

A

Dan Ports <drkp@mit.edu>
6.033 handson #6 - 2004/04/22
Karger TR1

1.A. c-entry.mit.edu# ./bw -b /dev/hdc7 128
Timing device /dev/hdc7: 129 MB in 9.02 seconds = 14.30 MB/sec

1.B. c-entry.mit.edu# ./bw -l /dev/hdc7
Timing 10000 reads on /dev/hdc7: 10000 reads in 139.280000 secs =
13.928000 msec average latency

(/ (* 60 1000) (* 2 13.928)) = 2153.934

This suggests the disk is running at about 2154 RPM. This is sane; it's at least the right order of magnitude. And it's a slow disk.

← and it's an underestimate since it doesn't account for head seeking.

2.A. c-entry.mit.edu# ./smallfb 10000 1024
creating a temporary file used to flush the cache: done.
small file benchmark: 10000 1024-byte files
creat: 10000 1024-byte files in 669.071755 sec = 15.304786 KB/sec
read: 10000 1024-byte files in 73.892419 sec = 138.579845 KB/sec
delete: 10000 files in 0.974262 sec = 10264.179451 files/sec

2.B. c-entry.mit.edu# ./largefb 64
large file benchmark: 16384 4096-byte writes
writing 67108864 sequential bytes: 67108864 bytes in 9.562913
sec = 7017.617331 KB/sec
reading 67108864 sequential bytes: 67108864 bytes in 8.197006
sec = 8186.997057 KB/sec
writing 67108864 random bytes: 67108864 bytes in 301.692152 sec
= 222.441530 KB/sec
reading 67108864 random bytes: 67108864 bytes in 84.791916 sec =
791.453563 KB/sec

3.A. The ratio of the raw data rate to the small file transfer rate is:

(/ (* 1024 14.30) 138.579) = 105.66

So the actual data transfer rate is orders of magnitude smaller than the raw data rate. This reflects that accessing small files requires seeking to their location on the disk (including seeking to the inode in order to find the file's data blocks) before reading the small files, and they are not necessarily contiguous.

3.B. The limiting factor in the large file benchmark is the sustained data transfer rate, while the limiting factor in the small file benchmark is the seek time.

largefb random? → seek

3.C. We would expect a log-based file system to perform considerably better for small-file writes, since the writes could simply be appended (contiguously) to the log rather than seeking around the disk to the blocks where the file was originally written. Reads would still require seeking for the correct blocks in the log, and seek time is not the limiting factor for large-file operations, so the performance results would likely be similar.

good analysis.

\$Id: handson6.txt,v 1.1 2004/04/22 09:01:56 dan Exp \$