KIMMDY

bwRSE4HPC Project Proposal

Assigned RSEs:	Kai Riedmiller, Glen Hunter
Project Partners:	Eric Hartmann[12], Jannik Buhr[12], Frauke Gräter[12]
Scientific field:	Physics
Subfield:	Molecular Dynamics
Duration:	6 weeks
Type of Work	HPC porting
License:	GPL
Link:	https://github.com/graeter-group/kimmdy
Programming Language:	Python
Technologies:	Tensorflow, PyTorch, CUDA
Target Cluster:	Helix

1 Project Summary

KIMMDY is a program that makes molecular dynamics simulations reactive by combining them with kinetic Monte Carlo steps. The simulations are performed using GROMACS, which utilizes GPUs for performance. The plug-in system of KIMMDY allows for users to include various reactions to the simulation, such as hydrolysis and homolysis, allowing the code to be extensible. Several plugins make use of ML frameworks, such as Tensorflow and PyTorch, to implement reaction models or force field parametrization. Currently, KIMMDY is being used on a cluster at HITS and is in a close to publication state.

2 Problem Statement

1. The installation process is currently too complicated, requiring some older python dependencies. Additionally, GROMACS patched with plumed is also required.
2. Installations should be better tested. Unit tests do exist; however, there are currently no integration tests.
3. The GPUs are currently not used properly, with two GPUs being required to run KIMMDY in its current state, while only one is utilized at a time. This suggests that a process is not freeing GPU memory. Additionally, some ML tasks do not run on the GPU.

3 Suggested Solution

1. Dependencies must first be evaluated. Since PyTorch and Tensorflow are used, a compatible environment must be identified, including CUDA and cuDNN versions. Ideally, GROMACS compiled with plumed should be available at Helix.
2. Integration tests can be developed, validating that expected results can be reproduced.
3. GPU memory should be freed after use by the ML frameworks, so GROMACS does not require an additional GPU.

[1]Heidelberg Institute for Theoretical Studies, Am Schloss-Wolfsbrunnenweg 35, Heidelberg, 69117, Germany
[2]Institute for Scientific Computing, Heidelberg University, Im Neuenheimer Feld 205, Heidelberg, 69120, Germany

4 Milestones

Week 1 - Identification of environments and packages needed. Identify what is holding the GPU memory.

Weeks 2-3 - Update required packages and environments for KIMMDY (perhaps using the setup.cfg). Apply fixes to the relevant code causing the GPU issues. Meet with project partners at the end of week 3 to report current progress.

Weeks 4-5 - Identify missing unit tests (if any) and begin their implementation. Decide on integration test approach (big-bang, top-down, bottom-up, etc) and begin construction of the test.

Week 6 - Finish test implementations. Perform final checks and submit a pull request.

5 Deliverables

The assigned RSEs aim to provide a simple installation of KIMMDY that detects compatible Python dependencies and installs any that are missing with the latest and most compatible version for all plugins. The code should be able to run on a single GPU for both GROMACS and the ML models on a Helix GPU node. Finally, an (a set of) integration test(s) for KIMMDY and corresponding plugins will be provided by the assigned RSEs.

6 Expected Contributions

The project partners will provide the necessary run files for KIMMDY, including files to run the MD simulations. Any code changes by the assigned RSEs should be reviewed and merged into the main branch of KIMMDY by the project partners when necessary. In order for the assigned RSEs to develop on Helix, the project partners must provide the Rechnenvorhaben (RV) information for their project.