Contribution ID: **7** Type: **Poster**

SciFi: An Embedded Asset Store to Organize and Optimize Data Files for Machine Learning

Monday, 19 September 2022 18:00 (3 hours)

Data in machine learning scenarios is typically scattered over a large amount of files. This comes with a number of undesired side effects. First, operating systems are not designed for storing thousands of files in a flat file system. As a result, a simple scan of a directory does not terminate anymore in the worst case. Implicitly called operations like user name resolution and sorting increase the execution time of a scan significantly in the case of thousands of files. Second, storing small files wastes disc space. A file always occupies at least one disc cluster. Hence, all files which are smaller than a disc cluster, block space which is not used. Further, whenever metadata is involved, the connection between metadata and the stored files has to be implemented by the scientist. This leads to a development overhead for each individual dataset and application. Finally, the processes of storing and sharing data are increasingly inefficent the more files and individual scipts are involved.

For these reasons, digital asset management systems (DAMS) are already popular in other fields, such as photography or music. However, DAMS are hardly used in science, mainly due to a lack of available systems. To close this gap, we present *ScienceFiles (SciFi)*, an embedded DAMS specially developed for scientific data. It combines a traditional relational database and a key-value store to store data and query metadata efficiently. *SciFi* consists of an extensible framework and a shell which serves as a stand-alone DAMS. For providing access to the data stored in *SciFi* for machine learning, we extended the Dataset class of *PyTorch*. To ensure the usability of our solution, we chose a lightweight design that runs on laptops and lab PCs without requiring special permissions or an installation.

In detail, *SciFi* provides the following features and advantages over exclusively using a file system:

- Significant reduction of files on disc, i.e. all data can be stored in one file
- Backup peformance increased by up to two orders of magnitude
- Disc space usage reduced by up to 75%
- Filter data by metadata without additional scipts and return the according subset of the data
- Multiple write modes for intermediate data: disc, temporary file system in main memory, return value via API
- Access in *Pytorch* via DataLoader

Primary authors: UNGETHÜM, Annett (Universität Hamburg); POPPINGA, Martin (Universität Hamburg)

Co-author: RAREY, Matthias (Universität Hamburg)

Presenter: POPPINGA, Martin (Universität Hamburg)

Session Classification: Poster Session

Track Classification: Poster