Atomistic adaptive ensemble calculations of eutectics of molten salt mixtures

Sai Jayaraman

Massachusetts Institute of Technology

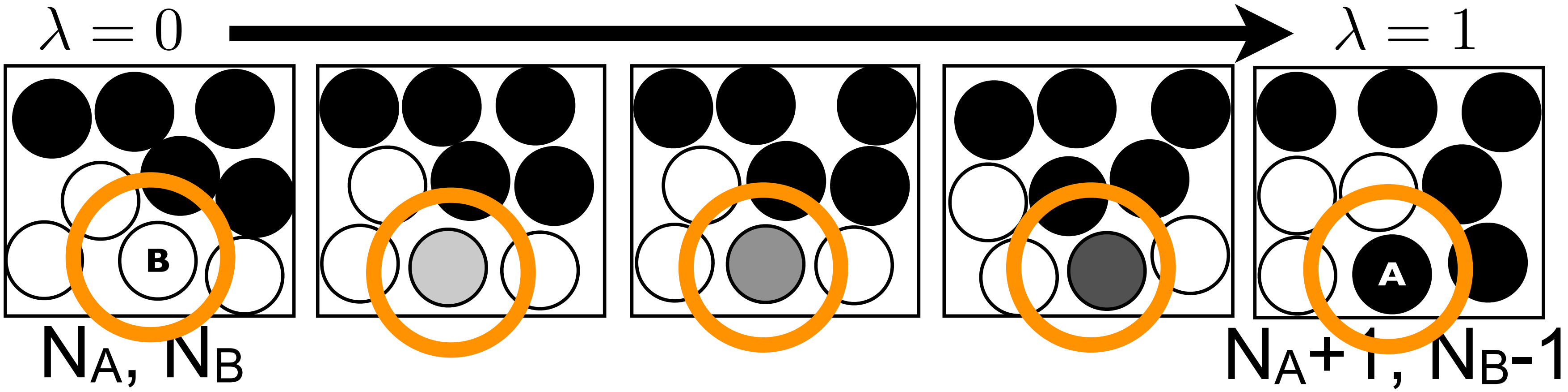
(formerly at Sandia National Laboratories)

A) Project Overview

INCITE project: **Atomistic adaptive ensemble calculations of eutectics of molten salt mixtures**

- Stated Goals
  - Study of molten salts as candidates for heat transfer and storage media in solar power plants
  - Enable fast exploration of molten salt mixture composition space for rapid discovery of new HTF candidates from MD simulations.
- The participants, description of team
  - Sai Jayaraman, Aidan Thompson, Steve Plimpton (SNL), Anatole von Lilienfeld (ANL)
- Previously demonstrated methodology on 2 and 3-component mixtures. Goal is to extend to higher dimensional mixtures, and overcome curse of dimensionality.

B) Science Lesson



- Uses “alchemical changes” to compute free energy of mixing for liquids.
- Thermodynamic integration at different compositions for each mixture – Each composition point ~ 10 “lambda” points.
- As no. of components in mixture increases, number of simulations required increases – Higher dimensional mixtures cannot be explored rapidly.
- Proposed an adaptive scheme in the composition and “lambda” dimensions to overcome this.

C+D) Parallel Programming Model and Code

- LAMMPS – Massively parallel molecular dynamics package written in C++.
- Runs efficiently on wide variety of clusters and architectures. (Intrepid BG/P for this application)
- Electrostatics handled using particle-particle-particle mesh (requires FFTW)
- Adaptive mesh implementation is external to LAMMPS and will not break its inherent massive parallelism.

E) I/O Patterns and Strategy

- I/O: initial configuration only major input, while snapshot-writing is major output. Tuning frequency of writes reduces I/O time.
- Each composition requires an input snapshot. Each lambda point writes to its own log file infrequently, so computation dominates.
- Input: trivial. Output ~ 1MB file per partition and ~100 partitions per run.
- Binary restart files snapshots written every 1000 timesteps.
- Native LAMMPS I/O used.

F) Visualization and Analysis

- How do you explore the data generated?
  - Python post-processing scripts which use scipy handle all ensemble averaging, integration, interpolation, etc. Plots generated using matplotlib and gnuplot, and publication quality surface plots generated using MATLAB.
- Current status and future plans for your viz and analysis
  - No significant changes planned to post-processing scripts.

G) Performance

- What do you believe is your current bottleneck to better performance/scaling?
  - Domain decomposition in LAMMPS achieves high performance. System size dictates upper bound on scaling per ensemble.
  - Overall, embarrassingly parallel scheme invests too much time on computing free energies of insignificant compositions.
- Future plans for improving performance
  - Adaptive mesh technique should overcome inefficient scheme.

H) Tools

- How do you debug your code?
  - Checkpoints, writes to standard output
- What other tools do you use?
  - Valgrind to catch memory errors
- Current status and future plans for improved tool integration and support
  - Very modular structure of LAMMPS makes it easy to track down bugs.

J) Roadmap

- Where will your science take you over the next 2 years?
  - Should be the first simulation of high-dimensional molten salt mixtures.
  - Enable fast screening across composition and component space for locating a low-melting eutectic.
- What do you hope to learn / discover?
  - Design a new , lower melting HTF which has not yet been discovered by experimentalists.
  - Find a rule of thumb for the design of a low-melting mixture.