Critique Week 8

Andres Calderon - SID:861243796

February 22, 2016

File System Design for an NFS File Server Appliance (Hitz et al, 1994)

The document presents a technical report by Network Appliance describing WALF (Write Anywhere File Layout) as a file system which implements Snapshots to work specifically with an NFS appliance. WALF uses a copy-on-write technique to minimize disk consumption by Snapshots and favor quick restart even after unclean shutdowns. WALF satisfies two main goals: usability and optimization for network file access.

The document stresses the importance that an NFS file server Appliance needs a file system with special requirements. To cover those requirements, WALF focuses on (1) provide a fast *NFS* service, (2) support large and scalable file systems, (3) support *RAID* with high performance and security, and (4) provide quick restarts. However, *NFS* and *RAID* have a high impact on write performance so WALF enables Snapshots to use non-volatile RAM (NVRAM) and write-anywhere design to deal with them.

Snapshots are read only copies of the entire file system. WALF creates and deletes Snapshots periodically and keeps a number of them available. Snapshots use a copy-on-write technique so it only records the changes on the system, if there is not modifications the blocks remain intact in the snapshot. As a result, Snapshots make easy to make backups, recovery files and restart quickly.

In the implementation section, the document defines WALF as a block-based, unix compatible file system which optimizes NFS. Although similar to FFS, it stores meta-data in files (inode, block maps and inode maps). It allows a write anywhere strategy which enables the copy-on-write technique used by Snapshot and allows efficient operation in RAIDs. The key of the WALF implementation is to see the system as a *tree of blocks* where a root inode refer to the other files in the system. This structure allows easy creation of Snapshots by duplicating just the root inode (figure 3 is self-explanatory). It provides a very quick implementation with little disk I/O use.

WALF also creates Consistency Points at regular basis. They are similar to Snapshots but with no name and no NFS access. They provide fast system consistency even at unclean shutdowns thanks to the quick Snapshot creation mechanism and use of NVRAM to keep a log for NFS requests. WALF implementation also pay special attention to write allocation. WALF can write any block to any location in any order and allocate space for many NFS operations at once. It gives great flexibility and provide more opportunities to implement policies and new strategies, for example to take advantage of locality and scheduling.

The document provide a brief section discussion about performance. Although the authors clarify that compare WALF's performance with other systems is unfair due to the specificity of the appliance development, it should be appropriate to provide some comparison to illustrate the advantages of the implementation. For example, in section 3.4 says that WALF should be inefficient if it does not apply write episodes. It should be interesting to compare the performance with and without this capability. Similarly, the impact and use of the NVRAM could be better understood by the use of graphs and comparisons.