File Double Locking

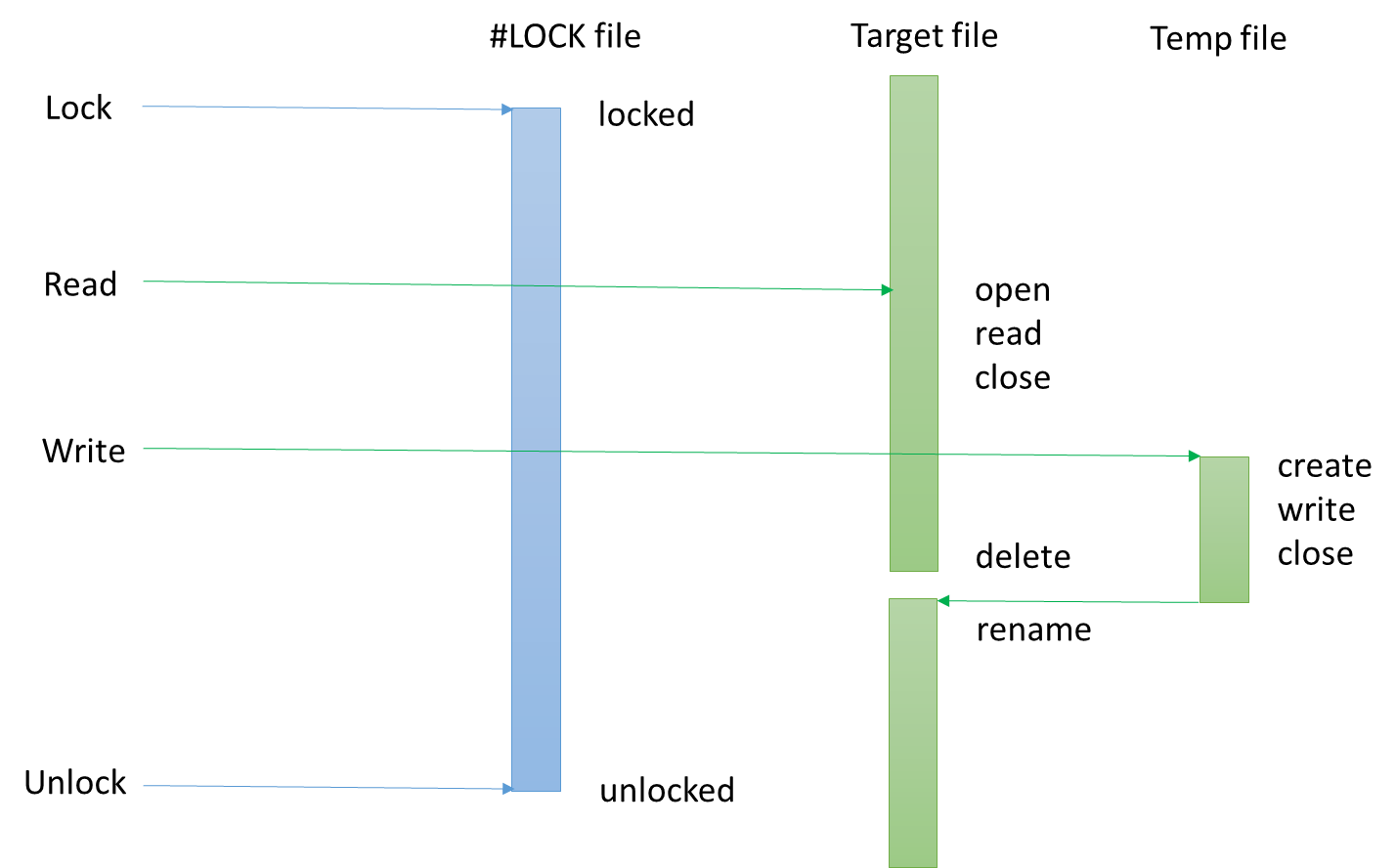
Keith D Swenson

Aug 13, 2018

This document describes the extra locking that is done in an attempt to address a low reliability file system, and to properly run reading and writing files on that.

# Previous

Previously, the LockableJSONFile class would implement the following locking:



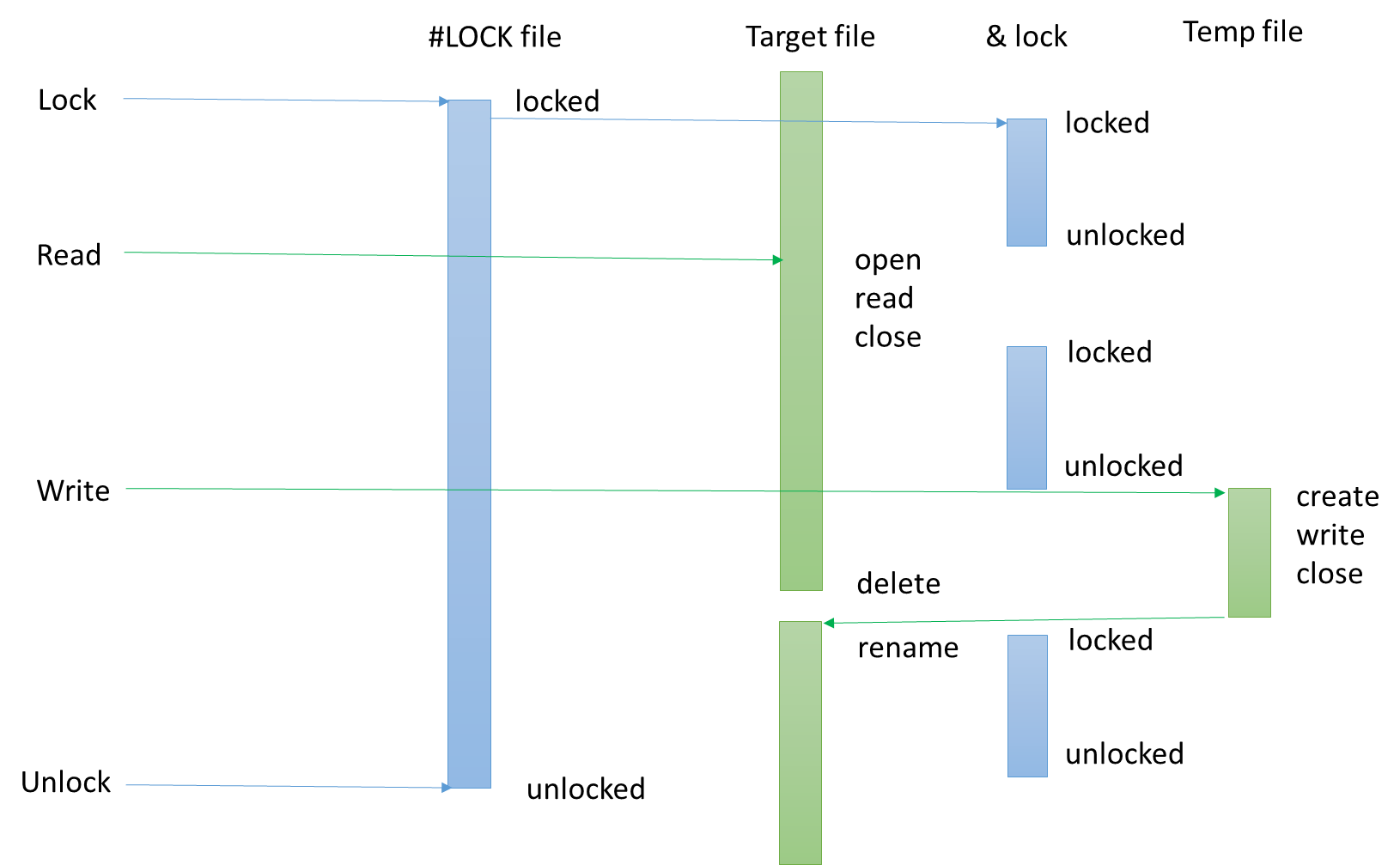
This is the standard pattern.

* First the #LOCK file is locked before the target file is touched.
* After the lock have been gotten, the target file will be opened and read and closed.
* You might have some small amount of processing.
* Create and write to a temp file
* Delete the old target file
* Rename the temp file to be the name of the target file.
* Finally, unlock the #LOCK file

This assured that no process touches the target file unless it has the lock on the #LOCK file. All the processing on the target file is completed before the lock file is released.

# Double Locking

Double licking locks both the #LOCK file, as well as the target file. This is done to attempt to force the file system to have the updated file available in the cache.



* The call to lock the file will first lock the #LOCK file, and then it will lock the target file.
* When reading the file, it will unlock the target file, read it, and then lock the target file again.
* you can do processing here
* When writing, it unlocks the target file, writes out the file using the normal temp file, delete, and rename.
* When renaming is done, it lock the target file again.
* Finally, when everything is unlocked, the target file is unlocked first, and then followed by the #lock file.