The Journal of Open Source Software

LiberTEM: Software platform for scalable multidimensional data processing in transmission electron microscopy

Alexander Clausen[1], Dieter Weber[1], Karina Ruzaeva[1], Vadim Migunov[1, 3], Jan Caron[1], Rahul Chandra[2], Magnus Nord[4], Knut Müller-Caspary[1], and Rafal E. Dunin-Borkowski[1]

1 Forschungszentrum Jülich, Ernst Ruska-Centre for Microscopy and Spectroscopy with Electrons **2** Chandigarh University **3** Central Facility for Electron Microscopy (GFE), RWTH Aachen University **4** University of Antwerp, EMAT

DOI: 10.21105/joss.02006

Software

- Review
- Repository
- Archive

Editor: Mark A. Jensen

Reviewers:

- @alvarolopez
- @mamcdona77

Submitted: 16 December 2019
Published: 31 January 2020

Summary

Increases in the data rates of detectors for electron microscopy (EM) have outpaced increases in network, mass storage and memory bandwidth by two orders of magnitude between 2009 and 2019 (Weber, 2018). The LiberTEM open source platform (Clausen et al., 2019) is designed to match the growing performance requirements of EM data processing (Weber, Clausen, & Dunin-Borkowski, 2020).

Motivation

The data rate of the fastest detectors for electron microscopy that are available in 2019 exceeds 50 GB/s, which is faster than the memory bandwidth of typical personal computers (PCs) at this time. Applications from ten years before that ran smoothly on a typical PC have evolved into numerical analysis of complex multidimensional datasets (Ophus, 2019) that require distributed processing on high-performance systems. Furthermore, electron microscopy is interactive and visual, and experiments performed inside electron microscopes (so-called in situ experiments) often rely on fast on-line data processing as the experimental parameters need to be adjusted based on the observation results. As a consequence, modern data processing systems for electron microscopy should be designed for very high throughput in combination with short response times for interactive GUI use and closed-loop feedback. That requires fundamental changes in the architecture and programming model, and consequently in the implementation of algorithms and user interfaces for electron microscopy applications.

Description

The LiberTEM open source platform for high-throughput distributed processing of large-scale binary data sets is developed to fulfill these demanding requirements: Very high throughput on distributed systems, in combination with a responsive, interactive interface. The current focus for application development is electron microscopy. Nevertheless, LiberTEM is suitable for any kind of large-scale binary data that has a hyper-rectangular array layout, notably data from synchrotrons and neutron sources.

LiberTEM uses a simplified MapReduce (Dean & Ghemawat, 2008) programming model. It is designed to run and perform well on PCs, single server nodes, clusters and cloud services. On clusters it can use fast distributed local storage on high-performance SSDs. That way it achieves very high aggregate IO performance on a compact and cost-efficient system built from stock components. On a cluster with eight microblade nodes we could show a mass storage throughput of 46 GB/s for a virtual detector calculation.

LiberTEM is supported on Linux, Mac OS X and Windows. Other platforms that allow installation of Python 3 and the required packages will likely work as well. The GUI is running in a web browser.

Based on its processing architecture, LiberTEM offers implementations for various applications of electron microscopy data. That includes basic capabilities such as integrating over ranges of the input data (virtual detectors and virtual darkfield imaging, for example), and advanced applications such as data processing for strain mapping, amorphous materials and phase change materials. More applications will be added as development progresses.

Compared to established MapReduce-like systems like Apache Spark (Zaharia et al., 2016) or Apache Hadoop (Patel, Birla, & Nair, 2012), it offers a data model that is similar to NumPy (Walt, Colbert, & Varoquaux, 2011), suitable for typical binary data from area detectors, as opposed to tabular data in the case of Spark and Hadoop. It includes interfaces to the established Python-based numerical processing tools, supports a number of relevant file formats for EM data, and features optimized data paths for numerical data that eliminate unnecessary copies and allow cache-efficient processing. As a result, it can reach more than four times the throughput of Spark-based processing for the same operations on typical data sets.

Compared to tools such as Dask arrays for NumPy-based distributed computations (Rocklin, 2015), LiberTEM is developed towards low-latency interactive feedback for GUI display as well as future applications for high-throughput distributed live data processing. As a consequence, data reduction operations in LiberTEM are not defined as top-down operations like in the case of Dask arrays, but as bottom-up streaming operations that work on small portions of the input data. Since LiberTEM can work efficiently on smaller data portions that fit into the L3 cache of typical CPUs, it can be more than a factor of two faster than equivalent implementations based on Dask arrays.

When compared to Dask arrays that try to emulate NumPy arrays as closely as possible, the data and programming model of LiberTEM is more rigid and closely linked to the way how the data is structured and how reduction operations are performed in the back-end. That places restrictions on the implementation of operations, but at the same time it is easier to understand, control and optimize how a specific operation is executed, both in the back-end and in user-defined operations. In particular, it is easier to implement complex reductions in such a way that they are performed efficiently with a single pass over the data.

The main focus in LiberTEM has been achieving very high throughput, responsive GUI interaction and scalability for both offline and live data processing. These requirements resulted in a distinctive way of handling data in combination with an adapted programming model. Ensuring compatibility and interoperability with other solutions like Gatan Microscopy Suite (GMS), Dask, HyperSpy (Peña et al., 2019), pyXem (Johnstone et al., 2019), pixStem and others is work in progress. They use a similar data model, which makes integration possible, in principle. As an example, LiberTEM can be run from within development versions of an upcoming GMS release that includes an embedded Python interpreter, and it can already generate efficient Dask.distributed arrays from the data formats it supports.

The online documentation with more details on installation, use, architecture, applications, supported formats, performance benchmarks and more can be found at https://libertem.github.io/LiberTEM/.

Live data processing for interactive microscopy and automation of experiments is currently under development. The architecture and programming model of LiberTEM are already

developed in such a way that current applications will work on live data streams without modification as soon as back-end support is implemented.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreements No 780487 - VIDEO and No 856538 - 3D MAGiC).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 686053 - CritCat and No 823717 – ESTEEM3.

We gratefully acknowledge funding from the Initiative and Networking Fund of the Helmholtz Association within the Helmholtz Young Investigator Group moreSTEM under Contract No. VH-NG-1317 at Forschungszentrum Jülich in Germany.

We gratefully acknowledge funding from the Information & Data Science Pilot Project "Ptychography 4.0" of the Helmholtz Association.

We kindly acknowledge funding from Google Summer of Code 2019 under the umbrella of the Python software foundation.

STEMx equipment and software for 4D STEM data acquisition with K2 IS camera courtesy of Gatan Inc.

Forschungszentrum Jülich is supporting LiberTEM with funding for personnel, access to its infrastructure and administrative support.

Furthermore, we wish to thank a large number of people who contributed in various ways to LiberTEM. We maintain a full and continuously updated list of creators and contributors online.

References

Clausen, A., Weber, D., Ruzaeva, K., Migunov, V., Caron, J., Chandra, R., Nord, M., et al. (2019). LiberTEM/libertem: 0.3.0. Zenodo. doi:10.5281/ZENODO.3572855

Dean, J., & Ghemawat, S. (2008). MapReduce. *Communications of the ACM*, *51*(1), 107. doi:10.1145/1327452.1327492

Johnstone, D. N., Crout, P., Høgås, S., Martineau, B., Smeets, S., Joonatan Laulainen, Collins, S., et al. (2019). Pyxem/pyxem: PyXem 0.9.2. Zenodo. doi:10.5281/ZENODO.2649351

Ophus, C. (2019). Four-dimensional scanning transmission electron microscopy (4D-STEM): From scanning nanodiffraction to ptychography and beyond. *Microscopy and Microanalysis*, *25*(3), 563–582. doi:10.1017/s1431927619000497

Patel, A. B., Birla, M., & Nair, U. (2012). Addressing big data problem using Hadoop and Map Reduce. In *2012 nirma university international conference on engineering (NUiCONE)*. IEEE. doi:10.1109/nuicone.2012.6493198

Peña, F. D. L., Prestat, E., Fauske, V. T., Burdet, P., Jokubauskas, P., Nord, M., Ostasevicius, T., et al. (2019). Hyperspy/hyperspy: HyperSpy v1.5.2. Zenodo. doi:10.5281/ZENODO.3396791

Rocklin, M. (2015). Dask: Parallel computation with blocked algorithms and task scheduling. In *Proceedings of the 14th python in science conference*. SciPy. doi:10.25080/majora-7b98e3ed-013

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. doi:10.1109/mcse.2011.37

Weber, D. (2018). Development of IT system and TEM camera performance. Zenodo. doi:10.5281/zenodo.2450624

Weber, D., Clausen, A., & Dunin-Borkowski, R. (2020). Handbook on Big Data and machine learning in the physical sciences. In K. K. Van Dam, K. Yager, S. Campbell, R. Farnsworth, & M. van Dam (Eds.), (Vol. Volume 2: Advanced analysis solutions for leading experimental techniques). World Scientific. doi:10.1142/11389

Zaharia, M., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., Stoica, I., Xin, R. S., et al. (2016). Apache Spark. *Communications of the ACM*, *59*(11), 56–65. doi:10.1145/2934664