

Time to Setup Connection and Tear it Down

Design: I wrote a simple bash script that connected 50 times in a row and reported the sum back. To remove any execve time interfering, I commented out the code from the server for these runs.

Results: I recorded the data for 18 trials (each the average of the 50 connections) below

Trial #	Time (ms)	Trial #	Time (ms)
1	4.23	10	4.47
2	5.14	11	5.45
3	3.89	12	3.55
4	4.39	13	4.48
5	3.36	14	4.65
6	4.91	15	5.68
7	5.21	16	4.81
8	6.67	17	4.11
9	4.96	18	5.29

Mean: 4.73 ms

Standard Deviation: 0.799 ms

Analysis: The time to establish a connection with authentication and then tear it down is fairly fast. The variation is also rather small. I had to place a small (0.5 second) delay between connection attempts as otherwise the server would shoot up. I wonder how the time would vary if concurrent connections were used instead of consecutive.

Throughput of Server to Client

Design: I wrote a simple bash script that ran: `./dsh -c "cat file" -s localhost > file2`, where file was a 10 Mb file created with piped output from `/dev/random`. I uncommented the code for this test.

Results: I recorded the runs in the table below

Trial #	Time (seconds)
1	1.243
2	1.834
3	1.293
4	2.123
5	1.921
6	1.129
7	1.532
8	1.021
9	1.321
10	1.403

Average: 1.482 seconds

Std Dev: 0.364 seconds

Calculated Throughput: $(\text{file size})/(\text{Average} - \text{latency}) = (10\text{Mb})/(1.482 - 0.004) = 6.766 \text{ Mb/s}$

Analysis: The throughput using my program is 6.766 Mb/s, which is rather low considering it is a localhost to localhost file transfer. I wonder what was the confounding variable here.

Notes: For all tests, my system was at idle, my specs are a fast i7 CPU, and a solid state drive.