Directory Contents

Introduction

The top level directory chunk contains tests performed with the archiving system, while the directory normal contains tests that uploading the files one-to-one to the provider. The tests are again separated into directories with names indicating the specific implementation. These folder contain at least two images:

1. **upload_rates.pdf**: Displays the upload rate of CloudFusion with a resolution of about one minute. The upload rates are the raw upload of the specific data by CloudFusion without overhead, measured through the statistic file interface, comparing the amount of uploaded data each minute.
2. **internal_upload_rates.pdf**: Displays the average internal upload rate from the start of the CloudFusion instance. More precisely, the internal upload rate is the upload rate measured while at least one upload worker process is active.

Each image is a graph with lines through the measured points. Each line is described by a tuple with the name of the folder containing the measurements, and the size in KB of the files that were uploaded during this test.

chunk/slowdown_laptop:

Upload rate writing 1 MB files to Tonline webdav account. At the beginning there is a peak, then the upload rate detoriorates quickly. The initial hypothesis to explain this behaviour is as follows. The cache expiration date is set to four minutes, so during this time no files are uploaded. But during this time, files are written to the file system. So when the synchronization thread starts, many files need to be uploaded. But the synchronization thread quickly catches up, so that it needs to wait until new files written to the file system are expired. To verify the hypothesis, the tests were set up on a more performant computer, and another provider was used in case there is a provider specific explanation for the decreased throughput or an issue in the provider specific implementation.

normal/sugarsync:

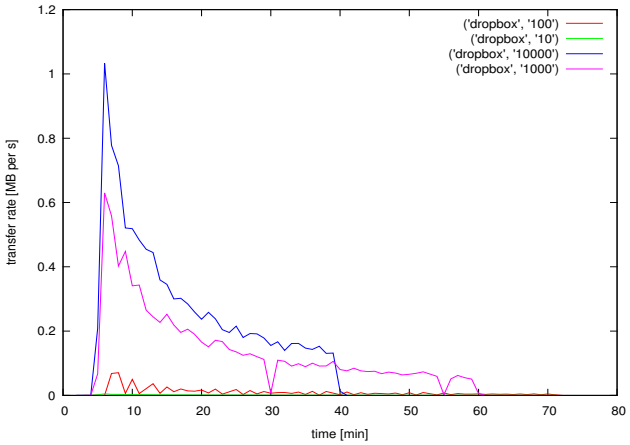
On the more powerful computer, testing the sugarsync implementation by writing 10 KB files shows a similar upload rate trend as the test with webdav on the laptop. Though we suspect that this behaviour might be influenced by the specific implementation. During the test, several gigabytes of data were downloaded. For reference, the file ifstat_out shows the up and download rates per second. The cause of this misbehaviour still needs to be researched. Still, the similar upload rate trend is a first indication that the hardware is not the reason for this behaviour.

normal/local1:

To exclude the possibility that the slowdown can be reproduced and that it is not due to network issues, the same tests are performed with a local webdav server at the laptop.

normal/dropbox1 and normal/dropbox2

Again on the more powerful computer, the results are cross-checked with yet another implementation and also with different file sizes (10, 100, 1000, 10000). By using larger files, we try to increase the speed of writing data to the file system relativ to the upload speed, so that more files are available for upload and the synchronization thread cannot catch up with the writer thread. Again, the results show a similar deterioration in upload rate, though the results show that by using a higher file size, the upload rate can be increased. Nevertheless, the amount of files that are not already uploaded was observerd during the tests through the statistics interface. More precisely through the virtual file **stats/notuploaded**. The amount of files that were being uploaded increased as the tests went on, which disproves the initial hypothesis that the synchronization thread catches up with the thread writing to the local file system.



normal/webdav2

To show the difference of using bigger files with another provider, 1 MB and 10 MB files are uploaded to T-Online on the fast computer. The upload rate deterioration is less acute when using larger files, but still visible. The next hypothesis is that the decay in upload performance is because of the synchronization thread cycles being too slow. Not to hinder the writer thread, the synchronization thread waits a certain amount of time between each cycle before starting new upload processes.

normal/local2:

By decreasing the amount of time the synchronization thread waits between cycles, and trying to adapt it, so that it stays the same even though the cycle itself takes longer to complete, the upload rate increases significantly. But still, the upload rate curve shows a degradation over time, disproving the last hypothesis. Also, by profiling and logging, no clear candidate for the deterioration could be determined, except that the synchronization overhead for the local cache seemed to be very high. The next hypothesis is that the local cache slows down synchronization. So we tried to check if the metadata cache is the problem. Thus we used the local test setup and instead of deleting the file at the end, we uploaded yet more files, but to another directory. At the start, the upload rate was quite high again, though not as high as the first peak. Next, to clear the meta data cache we could restart CloudFusion to again write files to it.

normal/small_cache

In the meantime, another local test on the fast computer is performed, trying to decrease the amount of files in the data cache by reducing the initial waiting time before files are uploaded form four to one minute. This waiting time is determined by the cache expiration time. Also, the cache size is reduced from 200 MB to 1 MB, though this is only the soft cache limit which means that more data can still be written to the cache, but as soon as it is synchronized to the storage provider it will be deleted (after waiting for the complete cache expiration time again, to account for eventual consistency). The writer thread is stopped after 30 minutes to see the effect of a decreasing cache size, though unexpectedly, the upload rate does not increase while the cache size decreases.