doi:10.1017/S1431927619007669
Microsc. Microanal. 25 (Suppl 2), 2019

Collaborative Editing and Distributing Large Image-Based Data for Connectomics

William T. Katz[1*], Stuart E. Berg[1] and Stephen K. Plaza[1]

[1.] FlyEM Team, Janelia Research Campus, Ashburn, VA, USA.
* Corresponding author: katzw@janelia.hhmi.org

Advances in biological imaging are resulting in much larger datasets and it's likely that the pace will continue to accelerate. In EM-based connectomics, we turn first to automated methods for segmentation like deep learning approaches to interpret that flood of data [1]. Although segmentation methods continue to improve, there is a non-negligible error rate that requires human input to increase accuracy. The scale of imagery dictates that only a small percentage of the dataset is moderated by humans, and after initial work by large groups, those interested in continued accuracy improvement can be distributed throughout the world in research groups focused on particular neural circuits.

The FlyEM Team at Janelia has produced some of the largest dense reconstructions to date [2] and we are currently pursuing much larger reconstructions that will cover the entire fruit fly brain. New approaches are needed to allow efficient access, distribution, and collaboratively editing of these massive datasets, and we believe open-source software development tools like git and community websites like github.com provide a template. We have created the open-source DVID (Distributed, Versioned, Image-Oriented Dataservice) system and associated tools as the first steps toward a distributed versioning approach to FIB-SEM connectomics datasets [3].

DVID provides a high-level Science API and translates requests through a number of data type modules into underlying key-value pairs (Figure 1). The use of a Science API frees clients from duplicating domain-specific calculations like how to merge multiple label fragments or extract a 2D slice of a given orientation from a 3D volume. Also, by shielding the clients from how the data is actually held, we can tailor storage solutions to a range of users from the postdoc with just a laptop to a large institutional cluster. DVID persists data as key-value pairs and is extremely flexible in how it stores data across different underlying key-value datastores. For example, the aligned grayscale image volume, which tends to be the largest use of storage, can be handled by petabyte-capable cloud services like Google Cloud Storage. The highly compressible segmentation, which is frequently mutated during proofreading, is typically stored in very low-latency Non-Volatile Memory Express (NVMe) solid-state drives.

Branched versioning (modeled as a directed acyclic graph or DAG) is a central feature of DVID. Data at any version is fully readable and also writable at uncommitted leaf versions. We will present a number of advantages of branched versioning including proofreader training and efficient data handling due to the immutability of committed versions. We are also developing ways to reintegrate remote changes (Figure 2) and obtain requested data either locally or from remote services.

DVID currently serves as our primary data service, handling low-latency requests to support interactive proofreading. It is supported by a number of tools that allows offloading heavy computation to a compute cluster before storing the results back into DVID. These offloaded computationally-expensive jobs include label compression and indexing during ingestion, neuron skeleton and mesh construction, and the image processing required to produce multi-scale image pyramids. DVID can also publish events to a Kafka distributed logging system, which allows other tools to subscribe to data changes.

https://doi.org/10.1017/S1431927619007669 Published online by Cambridge University Press

Given the above features, we will present the open-source DVID ecosystem as a flexible way to store, move, access, and collaboratively edit large image-based datasets.

References:

[1] M Januszewski, Nat. Methods **15** (2018), p. 605.

[2] SY Takemura et al., eLife **6** (2017), p. e26975.

[3] WT Katz and SM Plaza, Front. Neural Circuits **13** (2019), p. 5.

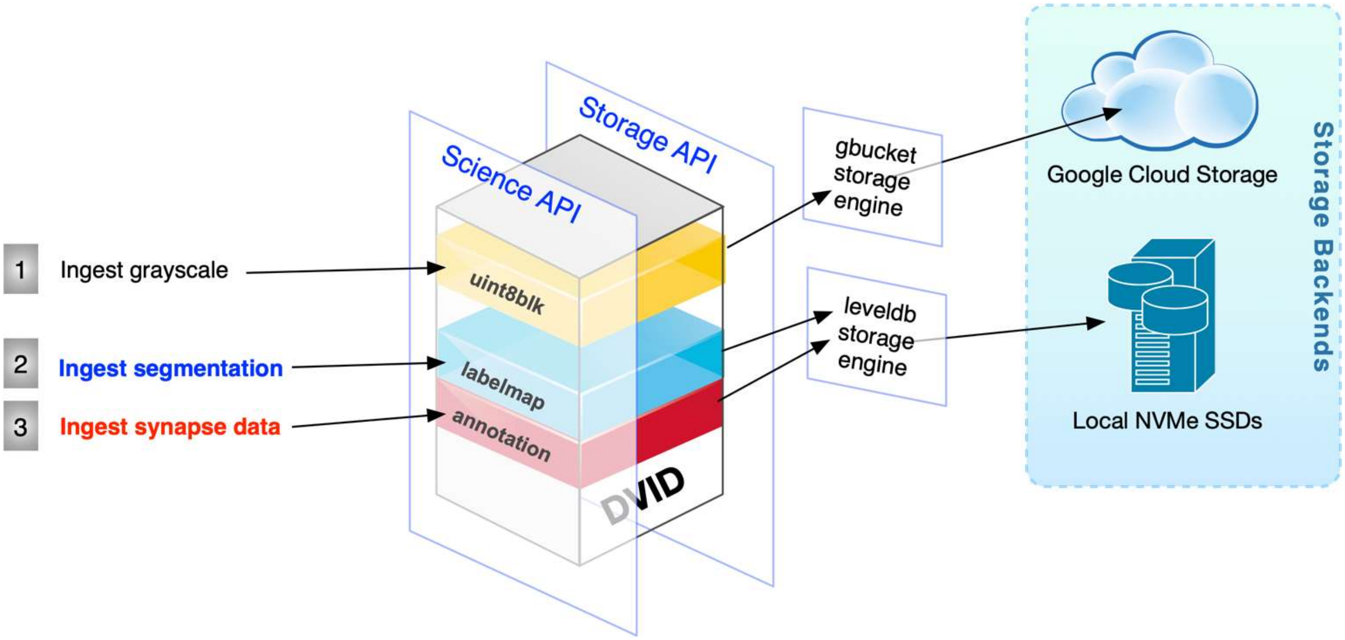

Figure 1. High-level view of DVID. Data types uint8blk, labelmap, annotation provide a Science API for different types of data like grayscale, segmentation, and 3d point data, respectively. These data type packages transform data into key-value pairs and can store them in a variety of key-value stores from petabyte-scale cloud services (Google Cloud Storage) to embedded databases on local solid-state drives.

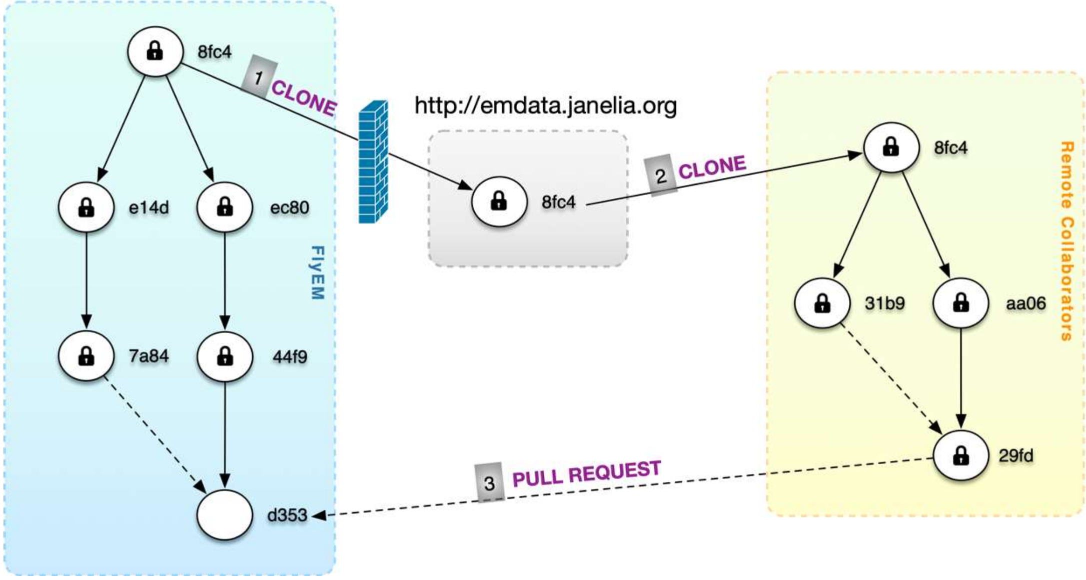

Figure 2. Branched versioning allows cloning a committed version (8fc4) to a public server. Work in progress: remote collaborators could clone the available data to their own computer, make modifications, and then initiate a request to merge the changes back, either to the public server or a private repository.

https://doi.org/10.1017/S1431927619007669 Published online by Cambridge University Press