CSC494 Report

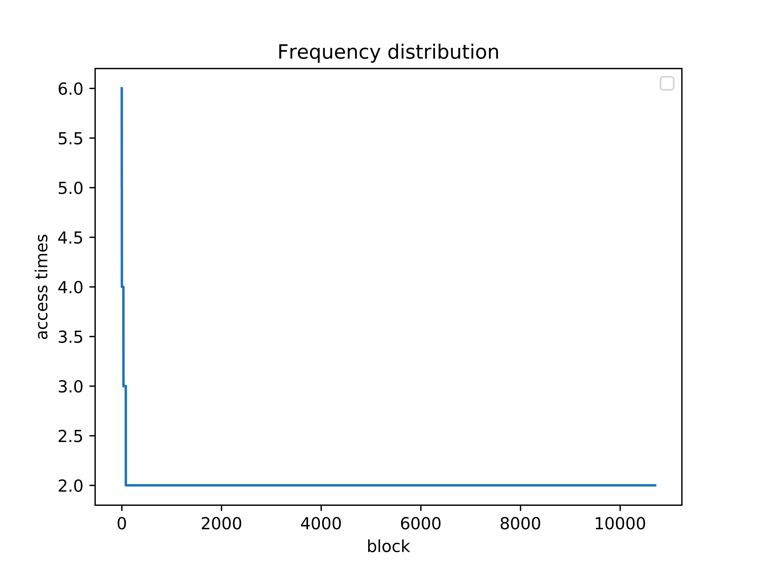
Fengjia Zhang

Yuhan Shao

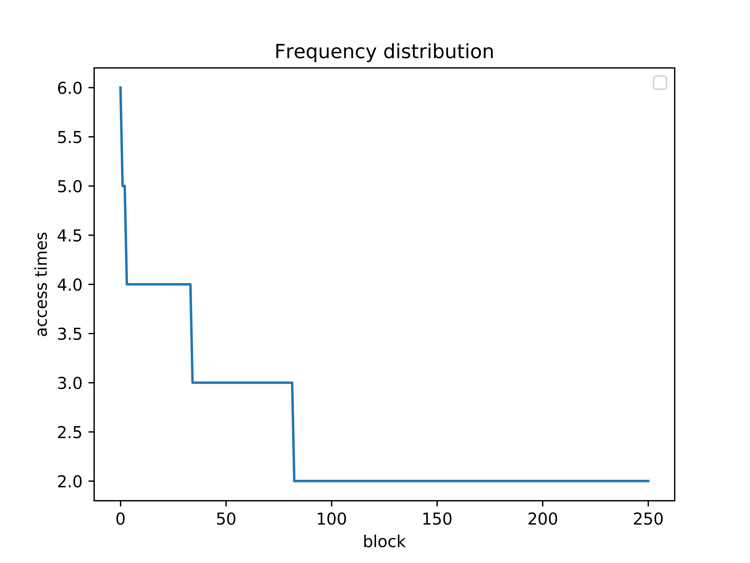
1. DISK\_ONLY and MEMORY\_AND\_DISK comparison in spark command
   1. In ext4

|  |  |  |
| --- | --- | --- |
|  | 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 |

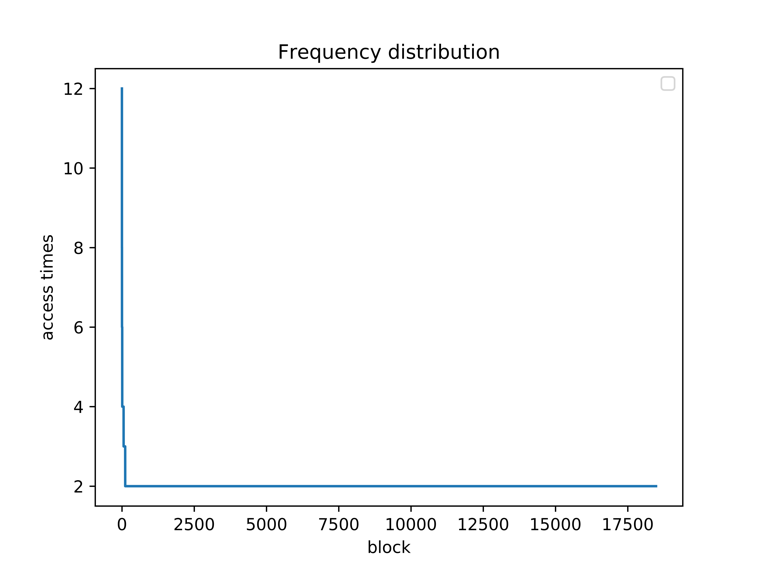
Frequency distribution for MEMORY\_AND\_DISK:



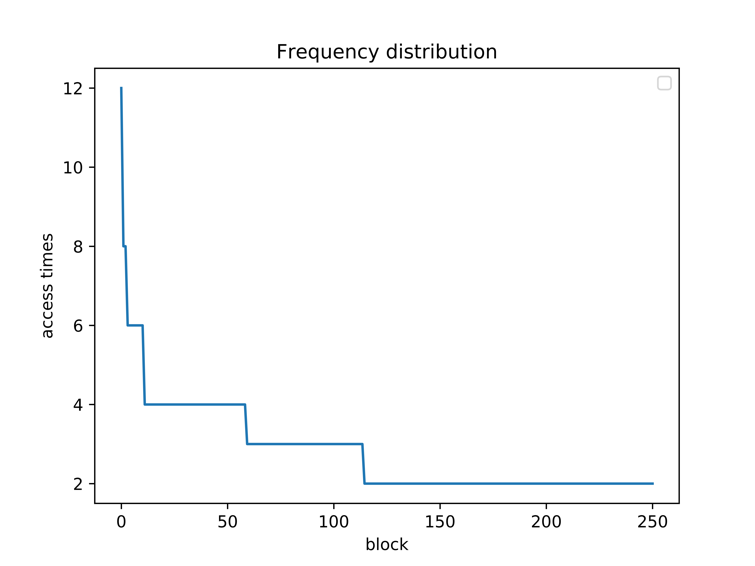
Focusing on the most frequent 250 blocks:



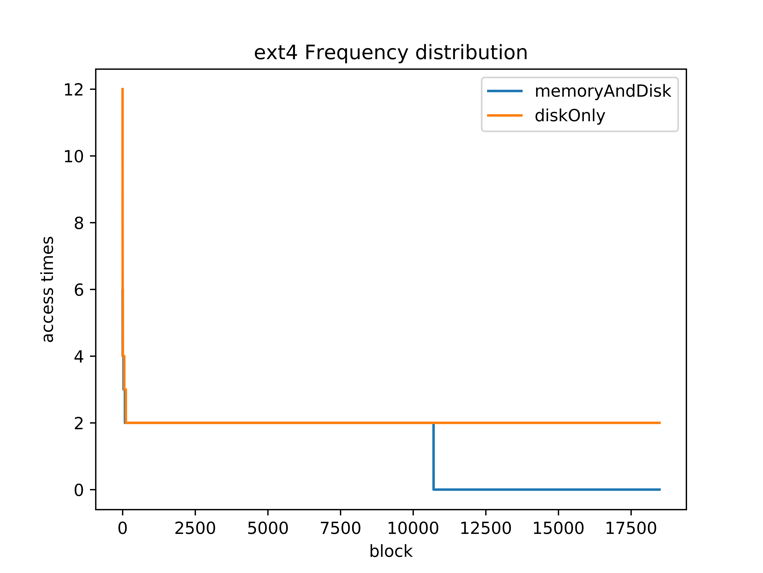
Frequency distribution for DISK\_ONLY:



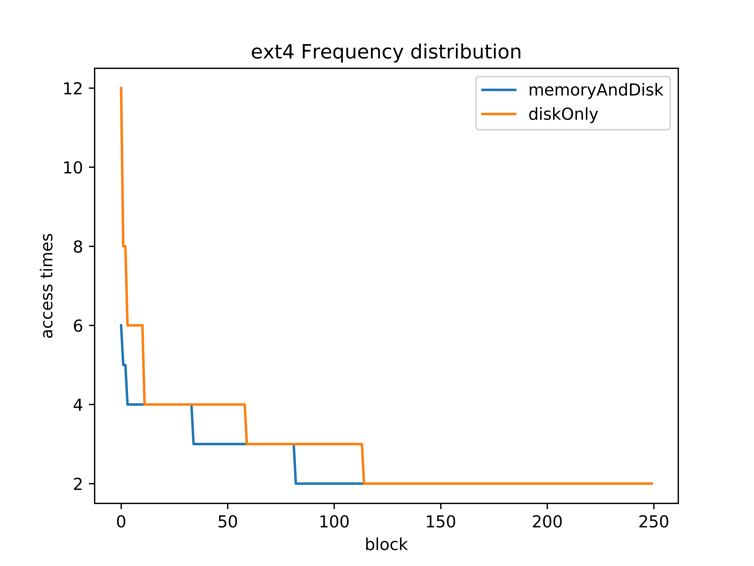
Focusing on the most frequent 250 blocks:



Overall Comparison:



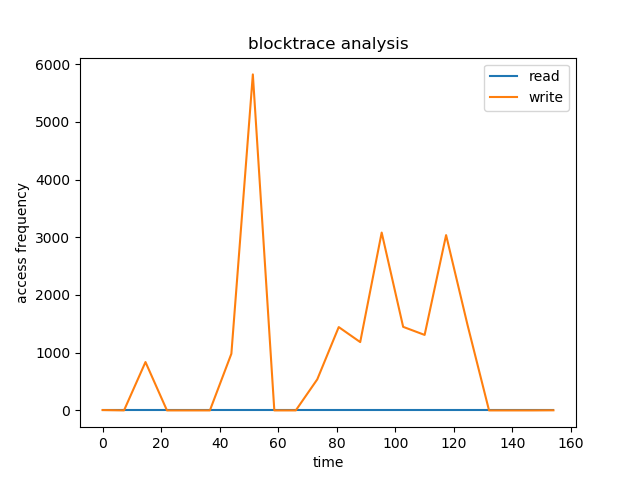
Focusing on the most frequent 250 blocks:



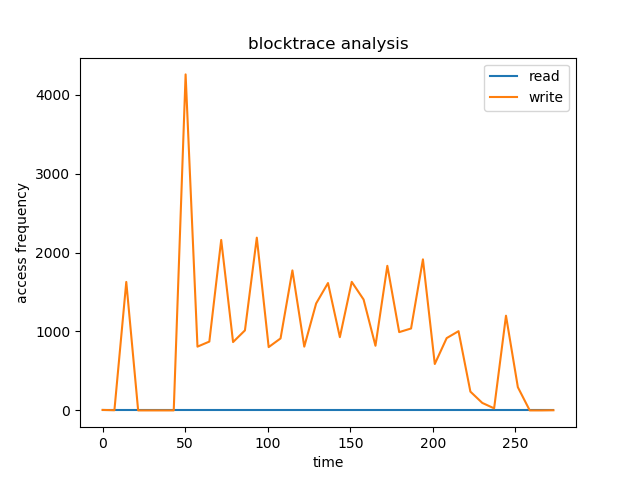
Analysis:

Text

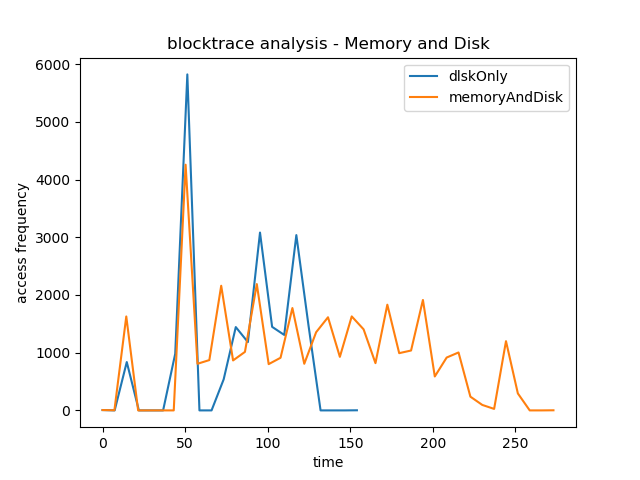
Read and write ratio for MEMORY\_AND\_DISK:



Read and write ratio for DISK\_ONLY:



Overall Comparison:



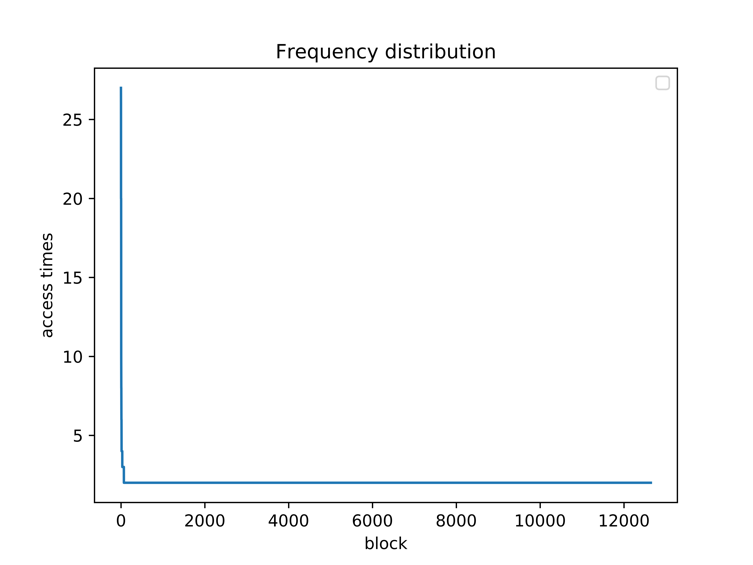
Analysis:

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

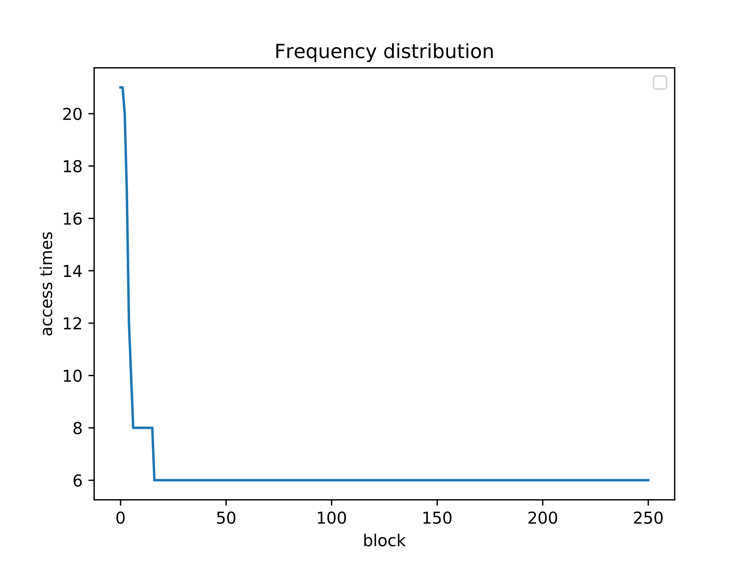
* 1. In btrfs

|  |  |  |
| --- | --- | --- |
|  | 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 |

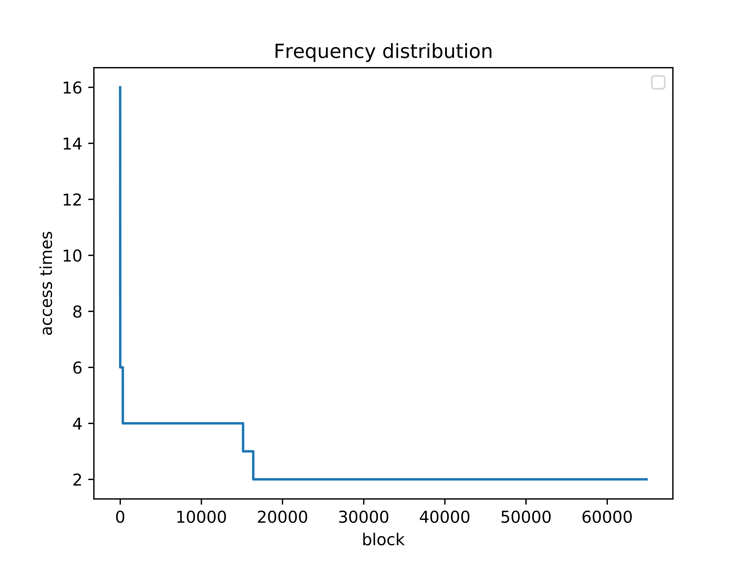
Frequency distribution for MEMORY\_AND\_DISK:



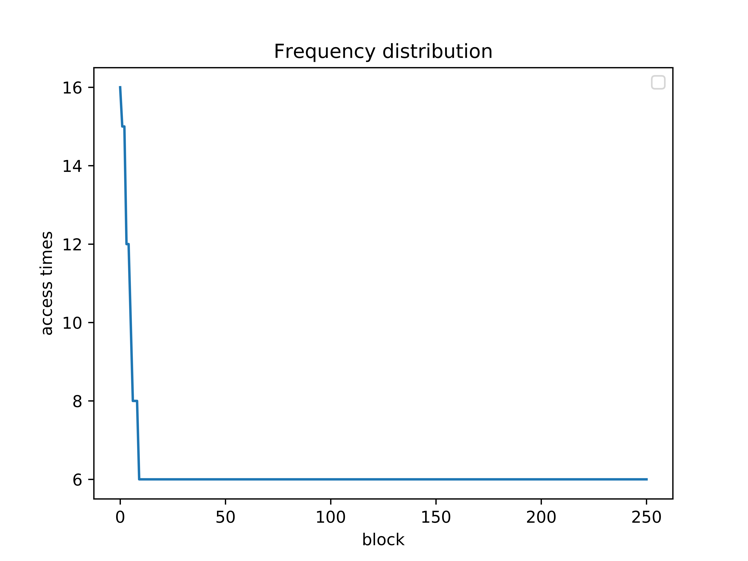
Focusing on the most frequent 250 blocks:



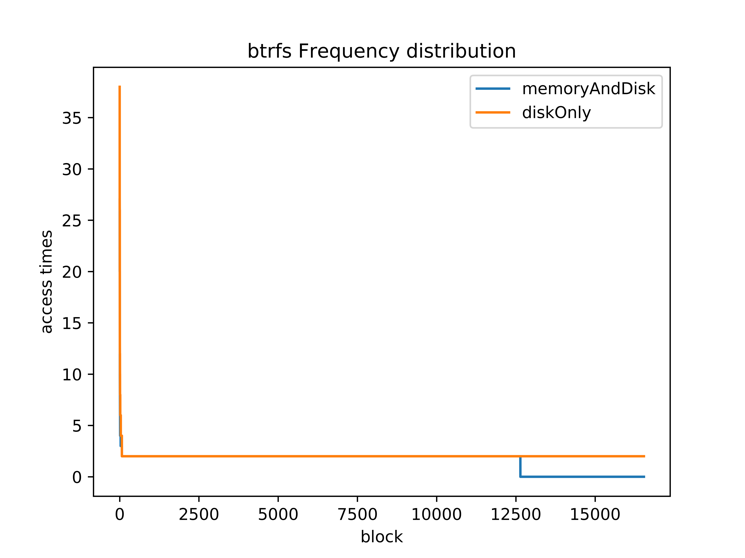
Frequency distribution for DISK\_ONLY:



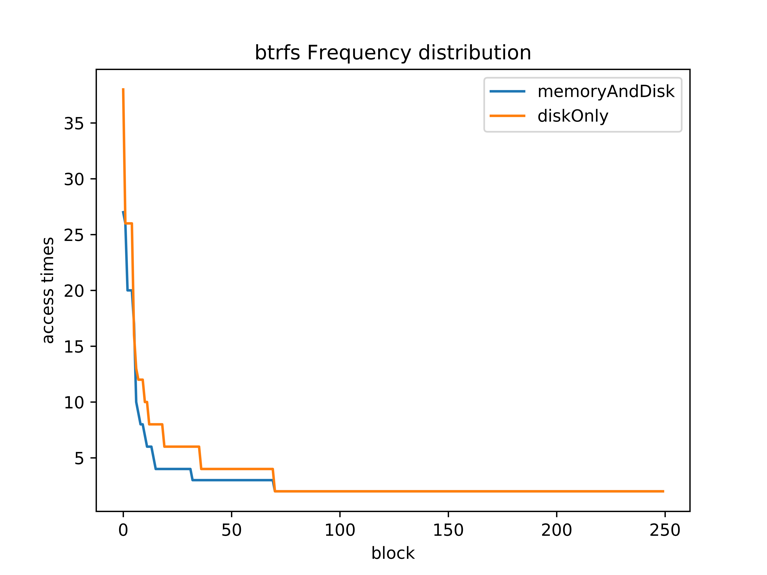
Focusing on the most frequent 250 blocks:



Overall Comparison:



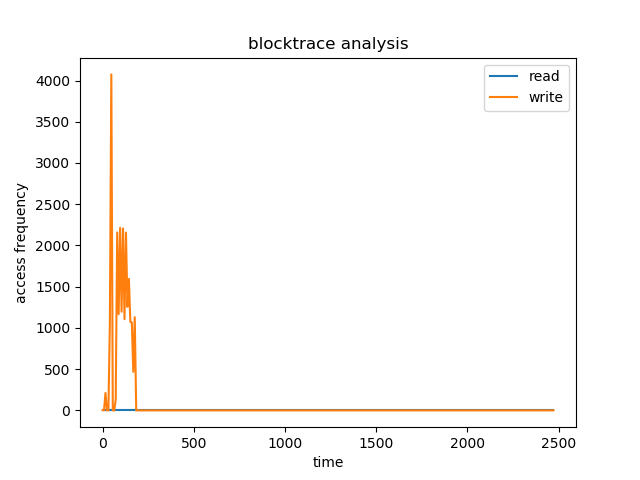
Focusing on the most frequent 250 blocks:

****

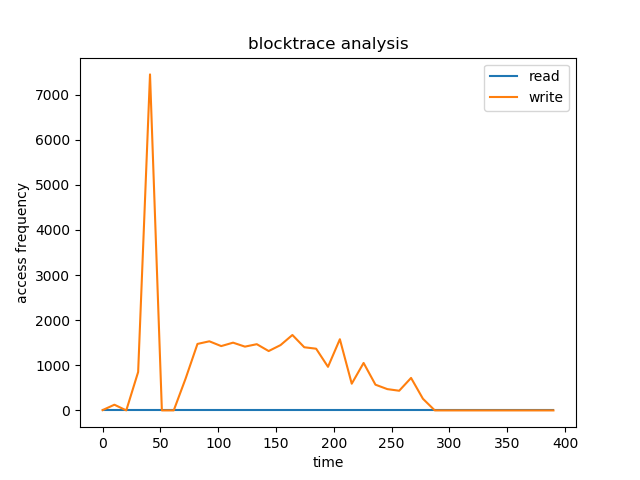
Analysis:

Text

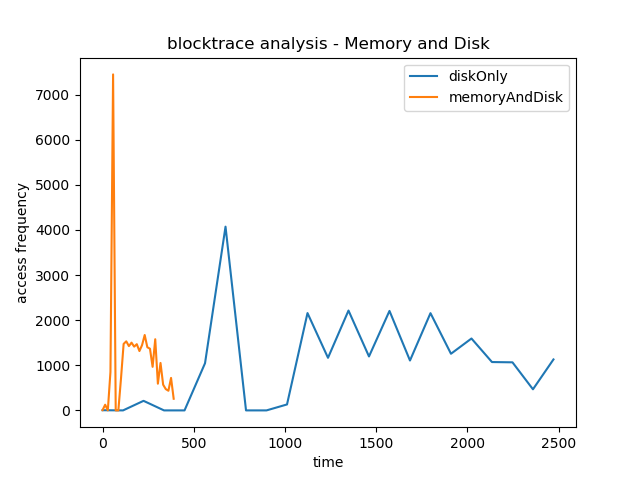
Read and write ratio for MEMORY\_AND\_DISK:



Read and write ratio for DISK\_ONLY:



Overall Comparison:



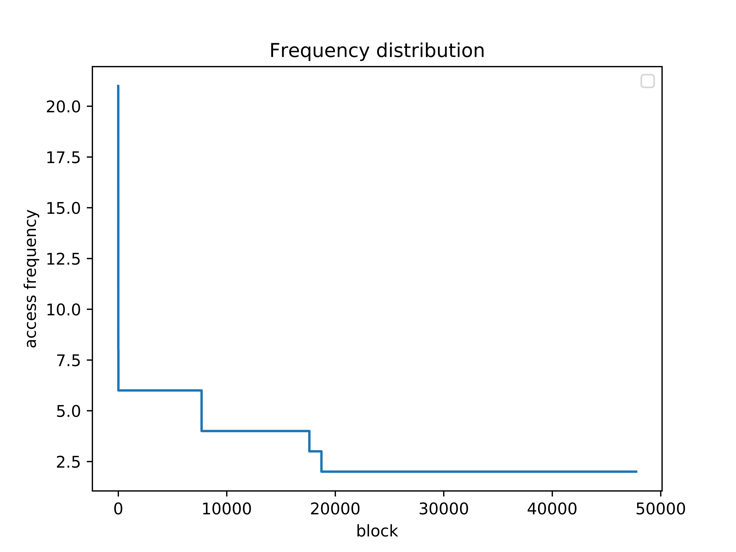
Analysis:

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

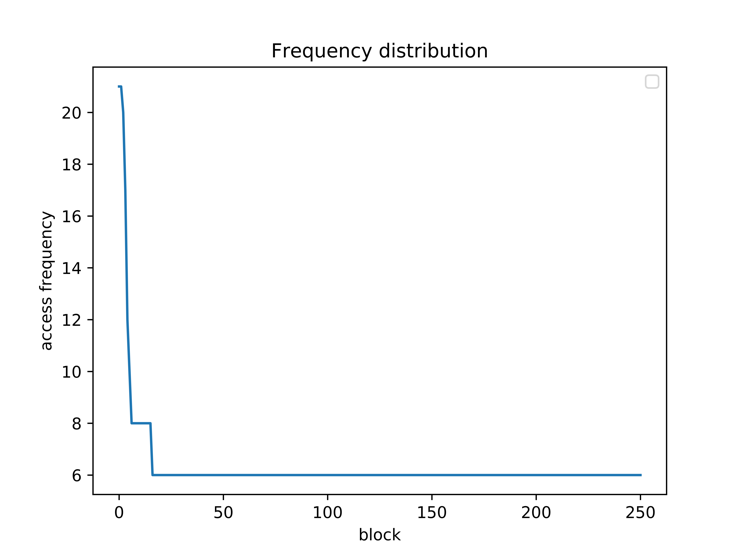
* 1. In f2fs

|  |  |  |
| --- | --- | --- |
|  | 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  sys 1m6.234s | Without blktrace  real 5m14.525s  user 82m4.652s  sys 4m46.754s  With blktrace  real 4m58.157s  user 82m25.109s  sys 4m43.383s |
| Blktrace file size | 10028000 | 11107003 |
| blocks written | 73,079, 10,945MiB | 80,354, 19,365MiB |
| blocks read | 0, 0KiB | 0, 0KiB |

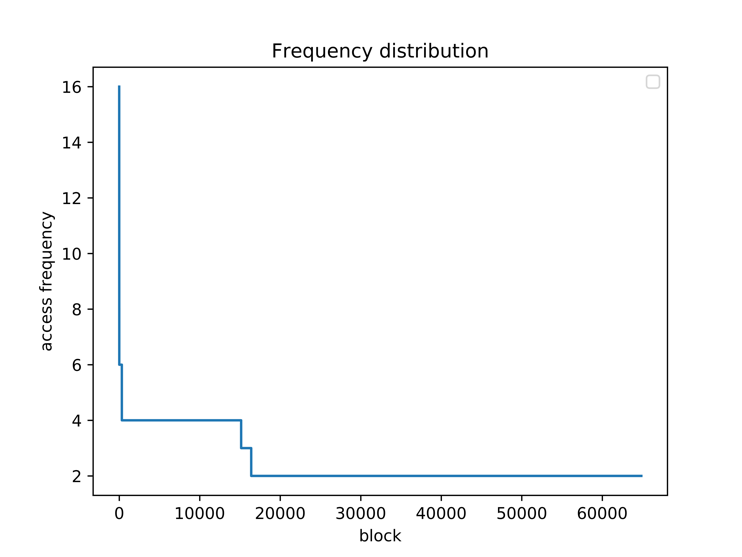
Frequency distribution for MEMORY\_AND\_DISK:



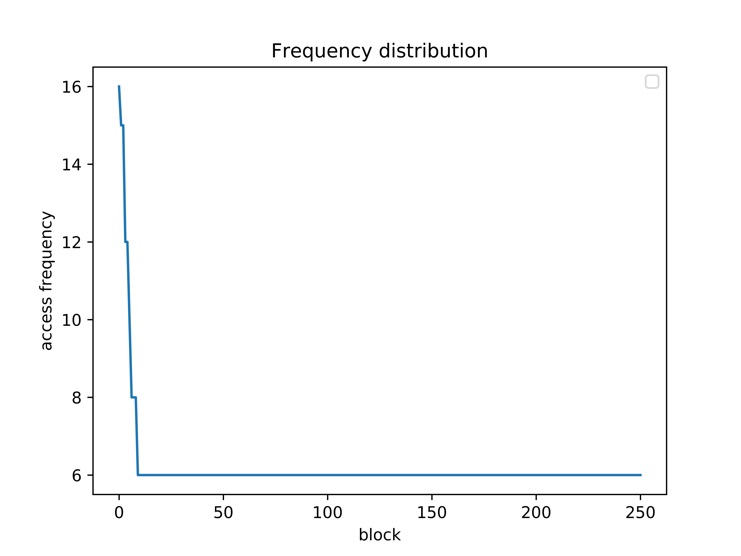
Focusing on the most frequent 250 blocks:



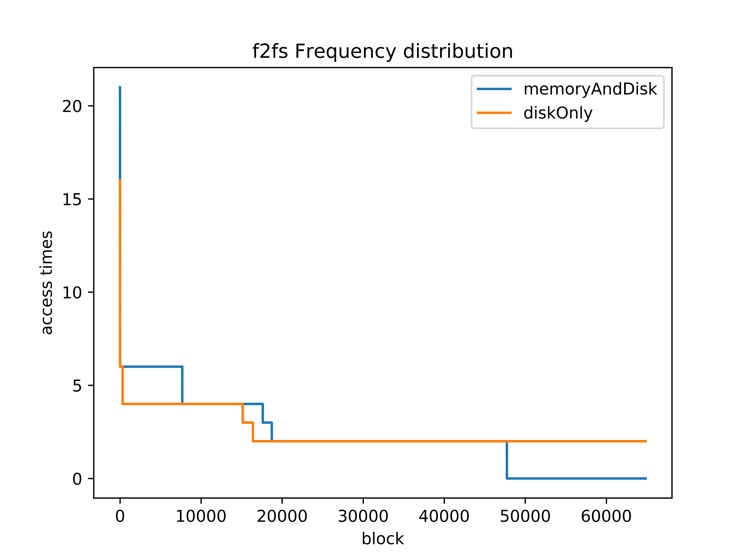
Frequency distribution for DISK\_ONLY:



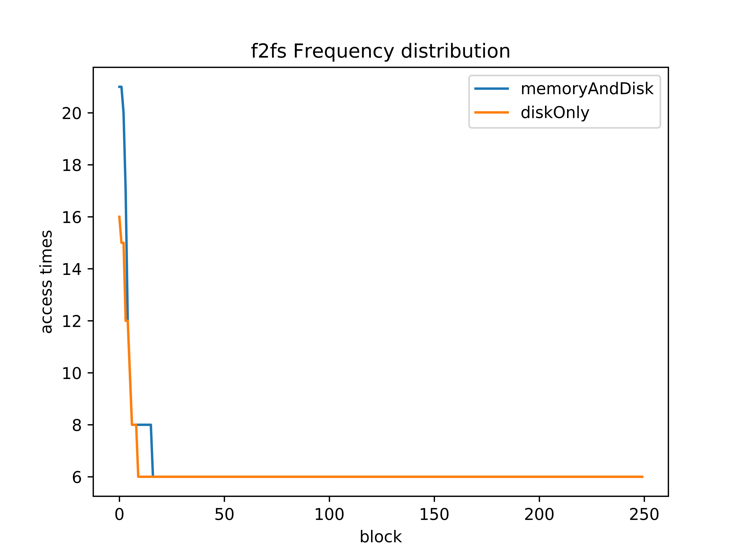
Focusing on the most frequent 250 blocks:



Overall Comparison:



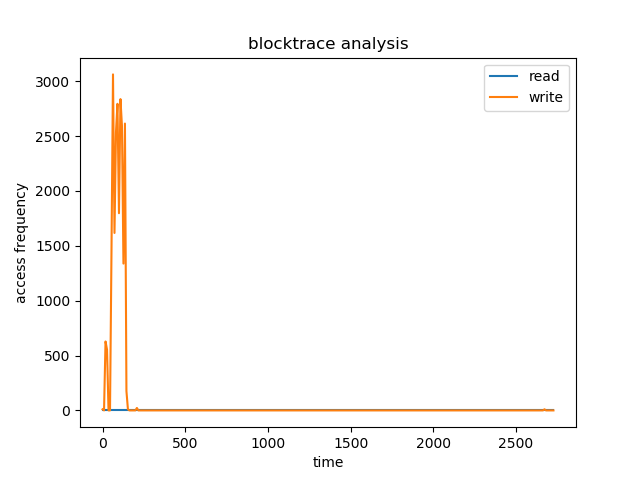
Focusing on the most frequent 250 blocks:

****

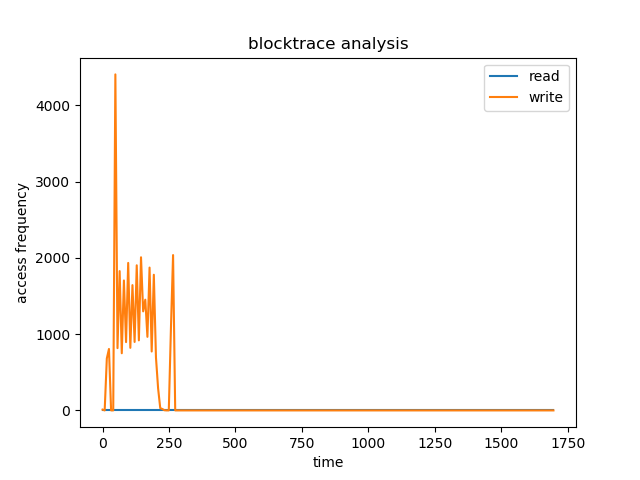
Analysis:

Text

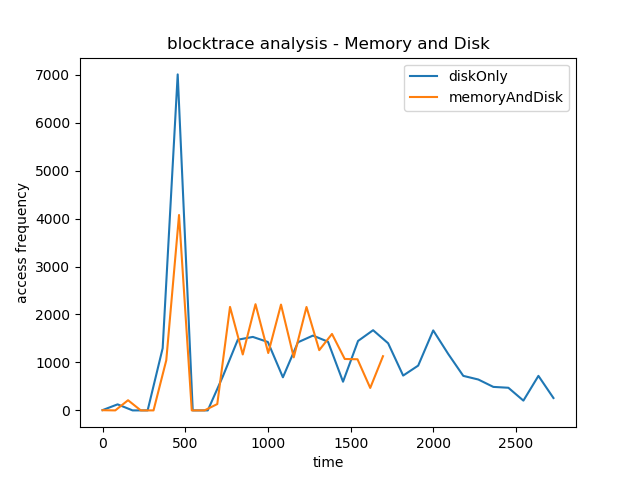
Read and write ratio for MEMORY\_AND\_DISK:



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.

1. File System Comparison
   1. Command:
      1. Blocktrace

mkdir blocktrace

sudo mkfs.ext4 /dev/nvme2n1

sudo mount /dev/nvme2n1 blocktrace

sudo blktrace -d /dev/nvme1n1 -w 3600 -o - | blkparse -a fs -i - > blocktrace/filename

* + 1. Spark on different file system

mkdir spark

sudo mkfs.ext4 /dev/nvme1n1 //ext4

sudo mkfs.btrfs -f /dev/nvme1n1 // Btrfs

sudo mkfs.f2fs /dev/nvme1n1 // F2FS

sudo mount /dev/nvme1n1 spark

sudo mount // to check whether the file system is formatted correctly

// sudo umount spark

// copy the spark to the folder spark

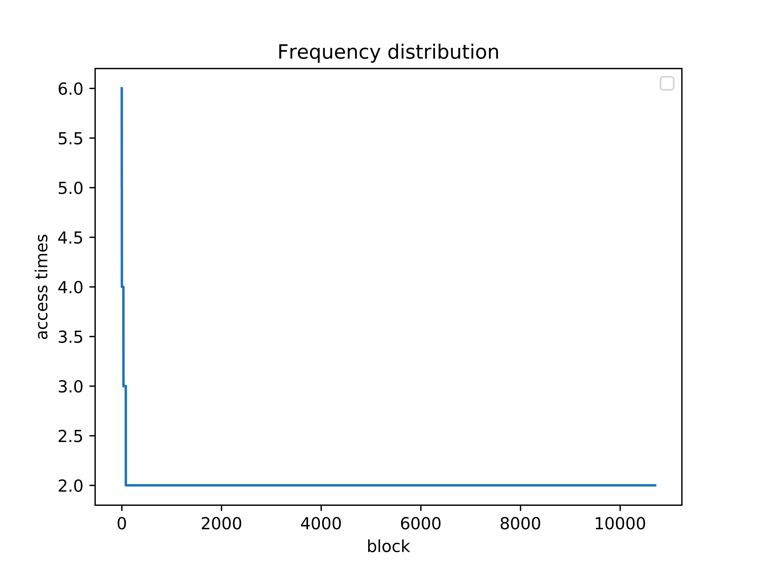
scp -r shaoyuha@swift-014.syslab.sandbox:/home/shaoyuha/spark-2.4.0-bin-hadoop2.7 ./spark

time ./bin/run-example graphx.Analytics pagerank ../twitter\_rv.net --numEPart=-1 --vertexStorageLevel=MEMORY\_AND\_DISK --edgeStorageLevel=MEMORY\_AND\_DISK

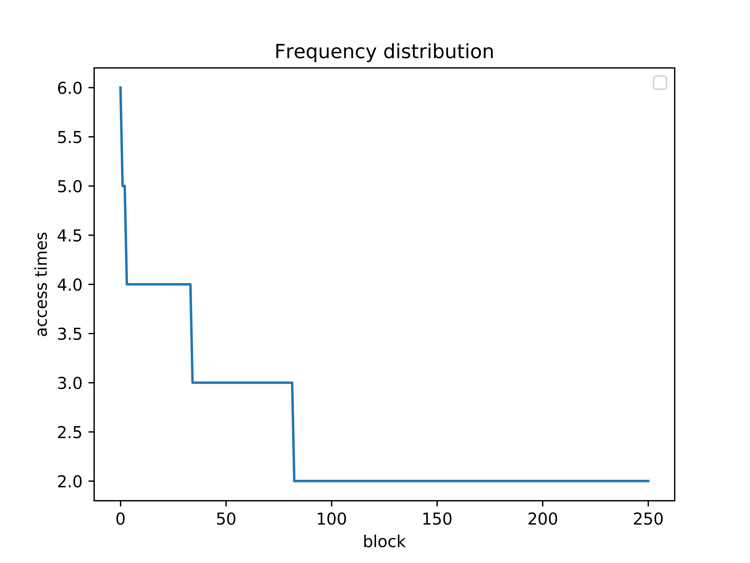
* 1. Output:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 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 |

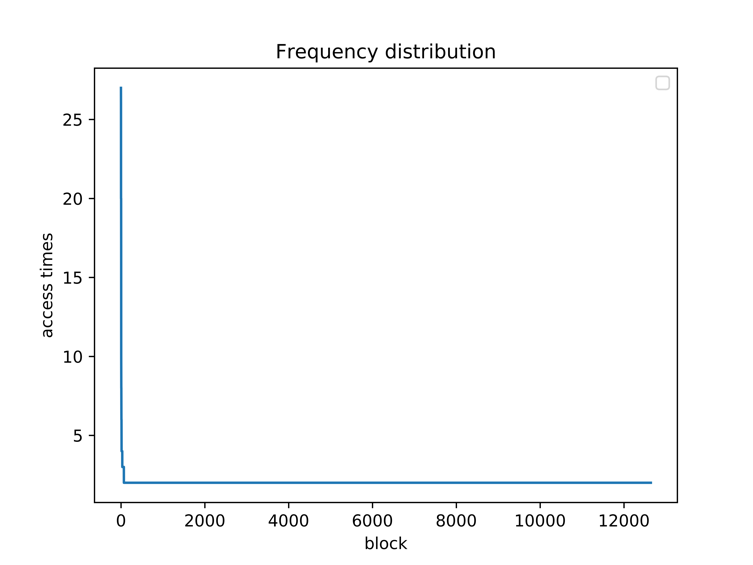
Frequency distribution for ext4:



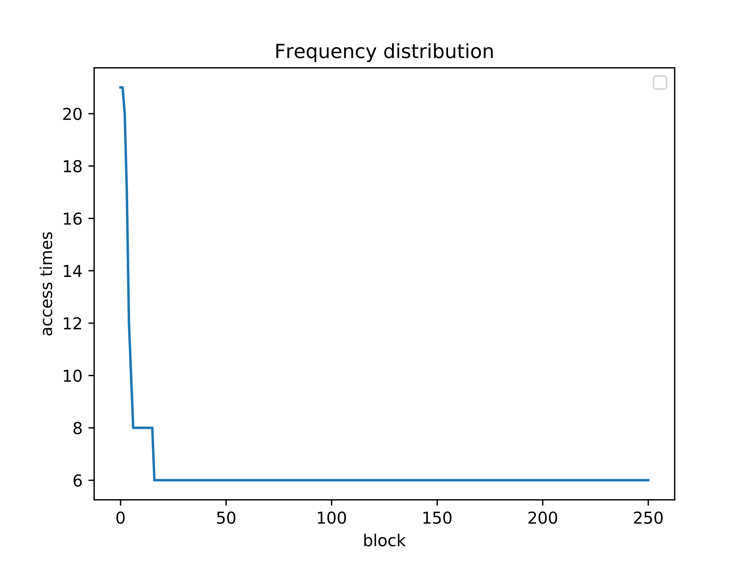
Focusing on the most frequent 250 blocks:



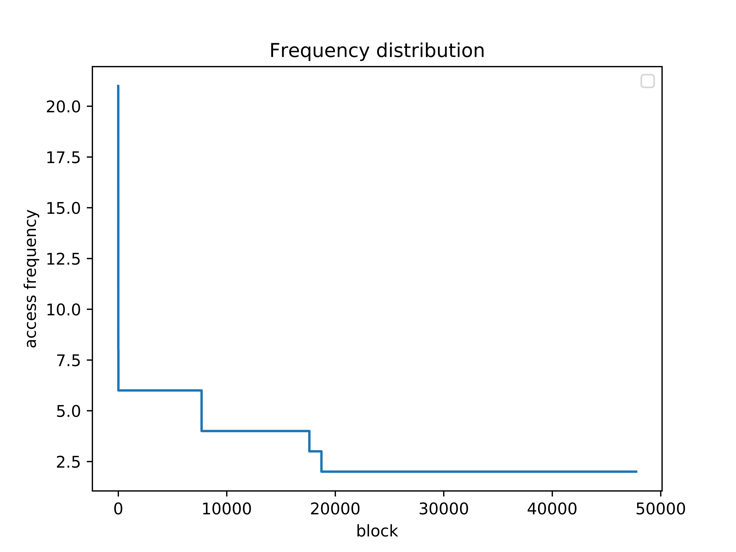
Frequency distribution for Btrfs:



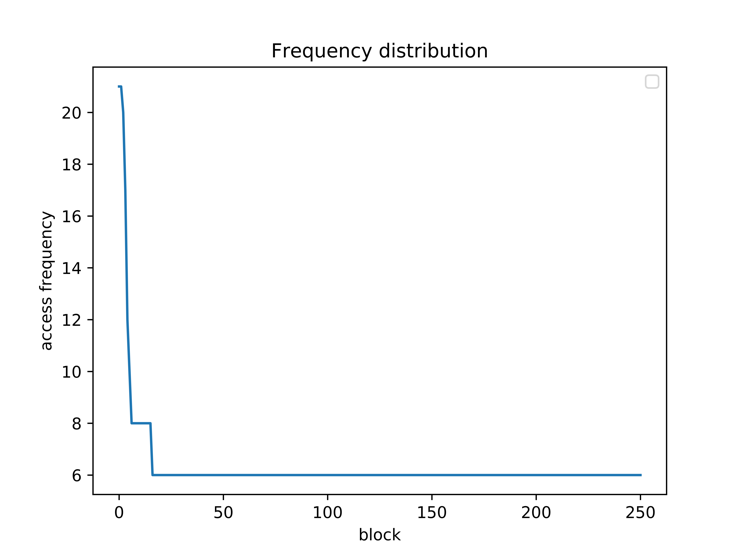
Focusing on the most frequent 250 blocks:



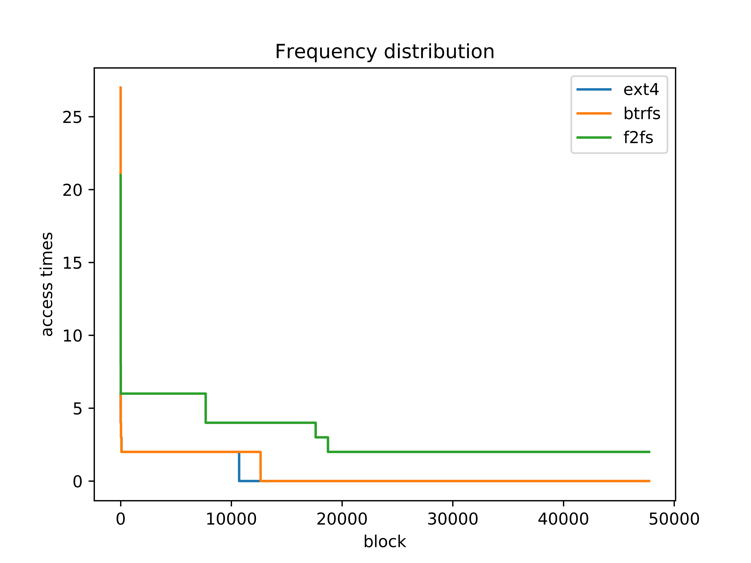
Frequency distribution for F2FS:



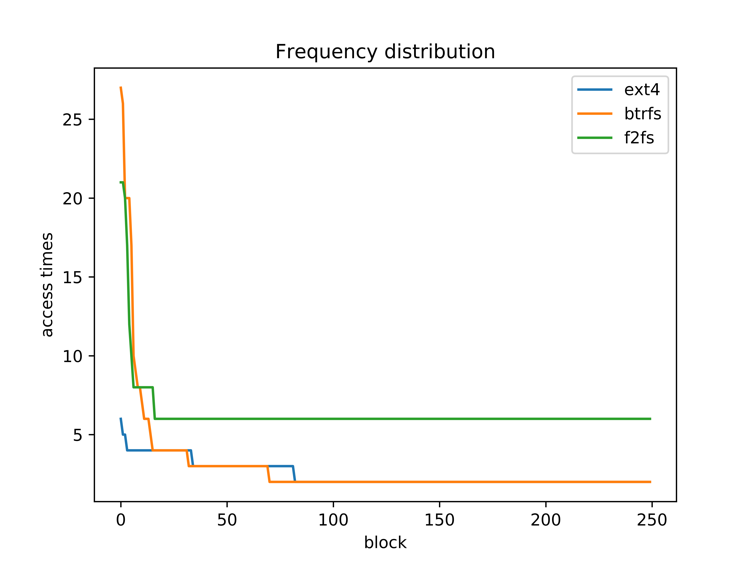
Focusing on the most frequent 250 blocks:



Overall Comparison:



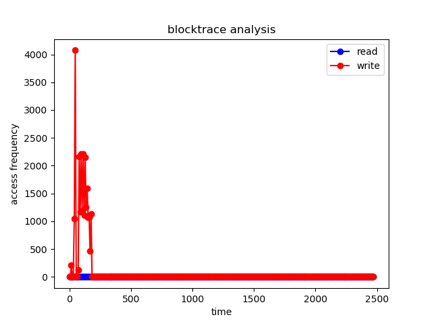
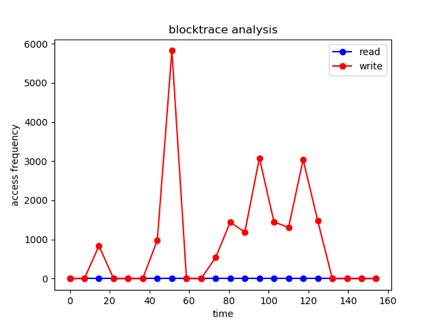
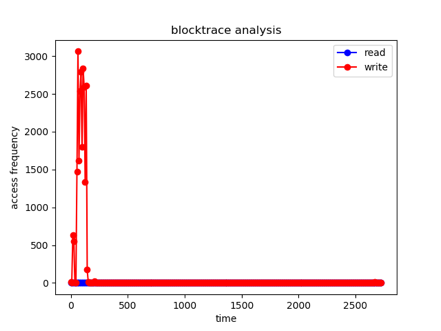
Focusing on the most frequent 250 blocks:



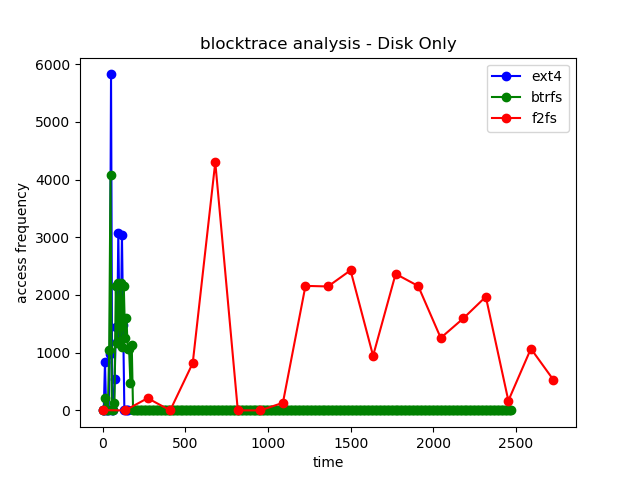
Analysis:

Text

R/W ratio for ext4: R/W ratio for Btrfs: R/W ratio for F2FS:

Overall Comparison:



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.

1. Related Files
   1. Workload

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

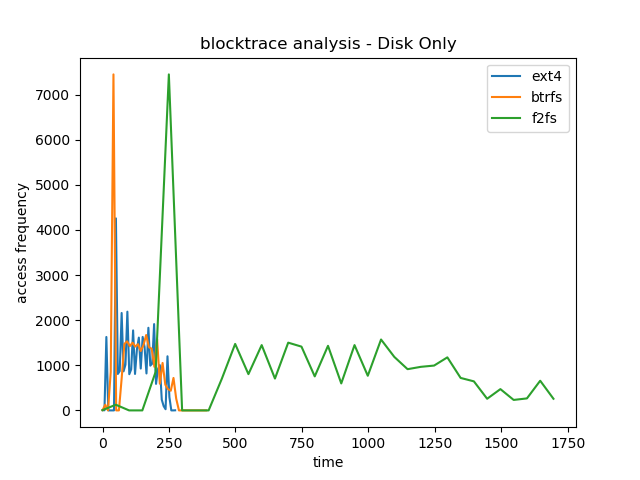
Stored under zhangf68@swift-014:~/spark$

* 1. Blktrace’s, Png’s, Csv’s

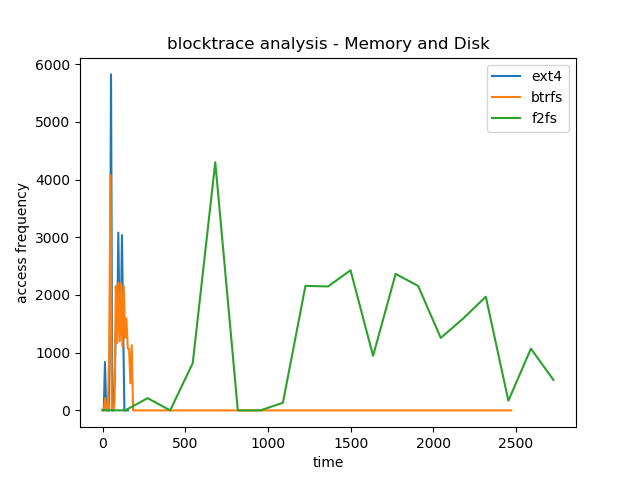
Stored under zhangf68@swift-014:~$

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

* Disk Only blocktrace of ext4, btrfs and f2fs:



* Memory and Disk blocktrace of ext4, btrfs and f2fs:



1. Final Conclusion:

We would like to choose ext4 for diskOnly access, while choose btrfs for memoryAndDisk access.