Learning Everywhere: Pervasive Machine Learning for Effective High-Performance Computing

Geoffrey Fox, … many others …., Shantenu Jha*
Rutgers University and Brookhaven National Lab.

http://radical.rutgers.edu

Outline

- Learning Everywhere: Motivation and Classification
- Molecular Science Examples
  - Adaptive Sampling: Predicting go next in MD (**MLaroundHPC**)
  - Using deep learning approaches for MD trajectory (**MLafterHPC**)
  - Objective Driven Drug Candidate Selection (**MLControlHPC**)
  - Nanoparticles Ionic distribution: ANN regression models (**MLAutoTuned**)
- ML-HPC Reference Architecture
- Learning Everywhere! Open Issues and Challenges
  - Enhance “Effective Performance” Performance Challenges
  - System and Software Challenges

Learning EveryWhere: Classification

- **HPCforML**: Using HPC to execute and enhance ML performance, or using HPC simulations to train ML algorithms (theory guided machine learning), which are then used to understand experimental data or simulations.
- **MLforHPC**: Using ML to enhance HPC applications and systems; Big Data comes from the computation
- *Context: Computational Science Effective consumer of **HPCforML**; innovative producers of **MLforHPC***

HPCforML: Classification

HPCforML can be further subdivided

- **HPCrunsML:** Using HPC to execute ML with high performance
- …
- ...
- **SimulationTrainedML:** Where the simulations are performed to directly train an AI system, rather than the AI system being added to learn a simulation.
    - Train ML algorithms, which are then used to understand experimental data or simulations.

MLforHPC: Classification

MLforHPC: Using ML to enhance HPC applications and systems

- **MLAutoTuning:** Using ML to configure (autotune) ML or HPC simulations.
- **MLafterHPC:** ML analyzing results of HPC as in trajectory analysis and structure identification in biomolecular simulations
- **MLaroundHPC:** Using ML to learn from simulations and produce learned surrogates for the simulations or parts of simulations.
- **MLControl:** Using HPC simulations in control of experiments and in objective driven computational campaigns. Simulation surrogates allow real-time predictions.
- Latter two arguably most important, rewarding (“effective perf”), difficult

MLAutoTuning: Examples

- **MLAutoTuningHPC: Learning Configurations (**classic auto-tuning)
  - Optimizes mix of performance & quality of results
  - Includes initial values, dynamic choices, e.g., block sizes for cache use, variable step sizes in space and time.
  - Also include discrete choices as to the type of solver to be used.
- **MLAutoTuningHPC: Active Learning**
  - Choose the best set of computation defining parameters to achieve some goal, e.g., providing the most efficient training set with defining parameters spread well over the relevant phase space.
- **MLAutoTuningHPC: Learning Model Setups from Observation**
  - Seen when simulation set up as a set of models; parameters to optimize outputs to available empirical data presents one of the greatest challenges in model construction.

MLaroundHPC: Examples

- **MLaroundHPC: Learning Outputs from Inputs**:
  - Simulations performed to directly train an AI system, rather than AI system being added to learn a simulation (includes **SimulationTrainedML)**
- **MLaroundHPC: Learning Simulation Behavior**
  - ML learns behaviour replacing detailed computations by ML **surrogates.**
- **MLaroundHPC: Learning Effective Potentials**
  - Effective potential is analytic, quasi-empirical or quasi-phenomological potential that combines multiple effects into a single potential.
  - Classic Coarse-graining: Effective potential typically defined using physical intuition, e.g., a model specified at a microscopic scale, define **coarse graining** to a different scale with macroscopic entities defined to interact with effective dynamics specified in some fashion such as an effective potential or effective interaction graph

MLaroundHPC: Further Examples

- **MLaroundHPC: Learning Agent Behavior – a Predictor-Corrector approach**
  - At each step optimize the parameters to minimize divergence between simulation and ground truth data. The ground truth here may be in the form of experimental data, or from highly detailed (and expensive) quantum or micro-scale calculations. The time series of parameter adjustments define information missing from the model.. This is an extended **data assimilation** approach.
- **MLaroundHPC: Inference of Missing Model Structure:**
  - In this case we aggregate the Learned Predictor Corrector MLs,
  - Infer unknown model structure from the aggregation of individual learned predictor corrector models. Add inferred mechanisms to the base model structure and repeat the basic predictor-corrector steps.

MLControl: Examples

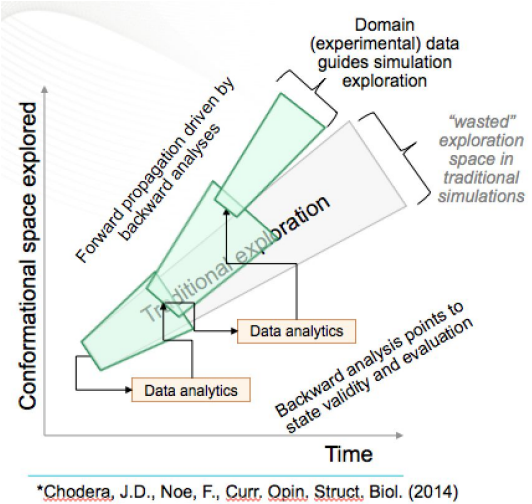
- **MLControl:** Using HPC simulations in control of experiments and in objective driven computational campaigns
- **MLControl: Experiment Control**
  - Using HPC simulations in control of experiments and in objective driven computational campaigns.
  - Simulation surrogates are very valuable to allow real-time predictions. Applied in Material Science and Fusion
- **MLControl: Experiment Design**
  - Challenges is uncertainty in precise model structures and parameters.
  - Model-based design of experiments (MBDOE) assists in the planning of highly effective and efficient experiments. MBDOE with ML assistance identifies the optimal conditions for stimuli and measurements that yield the most information about the system given practical limitations on realistic experiments

Outline

- Learning Everywhere: Motivation and Classification
- Molecular Science Examples
  - Adaptive Sampling: Predicting go next in MD (**MLaroundHPC**)
  - Using deep learning approaches for MD trajectory (**MLafterHPC**)
  - Objective Driven Drug Candidate Selection (**MLControlHPC**)
  - Nanoparticles Ionic distribution: ANN regression models (**MLAutoTuned**)
- ML-HPC Reference Architecture
- Learning Everywhere! Open Issues and Challenges
  - Enhance “Effective Performance” Performance Challenges
  - System and Software Challenges

Case Study: Enhanced Conformational Sampling

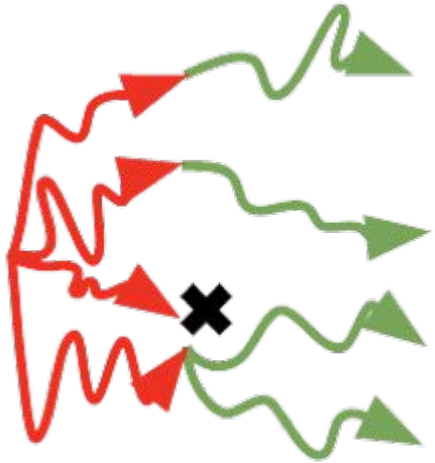
- **Adaptive Sampling**
  - Better, Faster, Greater sampling
- **Better Sampling**
  - Drive systems towards unexplored regions, don't waste time sampling behaviour already observed
- **Faster Sampling**
  - Statistically equivalent parts of conformational space sooner.

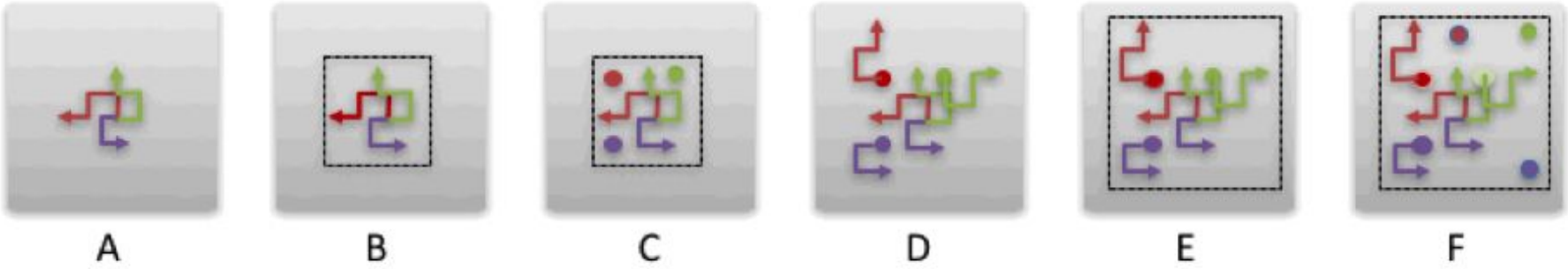


*Chodera, J.D., Noe, F., Curr. Opin. Struct. Biol. (2014)

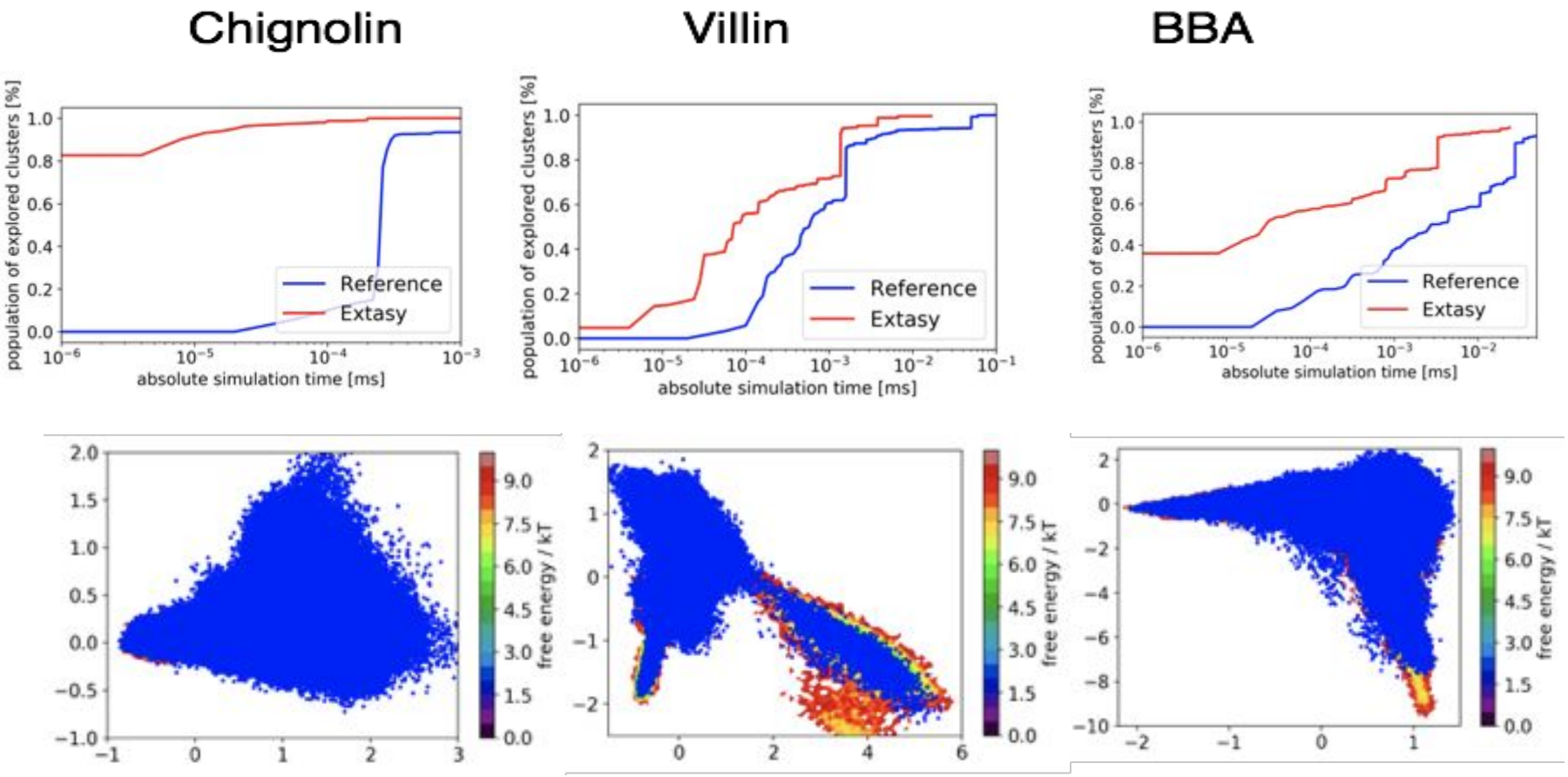
Case Study: Enhanced Conformational Sampling

- **Adaptive Sampling**
  - Better, Faster, Greater sampling
- **Better Sampling**
  - Drive systems towards unexplored regions, don't waste time sampling behaviour already observed
- **Faster Sampling**
  - Statistically equivalent parts of conformational space sooner.



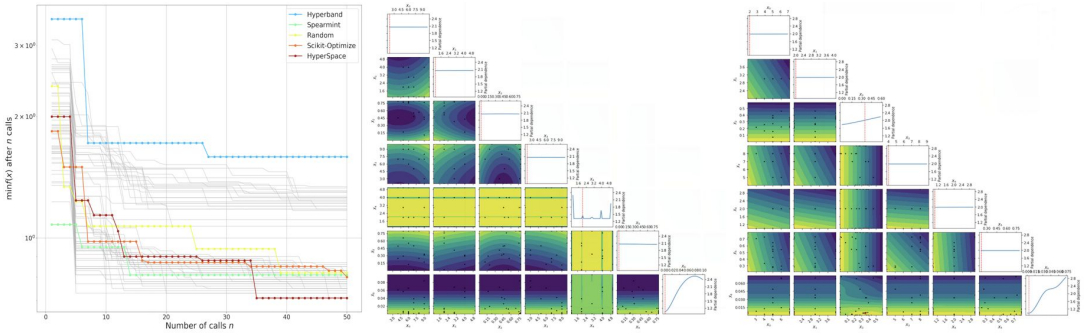


Adaptive Ensemble MD (MLaroundHPC)



Deep Clustering of Protein Folding (MLafterHPC)

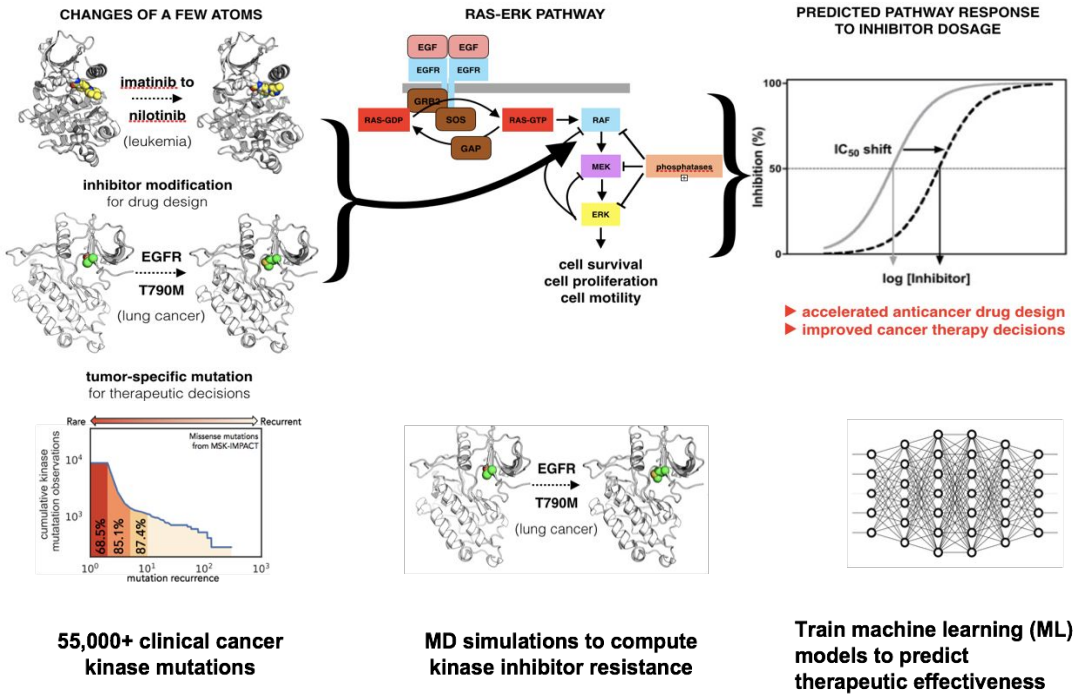
- Using DL to improve MD simulations
  - Deep clustering of protein folding simulations using CVAE (ORNL) and Bayesian Hyperparameter Optimization using RADICAL-Cybertools on Summit
  - Building low dimensional representations of states from simulation trajectories.
    - CVAE can transfer learned features to reveal novel states across simulations
  - HPC Challenge: DL approaches to achieve near real-time training & prediction!



Deep clustering of protein folding simulations, Debsindhu Bhowmik et al, https://doi.org/10.1101/339879

INSPIRE: Integrated (ML-MD) **S**calable **P**rediction of **RE**sistance

- Chemical space of drug design in response to mutations very large. 10K -100K mutations; too large for HPC simulations alone!
- Developed methods that use: (i) simulations to train machine learning (ML) models to predict therapeutic effectiveness; (ii) use ML models to determine which drug candidates to simulate.

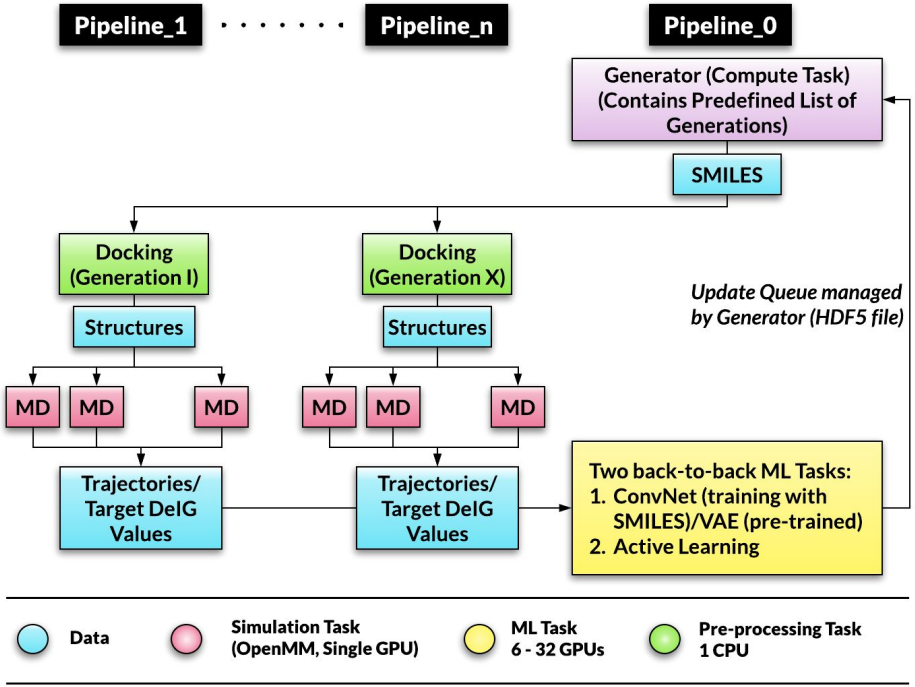


Early Science Project on NSF Frontera. DD Award on Summit.

A collaboration between BNL/Rutgers (Jha), Chicago (Stevens), Memorial Sloan Kettering (Chodera), UCL (Coveney)

INSPIRE: Integrated (ML-MD) **S**calable **P**rediction of **RE**sistance

- Chemical space of drug design in response to mutations very large. 10K -100K mutations; too large for HPC simulations alone!
- Developed methods that use: (i) simulations to train machine learning (ML) models to predict therapeutic effectiveness; (ii) use ML models to determine which drug candidates to simulate.
- **MLControlHPC**

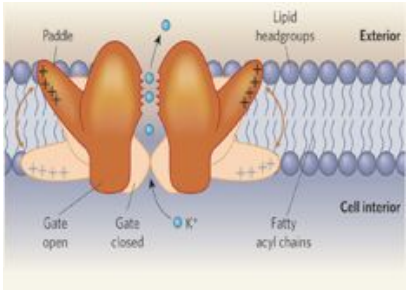


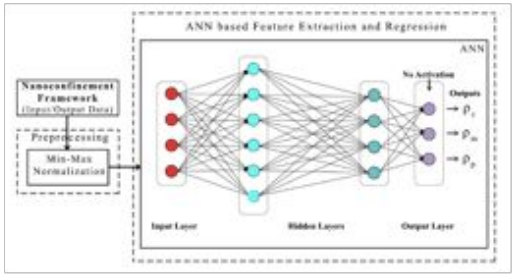
Early Science Project on NSF Frontera. DD Award on Summit.

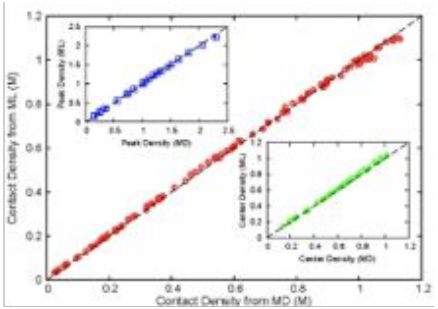
A collaboration between BNL/Rutgers (Jha), Chicago (Stevens), Memorial Sloan Kettering (Chodera), UCL (Coveney)

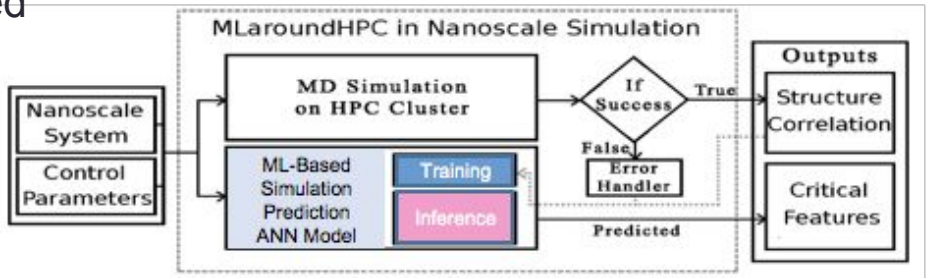
MLAutoTunning and MLaroundHPC: ML for performance enhancement with Surrogates of MD Simulations

- Integration of ANN based regression model for prediction for MD simulations of ions near polarizable nanoparticles
- Predict dynamics of ions for 10 million steps
- Reduced computational time of simulating systems with 1000 of ions and induced charges from 1000 of hours to 10 of hours, yielding a maximum speedup of 3 from MLAutoTuning and a maximum speedup of 600 from the combination of ML and parallel computing.
- ANN based regression model learns desired features of ionic density distribution
- Integration of ANN with simulations allows real time and any time engagement with simulation framework









Effective Performance

T_{seq} is sequential time

T_{train} time for a (parallel) simulation used in training ML

T_{learn} is time per point to run machine learning

T_{lookup} is time to run inference per instance

N_{train} number of training samples

N_{lookup} number of results looked up

N_{train} is 7K to 16K in our work

$$\text{Effective Speedup } S = \frac{T_{seq}(N_{lookup} + N_{train})}{T_{lookup}N_{lookup} + (T_{train} + T_{learn})N_{train}}$$

Becomes T_{seq}/T_{train} if ML not used

Becomes T_{seq}/T_{lookup} (10^5 faster in our case) if inference dominates (will overcome end of Moore's law and win the race to zettascale)

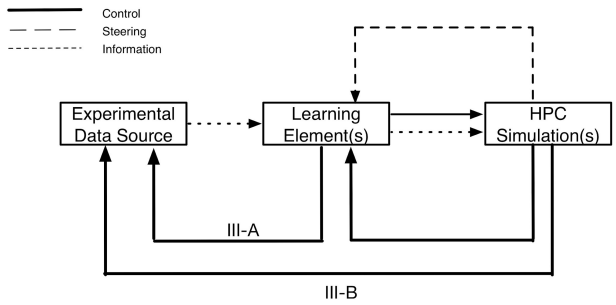
Outline

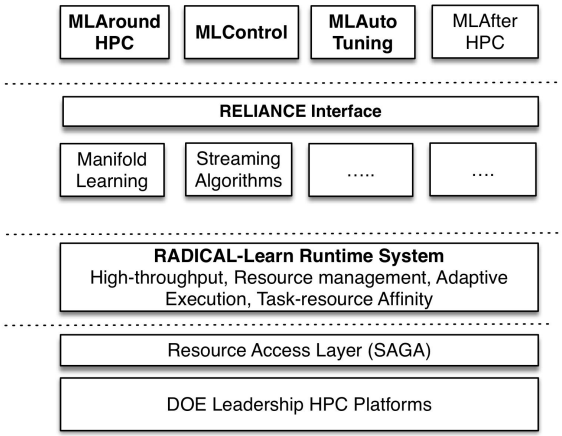
- Learning Everywhere: Motivation and Classification
- Molecular Science Examples
  - Adaptive Sampling: Predicting go next in MD (**MLaroundHPC**)
  - Using deep learning approaches for MD trajectory (**MLafterHPC**)
  - Objective Driven Drug Candidate Selection (**MLControlHPC**)
  - Nanoparticles Ionic distribution: ANN regression models (**MLAutoTuned**)
- ML-HPC Reference Architecture
- Learning Everywhere! Open Issues and Challenges
  - Enhance “Effective Performance” Performance Challenges
  - System and Software Challenges

Learning EveryWhere: Reference Architecture

MLforHPC: For all four scenarios

- Reference Architecture:
  - N_L (L = Learning Element)
  - N_S (S = Simulation Element)
  - N_D (E = Exp. Data Source)
  - N_L / N_S / N_D can be time dependent and so is the coupling between them.
- Coupling:
  - I: E couples to S or L
  - II: L couples to S
  - III : E controlled by L (III-A) or S (III-B)
- Control: Temporal constraints determine scaling considerations





Reference Architecture: Scaling Considerations

- Strong Scaling Considerations:
  - Strong Scaling of individual L: Enabling L to achieve near real-time training and prediction to control or steer S
    - Build low dimensional representation of states from trajectory analysis
  - Strong Scaling of Integrated L + S: Enabling simulation-trained models to determine where to sample in space
    - INSPIRE -- where space / number of drug candidates is very large.
- Weak Scaling Considerations:
  - Weak Scaling of individual L: Many learning models concurrently
    - Multiple surrogates, may the best surrogate win
  - Weak Scaling of Integrated L + S: Multiple instantiations of L and S
    - Model-based design of experiments (MBDOE), objective driven experiments and learning effective potentials

Reference Architecture: Resource Management Considerations

- **Resource Management Considerations:**
  - Must consider streaming data so as to include experimental and observational data
  - General Properties of applications
    - Adaptive: Task graph and plan will change based upon intermediate results and data availability
    - Dynamic: Resource availability and performance is time dependent
    - Heterogeneous workflows: Multiple distinct components (E, L and S), and different instances of each component
  - Resource management and system software challenges are similar to adaptive + streaming workflow!

Open Issues and Challenges

- Which learning methods are most effective?
- New algorithmic approaches based upon “effective learning” ?
- Is there a general multi-scale approach using surrogates (MLaroundHPC) ?
- Advances in Uncertainty Quantification
- What are appropriate system frameworks to implement interaction between E, S and L components?
  - Single reference architecture for all 4 categories?
- Runtime system challenges for balanced execution of real & surrogate models?
  - Workload management, resource management and scheduling
  - Strong and weak scaling challenges
- Application / scenario agnostic definition of Effective Performance
- ….

Thank You!