

# Are there ways to improve performance when accessing a file that has many groups and many attributes?

If you have performance problems accessing an HDF5 file with many groups and many attributes, there are some things you can do:

1. Make sure you set `H5Pset_libver_bounds(...,H5F_LIBVER_LATEST, H5F_LIBVER_LATEST)` when creating the file such that the newest library features are used.
2. Make sure you close objects that you have opened and no longer need to access.
3. Compress the group metadata, which may reduce disk reading times (some high-speed compressor such as LZ4 is recommended)
4. Use the split file driver to place metadata into different physical files than the raw data, so metadata is always compact on disk.
5. Reorganize your attributes into datasets instead. When appending data it may be less inefficient to update the datasets, so during an incremental data update you can still write attributes, then do some postprocessing that builds a "cache" of the attributes into a dataset instead, and during reading you read that dataset instead of the attributes. This postprocessing to build the "attribute cache" would take some time, but if it needs to be done once only while reading happens frequently, it is worth the effort. This would depend on the use case scenario.