Journalling file systems

Devesh Sukhwal(00005002)
Sudeep Goyal(00d05013)

Computer Science & Engineering Department,
Indian Institute of Technology,
Bombay.
www.cse.iitb.ac.in/
November 13, 2003

Dirty Buffers

1. To improve speed, data temporarily allocated in RAM by means of page cache / buffer cache subsystem.
2. Dirty buffers :- buffers whose data has not been transferred to the disk yet.
3. If the system crashes they can cause inconsistency on disk
4. e.g. file removed, but inode remains.

Solution : Journalling File Systems

1. fsck can recover, but has to scan the whole partition and can be very time consuming for servers(terabytes of storage).
2. journal :- File metadata written to a serial log before updating the original disk blocks.
3. Recovery after crash involves only analyzing this metadata, and trying to clean up only those files.
4. Earliest journalling filesystems were Veritas(VxFS), Tolerant, IBM(JFS). Now Namesys(ReiserFS), ext3 currently popular.

EXT3

1. ext3 standard in most linux distributions
2. Ext3 is compatible with Ext2, actually it is an ext2fs with a journal file
3. Maintains order consistency in both data and metadata including content of file.
4. The level of journaling can be controlled with mount options.
5. Ext3 partitions do not have a file structure different from ext2, so porting or backing out to the old system, by choice or in the event the journal file were to become corrupted, is straightforward.
6. Any i-node can be used for journalling

ReiserFS

1. only provides metadata journalling(file contents not ensured).

References:

1. http://bulmalug.net/body.phtml?nIdNoticia=1154nIdPage=7
2. http://redwood.snu.ac.kr/ sshong/courses/bib-linux/jfs.pdf