Volume 5, No. 11, November 2014

Journal of Global Research in Computer Science

ISSN-2229-371X

REVIEW ARTICLE

Available Online at www.jgrcs.info

TECHNIQUES FOR FILE SYNCHRONISATION: A SURVEY

Gyanaranjan Shial[1] and Santosh Kumar Majhi[2]

[1]VSS University of Technology, Burla,INDIA-768018

[2]Indian Institute of Technology, Bhubaneswar, INDIA-751013

Abstract:We are inhabiting in the information age.Large amount of data of the order of Terabytes, Exa- bytes and Zettabytes are of contents are transported from one gimmick to another.The bad news is that the network bandwidth provided by internet providers is determined.It will take long time to remove all files among different computer organization. The immediate solution is to synchronize the files in a better way in the clusters of a computer system which will consume less bandwidth and will give faster file synchronization operation. The main problem of file synchronization is to transfer media files in a bandwidth efficient manner. Apart from this file need to be transferred in small time, so that the client should not wait for file transfer to happen. There is a very wide research on this topic in Rsync [1][7][3]. For this we need to develop some synchronization technique between institutional hubs, so that files can be transferred from any remote location to the institutional hubs in less time and in a bandwidth efficient manner. And simultaneously the files need to be transmitted to the student tablets with all the required privilege. After all, the aim of the project is to transfer the updated content to the student tablets and corresponding institutional hubs and deliver content like quiz materials, video lectures, class notes, application software along with collecting student response with less effort and in less time.

1. INTRODUCTION

A common method is to transmit just the differences between A and B down the nexus and then use this list of differences to re-construct the file. The problem of file synchronization is easier if all the updates are logged in the log file that can be transmitted to the remote device. But the problem here is that not all devices can maintain log file for all files. And it will become critical to maintain separate log file if the number of files is very large. To determine which file is altered in a particular time one can hold the change time of the file. Based on the modification the recent modified file is taken as the latest file to be transferred to the remote location. But the problem is not exactly the same, we have to consider the modified content in a particular document, i.e. if it is a content like a Google spreadsheet where the updated content should be transferred rather than the whole file. Some other techniques are proposed in sync are as follows.

2. VARIOUS SYNCHRONIZATION TECHNIQUES

2.1 Checksum matches

In this technique the difference between the two files are calculated and delta is transferred to the other devices. In Rsync two checksums are required to compare the files between two remote devices. The first ace is a week checksum and the second single is strong checksum.The week checksum is a rolling checksum[1] which is quicker to compute from the previously received checksum. The strong checksum[2] is an MD4 checksum and it will cost more than calculating a weak checksum. Therefore the protocol here first checks the matching in the week checksum and when it finds any matching, then it will use the strong checksum for verification.When a match is found, the source machine will transmit the file offset, previous match and watched by the index of block of target machine that checked. If no match is found in particular offset, then the offset is updated to the next position of the file and search proceeds. If a match is found, then search proceeds further at the end of the match block. This technique saves considerable amount of computation for matching file when the two files are almost indistinguishable.The above procedure will require more time for checking if there are a number of files.The above problem can be resolved by using pipe-lining the processes for considerable latency advantages.

2.2 Improve Method over Rsync: In Rsync[6] the client sends hashes to the server, wherein file synchronization framework the hashes are transmitted from server to client. File synchronization framework tries to ascertain out the difference of files using Map Construction [6] and delta Compression [6] techniques.

2.3 Map Construction

Here according to map construction technique a map is maintained in server side. Suppose the server has a file $f_{new}$and the client has filed F_{old}, then the server will send the hash to the client and the client will compare the hashes with its own map file. For example, server S has a file which contains abcxyzbbb and the client C contains a file whose contents are pppxyzaaa, then C can create a map of the minethat looks as follows: ??? xyz??? and transmit it to S.

[1]rolling checksum

[1]strong checksum

Then C now can send 010 to server, based on that server will send the missing part to C. Here 1 means block exits and 0 means block not exist.

2.4 DELTA COMPRESSION

Then the server will create a F_{ref}, consisting of all part of F_{new}that are unknown to the client and based on that it will create a $F_{target}$and tries to compress the file by referring to F_{ref}, then send it to the client. The client uses its map to recreate the file and merge it to create the new file F_{new}.

3. PROPERTIES OF DEVICE SYNCHRONIZATION:

3.1 Scalability:

All the cloud based solutions are more scalable than any other way of file synchronization. Slow-sync is less scalable than other methods of file synchronization. Scalability is also a major advantage of cloud storage, because it looks to the users that storage capacity is unbounded.

3.2 Cost-effectiveness:

HadoopRsync[2] will cost more in the initial level as expected. But it is more effective afterwards. Cloud storage needs low maintenance cost, so more cost-effective than others.

3.3 Data Transmission Load:

It is the quantity of data transferred between any two devices.It is an important metric because it directly touches on the time required for data transmitting. This metric will affect directly to the willingness for using the particular synchronization application.

3.4 Computation:

In slow-sync no more calculation is required during file synchronization.It is less significant if the files are synced between different institutional hubs in our example. And is more important when the file is synchronized between student tablets and institutional hubs because the processing power in student tablet (Aakash tablet) is comparatively lower than a dual-core or core-to-duo processor.

3.5 Network Size:

Among all the techniques of file synchronizationHadoopRsync[2] is more scalable in network size than other methods of file synchronization. Slow sync network is least scalable in network size.

4. FILE SYNCHRONIZATION PROTOCOLS

Content synchronization protocol is used for identifying changes, resolve possible conflicts, and propagate updates to the various synchronizing devices. Moreover, Aakash tablets have limited CPU, storage and power resources and, therefore, are unable to quickly process or transmit large amount of information. In a networked setting, scalable and efficient data synchronization protocols are essential for data intensive applications such as e-mail editors, spreadsheets, and even collaborative work environments. Synchronizing content with just one desktop machine to be easy, but here we give more importance to synchronize contents over multiple devices i.e. multi-device synchronization. The motive of designing protocol is for addressing the scalability issues inherent to synchronizing data over large, heterogeneous, tether less networks.

4.1 HotSync

SlowSync[4]: Taking into consideration the file synchronisation between two devices, the file synchronisation technique suggest when there is any change in a file, the corresponding old file will be replaced by the new file in the other devices. This method of file synchronization will consume more bandwidth as well as time. **Slow sync** will give more latency time and more bandwidth usage when the size of content is very high, It may stop synchronizing in some point of time due to network bottleneck. **Fast Sync[4]:** In this technique the difference between the two files will be transmitted if the files are similar in some part. This is useful when more user is editing a single file and the change in the all the documents are transmitted to the central device here it is an institutional hub. And by this technique the bandwidth usage in comparatively lower than slow sync. Many more synchronization applications use this technique for file synchronization with some extra modification to it. This needs a previous synchronization with same device means synchronizing contents with last synchronization. First sync doesn’t give efficient solution when a number of devices want to change the content to the same database, for this we need some efficient protocol for data synchronizationbetween devices.

4.2 Intellisync

It maintains a client server architecture where all devices don't use peer to peer connection, rather devices who had an update information need to synchronize with the Intellisync[4] anywhere server in the network. And the Intellisync server will synchronize with the Microsoft exchange server, not periodically, but within a certain time duration. This synchronization between Intellisync and Microsoft It maintains a client server architecture where all devices don't use peer to peer connection, rather devices who had an update information need to synchronize with the Intellisync Anywhere server in the network. And the Intellisync server will synchronize with the Microsoft exchange server, not periodically, but within a certain time duration. This synchronization between Intellisync and Microsoft

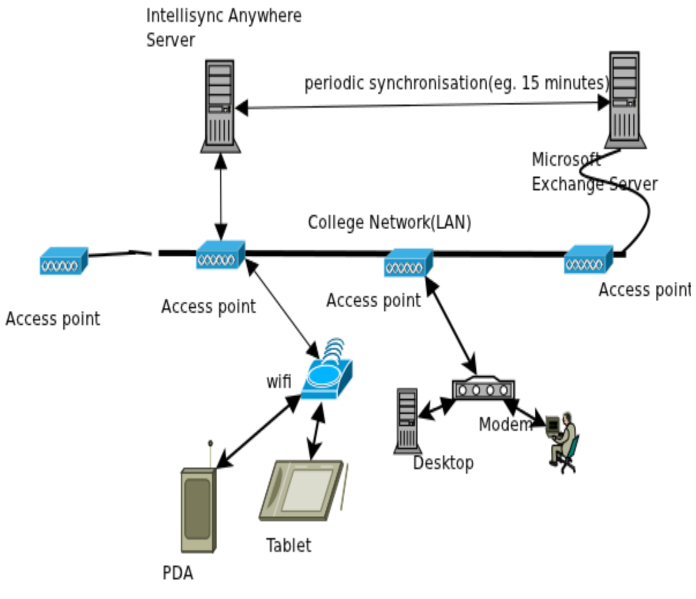

Figure 1: Shows Intellisync architecture

4.3 SyncML

SyncML[4] is open industry standard for file synchronization. In this type of file synchronization protocol every device will maintain some status flag for each record.During modification of the file will toggle the status flag with regard to every other device in the mesh.It's different from fastsync[3][4] of Hot-sync[4][4] protocol by keeping a set of status flags across a lot of gimmicks. Modifying a set of file in a device will toggle not only the status flag of that device, but also the status flags of other devices. When two device finishes synchronization then the status flag of these two devices will be cleared. Maintaining a set of status flags, which is a substantial amount of memory for a portable device. The quantity of storage needed by a network of and devices, each with or recorded would be roughly n*r units per device.

5. A mean to faster file synchronization

Synchronizing contents over two devices is extremely important because its efficiency has a multiplicative effect on the overall efficiency of synchronizing the network. By scaling down the file size, the amount of petitions can be slimmed down significantly and thereby speed up the upload (and download) of a certain file set. Though the size of the server is limited, so by introducing a concept of reduction on the number of contents in the remote device can also solve the problem by eliminating a number of requests to the server. De-duplication is a valid technologyfor end-user applications if controlled properly.

5.1 Chunking [5]:

In this technique file is divided into parts, if similar part is found, then the splitting is stopped there. Otherwise recursive splitting is carried on that unmatched blocks by which communication cost will be minimized. Selecting the precise size of the block when to stop splitting is a related proposal over rsync.

5.2De-duplication:

By using fast sync the difference between the files will be sent to the server or any device. De-duplication[5]works withmultichunking[6] method. In some cases the chunks of the files may be alike because it’s of similar file format or similar operating systems utilized to create the files. In that case, before uploading the bunch of chunks one needs to reduce the size. So one technique is reducing the number of similar blocks to a single block and let the other refers to this block, it can reduce the size before uploading the files. The similarity between the file chunks depends on the size of the chunks. The less the size of the chunks, the more the similar chunks we have. This is used by Syncany. So experiment can be carried on for determining the size of chunk size for getting maximum similarity between the files.Synchronizing large replicated collections of file over slow network can be created by some more techniques that are described in Improved filesynchronization[6] techniques.

[3]fastsync
[4]hot-sync
[5]De-duplication
[6]multichunking

Future Work:

De-duplication and reconstruction of filesare found to be the best fit for file synchronization and choosing a best chunk size of a particular file type is a challenging task over a variety of file system.Secondly reconstructing the same file over a limited number of chunks with maximum number of similarity is also a challenging task. These two problems can be taken forward to future work.

CONCLUSION:

Across all the file synchronization methods, checksum method works better for files where maximum similarity is found. De-duplication works better for similar types of files, choosing a best chunk size may produce the maximum number of chunks. So the probability of getting maximum number of similar chunks depends upon the chunk size. So finding best chunk size is itself a problem and it variesaccording to the type of files used for synchronization. One more problem is to reconstruct the same file in the destination device using the same chunk. Chunking method works well over all file synchronisation techinique. Apart from other file synchronizationmethods, Intellisync needs to setup a client server architecture for the same. Whereas SyncML needs extra communication for handling the device flags over a network.

REFERENCE:

[1] Alternative to file sharing, http://alternativeto.net/tag/file-sharing/?platform=android-tablet

[2]J. Zhang, X. Yu, Y. Li, and L. Lin. Hadooprsync. In Cloud and Service Computing (CSC),2011 International Conference on, pages 166–173. IEEE, 2011.

[3] Rsync, 2014,19th Nov, http://rsync.samba.org

[4] S. Agarwal, D. Starobinski, and A. Trachtenberg.On the scalability of data synchronization protocols for pdas and mobile devices. Network, IEEE, 16(4):22–28, 2002.

[5] Philipp C. Heckel. Minimizing remote storage usage and synchronization time using deduplication and multichunking: Syncany as an example. Technical Report TR-CS-96-05, universitat mannheim, School of Business Informatics and Mathematics Laboratory for Dependable Distributed Systems University of Mannheim, 2012. http://www.syncany.org/.

[6] T. Suel, P. Noel, and D. Trendafilov. Improved file synchronization techniques for maintaining large replicated

collections over slow networks. In Data Engineering, 2004. Proceedings. 20Th International Conference on, pages 153–164. IEEE, 2004.

[7] A. Tridgell and P. Mackerras. The rsync algorithm. Technical Report TR-CS-96-05, Australian National University, Department of Computer Science, June 1996. http://rsync.samba.org.