Information on the metadata cache

The metadata cache exists to cache metadata for an entire HDF5 file, and exists as long as the file is open.

As the working set size for HDF5 files varies widely depending on both structure and access pattern, it is necessary to add facilities for cache size adjustment under either automatic or "manual" control.

Structurally, the metadata cache can be thought of as a heavily modified version of the UNIX buffer cache as described in chapter three of M. J. Bach's "The Design of the UNIX Operating System". In essence the UNIX buffer cache uses a hash table with chaining to index a pool of fixed size buffers. It uses the LRU replacement policy.

Since HDF5 metadata entries are of no fixed size, and may grow arbitrarily large, the size of the metadata cache cannot be controlled by setting a maximum number of entries. Instead the cache keeps a running sum of the size of all entries, and will attempt to evict entries as necessary to stay within the specified maximum size. Candidates for eviction are chosen by the LRU replacement policy, and a LRU list is maintained for this purpose.

The cache cannot evict entries that are locked, and thus it will temporarily grow beyond its maximum size if there are insufficient unlocked entries to evict.

To help avoid generating writes in response to a read while running in parallel, the cache also maintains a clean LRU list. This list contains only clean entries, and is used as a source of candidates for eviction when servicing a read request in parallel mode. If the clean LRU list is exhausted, the cache will temporarily exceed its specified maximum size.

To increase the likelihood that this will not happen, the cache allows the user to specify a minimum clean size -- which is a minimum total size of all the entries on the clean LRU. Note that the clean LRU list is only maintained in the parallel version of the HDF5 library, and thus that the minimum clean size is only relevant when running the parallel version of the library.