Eco-Comp: Towards Responsible Computing in Materials Science

El-Tayeb Bentria[1], Sai Surag Lingampalli[2],, Fadwa El-Mellouhi[1]

[1] Qatar Environment and Energy Research Institute, Hamad Bin Khalifa University, Qatar
[2] College of Science and Engineering, Hamad Bin Khalifa University, Qatar

QEERI
معهد قطر لبحوث البيئة والطاقة
Qatar Environment & Energy Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Abstract:

Computational methods such as Density functional theory and Molecular dynamics (MD) simulations have become a key focus in material science, especially with the rise of machine learning interatomic potentials that enables the simulation of multi-million atomic systems. The computational intensity of these simulations necessitates their deployment in high-performance computers (HPCs) and the usage of multi node runs. However large number of submitted calculation by users are targeting speed regardless of efficiency. To foster sustainable and ethical computing practices. We built Eco-Comp, a user-friendly automated Python tool that allows material scientists to optimize their simulations' computing power with one command. In this study, we employed the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) to find the optimal allocation of computing resources based on the simulation input. Through the analysis of bulk metallic systems and surface reactions, we identified various factors that affected parallel efficiency. Through this, we propose rules for responsible computing in HPC architecture that Eco-Comp uses. This Poster gives a broad overview of the Eco-Comp software and its potential use for the material science community through an interactive guide.

Background:

Thousands of High-Performance Computers allocate more than 50% to atomistic research, yet few perform benchmarking before running their simulation.

How to ensure HPCs are used sustainably?

9 INDUSTRY, INNOVATION AND INFRASTRUCTURE

Methodology:

Why are atomistic simulation not benchmarked?

- New users have limited knowledge about the system architecture.
- False assumptions: more computational power = more speed
- Scheduling files reused to save time, impacting performance
- Difficult to distinguish between efficiency and CPU time taken
- Lack of awareness surrounding Responsible Computing

The goal is to build a tool that can:

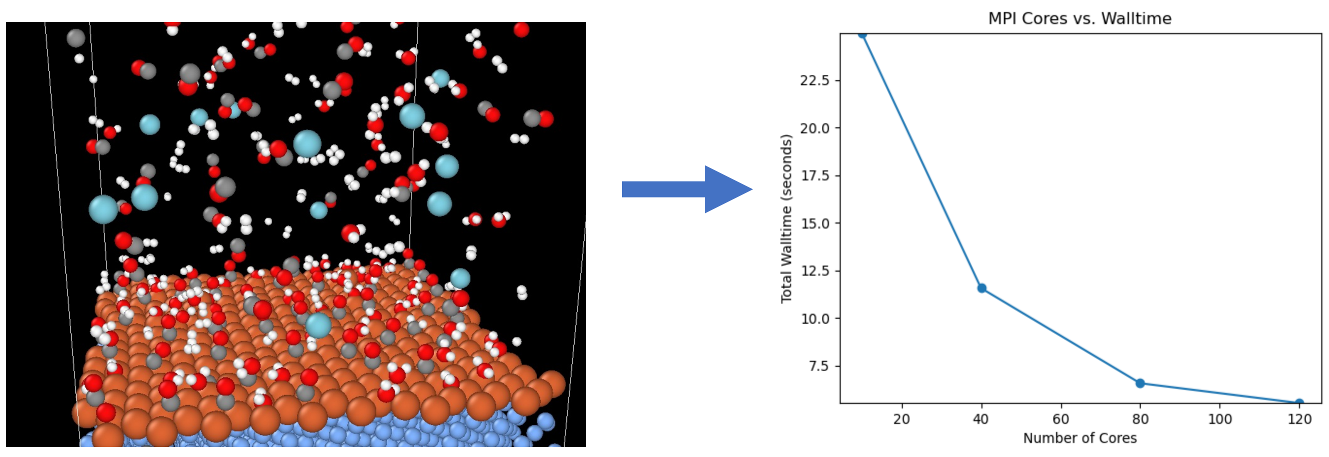
1. Read the simulation data and extract required information
2. Predict benchmark category based on system complexity
3. Run benchmarking on HPC
4. Extract & plot information to select the optimal configuration
5. Build a report and a ready-to-use job submission file

Requirements for building the tool are to understand the:

I. Impact of vacuum space in different atomic systems
II. Potentials used such as ReaxFF, ML-SNAP, etc.
III. Packages used such as Kokkos, MPI, Open-MPI
IV. Parallel efficiency of runs using various MPI cores
V. Possible bottlenecks and impact on parallel efficiency

$$Strong-scaling\ speedup\ on\ N\ nodes = \frac{Time\ on\ 1\ Node}{Time\ on\ N\ nodes}$$

$$Parallel\ Efficiency\ with\ N\ nodes\ (\%) = \frac{Strong\ scaling\ speedup\ on\ N\ nodes}{Number\ of\ nodes} \times 100$$



Acknowledgement:

This work is supported by the Qatar National Research Fund (QNRF) through the National Priorities Research Program (NPRP) under project number NPRP12S-0209-190063. The advanced computing facilities of Hamad Bin Khalifa University ("Hazeem Supercomputer") and Texas A&M University at Qatar are used for all calculations.

Contact Information:

Dr. El-Tayeb Bentria :ebentria@hbku.edu.qa

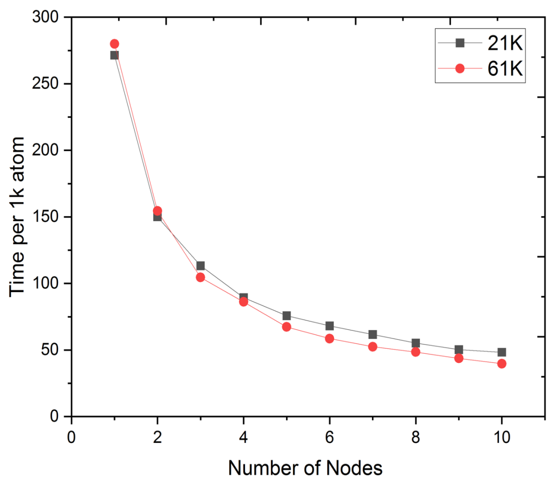
Sai Surag Lingampalli: sali35214@hbku.edu.qa

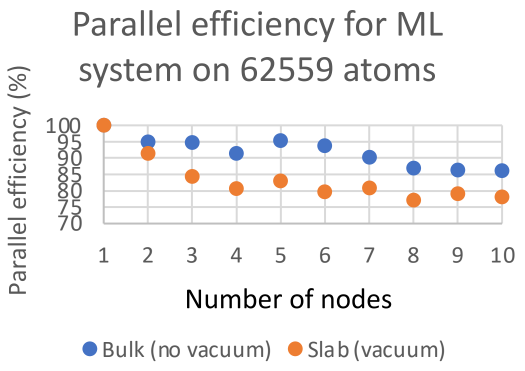
Dr. Fadwa El-Mellouhi: felmellouhi@hbku.edu.qa

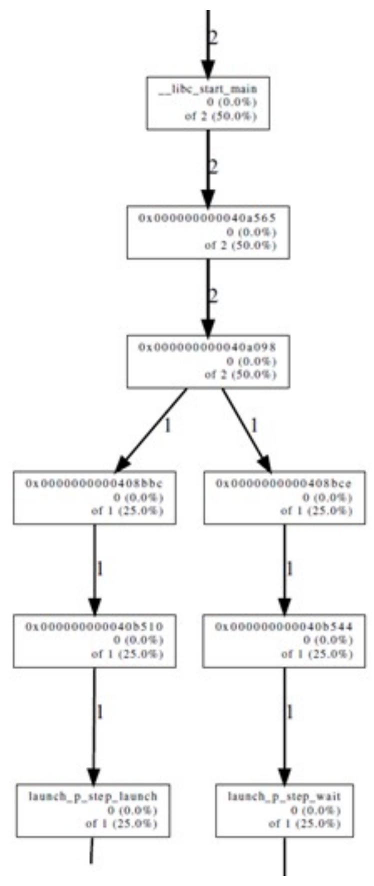
Result and Discussions:

Key takeaways:

- Performance degradation insignificant on small systems
- Kokkos package improves efficiency and speed by ~2× for large atomic systems.
- Slab system with vacuum vs. bulk system without vacuum:
  - 1.3× scale factor using ReaxFF
  - 1.45× scale factor using ML-SNAP
- 66% efficiency threshold set after profiling analysis







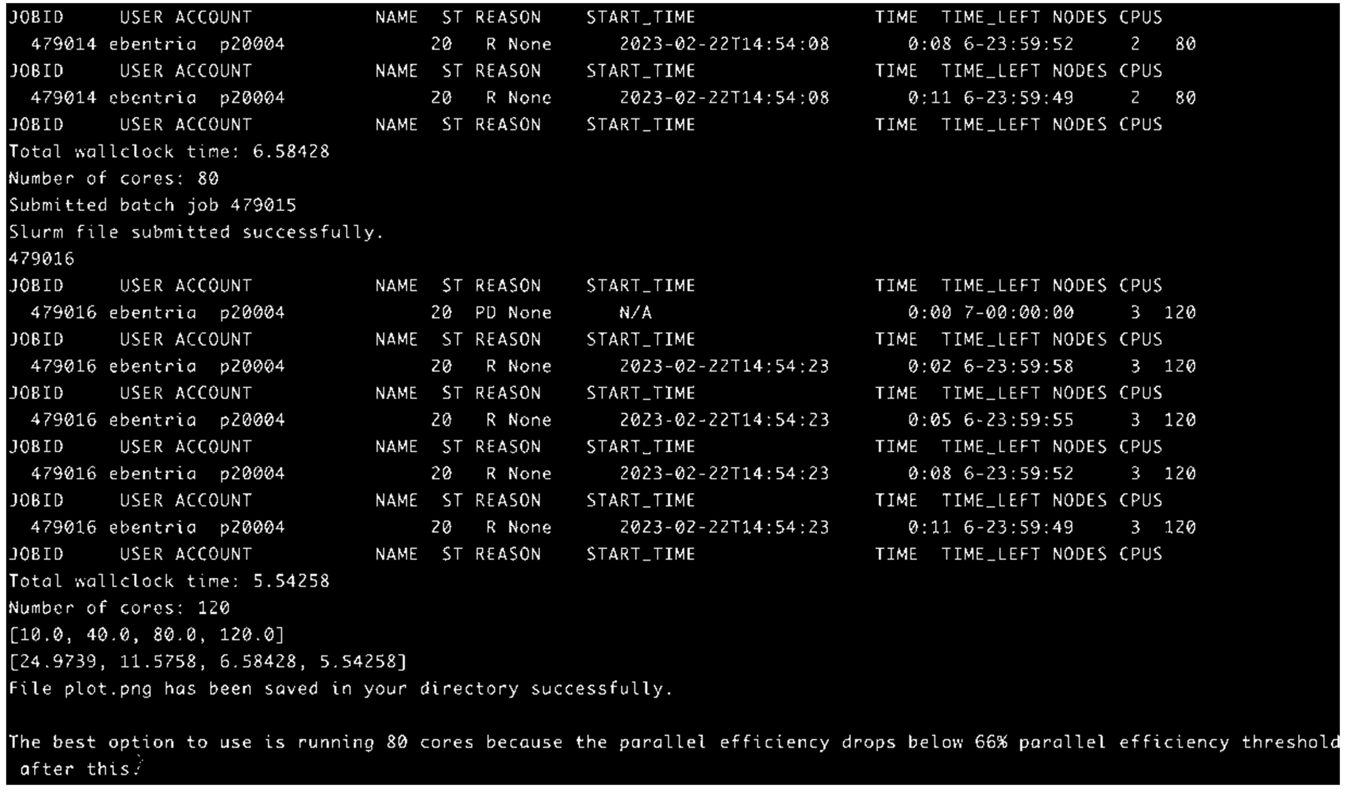
Conclusion and Future Work:

- Eco-Comp is a user-friendly Python tool that optimises simulations' computing power for materials scientists.
- User just needs to run Eco-Comp, and the tool will automatically submit jobs via schedulers and report benchmarking results for sustainable computing settings.
- If this tool gains popularity within the materials science community, the HPC footprint will be reduced considerably.
- Further efforts can be focused on incorporation into the tool, parallel programming such as OpenMP, methods such as EAM, Lennard-Jones, etc., packages such as VASP, and architectures such as GPUs.
- Raise awareness about Responsible Computing!

Outcomes:

- Eco-Comp has been elected for software copyright protection under Qatar Foundation.
- Presented at International Conference on Innovation and Technological Advances for Sustainable Development (ITAS 2023) – World Top 15 Finalist in the Arab Youth Competition.

Implementation of Eco-Comp:



```
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
  479014 ebentria  p20004            20    R None     2023-02-22T14:54:08     0:08 6-23:59:52     2   80
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
  479014 ebentria  p20004            20    R None     2023-02-22T14:54:08     0:11 6-23:59:49     2   80
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
Total wallclock time: 6.58428
Number of cores: 80
Submitted batch job 479015
Slurm file submitted successfully.
479016
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
  479016 ebentria  p20004            20   PD None      N/A                  0:00 7-00:00:00     3  120
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
  479016 ebentria  p20004            20    R None     2023-02-22T14:54:23     0:02 6-23:59:58     3  120
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
  479016 ebentria  p20004            20    R None     2023-02-22T14:54:23     0:05 6-23:59:55     3  120
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
  479016 ebentria  p20004            20    R None     2023-02-22T14:54:23     0:08 6-23:59:52     3  120
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
  479016 ebentria  p20004            20    R None     2023-02-22T14:54:23     0:11 6-23:59:49     3  120
JOBID     USER ACCOUNT          NAME  ST REASON    START_TIME            TIME  TIME_LEFT NODES CPUS
Total wallclock time: 5.54258
Number of cores: 120
[10.0, 40.0, 80.0, 120.0]
[24.9739, 11.5758, 6.58428, 5.54258]
File plot.png has been saved in your directory successfully.

The best option to use is running 80 cores because the parallel efficiency drops below 66% parallel efficiency threshold
 after this.
```

References:

1. "GPerfTools." Gperftools CPU Profiler.
2. He, Helen. "Performance Engineering of Reactive Molecular Dynamics Simulations." MIT Libraries, Massachusetts Institute of Technology.
3. K. Cha, "Performance Evaluation of LAMMPS on Multi-core Systems, "2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2013, pp. 812-819, doi: 10.1109/HPCC.and.EUC.2013.117.
4. LAMMPS molecular dynamics simulator. LAMMPS Documentation (23 Jun 2022 version). (n.d.).