Uintah Framework Hybrid Task-based Parallelism Algorithm

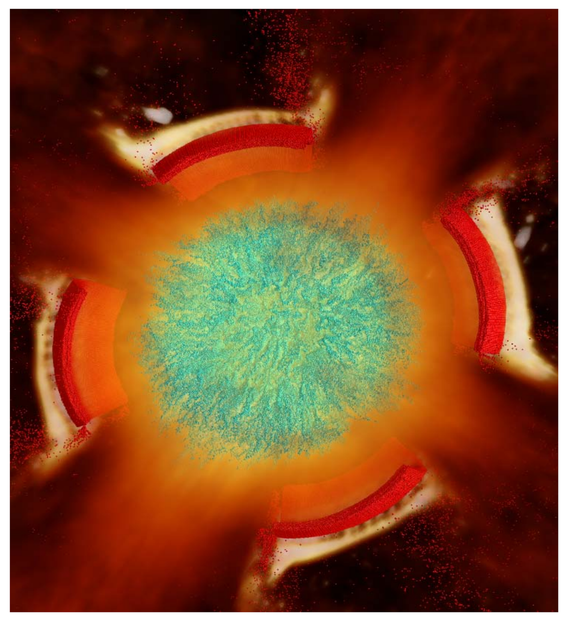
Qingyu Meng and Martin Berzins. Scientific Computing and Imaging Institute, University of Utah

Introduction

The Uintah Computational Framework (UCF) is a software framework that provides an environment for solving fluid-structure interaction problems on structured adaptive grids on large-scale science and engineering problems involving the solution of partial differential equations.

Uintah Applications:

- Explosions
- Plume Fires
- Industrial Flares
- Shape Charges
- Virtual Soldier
- CPU Mircopin Flow
- Foam Compaction
- Angiogenesis
- Sandstone Compaction



Uintah uses a combination of fluid-flow solvers and particle-based methods for solids, together with adaptive meshing and a novel asynchronous task-based approach with fully automated load balancing.

Challenges

Solve complex fluid structure interaction problems on parallel computers.

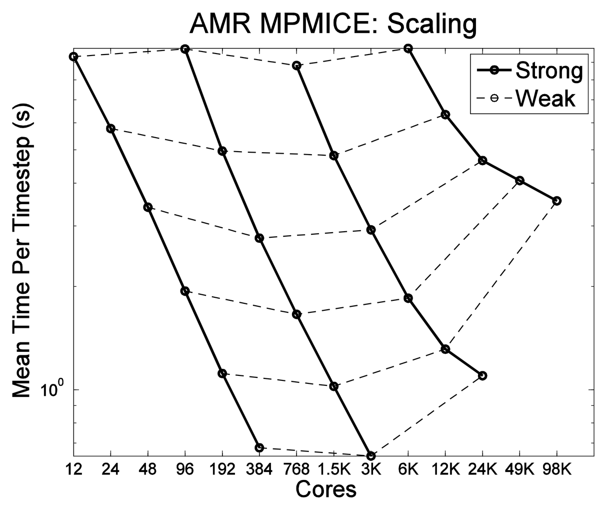
- Full physics - strong coupling between the fluid and solid phases with a full Navier-Stokes representation of fluid phase materials and the transient, nonlinear response of solid phase materials include chemical or phase transformation between the solid and fluid phases
- Multi-material - each material is given a continuum description and is defined over the complete computational domain.

With original MPI only approach, Uintah can successfully scale up to 98K.

- Adaptive Mesh Refinement Algorithms
- Measurement-based Load Balancing
- Out-of-order Task Execution
- Data Migration

Poor weak scaling efficiency – 23% at 98K cores.

- Hard to load balance
- High communication cost

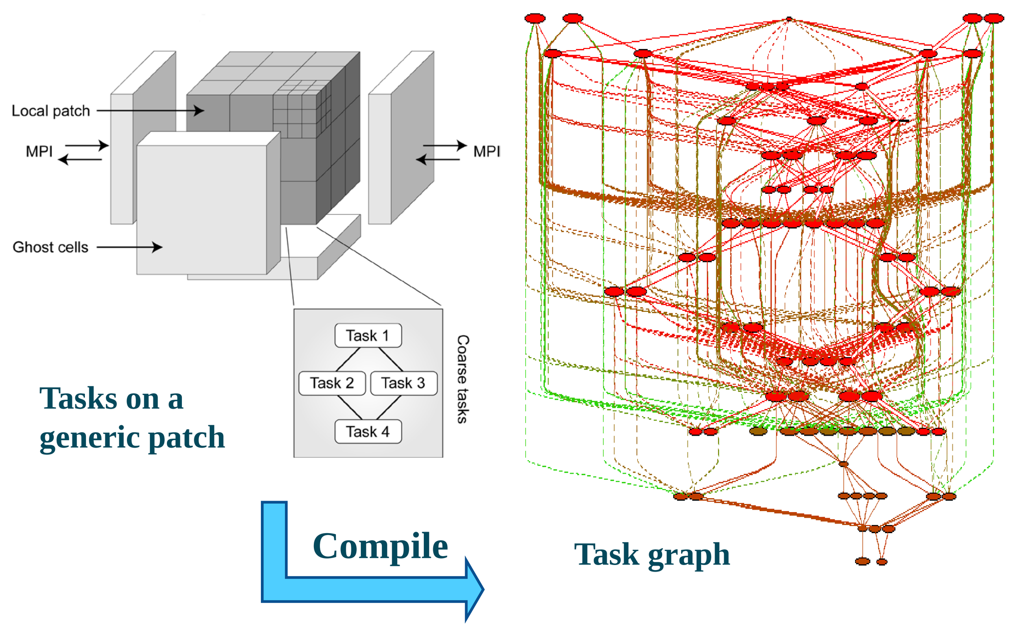


Solution

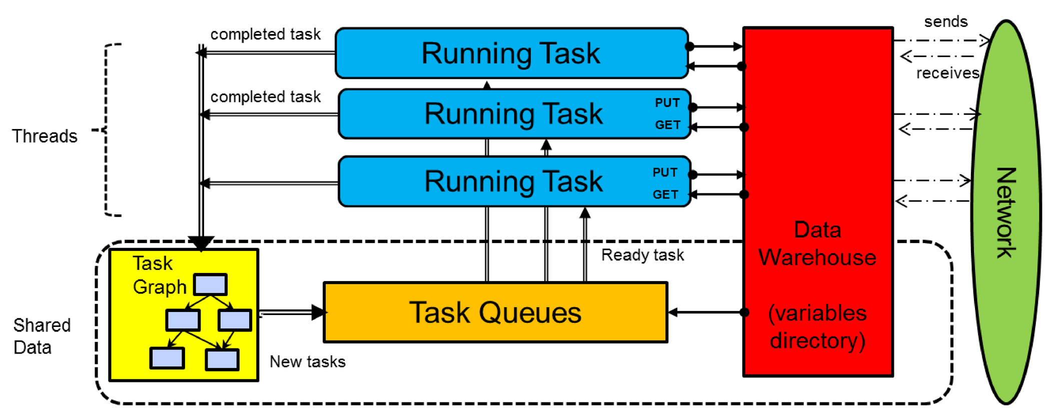
Hybrid MPI/threads approach: Re-design Runtime System

Runtime System

Uintah uses directed acyclic graph (DAG) based asynchronous-task work queue model.



- Simulation component defines computational tasks on a generic grid patch.
- Task defines the variables that are required and computed on this generic patch and its ghost cells.
- Framework creates task instances on the patches of a continually adapting grid and then maps on to the parallel machine.
- Tasks on local patches are compiled to form a DAG(Task Graph).
- Uintah scheduler uses the task graph to determine the order of execution and perform MPI.
- Data warehouse is dictionary-based data structure which manages all task variables and MPI buffers.



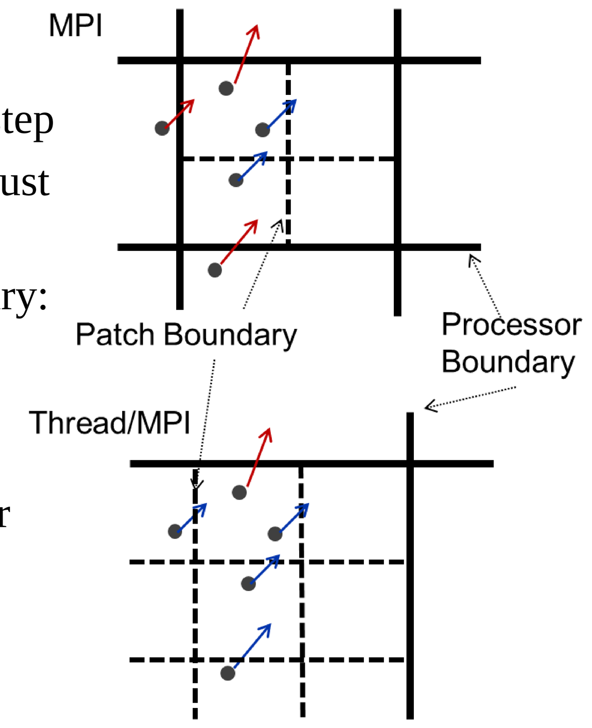
Hybrid MPI/threads Task Scheduler

- De-centralized model (Fully distributed)
- All threads directly pull tasks from task queue
- All threads process MPI sends/receives
- Thread-safe data warehouse

Improvements

Reducing Particle Relocation Communication Costs

- Move particles to new patches after each timestep
- Cross patch boundary: just re-indexing
- Cross processor boundary: MPI scatter record (expensive!)
- Thread/MPI: Fewer particles cross processor boundary



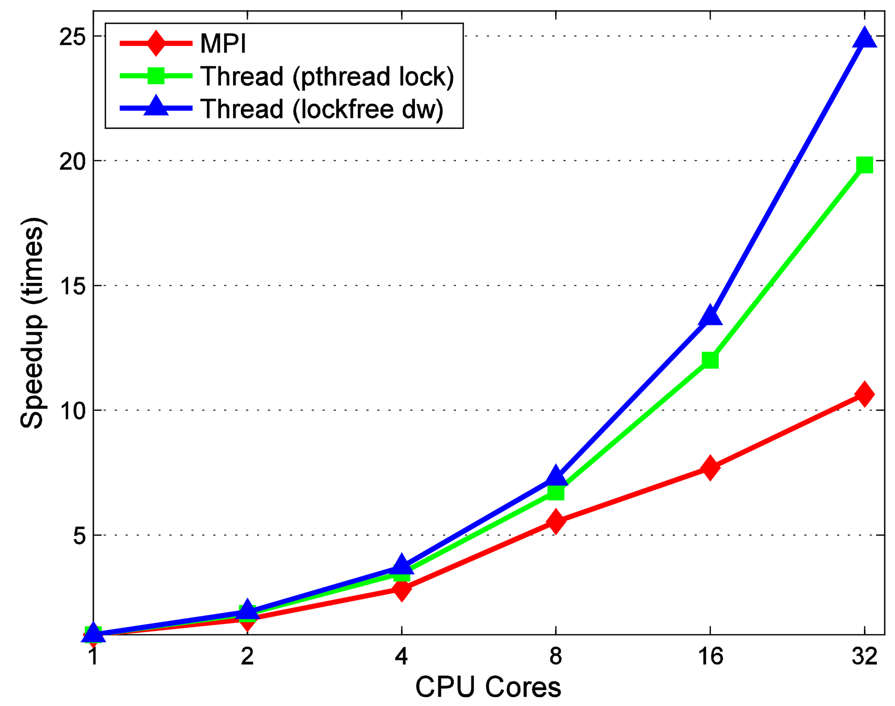
Load Balancing Improvements

- Enable work stealing: all tasks in the same node can be executed by any idle cores on that node
- Larger workload region: more patches on each node, easier to make them even
- The average load imbalance value was reduced from 60% to 25%

Using Lock-free Data Structures

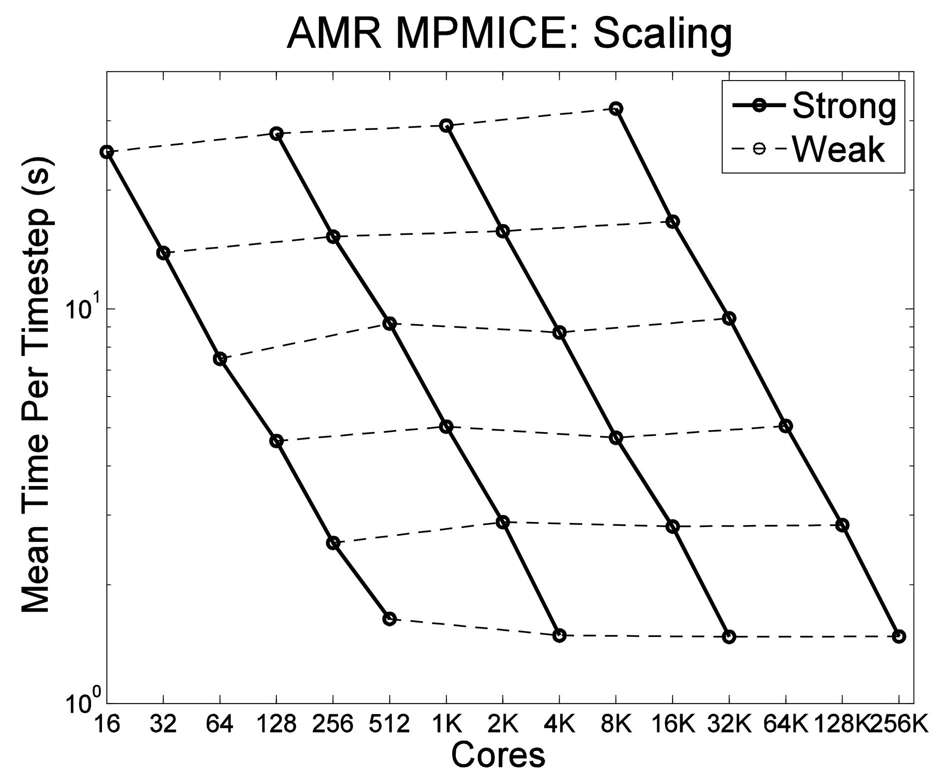
- Overhead of de-centralized scheduler: pthread read/write locking cost on shared data
- Using atomic instruction set
- Variable reference counting: *fetch_and_add*, *fetch_and_sub*
- Redesign data warehouse variable container
- Allow multiple threads update without waiting
- Add variable: *compare_and_swap*
- Reduce variable: *test_and_set*

Speed up 2.4X vs MPI only

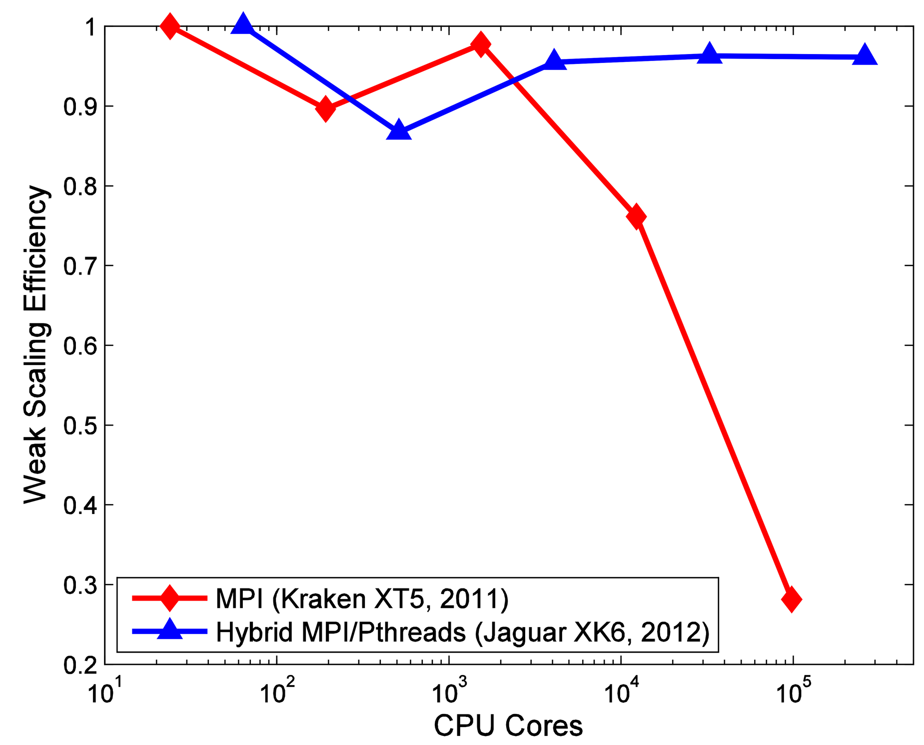


Results

With fully distributed hybrid MPI/thread scheduler and lock-free data warehouse, Uintah can successfully scale up to 256K cores with 95% weak scaling efficiency and 68% strong scaling efficiency.



Weak scaling efficiency comparison with new hybrid scheduler and old MPI only scheduler.



References

1. Qingyu Meng, Justin Luitjens and Martin Berzins. "Dynamic task scheduling for the Uintah framework." In Proceedings of Many-Task Computing on Grids and Supercomputers (MTAGS), 2010 IEEE
2. Qingyu Meng, Martin Berzins, and John Schmidt. "Using hybrid parallelism to improve memory use in the Uintah framework." In Proceedings of the 2011 TeraGrid Conference, 2011 ACM
3. Qingyu Meng and Martin Berzins. "Scalable Large-scale Fluid-structure Interaction Solvers in the Uintah Framework via Hybrid Task-based Parallelism Algorithms." Submitted to Concurrency and Computation: Practice and Experience.

Acknowledgement

This work was supported by the National Science Foundation under subcontracts No. OCI0721659, the NSF OCI PetaApps program, through award OCI0905068 and by DOE INCITE award CMB015 for time on Jaguar.