

CS344 Assignment 4

Group M20

Aditya Pandey: 200123004

Aman Bucha: 200123006

Sarthak Diwan: 200123076

The file system we have considered in order to give analysis on different features is the zfs file system.

ZFS

ZFS (Zettabyte file system) is a file system with volume management capabilities. The file system was first introduced as part of the Sun Microsystems Solaris operating system in 2001. During 2005 to 2010 the open source version of ZFS was ported to Linux, Mac OSX(continuing as the MacZFS) and FreeBSD. In 2013, OpenZFS was founded to coordinate the development open source ZFS. OpenZFS maintains and manages the core ZFS code, while organizations using ZFS maintain specific code and validation processes required for ZFS to integrate within their systems. OpenZFS is widely used in Unix-like systems.

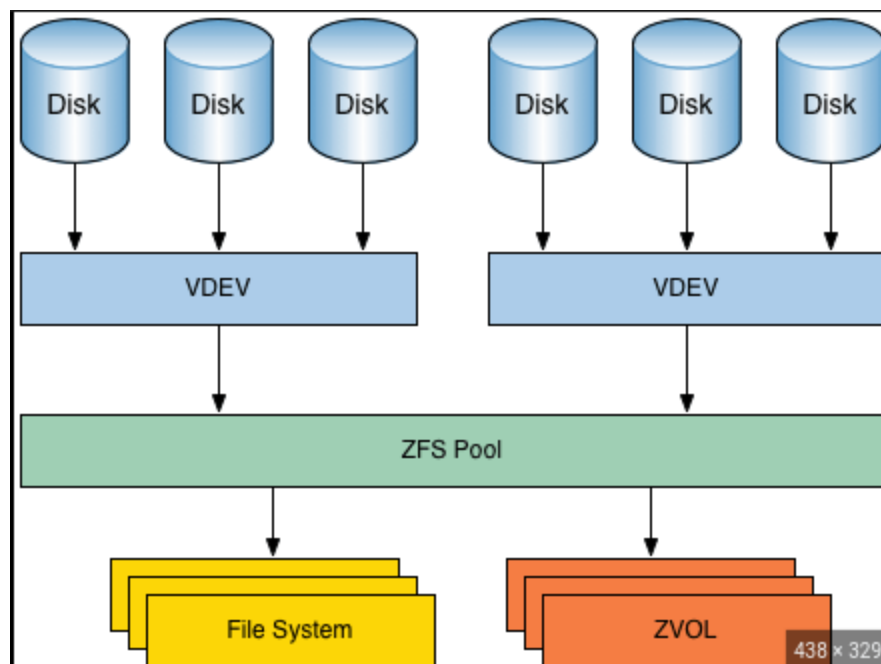
ZFS is different from other file systems as unlike most storage systems it unifies the role of both the volume manager and the file system. Thus, it has complete knowledge of both physical disks and volumes and also of the file stored on them. ZFS is designed to ensure that the data stored on disks cannot be lost due to physical errors or misprocessing by the hardware or operating system.

ZFS also includes the mechanism for dataset and pool-level snapshots and replication, including cloning which maybe described as one of its most powerful features, implementing features that other file systems even with the snapshot feature lack.

Some of the features of the ZFS file system can be enumerated in the following manner:

1. Data Integrity

2. RAID-z
3. Capacity
4. Encryption
5. Read/Write Efficiency
6. Caching mechanism
7. Copy on write transactional model
8. Snapshots and clones
9. Dynamic striping
10. Variable block sizes
and so on.....



The first feature of ZFS we wanted analyse was the deduplication feature:

The researched details of performance for the deduplication feature have been referenced from the following paper:

<https://www.usenix.org/system/files/login/articles/1916-galvin.pdf>

EXT4

This file system primarily focuses on performance and capacity. In this system, data allocation is in the form of extents, instead of fixed size blocks. Extents are described starting and ending places on the hard drive. This reduces fragmentation of memory allocated by the EXT4 file system and thus helps in storing location of file with the help of a small number of pointers, instead of using a pointer pointing to all the blocks of memory occupied by the file.

Deduplication

Deduplication works by “thumb printing” in which an entity is checksummed, resulting in a hash value. Hashing is very effective, providing unique values for the data stored in almost all cases. While there are chances for collision, it is fairly reasonable to say that if the hash values for 2 data entities are same then the data stored is same as well. If the hash value for the data is computed and the value already exists in the data deduplication table, a pointer pertaining to the same data is stored rather than a new copy of the data in the same location.

Deduplication is done by file systems in two manners:

1. Performing deduplication on blocks post processing, that is, they store all entities on write request and then later compare the entities and remove duplicates. This implementation is used by the NetApp FAS.
2. ZFS implements deduplication at the time of writing. Although this takes a penalty at the time of writing, it is space efficient in terms of memory usage in comparison to post processing deduplication.

ZFS deduplication, as with other features of ZFS such as compression, only works on data written after the specific feature is enabled. If a lot of data already exists in a ZFS

pool, there is no native way to have that deduplicated. Any new data will be deduplicated rather than written, but for the existing data to be deduplicated, that data would need to be copied to another pool (for example) or replicated to a ZFS file system with enabled deduplication.

In zfs, once deduplication is enabled the variable `dedupratio` shows how much effect deduplication is having on data in a ZFS pool. ZFS has inbuilt file system checksumming. Deduplication enables a stronger checksum for the file system when enabled. By default deduplication uses SHA256. Hashing almost always results in matches only when the hashed entities exactly match. However the almost always and always have a huge gap between them, meaning there can still be collisions leading to corruption of data.

First we want to experimentally show the difference between zfs with its deduplication feature and ext4 without its deduplication feature.

1. First we need to setup the deduplication feature by creating a zfs file system with the parameter `dedup = on`.

Large file creation:

The EXT4 file system allows for a maximum file size of 16 TiB (TebiBytes) - 2^{44} Bytes with the common 4 KiB blocks and 48 bit block addressing, as well as a maximum volume of 1 EiB (ExbiByte) = (2^{60}) Bytes.

EXT3 only permits file sizes of 2 TiB and a file system size of 16 TiB.

ZFS supports file systems with a 16 TiB size.

Very huge files can be created and handled very efficiently with EXT4.

This is because large files take up a lot of space and time to save and access due to extensive mapping of data blocks.

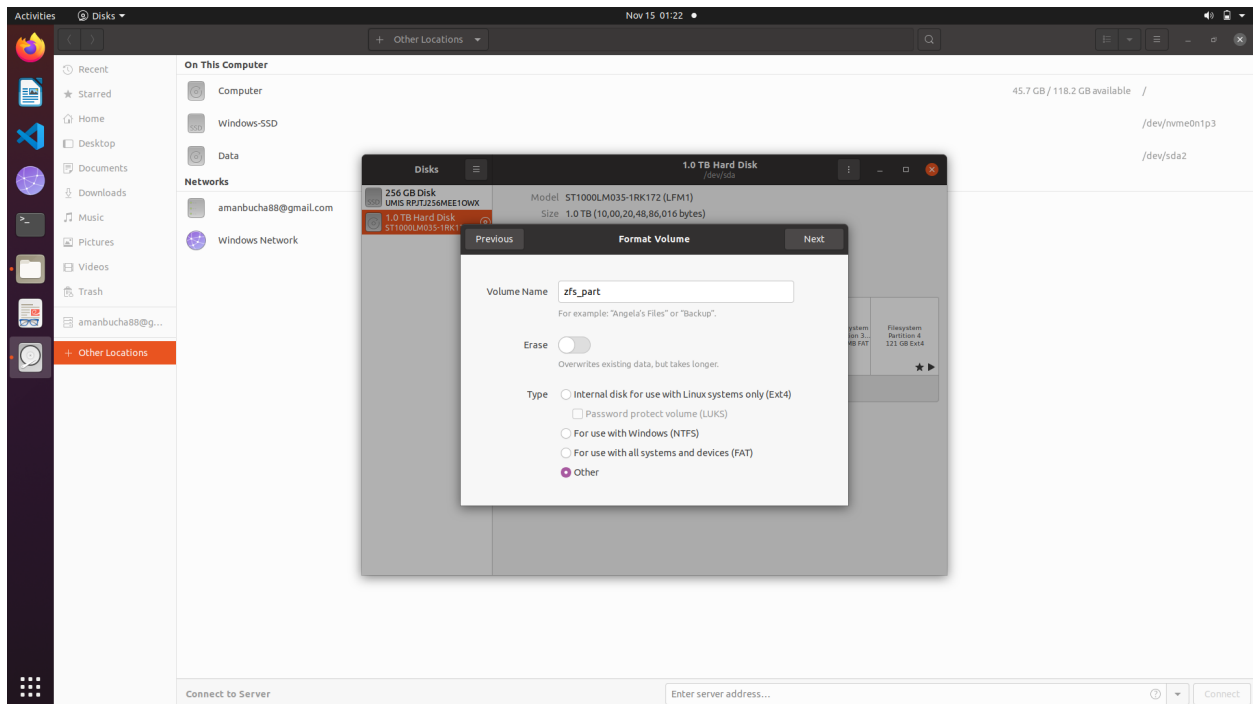
Other features in EXT4 also aid in the proper and effective operation of this new extent-based mapping system.

Multiblock allocation is a feature of EXT4 that allows for easy allocation of contiguous blocks of memory while avoiding a significant amount of cost by allocating many blocks in a single call as opposed to one block each call.

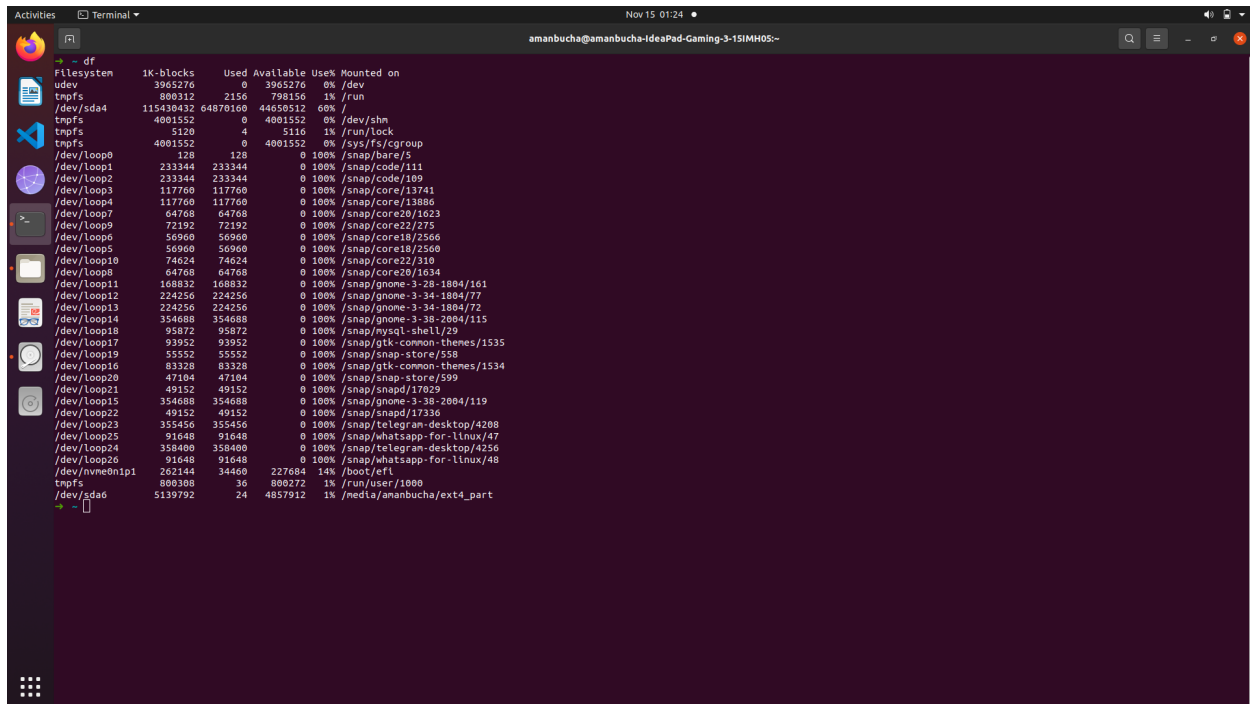
This works in conjunction with delayed allocation, which doesn't write to disc on every write operation instead noting the data to be written before using multiblock allocation to write a sizable amount of data into a contiguous memory segment.

Steps for experiment:

We create the partitions of the disk: `zfs_part` and `ext4_part`

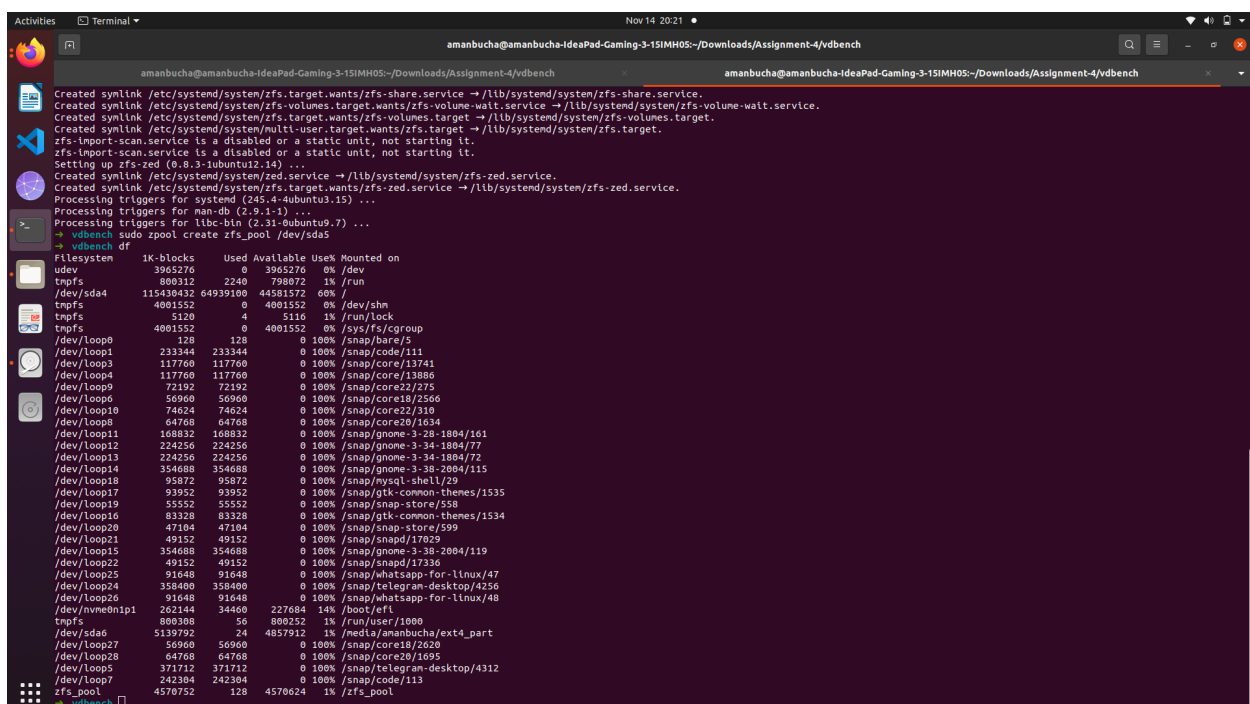


We noticed that zfs file system was not listed when we ran `df` command



```
amanbucha@amanbucha-IdeaPad-Gaming-3-15IMH05:~$ df
Filesystem            1k-blocks      Used Available Use% Mounted on
udev                  3965276        0  3965276   0% /dev
tmpfs                  800312      2156    798156   1% /run
/dev/sda4             115430432 64870160  44650512  60% /
tmpfs                 4001552        0   4001552   0% /dev/shm
tmpfs                  5120         4     5116    1% /run/lock
tmpfs                 4001552        0   4001552   0% /sys/fs/cgroup
/dev/loop0             128         0     100% /snap/bare/5
/dev/loop1             233344      233344    100% /snap/code/111
/dev/loop2             233344      233344    100% /snap/code/109
/dev/loop3             117760      117760    100% /snap/core/13741
/dev/loop4             117760      117760    100% /snap/core/13886
/dev/loop7             64768       64768    100% /snap/core20/1622
/dev/loop9             72192       72192    100% /snap/core22/275
/dev/loop6             56960       56960    100% /snap/core18/2566
/dev/loop5             56960       56960    100% /snap/core18/2560
/dev/loop10            74624       74624    100% /snap/core22/310
/dev/loop8             64768       64768    100% /snap/core20/1634
/dev/loop11            168832      168832    100% /snap/gnome-3-28-1804/161
/dev/loop12            224256      224256    100% /snap/gnome-3-34-1804/77
/dev/loop13            224256      224256    100% /snap/gnome-3-34-1804/72
/dev/loop14            354688      354688    100% /snap/gnome-3-38-2004/115
/dev/loop18            95872       95872    100% /snap/mysql-shell/29
/dev/loop17            93952       93952    100% /snap/gtk-common-themes/1535
/dev/loop19            55552       55552    100% /snap/snap-store/558
/dev/loop16            83328       83328    100% /snap/gtk-common-themes/1534
/dev/loop20            47104       47104    100% /snap/snap-store/599
/dev/loop21            49152       49152    100% /snap/snapd/17029
/dev/loop15            354688      354688    100% /snap/gnome-3-38-2004/119
/dev/loop22            49152       49152    100% /snap/snapd/17336
/dev/loop23            35456       35456    100% /snap/telegram-desktop/4288
/dev/loop25            91648       91648    100% /snap/whatsapp-for-linux/47
/dev/loop24            358400      358400    100% /snap/telegram-desktop/4256
/dev/loop26            91648       91648    100% /snap/whatsapp-for-linux/48
/dev/nvme0n1p1        262144      34460    227684  14% boot/efi
tmpfs                  800308       36    800272   1% /run/user/1000
/dev/sda6             513972      24    485792   1% /media/amanbucha/ext4_part
```

We corrected this by initialising zfs file system on `zfs_part` volume:



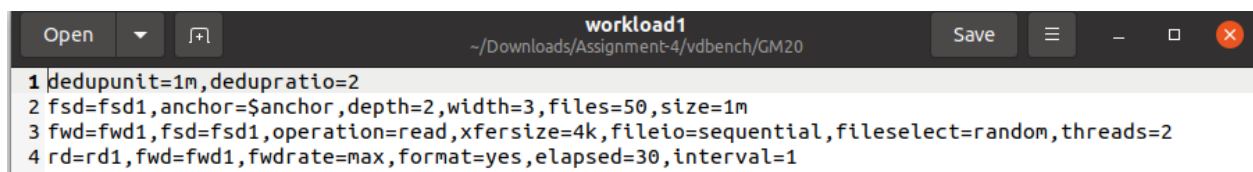
```
amanbucha@amanbucha-IdeaPad-Gaming-3-15IMH05:~/Downloads/Assignment-4/vdbench$ df
Filesystem            1k-blocks      Used Available Use% Mounted on
udev                  3965276        0  3965276   0% /dev
tmpfs                  800312      2240    798072   1% /run
/dev/sda4             115430432 64939100  44581572  60% /
tmpfs                 4001552        0   4001552   0% /dev/shm
tmpfs                  5120         4     5116    1% /run/lock
tmpfs                 4001552        0   4001552   0% /sys/fs/cgroup
/dev/loop0             128         0     100% /snap/bare/5
/dev/loop1             233344      233344    100% /snap/code/111
/dev/loop3             117760      117760    100% /snap/core/13741
/dev/loop4             117760      117760    100% /snap/core/13886
/dev/loop9             72192       72192    100% /snap/core22/275
/dev/loop6             56960       56960    100% /snap/core18/2566
/dev/loop10            74624       74624    100% /snap/core22/310
/dev/loop8             64768       64768    100% /snap/core20/1634
/dev/loop11            168832      168832    100% /snap/gnome-3-28-1804/161
/dev/loop12            224256      224256    100% /snap/gnome-3-34-1804/77
/dev/loop13            224256      224256    100% /snap/gnome-3-34-1804/72
/dev/loop14            354688      354688    100% /snap/gnome-3-38-2004/115
/dev/loop18            95872       95872    100% /snap/mysql-shell/29
/dev/loop17            93952       93952    100% /snap/gtk-common-themes/1535
/dev/loop19            55552       55552    100% /snap/snap-store/558
/dev/loop16            83328       83328    100% /snap/gtk-common-themes/1534
/dev/loop20            47104       47104    100% /snap/snap-store/599
/dev/loop21            49152       49152    100% /snap/snapd/17029
/dev/loop15            354688      354688    100% /snap/gnome-3-38-2004/119
/dev/loop22            49152       49152    100% /snap/snapd/17336
/dev/loop25            91648       91648    100% /snap/whatsapp-for-linux/47
/dev/loop24            358400      358400    100% /snap/telegram-desktop/4256
/dev/loop26            91648       91648    100% /snap/whatsapp-for-linux/48
/dev/nvme0n1p1        262144      34460    227684  14% boot/efi
tmpfs                  800308       36    800272   1% /run/user/1000
/dev/sda6             513972      24    485792   1% /media/amanbucha/ext4_part
/dev/loop27            56960       56960    100% /snap/core18/2620
/dev/loop28            64768       64768    100% /snap/core20/1695
/dev/loop5             371712      171712    100% /snap/telegram-desktop/4312
/dev/loop7            242304      242304    100% /snap/code/113
zfs_pool              4570752      128  4570624   1% /zfs_pool
```

The mount point of zfs is /zfs_pool

The mount point of ext4 is /media/\$USER/ext4_part

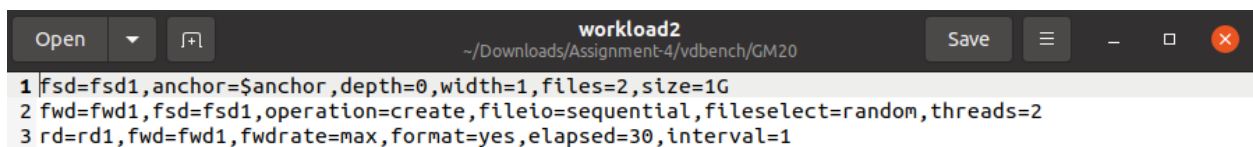
We use the following workloads and accordingly compare the two file systems based on the output:

1) Workload 1



```
1 dedupunit=1m,dedupratio=2
2 fsd=fsd1,anchor=$anchor,depth=2,width=3,files=50,size=1m
3 fwd=fwd1,fsd=fsd1,operation=read,xfersize=4k,fileio=sequential,fileselect=random,threads=2
4 rd=rd1,fwd=fwd1,fwdrate=max,format=yes,elapsed=30,interval=1
```

2) Workload 2



```
1 fsd=fsd1,anchor=$anchor,depth=0,width=1,files=2,size=1G
2 fwd=fwd1,fsd=fsd1,operation=create,fileio=sequential,fileselect=random,threads=2
3 rd=rd1,fwd=fwd1,fwdrate=max,format=yes,elapsed=30,interval=1
```

1. Deduplication

- a. There is a data deduplication feature in `zfs` which is turned on by the command

```
sudo zfs set dedup=on zfs_pool
```

- b. We now compare the space utilization of `zfs` and `ext4` using the following workload
- c. In the workload, we are creating **450 files** ($50 * 3 * 3$) each of size 1MB in a nested folder with depth = 2 and width = 3. We are then reading the files in sequence for 30 seconds.
- d. `dedupunit` is set to 1MB and `dedupratio` is set to 2.
- `dedupratio` is the ratio of the total number of blocks (of size `dedupunit`) with the number of blocks containing unique data.

dedupunit is the size of the block which will be compared with pre-existing blocks to check for duplicates. So in our case, half of the files are duplicates since the file size is 1MB.

e. Setting anchor to the ZFS Pool directory, we execute this workload on the ZFS file system using the following command.

```
sudo ./vdbench -f workload1 anchor=/zfs_pool
```

f. Setting anchor to the ext4 drive, we execute this workload on the ext4 file system using the following command.

```
sudo ./vdbench -f workload1 anchor=/media/amanbucha/ext4_part
```

Results:

1. zfs

- The empty ZFS folder initially contained **396 KB** of data.
- After the workload was executed, the ZFS folder had **229 MB** of data.
- This means files took **228.6 MB**
- We noticed a deduplication ratio of 2.0x, which was the desired value.
- Therefore instead of **450 MB** the new files took **223 MB** of space.

Before Workload

```
→ vdbench zpool list
NAME      SIZE  ALLOC   FREE CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP   HEALTH  ALTROOT
zfs_pool  4.50G   396K   4.50G      -          -         0%    0%   1.00x   ONLINE   -
```

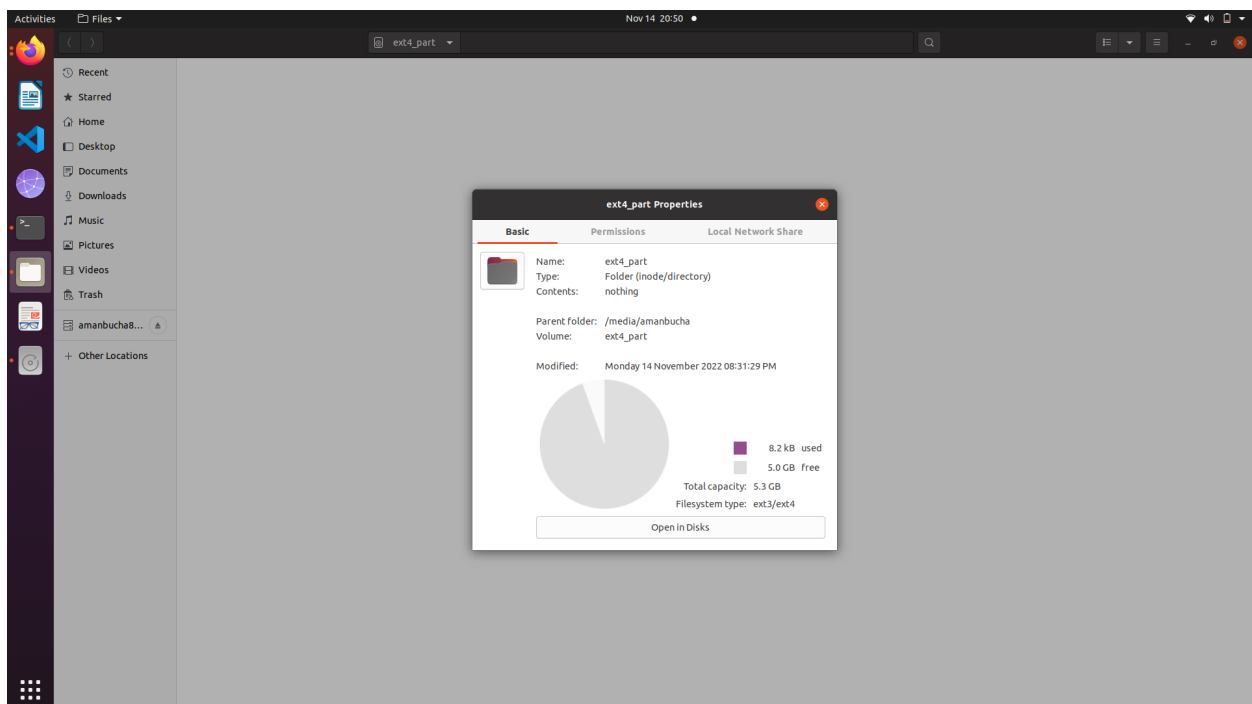
After Workload

```
→ vdbench zpool list
NAME      SIZE  ALLOC   FREE CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP   HEALTH  ALTROOT
zfs_pool  4.50G   229M   4.28G      -          -         0%    4%   2.00x   ONLINE   -
```

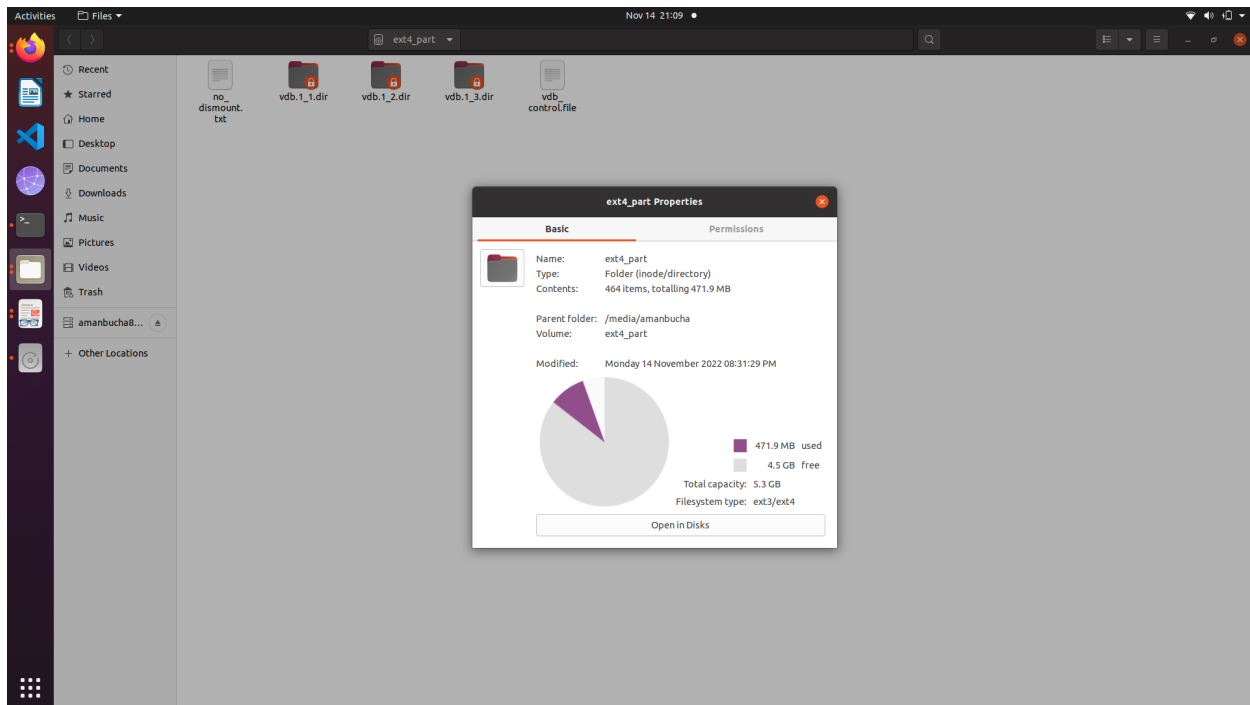
1. ext4

- The empty ext4_part folder initially contained **8.2 KB** of data.
- After the workload was executed, the ext4_part folder had **471.9 MB** of data.
- Therefore, the new files took **471.8 MB** of space. It is a little higher than 450 because of metadata.

Before Workload



After Workload



2. Large File Creation

- We developed the following workload to test the production of huge files:
- In this case, we are producing two 1GB files in a same folder. Since we are testing file creation, the `create` action is utilised.
- Setting anchor to the ZFS Pool directory, we execute this workload on the ZFS file system using the following command.

```
sudo ./vdbench -f workload2 anchor=/zfs_pool
```

- Setting anchor to the ext4 drive, we execute this workload on the ext4 file system using the following command.

```
sudo ./vdbench -f workload2 anchor=/media/amanbucha/ext4_part
```

Results:

1. zfs

- Time taken to create files: **47.85 seconds**
- Average write rate: **42.80 MB/s**

```
Activities | Firefox Web Browser | Nov 14 22:35
Log In | Microsoft | Assignments | CS344_OSLab | CS344 Assignment | Vdbench output/su | Google Docs | Untitled docum... | Google Keep | CS344_OSLab | Vdbench output/su
file:///home/amanbucha/Downloads/Assignment-4/vdbench/zfs work2/summary.html
rd1 For Loops: None
Link to config output: config
22:15:32.003 Starting RD=Format_for_rd1

Nov 14, 2022 ..Interval...Reqstdps...cpu...read...read...write...mb/sec...mb/sec...xfer...mmdir...mmdir...create...open...close...delete...
rate resp total sys pct rate resp rate resp read write total size rate resp rate resp rate resp rate resp rate resp
22:15:33.009 1 1325.0 0.005 13.9 5.74 0.0 0.0 0.000 1315.0 0.005 0.00 164.3 164.38 131072 1.0 0.107 12.0 0.075 0.0 0.000 2.0 3.374 0.0 0.000 450.0 0.103
2 0.0 0.000 10.1 2.78 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:34.010 3 0.0 0.000 6.1 1.25 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:35.014 4 0.0 0.000 4.4 1.01 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:36.014 5 0.0 0.000 2.9 0.50 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:37.016 6 1336.0 7.680 14.6 11.7 0.0 0.0 0.000 1336.0 7.680 0.00 167.0 167.00 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:38.018 7 0.0 0.000 4.4 0.63 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:39.008 8 0.0 0.000 1.8 0.25 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:40.009 9 0.0 0.000 2.6 1.51 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:41.013 10 1242.0 6.407 19.4 11.6 0.0 0.0 0.000 1242.0 6.407 0.00 155.2 155.25 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:42.017 11 0.0 0.000 6.4 2.65 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:43.015 12 1214.0 3.855 14.3 11.2 0.0 0.0 0.000 1214.0 3.855 0.00 151.7 151.75 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:44.014 13 0.0 0.000 7.7 2.00 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:45.020 14 0.0 0.000 2.7 0.20 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:46.010 15 692.0 9.540 3.6 1.51 0.0 0.0 0.000 692.0 9.540 0.00 86.50 86.50 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:47.009 16 497.0 0.059 14.6 11.1 0.0 0.0 0.000 497.0 0.059 0.00 62.00 62.00 130808 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:48.014 17 0.0 0.000 6.2 2.86 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:49.014 18 0.0 0.000 2.9 1.13 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:50.010 19 1201.0 6.230 20.3 12.3 0.0 0.0 0.000 1201.0 6.230 0.00 150.1 150.12 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:51.014 20 0.0 0.000 6.5 2.12 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:52.011 21 0.0 0.000 4.4 2.25 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:53.008 22 1161.0 4.074 13.9 11.9 0.0 0.0 0.000 1161.0 4.074 0.00 145.1 145.12 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:54.010 23 0.0 0.000 3.1 1.51 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:55.008 24 0.0 0.000 3.6 1.63 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:56.005 25 1114.0 5.430 17.0 13.3 0.0 0.0 0.000 1114.0 5.430 0.00 139.2 139.25 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:57.012 26 0.0 0.000 5.4 2.76 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:15:58.000 27 1005.0 4.232 18.3 13.1 0.0 0.0 0.000 1005.0 4.232 0.00 133.1 133.12 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:00.007 28 0.0 0.000 3.9 2.02 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:01.011 29 0.0 0.000 4.5 1.51 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:02.004 30 1022.0 6.648 11.3 10.0 0.0 0.0 0.000 1022.0 6.648 0.00 127.7 127.75 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000

Nov 14, 2022 ..Interval...Reqstdps...cpu...read...read...write...mb/sec...mb/sec...xfer...mmdir...mmdir...create...open...close...delete...
rate resp total sys pct rate resp rate resp read write total size rate resp rate resp rate resp rate resp rate resp
22:16:03.011 31 0.0 0.000 7.3 5.88 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:04.012 32 0.0 0.000 3.5 1.63 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:05.005 33 976.0 5.620 13.6 11.9 0.0 0.0 0.000 976.0 5.620 0.00 122.0 122.00 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:06.009 34 0.0 0.000 5.4 3.51 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:07.007 35 0.0 0.000 2.3 0.75 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:08.000 36 937.0 5.073 14.1 12.8 0.0 0.0 0.000 937.0 5.073 0.00 117.1 117.12 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:09.011 37 0.0 0.000 4.6 3.14 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:10.010 38 896.0 5.370 10.7 9.60 0.0 0.0 0.000 896.0 5.370 0.00 112.0 112.00 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:11.010 39 0.0 0.000 4.4 2.54 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:12.011 40 860.0 4.342 8.9 7.54 0.0 0.0 0.000 860.0 4.342 0.00 107.5 107.50 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:13.010 41 0.0 0.000 4.0 2.62 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:14.010 42 0.0 0.000 3.8 2.14 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:15.007 43 824.0 7.538 10.1 8.17 0.0 0.0 0.000 824.0 7.538 0.00 103.0 103.00 131072 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:16.007 44 0.0 0.000 4.3 2.00 0.0 0.0 0.000 0.0 0.000 0.00 0.00 0.00 0 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:17.007 45 31.0 114.88 3.1 1.88 0.0 0.0 0.000 31.0 114.88 0.00 3.88 3.88 131072 0.0 0.000 0.0 0.000 2.0 44408 0.0 0.000 2.0 124.22 0.0 0.000
22:16:18.001 46 0.0 0.000 7.7 4.95 0.0 0.0 0.000 0.0 0.000 0.00 42.00 42.00 131053 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000 0.0 0.000
22:16:17.010 std 2-45 497.2 135.51 497.2 135.51 0.107 0.148 0.3 0.846 0.3 0.998
22:16:17.010 max 2-45 1336.0 5110.2 1336.0 5110.2 3.375 2.0 124.92 3.418

22:16:18.001 Starting RDrdr1; elapsed=30; fdrdrname: For Loops: None
```

1. ext4

- Time taken to create files: **39.07 seconds**
- Average write rate: **52.41 MB/s**

21:10:29.003 Starting RD=rdformat_for_rdi

Nov 14, 2022	Interval	ReqstDrops	cpu%...	read	write	mb/sec	mb/sec	xfer	mkdir	rmkdir	create	open	close	delete
	rate	resp	total	sys	pct	rate	resp	rate	resp	rate	resp	rate	resp	rate
21:10:30.061	1	2967.0	0.252	13.7	3.13	0.0	0.0	0.000	2967.0	0.252	0.0	0.000	376.88	131072
21:10:31.017	2	2959.0	0.950	14.5	2.24	0.0	0.0	0.000	2959.0	0.950	0.0	0.000	256.25	131072
21:10:32.021	3	334.0	5.999	4.0	0.39	0.0	0.0	0.000	334.0	5.999	0.0	0.000	41.75	131072
21:10:33.017	4	153.0	13.056	4.1	0.64	0.0	0.0	0.000	153.0	13.056	0.0	0.000	19.12	131072
21:10:34.010	5	188.0	9.055	3.3	0.51	0.0	0.0	0.000	188.0	9.055	0.0	0.000	23.50	131072
21:10:35.025	6	249.0	8.733	4.1	1.02	0.0	0.0	0.000	249.0	8.733	0.0	0.000	31.12	131072
21:10:36.018	7	226.0	8.798	2.8	0.77	0.0	0.0	0.000	226.0	8.798	0.0	0.000	28.25	131072
21:10:37.199	8	340.0	7.034	3.1	0.75	0.0	0.0	0.000	340.0	7.034	0.0	0.000	42.50	131072
21:10:38.017	9	274.0	5.845	3.0	1.58	0.0	0.0	0.000	274.0	5.845	0.0	0.000	34.25	131072
21:10:39.148	10	374.0	5.486	3.7	0.77	0.0	0.0	0.000	374.0	5.486	0.0	0.000	46.75	131072
21:10:40.020	11	335.0	5.902	3.9	0.89	0.0	0.0	0.000	335.0	5.902	0.0	0.000	41.88	131072
21:10:41.012	12	426.0	4.091	4.5	1.03	0.0	0.0	0.000	426.0	4.091	0.0	0.000	53.25	131072
21:10:42.017	13	416.0	4.822	3.4	1.03	0.0	0.0	0.000	416.0	4.822	0.0	0.000	52.00	131072
21:10:43.008	14	503.0	3.960	6.4	0.64	0.0	0.0	0.000	503.0	3.960	0.0	0.000	62.88	131072
21:10:44.016	15	514.0	3.914	4.4	1.41	0.0	0.0	0.000	514.0	3.914	0.0	0.000	64.25	131072
21:10:45.000	16	497.0	3.966	4.7	0.91	0.0	0.0	0.000	497.0	3.966	0.0	0.000	62.12	131072
21:10:46.013	17	435.0	4.671	3.9	1.04	0.0	0.0	0.000	435.0	4.671	0.0	0.000	54.12	131072
21:10:47.000	18	342.0	5.067	3.0	1.29	0.0	0.0	0.000	342.0	5.067	0.0	0.000	42.75	131072
21:10:48.016	19	355.0	5.648	3.2	1.03	0.0	0.0	0.000	355.0	5.648	0.0	0.000	44.38	131072
21:10:49.007	20	307.0	5.785	4.5	0.77	0.0	0.0	0.000	307.0	5.785	0.0	0.000	38.38	131072
21:10:50.015	21	378.0	5.805	3.3	1.29	0.0	0.0	0.000	378.0	5.805	0.0	0.000	47.25	131072
21:10:51.000	22	343.0	5.869	3.5	0.77	0.0	0.0	0.000	343.0	5.869	0.0	0.000	42.88	131072
21:10:52.007	23	420.0	4.751	3.8	2.05	0.0	0.0	0.000	420.0	4.751	0.0	0.000	52.50	131072
21:10:53.011	24	517.0	3.862	3.6	1.03	0.0	0.0	0.000	517.0	3.862	0.0	0.000	64.62	131072
21:10:54.000	25	350.0	5.756	4.7	1.78	0.0	0.0	0.000	350.0	5.756	0.0	0.000	43.75	131072
21:10:55.011	26	599.0	3.306	5.3	1.80	0.0	0.0	0.000	599.0	3.306	0.0	0.000	74.88	131072
21:10:56.014	27	382.0	5.220	6.5	1.28	0.0	0.0	0.000	382.0	5.220	0.0	0.000	47.75	131072
21:10:57.010	28	402.0	5.017	6.4	2.05	0.0	0.0	0.000	402.0	5.017	0.0	0.000	50.25	131072
21:10:58.007	29	376.0	5.245	3.6	1.16	0.0	0.0	0.000	376.0	5.245	0.0	0.000	47.00	131072
21:10:59.014	30	443.0	4.531	2.7	1.04	0.0	0.0	0.000	443.0	4.531	0.0	0.000	55.38	131072

21:11:00.000

Nov 14, 2022	Interval	ReqstDrops	cpu%...	read	write	mb/sec	mb/sec	xfer	mkdir	rmkdir	create	open	close	delete
	rate	resp	total	sys	pct	rate	resp	rate	resp	rate	resp	rate	resp	rate
21:11:00.285	31	506.0	3.964	4.6	1.93	0.0	0.0	0.000	506.0	3.964	0.0	0.000	63.25	131072
21:11:01.011	32	364.0	3.594	4.2	1.41	0.0	0.0	0.000	364.0	3.594	0.0	0.000	45.50	131072
21:11:02.009	33	21.0	3.838	3.8	0.75	0.0	0.0	0.000	21.0	3.838	0.0	0.000	2.63	131072

21:11:02.012 std 2-33 319.6 11.969 319.6 11.969 0.051 0.040 1.0 31449 0.2 537.17 0.2 0.061 0.476

21:11:02.012 max 2-33 2050.0 490.44 2050.0 490.44 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

21:11:06.003 Starting RD=rdi; elapsed=30; fd=rate=0; For loops: None

Nov 14, 2022	Interval	ReqstDrops	cpu%...	read	write	mb/sec	mb/sec	xfer	mkdir	rmkdir	create	open	close	delete
	rate	resp	total	sys	pct	rate	resp	rate	resp	rate	resp	rate	resp	rate
21:11:07.011	1	0.0	0.000	3.7	0.88	0.0	0.0	0.000	0.0	0.000	0.0	0.000	0.0	0.000
21:11:07.016	avg 2-1	NaN	0.000	NaN	NaN	0.0	NaN	0.000	NaN	0.000	NaN	0.000	NaN	0.000
21:11:07.016	std 2-1	NaN	0.000	NaN	NaN	0.0	NaN	0.000	NaN	0.000	NaN	0.000	NaN	0.000
21:11:07.016	max 2-1	NaN	0.000	NaN	NaN	0.0	NaN	0.000	NaN	0.000	NaN	0.000	NaN	0.000

21:11:07.487 Vdbench execution completed successfully

Disadvantages of deduplication:

1. Performance:

In the second workload (large file optimization), **ext4** was quicker. This is largely the result of **ext4**'s file optimization and **ZFS**'s deduplication costs.

This demonstrates that deduplication has a negative impact on the performance of a file system owing to its overhead.

Disadvantages of large file creation optimization:

1. Metadata Overhead:

In workload1 (in **ext4**), only **450 MB** was needed for the files, but the metadata overhead was much higher which made the total space utilized to be **471.8 MB**.

However, **ZFS** has a very minimal overhead.

Maintaining the extent trees (in **ext4**) requires a substantial amount of extra storage space compared to the real data (for small files).

2. No data correction mechanisms:

`ext4` optimises the construction of huge files via the use of delayed and contiguous allocation and extents. This makes it difficult for any data correction procedures to exist, since huge files with numerous contiguous blocks retain relatively little information.