iSCSI: Backing blocks with files

Paul Forgey
EMC / Isilon Storage Division
Levels of indirection

- Databases perform best against raw block storage devices
- A filesystem is a type of database
- Filesystems on filesystems are another layer of indirection
Responsibilities of the cluster

- Contents must be accessible in absence of any one node (or more)
- Update mirrors and/or FEC data
- Cost of synchronous writes are much higher
Single block writes are not simple. Commits are read-modify-write

Write-cache-enable or not? WCE can coalesce RMW at the expense of data guarantees.
Filesystem writes

Filesystems not demanding synchronous writes but using SYNCHRONIZE-CACHE instead could see better performance through reduced latency. The filesystem is given application hints (such as O_SYNC, fsync(), etc) about expected data guarantees. The storage layer otherwise has to assume every write needs a data guarantee upon acknowledgement.
OneFS specific

While these are OneFS specific issues, some aspects of these limitations may apply to other filesystems on other platforms

- OneFS is file oriented. Lockers, journaling, caches are per file.
- Write latency may be reduced by spreading data among multiple files.
- Is there an optimal way to tune a multiple file backed LUN to be workflow agnostic?