Dataset gets filtered twice when creating filtered chunk dataset

USER: When I create a filtered chunked dataset, I noticed that the dataset gets filtered twice: when the empty dataset gets created, and when it gets written.

For efficiency, I would like to remove the first filter call (dataset creation) and if possible, avoid the initial disk allocation with empty data. I've tried options such as H5D_ALLOC_TIME_INCR, but without success so far.

I'm using the C++ wrappers of HDF5 1.10.3 with collective calls.

Answer: You should consider precreating the dataset (in serial) on rank 0, and then re-open in parallel. You can avoid writing fill values via H5Pset_fill_time(dcpl, H5D_FILL_TIME_NEVER).

USER:

I've implemented what you suggested.

The sequential file and dataset creation works as expected. The filter is not called at this stage.

I reopen the file collectively, and the the dataset like this:

```
H5::FileAccPropList fileAccPropList;
H5Pset_fapl_mpio(fileAccPropList.getId() , comm, info);
H5Pset_all_coll_metadata_ops(fileAccPropList.getId() , true);
H5::H5File h5file("test.h5", H5F_ACC_RDWR, H5::FileCreatPropList::DEFAULT, fileAccPropList);
H5::DSetMemXferPropList xfer_plist;
H5Pset_dxpl_mpio(xfer_plist.getId() , H5FD_MPIO_COLLECTIVE);

H5::DataSet dataset = h5file.openDataSet(dataset_name);
```

Unfortunately, the openDataSet calls the filter to compress data. I don't know how to avoid this call.

Answer: I believe that there is currently no way to get around the first call to the filter.

If you look in the document Fill Value and Dataset Storage Allocation Issues in HDF5 under section VI, the first part of the first bullet point mentions that certain VFDs (such as the MPI-I/O one) require that the space for the dataset be allocated upon creation time. From what I understand, it seems that in your case the library is realizing upon dataset open that the space wasn't allocated. At that point, the MPI-I/O driver forces the space to be allocated, the side effect of which will be that the chunks in the dataset get filtered during allocation time. In this manner, later reads of those chunks will see filtered data as expected, instead of unfiltered data.

You may be able to optimize by using your suggestion of testing the buffer, but in that case, if fill values weren't written, you may not be guaranteed that the data buffer for the chunk is actually zero-filled.