Facilitating the Scalability of ParSplice for Exascale Testbeds

Vinay B. Ramakrishnaiah1,*, Jonas L. Landgesell2, Ying Zhou3, Iris Linck4, Mouad Rami5, Joshua Bevan6, Danny Perez7, Louis J. Vernon7, Thomas D. Swinburne7, Robert S. Pavel7, and Christoph Junghans7

1UWYO, Laramie, WY, USA, 2Univ. of Stuttgart, Stuttgart, Germany, 3Loughborough University, Leicestershire, UK, 4CU Denver, Denver, CO, USA, 5École des ponts ParisTech, Champs-sur-Marne, France, 6UIUC, Champaign, IL, USA, 7LANL, Los Alamos, NM, USA.

Introduction

Parallel trajectory splicing, or ParSplice, is an attempt to solve the enduring challenge of simulating the evolution of complex atomistic systems over long time scales.

- Conventional molecular dynamics (MD) suffer from time scale limitations.
- Typical simulations can only be performed for durations on the order of nanoseconds.
- Hinders physical insights.
- Alleviated using accelerated-MD (AMD) methods.
- ParSplice aims at improving the performance of AMD methods for systems with heterogeneous distributions of barriers.
- Parallelize the generation of long trajectories in a parallel fashion.
- Employ speculative execution strategy.

The preliminary results of our attempts to enhance the scalability of ParSplice are presented in this poster.

Parallel Trajectory Splicing

- Conventional trajectory can be decomposed into segments.
- ParSplice uses this property to concurrently generate segments.
- The segments are spliced together to form a trajectory.
- Current implementation:
  - Producers complete requests for segments
  - Splicer uses Markov chain based predictor to preemptively schedule production of segments
  - Segments are stored in a database.

Motivation

- Large number of atoms can be simulated, but the temporal reach of MD is limited. The performance of parsplice can be further improved by:
  - Efficient prediction of the MD trajectory
  - Large scale MD segment generation

Improving Predictor Efficiency

- The predictor builds a Markov chain based on the previously visited states by the MD segments and performs a Kinetic Monte Carlo (KMC) analysis to predict the next probable state of the trajectory.
- Physics can be accelerated using elevated temperature.
- Assign a fraction of workers to perform ParSplice runs at an elevated temperature.
- Update KMC predictor using elevated temperature runs.

Incorporate Bayesian estimator that takes the inherent model uncertainty into account (Fig. 3).

MPI parallelization

- Most functions are asynchronous
- The splicer, databases, and work manager are executed in parallel process
- Work manager spawns parallel workers
- Good weak scaling, but relatively poor strong scaling

Exploting Latent Parallelism

- Improved KMC predictor using message-passing + multi-threading (Fig. 6)

Conclusions

A two-pronged approach of using heterogeneous architectures, and improving the efficiency of the predictor in ParSplice was explored and currently we are investigating:

- Use of different many-core architectures like Intel Knights Landing (KNL).
- Optimized dynamic load balancing between different architectures.
- Issue of inherent uncertainty in the prediction model, as the current predictor only takes into account the previous observations to formulate the problem.