MS051.O05

ParSeq: parallel execution of sequential data analysis applied to XAFS

Konstantin Klementiev[1]
[1]*MAX IV Laboratory, Lund, Sweden*
E-mail: konstantin.klementiev@maxiv.lu.se

ParSeq is a Python software library that provides parallel execution of sequential data analysis. The sequential aspect refers to a data analysis pipeline consisting of nodes and transforms that connect the nodes. The parallel aspect refers to simultaneous calculations on multi-task computing units applied to a large data set. The intention is to work on data sets of ~103 spectra of ~103 – 104 data points without delays in data treatment and plotting.

The following features are missing, at least partially, in other existing analysis platforms and will be implemented in ParSeq:

1. Application of any parameter represented by a corresponding GUI element, e.g. a check box or a spin box, to one or several previously selected active data. Alternatively, there is a way of applying the last action to an a posteriori selected data subset.

2. Undo and redo for all treatment steps.

3. Entering into the analysis pipeline at any node, not only at the most upstream one. This feature is useful for comparing heterogeneous data: experimental, post-processed (also saved in data libraries) and calculated ones. The latter kind of data typically appears closer to the end of an analysis pipeline whereas experimental data enter a pipeline from its start.

4. Creation of cross-data combinations, e.g. averaging or PCA, and their propagation downstream the pipeline together with the parental data. The possibility of termination of the parental data at any selected downstream node. This feature is useful for data quality assessment for each measurement channel with the simultaneous focus at the data treatment of the net (averaged multiple channels) spectra.

5. Parallel execution of data analysis on GPUs, also multiple, or multi-core CPUs.

6. Fast plotting software library capable of quick handling of ~103 curves.

The pipeline of ParSeq can be operated via Python scripts or GUI that is currently implemented in PyQt application framework. The mechanisms for creating nodes and transforms and connecting them together are given by Python scripts and thus are extensible by the user. The parallelization is achieved by utilizing OpenCL framework via pyopencl Python binding.

I present the main concepts of the software architecture of ParSeq, exemplify its application to a large data set on a laptop and brief on the status of this open-source project.

Keywords: XAFS, parallel data analysis, large data sets