SAND2019-8772C

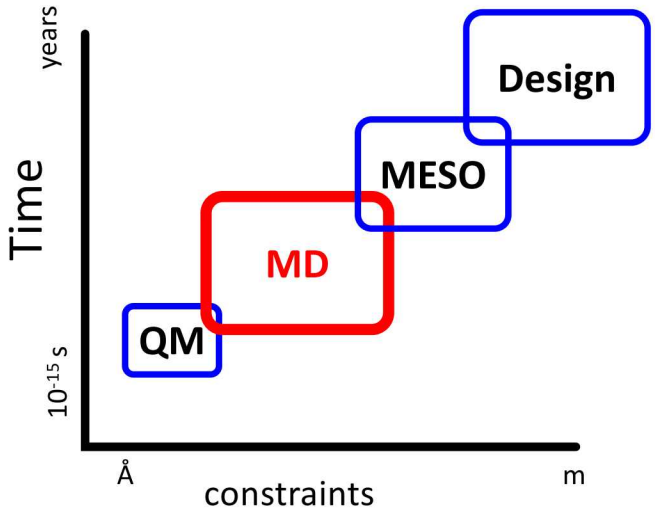
Sandia National Laboratories

Development of Validated High-accuracy Interatomic Potentials Using DAKOTA for Large-Scale Atomistic Simulation of Materials

Aidan Thompson
Center for Computing Research,
Sandia National Laboratories,
Albuquerque, New Mexico

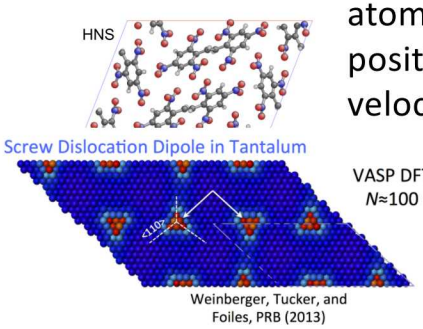
Dakota Symposium, USNCCM, Austin TX
(7/29/2019)

What is Molecular Dynamics Simulation?

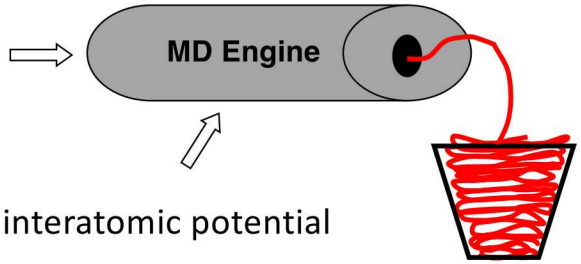


constraints

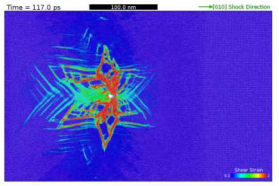
Distance



atoms,
positions,
velocities

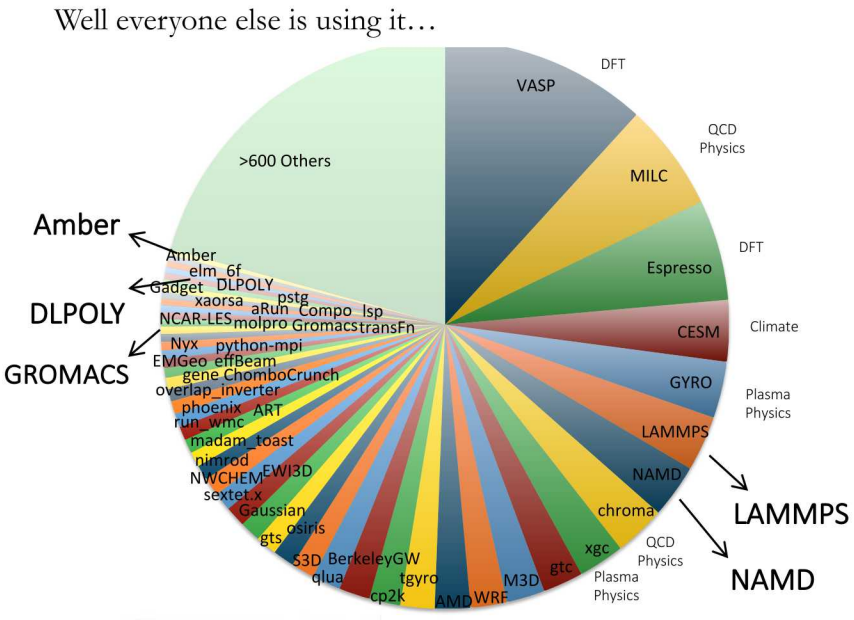


interatomic potential



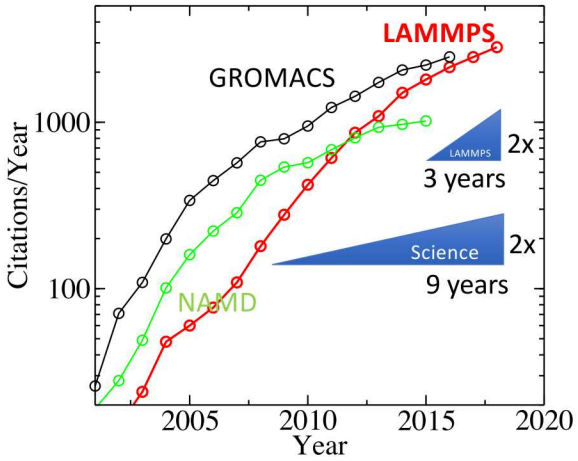
Positions, velocities and forces at many later times

But is it useful?



2014 Top Application Codes at NERSC (Hopper and Edison)

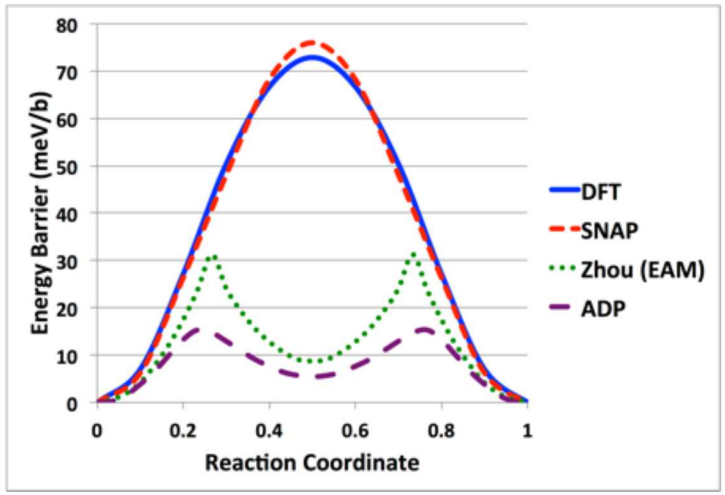
"Global scientific output doubles every nine years or 8-9% per year"
Nature, May 2014



Papers citing LAMMPS, NAMD, GROMACS
Measured using Web of Science citation reports

SNAP Tantalum

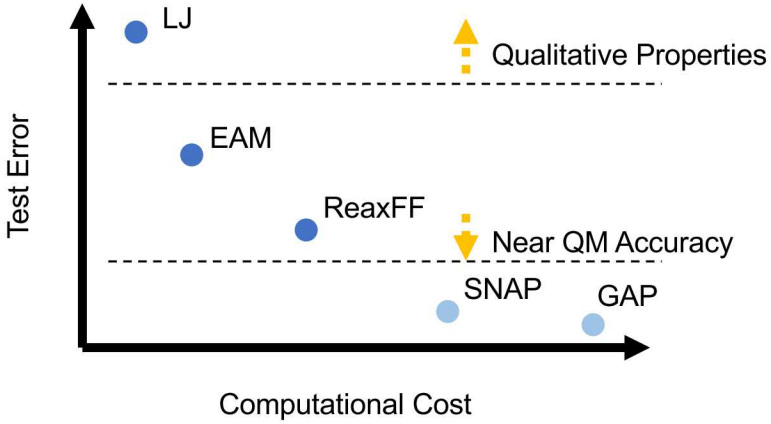
- Training data:
  - Energy, force, stress
  - ~5,000 data points
  - Deformed crystals phases
  - Generalized stacking faults
  - Surfaces
  - Liquid
- Peierls barrier is the activation energy to move a screw dislocation
- Not included in training data
- SNAP potential agrees well with DFT calculations



SNAP Potentials have been developed at Sandia and elsewhere for a variety of material systems:
Sandia: Ta, InP, W/Be/H
UCSD: Co, Li, Mo, Cu, Ni, Si, Ge

A. P. Thompson , L.P. Swiler, C.R. Trott, S.M. Foiles, and G.J. Tucker, *J. Comp. Phys.,* **285** 316 (2015) .

Evolution of ML Potentials

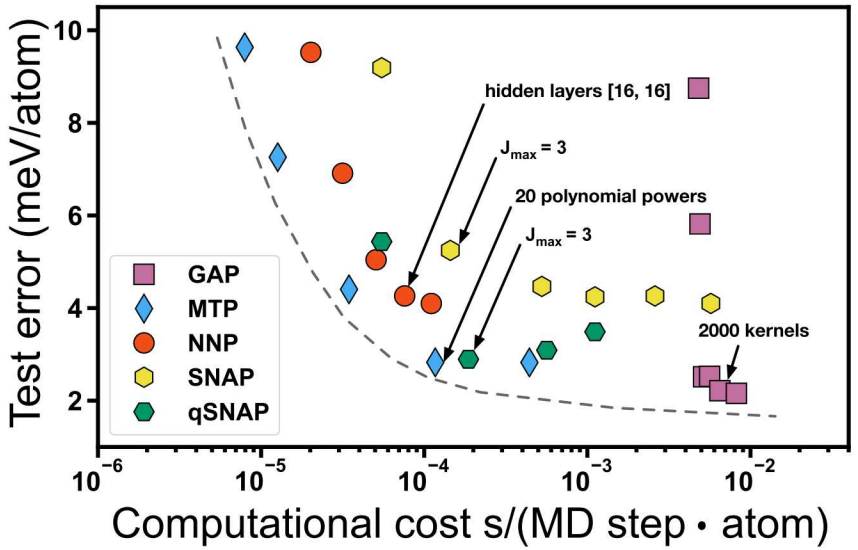


Shyue Ping Ong (UCSD), with:

Csanyi (2010)	**GAP**
Shapeev (2015)	**MTP**
Behler(2007)	**NNP**
Thompson (2015)	**SNAP**
Wood (2018)	**qSNAP**

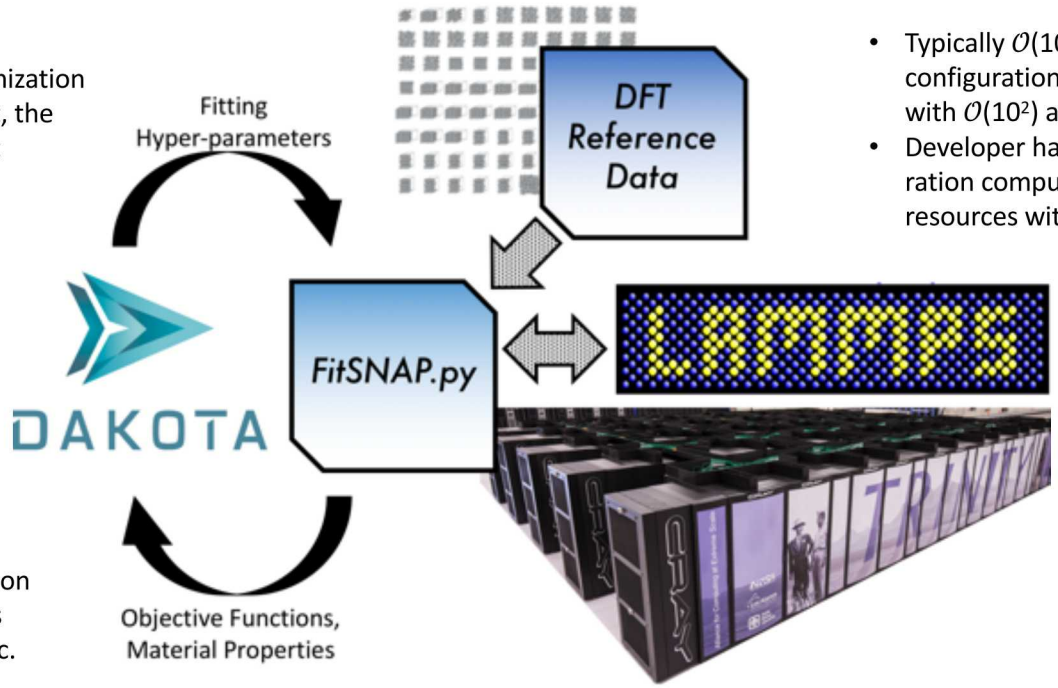
Submitted J.Chem. Theory.Comp., 2019

https://arxiv.org/abs/1906.08888
https://github.com/materialsvirtuallab/mlearn



Automated Generation of SNAP Interatomic Potentials

- Even if the optimization routine is robust, the process still isn't transparent.
- The importance on each objective is part of the magic.

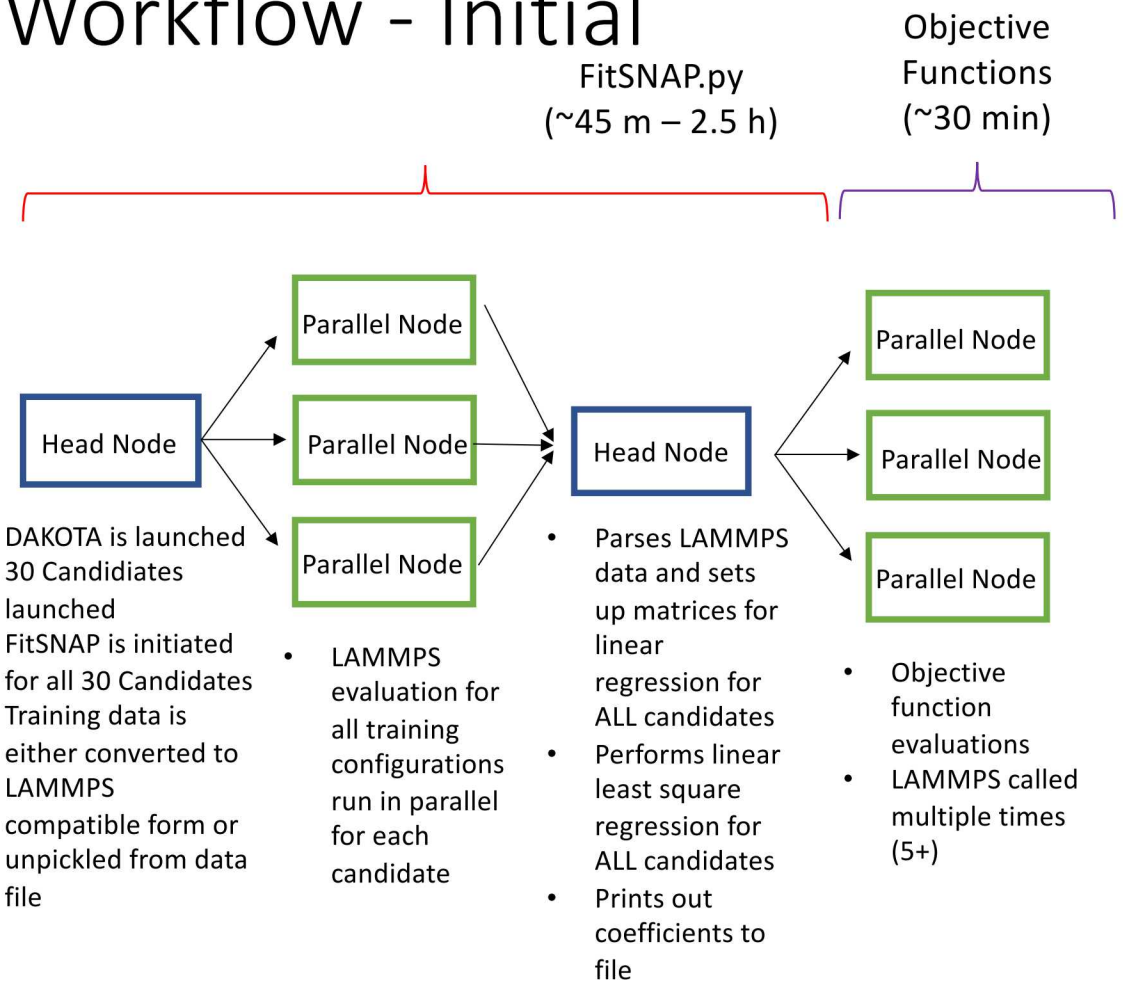


- Typically $\mathcal{O}(10^4)$ configurations, each with $\mathcal{O}(10^2)$ atoms
- Developer has to ration compute resources with DFT

W-Be-H DAKOTA Parameters/Data

- Nodes: 31
  - Limited due to memory issues on head node
- Training set size: ~40,000 configurations
  - Large dataset has lead to memory issues with the way DAKOTA is set up
- GA Parameters:
  - max_iterations = 270
  - max_function_evaluations = 25000
  - population_size = 200
  - Variables: 5 (hyperparameter fit), 34 (groupweight fit)
  - Objective functions: 6 (hyperparameter fit), 9 (groupweight fit)
    - Evaluate things like: energy/force errors, defect formation energies for W, W-Be, and W-H, H2 and H3 binding energies, H/H2 adsorption energy (important parameters for fusion material modeling)
- Evaluation time per candidiate: 0.75-2.5 hours
  - Shorter time during groupweight fitting since we use previous Data and DumpSnap files so we don't have to convert JSON to LAMMPS data files or run LAMMPS again
  - Longer time for hyper parameter fitting is only for half the dataset but shorter time for groupweight fitting is for entire dataset
- Number of Candidates needed
  - Hyperparameters: ~100s
    - Cost: ~500 node-hours (200 candidates, 2.5 hours)
  - Groupweights: ~15,000-20,000
    - Cost: ~20,000 node-hours (20,000 candidiates, 1 hour)

Workflow - Initial

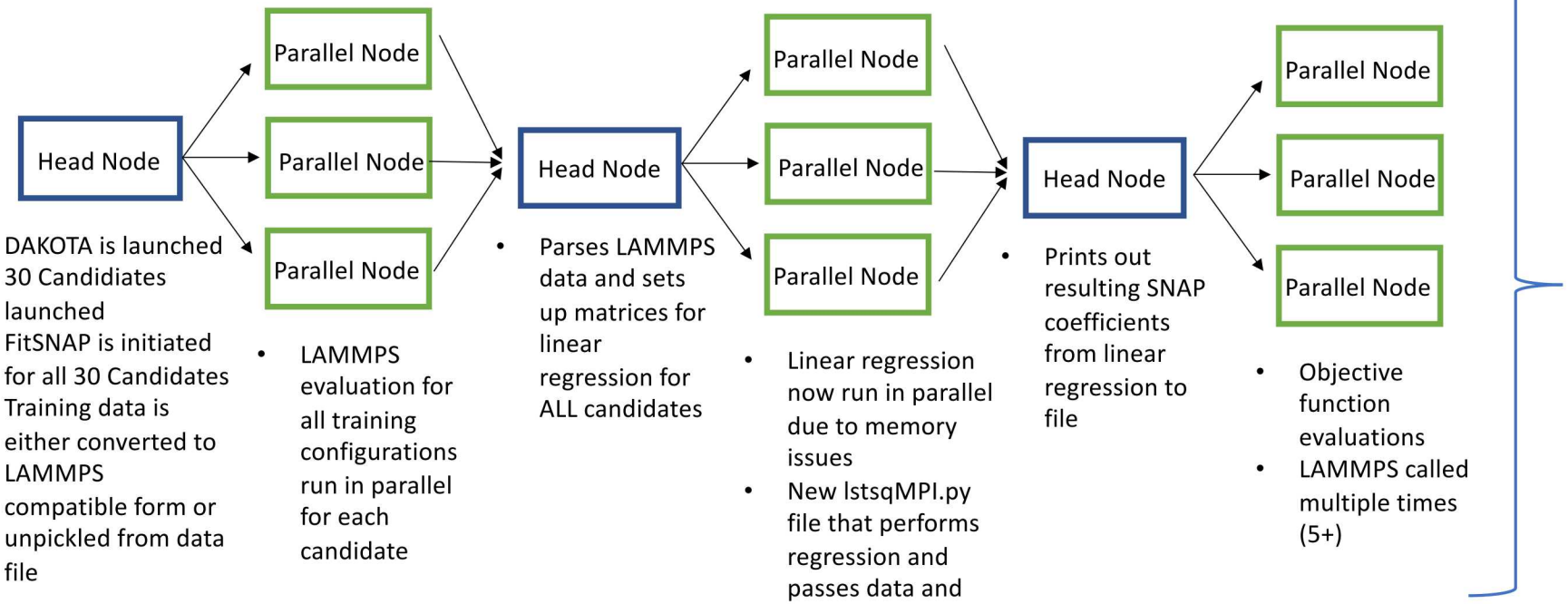


- Initial workflow for generating candidate potentials
- Works well for initial W-Be training data
  - Could run at least 50 concurrent candidates
- Increases in training data from additional hydrogen configurations
  - Training set now ~40,000 configurations
- Head node would run out of memory for more than ~15 concurrent candidates
  - High memory usage from linear least squares on head node for all candidates
- Modification needed to also run linear least squares in parallel

Workflow - Modified

FitSNAP.py
(~45 m – 2.5 h)

Objective Functions
(~30 min)



- DAKOTA is launched
- 30 Candidiates launched
- FitSNAP is initiated for all 30 Candidates
- Training data is either converted to LAMMPS compatible form or unpickled from data file

- LAMMPS evaluation for all training configurations run in parallel for each candidate

- Parses LAMMPS data and sets up matrices for linear regression for ALL candidates

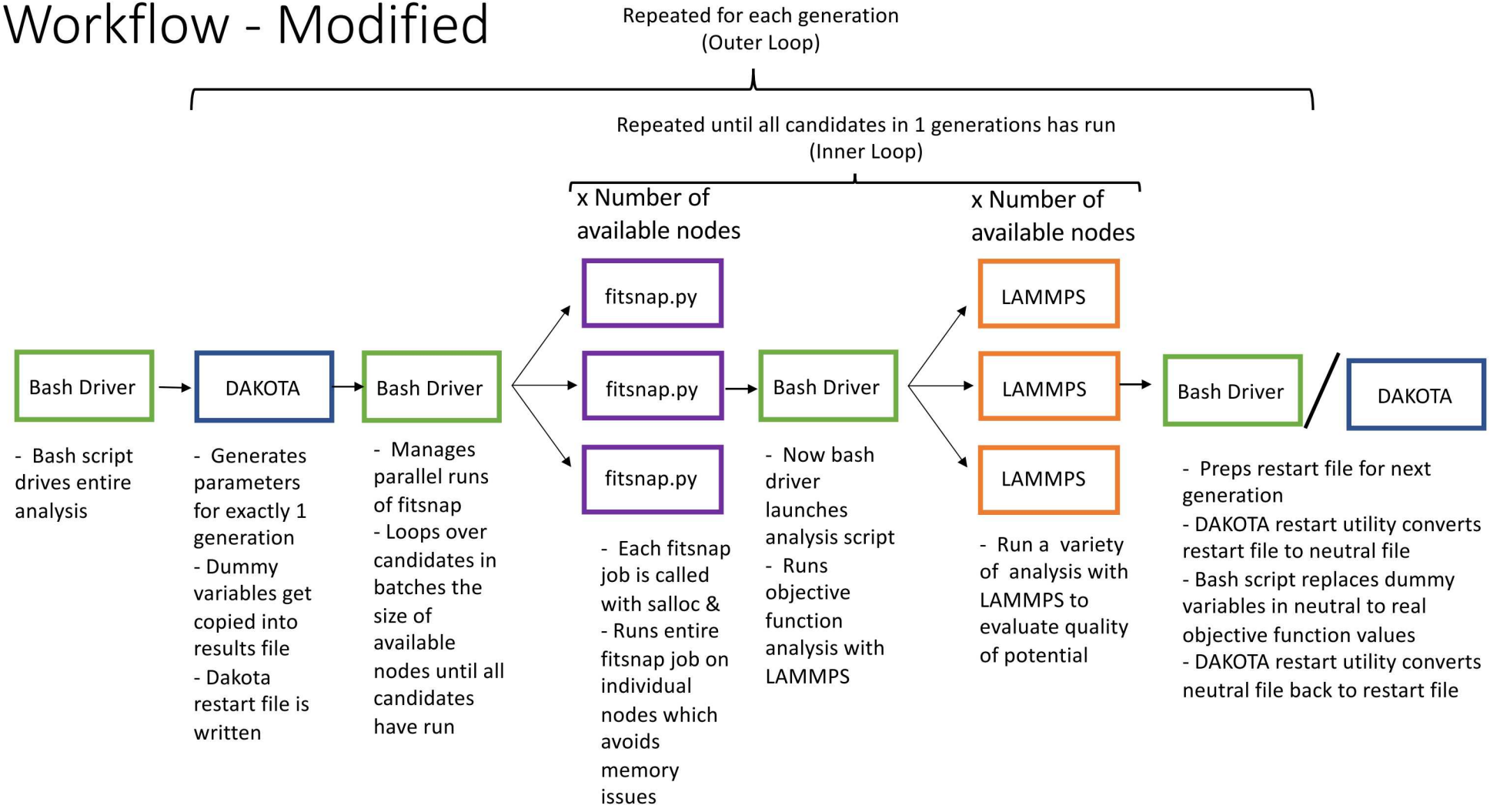
- Linear regression now run in parallel due to memory issues
- New lstsqMPI.py file that performs regression and passes data and can be called in parallel

- Prints out resulting SNAP coefficients from linear regression to file

- Objective function evaluations
- LAMMPS called multiple times (5+)

Can now run more candidates at once with new method (15 vs. 30)

Workflow - Modified



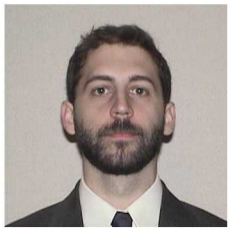
Conclusions

- DAKOTA is a powerful toolbox for SNAP hyperparameter optimization
- The SNAP multi-step workflow is relatively complex
- Increase in training data size has exposed non-parallel bottlenecks
- Solution 1: consecutive MPI launches within DAKOTA analysis driver
- Solution 2: Combine single-generation DAKOTA stage with separate analysis stage

Acknowledgements

Adam Stephens

Mitch Wood



Mary Alice Cusentino