doi:10.1017/S1431927622010911
Microsc. Microanal. 28 (Suppl 1), 2022


# Seamless Communication Between High-Performance Computing System and Electron Microscopes for On-Demand Automated Data Transfer and Remote Control

Debangshu Mukherjee[1*], Anees Al-Najjar[1], Kevin M. Roccapriore[2], Jacob D. Hinkle[1], Andrew R. Lupini[2], Chris Meyer[3], Sergei V. Kalinin[2], Olga S. Ovchinnikova[1] and Nageswara S. Rao[1]

[1.] Computational Sciences & Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN, United States.
[2.] Center for Nanophase Materials Sciences, Oak Ridge National Laboratory, Oak Ridge, TN, United States.
[3.] NION R&D, Kirkland, WA, United States.
* Corresponding author: mukherjeed@ornl.gov

A quarter-century back, anticipating the capabilities of the next generation of electron microscopes, Mick Brown memorably coined the phrase "A synchrotron in a microscope" [1]. Since then, with newer modalities such as 4D-STEM and momentum-resolved EELS becoming increasingly available, the flood of data coming out of electron microscopes has increased by several orders of magnitude [2-4]. As a result of these advances, modern transmission electron microscopy has increasingly become an all-digital and data driven. Currently installed high-speed direct electron detectors can capture tens of thousands of frames in a single second and thus, at full throttle, can produce terabytes of data in a single session [5]. This problem is especially acute in the physical sciences applications, where microscopes and high-performance computing have operated in separate, disconnected silos due to historical reasons.

Even something as seemingly trivial as data transfer and storage is a challenge. Consequently, electron microscopy and its corresponding data analysis often have been two discrete steps performed sequentially. Thus, the sequence of events usually occurs in discrete steps: microscope operation/data collection, data transfer to a separate analysis computer, and subsequent data analysis. Each step in this pipeline is time-consuming and happens on disconnected systems. Thus, feedback loops to steer microscopes based on the analysis results can be extremely challenging to implement or are limited to less data-intensive experiments [6].

In this paper, we demonstrate a prototype solution we implemented at ORNL. We connected transmission electron microscopes across the campus to high-performance compute clusters in data centers for real-time data transfer, analysis, and feedback loops. We implemented our solution on NION microscope systems using the Swift microscope software which provide open Application Programming Interfaces (APIs) and a scriptable Python console [7]. We remote mounted the Network-attached Storage (NAS, Synology DS200) connected to the control computer to automatically transfer data to data centers connected over campus network. The control computer is a Windows machine, as the control software is limited only to that operating system. The NAS systems are visible to Windows systems on the ORNL internal network, and are samba mounted on Linux-based GPU workstation located in the data center. When operating the microscope from the control computer, the data is directly collected on the NAS and thus is visible transparently to the workstation post-acquisition without requiring any manual inputs. A Jupyter notebook is run on the workstation to control the microscope acquisition. We incorporate the Remote Procedure Call (RPC) paradigm as a distributed software communication to send the microscope controls remotely. Python Pyro library is a type of RPC leveraged in our implementation through running a Pyro server as a daemon inside the NION Swift computer to expose the microscope controls across the network and running the Pyro client from the Jupyter notebook at the workstation to access these controls. As can be observed in **Figure 1**, the networking set up here consists of three distinct parts – *Control Plane,* which controls the microscope itself; *Data Plane,* where microscopy data is transferred through gigabit ethernet and the *Management Plane*, where the status of an ongoing job is communicated back to its parent process.

Thus, in this setup, the operator, the microscope, and the analysis computer can all be in geographically separate areas yet can talk to each other and transfer data and commands in real-time with no operator involvement. Thus, the workflow is executed as follows: the operator starts a Jupyter session on the data center and enters the desired microscope operation, with the notebook session itself operable from any system on the network with verified credentials, as visualized through process 1 in **Figure 1**. This operation script is transmitted through Pyro client to server running on NION Swift on the microscope control computer, as shown by process 2 in **Figure 1**. Swift running on the control computer executes this script through processes 3 and 4. The output data from the microscope detectors is streamed to the NAS (process 5), with the state of the data collection communicated to Swift (process 6). Since the NAS is also samba mounted on the cluster, this data is also available immediately, as shown through process 7. As the data is being collected on the NAS, the cluster starts processing the data, with the results being available to the operator as outputs in the Jupyter session. Suppose the operator decides on performing an automated experiment. In that case, the analysis results can be used directly to guide the next experiment, again through the Pyro communication channel until a pre-determined endpoint is reached, thus enabling the whole feedback loop within this workflow.

The solution that we have thus demonstrated here is scalable, from a single GPU workstation in a data center to multiple workstations working together in parallel, all the way to a supercomputing cluster. Additionally, this solution enforces a separation between the microscope control computer and the analysis computer, keeping the control and processing parts separate and utilizing the superior computing power available at remote systems. We envision future work in this avenue to completely abstract away the underlying network to the end user, with the end user operating Jupyter notebooks and getting access to computed data in real-time [8].

https://doi.org/10.1017/S1431927622010911 Published online by Cambridge University Press

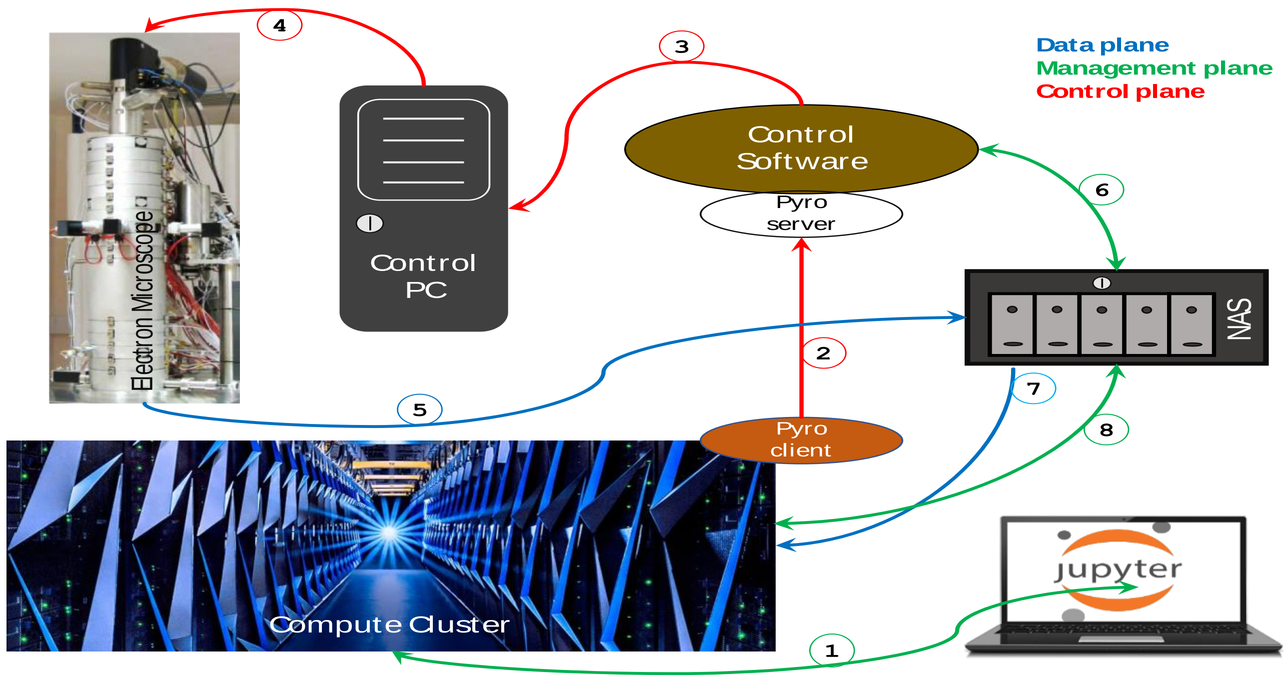


**Figure 1.** Interconnectivity workflow between the microscope and the computational facilities. The connectivity can be split into three distinct planes – *data* (moving around the microscope data), *management*, and *control* (commands to the microscope for operation).

References:

[1] LM Brown in "Electron Microscopy and Analysis 1997" (CRC Press, 1997) p. 17. DOI: 10.1201/9781003063056-4
[2] C Ophus, Microscopy and Microanalysis **25** (2019) p. 563. DOI: 10.1017/S1431927619000497
[3] FS Hage et al., Science Advances **4** (2018), p. eaar7495. DOI: 10.1126/sciadv.aar7495
[4] SR Spurgeon et al., Nature Materials **20** (2021), p. 274. DOI: 10.1038/s41563-020-00833-z
[5] P Ercius et al., Microscopy and Microanalysis **26** (2020), p. 1896. DOI: 10.1017/S1431927620019753
[6] KM Roccapriore et al., arXiv:2112.04479 (2021).
[7] C Meyer et al., Microscopy and Microanalysis **25** (2019), p.122. DOI: 10.1017/S143192761900134X
[8] Research is supported by ORNL's Laboratory Directed Research and Development Program (INTERSECT Initiative), managed by UT-Battelle, LLC for the U.S. Department of Energy (DOE). Electron microscopy was conducted as part of a user proposal at Oak Ridge National Laboratory's Center for Nanophase Materials Sciences (CNMS), which is a U.S. DOE Office of Science User Facility.