Matthew Koken

COEN 177L

Assignment 4

How fast is I/O when using the filesystem?

The speed of I/O depends on how the filesystem is being accessed. If the data is already stored in RAM, then it would be much faster to read/write. If the data has not been brought into RAM, it must be requested from the hardware – the HDD or possibly SSD. In both, the request will be queued before it is retrieved and returned. To the program, a read/write may be much slower or faster due to queuing, caching, and read/write requests. A write may appear fast, but it's possible that the request was only queued and not actually performed yet. Given these variables, is it possible to properly gauge the I/O speed of the filesystem? In order to gauge this properly, the data to be read/written cannot be cached, and we want to know the actual time for the data to be read/written. This excludes wait time, and we also do not want simply the return time that the write was requested to hardware.

**Writing**

In Linux, there are several tools available through the command line that will be able to gauge the data we need. In order to test the drive write speed, we can utilize the “dd” command. This will write a specified amount of data to the hard drive given a source, such as “/dev/null/”. However, this may only queue the data write. Another tool can be used in conjunction with this to get the results we want: “sync”. Sync flushes the buffer and commits all data to the disk, meaning we will not have the queue time, but the true write time. This will dump data from the buffer into a file for a true write speed test, without any read necessary. Altogether, the command used would look like:

$ sync; dd if=/dev/zero of=tempfile bs=1M count=1024; sync

Where the byte size, bs, is 1M, and we write count, or 1024, of those bytes. Usefully, the “dd” command prints some important performance metrics for this test. It validates the amount of data written, and includes the total amount of time spent writing the data, with average write speeds.

|  |  |  |
| --- | --- | --- |
| BS | Count | Time (s) |
| 1M | 32 | 0.0164485 |
| 1M | 64 | 0.851797 |
| 1M | 128 | 1.44376 |
| 1M | 256 | 3.72068 |
| 1M | 512 | 8.11785 |
| 1M | 1024 | 21.8966 |
| 1M | 2048 | 46.4164 |

Small amounts of data are able to be written quite quickly. However, as the amount of data to be written increases, the speed of the write decreases. It becomes harder to find enough data blocks to write the file, so a larger file may need to be broken up and cannot be written contiguously, like a smaller file. In this case, the data must be processed into blocks and written, all while keeping track of where the data is on the drive. Therefore, it makes sense that the larger files will take longer to write, especially as the drive reaches it's maximum data transfer rate. If the sync were not utilized, the data would be put into RAM and the data queued to be written. It would appear that GB/s of data were written, taking very little time. However, no data may have actually been written yet, as it could still be waiting in the queue to be written by the drive.

**Reading**

In order to properly test read speeds, we can transfer data from the written file to the buffer in the same manner as before. However, now that the data has been read, it is cached in RAM. It will need to be cleared from RAM so that it is only read from the drive. Using the Linux utility, we can drop the cached data:

$ sudo /sbin/sysctl -w vm.drop\_caches=3

Now that the data is cleared from RAM (and must be done for each subsequent test), we can test the read speed of the hard drive. Like before, we use “dd” to read the data into the buffer:

$ dd if=tempfile of=/dev/null bs=1M count=1024

|  |  |  |
| --- | --- | --- |
| BS | Count | Time (s) |
| 1M | 32 | 0.745276 |
| 1M | 64 | 1.31208 |
| 1M | 128 | 2.55209 |
| 1M | 256 | 5.22585 |
| 1M | 512 | 6.9469 |
| 1M | 1024 | 19.0644 |
| 1M | 2048 | 30.5209 |

Here, we see that the read speed is decently faster than the read speed. Since the data can be transferred directly to RAM from the disk without the need for additional processing, this makes sense. The data only needs to be read from the disk, with no processing to find where to put the next piece and log where the piece is, only the next address has to be read and moved to. Again, the speed slows as the amount of data increases, though much more slowly than the write speeds. Since there is little data to process, the maximum transfer rate of the drive can be better utilized. If the cache was not cleared before reading, then we would be seeing the RAM speed instead of the hard drive speed. In that case, the read speeds would appear to be in the GB/s range instead of the MB/s range that we see in the mechanical drive.