

Veeam backup methods and the impact on destination storage I/O



Luca Dell'Oca (vExpert 2011-2012, VCP, CISSP) works in ICT since 2000.

Virtualization architect and consultant since 2006, specialized on VMware design and management, Backups and Disaster Recovery.

VMware Italian User Group founder and board member, passionate blogger, runs his personal blog www.vuemuer.it where he writes about everything related to virtualization and cloud computing.



Veeam Backup & Replication v6 allows users to choose among several different backup methods to fit their need.

Each method has its own strengths and weaknesses, and there is always a tradeoff between used space on the backup storage and the amount of activity (both from the cpu and the IOPS generated on the storage) required to complete the backup.

However, backup designers often evaluate only the used space, keeping low attention on the impact each chosen method has on destination backup I/O. This is fundamental in the choice of the backup appliance so that backup times do not go over the desired backup window.

We will analyze in this paper the available method and their pros and cons. We will assume standard disk storage is used; different analysis may be required if deduplication storage appliances or other different systems are used.

Graphics reproduced in this paper are taken from the Official “Veeam Backup & Replication Version 6.0 User Guide for VMware Environments November, 2011”.

Copyright © 2005–2011 Veeam Software Inc.



“Veeam backup methods and the impact on destination storage I/O” by [Luca Dell'Oca](#) is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](#).

Forward Incremental Backup

This is the default method when a new backup job is created.

Also indicated simply as “Incremental Backup”, this is by far the most common method used in Veeam Backup projects.

During the first run, Veeam Backup & Replication creates a full backup file (.vbk). At subsequent backups, it only gets changes that have taken place since the last performed backup (whether full or incremental) and saves them as incremental backup files (.vib) next to the full backup.



It's the best choice if the backup files need to be moved or saved to tape, since only the new incremental backup file is created, and the previous files are not touched anymore, so are their archive bits.

On the other hand, restores are slower than in other methods (such as Reverse Incremental) because Veeam Backup server needs to aggregate the full backup and its increments to show to the backup administrator the desired restore point.

Also, based on the desired retention policies, an old full backup must be kept until the retention period is reached, resulting in a higher used space if compared to other backup methods.

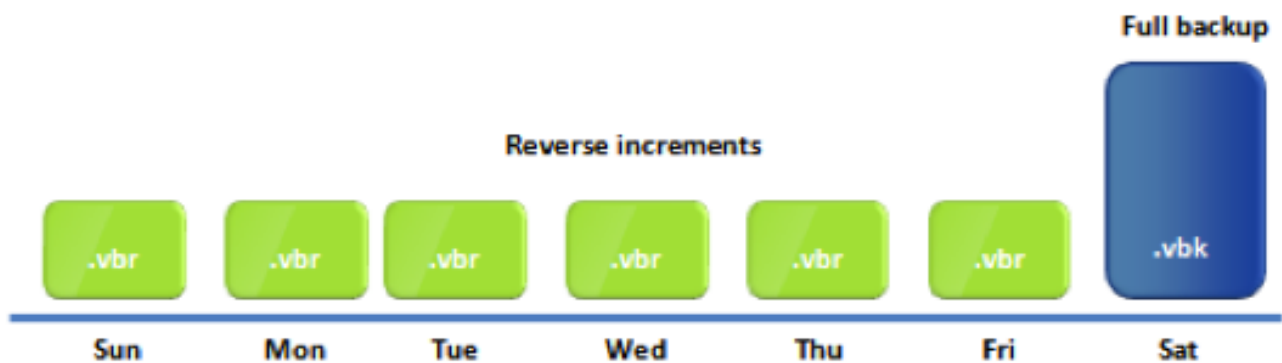
This is the lowest impacting method on backup storage: for every bit we have to save, 1 IO is used to write it on the storage itself. This makes the forward incremental method the best choice for backups involving low-end NAS.



Reversed Incremental Backup

This is the best choice if the backup storage is space-constrained and if fast restore of the last backup version is the highest goal of the users.

During the first run of a backup job a full backup of a VM is created and stored in a resulting full backup file (.vbk). All subsequent backups are incremental – that is, Veeam Backup & Replication copies only those data blocks that have changed since the last job run. During incremental backup, Veeam Backup & Replication “injects” changes into the .vbk file to rebuild it to the most recent state of a VM. It also creates a reversed incremental backup file (.vbr) containing data blocks that are replaced when the full backup file is rebuilt. Therefore, the most recent restore point is always a full backup, and it gets updated after every backup cycle.



This backup method lets you perform forever-incremental backup and save disk space as you have to store only one full backup. With reversed incremental backup, you do not have to perform periodic full backups to guarantee safety of data and to keep up with the specified retention policy. If the number of restore points allowed by the retention policy is exceeded, Veeam Backup & Replication will simply delete the earliest reversed increment.

However, this method uses 3 I/O for every written bit:

- 1° I/O to read the bit in the .vbk file that needs to be replaced
- 2° I/O to substitute that bit with the new one extracted from the VM disk
- 3° I/O to write the old bit into the newly created .vbr file



Also, since substitutions comes from different and unpredictable blocks of the production VM, we will have random access to the saved data, and this will even worse the performances of backups if compared to the sequential writes done during a forward incremental.

The real value can be 4x of the original I/O.

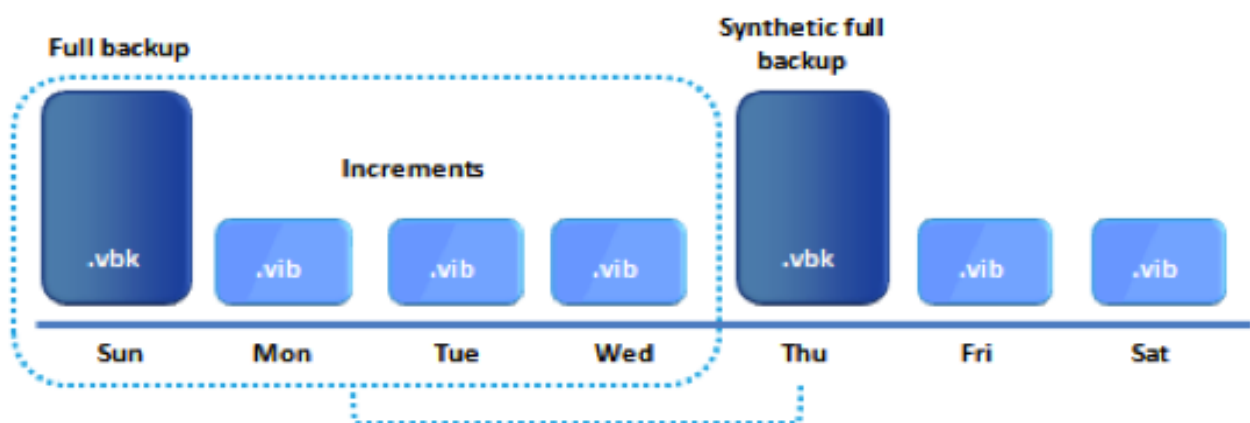
Forward Incremental Backup with Syntethic Full

Forward incremental is the best choice for offloading backups to tape, but one of its cons is the lack of a full backup if you need to do a fast restore of the last version of a VM.

To bypass this problem, you can use Syntethic Full: along with the usual forward incremental schedule, you can choose a day to perform a syntethic full. This combines the advantages of forward incremental (tape-friendly) and reverse incremental.

The full version is not created reading again from the production storage, but Veeam Backup & Replication uses the chain of full and incremental backups that are already kept on backup storage to create a new full backup.

Every next backup job run will create an incremental backup starting from the new full backup until next schedule of the new syntethic full.



This method uses 1 I/O for every written bit during the forward incremental, and 2 I/O during the syntethic full:

- 1° I/O to read the bit from the .vbk or .vib files
- 2° I/O to write the same bit into the resultant .vbk file

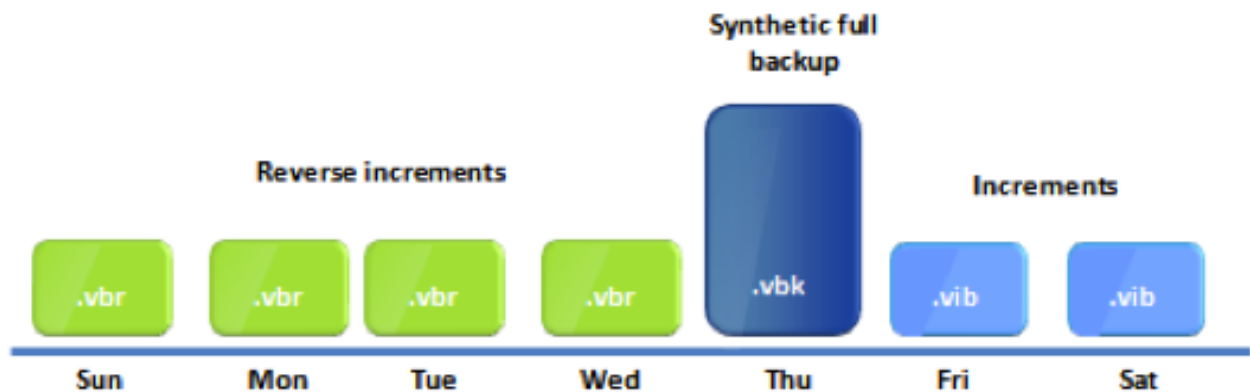


Forward Incremental Backup with Synthetic Full and Transform

If you select to create a synthetic full backup, you can additionally choose to transform all previous full backup chains to a reversed incremental backup sequence.

Both .vib files as well as the full backup will be transformed to reversed increments (.vrb files), and you will only have a full backup. This option enables you to keep only one full backup image on disk and so reduce the amount of space required to store backups.

However, such transformation takes more time than simply creating a periodic full synthetic backup and is a really heavy operation on both the Veeam Proxy doing the transformation and the backup storage.



From an I/O perspective, the combined use of synthetic full and transform on a forward incremental requires 4 I/O for every bit:

- 1° I/O to read the bit from the .vbk or .vib files
- 2° I/O to write the same bit into the resultant .vbk file
- 3° I/O to read again the bit from the .vbk or .vib files
- 4° I/O to write the resultant .vrb file



Conclusions

As I have described in the previous scenarios, the method you choose to backup your VMs have a deep impact on the backup storage.

If you take for example a small company, they usually have a constrained budget and they will likely choose a cheap NAS for their backup storage. That appliance will be filled with off-the-shelf SATA disks, and they will setup a raid5 to save space. This design is completely reasonable; however, if their IT guy will choose for example the Reversed Incremental to further save on disk space, there are chances the cheap NAS will perform really bad because it's not powerful enough to load the stress of 2 I/O per saved bit.

There is not a best choice from the 4 ways you can do a backup with Veeam Backup & Replication, it always depends on your environment: for example if you do not need to offload to tape or off-site and you have fast storages, you can probably go for the Reversed Incremental. But, if you are going to save to a deduplication appliance, reversed is not the best choice, so you will go back to the usual Forward Incremental.

Just like in any (well designed) IT project, you need to think about every piece of the whole design, and the goal of this paper has been to highlight the importance of I/O load on the backup storage: this is definitively one of the elements you need to care about.

