OSD Persistence Strawman 04-005r0
Notes on FUA Option

John Muth, VERITAS Software

23 Sept 2003 (Updated 16 Oct 2003)
---

The abstract persistence model for OSD contains a two level memory hierarchy, Volatile Cache and Stable Storage (SS). Stable Storage is memory that survives non-catastrophic failure of the OSD such as a crash and restart. Volatile Cache is memory that is lost if the OSD crashes. This model is not meant to constrain OSD implementations. Individual OSD implementations are free to use whatenver technologies they choose to implement stable storage. For example, an OSD could choose to implement stable storage as a combination is NVRAM and disk devices. Volatile Cache on an OSD is optional.

The FUA bit (Force Unit Access) in the Options Byte controls whether or not the results of an operation must be committed to Stable Storage (SS) in the target OSD before success is returned to the initiator. FUA=1 requires that updates be committed to stable storage before the OSD returns success. FUA=0 allows the OSD to return success for updates that are only contained in Volatile Cache, although an OSD implementation is free to commit FUA=0 updates to stable storage. The FUA bit effects both object data and object attributes.

The following commands define the Options Byte:

> APPEND,CREATE, CREATE_AND_WRITE, CREATE_COLLECTION, CREATE_PARTITION, FORMAT_OSD, READ, REMOVE, REMOVE_COLLECTION, REMOVE_PARTITION, WRITE

All compliant OSD devices must support both FUA=0 and FUA=1 for these commands.

For operations other than READ, if FUA=1 then the operation in question must be committed to Stable Storage before success can be returned to the caller. If FUA=0 then the OSD may or may not commit the operation to SS before returning success to the initiator.

The READ operation always returns the result of the most recent WRITE operation. READ with FUA=1 requires that any data returned has been flushed to stable storage before the READ operation returns success. READ with FUA=0 does not require that returned data has been flushed to stable storage.

FLUSH_OBJECT forces updates to an object to be committed to stable storage before success is returned to the initiator.

If an error occurs during processing of command, partial results may be written to an OSD's persistent storage. It is up to higher level software to destect and correct the error.