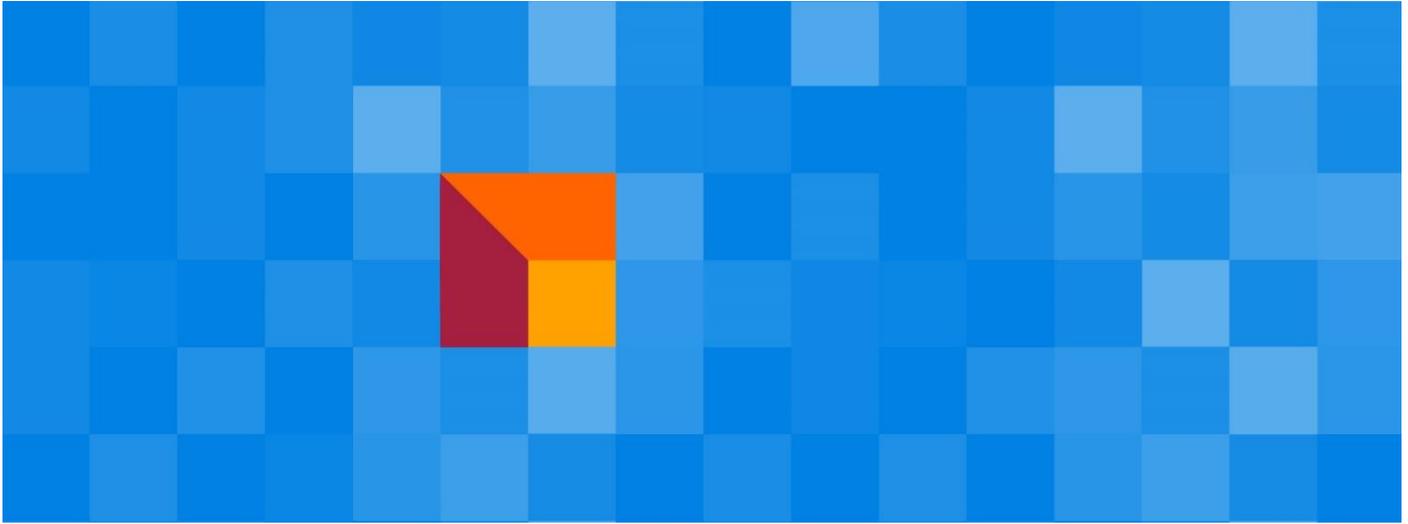


# Microsoft Azure Storage Account benchmarks and good practices

Quincy HSIEH



If you ever have worked on the Microsoft Azure as infrastructure as a service (IaaS) for your insurance APIs or other online claim applications, you might have used Azure Storage Account to store the files for the data persistence.

Milliman recently re-designed and enhanced an old application system to a new API system with benefits of the scalable Azure Scale Set. In this project, we encountered some disk performance latency issues in the new API system upon copying files from directory to directory within the storage account. Our system needs to frequently copy a total of 10 MB of 600 files within the same storage account to prepare an execution environment. Each copy took about 35 seconds, which was very long in comparison to the copy within the system disk of a virtual machine.

This prompted us to do some tests to find the optimal solution with Azure Storage Account. This article is about sharing what we have experienced.

## Deployment of Azure Storage Accounts

Microsoft Azure Storage Account (SA) provides several data types (blobs, files, queues, tables or Spark data lake) for the storage data that is accessible from anywhere via HTTP or HTTPS. SA ensures that data are properly stored when your Virtual Machine (VM) reallocated or scaled up. It also ensures that files are durable and that there is persistence in case of physical damage like power failure with data loss or natural disaster. Those are reasons we use SA over VM system disk for data persistence on Azure.

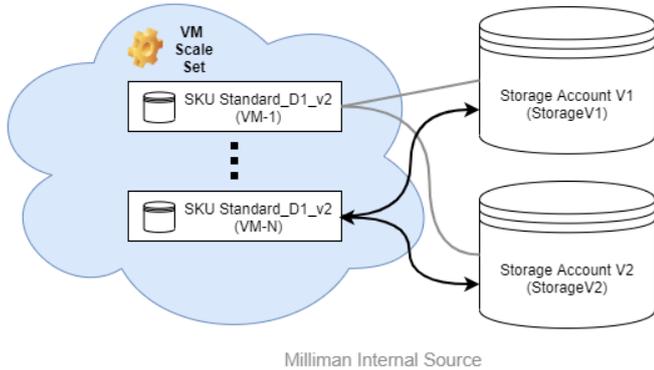
<sup>1</sup> Azure Storage redundancy. (June 22, 2020). Retrieved on July 6, 2020, from <https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy>

Since 2017, Microsoft has created the second version of Storage Account, the General-purpose v2 account, which added 2 tiers of hot and cold access. We considered both versions, the General-purpose v1 account and the General-purpose v2 account, to benchmark the performance of file copy.

Because our system is an API application with various possible sizes of files of random access, the data type “files” of SA is suitable for our needs. To limit the budget of every byte we stored, we use “Standard storage” instead of “Premium storage”. For the storage redundancy, we choose Locally Redundant Storage (LRS) <sup>\*1</sup> to simplify the test by avoiding internet delay over storage replication. The test VM and SA are in the same zone within the same network security group.

With all the specifications of the choices above, we have prepared the following Azure devices for the test as Figure 1.

FIGURE 1: DEVICE DEPLOYMENTS



Within the virtual machine, we created two mounting directories for our tests. The directory /media/storageV2 mounted to StorageV2 and directory /media/storageV1 for StorageV1.

This document focuses on the SA benchmark to know how to mount the SA with SMB 3.0 to a VM directory, please refer to the Microsoft document<sup>2</sup>.

### File copy benchmarks

We defined the following scenarios to benchmark the copy of the same size of files (10MB of 600 files in our case):

- Scenario 1 - Copy files from the directory mounted at the root node "storage" of SA to the same mounted directory. The fstab mount configurations are as follows:

```
//StorageV1.file.core.windows.net/storage on
/media/storageV1 type cifs (rw, relatime, vers=3.0,
sec=ntlmssp, cache=strict, uid=0, noforceuid, gid=0,
noforcegid, file_mode=0777, dir_mode=0777, soft,
persistenhandles, nounix, serverino, mapposix,
mfsymlinks, rsize=1048576, wsize=1048576,
echo_interval=60, actimeo=1)

//StorageV2.file.core.windows.net/storage on
/media/storageV2 type cifs (rw, relatime, vers=3.0,
sec=ntlmssp, cache=strict, uid=0, noforceuid, gid=0,
noforcegid, file_mode=0777, dir_mode=0777, soft,
persistenhandles, nounix, serverino, mapposix,
mfsymlinks, rsize=1048576, wsize=1048576,
echo_interval=60, actimeo=1)
```

- Scenario 2 - Copy files from the directory mounted at the sub-directory "shared" under the root node "storage" of the SA to the same mounted directory. The fstab mount configurations are as follows:

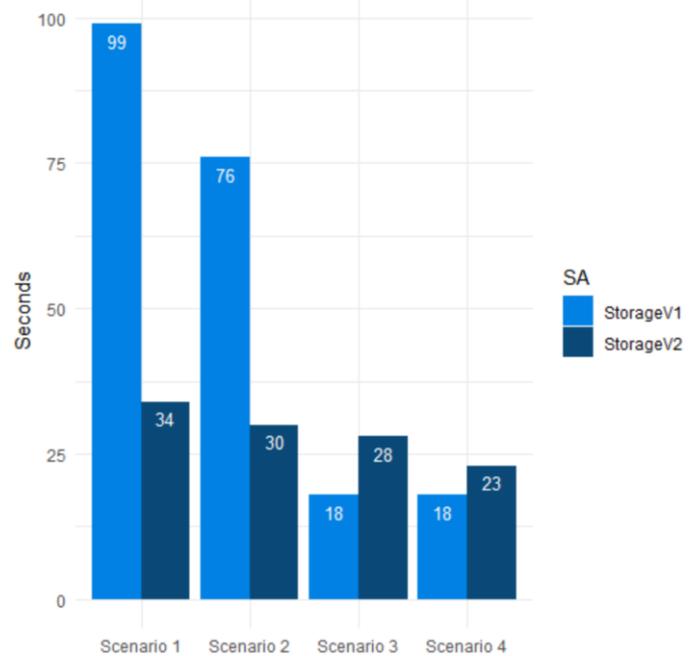
```
//StorageV1.file.core.windows.net/storage/shared on
/media/storageV1 type cifs (rw, relatime, vers=3.0,
sec=ntlmssp, cache=strict, uid=0, noforceuid, gid=0,
noforcegid, file_mode=0777, dir_mode=0777, soft,
persistenhandles, nounix, serverino, mapposix,
mfsymlinks, rsize=1048576, wsize=1048576,
echo_interval=60, actimeo=1)
```

```
//StorageV2.file.core.windows.net/storage/shared on
/media/storageV2 type cifs (rw, relatime, vers=3.0,
sec=ntlmssp, cache=strict, uid=0, noforceuid, gid=0,
noforcegid, file_mode=0777, dir_mode=0777, soft,
persistenhandles, nounix, serverino, mapposix,
mfsymlinks, rsize=1048576, wsize=1048576,
echo_interval=60, actimeo=1)
```

- Scenario 3 - Copy files from the VM home directory on the system disk to the SA mounted at root node "storage". The mount fstab configurations are the same as Scenario 1.
- Scenario 4 - Copy files from the VM home directory on the system disk to the SA mounted at the sub-directory "shared" under the root node "storage". The mount fstab configurations are the same as Scenario 2.

In each copy scenario, we compared the copy between Storage Account V1 and V2. Figure 2 presents the test result.

FIGURE 2: TEST RESULT



As we can see from the test result, if we copy files within the SA (scenarios 1 and 2), Storage Account V1 is very slow. However, if we copy files from VM to the SA (scenarios 3 and 4), Storage Account V1 is equal or faster than Storage Account V2. More interestingly, if we mount the directory with a sub-directory in the SA, the latency of copy is slightly reduced.

<sup>2</sup> Mount Azure File storage on Linux VMS using SMB. (June 28, 2018). Retrieved on July 6, 2020, from

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mount-azure-file-storage-on-linux-using-smb>

## Good practices

From the test result and our previous experiences, we suggest the following practices for better performance:

- Create a sub-directory in the SA and mount VM directory on this sub-directory of the SA. This will increase performance.
- Use Storage Account V1 if there are a lot of VM file copies from VM system disk to SA in your application.
- Use Storage Account V2 if there are a lot of file copies within SA in your application.
- Users should also consider implementing a cache mechanism in the system using VM system disk as a cache with Storage Account V1 – to benefit the speed of copy from VM system disk to SA. The cache mechanism with VM Scale Set is illustrated in Figure 3 and Figure 4 for its logic flow.

FIGURE 3: CACHE MECHANISM ON VM SCALE SET

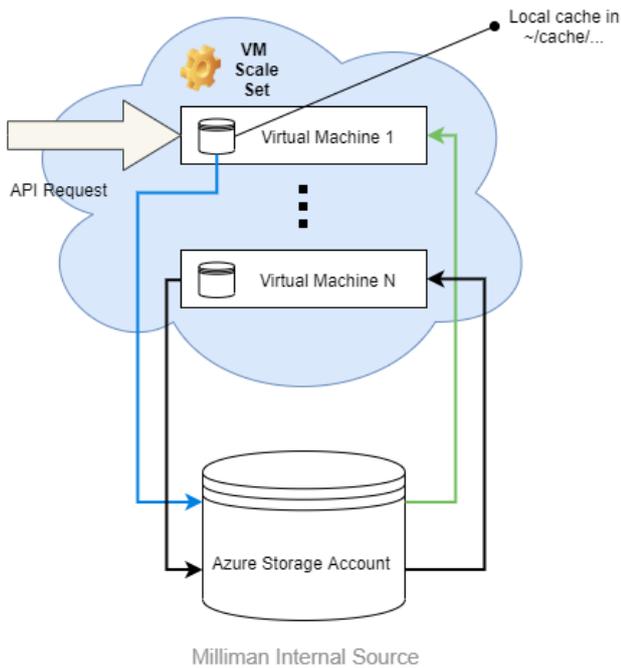
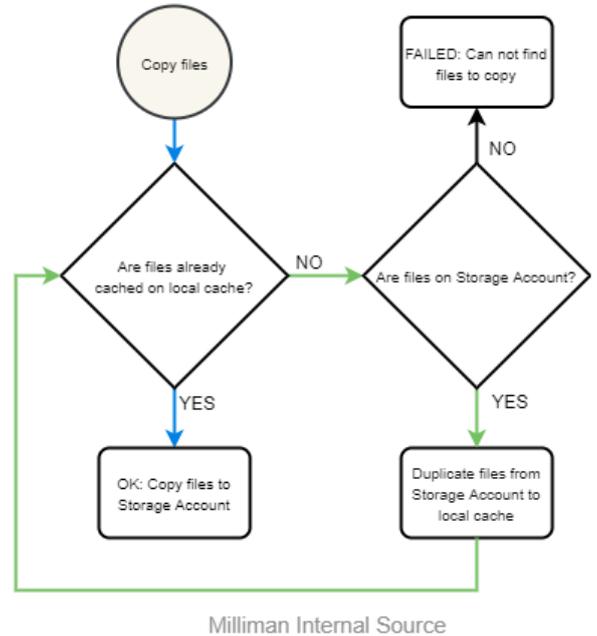


FIGURE 4: LOGIC FLOW OF CACHE MECHANISM



The blue arrow is the most frequent way of copying cached files. The green arrow is the first time when files are not in the cache. This cache mechanism will be helpful when mounting a SA on a VM Scale Set to avoid a frequent copy of the same files within the SA. With this cache-on-system-disk mechanism, only the first time copy (from SA to system disk cache, and then from system disk cache to SA) will take slightly longer than direct copy within the SA. Once the files are in the cache, we can benefit from the speed of IO from the VM system disk to the SA.

## Our cloud experience

As more and more companies move their IT infrastructure to the cloud, Milliman helps its clients to implement their projects efficiently with best practices. Our analytics teams are skilled on these topics and will share more feedback in the future. For further information, please contact any of the consultants below or your usual Milliman consultant.

Milliman is among the world's largest providers of actuarial and related products and services. The firm has consulting practices in life insurance and financial services, property & casualty insurance, healthcare, and employee benefits. Founded in 1947, Milliman is an independent firm with offices in major cities around the globe.



## CONTACT

**Remi BELLINA**  
[remi.bellina@milliman.com](mailto:remi.bellina@milliman.com)

**Quincy HSIEH**  
[quincy.hsieh@milliman.com](mailto:quincy.hsieh@milliman.com)

[milliman.com](http://milliman.com)

© 2020 Milliman, Inc. All Rights Reserved. Milliman makes no representations or warranties to the reader with respect to the information contained in this document ("Information") or to any other person or entity, as to the accuracy, completeness or merchantability of the Information. The reader of this document should not construe any of the Information as investment, legal, regulatory, financial, accounting or other advice and persons should consult qualified professionals before taking any specific actions. Milliman shall not be liable to the reader of the Information or any person or entity under any circumstances relating to or arising, in whole or in part, from any circumstance or risk (whether or not this is the result of negligence), or, for any losses, damages or other damages caused in connection with the publication of the Information or its distribution. The holder of this document agrees that it shall not use Milliman's name, trademarks or service marks, or refer to Milliman directly or indirectly in any media release, public announcement or public disclosure, including in any promotional or marketing materials, customer lists, referral lists, websites or business presentations without Milliman's prior written consent for each such use or release, which consent shall be given in Milliman's sole discretion.

This Information contained therein is protected by Milliman's and the authors/co-authors' copyrights and must not be modified or reproduced without express consent.