OCFS2 filesystem

OCFS2 is a general purpose extent based shared disk cluster file system with many similarities to ext3. It supports 64 bit inode numbers, and has automatically extending metadata groups which may also make it attractive for non-clustered use.

You'll want to install the ocfs2-tools package in order to at least get "mount.ocfs2" and "ocfs2_hb_ctl".

Project web page: http://ocfs2.wiki.kernel.org Tools git tree: https://github.com/markfasheh/ocfs2-tools OCFS2 mailing lists: https://oss.oracle.com/projects/ocfs2/mailman/

Credits

Lots of code taken from ext3 and other projects.

Authors in alphabetical order:

- Joel Becker <joel.becker@oracle.com>
- Zach Brown <zach.brown@oracle.com>
- Mark Fasheh <mfasheh@suse.com>
- Kurt Hackel <kurt.hackel@oracle.com>
- Tao Ma <tao.ma@oracle.com>
- Sunil Mushran <sunil.mushran@oracle.com>
- Manish Singh <manish.singh@oracle.com>
- Tiger Yang <tiger.yang@oracle.com>

Caveats

Features which OCFS2 does not support yet:

- Directory change notification (F_NOTIFY)
- Distributed Caching (F_SETLEASE/F_GETLEASE/break_lease)

Mount options

OCFS2 supports the following mount options:

(*) == default

barrier=1	This enables/disables barriers. barrier=0 disables it, barrier=1 enables it.
errors=remount-ro(*)	Remount the filesystem read-only on an error.
errors=panic	Panic and halt the machine if an error occurs.
intr (*)	Allow signals to interrupt cluster operations.
nointr	Do not allow signals to interrupt cluster operations.
noatime	Do not update access time.
relatime(*)	Update atime if the previous atime is older than mtime or ctime
strictatime	Always update atime, but the minimum update interval is specified by atime_quantum.
atime_quantum=60(*)	OCFS2 will not update atime unless this number of seconds has passed since the last update. Set to zero to always update atime. This option need work with strictatime.
data=ordered (*)	All data are forced directly out to the main file system prior to its metadata being committed to the journal.
data=writeback	Data ordering is not preserved, data may be written into the main file system after its metadata has been committed to the journal.
preferred_slot=0(*)	During mount, try to use this filesystem slot first. If it is in use by another node, the first empty one found will be chosen. Invalid values will be ignored.
commit=nrsec (*)	Ocfs2 can be told to sync all its data and metadata every 'nrsec' seconds. The default value is 5 seconds. This means that if you lose your power, you will lose as much as the latest 5 seconds of work (your filesystem will not be damaged though, thanks to the journaling). This default value (or any low value) will hurt performance, but it's good for data-safety. Setting it to 0 will have the same effect as leaving it at the default (5 seconds). Setting it to very large values will improve performance.
localalloc=8(*)	Allows custom localalloc size in MB. If the value is too large, the fs will silently revert it to the default.
localflocks	This disables cluster aware flock.

inode64	Indicates that Ocfs2 is allowed to create inodes at any location in the filesystem, including those which will result in inode numbers occupying more than 32 bits of significance.
user_xattr (*)	Enables Extended User Attributes.
nouser_xattr	Disables Extended User Attributes.
acl	Enables POSIX Access Control Lists support.
noacl (*)	Disables POSIX Access Control Lists support.
resv_level=2 (*)	Set how aggressive allocation reservations will be. Valid values are between 0 (reservations off) to 8 (maximum space for reservations).
dir_resv_level= (*)	By default, directory reservations will scale with file reservations - users should rarely need to change this value. If allocation reservations are turned off, this option will have no effect.
coherency=full (*)	Disallow concurrent O_DIRECT writes, cluster inode lock will be taken to force other nodes drop cache, therefore full cluster coherency is guaranteed even for O_DIRECT writes.
coherency=buffered	Allow concurrent O_DIRECT writes without EX lock among nodes, which gains high performance at risk of getting stale data on other nodes.
journal_async_commit	Commit block can be written to disk without waiting for descriptor blocks. If enabled older kernels cannot mount the device. This will enable 'journal_checksum' internally.