Terra-Neo - Integrated Co-Design of an Exascale Earth Mantle Modeling Framework

Nils Kohl, D. Thönnes, D. Drzisga (TUM), D. Bartuschat, U. Rüde
Chair for System Simulation, University of Erlangen-Nürnberg
March 21, 2018

joint work with
Markus Huber, Barbara Wohlmuth (TUM)
Simon Bauer, Marcus Mohr, Hans-Peter Bunge (LMU)
Motivation: simulation of earth mantle convection

[Source: Terra-Neo reference implementation]
TerraNeo Project

re-design?

Problems with reference code:
• not extensible to higher order FE
• ill-designed load balancing approach
• hard to maintain

Fast progress through reuse of well-maintained waLBerla (http://walberla.net) core library!
TerraNeo Project

Building Blocks

- hierarchical hybrid grids
- fully distributed data structures
- support for different discretizations
TerraNeo Project

Building Blocks

• hierarchical hybrid grids
• fully distributed data structures
• support for different discretizations
Hierarchical Hybrid Grids
Hierarchical Hybrid Grids
Hierarchical Hybrid Grids

unstructured grid  graph of *primitives*
Hierarchical Hybrid Grids

unstructured grid

graph of \textit{primitives}

edge primitives
Hierarchical Hybrid Grids

- marks ownership
- ghost points
Hierarchical Hybrid Grids

same DoF
TerraNeo Project

Building Blocks

• hierarchical hybrid grids
• fully distributed data structures
• support for different discretizations
From mesh to primitives (2D)

(unstructured)
input mesh
From mesh to primitives (2D)

(input mesh) → (primitive graph)

primitive assembly

(unstructured)
From mesh to primitives (2D)

- (unstructured) input mesh
- Primitive assembly
- Primitive graph
- Load balancing
- Fully distributed domain
Load Balancing 2D

face primitives

edge primitives
Data Handling

Primitive

Types: Vertex, Edge, Face, Cell / Tetrahedron
Data Handling

Primitive

Metadata:
- globally unique ID
- direct neighborhood (IDs)
- geometric information (e.g. vertex coordinates, orientation, …)

Types: Vertex, Edge, Face, Cell / Tetrahedron

Lightweight metadata
Data Handling

Primitive

Metadata:
- globally unique ID
- direct neighborhood (IDs)
- geometric information (e.g. vertex coordinates, orientation, …)

Types: Vertex, Edge, Face, Cell / Tetrahedron

Lightweight metadata

Registered / Allocated Data

Arbitrary data structures

Actual simulation data

P1 FE

P2 FE

Flag Field

P1 FE

P2 FE

Flag Field
Abstraction Data - Topology

Data

*intra*-primitive building blocks

- Serialization (Buffer / File)
- (Simulation) Data
- Calculations

Topology

*inter*-primitive building blocks

- Communication
- Load Balancing
- Neighborhood

Macro Primitives
TerraNeo Project

Building Blocks

• hierarchical hybrid grids

• fully distributed data structures

• support for different discretizations
Data access abstraction

Abstract index

Actual memory index

\[
\text{face\_index}(\text{level}, x, y) \Rightarrow \text{linearized index}
\]

\[
\text{face\_index}(2, 3, 1) \Rightarrow 8
\]
Splitting of unknowns

Vertex DoF

Edge DoF

Cell DoF
Splitting of unknowns

Vertex DoF

Edge DoF

Cell DoF

P2 FE

P3 FE

2x
Outlook: code generation
(in communication with ExaStencils)

dirs = ['S', 'SE', 'W', 'C', 'E', 'NW', 'N']

src = VertexTriangleField('p1FaceSrc')
dst = VertexTriangleField('p1FaceDst')
stencil = StencilField('p1FaceStencil', dirs)

s = sum([stencil[dir] * src[dir] for dir in dirs])

update = sp.Eq(dst['C'], s)
loop = createTriangleKernel('kernel', [update])
Outlook: code generation

(in communication with ExaStencils)

dirs = [ 'S', 'SE', 'W', 'C', 'E', 'NW', 'N' ]

src = VertexTriangleField( 'plFaceSrc' )
dst = VertexTriangleField( 'plFaceDst' )
stencil = StencilField( 'plFaceStencil', dirs )

s = sum( [ stencil[dir] * src[dir] for dir in dirs ] )

update = sp.Eq( dst[ 'C' ], s )
loop = createTriangleKernel( 'kernel', [ update ] )

void kernel( double * fd_plFaceDst, double * fd_plFaceSrc, double * fd_plFaceStencil, int64_t level )
{
    for ( int ctr_2 = 1; ctr_2 < ( 1 << level ) - 1; ctr_2 += 1 )
    for ( int ctr_1 = 1; ctr_1 < -ctr_2 + (1 << level); ctr_1 += 1 )
    {
        fd_plFaceDst[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2)] =
            fd_plFaceSrc[ctr_1 + (ctr_2 + 1)*((1 << level) + 2) - ((ctr_2 + 1)*(ctr_2 + 2) / 2) - 1]*fd_plFaceStencil[5]  
            + fd_plFaceSrc[ctr_1 + (ctr_2 + 1)*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2)]*fd_plFaceStencil[6]  
            + fd_plFaceSrc[ctr_1 + (ctr_2 - 1)*((1 << level) + 2) - (ctr_2*(ctr_2 - 1) / 2) + 1]*fd_plFaceStencil[1]  
            + fd_plFaceSrc[ctr_1 + (ctr_2 - 1)*((1 << level) + 2) - (ctr_2*(ctr_2 - 1) / 2)]*fd_plFaceStencil[0]  
            + fd_plFaceSrc[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2) + 1]*fd_plFaceStencil[4]  
            + fd_plFaceSrc[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2) - 1]*fd_plFaceStencil[2]  
            + fd_plFaceSrc[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2)]*fd_plFaceStencil[3];
    }
}
Annulus convection

- P1-P1 PSPG Elements for Stokes alternatively Taylor-Hood
- Finite volumes for heat transport
- 256 face primitives
- 2145 DoFs on each face p. (level 6)