Lab 7 - Understanding Filesystems

-- 140050002, 140050007

File Systems for Comparison :

1. ZFS with data deduplication
2. ZFS without data deduplication

How Data Deduplication is implemented in ZFS :

When storing similar files on disk it is likely that many blocks have same data. We can exploit this fact using data deduplication to avoid creating new blocks having same data. ZFS implements Block-level data deduplication.

Each block is hashed with a cryptographic hashing algorithm, such as SHA-256. If the hash matches for multiple blocks, rather than storing the new block on disk, ZFS references new block to the original block. This can have significant savings of disk space.

Workload : Writing similar files to disk

Experiment :

We created two ZFS file systems (*testPoolWith* and *testPoolWithout*) in which one will have data deduplication on and another will be without it. File “foo.txt” containing random data is generated using a python code in the current directory. We copy this same file twice in each of the systems and observe the disk space used and deduplication ratio.

Metric : Disk space occupied, Deduplication Ratio

Result :

The total available space on disk to both the filesystems is initially is 464 MB. This is observed when we “zfs list” the two file systems. After adding 2 files (both same and each ~118MB) to these file systems we observe that the available space on disk to the pool with data deduplication on is 350MB but without it is 246MB.

We also observe that the data deduplication ratio is 1.94 when we use “zpool get dedupratio” on the filesystem with deduplication ON. Other file system has deduplication ratio of 1.

For no deduplication file system; Used Space is 218MB and Available disk space is 246MB , which implies that their sum is 464MB which is the initial allocated space on disk. Similarly for using deduplication we get Virtual Used disk space of 221MB and Available space 350MB. Considering deduplication ratio of 1.94 we get the actual value of (221 / 1.94) ~= 114 MB of disk space used. Which adds up to 464MB which is also the initial allocated space on disk.

File System (ZFS)	Real Disk Space Available	Real Disk Space Used	Deduplication Ratio	Disk Space Saved
Without Deduplication	246 MB	218 MB	1.00	--
With Deduplication	350 MB	114 MB	1.94	47%

Disadvantage :

The deduplication table used to maintain deduplication records inside RAM is an extra heavy burden. In our experiment in case of ZFS with data deduplication, we observe a bp count of 1999 when we use “zdb -b”.

Since each deduplication table entry is 320 B in size we observe that these deduplication tables occupy (1999 * 320) B. The percentage occupancy is ((1999*320 B) / 221 MB)*100 ~= 0.3%
Hence when we want to store huge data say of size 1 TB, it will occupy 3 GB of RAM to just store the deduplication table. Also the deduplication table is capped at 25% size of ARC.
Thus 3 GB *4 = 12 GB RAM is required. This is a huge disadvantage.

How to run code :

1. Install ZFS. (link : http://serverascode.com/2014/07/01/zfs-ubuntu-trusty.html)
2. Open directory containing Makefile and createFile.py in terminal.
3. Run command "make test"
4. Run command "make clean" after observing results.