Online Analytics for Complex Simulation Workflows using DataSpaces

Tong Jin, Fan Zhang, Qian Sun, Hoang Bui, Manish Parashar
Rutgers University

Hongfeng Yu
University of Nebraska

Scott Klasky, Norbert Podhorszki, Hasan Abassi
Oak Ridge National Laboratory

Motivation

- Advanced coupled simulation workflows running at extreme scale generate large amounts of data that must be managed and analyzed to get insights
- Online data analysis approaches based on in-memory data staging and in-situ/in-transit processing becomes promising
- Simulations based on dynamic formulations such as Adaptive Mesh Refinement (AMR) present data management challenges for online analytics
  - Dynamically changing volume of data
  - Imbalanced data distribution
  - Heterogeneous resource requirements

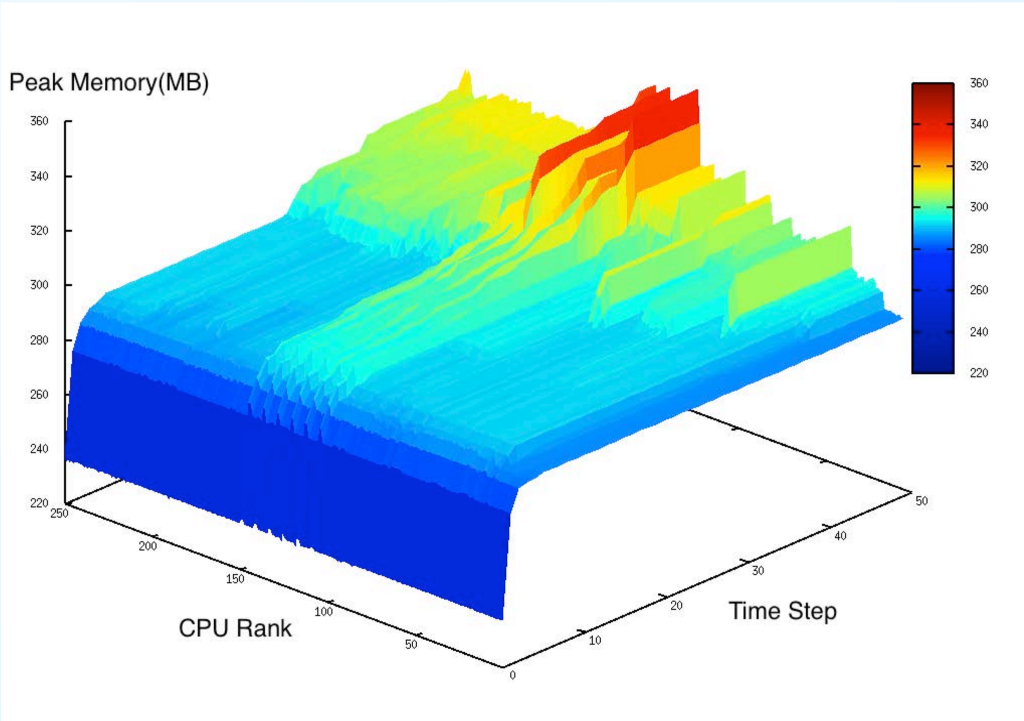

Figure. Distribution of peak memory consumption for a 3D AMR-based Polytropic Gas simulation using the Chombo library. Overall 4K CPU cores over 50 time steps.

Cross-layer Adaptations for Dynamic Data Management

- Problem: dynamic runtime behaviors of AMR-based simulation increase the complexity of managing staging resources and scheduling in-situ/in-transit data processing
- Objective: manage online data analytics using cross-layer adaptations that respond at runtime to the dynamic data management and processing requirements

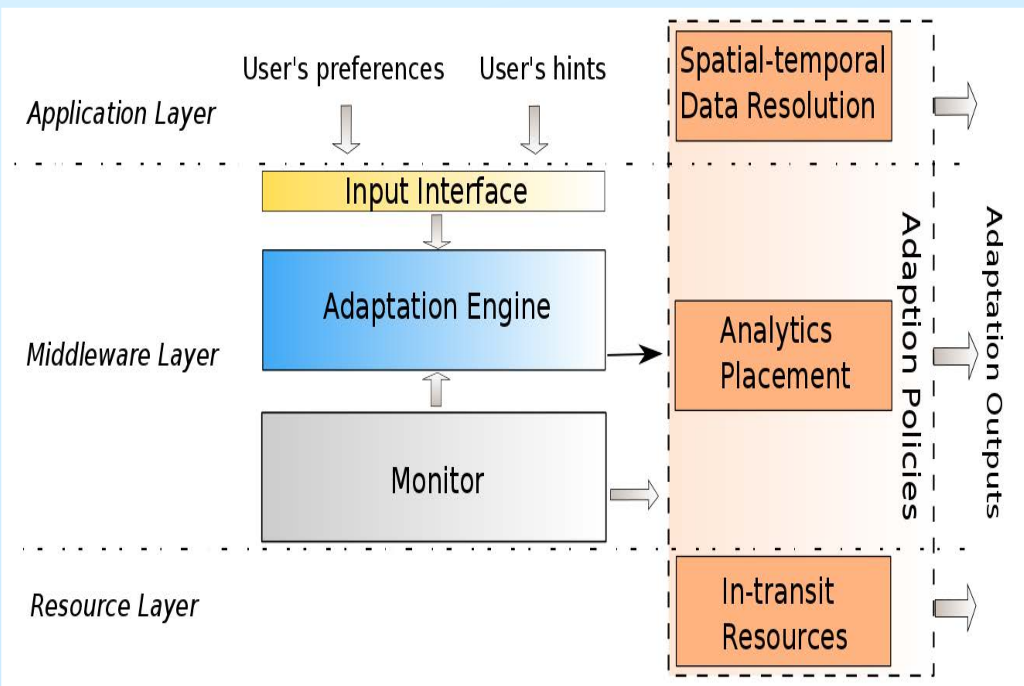

Figure. A conceptual architecture for realizing runtime adaptations for in-situ/in-transit implementations of coupled simulation workflows.

- Runtime adaptation can be performed at three different layers
  - **Application layer**: Adaptive spatial data resolution
  - **Middleware layer**: Dynamic in-situ/in-transit placement and scheduling
  - **Resource layer**: Dynamic allocation of in-transit resources
- Dynamic cross-layer adaptation employs **coordinated approach** to combine adaptations in a cross-layer manner to further optimize the end-to-end performance
- Dynamic cross-layer adaptation consists of three main components
  - **Monitor**: captures runtime status information at different layers
  - **Adaptation engine**: selects and executes appropriate adaptations based on user preference and hints
  - **Adaptation policies**: specify which adaptation mechanisms should be executed based on user inputs and the operational state

Experiments Setup

- Platform: ORNL Titan Cray XK7 system
- Application
  - Chombo-based AMR simulations
- Analysis
  - Visualization: marching cubes algorithm, the de facto standard isosurface extraction algorithm

Application Layer Adaptation

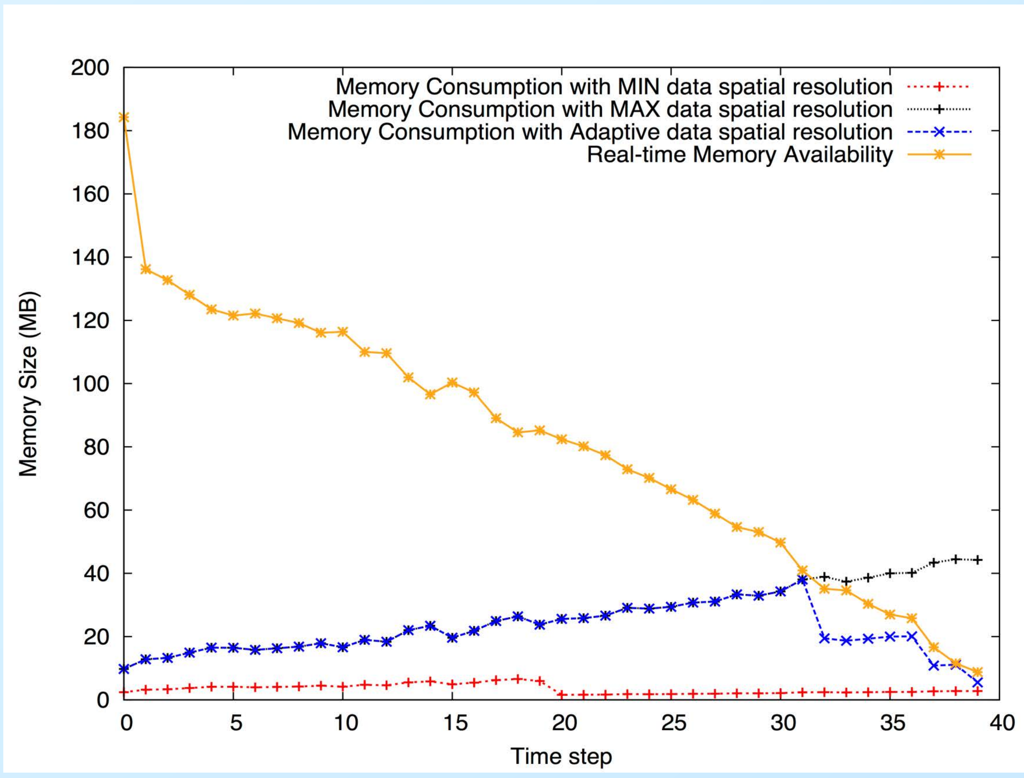

Figure. User-defined down-sampling based on runtime memory availability. At the 31st timestep, spatial data resolution is reduced, then at the 40th time step, it reaches the minimal value.

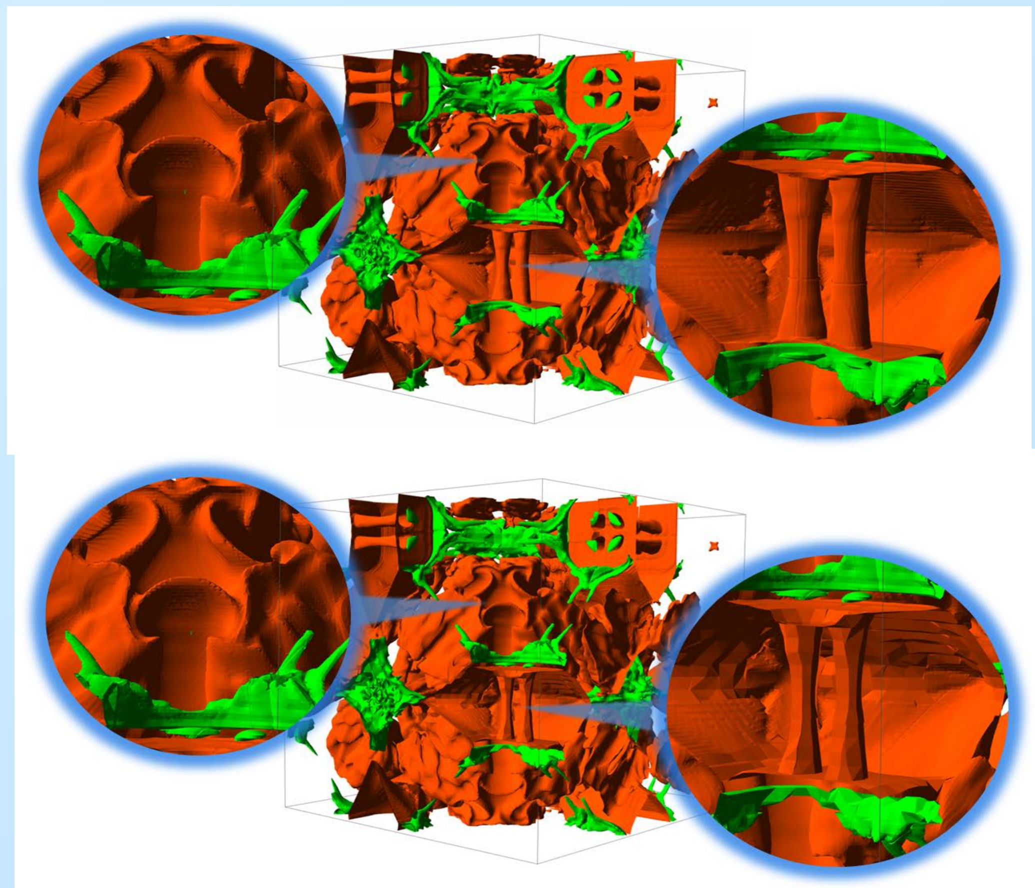

Figure. Entropy based data down-sampling. Data is automatically translated from full resolution (top) to the reduced resolution (bottom) to meet the limited memory availability based on the contained internal information

Resource Layer Adaptation

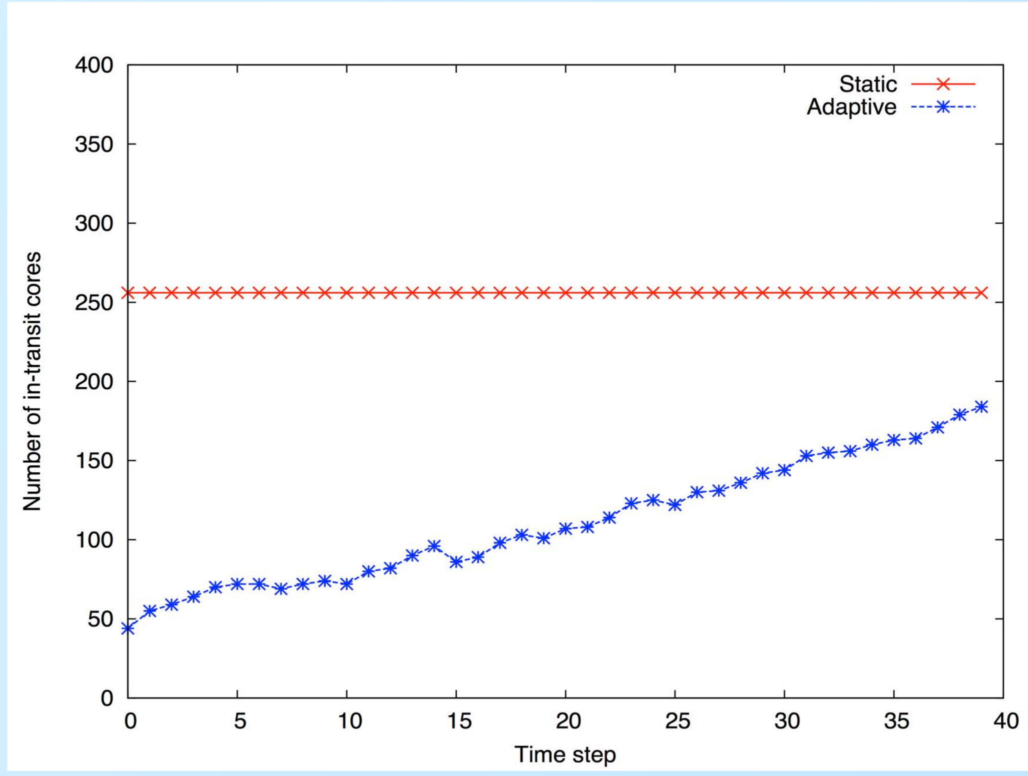

*Figure. Comparison of in-transit cores usage between static resource allocation and adaptive resource allocation. utilization efficiency of static allocation: **54.57%** utilization efficiency of adaptive allocation: **87.11%**.*

Cross-layer Adaptation

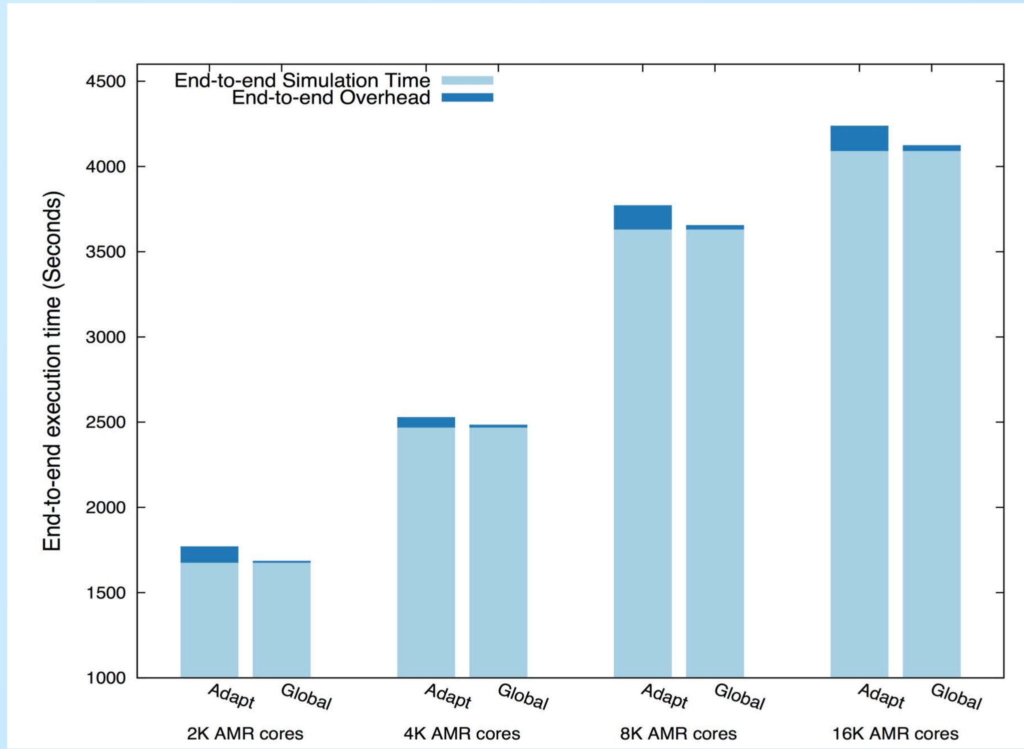

*Figure. Compared with adaptive placement, end-to-end overhead is decreased by: **52.16%, 84.22%, 97.84%, 88.87%***

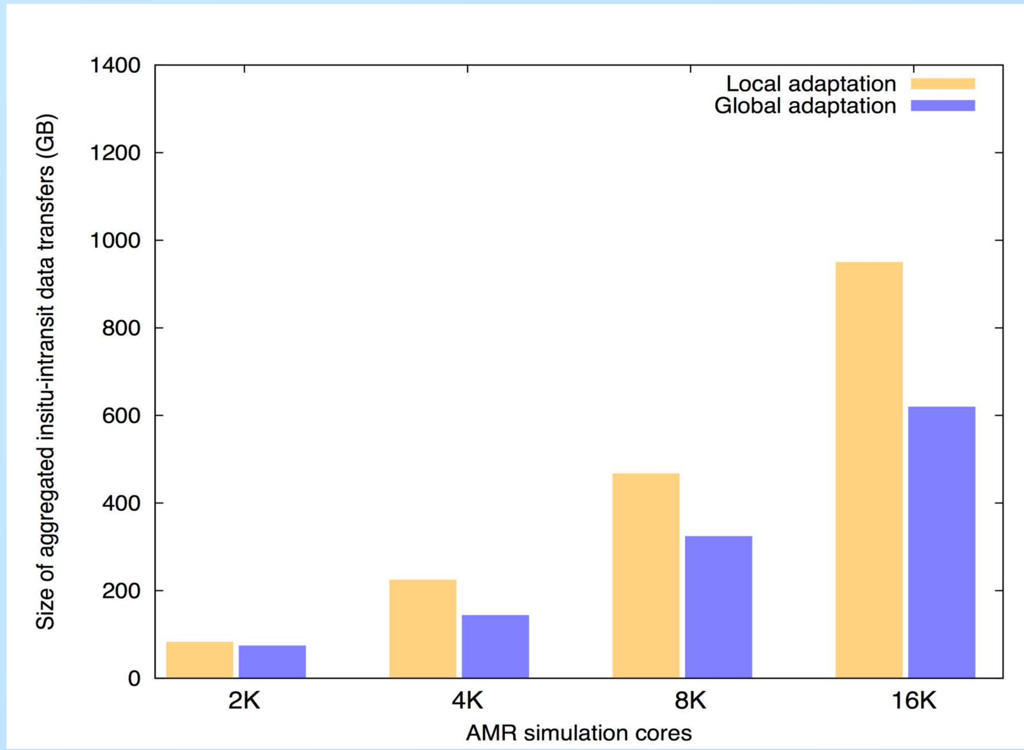

*Figure. Compared with adaptive placement, data movement is reduced by: **45.93%, 17.25%, 5.76%, 32.41%***

Conclusions

- Manage dynamic data processing requirements at extreme scales using coordinated algorithm, middleware and resource layer adaptations
- Accelerated the data-to-insights process by up to 75% for a large-scale AMR-based simulation-analytic workflow
- Reduced overall data movement between the AMR-based simulation and in-situ analytics by 45%