Module1 Report

Jianghong Wan PID: A59006198 GitHub: JWRickyWan server:Oregon client: Seoul

Zhishan Chen PID: A53297692 GitHub: CZhishan server:Oregon client: Ireland

Graphs

The numByte Size vs Effective Bandwidth bar graphs for serial/concurrent experiments are as follows:

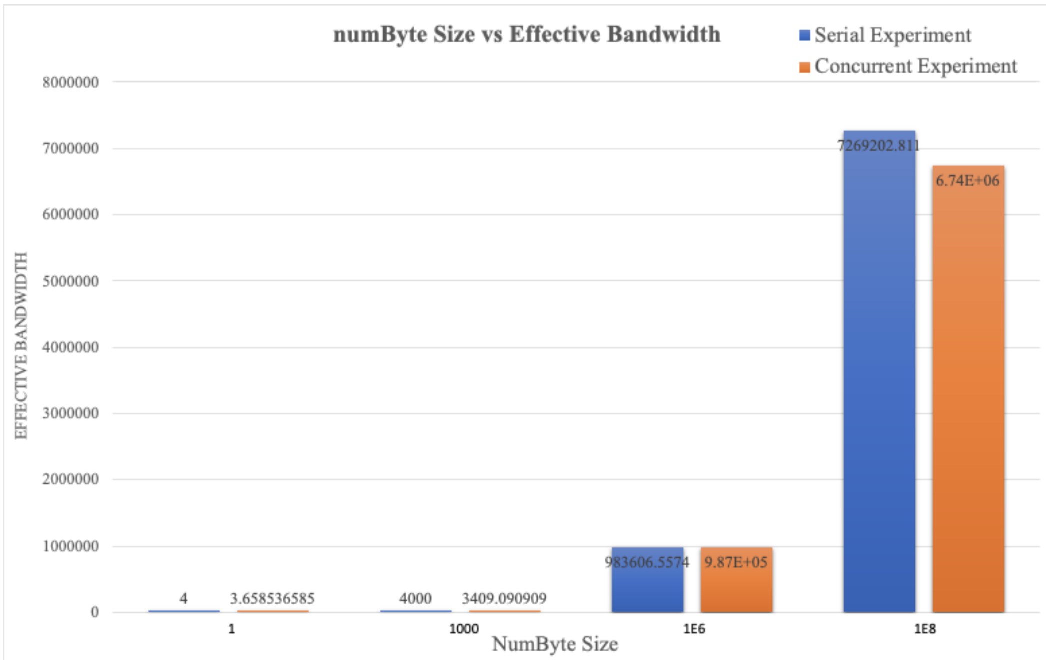

Fig 1. numByte(byte) vs effective bandwidth(byte/s), server:Oregon, client: Seoul

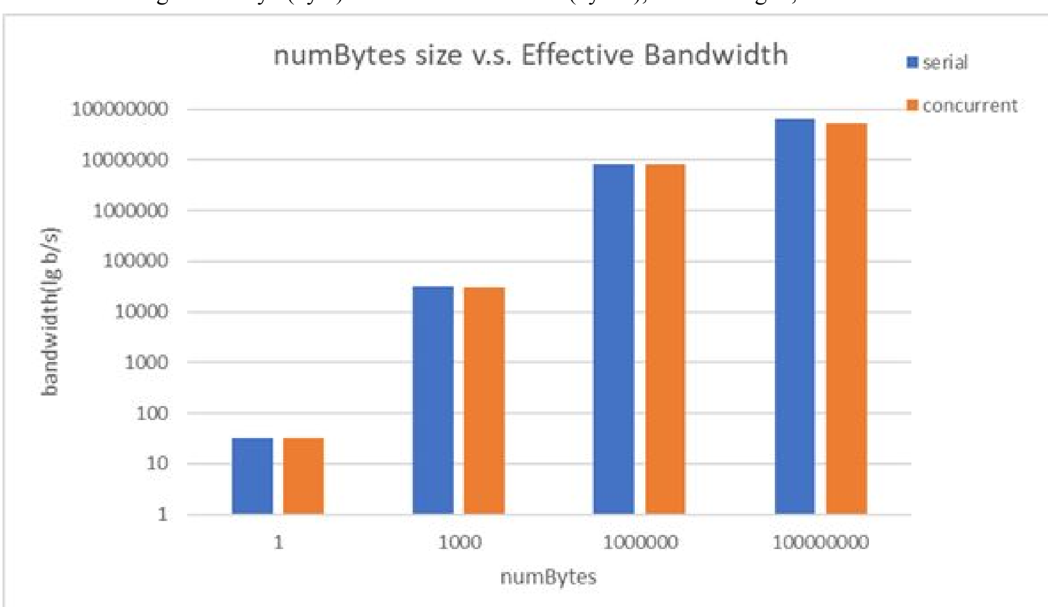

Fig 2. numByte(byte) vs effective bandwidth(byte/s), server:Oregon, client: Ireland

Q1. For the data from the serial experiment, how was the effective bandwidth affected by the size of numBytes ?

The relationship between the size of numBytes and the effective bandwidth is a positive correlation. Bigger size of numBytes means a larger effective bandwidth.

Q2. Using the data from only the *Serial* experiment, estimate the bandwidth between your client and your server. Now incorporate the data from your *Latency* experiment to increase the accuracy of your bandwidth estimate. Describe how data from the *Latency* experiment improves accuracy.

The more data sent, the more accurate the estimate will be. So we use the data from the serial experiment when numBytes=100000000 to estimate the bandwidth, which is 68.8Mbps. The round-trip time between our virtual machines is 0.123s, and the one-way latency is estimated as 61.5ms. After subtracting the one-way latency from the request time, the bandwidth is calculated as 69.1Mbps. The bandwidth we measured using iperf is around 68Mbps. The request time is the sum of latency and the actual transmission time. Since we got rid of the latency when calculating the bandwidth, the time period to transmit data of certain size is more precise.

Actually, we have tried other sizes of numbytes and the bandwidth is around 69~76Mbps.

Size in byte	Bandwidth in byte/s	Bandwidth in Mbp/s
3.00E+07	8.30E+06	66.4
5.00E+07	8.69E+06	69.5
8.00E+07	9.54E+06	76.3
1.00E+08	8.64E+06	69.1

Table 1. Additional measurements for estimating bandwidth. Server: Oregon. Client: Seoul

Q3. How did the data from the *Serial* experiment compare from the *Concurrent* experiment? Similar? Dissimilar? Explain these results as best you can.

The data from serial experiment and concurrent experiment is basically similar. From the bar graph, we can see that when numBytes equals 1, 1000 and 1000000, the request time is basically identical. And when numBytes equals 100000000, the request time from concurrent experiment is a little longer, which we think is because the bandwidth is divided for other concurrent requests.

Q4. After carrying out these experiments, what is something that you learned about performance and networked applications?

We' ve learned that the effective bandwidth depends on the data requested. And latency has a greater impact when transmitting a smaller amount of data. To have a better estimation of the bandwidth, one should increase the size of data to have a more precise estimation.

Q5. After carrying out these experiments, what is one (or more) unanswered question(s) you still have about network performance?

According to our results, the difference between serial and concurrent experiments is not obvious. They both vary positively with the size of data, and request time doesn't differ much from each other. This raises our question: under what conditions should one decide to call serially or concurrently, and what the pros and cons for those two styles are.

Additionally, we are still confused about the difference between one-way latency estimated from ping command and the request time of 1 byte data.

Q6. Any other comments/observations?

Generally, we conclude that the effective bandwidth has a positive correlation with the request data size. However, when we try other sizes of numBytes, like 80000000, the bandwidth calculated is 76Mbps, which is larger than that calculated from numBytes=100000000. So we wonder what other factors may affect the effective bandwidth.