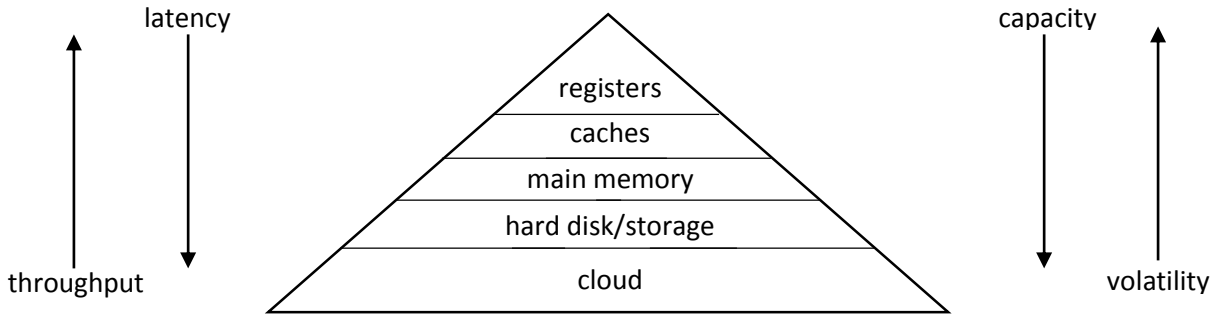
Assignment 4: Cloud Clocking

Bryan Mahoney and Tara Schneider
Team: 0000001
Due: 12/12/14
CE 290I
Dr. Christoph Kirsch

Abstract: We wrote a program to demonstrate the latency of the cloud in several different scenarios and to compare cloud storage services.

Background: This graphic from class shows the pros and cons of the five main types of memory. Arrows show the direction of increase:



Although many people choose to use the cloud for increased storage capacity and reduced volatility among other reasons, the disadvantages of the cloud—increased latency and decreased throughput—must also be considered. This experiment addresses the latency issue, not throughput, i.e. it sends many small data packets, not large data packets.

Architecture: Two separate laptops were used, one utilizing Windows 8.1 and one using Windows 7. In order to reduce the demand on the processors, the only processes running were basic Windows OS processes and those associated with running python code. All tests were performed on this architecture.

Procedure: The two laptops were connected to the internet through a single modem. Each laptop was loaded with a python program that would interact with files in a local folder that synced with a cloud storage service: DropBox, Box, Google Drive, and Microsoft OneDrive. The design and execution of the python programs is as follows, and the code is attached:

1. The Win8.1 laptop was loaded with python code called CE290I_starter. This code copies a small file called "startfile#.txt" inside of the synced folder and proceeds to search for a file called "cloudfile#.txt"
2. The Win7 laptop was loaded with python code called CE290I_checker. This code checks for the "startfile#.txt" file and, when found, creates a copy of a ~35mb file using the name "cloudfile#.txt"
3. When the Win8.1 laptop finds "cloudfile#.txt," it checks the filesize and the very last value in the file. Only when the filesize and last value in the file match the correct values does the CE290I_starter program repeat itself, creating a new "startfile#.txt" and repeating steps 1-3 for 20 iterations.
4. This process was repeated 10 times on each cloud service to obtain the mean and standard deviation of the data.
5. As a control, we performed the same file interaction and timing experiment on a file stored locally.

Statistical Method: We calculate the mean time it takes to run the program using various cloud storage servers and the mean time it takes to run the program using local hard disk storage. Standard deviation is also calculated. We then compare the time for each of the cloud storage services to the time for the hard disk. To prove that the cloud demonstrates more latency than the hard disk, we perform a one-sided t-test with the unequal-variance assumption as follows:

The null and alternative hypotheses, respectively, are:

$$H_0\colon\ \mu_C \leq \mu_{HD}$$
$$H_1\colon \mu_C > \mu_{HD}$$

where μ_C is the mean time it takes to run the program using cloud storage and μ_{HD} is the mean time it takes to run the program using hard disk storage.

A rejection of the null hypothesis proves that using cloud storage takes longer than using hard disk storage. We consider results to be statistically significant if the probability of obtaining the observed results if the null hypothesis were true is less than 5%, i.e. if the p-value is less than .05. P-values were calculated using python's scipy.stats.ttest_ind function (Attachment: StatsCalculator.py and TestData.csv), but because this function gives p-values for a two-tailed t-test, we halved the p-values to reflect a one-tailed t-test.

We also compare the mean program run times for various cloud storage servers to each other to draw preliminary conclusions about the speed of each.

Factors Affecting Data:

Due to the nature of modern operating systems, computer hardware and the internet, this test has a number of variables that can affect the data obtained. The times shown in Table 1 below are an aggregate of the amount of time associated with the Windows operating system, looping through Python code, communicating with a modem, communicating with the hard drive on each laptop, communicating with the RAM on each laptop, and communicating with the servers of each cloud storage provider.

In order to minimize the amount of error associated with the test, both laptops were only running the Windows OS, the processes associated with running python code, and the respective cloud storage service process (for syncing files). Additionally, between the tests, the python kernel was reset to avoid any stored information about file names, etc. While the tests were running, no movements or input was provided to the laptops.

Results and Conclusion:

	DropBox	Box	Google Drive	Microsoft OneDrive	Local
Average	455	2750	1515	2346	244
Standard Deviation	246	431	115	274	13
p-Value	0.0119	9.25E-09	2.06E-11	7.80E-10	

Table 1: Results of running program using various memory storage locations. Average and standard deviation are times in seconds. The p-value listed for each cloud storage service is the probability that the null hypothesis, $\mu_C \leq \mu_{HD}$ is true.

As shown in Table 1, the p-values for all of the cloud storage servers are much less than 5%, therefore we have proven that interacting with a file on the cloud takes longer than interacting with a file on one's local hard disk. In addition, we can see that during this experiment the speed of each storage type can be ranked as follows (from fastest to slowest): local hard drive, DropBox, GoogleDrive, Microsoft One-Drive, and Box. However, one iteration of the experiment is not enough to prove that any cloud storage service is always going to be faster than another. The latency of a cloud storage service can change with the amount of traffic both on the cloud server and on all the fiber optic cables and wi-fi connection between a user's computer and the cloud server, etc. In general, one can see that there is not only a difference in latency between hard disk and cloud but also among cloud servers.