NASA Postdoctoral Program

A Partitioned - Task Parallel Implementation of the NASA Multiscale Analysis Tool for High Performance Computing

Ibrahim Kaleel[1]
Trenton M. Ricks[1]
Peter A Gustafson[2]
Evan J. Pineda[1]
Brett A. Bednarcyk[1]
Steven M. Arnold[1]

[1]NASA Glenn Research Center

[2]Western Michigan University, Kalamazoo, MI, 49008, USA

2022 American Society for Composites Conference,
September 19-21, 2022, Tucson, Arizona

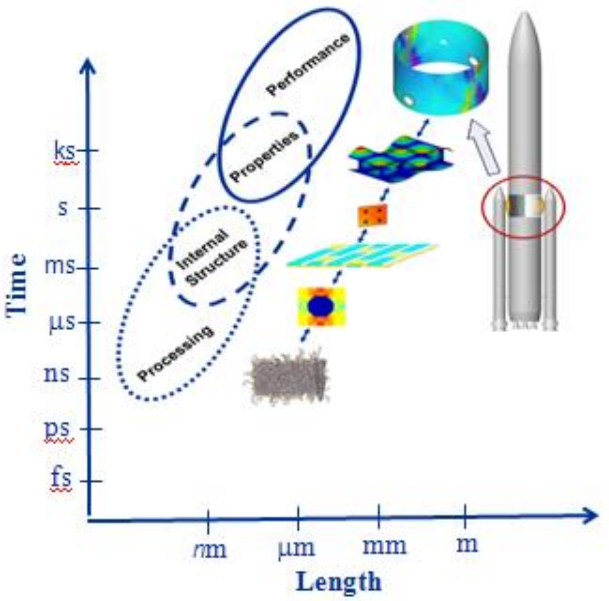
Introduction

- Massively multiscale modeling often constrained by scalability issues
- Real engineering problems with multiscale models are still computationally infeasible
- NASA Vision 2040 recognizes following gaps under Key Element 9: Computational Infrastructure
  - Scalability and Computational Efficiency
  - Linkage and Integration
- Multiple scales and their interdependence (hierarchical task dependence) poses additional restrictions in terms of implementation
  - Efficient parallelization of hierarchical tasks still an open problem in computer science

Liu, Xuan, et al. *Vision 2040: a roadmap for integrated, multiscale modeling and simulation of materials and systems*. NASA/CR-2018-219771.

NASA Multiscale Analysis Tool (NASMAT)

- Successor to MAC/GMC and FEAMAC toolsets
- Designed to support massively multiscale modeling (M3) on high-performance computing (HPC) systems
  - Solves real, large-scale, non-linear, thermo-mechanical problems
- Modular design to support “plug-and-play” capabilities
  - Operational components categorized into NASMAT procedures
- Developed for enhanced interoperability
  - Integrates with 3rd party structural analysis codes (e.g., FEA)
  - Library of constitutive laws/damage models
  - Data output in HDF5 file format
- Ideal for “design with” or “design of” the material
  - Enables Integrated Computational Materials Engineering (ICME)
  - Developed to support NASA Vision 2040

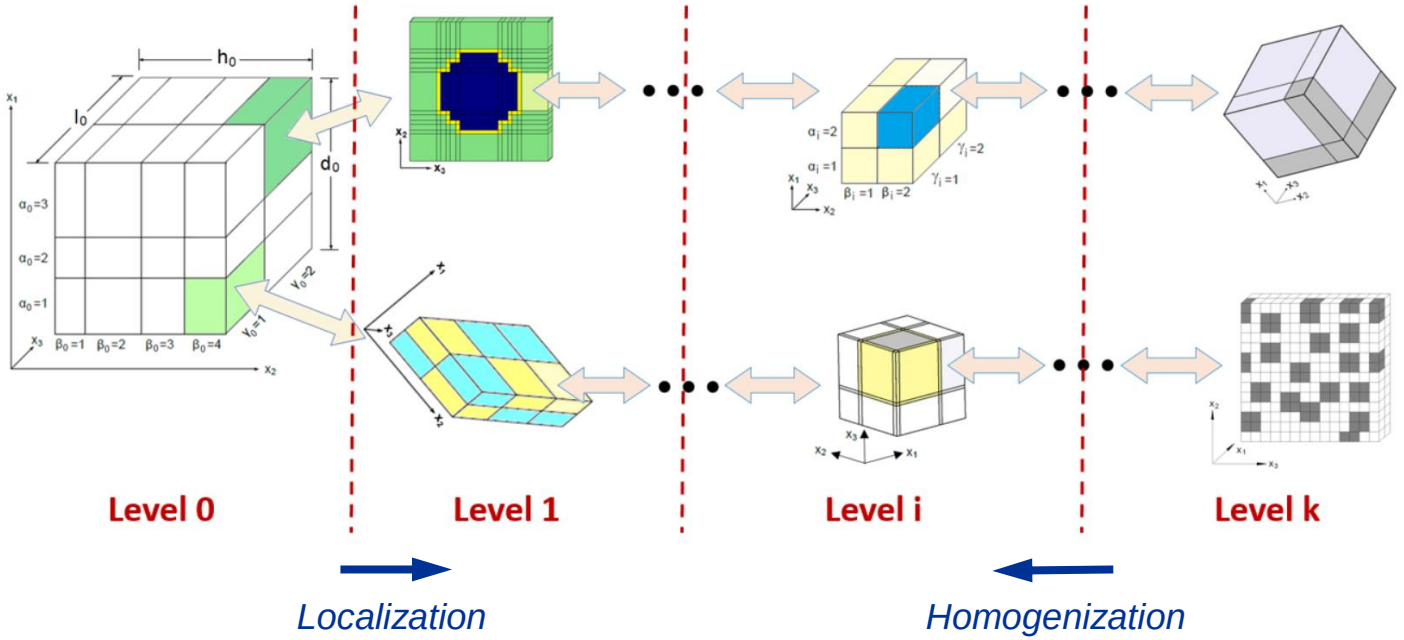


Multiscale localization/homogenization

Liu, Xuan, et al. *Vision 2040: a roadmap for integrated, multiscale modeling and simulation of materials and systems*. NASA/CR-2018-219771.

Multiscale Recursive Micromechanics (MsRM)

- Suite of micromechanics theories available
- Recursive in nature
- Arbitrary number of scales



Current Challenges and Shortcoming

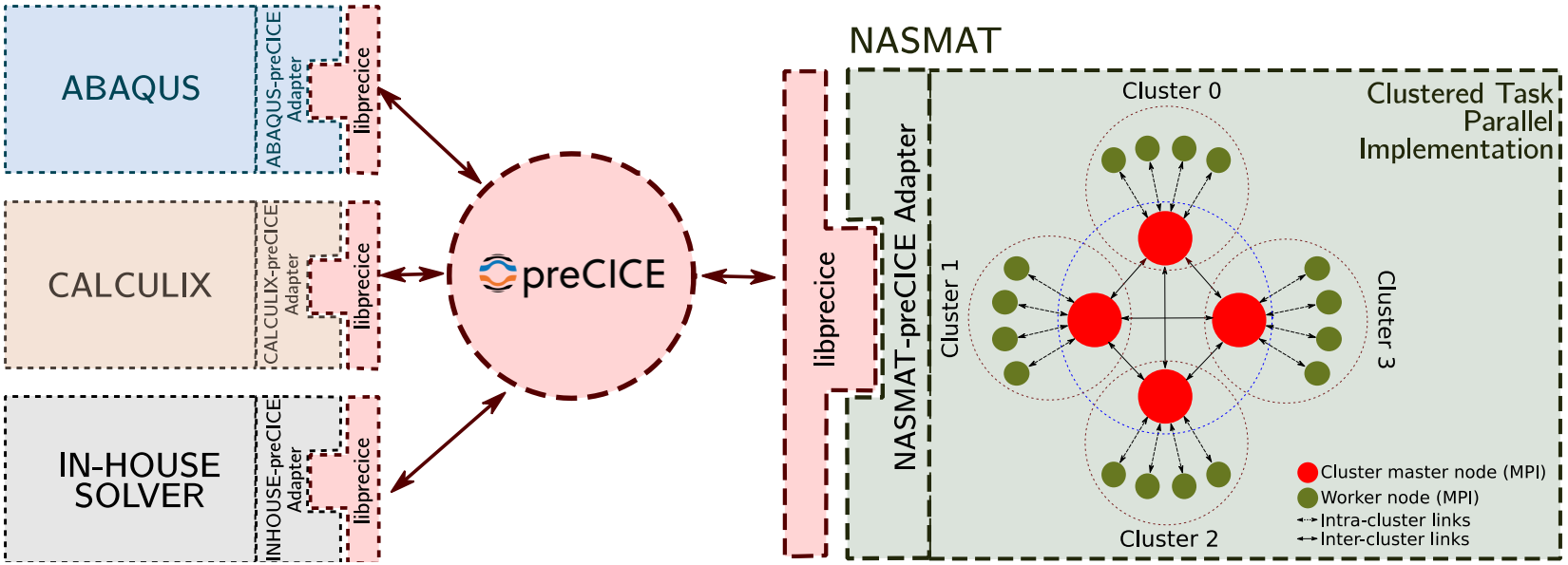
- Undertaking massively multiscale modeling still a challenge in terms of computational feasibility

- Conventional Monolithic approach (UMAT-like)
  - Heavily constrained by architecture of macro solver
  - Limits parallelization opportunities (not more than 1 core/thread available)
  - Introduce macro solver specific changes
  - Disparate resource management (reduced parallelization efficiency)

- Adapting to heterogenous high performance computing system (CPU/GPU, multi core/threads etc.)

Solution

- A partitioned task-parallel architecture is proposed
- Task-parallel architecture - internal NASMAT parallelization
- Partitioned architecture – external integration with macro solver
- Modular and scalable architecture

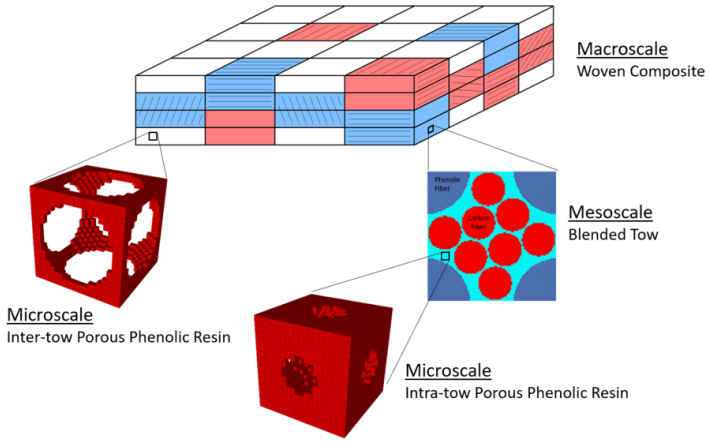


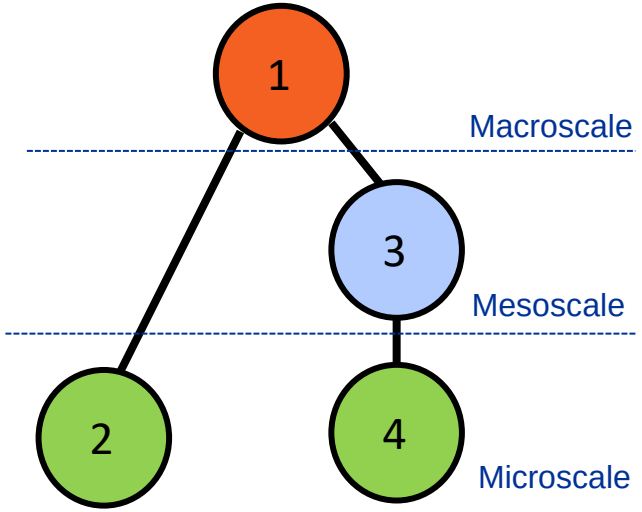
Planas, J., et al., (2009). Hierarchical Task-Based Programming With Starss. The International Journal of High Performance Computing Applications, 23(3), 284–299. https://doi.org/10.1177/1094342009106195

NASMAT 'Task-Parallel' Architecture

Task-Parallel

- Corresponds to the parallelization scheme adopted within NASMAT
- Abstracted definition of Task - refers to a *homogenization/localization* call
- Multiscale RUC architecture is defined as a tree

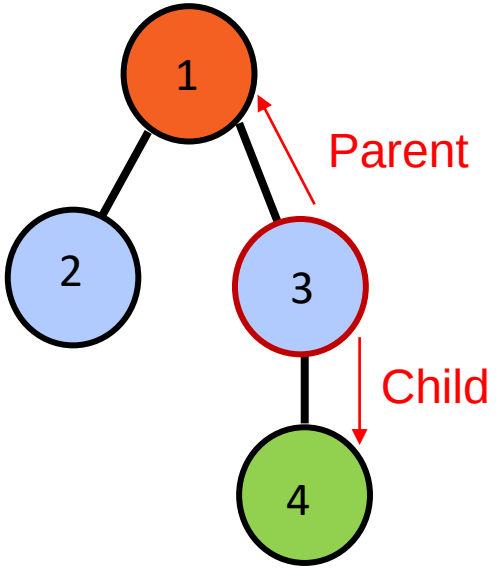




Example Tree for 3 scale system (Only unique RUCs represented)

NASMAT '**Task-Parallel**' Architecture

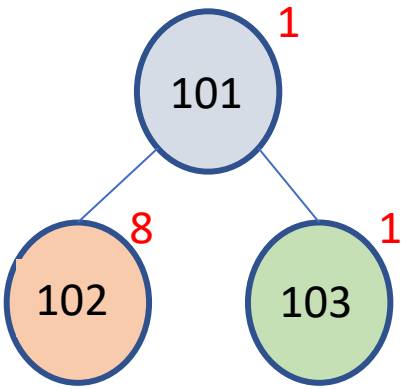
- Each RUC tree is decomposed into Nodes
- Parent-Child relationships are established between *Nodes*
  - Ex: *Node3* : Parent-*Node1*, Child-*Node4*
  - Each Node may have multiple children
- Different Master-Worker based parallelization is adopted to undertake homogenization/localization task
  - MPI based
  - OpenMP based
- Looping order decided based on type of task (Homogenization/Localization)



Example Tree

NASMAT 'Task-Parallel' Architecture: Algorithm

- Example: RUC tree with 2 scales with 9 subcell each



Tree Diagram: Runtime

Total tasks: 10

SCALE 0

MSM=0, ID:101

-17	-17	-17
-17	-18	-17
-17	-17	-17

SCALE 1

MSM=-17, ID:102

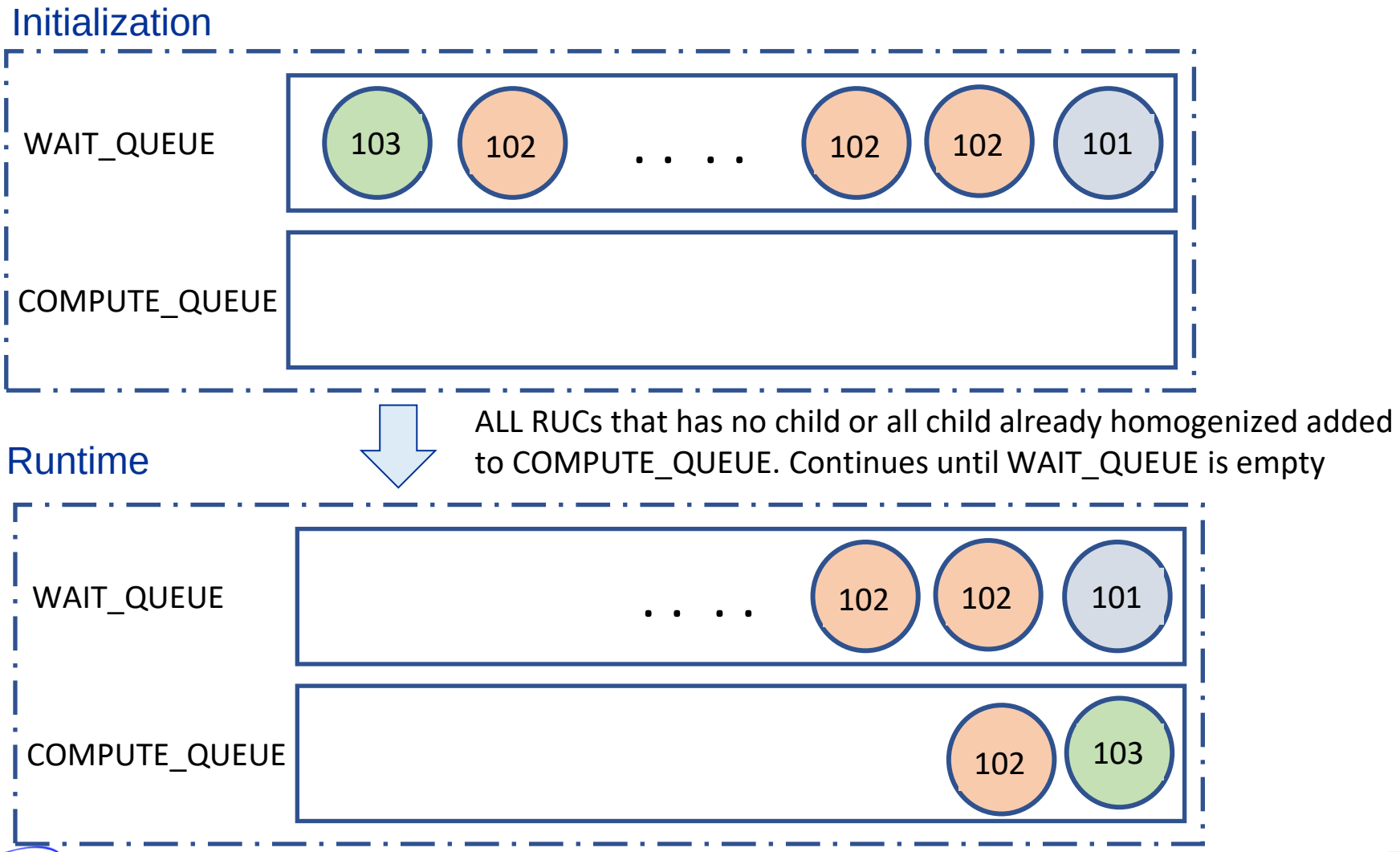
2	2	2
2	1	2
2	2	2

MSM=-18, ID:103

4	4	4
4	3	4
4	4	4

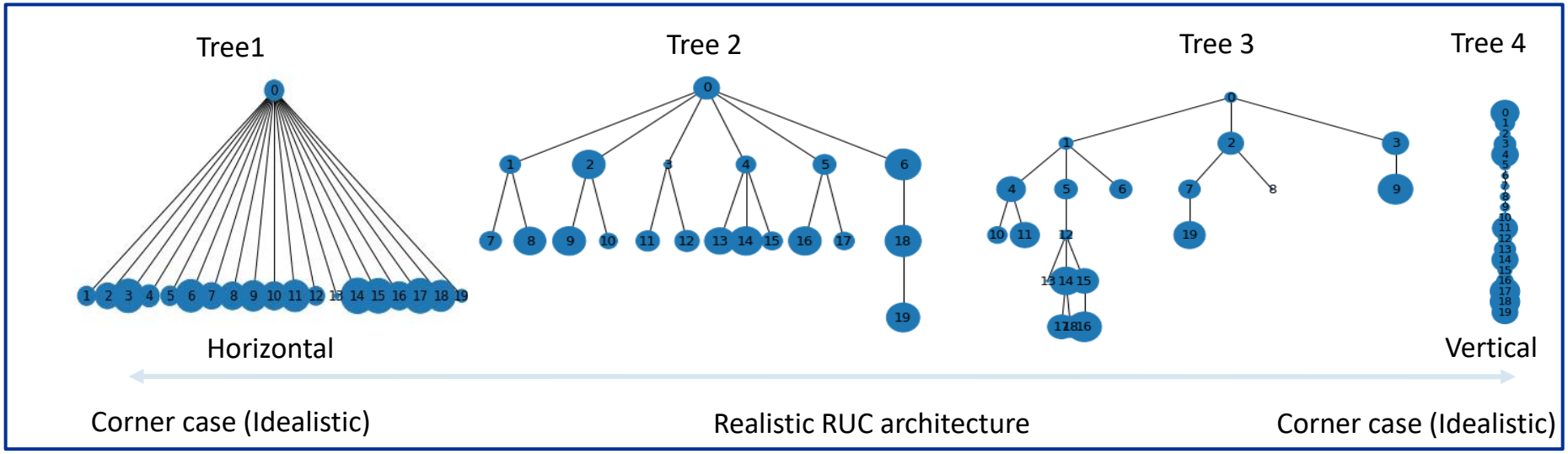
Positive material Tags: Constitutive models
Negative material Tags: RUC

NASMAT 'Task-Parallel' Architecture: Algorithm



Comparison of two different scheme for Homogenization Task

- Parallelization scheme
  - Tree Level
    - Each RUC associated with a macro GP is given “n” CPUs
  - Accumulated Tree scheme
    - All nodes from all macro GP are accumulated and distributed parallelization is undertaken with “n” CPUs

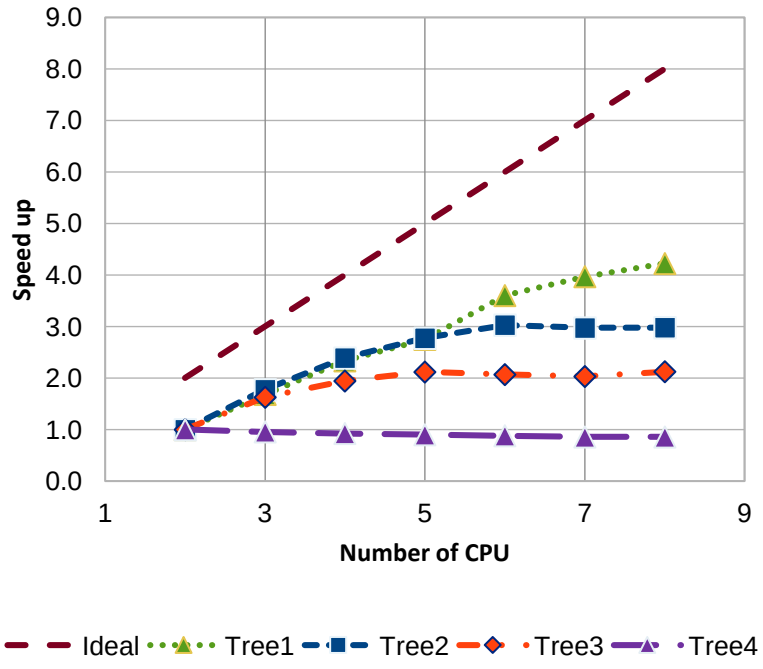


Each tree contains 20 nodes with DOF varying from 100-100K ((equivalent unit cell distribution) with 100 macro GP

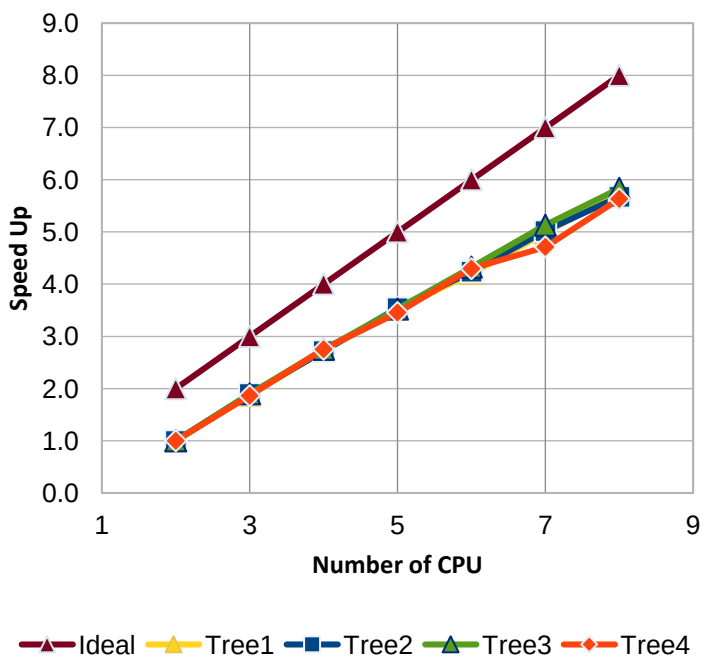
Comparison of two different scheme for Homogenization Task: Results

Micromechanics model: GMC 2D – Linear Elastic Homogenization

Tree-level



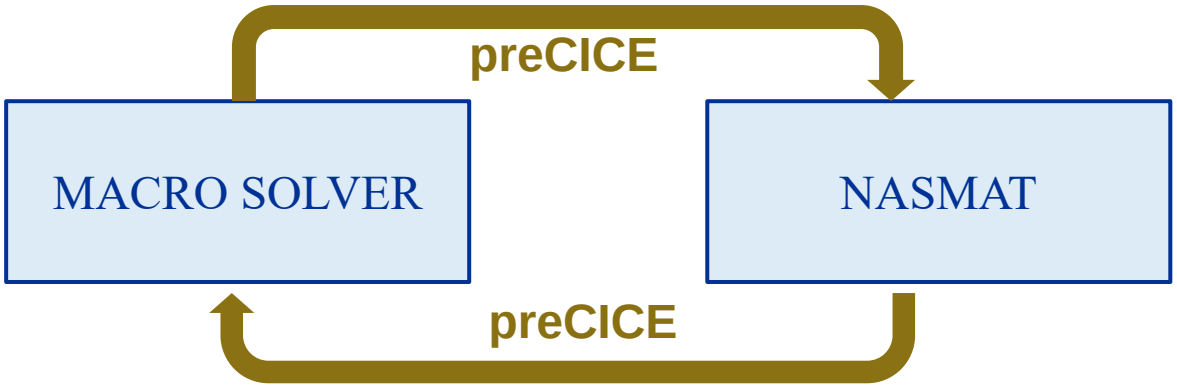
Accumulated Tree-level



- Scaling is RUC architecture independent for Accumulated Tree-Level scheme

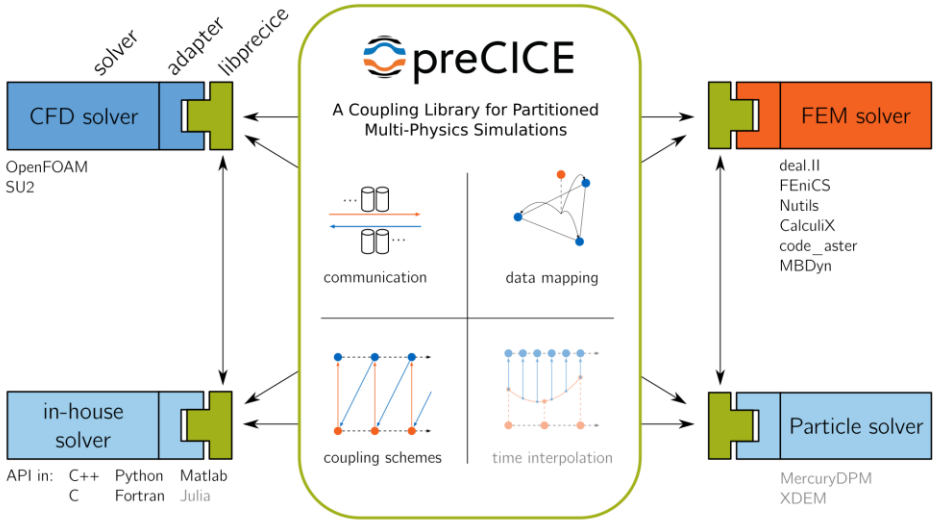
NASMAT ‘Partitioned’ Architecture

- Corresponds to the way NASMAT interacts with macro solver
- FSI (Fluid Structure Interaction)-like modeling approach
- Two separate processes are used for macro and multiscale solvers (Staggered scheme) – communication protocol established between the two processes
- Architecture and implementation are macro-solver independent
- Implemented using **preCICE** (open-source coupling library)



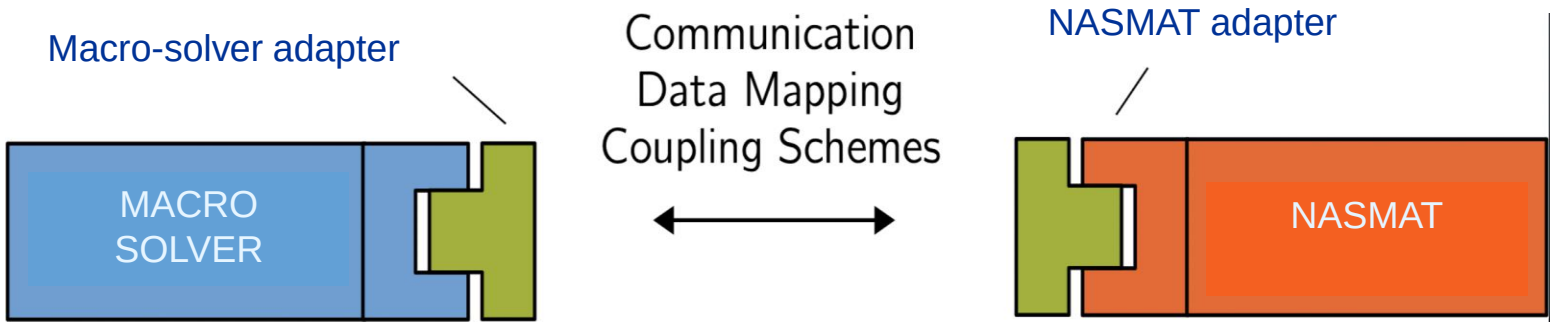
preCICE: Precise Code Interaction Coupling Environment

- Couples existing programs (solvers) which simulate a subpart of the complete physics involved in a simulation
- Open-source library
- High-level API calls – Multiple language support
- Minimally invasive integration
- Developed at University of Stuttgart and Technical University of Munich



Chourdakis G, *et al.* preCICE v2: A sustainable and user-friendly coupling library Open Res Europe 2022, **2**:51

NASMAT-preCICE Integration



Macro Solver:

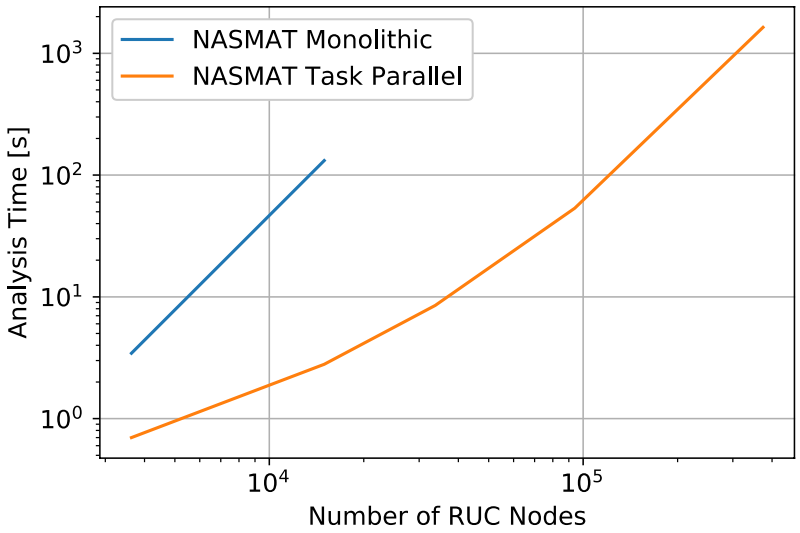
1. Macro solver can be written in any language supported by preCICE API
2. preCICE adapter for Calculix is readily available

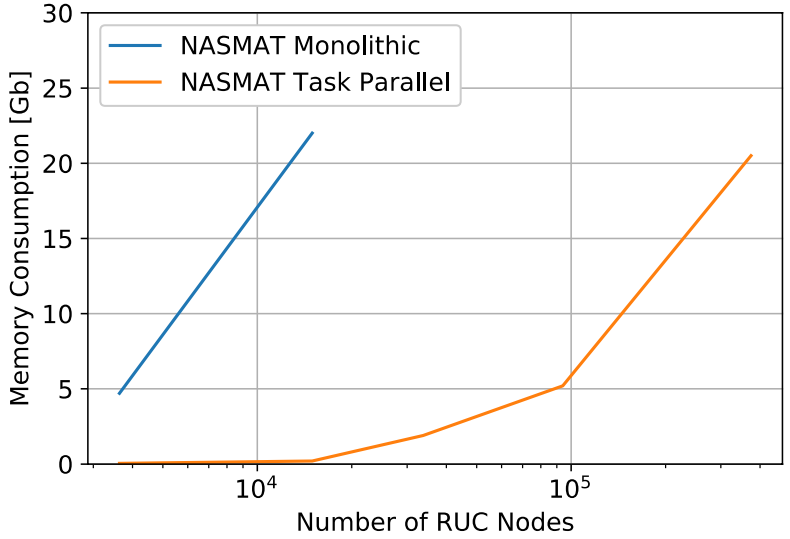
NASMAT

1. preCICE adapter for NASMAT is implemented
2. preCICE adapter calls NASMAT libraries – no changes to original codebase

Comparison of NASMAT Monolithic vs NASMAT Task Parallel

Homogenization task for GMC 2D with an RUC with 3 levels





- Significant improvement in terms of analysis time and memory requirement [~1 order of magnitude] for GMC2D models

Summary and Conclusions

- NASMAT Partitioned Task-Parallel architecture addresses the scalability and integration associated with massively multiscale modeling architectures
- Macro-solver integration of NASMAT through preCICE, an open source coupling library allows for easy integration of NASMAT to any third-party external library
- Task-based parallelization of homogenization/localization calls allows for various shades of parallelization
- In comparison to previous iteration of NASMAT, the proposed architecture shows significant (~1 order of magnitude) reduction in analysis time and memory requirement

Future Work

- Extending task parallel implementation to HFGMC micromechanics models within NASMAT
- Improving memory efficiency of task parallel implementation using shared memory within MPI
- Building preCICE adapter for ABAQUS

Acknowledgements

- Primary NASMAT development has been supported through the NASA Aeronautics Research Mission Directorate's (ARMD's) Transformational Tools and Technologies (TTT) Project.

- Ibrahim Kaleel's research was supported by an appointment to the NASA Postdoctoral Program at the NASA Glenn Research Center, administered by Oak Ridge Associated Universities under contract with NASA.

Thank You For Your Attention

Extra Slides

Task Parallel Flowchart