CS4513 Distributed System

**Project2 Experiment Report**

Submitted by:

Kewen Gu

Date:

Feb 2, 2016

**Design:**

For the first part of this experiment, we are asked to measure the setup, authentication, and tear down time of the program we wrote for project 2. In order to do this, I measured the time elapsed using the gettimeofday() call in my client’s code, and store the data into files. The program will create (if not exist) a file called setup.txt to store measured setup time, a file call authen.txt to store measured authentication time, and a file called teardown.txt to store measured tear down time. For separate experiment, these files should be removed before running the program, since they might contain the data from the previous experiment.

I also wrote a bash script that will ask AWS server to send files with sizes 10M, 20M, 30M, 40M, 50M, 60M, 70M, 80M, 90M and 100M separately to the client which runs on my own machine (MAC OS). Each file was sent 5 time in order to reduce errors caused by the fluctuation of the network conditions.

**Results:**

The measured results from the network performance part are shown in the table below:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| File size | 10M | 20M | 30M | 40M | 50M | 60M | 70M | 80M | 90M | 100M |
| Time in mSecs | 5439.954 | 3483.571 | 4323.636 | 4312.009 | 4872.253 | 16530.12 | 13358.477 | 6364.783 | 19873.753 | 47591.718 |
| 3295.069 | 3148.872 | 2875.638 | 4177.758 | 20349.668 | 12109.195 | 6992.089 | 7296.456 | 29575.955 | 8416.091 |
| 2419.065 | 6856.227 | 6101.609 | 3549.391 | 6955.619 | 11670.302 | 6790.752 | 20637.017 | 19264.57 | 20977.186 |
| 2926.887 | 3333.212 | 3969.75 | 4954.591 | 4963.181 | 5062.823 | 23886.566 | 7356.715 | 7536.031 | 16827.969 |
| 1695.206 | 4990.815 | 6641.44 | 14976.891 | 21376.44 | 6028.96 | 13043.552 | 15295.48 | 21081.55 | 12408.403 |
| Mean | 3155.236 | 4362.539 | 4782.415 | 6394.128 | 11703.432 | 10280.28 | 12814.287 | 11390.090 | 19466.372 | 21244.273 |
| SD | 1411.140 | 1574.666 | 1557.478 | 4823.799 | 8410.740 | 4733.740 | 6948.472 | 6305.501 | 7863.717 | 15462.841 |
| Throughput  (Mb/sec) | 3.170 | 4.585 | 6.274 | 6.256 | 4.272 | 5.837 | 5.463 | 7.034 | 4.623 | 4.707 |

Then plot the transfer file size vs. average time, shown in the figure below:

The above figure shows a linear behavior of the network performance. With the increase of the transfer size, the average time spent also increases.

For the CPU and I/O performance part, the measured results using sysbench is shown in the table below

|  |  |  |
| --- | --- | --- |
|  | AWS Server | Local Machine |
| CPU Total Time (sec) | 29.8922 | 22.9425 |
| I/O Performance (Mb/sec) | 32.925 | 52.283 |

**Analysis**

According to the equation:

**Local\_CPU + Local\_File\_I/O = n \* Network + Remote\_CPU + Remote\_File\_I/O**

And plug in the measured data, for example, when the server transfers a 10M file to the the client, the left side of the equation is 22.9425 + 52.283 = 75.2255, and the right side of the equation is 10 \* 1.411 + 29.8922 + 32.925 = 76.9272. The right side and the left side are roughly equal. We verified on the other set of data. It shows that the measured data matches with the equation above. It also shows from the equation that when the remote CPU and I/O performance remains unchanged, the maximum network performance is limited by the local CPU and local file I/O capabilities.