Assessing Resource Provisioning and Allocation of Ensembles of In Situ Workflows

Tu Mai Anh Do, Loïc Pottier, Rafael Ferreira da Silva, Silvina Caíno-Lores, Michela Taufer, Ewa Deelman

International Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2)
September 9th, 2021

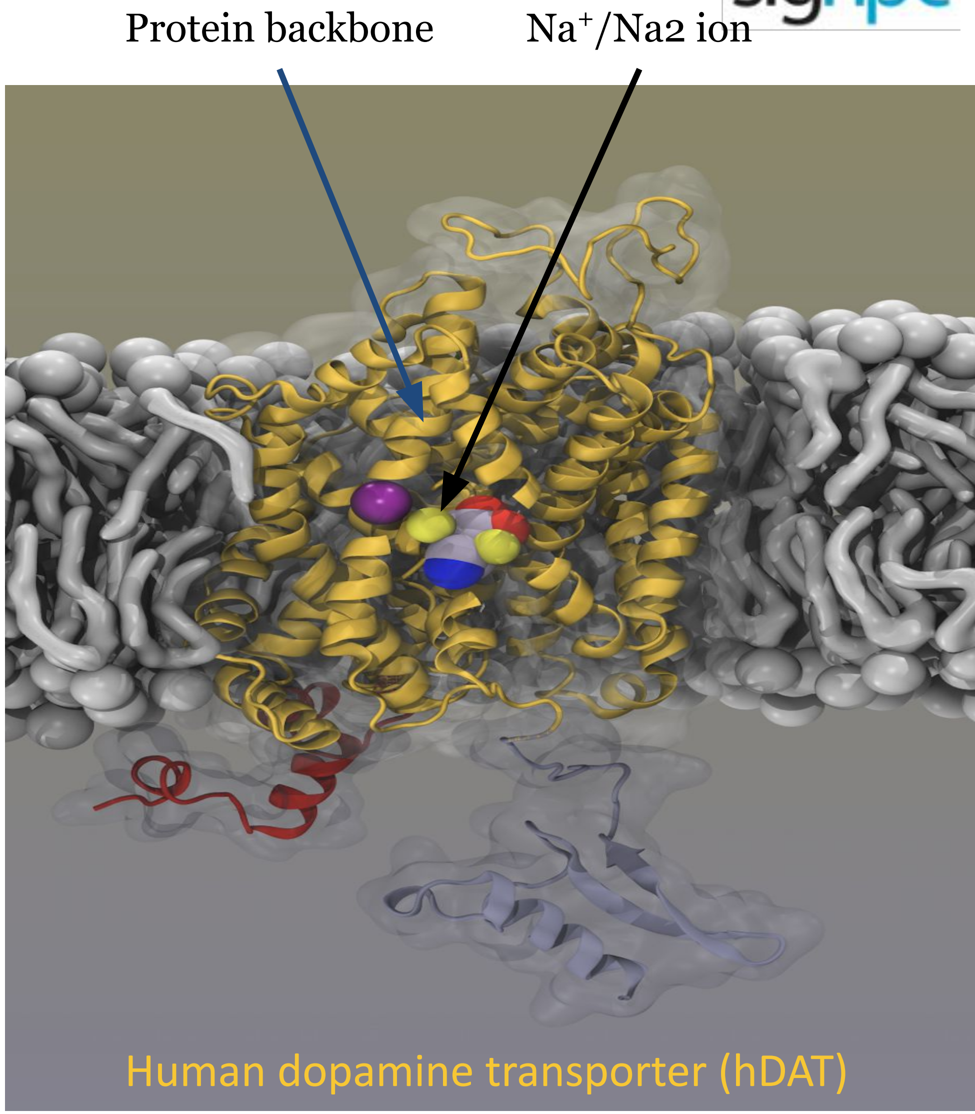
This work is funded by NSF contracts #1741040 and #1741057; and DOE contract #DE-SC0012636

1. Motivations and Background

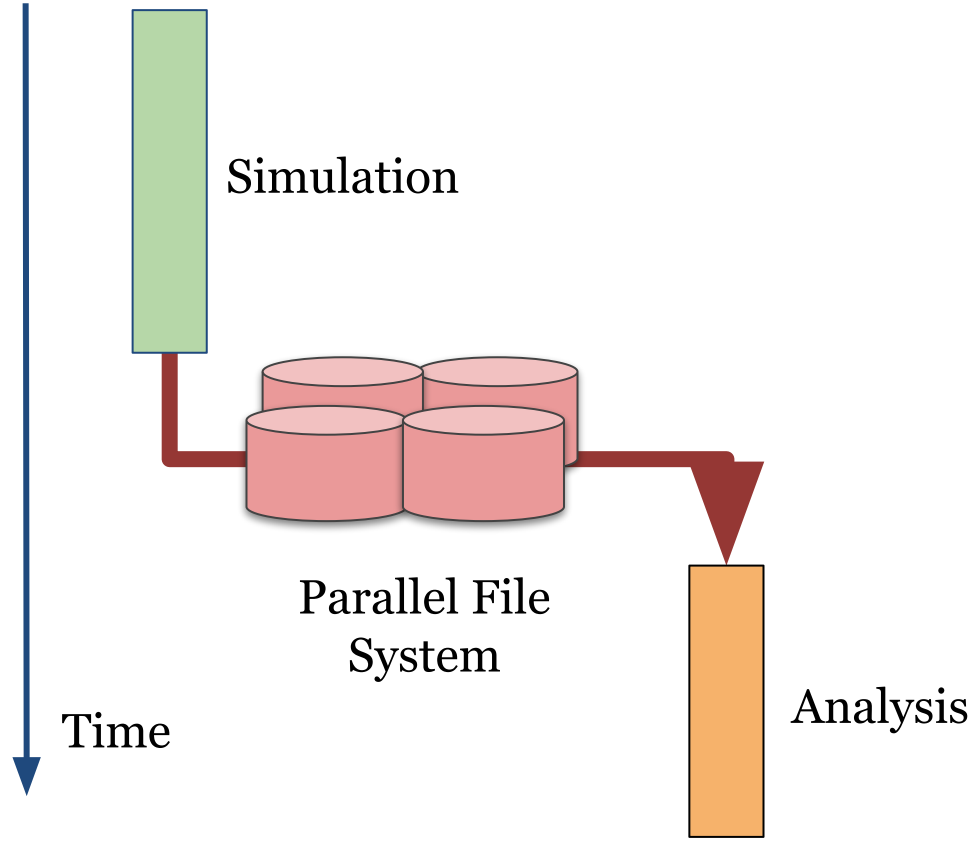
Molecular dynamics

- 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, such as temperature, pressure
  - Observe important processes at atomic resolution, such as conformational changes, phase transitions, or binding events
- To obtain these outcomes, the analysis of MD trajectories (snapshots of atomic positions) is needed to integrate into the simulation pipeline

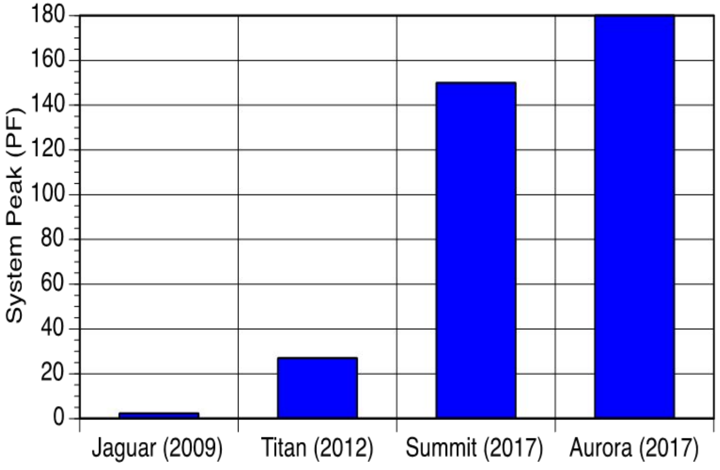
Razavi et al, 2017

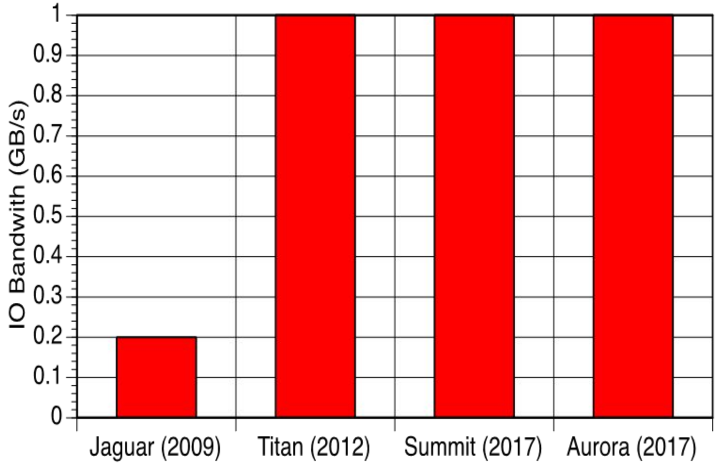


Post-processing



I/O stagnant on contemporary leadership computers. (Johnston et al., 2017)

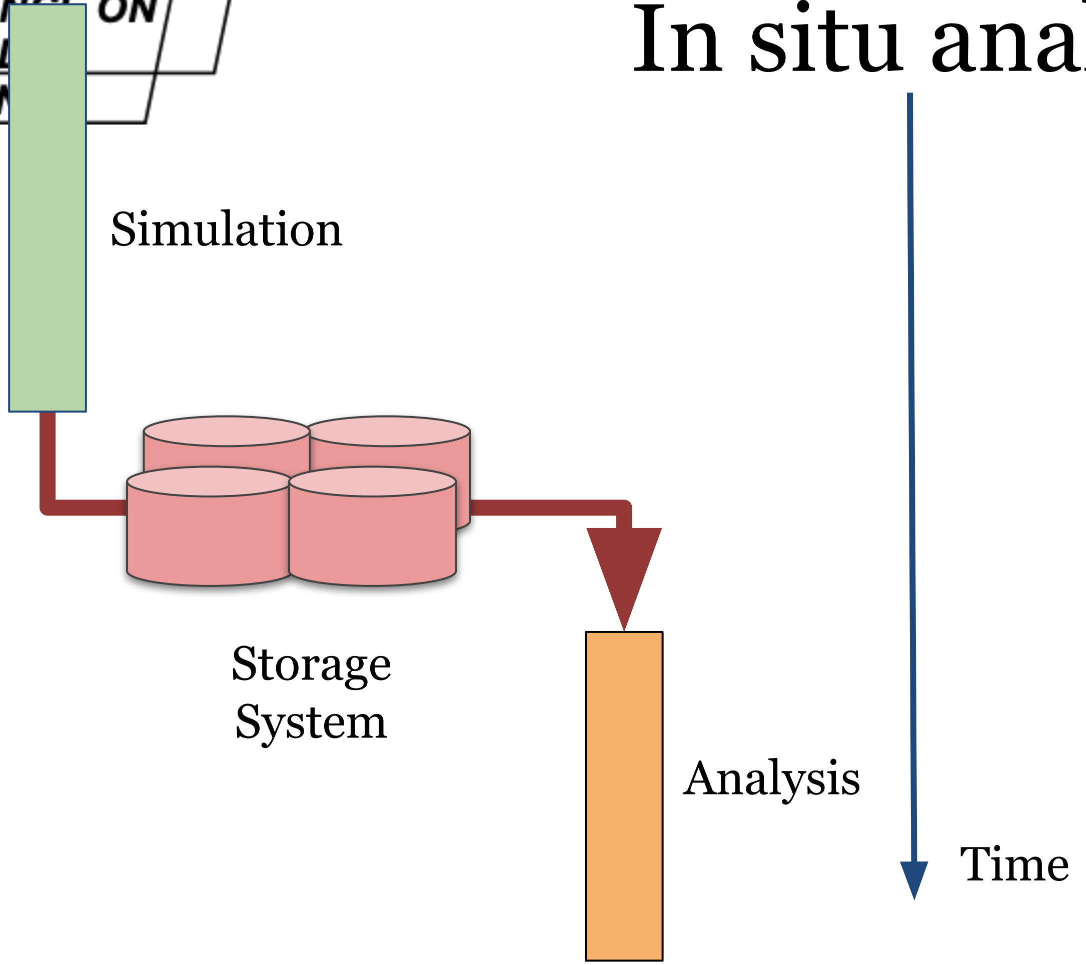


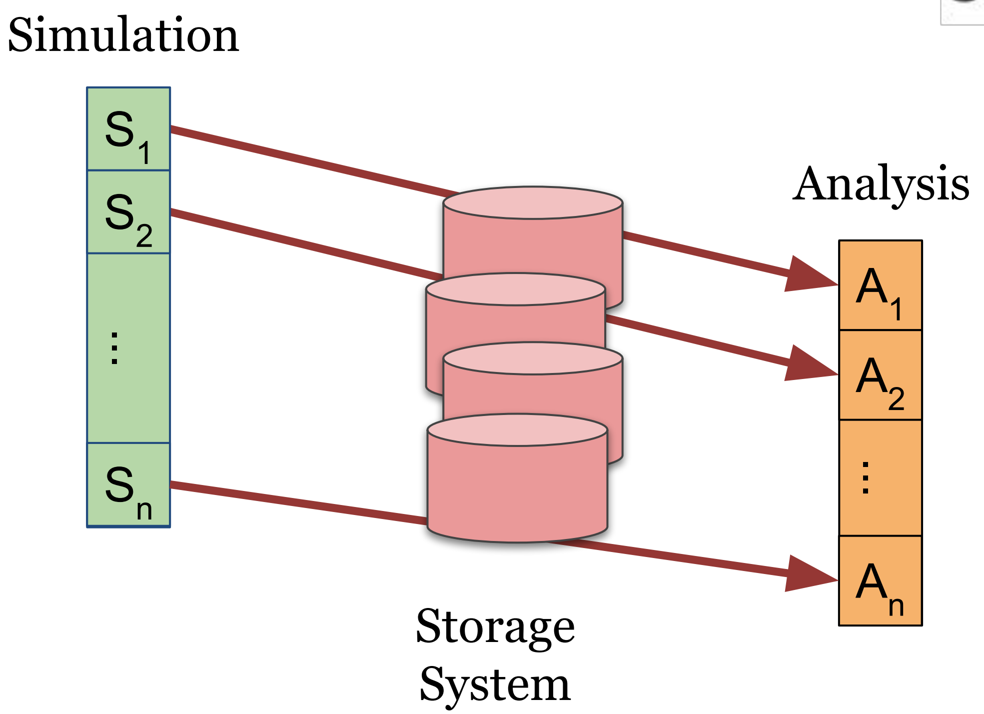


- In post-processing, frames are stored to file system for analyzing later

- 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



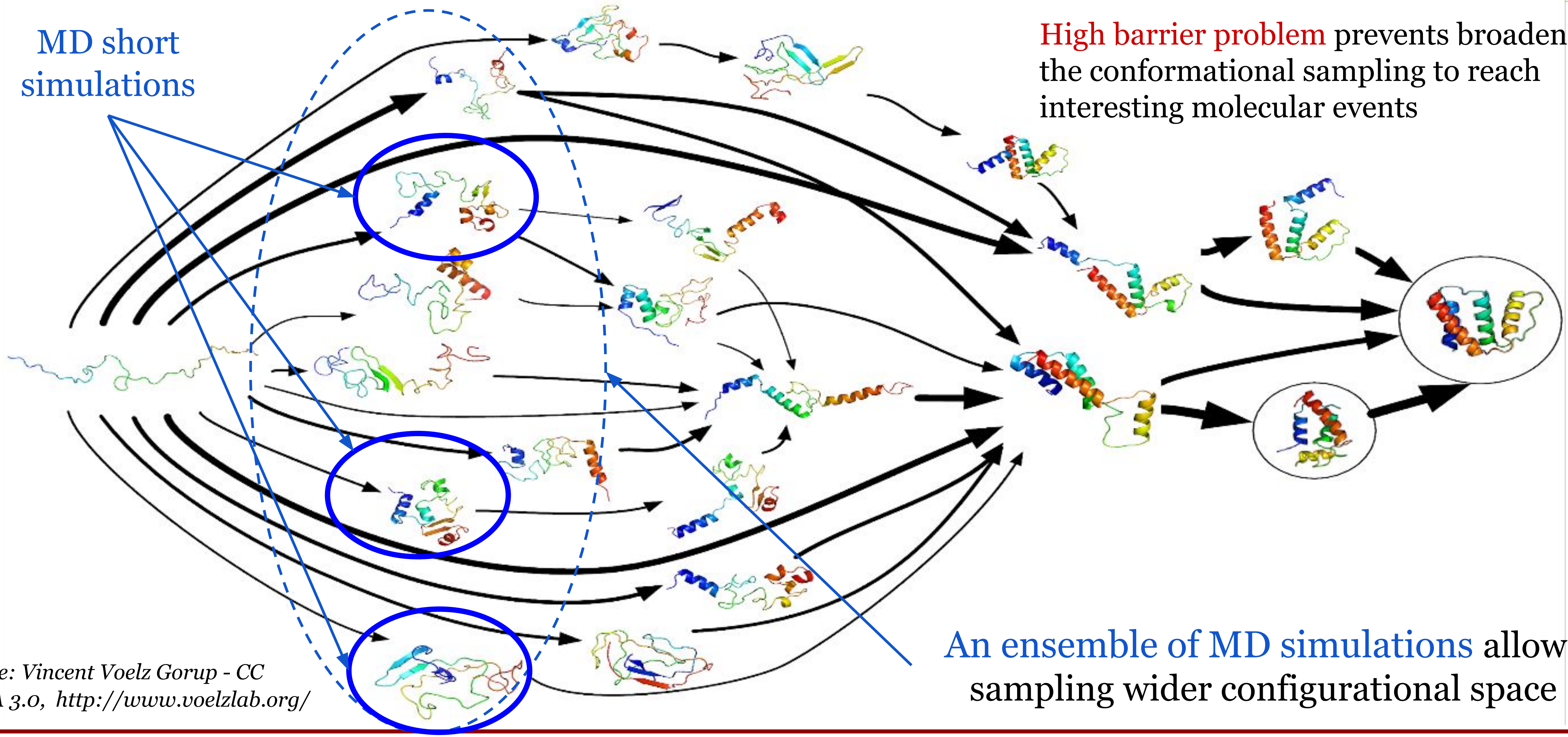


- Data is analyzed as soon as generated
- The simulation and analysis tasks are interleaved to reduce time-to-solution
- Performing analyses at simulation runtime helps to study insights into phenomena of the molecular system in a timely fashion → better science discovery

MD simulation ensemble

MD short simulations

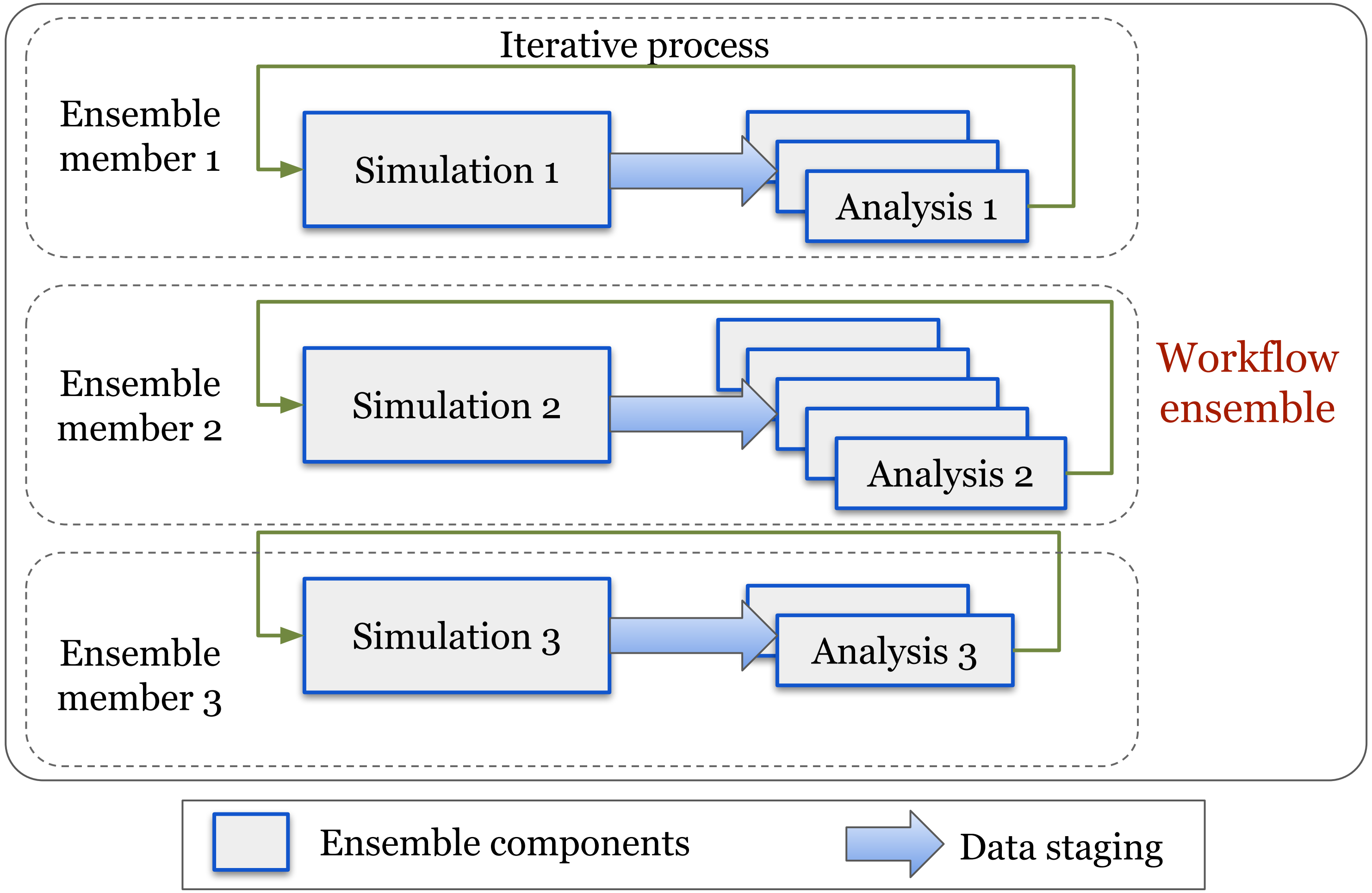
High barrier problem prevents broadening the conformational sampling to reach interesting molecular events



An ensemble of MD simulations allows sampling wider configurational space

Source: Vincent Voelz Gorup - CC BY-SA 3.0, http://www.voelzlab.org/

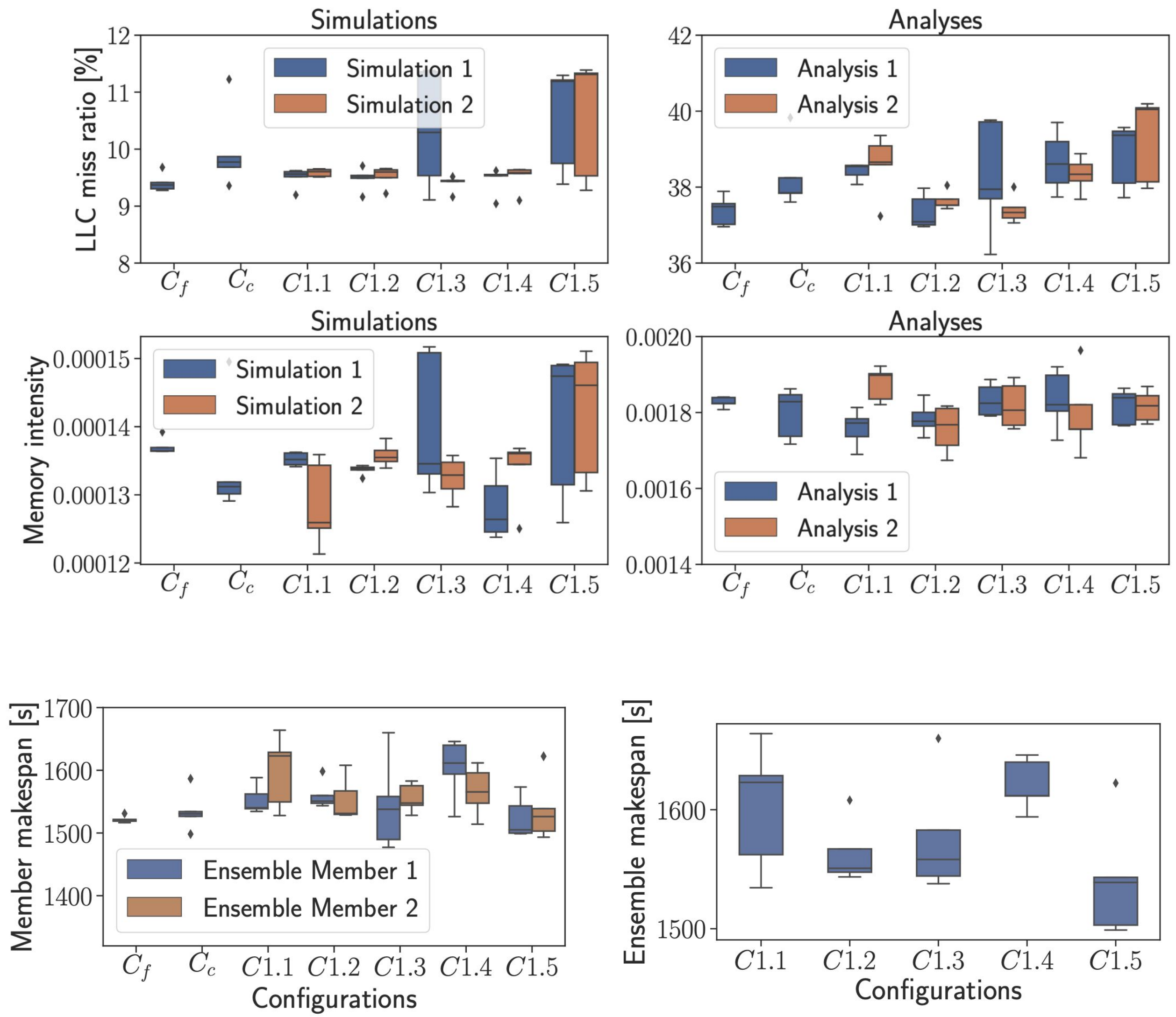
In situ Workflow Ensembles



Characterization

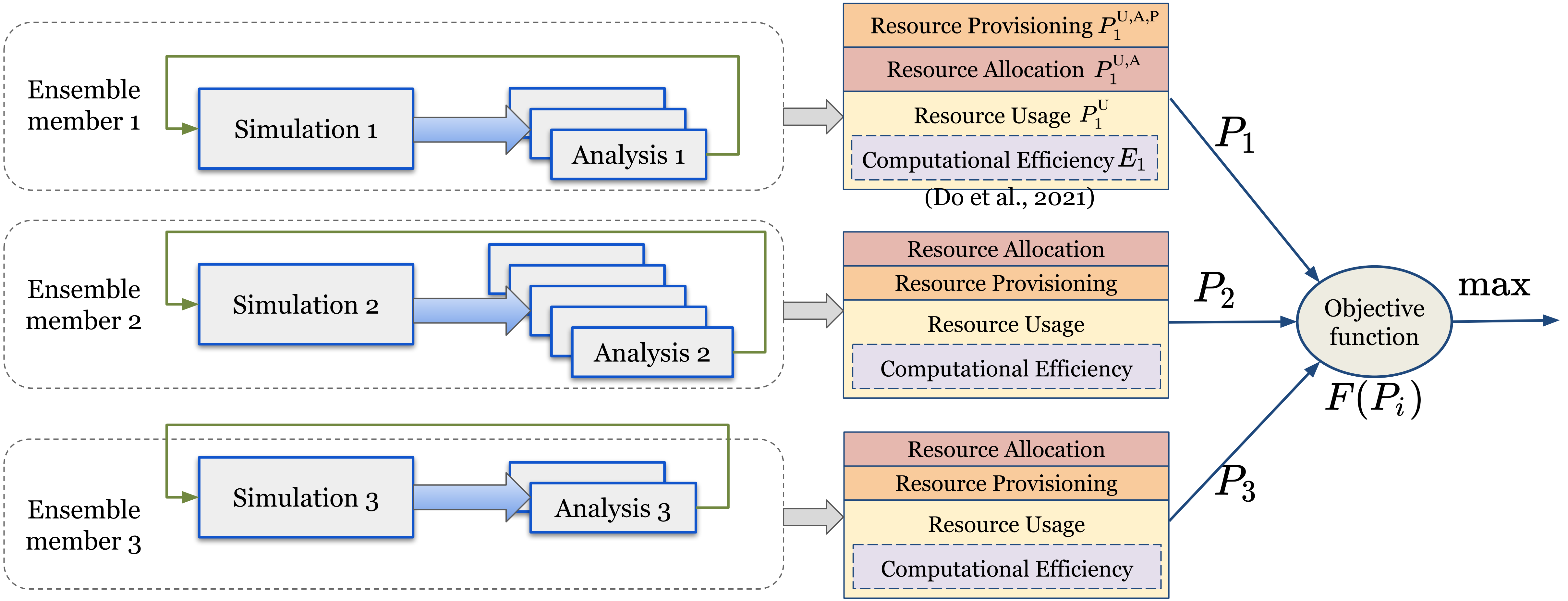
Evaluating each metric exclusively does not guarantee a thorough understanding of the workflow ensemble performance

→ A need for a method that captures performance at multiple levels of workflow ensembles



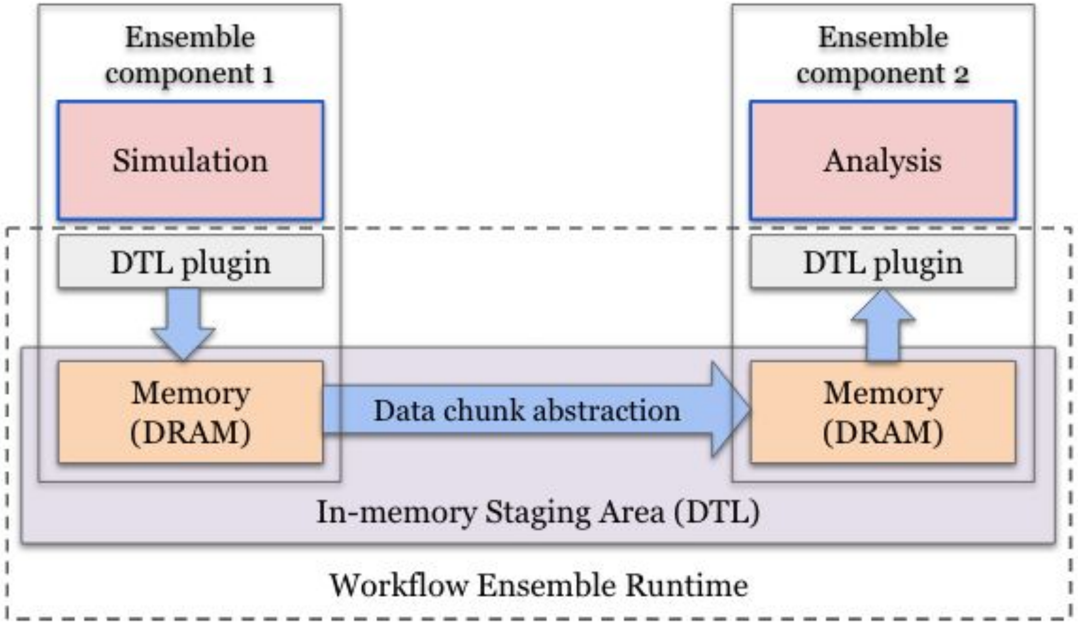
2. Performance Evaluation of Workflow Ensemble

Multi-stage performance indicators



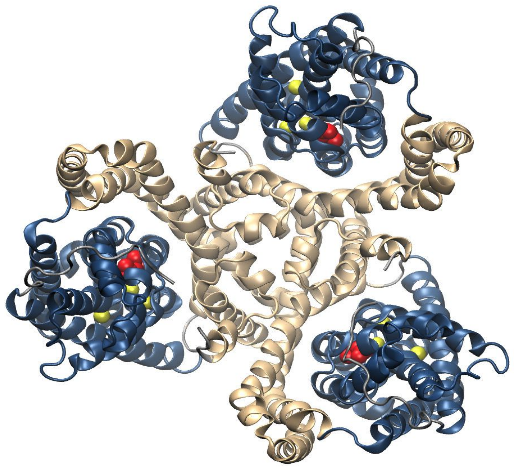
Experiment setup

Runtime system



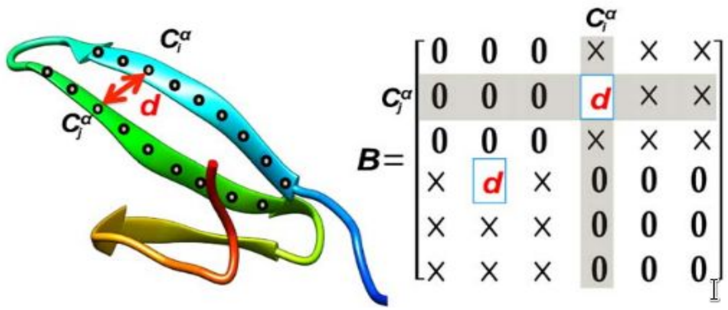
In-memory DTL is implemented with the help of DIMES (Fan Zhang et al., 2017.)

Simulations



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

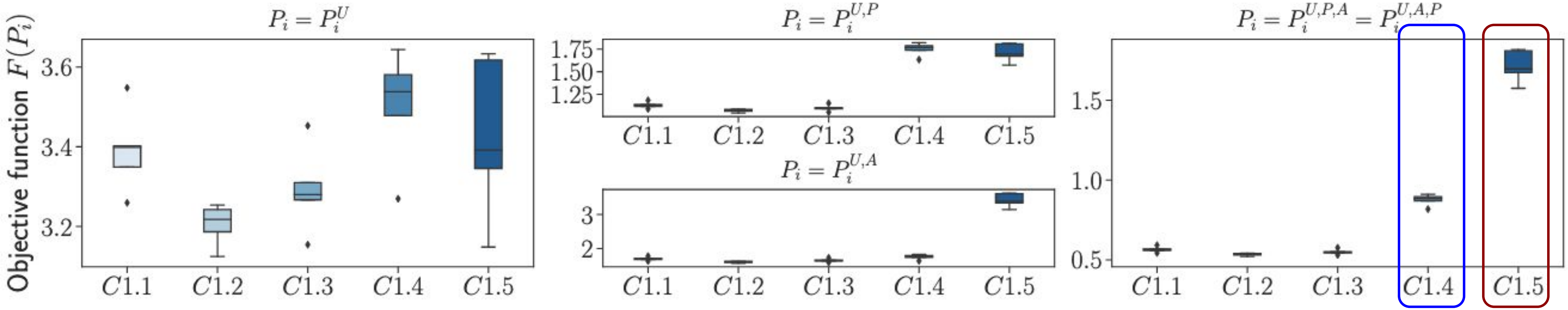
Analyses



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

- Our execution platform is Cori@NERSC. Each compute node is equipped:
  - 2 Intel Xeon E5-2698 v3 (16 cores each)
  - 128 GB of DRAM

One analysis per simulation



➔ $C1.5$ outperforms other configurations, which validates the benefit of co-locating coupled components

Configuration	Number of computing nodes	Number of ensemble members	Node indexes: Ensemble member 1: Simulation 1	Ensemble member 1: Analysis 1	Ensemble member 2: Simulation 2	Ensemble member 2: Analysis 2
C_f	2	1	n_0	n_1	-	-
C_c	1	1	n_0	n_0	-	-
C1.1	3	2	n_0	n_2	n_1	n_2
C1.2	3	2	n_0	n_1	n_0	n_2
C1.3	3	2	n_0	n_0	n_1	n_2
☆ C1.4	2	2	n_0	n_1	n_0	n_1
☆☆ C1.5	2	2	n_0	n_0	n_1	n_1

Conclusions

- Due to the capability of comparing different configurations in multiple resource aspects, the proposed indicators can be leveraged for evaluating scheduling decision of in situ ensemble under resource constraints
- The approach improves effectiveness of resource usage, thereby optimizing simulation exploration by deploying as many as possible MD simulations at a time

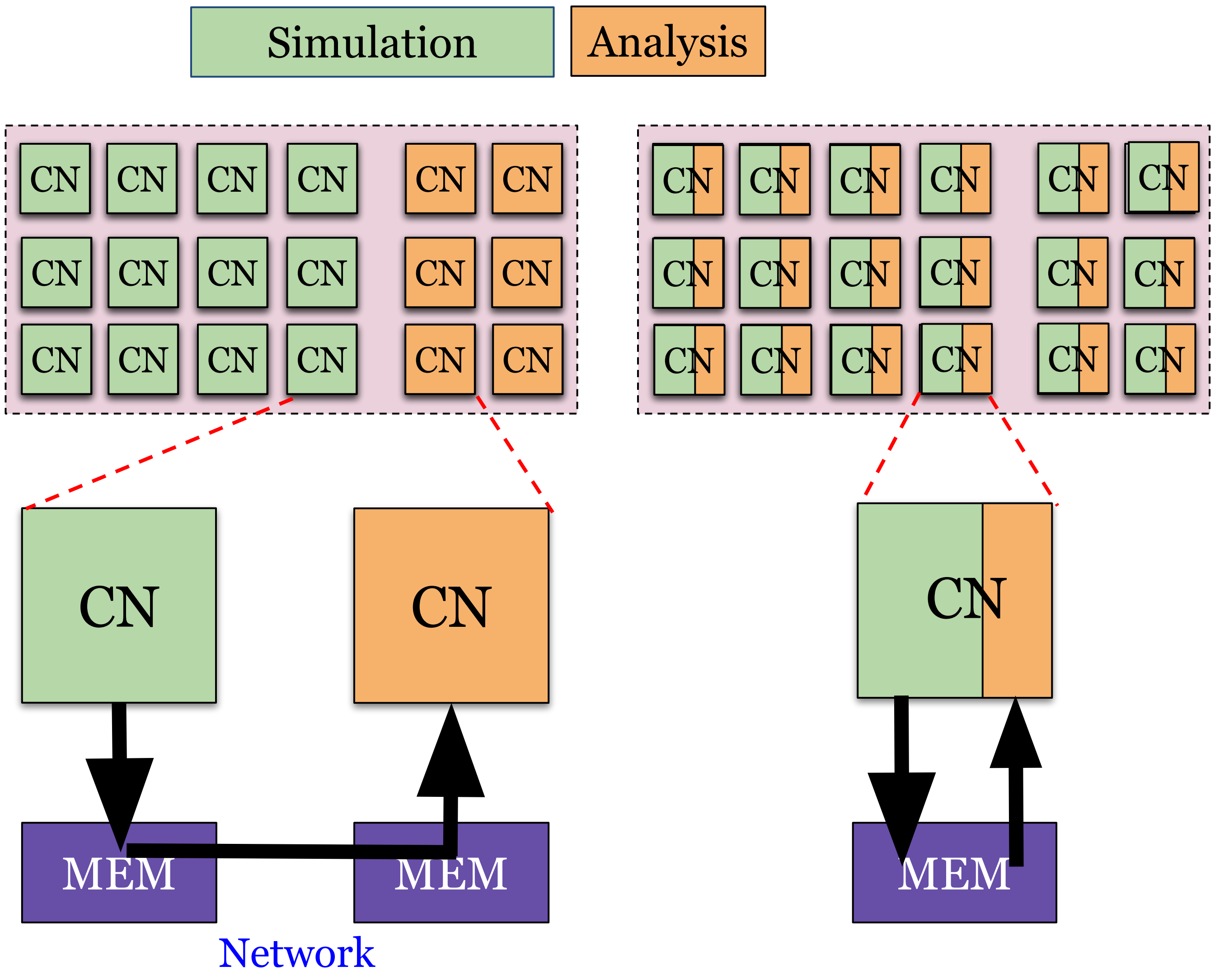
- Future work will consider leveraging the proposed indicators for scheduling in situ components of a workflow ensemble to enable high-throughput ensemble of simulations

Thank You!

References

- P Bjelkmar et al., 2010. Implementation of the CHARMM Force Field in GROMACS:Analysis of Protein Stability Effects from Correction Maps, Virtual Interaction Sites, and Water Models. J. Chem. Theory Comput.6, 2 (2010).
- Johnston et al. “In situ data analytics and indexing of protein trajectories”, Journal of Computational Chemistry, 38 (16), 1419-1430, (2017)
- Akyuz et al. 2015. Transport domain unlocking sets the uptake rate of an aspartate transporter. Nature 518, 7537(2015)
- Barducci et al. 2011. Meta-dynamics.WIREs Computational Molecular Science 1, 5 (2011)
- Johnston et al. In situ data analytics and indexing of protein trajectories. Journal of Computational Chemistry, 38 (16), 1419-1430, (2017)
- Fan Zhang et al., 2017. In-memory staging and data-centric task placement for coupled scientific simulation workflows.Concurrency and Computation: Practice and Experience 29, 12 (2017)
- Do et al., 2021. A Lightweight Method for Evaluating In Situ Workflow Efficiency. Journal of Computational Science, 48, 101259.

Placement variants



In transit

- Dedicated resources
- Increase data movement cost

Co-location (Helper-core)

- Resource contention
- Increases data locality

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

$$0 < \frac{\overbrace{|s|}}{\underbrace{|s \cup a|}} \leq 1$$

Set of node indexes where the coupled analysis is executed

Placement indicator of ensemble member i with K_i analyses

$$CP_i = \underbrace{\frac{1}{K_i} \sum_{j=1}^{K_i} \frac{|s_i|}{|s_i \cup a_i^j|}}$$

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.

INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING

acm In-Cooperation sighpc

Performance indicators

Efficiency of single core usage

1st stage

Resource usage (U)

$$P_i^{\mathrm{U}} = \frac{E_i}{\underbrace{c_i}_{\text{Total number of cores used by ensemble member i}}}$$

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

(Do et al., 2021)

Efficiency of allocating ensemble components

2nd stage

Resource allocation (A)

$$P_i^{\mathrm{U,A}} = P_i^{\mathrm{U}} \times \underbrace{CP_i}_{\text{Placement indicator}}$$

Minimizing resources provisioned

3rd stage

Resource Provisioning (P)

$$P_i^{\mathrm{U,A,P}} = \frac{P_i^{\mathrm{U,A}}}{\underbrace{M}_{\text{Total number of compute nodes used by all ensemble members}}}$$

Resource allocation (A) and resource provisioning (P) can be used interchangeably

INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING

acm In-Cooperation sighpc

Synthesis of performance indicators

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

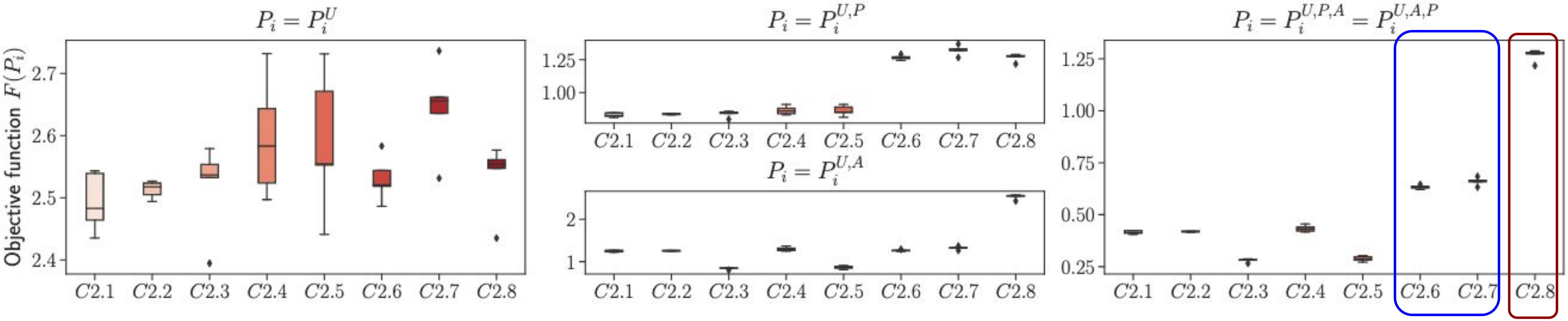
- The objective function of N ensemble members (the higher the better)

Maximize $$\underbrace{F(P_i) = \overline{P}}_{\text{Mean}} - \underbrace{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - \overline{P})^2}}_{\text{Standard deviation}} \quad \text{where} \quad \overline{P} = \frac{1}{N}\sum_{i=1}^{N} P_i$$

Maximize average performance of ensemble members ← Mean

Standard deviation → Minimize variability among ensemble members

Two analyses per simulation



	Configuration	Number of computing nodes (N)	Number of ensemble members	Node indexes					
				Ensemble member 1			Ensemble member 2		
				Simulation 1	Analysis 1.1	Analysis 1.2	Simulation 2	Analysis 2.1	Analysis 2.2
	C2.1	3	2	n_0	n_2	n_2	n_1	n_2	n_2
	C2.2	3	2	n_0	n_1	n_1	n_0	n_2	n_2
	C2.3	3	2	n_0	n_1	n_2	n_0	n_1	n_2
	C2.4	3	2	n_0	n_0	n_2	n_1	n_1	n_2
	C2.5	3	2	n_0	n_1	n_2	n_1	n_0	n_2
⋆	C2.6	2	2	n_0	n_1	n_1	n_0	n_1	n_1
	C2.7	2	2	n_0	n_0	n_1	n_1	n_0	n_1
⋆⋆	C2.8	2	2	n_0	n_0	n_0	n_1	n_1	n_1