

# Raw devices versus file systems in an Oracle Environment

RDB Consulting  
11/12/2003

## Raw device pro's:

- Improves I/O performance **if** system is I/O bound. When a reasonably well-configured db is I/O bound, it is normally the result of a poorly optimized application query. This kind of problem can sometimes be improved marginally by moving to raw devices, but correct solution is fixing the bad query.
- Provides superior write performance compared to file systems and/or RAID 3/4/5 etc.
- Guaranteed write to disk, no potential for block 'corruption' due to buffering.
- Does not require memory for file system buffering. This memory can be used more effectively elsewhere.
- Is hidden from view. It is not a file system so you can't see it. Will prevent clumsy administrators from removing files accidentally.
- No file system overhead is incurred in terms of inode allocation and maintenance or free block allocation and maintenance.

## Raw device con's

- Additional administrative overhead incurred, as a raw device needs to be configured and pre-allocated for each and every database data file and each and every potential or future data file.
- Cannot auto extend database data files on raw devices. Auto extension is an important Oracle feature for easier and automated database space management.
- Cannot make use of Oracle managed data files with raw devices.
- Cannot shrink database data files on raw devices to reclaim disk space. The raw device has to be recreated in order to reclaim disk

space.

- Easily resizing redo logs and or other database data files is not possible with raw devices without recreating the raw devices to the required size i.e. every change to a databases storage requirement needs a change to be made to the raw devices.
- Need to plan database space much more carefully and have preallocated and unused raw devices at all times in case space needs to be added to the database.
- Both DBA's and SE's need to understand the database and raw device requirements and both require being involved with database space management activities.
- Raw devices require a higher technical skill to manage, so new employees and junior personnel will have to learn and become familiar with the set-up pertinent to the databases requirements. Additional complexity and skill levels introduce additional risk.
- Transportable tablespaces are a lot harder to use with raw devices.
- You can't see disk space usage and availability without being a SE and without access to a volume manager.
- Harder Configuration Tuning. Upon finding that a particular disk drive is "hot" and that performance would benefit from movement of an ORACLE data file from that drive to some other; it is likely that no correctly sized section exists on the "cool" drive. Moving data files around, a simple and attractive option in a UNIX file system environment, is potentially impossible with raw devices.

### **File system pro's**

- You only need to create a file system once, that's it! No SE involvement from there onwards with respect to managing disk space and database data files.
- No need to carefully plan database data file sizing and preallocation, all space management issues with Oracle can be done, when needed, by the DBA only.
- Can use all of Oracles automated space management features such as auto extension and Oracle managed data files.
- Can shrink data files and reclaim disk space without having any file

system management considerations.

- Can add, change or resize database redo logs and data files on the fly without having any file system management considerations.
- File system space allocation and usage can be easily viewed.
- Journalled or Veritas VxFS file systems have a direct I/O option, which delivers file system write speeds equivalent to raw devices but without the management overhead.
- Transportable tablespaces are much easier to use with file systems.
- Junior personnel with less experience and technical skills can work in a simpler environment without the risk associated with their skills development and learning process.
- No block corruption or data loss with file systems. Oracle opens database data files with a data sync flag which means the any writes to disk by Oracle are guaranteed to go to disk and not be left in the file system cache. In other words, even when using file systems there will not be a chance of a corrupt block or lost data in the event of a server or database crash. If this was the case then I think Larry Ellison would have been sued into bankruptcy by now.
- UNIX systems have a UNIX kernel capability for direct reading and writing of UNIX mounted filesystems. This kernel feature allows application software to bypass the UNIX I/O buffering mechanism for disk performance that essentially matches that of using raw devices, without incurring the administrative costs of using unmounted disk sections. If this is available no reasons to use raw devices

File system con's

- Slow write speeds. This should not be taken into consideration as a disadvantage if there is not an actual I/O bottleneck. However, file system I/O performance can be brought on a par with raw devices if using VxFS with direct I/O. Raw devices only increase write performance and not read performance.
- Less secure than raw devices in the sense that a user with privileged permissions can easily remove the database data files, as they are easily visible. But this applies to any file on any file system whether it is a database file, an Oracle binary, an application server file or a UNIX O/S file. Using raw devices won't

necessarily be a foolproof way to protect against clumsy administrators, but a daily database backup will.

- File system overhead is incurred in terms of inode allocation and maintenance or free block allocation and maintenance. However, Oracle database data files are static in size throughout general database activity so this should not be a very big overhead.