CS344 Operating Systems Lab
Assignment 4

REPORT

GROUP 20

Anjali Godara -180101008
Niharika Bhamer 180101048
Varhade Amey Anant 180101087
Tanmay Jain 180123050

1.File System and Implementation details

We have considered the following two file systems :

A. **ZFS**
B. **ext4**

Brief description of file systems:

ZFS

ZFS is a local file system and logical volume manager created by Sun Microsystems Inc. to direct and control the placement, storage, and retrieval of data in enterprise-class computing systems. The ZFS file system and volume manager is characterized by data integrity, high scalability, and built-in storage features. ZFS is designed to run on a single server, potentially with hundreds if not thousands of attached storage drives. ZFS pools the available storage and manages all disks as a single entity. A user can add more storage drives to the pool if the file system needs additional capacity. ZFS is highly scalable and supports a large maximum file size.

ext4

The EXT4 filesystem primarily improves performance, reliability, and capacity. To improve reliability, metadata and journal checksums were added. To meet various mission-critical requirements, the filesystem timestamps were improved with the addition of intervals down to nanoseconds. In EXT4, data allocation was changed from fixed

blocks to extents. This makes it possible to describe very long, physically contiguous files in a single inode pointer entry, which can significantly reduce the number of pointers required to describe the location of all the data in larger files. EXT4 reduces fragmentation by scattering newly created files across the disk so that they are not bunched up in one location at the beginning of the disk, as many early PC filesystems did.

Features of interest

Features and Experiments performed :

1. **Data Deduplication:** It is a specialized data compression technique for eliminating duplicate copies of repeating data. Related and somewhat synonymous terms are intelligent (data) compression and single-instance (data) storage. This technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. In the deduplication process, unique chunks of data, or byte patterns, are identified and stored during a process of analysis. As the analysis continues, other chunks are compared to the stored copy and whenever a match occurs, the redundant chunk is replaced with a small reference that points to the stored chunk. Given that the same byte pattern may occur dozens, hundreds, or even thousands of times (the match frequency is dependent on the chunk size), the amount of data that must be stored or transferred can be greatly reduced.

2. **Large file Handling:** Ext4 uses 48-bit internal addressing, making it theoretically possible to allocate files up to 16 TiB on filesystems up to 1,000,000 TiB (1 EiB). Early implementations of ext4 were still limited to 16 TiB filesystems by some userland utilities, but as of 2011, e2fsprogs has directly supported the creation of >16TiB ext4 filesystems. As one example, Red Hat Enterprise Linux contractually supports ext4 filesystems only up to 50 TiB and recommends ext4 volumes no larger than 100 TiB.

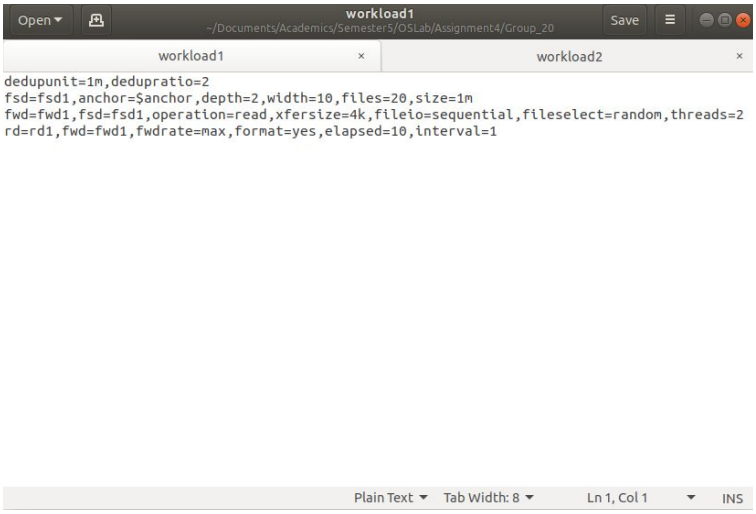
2.Observations and Comparisons

The experiments were done on an Ubuntu 18.04 system. ZFS was installed via the Ubuntu Software Center that gives access to zfs and zpool commands. We use an external USB disk (4 GB Pendrive) to provide the unallocated space for the ZFS filesystem.

Experiment 1

In this part, we want to experiment with the data deduplication feature of ZFS. For this we plan on recording the test results in two different cases, one with data deduplication off and the other with data deduplication on.

The workload used for this experiment is as shown below, we use a dedupratio of 2 and create 2000 files of 1MB each



ZFS with deduplication on

To get started, we need to create a pool.

```
$sudo zpool create pool_name /dev/sdb
```

To switch on data deduplication, we use the following command:

```
$sudo zfs set dedup=on [POOL NAME]
```

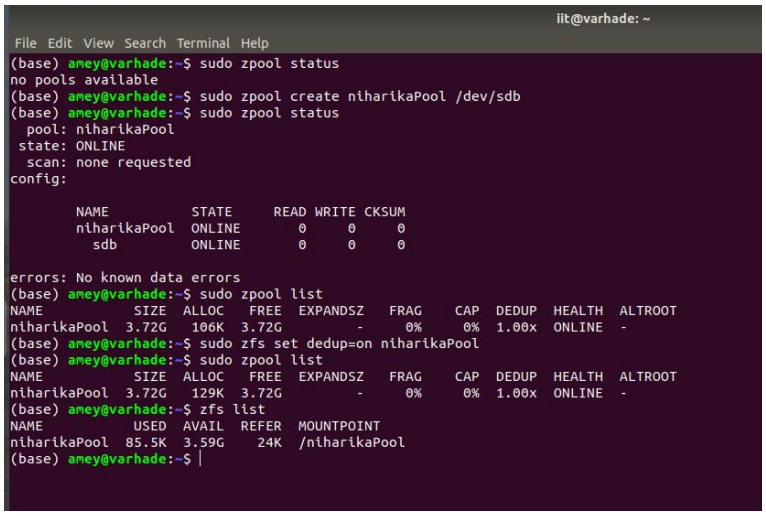
After switching it on, we now test it with a workload created in vdbench, where we read 2K files each of 1MB each and record the before and after of the memory assigned to our pool. For this we use the command:

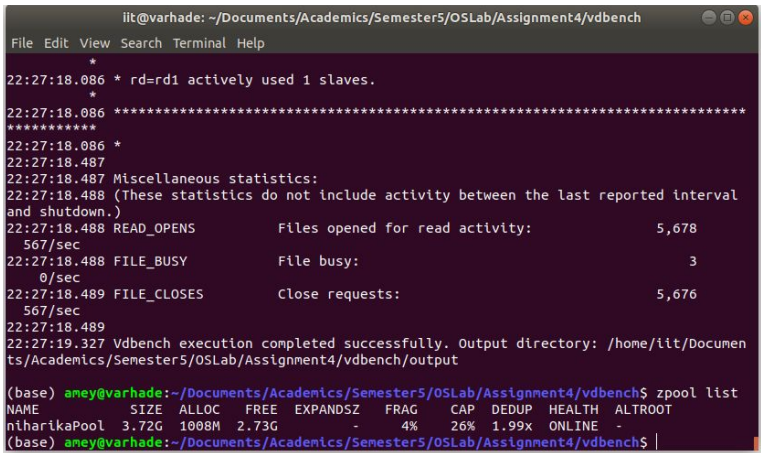
```
$sudo zpool list
```

We now repeat the same steps, but with data, deduplication switched off.
To switch off data deduplication, use the below command:

```
$sudo zfs set dedup=off [POOL NAME]
```

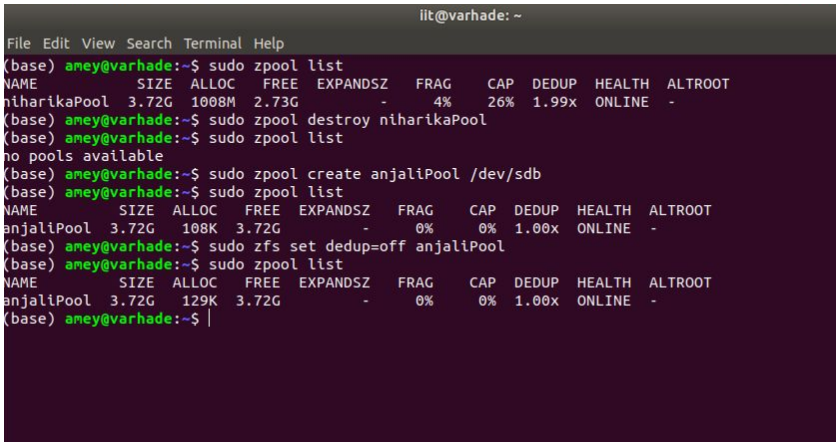
Before and after of the memory assigned to the pool are:

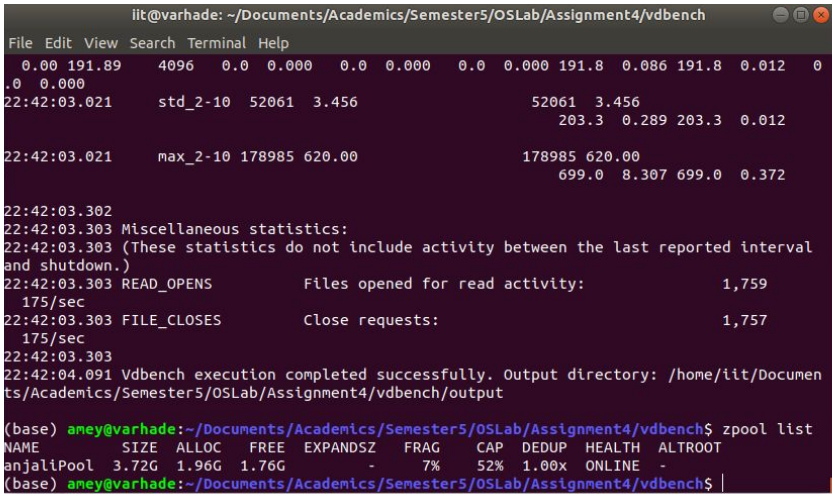




ZFS with deduplication off

To keep data deduplication off, we use the following command:
`$sudo zfs set dedup=off [POOL NAME]`





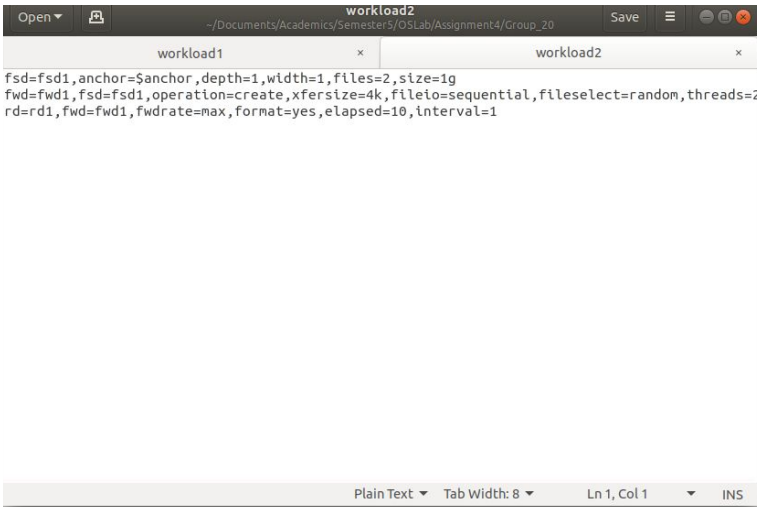
In the above experiment we can observe that when we perform a read operation of 2000 files on both the pools there is a difference in the memory usage of the disk. Along with taking less space than the actual space needed, even the time for copying is less while dedup is on. In the case when dedup is off, since all the duplicated data is copied too, space used on the disk is comparable to the actual size of the files and hence the time consumed is more.

Another interesting observation is the usage of CPU when dedup is turned on and off. It can be observed and argued that dedup on is more CPU intensive than dedup off.

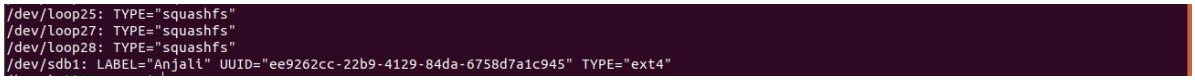
Experiment 2

In the second part of this experiment we want to observe the large file creation functionality of the ext4 filesystem.

The workload used for this experiment is as shown below, we create two large files of 1GB each

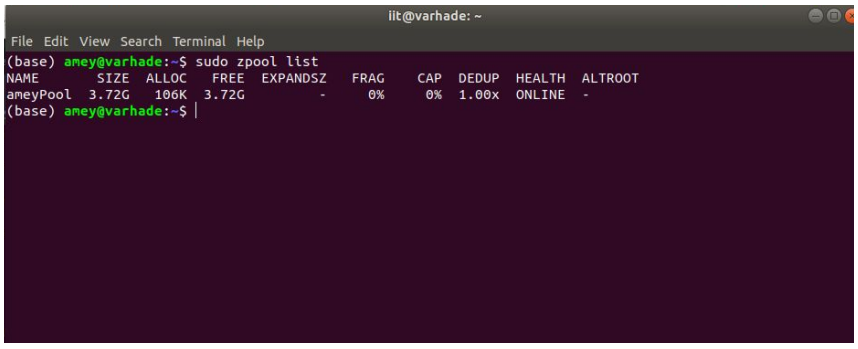


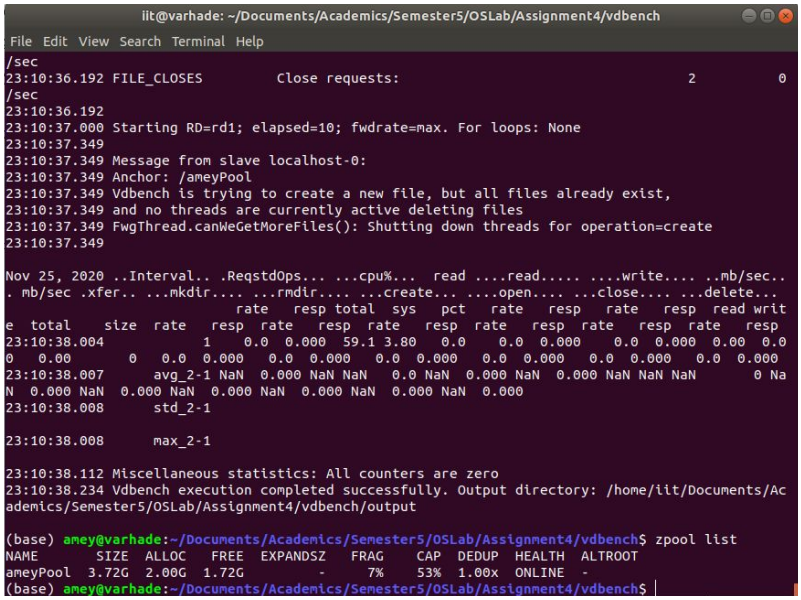
The file type in the external device is set to be ext4 as shown below

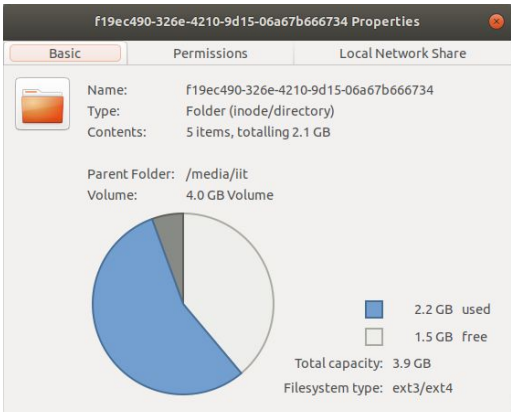


For this, we begin by installing the file system in an external USB drive of 4GB. We start off by creating a workload that creates 2 files each of 1GB. It is this workload whose results we would be observing.

Testing it on ext4, we observe that the files were created pretty fast and this we conclude by observing the write-rate in the summary.html file. Whereas, in ZFS creating such large files takes up much longer than ext4.







3.Conclusions

Experiment 1

One of the primary improvements in performance seen with deduplication is that it leads to a lower disk space utilization as compared to when there is no deduplication. There is a tradeoff between disk space and CPU utilization rate. The second observation of experiment 1 leads us to believe that when our data contains little to no duplicates, there won't be much difference in dedup on and dedup off disk space usage, and then to have better performance, we should go with dedup off and save yourself that extra overhead.

Disadvantages of deduplication :

a) If your data doesn't contain any duplicates, enabling dedup will add overhead without providing any benefit. The ZFS system takes more time than the ext4 to set up the filesystem. In the large file creation also ext4 was giving better **performance** in terms of speed. The average write speed is faster in case of ext4 filesystems.

b) **CPU utilization** is more when the deduplication is on due to deduplication overhead. In the large file creation also ZFS has more CPU utilization than the ext4 filesystem.

Experiment 2

Disadvantages of large file creations:

a)Ext4 is a refurbishment of technology developed in the early 1990s, it has limited capacity to manage modern loads of data. Its once-helpful journaling system now slows down its processes as it stores more data. Plus, ext4 can support a file size no larger than 18 terabytes, making it a modest storage space for a contemporary data-driven, digital company.

b) There is no possible recovery from corruption. Large file creation is optimized by Ext4 via delayed and contiguous allocation and extents.