PROJECT MINERVA

Accelerated Deployment of MFEM Based Solvers in Large Scale Industrial Problems Topic: 2/a

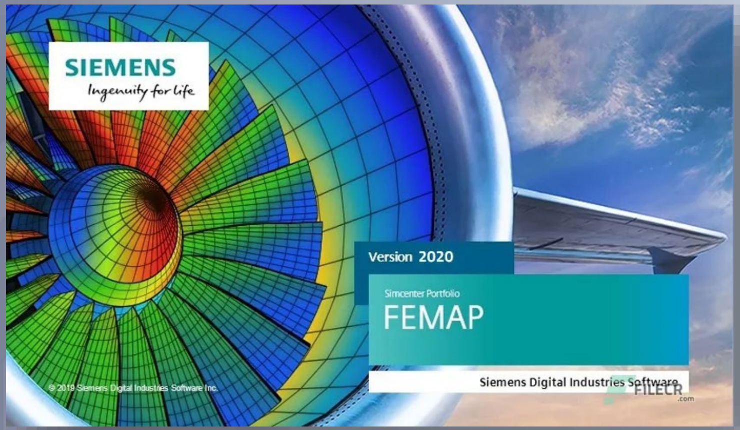
MINERVA

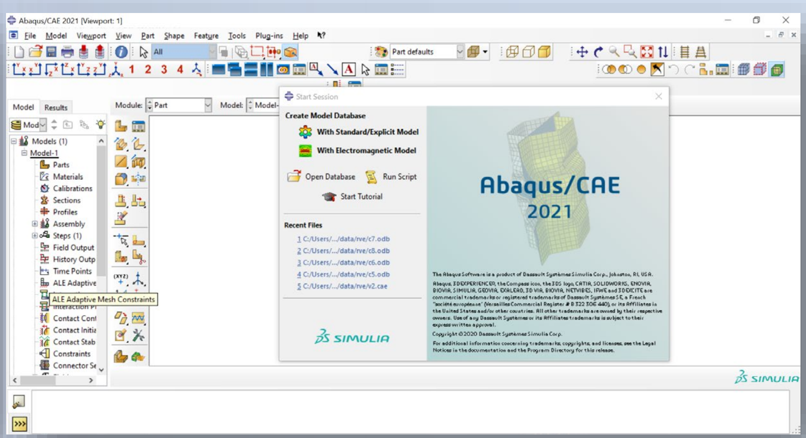
MINERVA

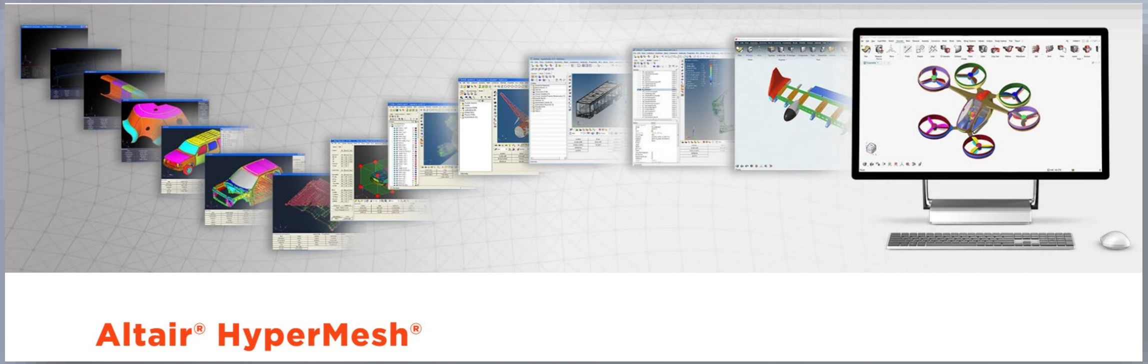
- Minerva is intended to be a secure, cloud deployable platform based on the MFEM software
  - The goal of Minerva is to accelerate HPC finite element research and application development for a wide variety of computational environments
  - It is anticipated that through collaboration, organizations in academia and industry will be early adopters of this platform to support a customer ecosystem focused on accelerating enhancements to MFEM

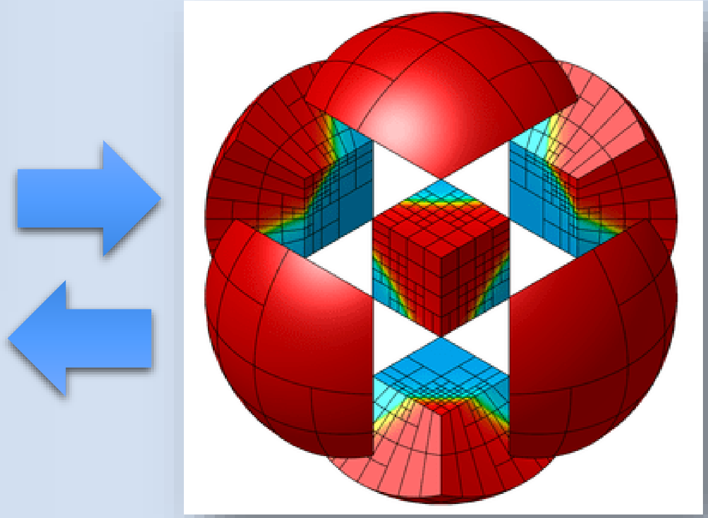
CONCEPT

- Leverage mature, commercially available CAD and FE pre/post software to develop MFEM models
- Disrupt current commercial software/hardware models for HPC FEA



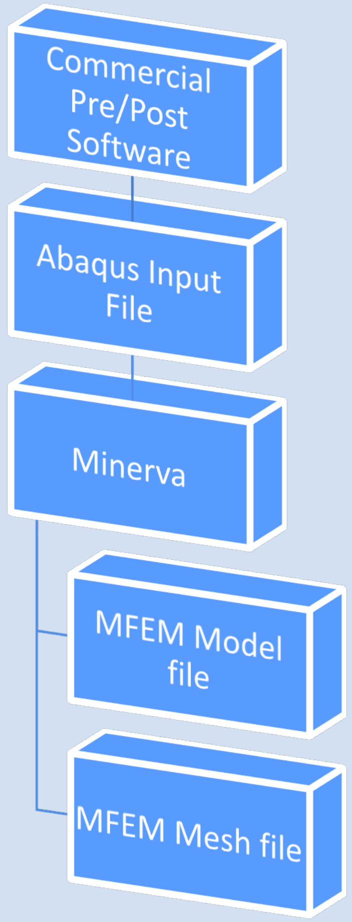






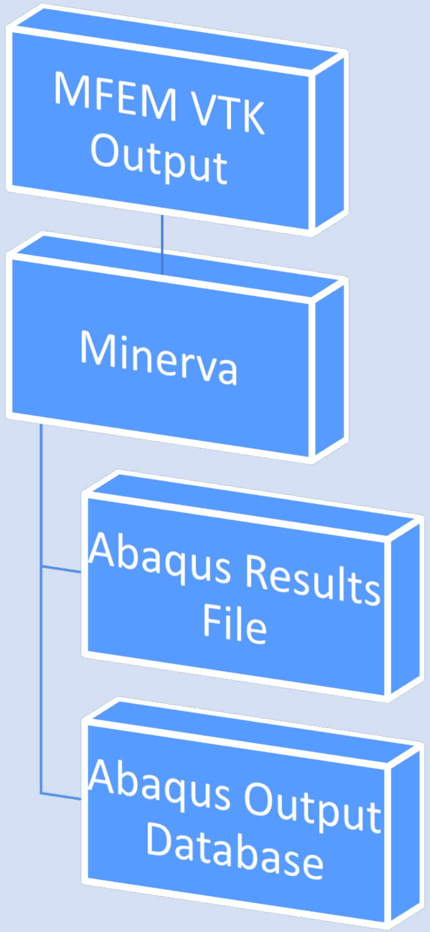
SOFTWARE LAYER

- The current layer translates an Abaqus input file into a MFEM mesh file and a MFEM model file
- The layer currently supports:
  - 3D solid continuum elements (tet, hex)
  - Essential BCs (surfaces not nodes)
  - Surface loads (pressure, traction, etc.)
  - Multiple isotropic materials definitions
  - Static, linear analysis
- The MFEM mesh file (*.mesh) is generated completely by the layer
- The MFEM model file (*.cpp) is generated by automatically populating fields in a template for static analysis
  - Serial
  - Parallel
  - Parallel + AMR



DATABASE TRANSLATOR

- The VTK output from MFEM is converted to Abaqus results file formats that are read by Hypermesh, FEMAP, Abaqus/CAE, etc.
  - A similar approach can be implemented in Phase II to support ANSYS pre/post software
- Currently supports:
  - 3D solid continuum elements (tet, hex)
  - Displacements
  - Stress tensors
  - Strain tensors

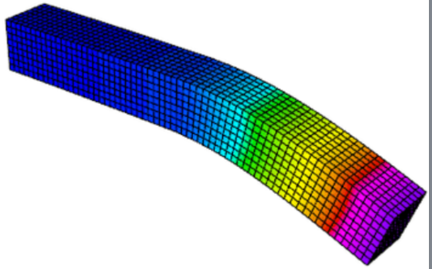


EXAMPLES

- MFEM Example-2
  - Model created in Abaqus/CAE
  - Run in MFEM
  - Visualized in Abaqus/CE

Example 2: Linear Elasticity

This example code solves a simple linear elasticity problem describing a multi-material cantilever beam. Specifically, we approximate the weak form of

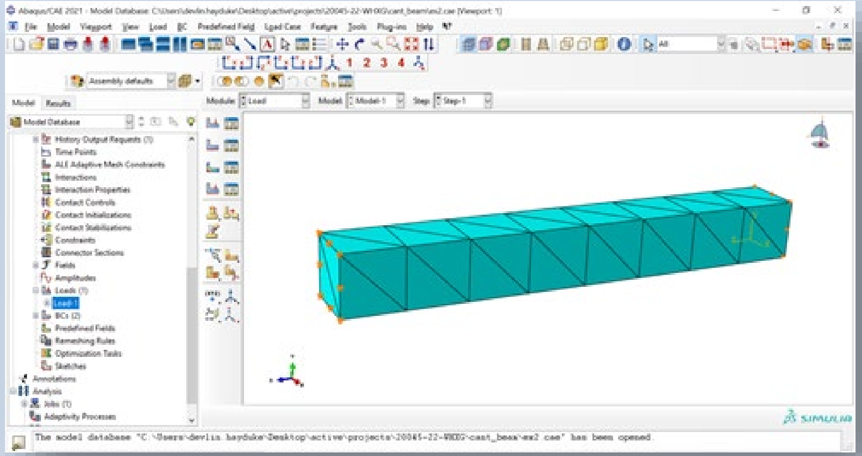


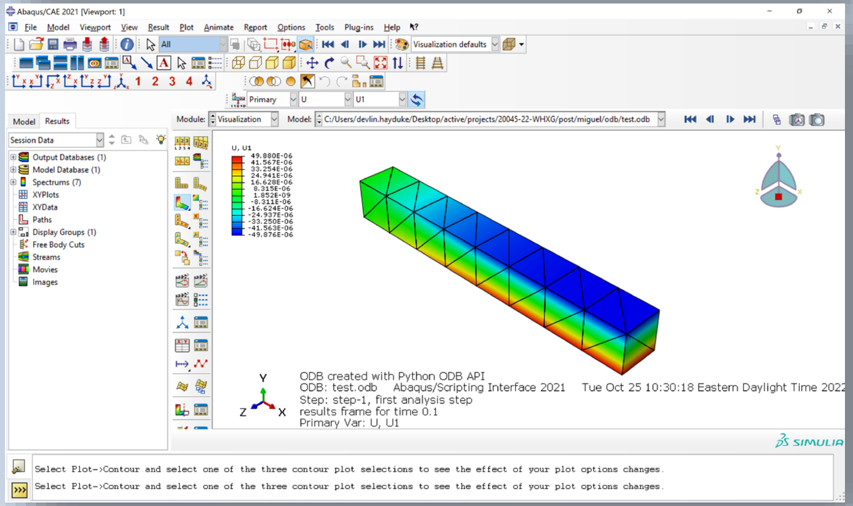
$$-\operatorname{div}(\sigma(\mathbf{u})) = 0$$

where

$$\sigma(\mathbf{u}) = \lambda \operatorname{div}(\mathbf{u})\, I + \mu\, (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$$

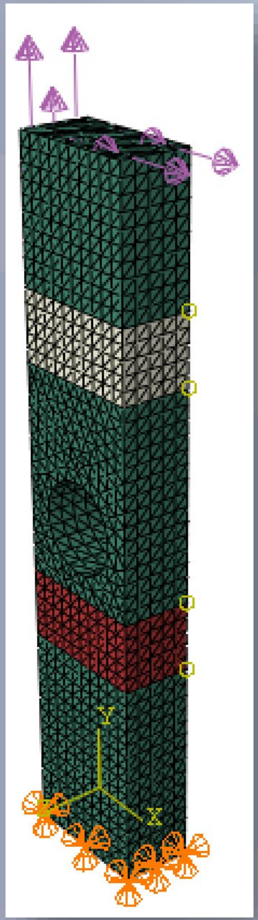
is the stress tensor corresponding to displacement field $\mathbf{u}$, and λ and μ are the material Lame constants. The boundary conditions are $\mathbf{u} = 0$ on the fixed part of the boundary with attribute 1, and $\sigma(\mathbf{u}) \cdot n = f$ on the remainder with f being a constant pull down vector on boundary elements with attribute 2, and zero otherwise. The geometry of the domain is assumed to be as follows:

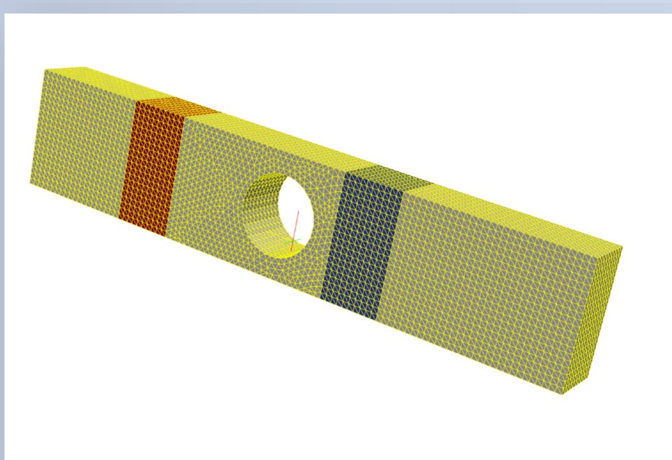


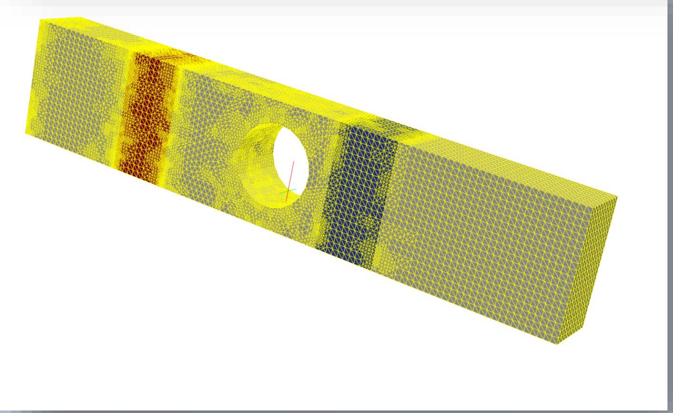


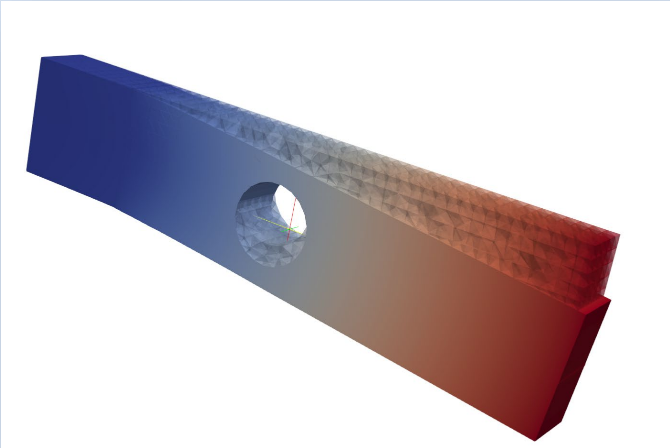
EXAMPLES

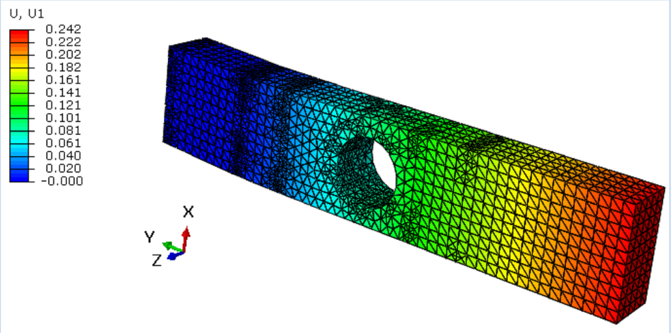
- Multi-Material + Multi-Load + AMR





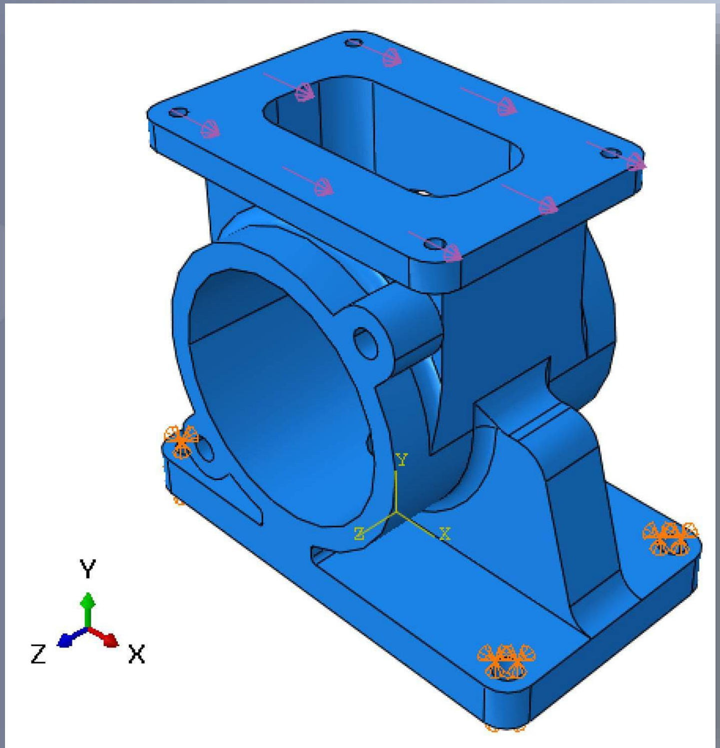


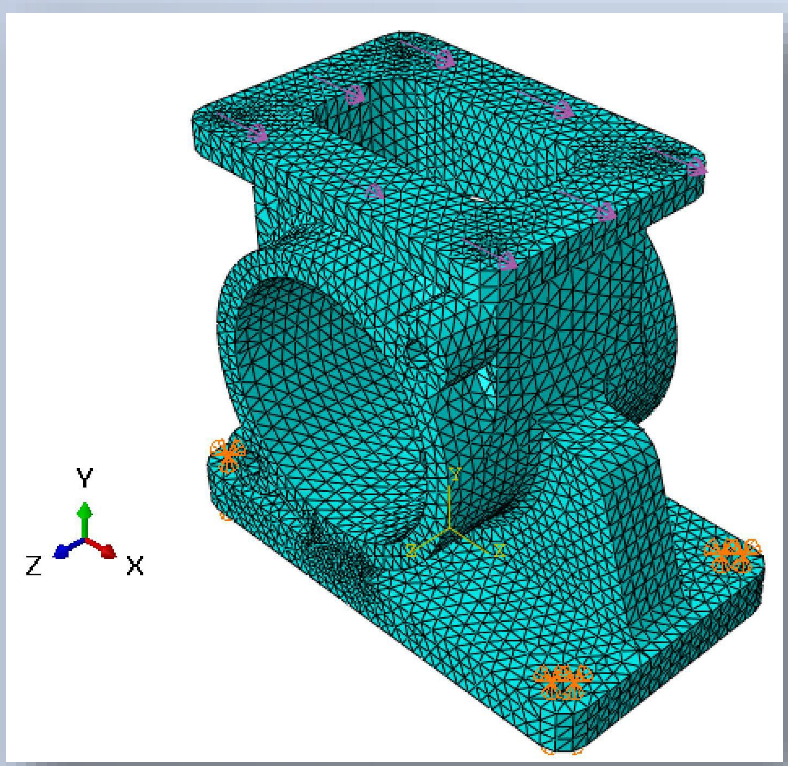


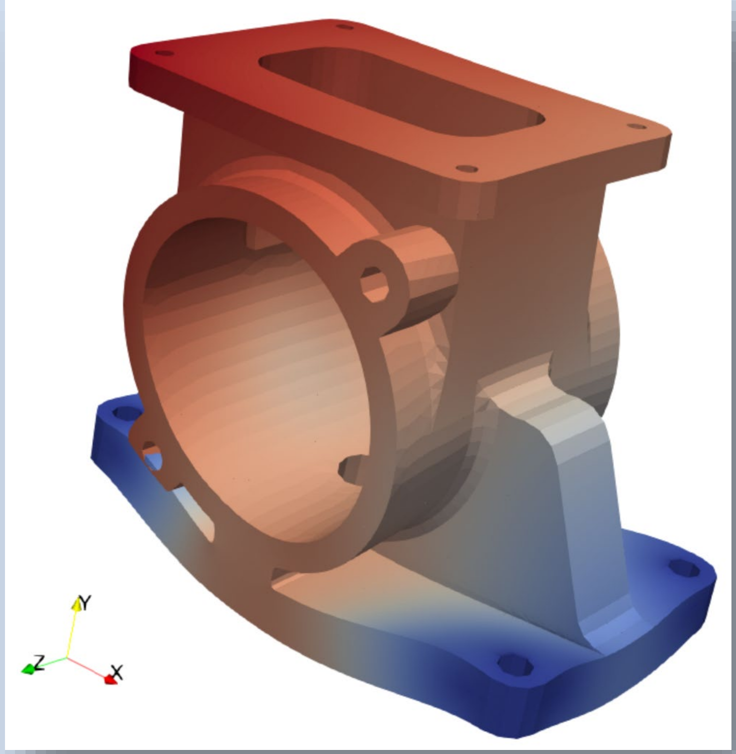


EXAMPLES

- Complex-ish Part







MFEM ENHACEMENTS

- The current effort is supporting the following MFEM enhancements:
  - Stress/strain coefficients allowing easy VTK output
    - ASCII VTK output fix
  - Local and Non-local elastoplastic solver
    - Explicit integration
    - Implicit integration
  - Support for different materials for the solver
    - Elastic-perfectly plastic
    - Orthotropic
  - Thermo-elasticity
  - Distributed loads

NON-LOCAL PLASTICITY

Governing equations:

$$-\nabla \boldsymbol{\sigma} = \mathbf{f} + \mathrm{BC}$$
$$-\nabla^{T} r^{2} \nabla \overline{\varepsilon_p} + \overline{\varepsilon_p} = \varepsilon_p$$
$$\nabla \overline{\varepsilon_p} \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial \boldsymbol{\Omega}$$

$\overline{\varepsilon_p}$ - regularized accumulated plastic strain

Constitutive behaviour:

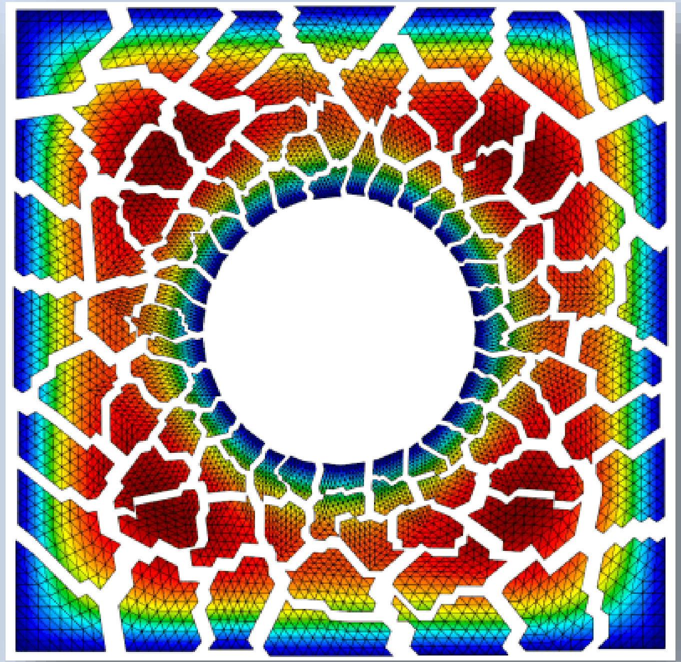
$$f\left(\boldsymbol{\sigma}, \overline{\varepsilon_p}\right) = \sigma_e\left(\boldsymbol{\sigma}\right) - \sigma_y\left(\overline{\varepsilon_p}\right)$$
$$\sigma_e\left(\boldsymbol{\sigma}\right) = \sqrt{\frac{3}{2}\boldsymbol{\sigma}^{\mathrm{D}} : \boldsymbol{\sigma}^{\mathrm{D}}}$$
$$\boldsymbol{\sigma}^{\mathrm{D}} = \boldsymbol{\sigma} - \mathrm{tr}\left(\boldsymbol{\sigma}\right)\mathbf{I}/3$$

$$\sigma_y\left(\overline{\varepsilon_p}\right) = e^{\beta \overline{\varepsilon_p}}\left(\sigma_{y,0} + H\varepsilon_p\right)$$
$$\dot{\varepsilon}_p = \sqrt{\frac{2}{3}\boldsymbol{\varepsilon}_p : \boldsymbol{\varepsilon}_p}$$
$$\boldsymbol{\sigma} = \mathbf{D}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_p\right)$$

- Highly desirable feature for structural analysis that is not available in commercial solvers
- Length scale, r, for mesh independent analysis directly relates to fracture criteria
- Currently being implemented and verified
  - Add to next MFEM release

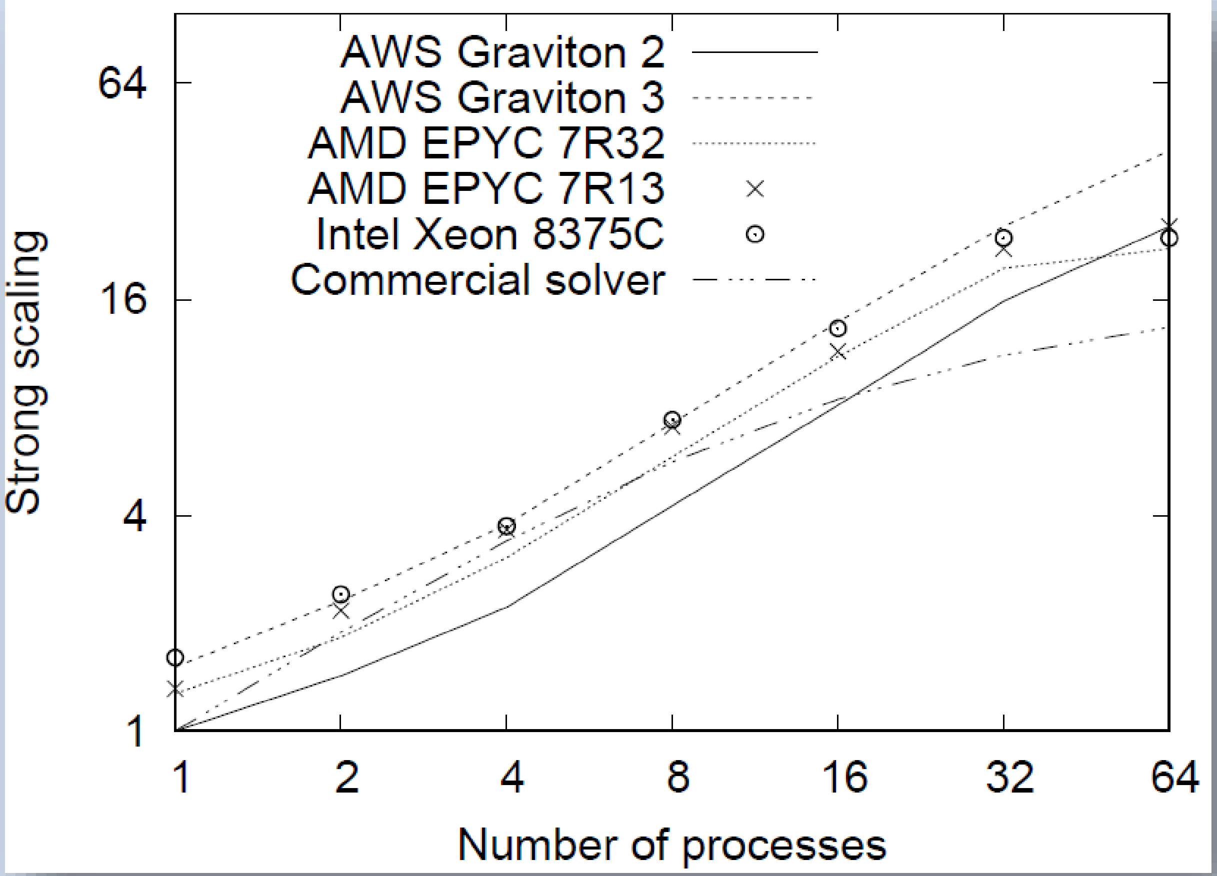
CLOUD-BASED PLATFORM

- The team is currently evaluating Amazon Web Services (AWS) and platforms that provide frontends to manage AWS for software deployment
- The scalability study is performed on five different AWS 64-virtual CPU machines with ARM and x86 architectures
  - The ARM machines are with Amazon's CPUs Graviton 2 and 3 with 64 cores, i.e., every virtual CPU corresponds to one CPU core
  - The x86 machines are multithreaded, and two virtual CPUs are mapped to one physical core
- The test code is an example from MFEM, executed on four times refined hex mesh with a total of 7.2M DOFs



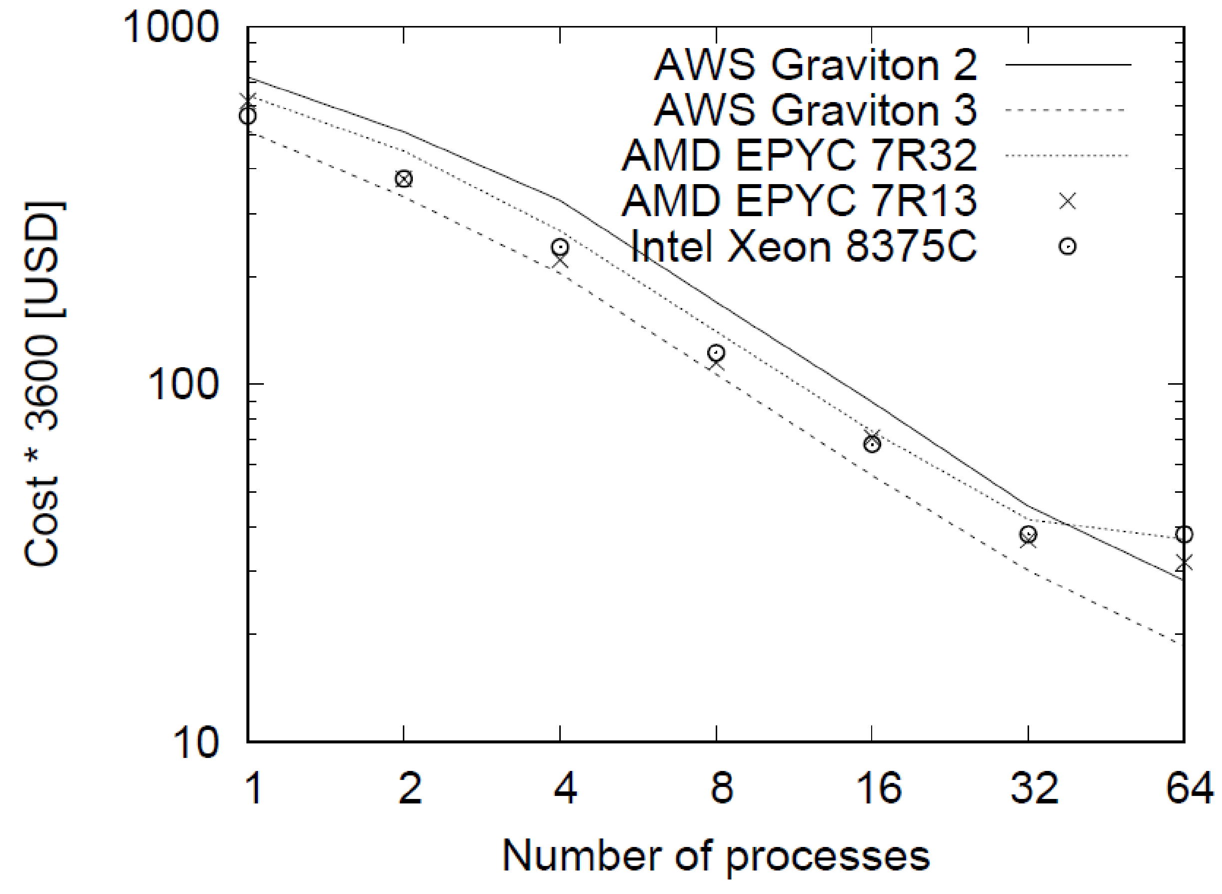
HPC PERFORMACE

- The scalability study is performed on five different AWS 64-virtual CPU machines with ARM and x86 architectures
  - The ARM machines are with Amazon's CPUs Graviton 2 and 3 with 64 cores, i.e., every virtual CPU corresponds to one CPU core
  - The x86 machines are multithreaded, and two virtual CPUs are mapped to one physical core
- On all platforms, the scalability is close to perfect, except for the x86 architectures
  - At 32 processes, the curve for the x86 machines is flattening, a well-known behavior on CPU with enabled multi-threading
- Commercial solvers do not offer such nice scalability
  - In comparison to MFEM, we can observe a factor of two differences in performance
- In addition, we are not aware of any major commercial solver available for ARM architectures



HPC COST

- At 64 CPUs, the Graviton 3 machine is approximately four times cheaper than any of the x86 machines, and the Graviton 2 machine is around two times cheaper
- Compared to an equivalent simulation with commercial software, the difference will be a factor of eight which clearly demonstrates the possibilities for cost saving in addition to proven parallel scalability
- Of course, to lower the cost, one can select smaller machines for runs on a smaller number of processes than 64
  - However, the available memory on such machines will limit the size of the problem
- Currently, commercial solver licenses are priced per CPU and cloud-based deployments are priced to more than annual licenses



PROJECT TEAM:

Devlin Hayduke, ReLogic (PI)

FE research:

- Material models and element formulations for analyses of composite structures
- Topology optimization for advanced manufacturing applications

Miguel Agulio, ReLogic

FE research:

- Former developer of the Plato optimization software at SNL
- Uncertainty quantification

Steve Pilz, ReLogic

FE product development:

- Former Lead Product Manger at ANSYS
- Adept at building coalitions and cooperative relationships between businesses and academia

Boyan Lazarov, LLNL

FE research:

- MFEM development team
- Optimization, computational mechanics, non-linear mechanics, structural reliability, etc.

Thank you!
relogicresearch.com

Chandler Wicks, CEO
chandler.wicks@relogicresarch.com

Devlin Hayduke, CTO
devlin.hayduke@relogicresearch.com