CSC494 Report

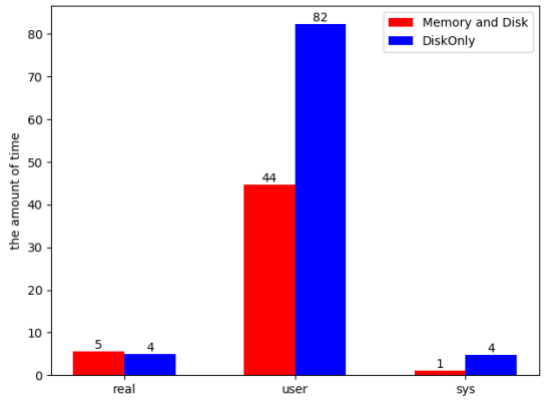
Fengjia Zhang
Yuhan Shao

1. DISK_ONLY and MEMORY_AND_DISK comparison in spark command

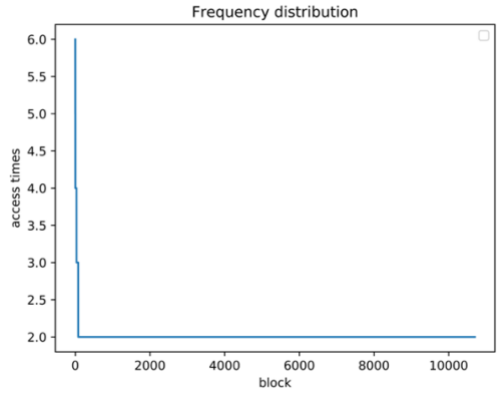
1.1. In ext4

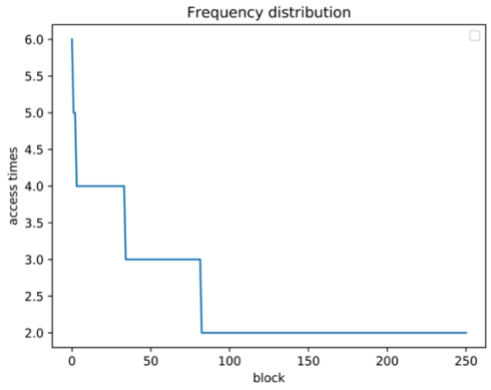
	MEMORY_AND_DISK	DISK_ONLY
Performance drop (the amount of time)	Without blktrace real 2m18.379s user 45m4.699s sys 1m7.734s With blktrace real 2m13.066s user 44m53.089s sys 1m8.120s	Without blktrace real 4m16.086s user 80m29.434s sys 5m0.749s With blktrace real 4m12.011s user 80m57.390s sys 5m1.255s
Blktrace file size	1634736	2811764
blocks written	10,761, 9,112MiB	18,599, 14,663MiB
blocks read	0, 0KiB	1, 4KiB

Amount of time needed for MEMORY_AND_DISK & DISK_ONLY with blktrace:



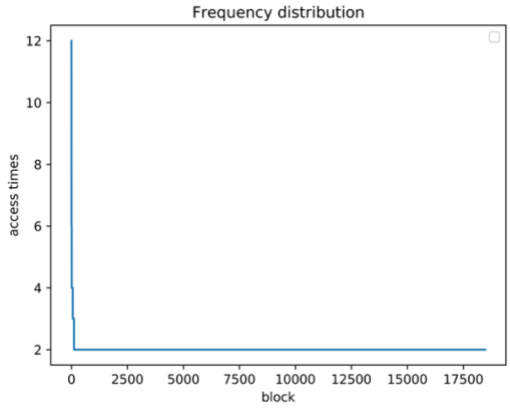
Frequency distribution for MEMORY_AND_DISK: Focusing on the most frequent 250 blocks:

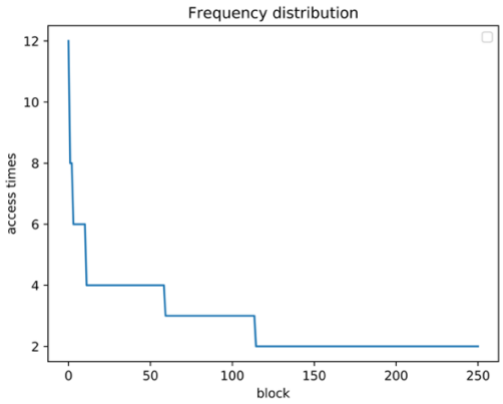




Frequency distribution for DISK_ONLY:

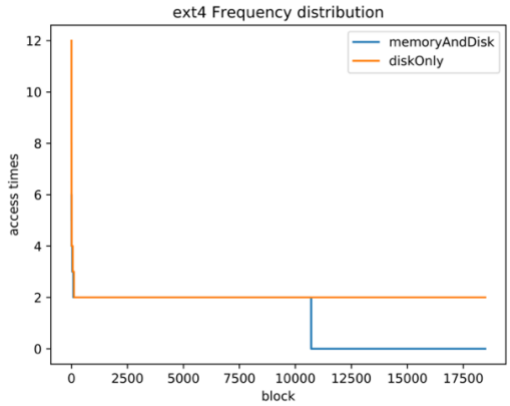
Focusing on the most frequent 250 blocks:

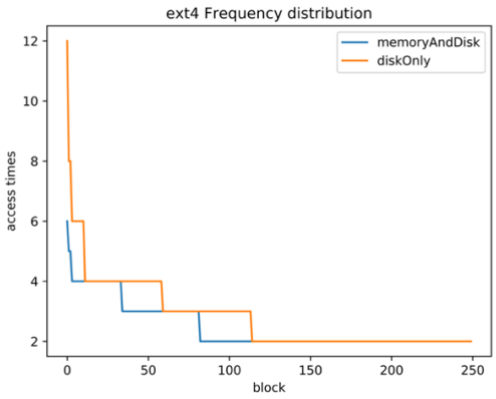




Overall Comparison:

Focusing on the most frequent 250 blocks:



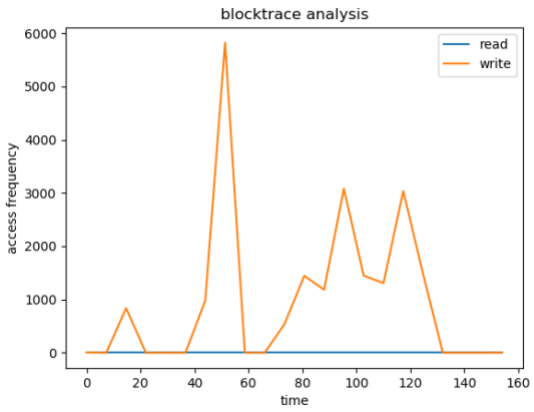


Analysis:

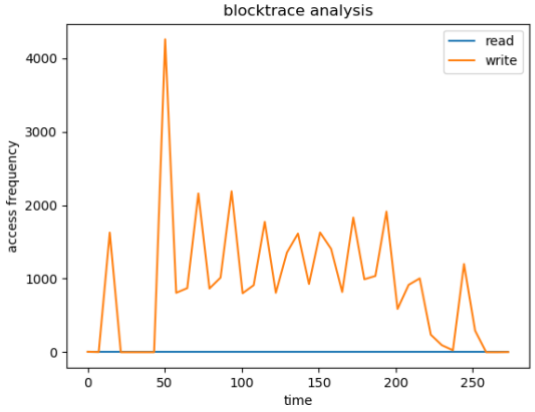
As shown by the graphs, DISK_ONLY mode accesses blocks more frequently than MEMORY_AND_DISK. In other words, more blocks get accessed(more than 17500 blocks vs more than 10000 blocks) and the most frequently accessed blocks get accessed much more frequently too(12 times vs 6 times) .

Both modes follow a typical zipfian curve, only 150/17500(less than 1%) of the blocks get accessed more than twice. A significant access frequency drop happens at the top 10 blocks.

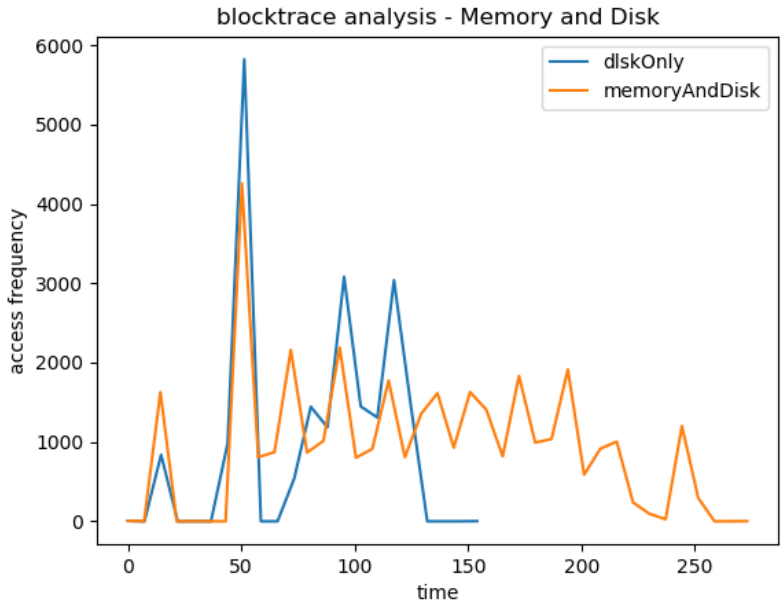
R&W ratio for MEMORY_AND_DISK:



R&W ratio for DISK_ONLY:



Overall Comparison:

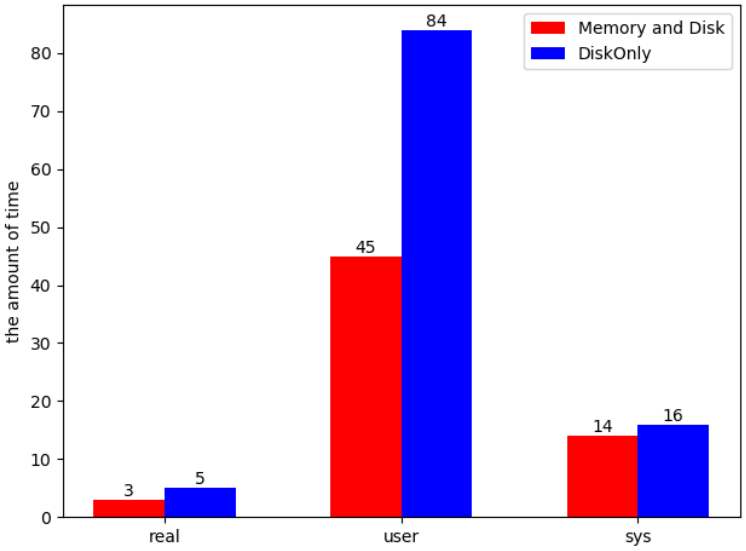


Analysis:
From the diagram of Overall Comparison in this part, at first diskOnly has larger access frequency than memoryAndDisk, while its access frequency vanishes to 0 much faster than memoryAndDisk.

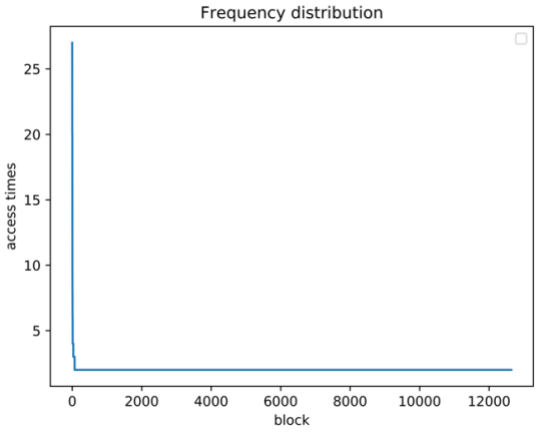
1.2. In btrfs

	MEMORY_AND_DISK	DISK_ONLY
Performance drop (the amount of time)	Without blktrace real 3m7.707s user 44m4.280s sys 13m56.787s With blktrace real 3m10.161s user 44m40.151s sys 13m50.994s	Without blktrace real 4m48.651s user 80m34.261s sys 16m9.468s With blktrace real 5m2.474s user 83m55.573s sys 15m57.412s
Blktrace file size	1919724	2519014
blocks written	12,743, 14,419MiB	16,736, 14,772MiB
blocks read	0, 0KiB	0, 0KiB

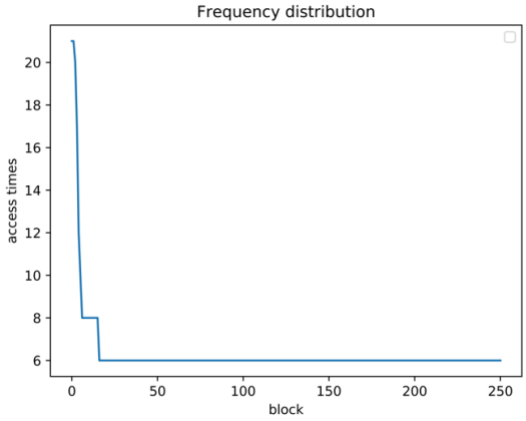
Amount of time needed for MEMORY_AND_DISK & DISK_ONLY with blktrace:



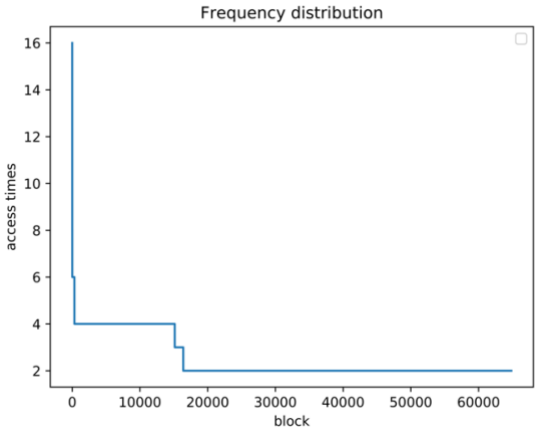
Frequency distribution for MEMORY_AND_DISK:



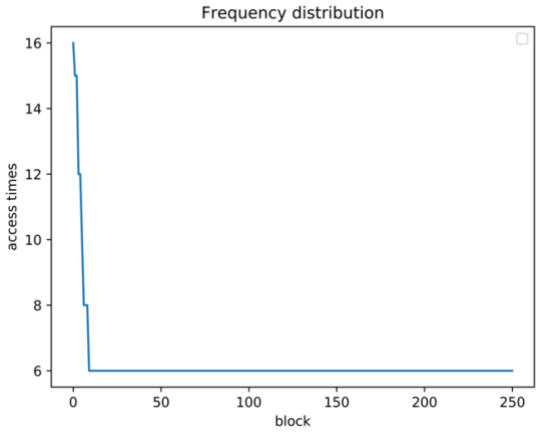
Focusing on the most frequent 250 blocks:



Frequency distribution for DISK_ONLY:

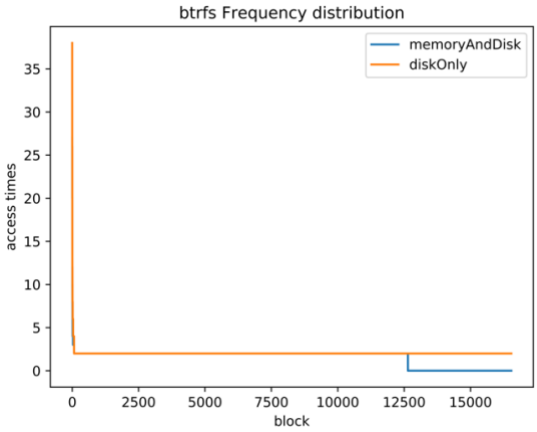


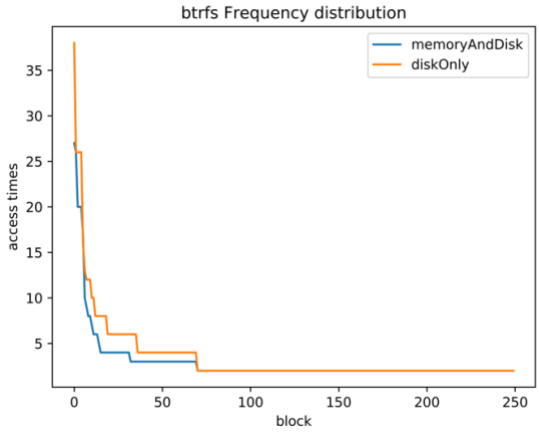
Focusing on the most frequent 250 blocks:



Overall Comparison:

Focusing on the most frequent 250 blocks:



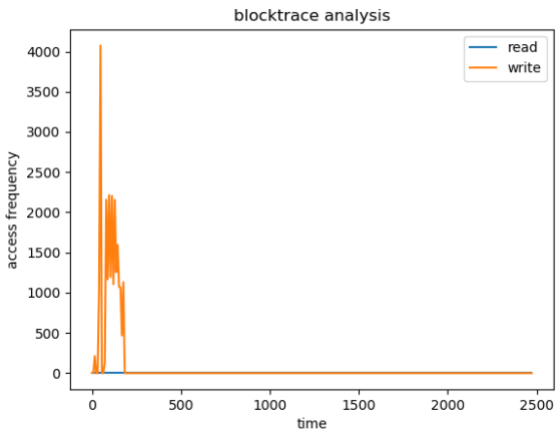


Analysis:

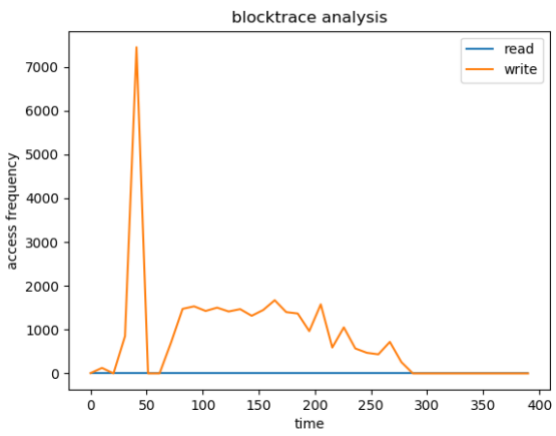
As shown by the graphs, DISK_ONLY mode accesses blocks more frequently than MEMORY_AND_DISK. In other words, more blocks get accessed(less than 17500 blocks vs more than 12500 blocks) and the most frequently accessed blocks get accessed much more frequently too(39 times vs 27 times) .

Both modes follow a typical zipfian curve, only 100/17500(less than 1%) of the blocks get accessed more than twice. A significant access frequency drop happens at the top 20 blocks.

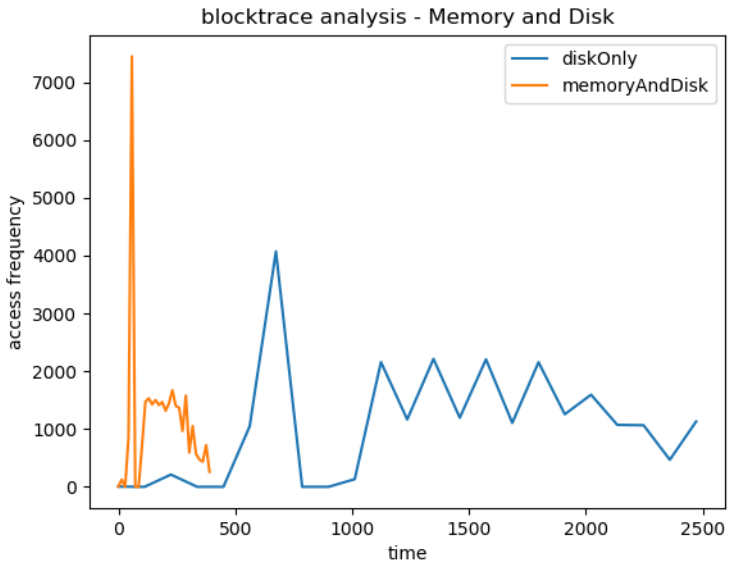
R&W ratio for MEMORY_AND_DISK:



R&W ratio for DISK_ONLY:



Overall Comparison:



Analysis:

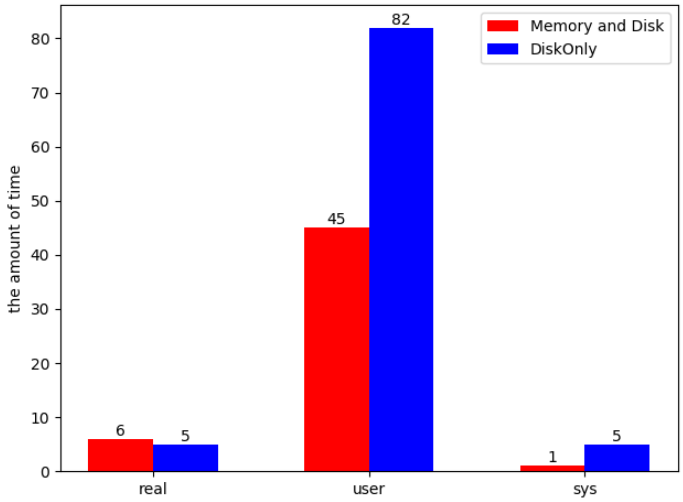
From the diagram of Overall Comparison in this part, at first diskOnly has less access frequency than memoryAndDisk, while its access frequency vanishes to 0 much slower than memoryAndDisk.

1.3. In f2fs

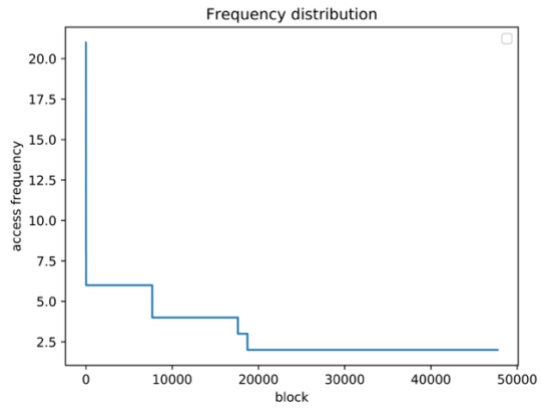
	MEMORY_AND_DISK	DISK_ONLY
Performance drop (the amount of time)	Without blktrace real 4m36.656s user 44m14.205s sys 1m6.152s With blktrace real 5m39.593s user 44m38.928s	Without blktrace real 5m14.525s user 82m4.652s sys 4m46.754s With blktrace real 4m58.157s user 82m25.109s

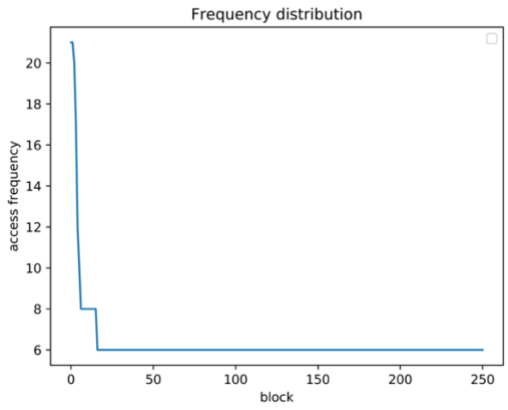
	sys 1m6.234s	sys 4m43.383s
Blktrace file size	10028000	11107003
blocks written	73,079, 10,945MiB	80,354, 19,365MiB
blocks read	0, 0KiB	0, 0KiB

Amount of time needed for MEMORY_AND_DISK & DISK_ONLY with blktrace:

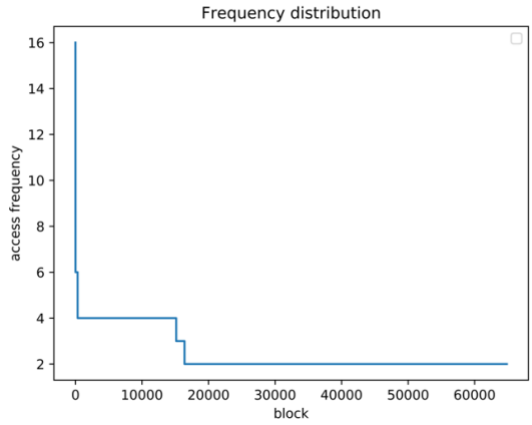


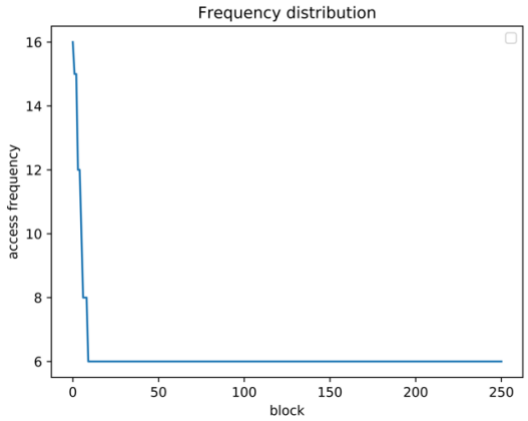
Frequency distribution for MEMORY_AND_DISK: Focusing on the most frequent 250 blocks:





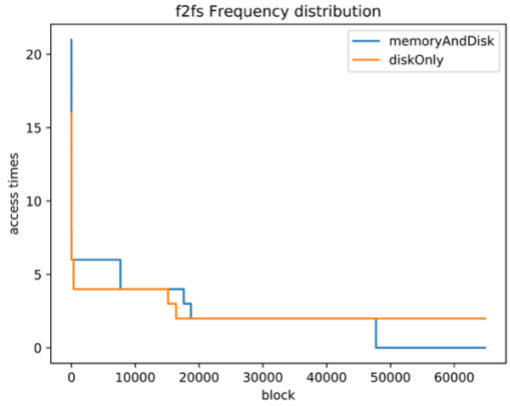
Frequency distribution for DISK_ONLY: Focusing on the most frequent 250 blocks:

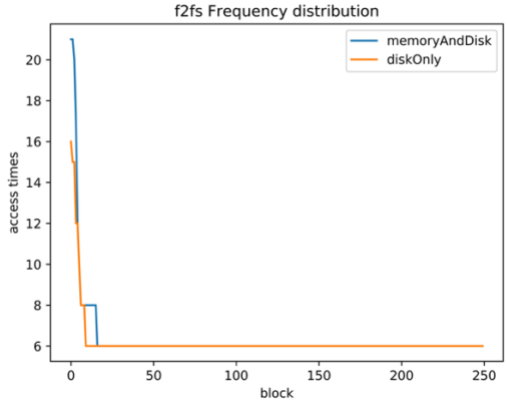




Overall Comparison:

Focusing on the most frequent 250 blocks:





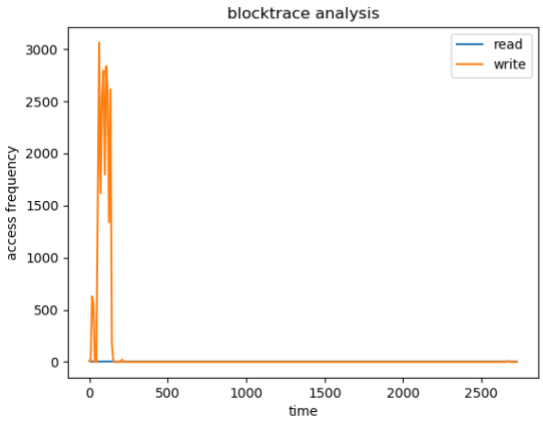
Analysis:

As shown by the graphs, DISK_ONLY mode accesses blocks more frequently than MEMORY_AND_DISK. In other words, more blocks get accessed(less than 70000 blocks vs more than 50000 blocks) and the most frequently accessed blocks get accessed much more frequently too(25 times vs 16 times) .

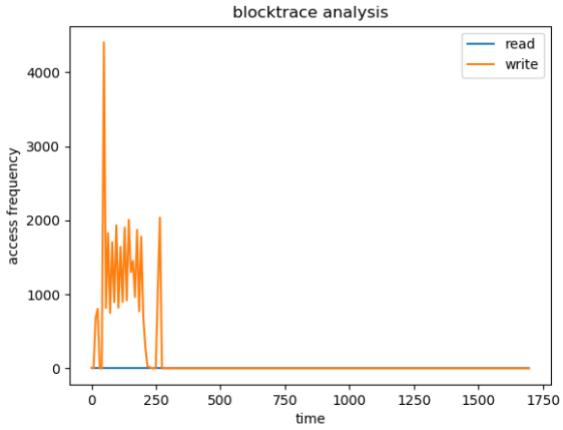
Both modes access blocks more evenly compared to ext4 and btrfs, less than 20000/70000(around 28%) of the blocks get accessed more than twice. A significant access frequency drop happens at the top 15 blocks.

The more detailed comparison between three file systems will be shown in part 2.

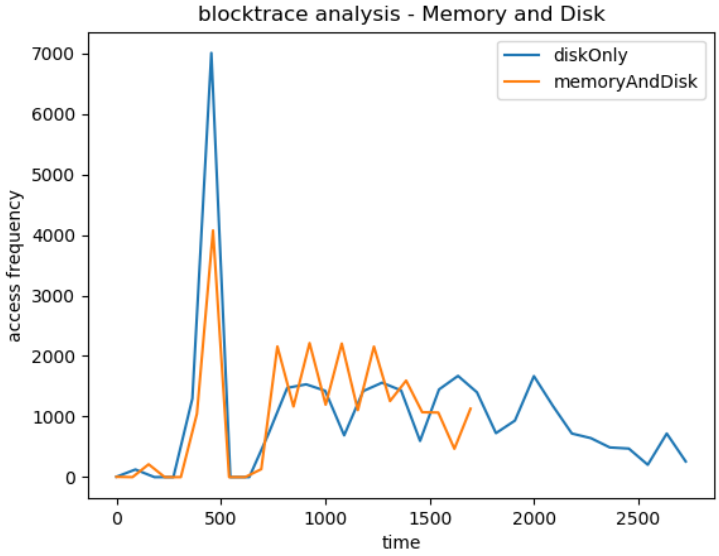
Read and write ratio for MEMORY_AND_DISK:



Read and write ratio for DISK_ONLY:



Overall Comparison:



Analysis:

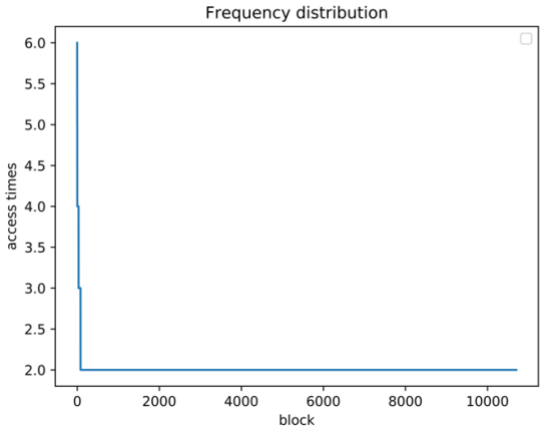
From the diagram of Overall Comparison in this part, at first diskOnly has larger access frequency than memoryAndDisk, but later memoryAndDisk has larger access frequency. The access frequency of memoryAndDisk vanishes to 0 much faster than diskOnly.

2. File System Comparison

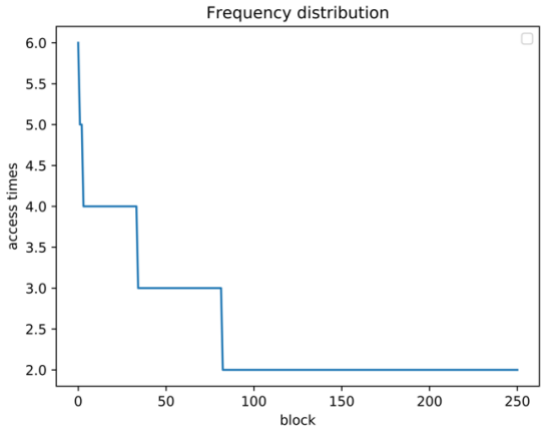
2.1. memoryAndDisk:

	ext4	Btrfs	F2FS
Performance drop (the amount of time)	Without blktrace real 2m18.379s user 45m4.699s sys 1m7.734s With blktrace real 2m13.066s user 44m53.089s sys 1m8.120s	Without blktrace real 3m7.707s user 44m4.280s sys 13m56.787s With blktrace real 3m10.161s user 44m40.151s sys 13m50.994s	Without blktrace real 4m36.656s user 44m14.205s sys 1m6.152s With blktrace real 5m39.593s user 44m38.928s sys 1m6.234s
Blktrace file size	1634736	1919724	10028000
blocks written	10,761, 9,112MiB	12,743, 14,419MiB	73,079, 10,945MiB
blocks read	0, 0KiB	0, 0KiB	0, 0KiB

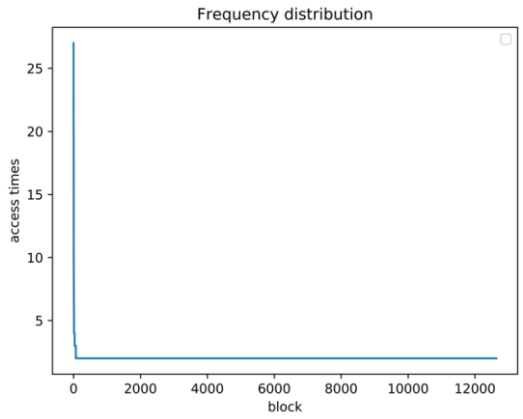
Frequency distribution for ext4:



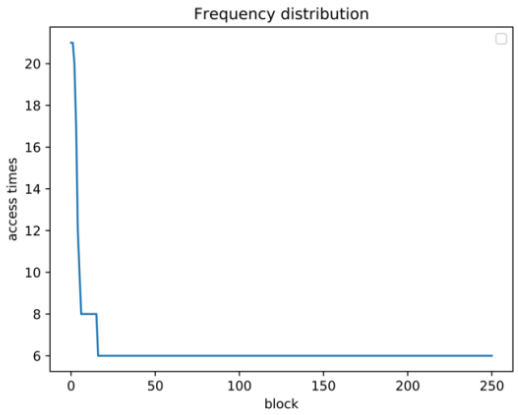
Focusing on the most frequent 250 blocks:



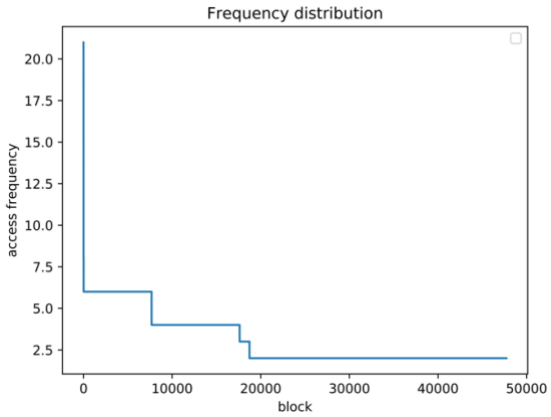
Frequency distribution for Btrfs:



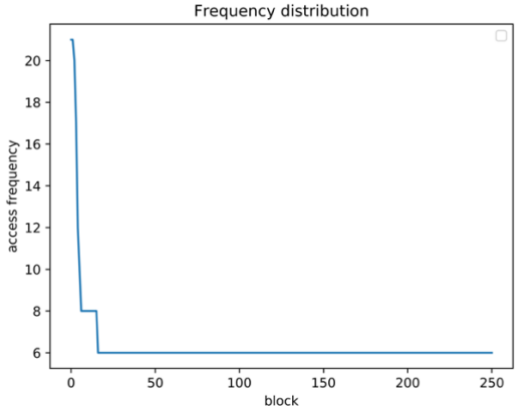
Focusing on the most frequent 250 blocks:



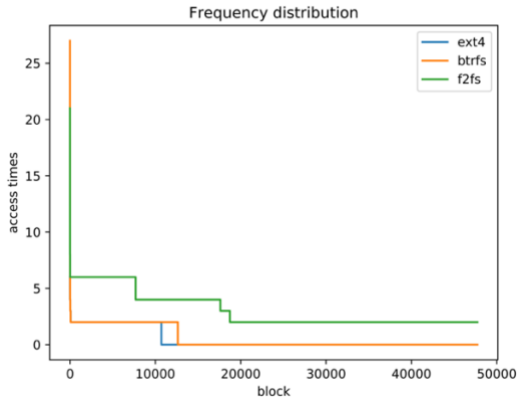
Frequency distribution for F2FS:



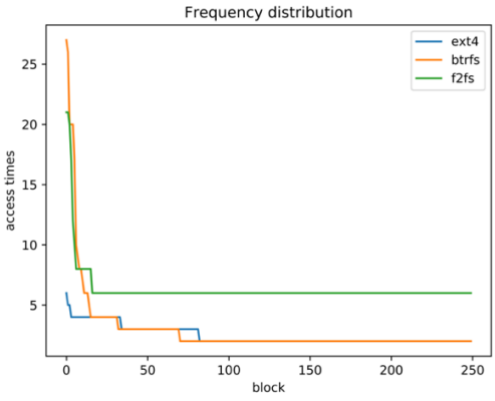
Focusing on the most frequent 250 blocks:



Overall Comparison:



Focusing on the most frequent 250 blocks:



Analysis:

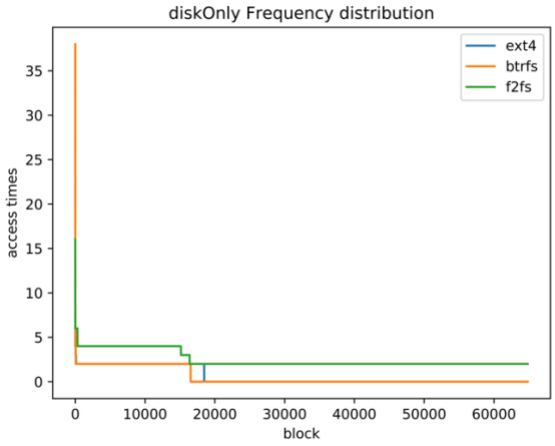
With regards to btrfs, it follows the zipfian curve most, since it has the highest access times of the most frequently used blocks and then drops fastest, using leaving most of the blocks accessed around twice.

With regards to ext4, it accesses blocks much less than btrfs and f2fs. This might be because more data is written via each block access. If this is the case, the block access frequency may change if the page size changes. This is related to the Access Granularity part in "*Avoiding the Streetlight Effect: I/O Workload Analysis with SSDs in Mind*".

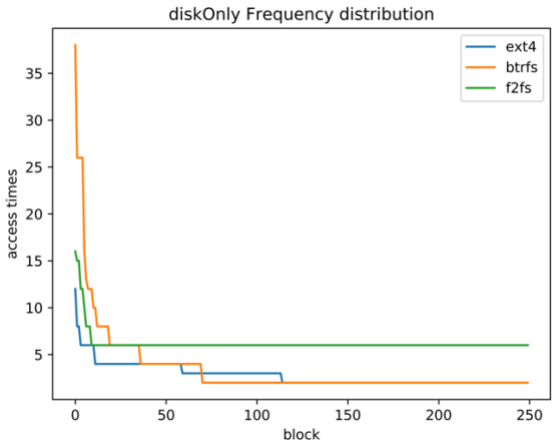
With regards to f2fs, it evens the block access frequency much better than btrfs.

For SSD, we are looking for the file system that avoids accessing some blocks very frequently since this causes the lifetime of those blocks. Therefore, we prefer ext4, than f2fs, and than btrfs.

2.2. diskOnly:



Focusing on the most frequent 250 blocks:

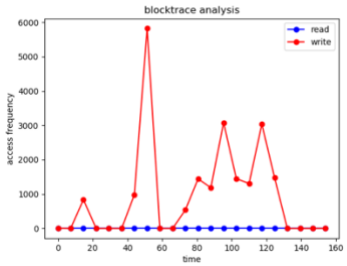


As we can see, the diskOnly mode shows the same pattern with memoryAndDisk

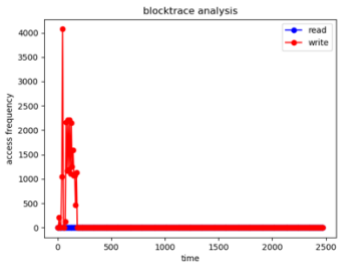
Analysis:

At very beginning, ext4 has the largest access frequency, then the btrfs, and the last is f2fs. ext4 and btrfs' s access frequencies vanish fast to zero. By contract, the access frequency of f2fs file system then increases and vanishes to zero much slower than the ext4 and btrfs.

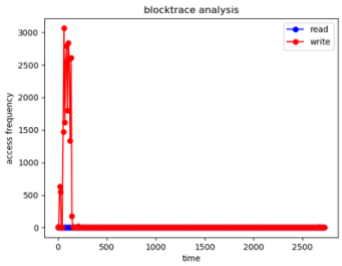
ext4 R/W ratio (DiskOnly):



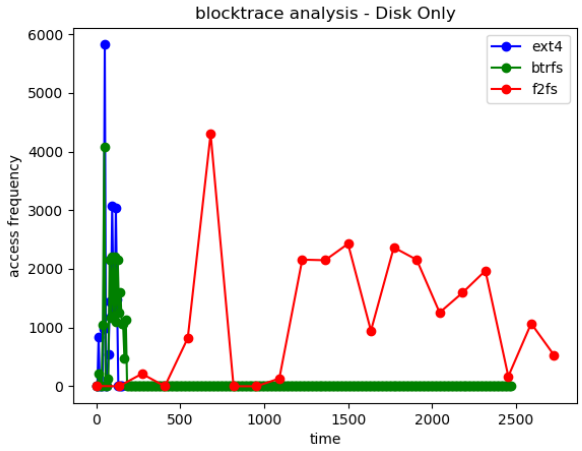
Btrfs R/W ratio (DiskOnly):



F2FS R/W ratio (DiskOnly):

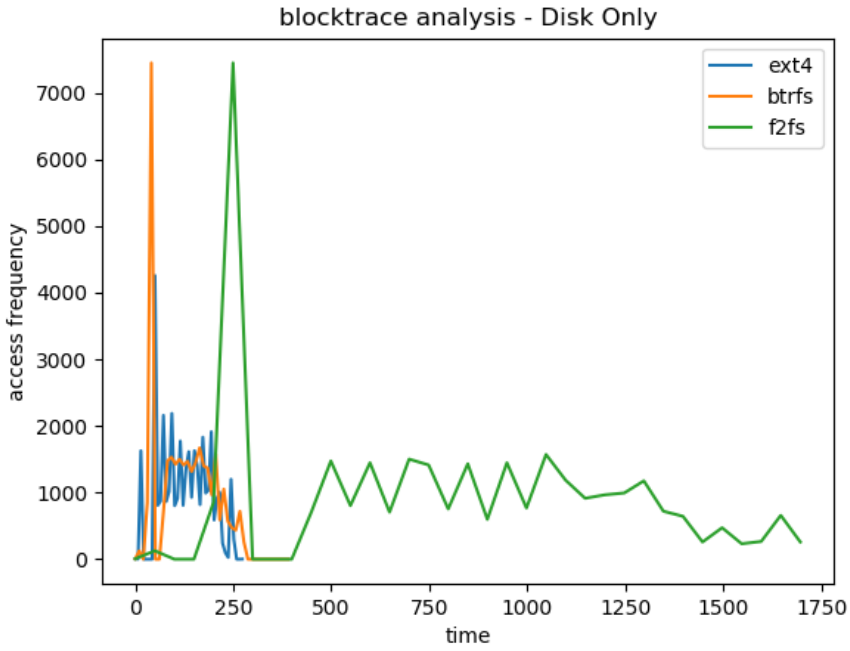


Overall Comparison of three file system (DiskOnly):

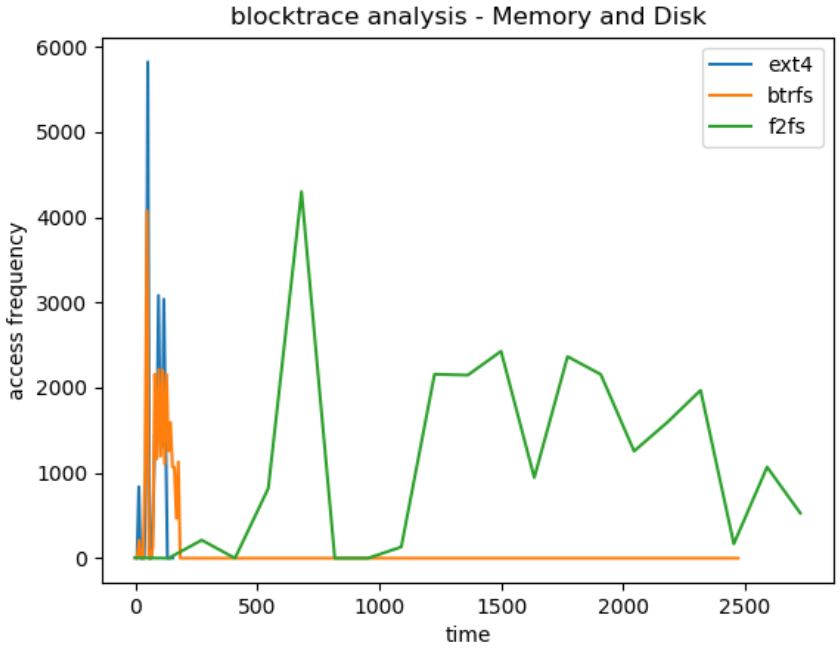


3. Comparison of diskOnly and memoryAndDisk of three file systems:

- diskOnly blocktrace of ext4, btrfs and f2fs:



- memoryAndDisk blocktrace of ext4, btrfs and f2fs:



4. Related Files

4.1. Workload

spark-2.4.0-bin-hadoop2.7 sx-stackoverflow.edges

Stored under zhangf68@swift-014:~/spark$

4.2. Blktrace's, Png's, Csv's

Stored under zhangf68@swift-014:~$

4.3. Commands

Named as *CSC494 commands*