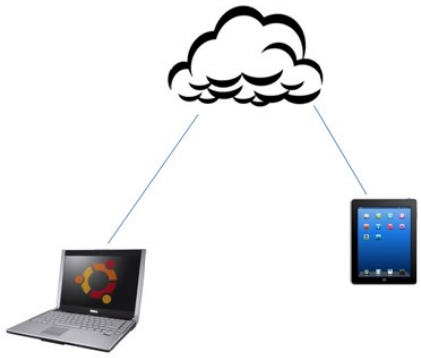
Cloud storage optimization – supervisor Keith Briggs



Cloud storage is already a huge business, and will certainly grow further. In the figure above, we imagine that a file is generated on the laptop. The cloud is supposed to keep a copy of this file, and automatically track any changes. Then when we view the file on the tablet, it is always synchronized to the laptop version. In reality, there are three copies of the file; but, to the user, it appears that there is only one.

This raises some interesting questions. How does the cloud know that its copy is out of sync with the laptop or the tablet? The answer is by using hash functions. These are short bit-strings (typically 256 or 512 bits) which have the property that if the hash values of two files are different, then the files are certainly different; where if the hash values are the same, then the files are the same with high probability (the low probability complementary event is called a *hash collision*). So it can be assumed that there is a method available to determine whether to files on remote devices differ, which uses little network traffic. The next problem is to update the cloud copy when the laptop copy changes, preferably without copying the entire file. This problem was solved by Andrew Tridgell in this PhD thesis (http://www.samba.org/~tridge/) – it is the rsync algorithm. This works (roughly), by breaking the file into blocks, doing a hash on each block, and only sending the blocks which have changed.

So where is the mathematics in this? One problem is that large files are often stored in compressed form (e.g. zip or gzip). This has the effect that when just one bit is changed in the uncompressed file, to an algorithm like rsync working on the compressed file, it appears that the whole file has changed. Furthermore, in file types like pdf, the internal format is compressed, so the same problem appears. There is thus a trade-off when using cloud storage: to compress or not to compress? Also, some parameters in the rsync algorithm are tunable, like the block size, and the hash string length. So the task of this project is to find optimum parameters settings, so that (when applied to typical file-size distributions), a cloud storage system can maintain file synchronization with minimal network traffic. Some theoretical analysis may be possible, but it is likely that the final solution will come from a numerical optimization algorithm.