Enabling Efficient Execution of In Situ Workflows

Tu Mai Anh Do
Information Sciences Institute, University of Southern California

UTK Seminar, September 3rd, 2021
Acknowledgements

This work is funded by NSF contracts #1741040 and #1741057; and DOE contract #DE-SC0012636
Outline

1. Motivations and Background
2. Contributions:
   a. Modeling framework for in situ workflows
   b. Performance indicators to determine performance of workflow ensembles
3. Conclusion
Molecular dynamics (MD) is a simulation model computing the atomic states of a molecular system evolving over time by observing interactions between atoms.

MD serves as a productive method to:
- Control the configurations of the molecular systems
- Observe important processes at atomic resolution

Human dopamine transporter (hDAT) 
Razavi et al, 2017
Traditional MD analysis

**Initial Structure** → **Changing Structure** → **Final Structure**

*Stride*

Larger *Stride* means less frequent data generation

**Frames** (snapshot of atomics positions)

*need to follow dynamics by analyzing the trajectories*

**Simulated Time**

**Post-hoc Analysis**
The increase in computing capability helps the MD simulations generate more data that needs to be analyzed (150,000 atoms + 500,000 snapshots would generate ~ 1.8TB data).

However, the I/O bandwidth does not grow at the same pace → I/O bottleneck
In situ analysis

- Data is analyzed as soon as generated
- Decoupling analysis from the simulation to interleave their executions → Reduce time-to-solution
- Study insights into phenomena of the molecular system in a timely fashion
Data staging

- We focus on **memory-to-memory** for the lowest latency

Comparison of capacity, latency for most important technologies of the memory and storage hierarchies. (Lüttgau et al., 2018)
Placement variants

Simulation | Analysis

CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN
CN | CN | CN | CN

In transit
Dedicated resources
Increase data movement cost

Co-location (Helper-core)
Resource contention
Increases data locality

In transit
Dedicated resources
Increase data movement cost

Network

Information Sciences Institute
In situ framework design

Problem: How to enable in situ execution of simulations and analyses?

Challenges/complexities:
- Decoupling analyses from the simulation to perform in situ requires to manage data staging to additional concurrent components
- Orchestrating data coupling over iterations
- Data incompatibility between decoupled components (simulations, analyses)

We model/design a framework that allows to decouple in situ analyses from the simulation to address these complexities:
- Data staging, coupling \(\rightarrow\) Data transport layer
- Data compatibility \(\rightarrow\) Data abstraction
In situ framework implementation

- Two main components:
  - Data transport layer (DTL) provides interfaces to support different storage tiers
  - DTL plugins interact with the DTL via data chunk abstraction for compatibility

- In this work
  - The DTL is implemented with the help of DIMES (Zhang et al., 2017) to enable in-memory data staging
  - The DTL plugin is integrated with Plumed (Tribello et al., 2014) to offer non-intrusive approach
An ensemble of MD simulations allows sampling wider configurational space.

High barrier problem prevents broadening the conformational sampling to reach interesting molecular events.

MD short simulations

In situ Workflow Ensembles

Iterative process

Ensemble member 1
Simulation 1 → Analysis 1

Ensemble member 2
Simulation 2 → Analysis 2

Ensemble member 3
Simulation 3 → Analysis 3

Workflow ensemble

Ensemble components
Data staging
Performance indicators for in situ workflow ensembles

Problem: How to evaluate performance of an in situ workflow ensembles?

Challenges:
• Using traditional metrics and capturing them separately are not straightforward to synthesize performance of entire workflow ensemble
• Need to take into account resources to characterize performance under resource cost

We introduce multi-stage performance indicators that capture performance of the entire in situ workflow ensembles in terms of multiple resource perspectives
  – Resource usage: How efficiently the resource is utilized?
  – Resource allocation: How efficiently the simulations, analyses are placed on allocated resources?
  – Resource provisioning: How many resources are provisioned to execute efficiently?
Multi-stage performance indicators

Ensemble member 1

Simulation 1 -> Analysis 1

Ensemble member 2

Simulation 2 -> Analysis 2

Ensemble member 3

Simulation 3 -> Analysis 3

Resource Provisioning $P_{1,U,A,P}$

Resource Allocation $P_{1,U,A}$

Resource Usage $P_{1,U}$

Computational Efficiency $E_1$

(Do et al., 2021)

Objective function

$F(P_i)$

$P_1, P_2, P_3$

max

Information Sciences Institute

Page 15

USC Viterbi School of Engineering
Experiment setup

Simulations

Medium-scale all-atom system containing the GltPh transporter protein (Akyuz 2015) implemented in GROMACS (P Bjelkmar et al., 2010)

Analyses

Collective variable (largest eigenvalue of bipartite distance matrices between two substructures) (Barducci 2011, Johnston 2017)

- Execution platform: Cori, a Cray XC40 supercomputer at NERSC. Each compute node is equipped with
  - 2 Intel Xeon E5-2698 v3 (16 cores each)
  - 128 GB of DRAM
- TAU (Shende et al., 2006) is leveraged to collect performance information
→ C1.5 outperforms other configurations, which validates the benefit of co-locating coupled components
Takeaways

• The proposed indicators can be leveraged for evaluating scheduling decision of in situ ensemble under resource constraints

• Provide hints to improve effectiveness of resource usage → optimizing simulation exploration by running many MD simulations at a time

• Future work will consider leveraging the proposed indicators for scheduling in situ components of a workflow ensemble to enable high-throughput ensemble of simulations
Thank You!


References

• Barducci et al. 2011. Meta-dynamics.WIREs Computational Molecular Science 1, 5 (2011)
References

• Taufer et al., 2019. Characterizing In Situ and In Transit Analytics of Molecular Dynamics Simulations for Next-Generation Supercomputers. 15th International Conference on eScience (eScience).
• Zou et al., 2014. FlexAnalytics: A flexible data analytics framework for big data applications with I/O performance improvement. Big Data Research 1, 4 – 13.
• Lofstead et al., 2008. Flexible io and integration for scientific codes through the adaptable io system (adios), 6th international workshop on Challenges of large applications in distributed environments.
In-line in situ analysis

- Data is analyzed as soon as generated
- The simulation and analysis interchangeably execute
- Analysis needs to be embedded in simulation code
- Less robust
IDLE SIMULATION

In situ step 5

Simulation step 4

Analysis step 4

In situ step 4

IDLE ANALYZER
Efficiency model

- Computing in situ step is lightweight and can be performed online.
- *In situ step* is leveraged to estimate:
  1. **Makespan**: duration of an in situ step $\times$ number of in situ steps - overlapped part between in situ steps
  2. **Useful computation**: measured by time for computation except idle time during an in situ step
- Evaluate computational efficiency:

$$E = \frac{\text{Estimation of useful computation}}{\text{Estimation of makespan}}$$

→ Minimize idle time
→ Minimize makespan
Characterization challenges

- Evaluating each ensemble component/member exclusively is hard to:
  - Generate a full picture of the workflow ensemble performance
  - Compare between different executions/configurations

- Without taking into account resources, the performance could be misleading

C1.4 & C1.5 use fewer nodes than other configurations
Component placement

- The simulation is co-located with the analysis, iff \( |s| = |s \cup a| \)
- The simulation and analysis are assigned to different nodes, iff \( |s| < |s \cup a| \)

Set of node indexes where a simulation is executed

\[
0 < \frac{|s|}{|s \cup a|} \leq 1
\]

Set of node indexes where the coupled analysis is executed

**Placement indicator of ensemble member \( i \) with \( K_i \) analyses**

\[
CP_i = \frac{1}{K_i} \sum_{j=1}^{K_i} \frac{|s_i|}{|s_i \cup a_i^j|}
\]

Mean of ratios forming by all (simulation, analysis) pairs

Maximize placement indicator prioritizes placements that minimize the number of computing resources (number of compute nodes) used by that ensemble member.
Performance indicator $P_i$ of ensemble member i

**1st stage**

Resource usage (U)

\[
P_i^U = \frac{E_i}{c_i}
\]

Total number of cores used by ensemble member i

**2nd stage**

Resource allocation (A)

\[
P_i^{U,A} = P_i^U \times CP_i
\]

Placement indicator

**3rd stage**

Resource Provisioning (P)

\[
P_i^{U,A,P} = \frac{P_i^{U,A}}{M}
\]

Total number of compute nodes used by all ensemble members

Resource allocation (A) and resource provisioning (P) can be used interchangeably
Synthesis of performance indicators

- $P_i$ can be either $P_i^U$, $P_i^{U,A}$, $P_i^{U,P}$, $P_i^{U,A,P}$ ($= P_i^{U,P,A}$)

- The objective function of $N$ ensemble members \textit{(the higher the better)}

$$F(P_i) = \bar{P} - \sqrt{\frac{1}{N} \sum_{i=1}^{N} (P_i - \bar{P})^2}$$

where $\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i$

Maximize average performance of ensemble members

Minimize variability among ensemble members
Two analyses per simulation