Toward an Exascale Solution for Heterogeneous Multiscale Modeling

ExMatEx

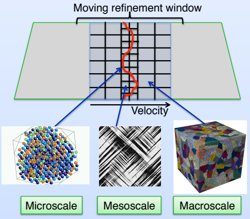
Exascale computing presents an enormous opportunity for solving some of today's most pressing problems, including clean energy production, nuclear reactor lifetime extension, and nuclear stockpile aging. At their core, each of these problems requires the prediction of material response to extreme environments. Our Center's objective is to establish the interrelationship between software and hardware required for materials simulation at the exascale while developing a multiphysics simulation framework for modeling materials subjected to extreme mechanical and radiation environments.

Heterogeneous Multi-Scale Modeling of Shockwave Propagation in Materials

- ❑ **Goal**: non-oscillatory high-resolution scheme for two-dimensional hyperbolic conservation laws to simulate shockwave propagation in the non-linear response regime
- ❑ **Method**: finite volume macro solver models the evolution in time and space of a coupled system of PDEs:

$$\partial_t \mathbf{A} \quad - \nabla_{x^0} \boldsymbol{v} \quad = 0$$
$$\partial_t \boldsymbol{q} \quad - \nabla_{x^0} \cdot \boldsymbol{\sigma} = 0$$
$$\rho_0 \partial_t e - \nabla_{x^0} \cdot \boldsymbol{j} \quad = 0$$

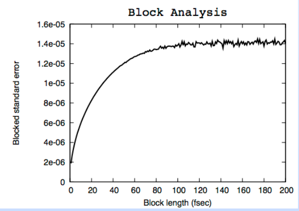
- ❑ **Finite volume**: conserved properties inside a cell, fluxes through the surface



- ❑ **Multi-Scale approach**: finite volume method calls MD simulation with input parameters strain **A**, momentum density ***q***, and energy density ***e***. The stress ***σ***, and the energy density flux ***j*** is then evaluated on the microscopic level and is given back to the macro solver.

Error Analysis

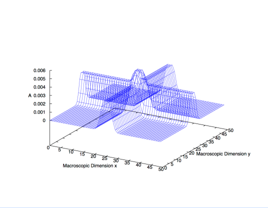
The block average analysis enables to quantify the uncertainty of the microscopic simulations, that can then be set to match the sensitivity of higher level simulations.



- ❑ To estimate the sensitivity of the macroscopic simulations we use an analytical response of the finer level with a Gaussian noise.
- ❑ The size and length of the molecular dynamics microscopic simulations is set accordingly, with respect to the potential used.

2D Macrosolver

Example of the two dimensional macro solver, with two shock fronts evolving through the system. The picture shows the total strain in the simulation after a couple of time steps.



For this simple example of 50x50 grid points, the Molecular dynamics (CoMD) has to be called 2500 times. In order to compute a single time step at macro level, we need to execute this operation 4 times.

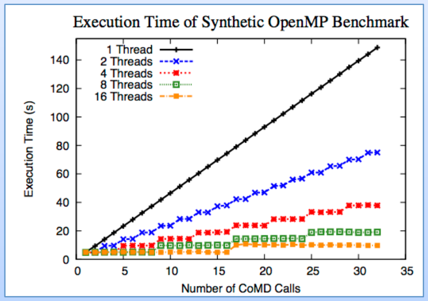
Co-Design and a Multi-Disciplinary Approach

Co-Design is the blurring of the user and designer. By ensuring that the users of the system are the ones designing it, the quality of the design increases as unnecessary features are culled and the researcher is able to ensure required functionality.

This summer school used a multi-disciplinary approach consisting of student interns trained in physics and computer science to develop a solution of the problem while using efficient computing techniques.

Identifying The Bottlenecks

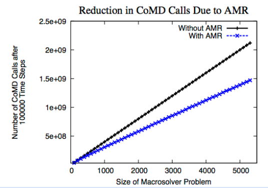
We developed a synthetic benchmark with OpenMP to identify where our time would best be spent



Clearly, our goal is to maximize parallelism and reduce the number of calls to the MD simulator while maintaining a high degree of accuracy.

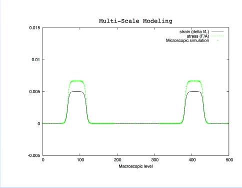
Adaptive Mesh

Mesh coarsening



Our first level of adaptive mesh uses a fine grid everywhere and coarsen it where there is no need for precise computation. The grid is left at its initial finest where the shock front passes, and is coarsened away from the shock front.

Mesh refinement



On a second level of mesh adaptivity, the results from the microscopic level are interpolated. Microscopic simulations are spawned where the interpolation requires more data to be effective. This grid refining approach considers microscopic outputs only, allowing numerical experiments where the relation between the simulation levels is unpredictable.

We acknowledge the support and guidance of our mentors, Tim Germann and Allen McPherson. We also thank Christoph Junghans, Jamal Mohd-Yusof and Kipton M. Barros for additional advisement. Finally, we extend our gratitude to Mike McKerns for his assistance in using the Pathos runtime.

Runtime and Programming Model

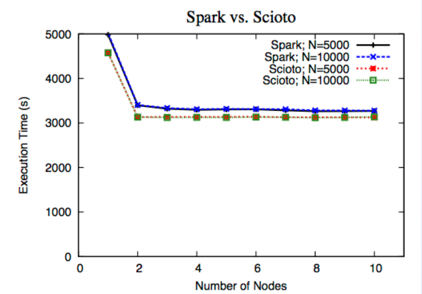
Requirements:

- ❑ No Emphasis on Locality
- ❑ Support for Distributed Systems
- ❑ Runtime-managed Job Scheduling and Load Balancing
- ❑ Resilience
- ❑ Potential for High Scalability

Runtime	Developer	Advantages	Disadvantages
Scioto	Ohio State University	Load balances asynchronous tasks uses Global Arrays for PGAS	Early development
Spark	AMPLab	Highly resilient framework for in-memory cloud computing	Full cluster computing system
Swarm	ET International	Codelet based model with proven performance	No distributed load balancing
Pathos	Mike McKerns	Uses map-reduce to distribute tasks via MPI	High overhead

Comparison of Runtime Performance

For performance and time purposes, we chose to compare the performance of a traditional model, Scioto, and a less traditional framework, Spark. Both achieved similar performance



All benchmarks were run for 30 steps at the macrosolver level on LANL's Darwin Cluster using 48 cores per node with visualization disabled and the number of grid points specified in the graph. Each grid point contained an MD simulation with 864 particles that converged after 1000 steps. The described Adaptive Mesh Refinement technique was used.

Fault Tolerance

As we approach the exascale era, system faults become a real possibility. We take a two-pronged approach to fault tolerance:

- ❑ **Task Level**
  - ○ One or more nodes go down, but simulation continues
  - ○ Failed MD simulations are restarted on remaining nodes
- ❑ **Job Level**
  - ○ Enough nodes go down that simulation stops
  - ○ Simulation state saved when visualizing output. Can recover from saved state

This work was approved for unlimited public release as per LA-UR-13-25736
and LA-UR-13-25606