Rutgers University

Department of Electrical and Computer Engineering

A Framework for Replica Exchange Simulations

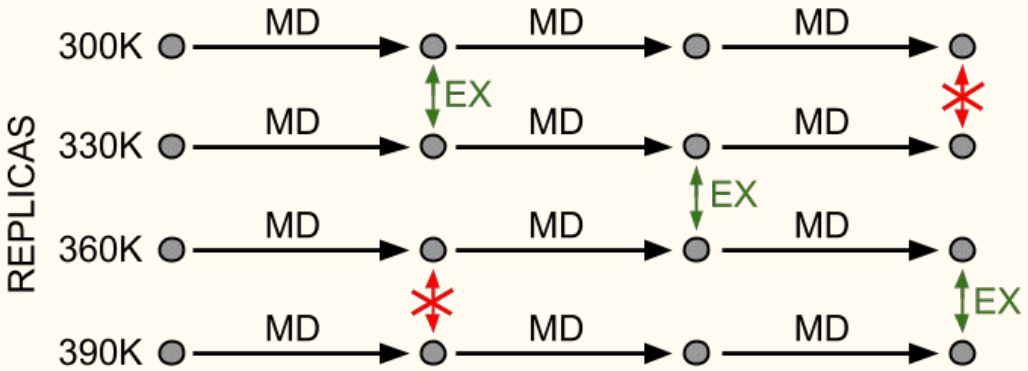
Antons Treikalis

Thesis advisor: Dr. Shantenu Jha

Outline

- What is Replica Exchange?
- Asynchronous Replica Exchange
- `RepEx` software package
- Experiments
- Conclusion

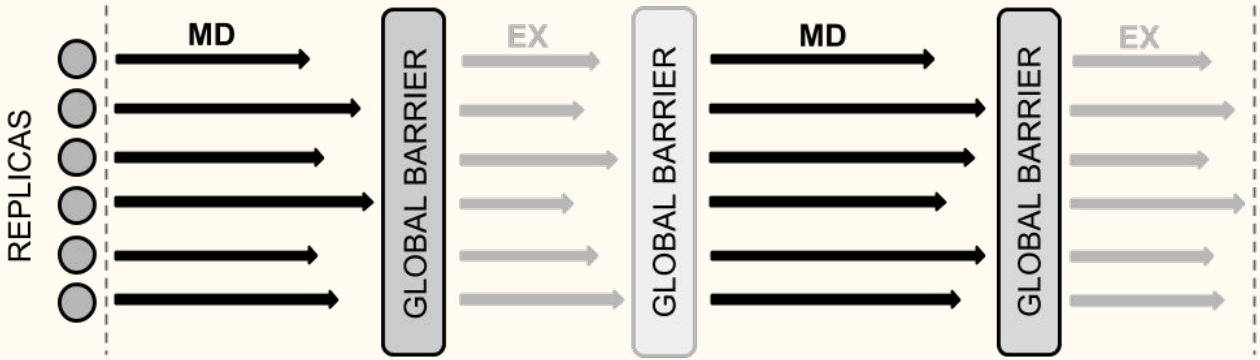
What is Replica Exchange?



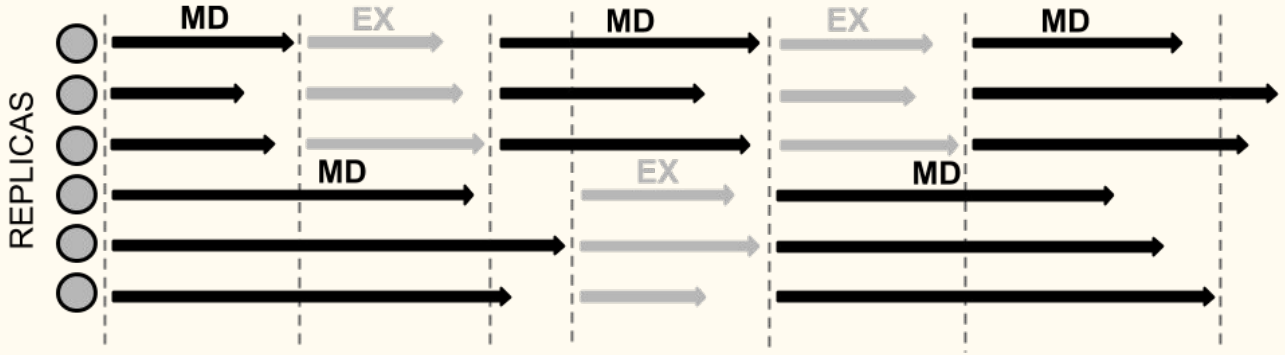
Replica Exchange Molecular Dynamics (REMD) introduced by Sugita and Okamoto in 1999 [1]

- The algorithm:
  - Concurrently run multiple Molecular Dynamics (MD) simulations at different temperatures for N time-steps
  - Select exchange partners
  - Accept (or reject) an exchange of temperatures according to probability given by the Metropolis criterion
  - Repeat

Synchronization in Replica Exchange



Top: **Synchronous** RE; Bottom: **Asynchronous** RE



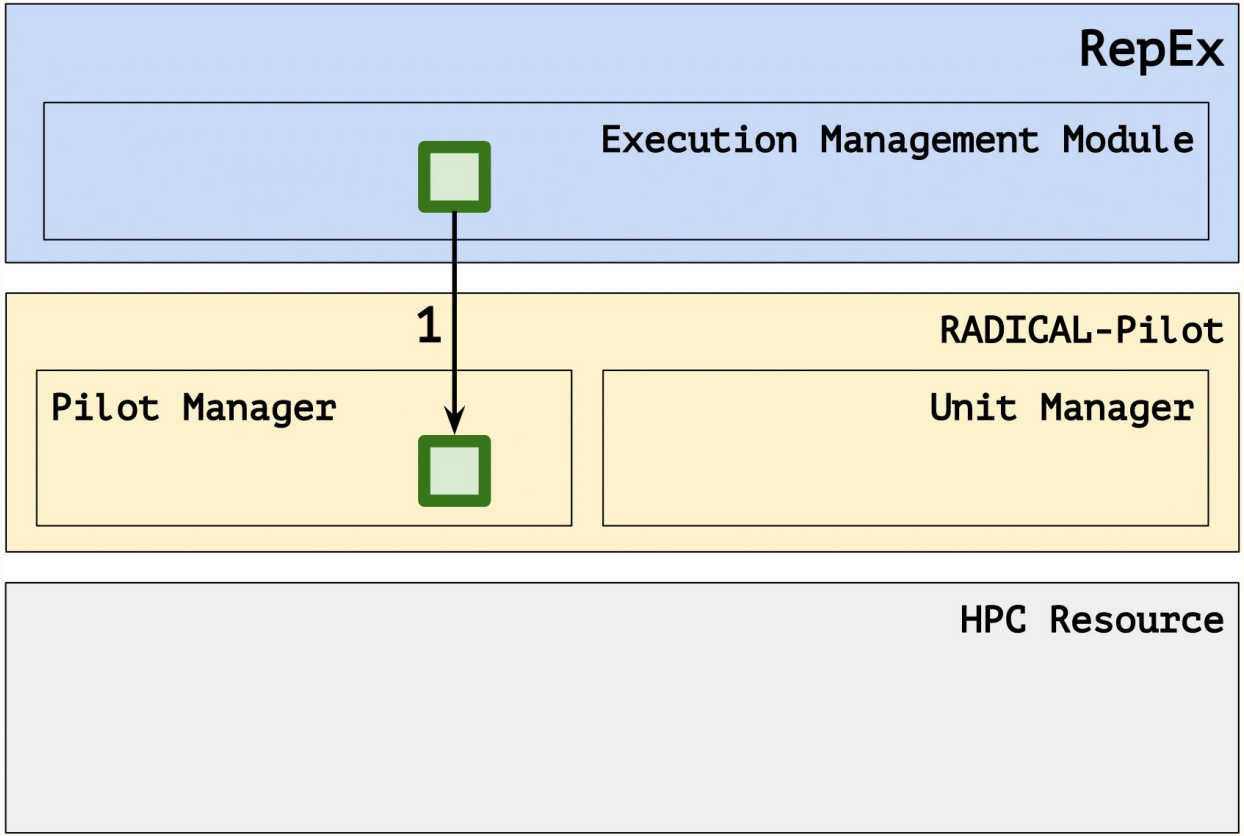
Asynchronous Replica Exchange

- Traditionally viewed as an algorithm for Grid Computing: **Heterogeneity of computational resources and fluctuation of resource availability**
- Is well suited for heterogeneous simulations: Quantum Mechanics (QM) / Molecular Mechanics (MM)
- Facilitates adaptive sampling: **replicas can be stopped and started dynamically and according to the simulation requirements**
- **Fast simulation recovery times** at the event of replica failures

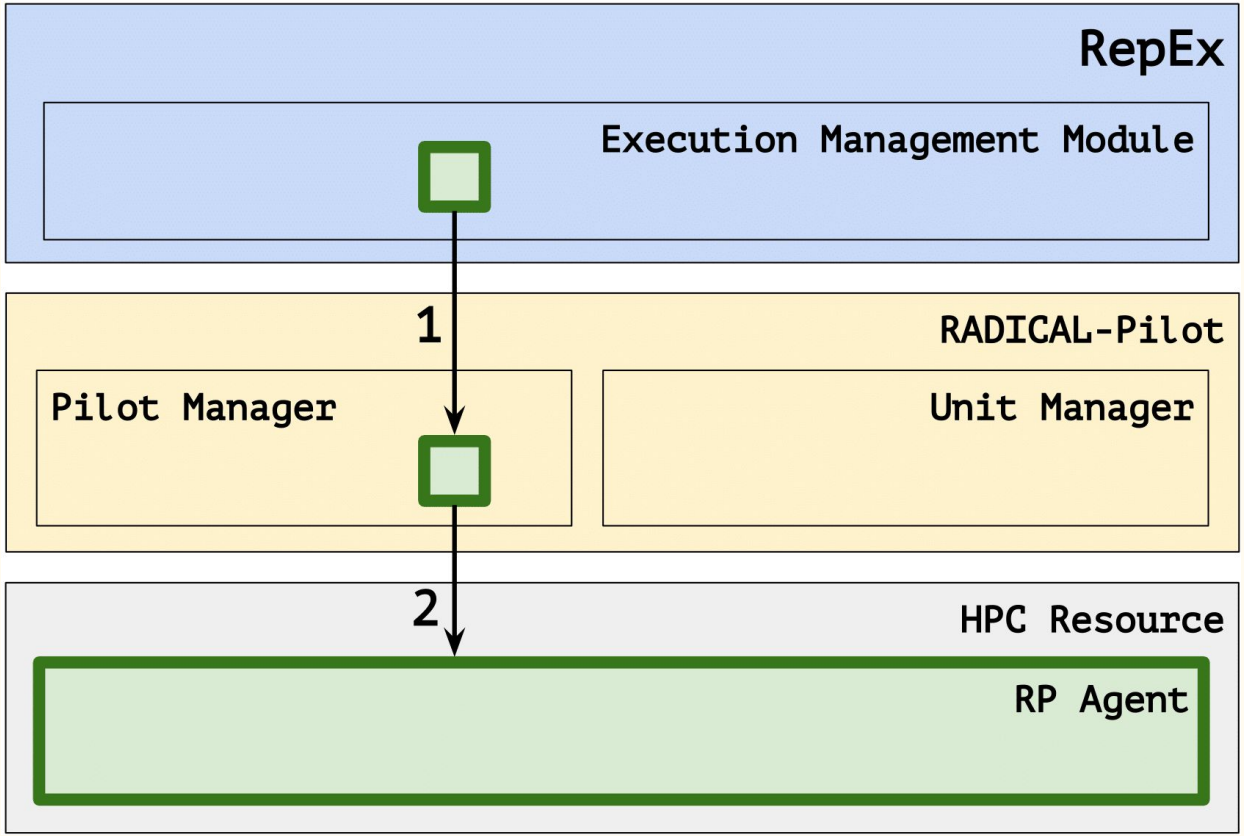
Our contribution

Molecular Simulation Software Packages with Integrated RE algorithm	`RepEx` **software package**
Advances in the RE methodology are tied to the particular MD code	Modules implementing RE are used by multiple MD engines
Can't take advantage of the advances in other MD simulation engines	Can be extended to support any MD engine
High barrier for implementation of REMD methods	Explicit decoupling of RE methodology from MD engine facilitates implementation of various RE methods
High barrier for addition of new exchange parameters	Can be extended to use any exchange parameter
No synchronization flexibility	Supports sync. and async. RE
No fault tolerance	Ability to restart replicas

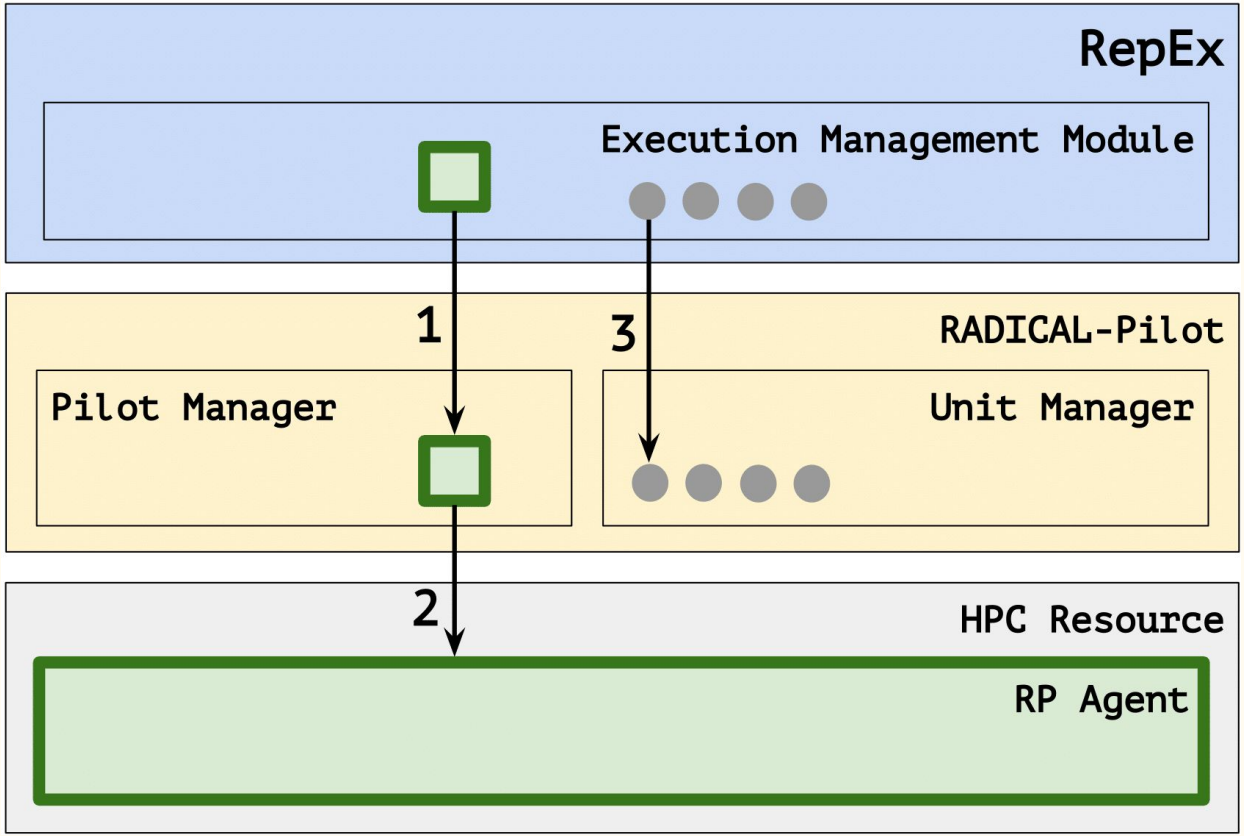
Task Execution Diagram



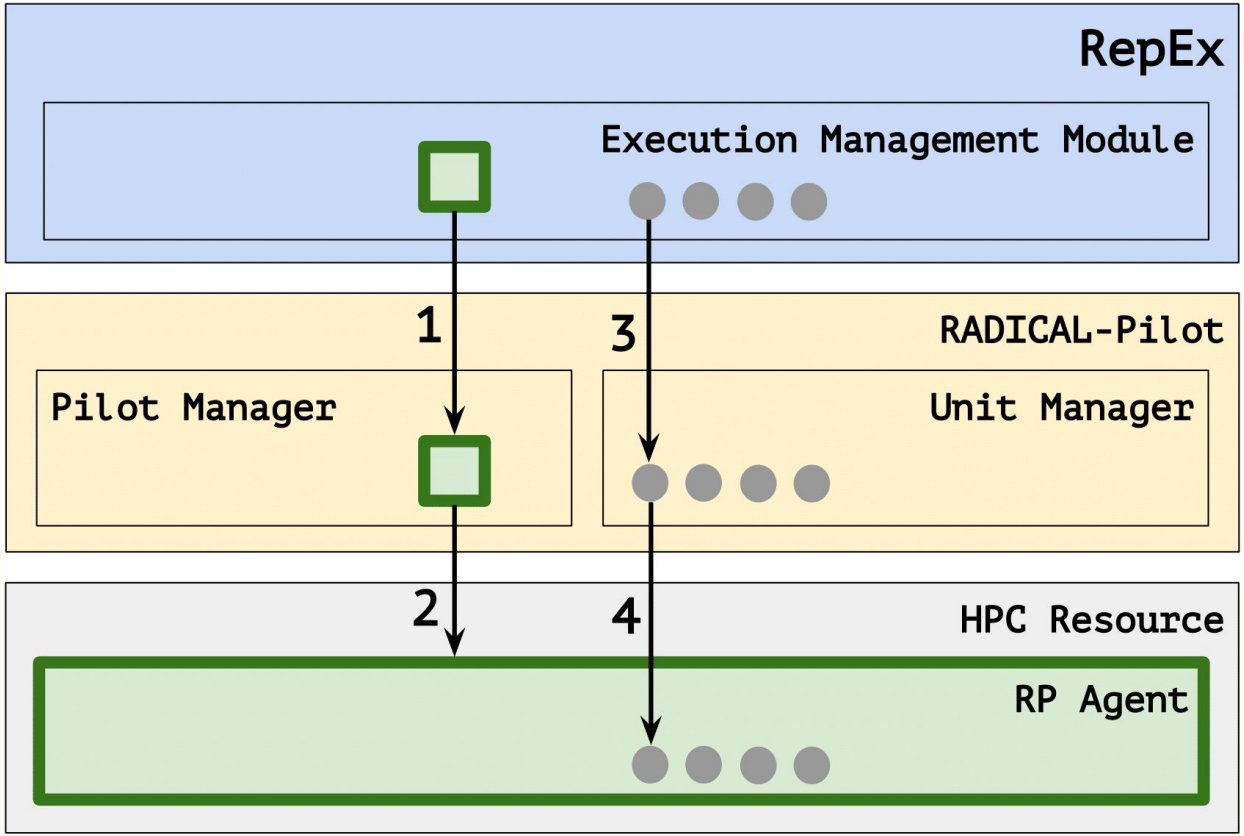
Task Execution Diagram



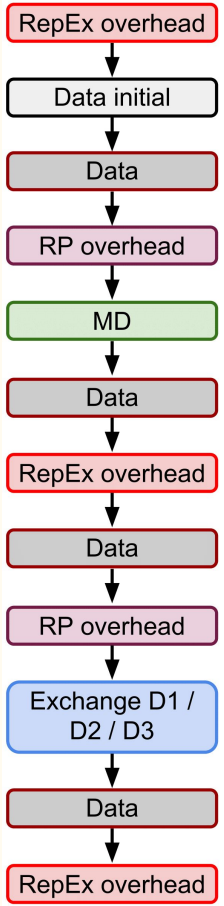
Task Execution Diagram



Task Execution Diagram

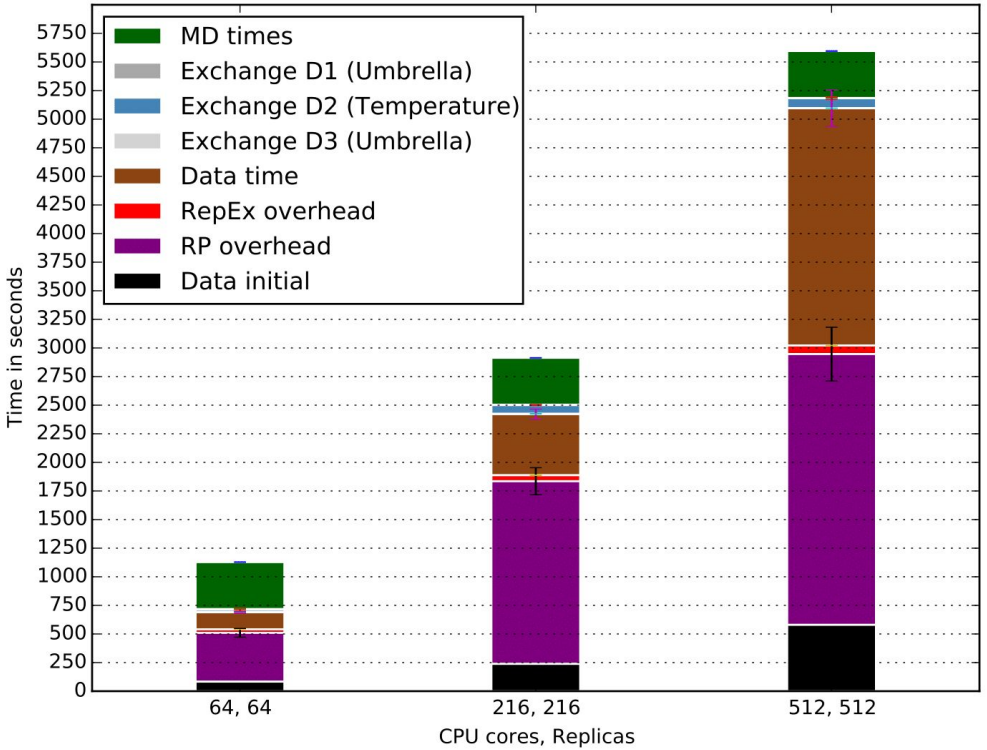


Performance



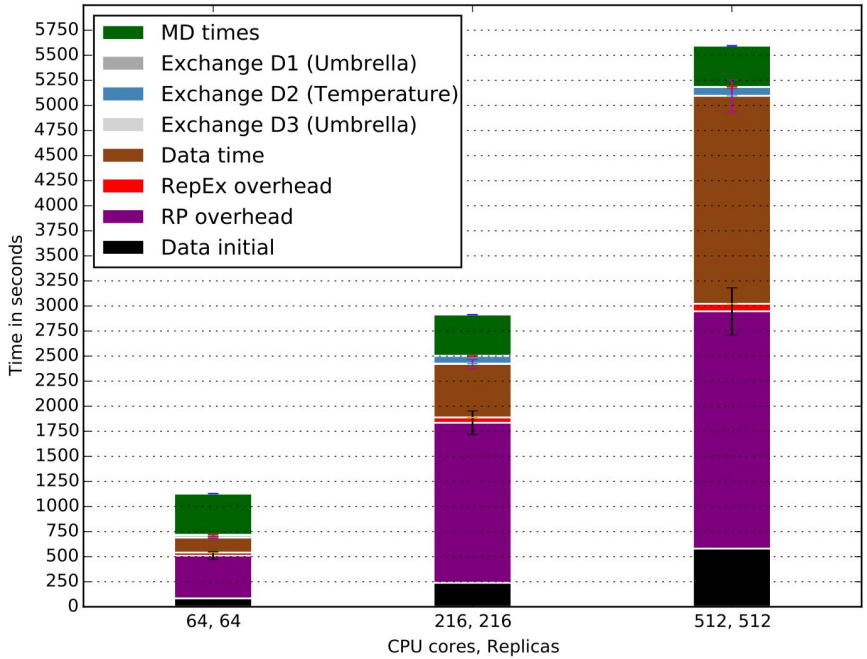
- **Data (initial)** - time taken to stage the files required by the simulation
- **Data** - time taken to stage the files required by each simulation cycle
- **MD** - time taken to perform N simulation time-steps before an exchange
- **Exchange D1** - time taken to determine umbrella exchanges
- **Exchange D2** - time taken to determine temperature exchanges
- **Exchange D3** - time taken to determine temperature exchanges
- **RepEx overhead (client side)** - time taken to perform local method calls
- **RP overhead** - task launching time and time for internal RP communication

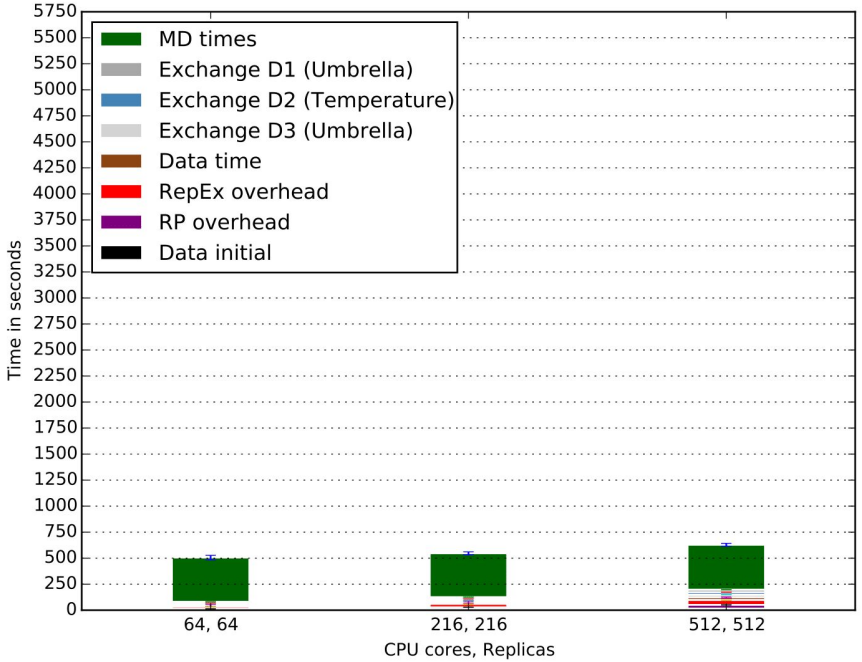
Performance: Initial



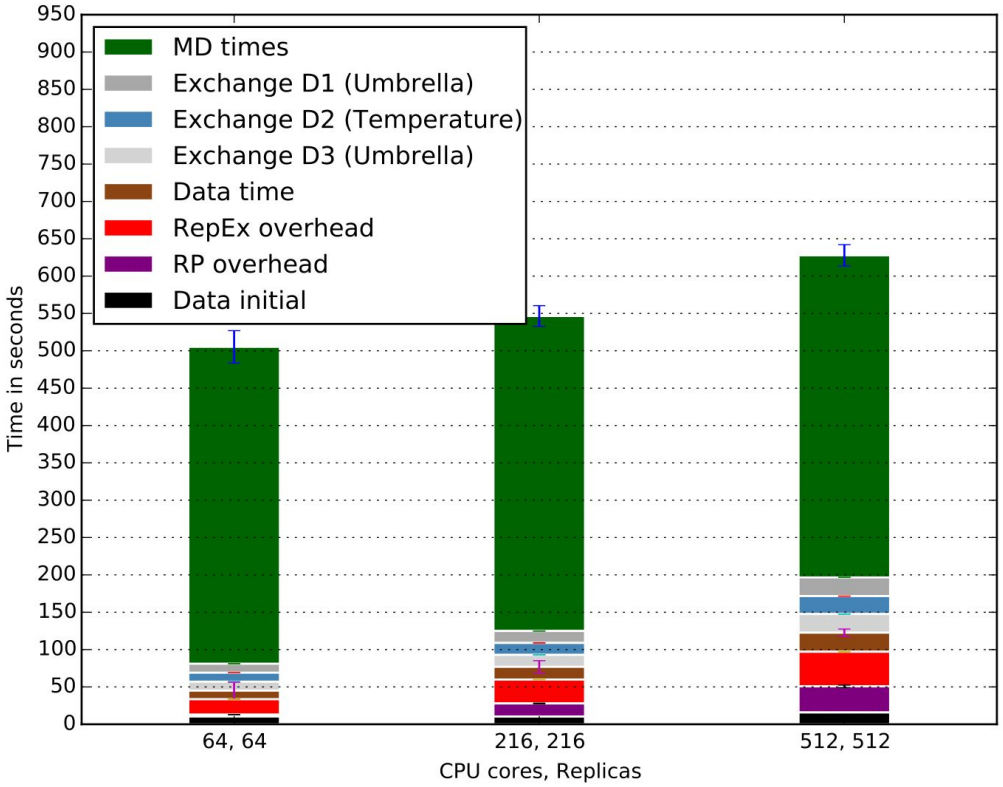
TUU-REMD simulation on Stampede HPC cluster. Averages of 4 runs with 6000 time-steps between exchanges.

Performance: Optimization





Performance: Optimization



Performance characterization

We calculate weak scaling parallel efficiency (% of linear scaling) as:

- $\mathbf{W_S = (T_1 / T_N) * 100\ \%}$

Where:

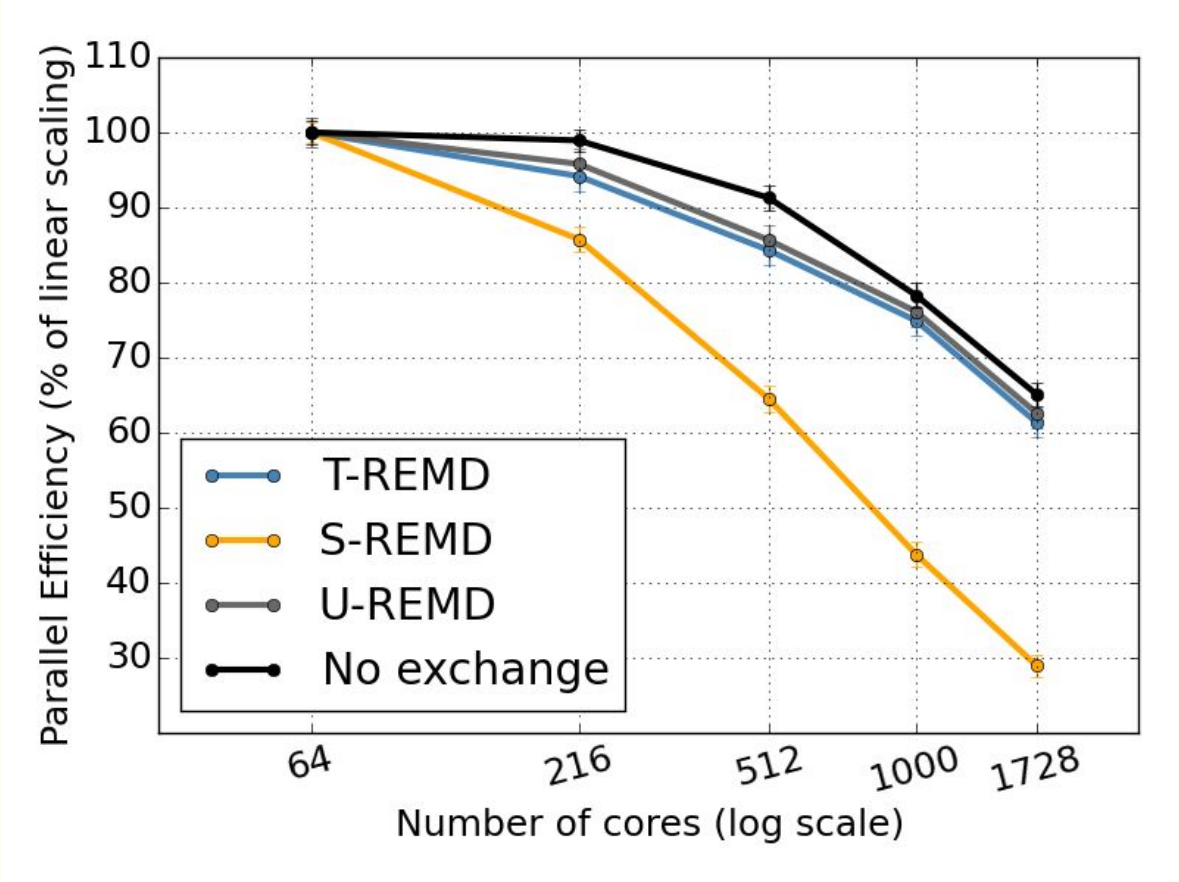
- $\mathbf{T_1}$ - time to complete simulation cycle with minimal number of replicas with number of CPU cores equal to the number of replicas (2 replicas on 2 CPUs)
- $\mathbf{T_N}$ - time to complete simulation cycle involving N replicas with N CPUs

We perform runs for:

- temperature exchange (T-REMD)
- salt concentration exchange (S-REMD)
- umbrella exchange (U-REMD)

We perform 6000 simulation timesteps between exchanges. Runs are performed on SuperMIC cluster using Amber MD engine.

Performance characterization



Resource Utilization: Quantum Mechanics / Molecular Mechanics

Resource utilization: percentage of maximal (ideal) simulation time, obtained by running MD 100 % of the time.

We define resource utilization as:

- $\mathbf{U_R = (T_S / T_{MAX}) * 100\%}$

where:

- $\mathbf{T_S}$ - total simulation time on $\boldsymbol{N}$ cores using (async/sync) RE
- $\mathbf{T_{MAX}}$ - maximal (ideal) simulation time, MD run on all cores 100% of the time

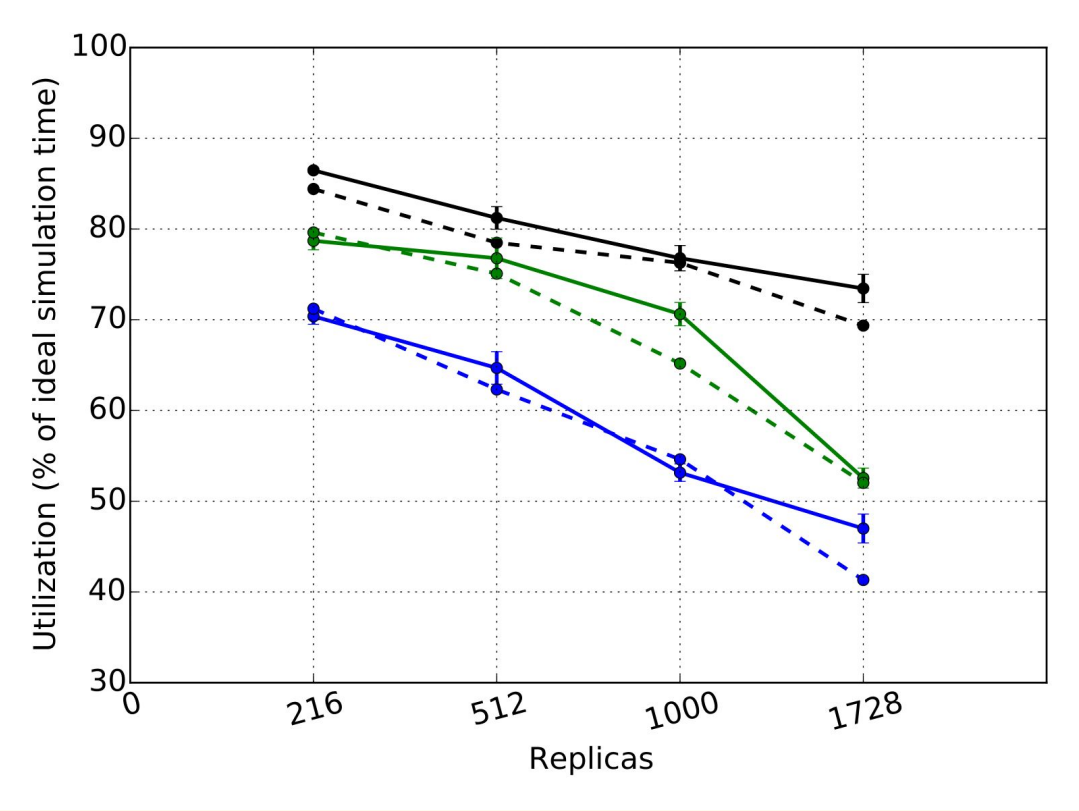
Frequency of exchanges in asynchronous RE is controlled by:

- $\mathbf{W_R = N_F / N_T}$

where:

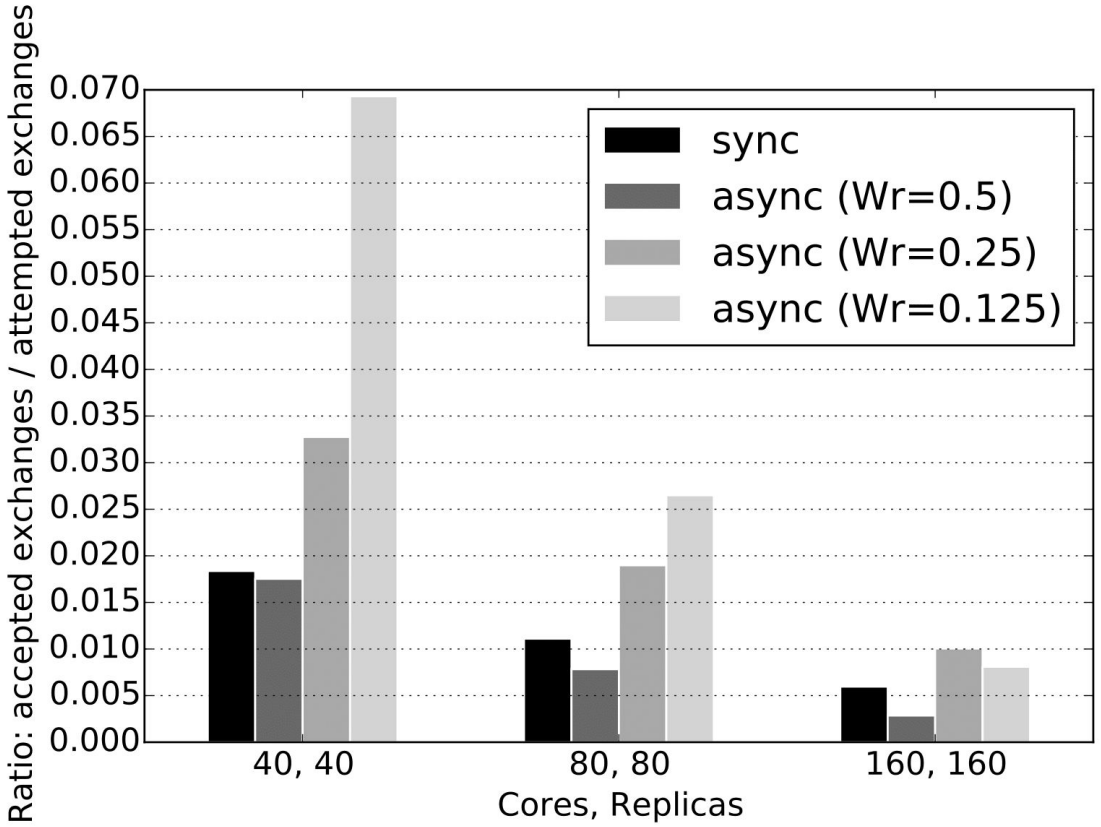
- $\mathbf{N_F}$ - number of replicas which have finished the MD phase
- $\mathbf{N_T}$ - total number of replicas

Resource Utilization: Quantum Mechanics / Molecular Mechanics



Resource utilization using **synchronous** (**solid lines**) and **asynchronous** (**dashed lines**) RE. We use **black** color for runs with **2000** simulation time-steps between exchanges, **green** color for runs with **1000** steps, and **blue** color for runs with **500** steps.

Sampling Efficiency



Ratio of accepted exchanges to attempted exchanges for 1D T-REMD with synchronous and asynchronous RE. Runs with synchronous RE (black bars), asynchronous RE with $\mathbf{W_R=0.5}$ (dark grey bars), $\mathbf{W_R=0.25}$ (light grey bars) and $\mathbf{W_R=0.125}$ (extra light gray bars).

Conclusion

- Performance is highly dependent on file staging and I/O
- Up to **1000** replicas, MD simulation time is above **45%** of total simulation time for all simulation types (including 3D simulations and QM / MM)
- Resource utilization using asynchronous RE is bounded by exchange frequency (exchange is expensive)
- Design of `RepEx` allows to:
  - Overcome barriers in the number and ordering of dimensions for multidimensional RE simulations
  - Overcome scalability limits of MD engines

Future work

- Characterization of **resource utilization** (sync. vs async.) with:
  - Different values of **resource idling**
  - Different numbers of simulation time-steps between exchanges
  - Different ratios of CPU cores to replicas
  - Different values of **Wr**
- Characterization of **sampling efficiency** for asynchronous RE and different values of $\mathbf{W_R}$:
  - Transition times
  - Relaxation time from empirical state transition matrix
  - Correlation time of the replica state index
- Explicit support for simulations concurrently utilizing multiple HPC resources
- Addition of interfaces exposing REMD capability of MD engines (Amber)
- Explicit support for multiple concurrent RE simulations (multi-level RE)
- Addition of the new exchange parameters (pH)
- Support for other MD simulation engines (GROMACS, LAMMPS, etc.)

References

1. Sugita, Yuji, and Yuko Okamoto. "Replica-exchange molecular dynamics method for protein folding." Chemical physics letters 314.1 (1999): 141-151.
2. RepEx, https://github.com/radical-cybertools/radical.repex
3. RADICAL-Pilot, https://github.com/radical-cybertools/radical.pilot
4. Chodera, John D., and Michael R. Shirts. "Replica exchange and expanded ensemble simulations as gibbs sampling: Simple improvements for enhanced mixing." The Journal of chemical physics 135.19 (2011): 194110.