Reducing Costs between Sequential Backups

Alexandru FILOTE
IT&C Security Master
Bucharest University of Economic Studies
ROMANIA
alexandru.filote@gmail.com

Abstract: This paper addresses the problem of costs between two sequential Backups, identifies the main reasons that create costs and the facts that influence these costs. For the problem used in this scenario are used two sequential backups Ti and Ti+1, where Ti and Ti+1 are created with a standard method of "Incremental Backup". In the second part, there are proposed solutions to reduce the costs, it is presented the impact of reduction of the costs and it is proposed a formula based on system configuration and the size of the file.

Key-Words: sequential, backup, reducing, cost, incremental, cloud

1. Introduction

In recent years, in Information Technology industry, backups have become an important part of every business, especially fields that are much regulated in maintaining the electronic records. Even high reliable computers might sometimes break down due to a large set of actions that can be performed over these or due to human errors and hardware faults, the backup technology have become quite popular. Nowadays, the backup technology is a very vast field and there are many types of backup, the most important are: full backup, incremental backup, differential backup and mirror backup. From the point of view of cost between two sequential backups, the most optimal technique is represented by incremental backup.

2. Problem Formulation

In recent years, the backup service has begun to be offered by many vendors as a Software as a Service product. Online backup systems are typically built around a client software program that runs on a schedule, typically once a day, and usually at night while computers aren't in use. This program typically collects, compresses, encrypts, and transfers the data to the remote backup service provider's servers or off-site hardware. There are many products on the market and they have gained popularity also over the mobile systems and they all have different features like: type of encryption, online access to the files, differential data compression, bandwidth usage, file-by-file restore or continuous backup. An independent study performed last year reveals that 31% of PC users have lost all of their files due to events beyond their control, 60% of the companies that have lost their data have shut down within 6 months since the disaster, 93% of companies that lost their data center for at least 10 days, have been filed for bankruptcy within one year, 140.000 hard drives crash every week only in US and one simple drive recovery can cost upwards $7500 and the success is not guaranteed[1]. The five most common reasons for backups failing or being unusable in the event of disasters are: a staff member leaves and no one continues the backup role, backup media is now swapped and then overwritten the next night, changes are made to the server or application and the backup is not adjusted to factor in this change, media is not taken offsite and the server is stolen, lost or damaged in a disaster and no one is checking the reports or performing test restores. The recommendations for nowadays to ensure that a backup is adequate are: media is swapped regularly, backups are stored offsite and securely, the backups are password protected, media should be clearly labeled and a record kept, ensure that someone is in charge and has been trained to understand and read the backup reports,

ISSN 2067 – 4074

run test restores on a regular basis and review the new programs and data, so that the backup includes these. Having this in mind, we can consider that Cloud Backups are the closest to what a backup Service should have. Anyway, when a service like this is procured, there must be taken in consideration the following facts: the predicted capacity of the backup, the bandwidth of the internet connection and the number of users for which the backup service will be used. In the Table 1 are shown the actual numbers used when is taken in consideration Cloud storage against Traditional Backup. [2]

Table 1: Online vs Traditional backup

Backup factor	Cloud storage	Traditional backup
Amount of data	Best when the total amount to protect is less than 100 GB per 1 Mb of network bandwidth. For example, 100 GB can be supported by a 1 Mb WAN connection (such as DSL)	For large amounts of data, or for environments with limited network connectivity, traditional backup techniques are more appropriate.
Rate of change	Best when the rate of change is less than 10% of the total data per month.	For data that changes frequently, traditional backup methods that use local disk and tape, with tape transport off-site are more appropriate

The above numbers are generated taking in consideration the most cost-efficient technique of data recovery: incremental backup.

2.1 Incremental Backup

The technique of Incremental Backup came later as a necessity to overcome the deficiencies that result from a traditional backup. Anyway, this new technique involves a more complex process in the creation of the backup. To understand this we will present step by step the technique of incremental backup and risks and costs that imply each of them.

One of the most important problems when using an incremental backup is that the files are backed-up in a sequential order, otherwise a file backed-up at a sequential time t+1 is inserted in the sequential backup t, and the file might corrupt the whole backup. This issue can happen especially if only one incremental backup is used over multiple synchronized systems or if there are multiple users that share one system or even if a user tries manually to create another backup, before the system has finished saving the previous backup.

Since the files of the backup should only be accessible to the user(s) that are associated with the systems for which the backup is provided, the applications can use either a private key that is stored on the local machine, or a username and a password associated with the backup. The first method, even if it's easier to be implemented, it can be very hard to store it on the local machine and can be very easily corrupted. The second solution can also cause a lot of issues especially when the user wants to change or restore the password or during the downtime of the internet connection.

Especially due to human errors some versions of the backup might not have the main purpose of the backup due to accidentally savings or the first backup after a recovery.

Every time a new backup is created the main file of the backup must be update with new content. Updating it at the beginning or at the end of the backup process can cause a lot of trouble if the backup fails during the process or if there are files that are changed during the backup period and a real issue is caused by temporary files and system files. Also, if the file is updated after each file is checked, the performance of the both systems (the one that is backed-up and the one that is used for backing up) will decrease and since the time between two sequential requests of updating is very

short, network issues can be caused due to trying to access the same files.
Between two subsequent times of the backup Tn and Tn+1, even if there are a few changes that have been performed by the users that access the system, the backup application might detect that there are a lot of files that have been modified. This situation is mostly determined by the Operating System files that are under continuous change. Another big problem is to detect which files are new, which files are a duplicate of another file in the system and if a specific file has been deleted or just has been moved to another location.
In order to save the space and to transfer the files easier over the network, the files should be archived to reduce the size. Beside the fact that specific types of archiving are different for each type of file (see Figure 2 below) the application used for backup must choose the most proper behavior: time consuming or space consuming. Another problem caused by the archiving is when the verification of the new files that should be updated, the files that are already in the backup, there is a delay caused by slower access to files archived.

Table 2: Analysis of compression method

Type of file	Best (compression)	Best (fastest)	Best (Overall)
Text	7-zip	lha	Tar+bzip2
Binary	7-zip	lha	Tar+bzip2
Image (png)	pngcrush	Tar+gzip	zip
C++ sources	7-zip	Tar+gzip	Tar+gzip

There should also be taken into consideration all other processes that are running in the host system, that can cause longer times needed to create the archived file.
Although the archiving process is optional in the process of creating a backup, the encryption is mandatory. In the Figure 3 is presented a comparison between most popular types of encryption nowadays.

Table 3: Encryption/time

Input(Kb)	AES	3DES	DES	Blowfish
49	56	54	29	36
59	38	48	33	36
100	90	81	49	37
247	112	111	47	45
321	164	167	82	45
694	210	226	144	46
899	258	299	240	64
963	208	283	250	66
5345.28	1237	1466	1296	122
7.310.336	1366	1786	1695	107
Average Time	374	452	389	60.3
Through put (MB/sec)	4,174	3,45	4,01	25,892

Based on first 8th values recorded in Table 3, thru the regression method we can generate the following equations:

$$AES(x) = -211.486968607 + 64.169486712 * \ln(x)$$

And with the following graph:

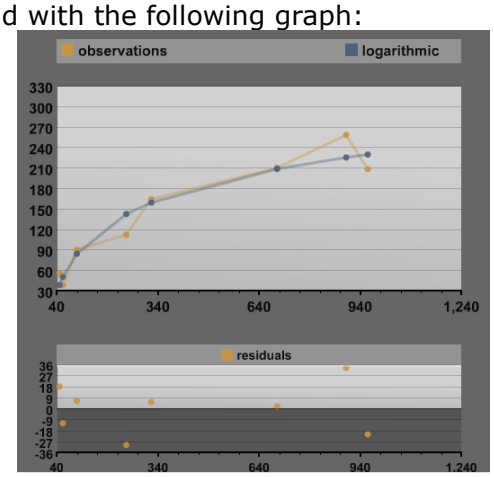

Figure 1. AES performance representation

$$3DES(x) = e^{(1.57127799656 + 0.596298097545 * \ln(x))}$$

And with the following graph:

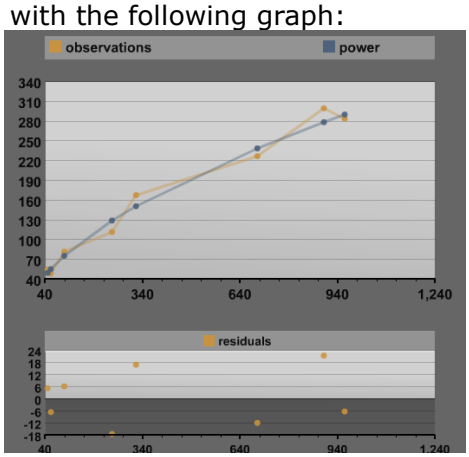

Figure 2. 3DES performance representation

$$DES(x) = (5.05854681036 + 0.011067141175 * x)^2$$

ISSN 2067 – 4074

And with the following graph:

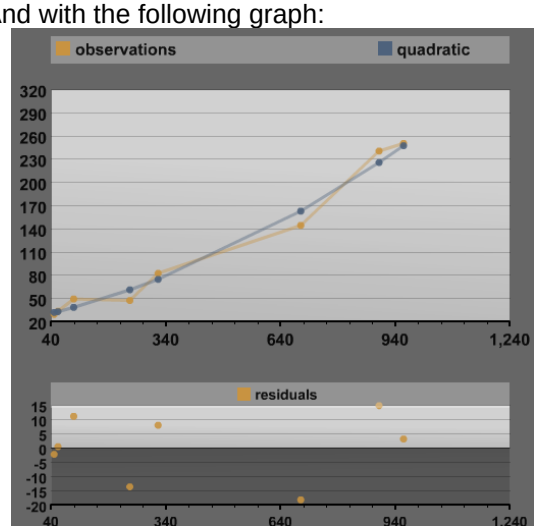

Figure 3. 3DES performance representation

$$Blowfish(x) = e^{(2.81261675475 + 0.183065093417 * \ln(x))}$$

with the following graph:

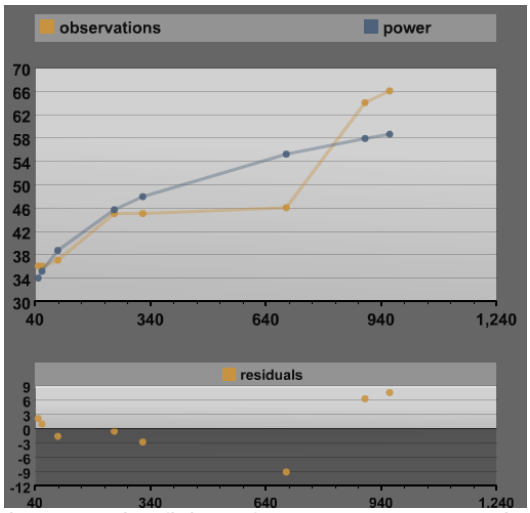

Figure 4. Blowfish performance representation

Even if Blowfish seems to be by far the fastest encryption method from the above, we must take into consideration the problem that appears when the leys are changed since each new key requires pre-processing. Also Blowfish is known to be susceptible to attacks on reflectively weak keys.

The main issue that can appear is that the files sent to the cloud server are not always saved properly. This includes also the networks attacks.

This must be done in the exact opposite way the locking process has been performed; otherwise the user might not be able to access it or to create new versions of the backup.

3. Problem Solution

Each of these steps is time-consuming or resource-consuming. In order to improve these costs, the backup application must take into consideration the following recommendations.

If the backup is used for systems with multiple users or synchronized systems or even systems that are backing up great amount of data, between locking and unlocking the backup, the cloud solution should be able to create a read-only image of the backup, so that the users might access it especially if the solution provides an online tool to access the files.

3.1.2 Increase the security

Since private keys saved on the local machine are causing a lot of troubles, it is recommended to use the alternative way and to enforce the security and token should be used. Although if the systems are using an automated way of backing-up or a token can't be used, the backup solution could use the operating systems security to protect the session private key against unauthorized access.

In order to be sure that the process is using the correct version of the backup, the system should provide to the user a detailed report regarding the version selected and warn him if based on the data retrieved the backup might be corrupted. To perform this, the application should provide the functionality to test the backup used.

Since almost all the processes are performed on the host system, to ease the continuous transfer of the index file to and from the cloud server, the file can be saved locally in a temporary location until all other files are saved to the cloud server. If the index file is not sent back to the server right before the unlocking, all the updated files should be quarantined until the index file is saved.

To reduce the costs of time and the redundancy in the backup, the application should have a multi-layered detector to identify if the file is new or it have been changed since last backup. First of all it

should be verified the date of last change of the file. Secondly the dimension of the file should be checked against the old value. Another important check that must be performed is to verify if the file is a duplicate of another file or if the file has been moved from the original location. If so, only the path should be copied/changed in the index file that will be save with the backup.
Anyway, another approach should be used over the operating system files, due to the possibility that it could be a file that would only contain flags, and if one of these flags are changed and even if the size wasn't changed, the possibility of saving an old value might create a system failure at the recovery.
If the backup application is also having the purpose of storage reduction, at the installation of the application it should record the performance of the host machine through 2 indicators:
-Ar (the average number of MB that are archived per second)
-Cp (the average compression rate)
-Ench (the average number of MB that are encrypted per second)
Also, before each session of backup another indicator should be calculated:
-Tr (the average number of MB that are sent to the server per second)
Based on these indicators, if:

$$\frac{filesize}{Tr}+\frac{filesize}{Ench}<Cp*\frac{filesize}{Tr}+Cp*\frac{filesize}{Ench}+\frac{fileseze}{Ar}$$

then the system should encrypt the file and send it to the server, and the server should handle the archiving of the file, otherwise the archiving should be performed at the host.
Another technique used for minimizing the value of files archived is to group them by the type, especially if the files are similar. In the table below can be checked a comparison between .jpeg files that where the percentage of changes between files is lower than 10%.

Table 4. Normal size vs archived size

Nr of files/Type of archive	Without archiving(KB)	Zip	7-zip
1	2593	2588	2605
2	4690	4673	4680
4	9206	9163	6175
8	16998	16948	11325
16	33587	33070	15466
16*	33996	33896	11330

* 50% of files are duplicates

Using a regression model over the values recorded above we can see that for the Zip archive, the best regression equation is a linear model:

$$Zip(x) = 91.0764336872 + 0.983788320833 * x$$

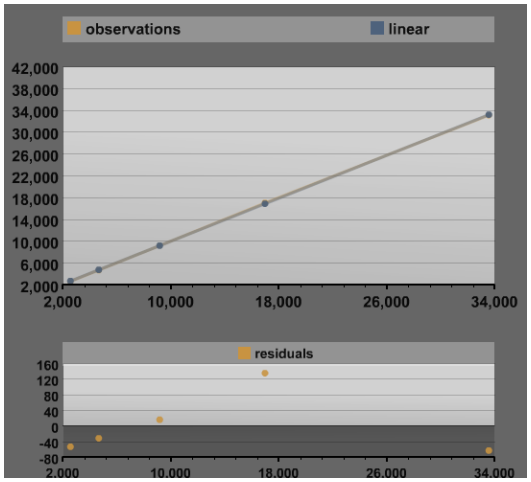

Figure 5. ZIP performance

While the 7zip compression method seems to have a power regression model:

$$7zip(x) = e^{(2.49707434326 + 0.691610862016 * \ln(x))}$$

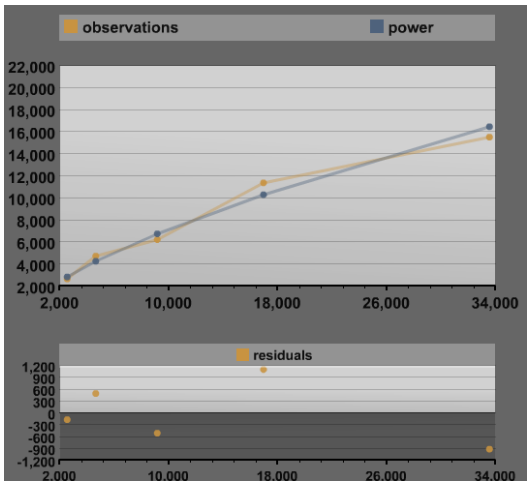

Figure 6. 7-ZIP performance

Even if as we can see that the zip files are faster to be archived and the report (compression/time used) is one of the

ISSN 2067 – 4074

best, the advantages of using the 7-zip archiving are starting to be noticed when the number of files is bigger, especially when a number of files are duplicates, this way the cost of storage can be reduced to almost 50%.
To ensure the integrity of the files that are sent from and to the cloud server the application should use a modern block cipher like AES, Salsa20 or Twofish instead of older version.
Another important aspect is the reduction of information lost between two sequential backups. To perform this without increasing the redundancy of the backup file and if the user allows it is to create a local version of the backup that can contain intermediary versions. This solution allows the system to perform a deeper optimization of the files and perform verification at the byte level for the files. Using this solution the files are actually saved in real-time and the performance of the server is increased, the application being activated only when the host isn't overloaded. Anyway, over the mobile devices it can be used the same main idea, it should only be used on the host mobile device a differential backup over an incremental backup due to lack of space and connectivity.

4. Conclusion

By using these techniques we can significantly improve the way how incremental backups are performed nowadays and to increase the security of files saved in cloud. These techniques are also significant for mobile devices where the user can have a lot of issues due to poor/lack of internet connection. If these are applied to personal data files the cost determined by data storage and bandwidth connection can be reduced up to 50%, while if these are used for business purpose the values are slightly lower but it depends a lot by the files that are stored. Also, the risk of losing data is lower, the user can choose to perform the backup activity more often and by having it saved in two places.

Acknowledgment

Parts of this research have been published in the Proceedings of the 6th International Conference on Security for Information Technology and Communications, SECITC 2013.

References

[1] Advantages you probably didn't know about the online data backup: *http://visual.ly/advantages-you-probably-didn%E2%80%99t-know-about-online-data-backup*
[2] Russ Fellows - Cloud backup tutorial: How to leverage cloud backup service: *http://searchdatabackup.techtarget.com/tutorial/Cloud-backup-tutorial-How-to-leverage-cloud-backup-services*
[3] Archive Comparison Test: *http://compression.ca/act/act-text.html*
[4] Simar Preet Singh, Raman Maini, Comparison of Data Encryption Algorithms, International Journal of Computer Science and Communication, Vol. 2, No. 1, 2011, pp125-127
[5] *Cunhua Qian, Yingyan Huang, Xufeng Zhao, Toshio Nakagawa*, Optimal Backup Interval for a Database System with Full and Periodic Incremental Backup, Journal of Computers, Vol. 5, No. 4, 2010, pp557-564
[6] Divya BHATT, A Revolution in Information Technology - Cloud Computing, Walailak Journal of Science and Technology, Vol. 9, No. 2, 2012, pp107-113