

Automated high-throughput computational workflows with Taskblaster

Ask Hjorth Larsen¹, Mikael Kuisma¹, Tara M. Boland¹, Fredrik A. Nilsson¹, Kristian S. Thygesen¹

¹CAMD, Computational Atomic-Scale Materials Design,
Department of Physics, Technical University of Denmark,
2800, Kgs. Lyngby, Denmark

asklarsen@gmail.com

We present Taskblaster, a Python tool for defining and running computational workflows [1]. Taskblaster facilitates scientific project work by allowing the user to express a complex series of computations as a workflow which in turn generates a directed acyclic graph of computational tasks. Workflows and tasks are written in Python. Taskblaster uses an intuitive storage model where tasks are stored in a directory structure with the help of a sqlite database for efficient indexing. A command-line utility provides means to inspect, run, or update tasks with a high degree of automation and different computational resources. I will explain the major features and insights in Taskblaster, show examples of computational workflows for computing properties of materials, and describe how to integrate with different computational codes.

References

[1] <https://taskblaster.readthedocs.io/>
