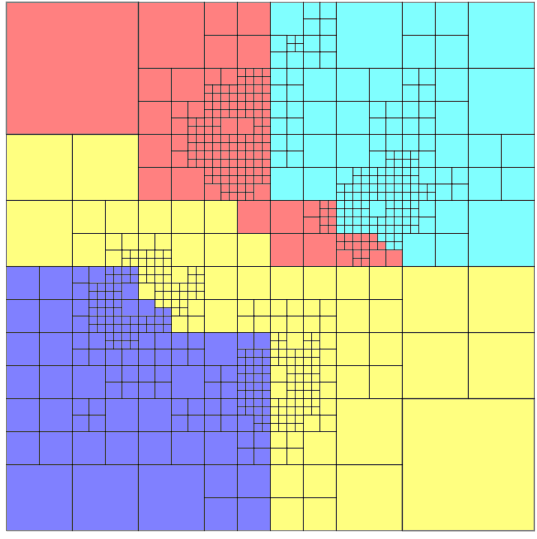




Distributed-Memory Semi-Lagrangian

For each Lagrangian particle:

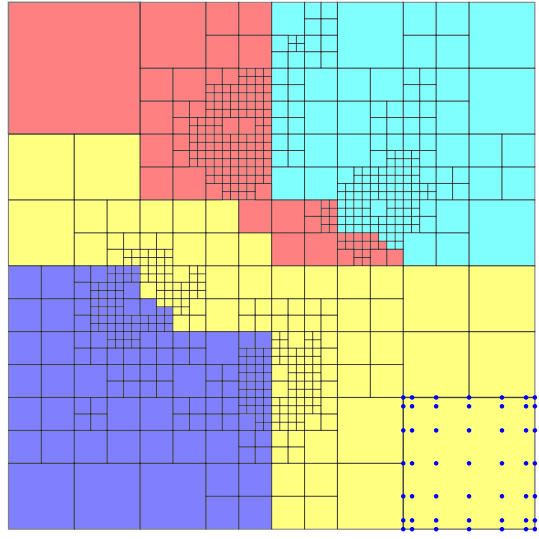
- 2 evaluation of velocity values (RK2)
- 1 evaluation of concentration values
- Tree evaluation requires communication across processes



Distributed-Memory Semi-Lagrangian

For each Lagrangian particle:

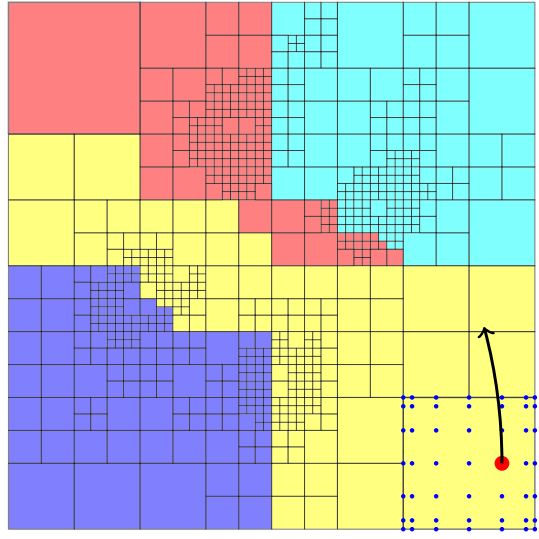
- 2 evaluation of velocity values (RK2)
- 1 evaluation of concentration values
- Tree evaluation requires communication across processes



Distributed-Memory Semi-Lagrangian

For each Lagrangian particle:

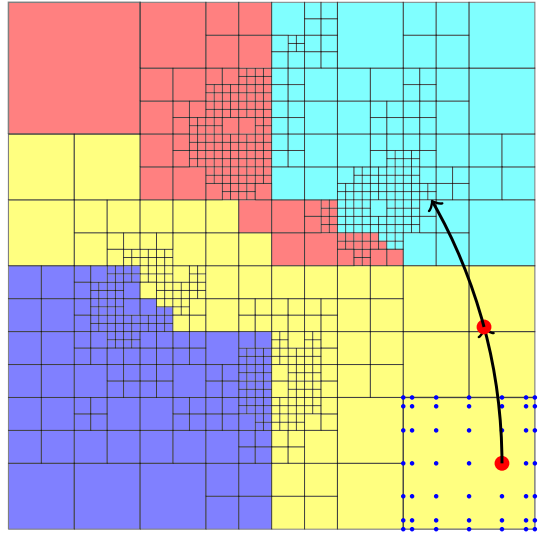
- 2 evaluation of velocity values (RK2)
- 1 evaluation of concentration values
- Tree evaluation requires communication across processes



Distributed-Memory Semi-Lagrangian

For each Lagrangian particle:

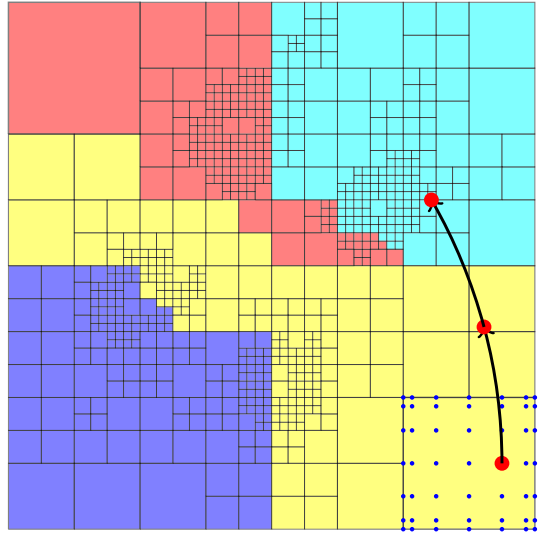
- 2 evaluation of velocity values (RK2)
- 1 evaluation of concentration values
- Tree evaluation requires communication across processes



Distributed-Memory Semi-Lagrangian

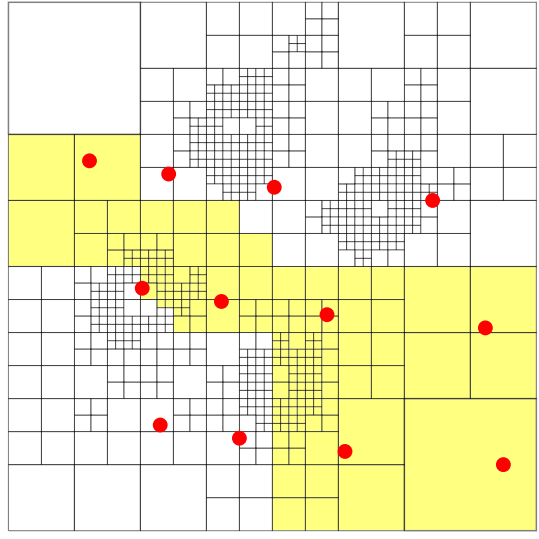
For each Lagrangian particle:

- 2 evaluation of velocity values (RK2)
- 1 evaluation of concentration values
- Tree evaluation requires communication across processes



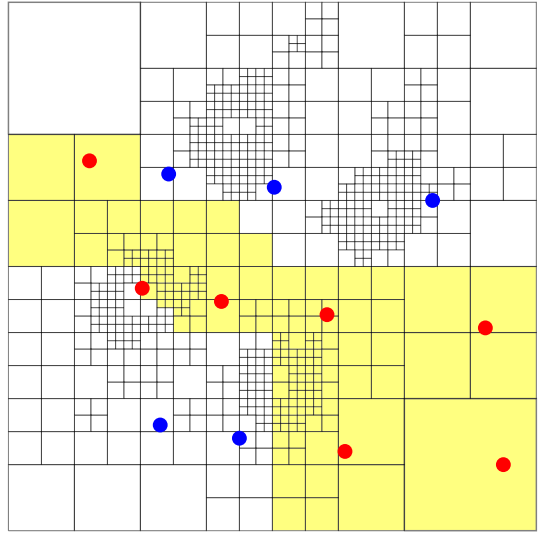
Distributed-Memory Tree Evaluation

- Separate the local and remote points
- Communicate remote point's positions
- Evaluate Chebyshev polynomial
- Communicate remote point's values



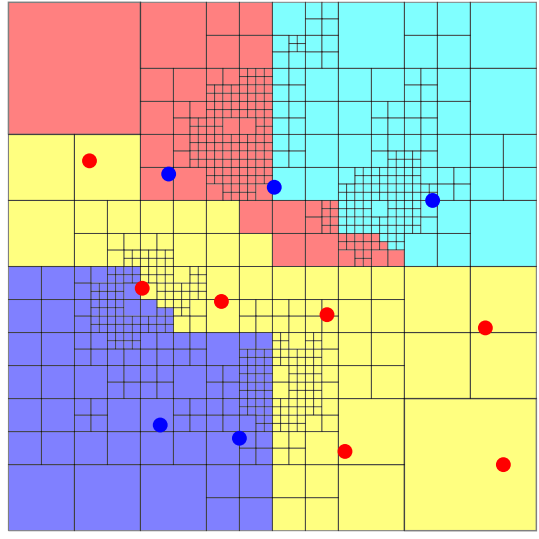
Distributed-Memory Tree Evaluation

- Separate the local and remote points
- Communicate remote point's positions
- Evaluate Chebyshev polynomial
- Communicate remote point's values



Distributed-Memory Tree Evaluation

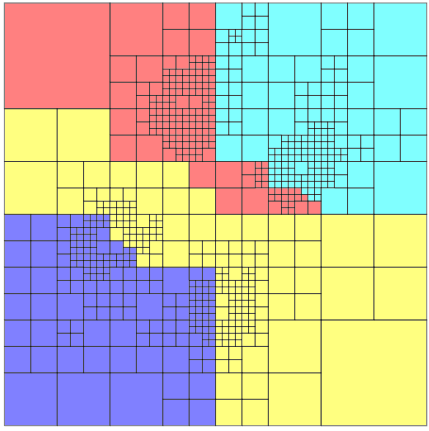
- Separate the local and remote points
- Communicate remote point's positions
- Evaluate Chebyshev polynomial
- Communicate remote point's values



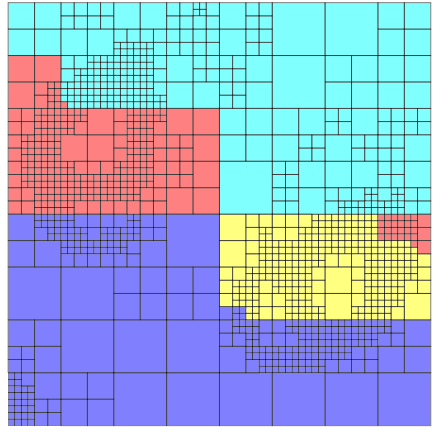
Distributed-Memory Semi-Lagrangian

- Two separate octrees for velocity and concentration fields

Concentration Tree



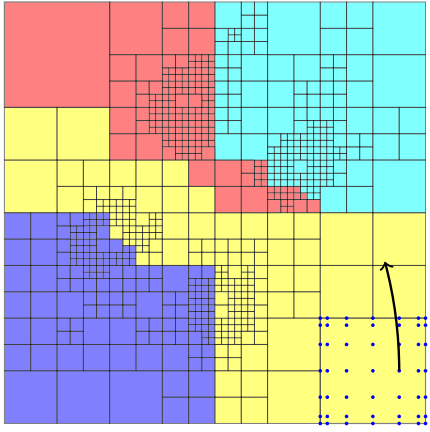
Velocity Tree



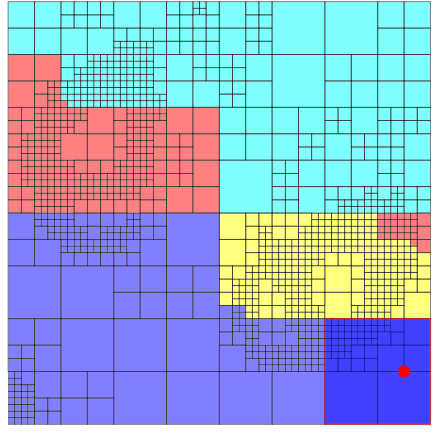
Distributed-Memory Semi-Lagrangian

- Two separate octrees for velocity and concentration fields

Concentration Tree



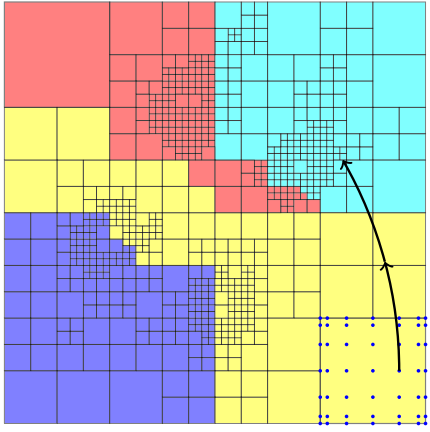
Velocity Tree



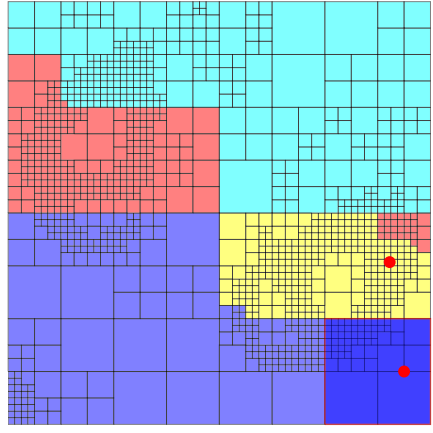
Distributed-Memory Semi-Lagrangian

- Two separate octrees for velocity and concentration fields

Concentration Tree



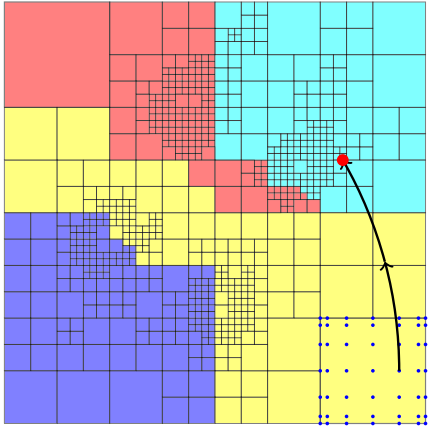
Velocity Tree



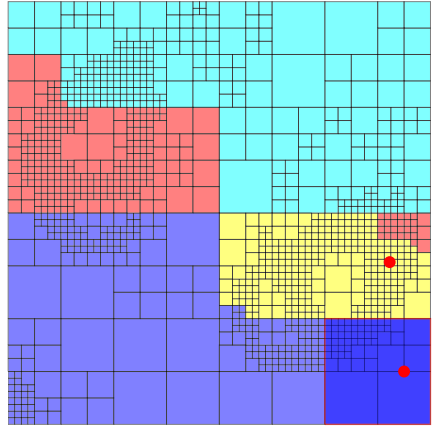
Distributed-Memory Semi-Lagrangian

- Two separate octrees for velocity and concentration fields

Concentration Tree

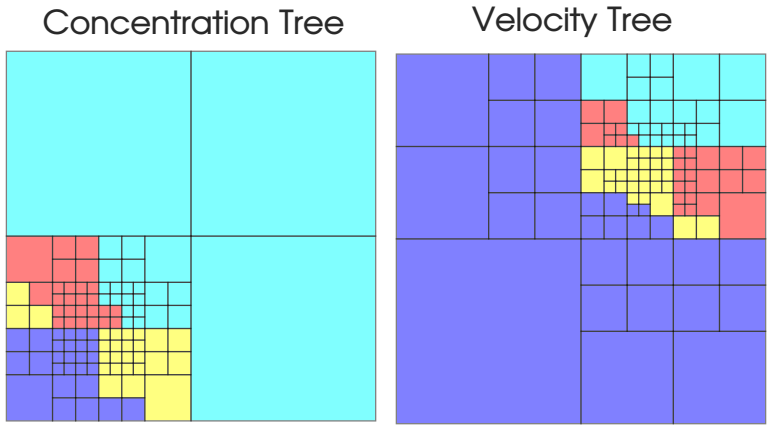


Velocity Tree



Octree Partitioning

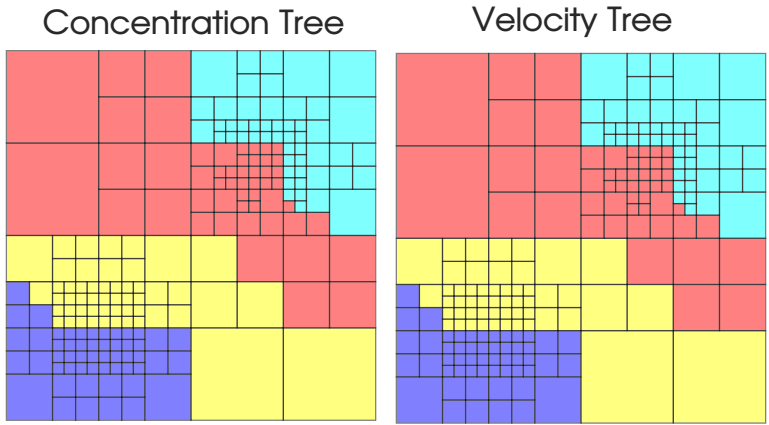
- **Separate Trees:**



- High communication overhead
- Exceed memory resources due to severe load imbalance

Octree Partitioning

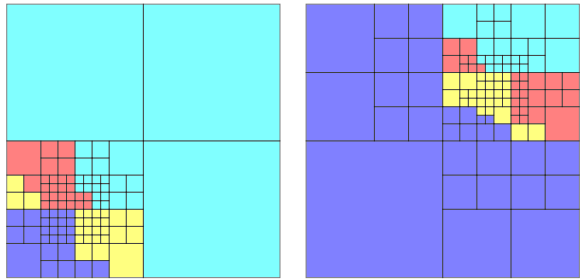
- **Complete Merge:** Both trees have the same refinement and partitioning



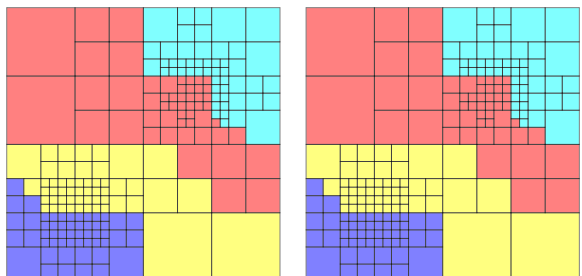
- Increases the computational cost

Octree Partitioning

Separate trees



Complete Merge

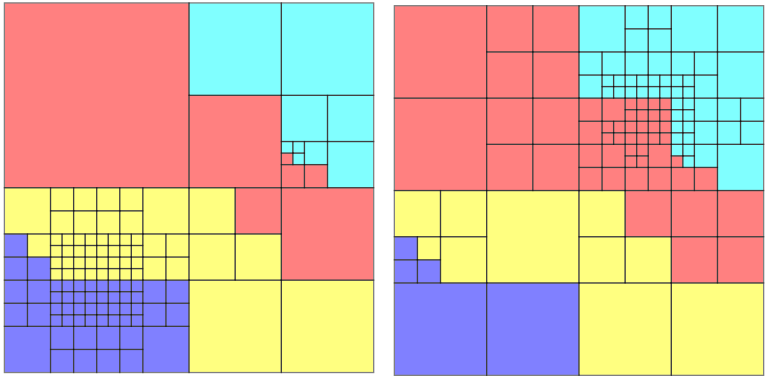


Octree Partitioning

- **Semi-Merge:** Merge partitions but not trees
- Refine the trees only at the new partition boundary

Concentration Tree

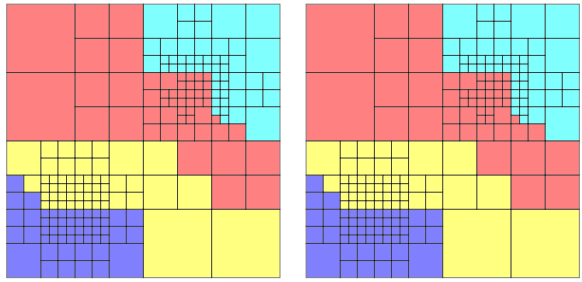
Velocity Tree



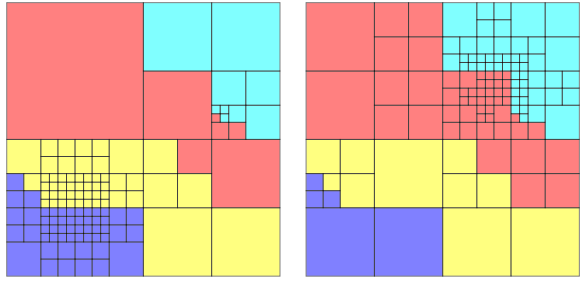
- Significantly fewer octants compared to complete merge

Octree Partitioning

Complete Merge

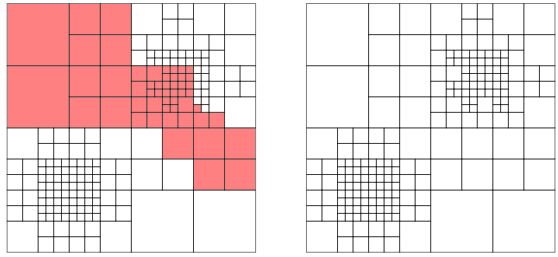


Semi-Merge

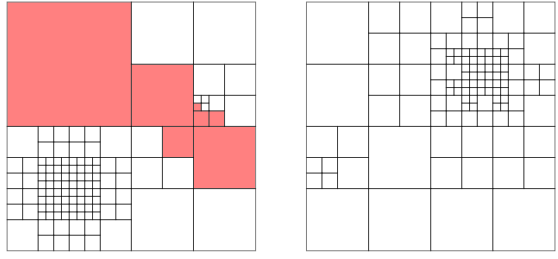


Octree Partitioning

Complete Merge



Semi-Merge

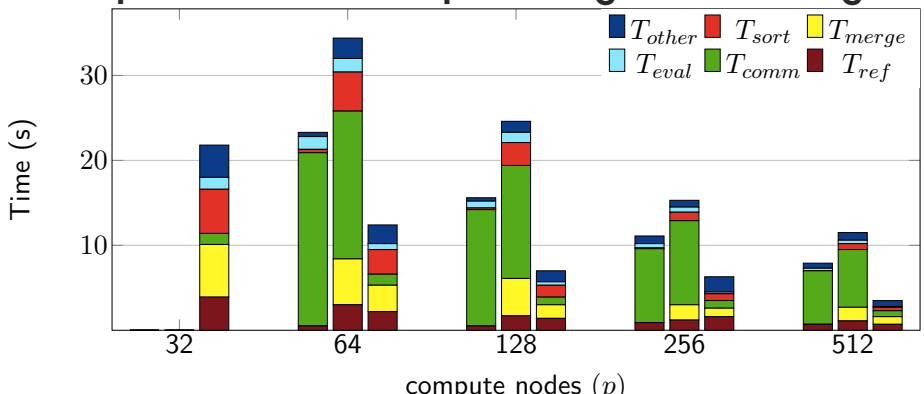


Comparison of Merging Schemes

Compute Nodes: 32 $\longrightarrow$ 512

Unknowns: 40 million

Separate Trees vs Complete Merge vs Semi-Merge:



up to 20x less communication, 3x faster

Convergence

advection equation

		Uniform Tree			Adaptive Tree		
dt	q	L	L^∞	T_{solve}	ϵ_{tree}	L^∞	T_{solve}
1.0E-2	14	3	1.3E-3	38.52	1E-3	6.0E-4	20.49
5.0E-3	14	4	7.9E-6	525.48	1E-5	2.3E-5	133.88
2.5E-3	14	5	2.8E-7	8059.58	1E-7	2.0E-7	809.80

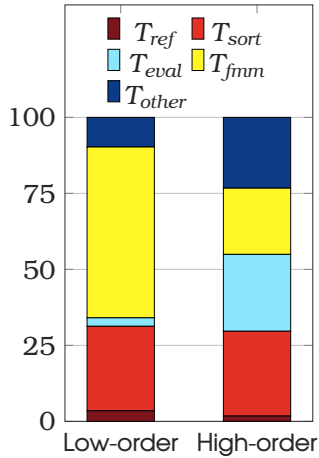
Uniform vs Adaptive: 10x faster

Convergence

advection-diffusion equation

q	δt	N_{dof}	L_T^∞	T_{solve} (GFLOP/s)
4	1E-1	1E+6	9E-3	41.8 (43)
4	6E-3	1E+7	4E-5	12241.0 (35)
14	1E-1	3E+5	9E-3	7.7 (60)
14	6E-3	8E+5	4E-5	310.4 (72)

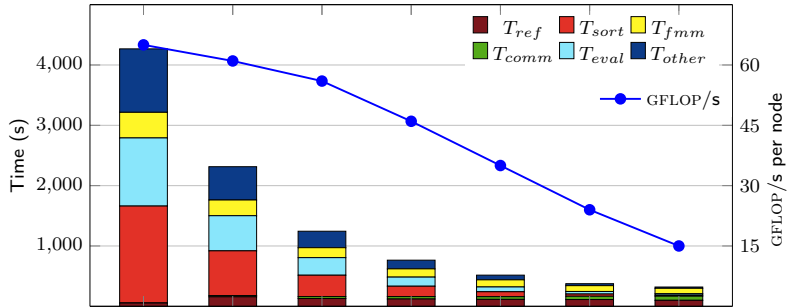
	Eval	FMM
Low-order	40GFLOP/s	80GFLOP/s
High-order	140GFLOP/s	140GFLOP/s



Low-order vs High-order: 15x fewer unknowns, 40x faster

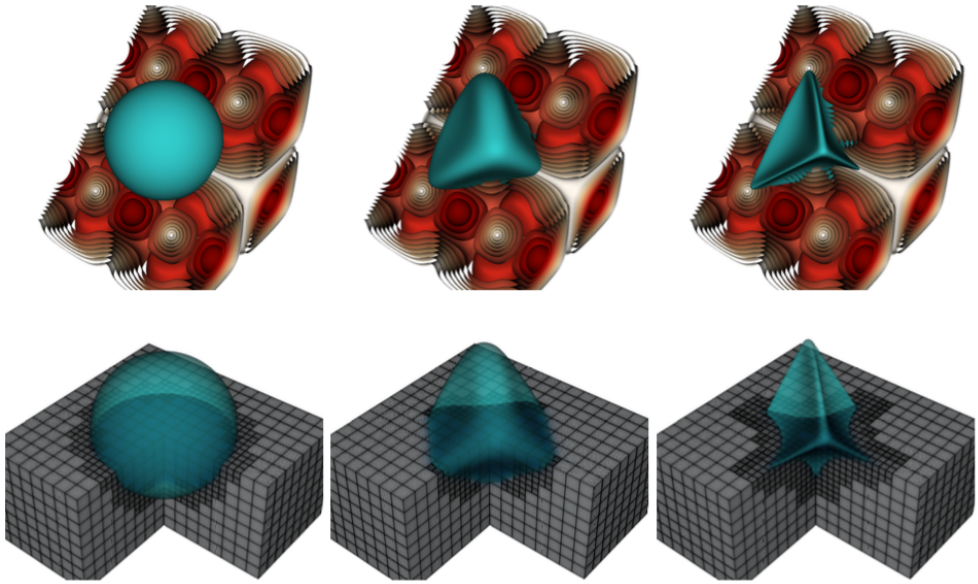
Strong Scaling

- 9.4 million unknowns
- 14^{th} order discretization
- 256 time-steps



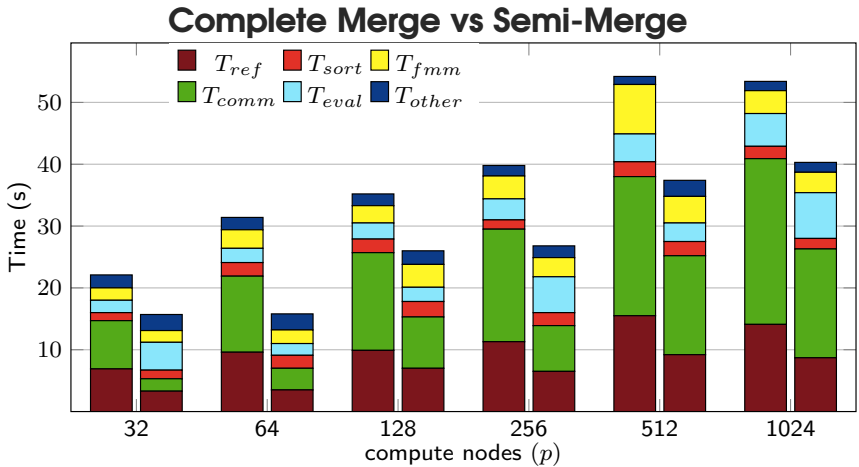
Compute Nodes	p	1	2	4	8	16	32	64
Refinement	T_{ref}	54.1	157.3	128.5	120.9	115.1	112.2	99.4
	remote%	0.0	0.4	1.4	2.5	4.1	7.1	13.4
Semi-Lagrangian	T_{comm}	0.0	20.4	32.2	40.2	44.1	54.5	68.7
	T_{sort}	1607.6	741.8	354.6	171.9	82.4	36.7	16.9
	T_{eval}	1129.0	582.1	291.2	151.0	79.1	42.4	24.3
	GFLOP/s	150.1	145.6	145.5	140.4	133.9	125.0	108.8
FMM	T_{fsetup}	54.4	51.2	49.7	64.4	68.5	55.8	52.0
	T_{fcomp}	380.3	209.2	115.2	72.2	49.5	40.6	37.2
	GFLOP/s	253.0	230.0	209.0	167.0	123.0	75.0	42.0

Weak Scaling



Weak Scaling

Compute Nodes:	32	$\longrightarrow$	1024
Unknowns:	47 million	$\longrightarrow$	1.4 billion



Limitations

- No time-adaptivity
- Second-order in time
- Cubic domains:
  free-space and periodic boundary conditions

Conclusions

Contributions:

- Semi-Lagrangian/FMM based advection-diffusion solver
- Robust load-balancing method for Lagrangian schemes

Features:

- High order in space
- Second order in time
- Unconditionally stable
- Dynamic AMR
- High performance
- Parallel & Scalable

Funding from:

IAS TUM Institute for Advanced Study

DEPARTMENT OF ENERGY · UNITED STATES OF AMERICA

NSF

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH · UNITED STATES AIR FORCE