Microsoft

Avere & PixStor: Configuring SMB Shares

December 2020

Introduction

A popular media NAS storage filer for M&E Rendering Studios is the Pixit Media PixStor. Artists create media content directly on the PixStor Filer using the NFS or SMB protocols. Content is then rendered on-premises or in the cloud on render nodes.

To help scale to tens of thousands of render nodes Avere Technology can mount a PixStor to expand throughput and IOPs. The Avere may only mount NFSv3 shares. When the PixStor only exposes NFSv3 shares the Avere can successfully mount and scale the IOPs and throughput. However, when a mixed mode environment of NFS + SMB is required, the default setup will result in permissions issues.

This document describes the two problems in the default configuration and their solutions.

Constraints

This solution has the following constraints:

1. **RID Mapping** – The PixStor uses the Samba rid mapping scheme and has the limitations outlined in the Samba Wiki [Samba Wiki 2020].

2. **Single Share / Single Group** – Due to the limitation of POSIX mode bits, there can only be a single group assigned. Since Avere can only access the PixStor using the NFSv3 protocol, this means that all users accessing a specific SMB Share through the Avere must have their primary group assigned to the group defined on the share.

   It may cause confusion that a user belonging to a different group can access the PixStor, but not the Avere SMB Share. This is explained by the fact that the extended groups contain additional groups, but the additional groups are not exposed through to the Avere.

3. **PixStor Kernel NFSv3** – Kernel space NFSv3 is required on the PixStor. The NFS-Ganesha in the container space mode of PixStor returns "NOENT" on readdirplus calls, and therefore NFS-Ganesha + GPFS is unsupported on Avere.

   Here is a related bug:

   https://review.gerrithub.io/c/ffilz/nfs-ganesha/+/472446 - *NFS readdir operation – "Sometimes we receive invalid handle from lookup of "..", handle it by sending DELAY error and hope it goes away in a retry!"*

The User / Group Problem

The PixStor provides a method to automatically map the Windows Security Identifier (SID) to unix based UIDs and GIDs. It does this by combining a unique integer offset with the Relative Identifier (RID) [Samba Wiki 2020].

To implement this mapping, there are two approaches.

The first approach is automatic and will generate the user and group files on cluster creation. To enable assign your base RID integer to the cifs_rid_mapping_base_integer attribute as shown in the following example:

```
cifs_rid_mapping_base_integer = 1087660000
```

The second approach is manual and requires the following two-step approach:

1. Run the following powershell script against your Active Directory service using the integer offset with: https://github.com/Azure/Avere/blob/main/src/terraform/examples/houdinienvironment/Get-AvereFlatFiles.ps1. This will generate a user and group file with the correct mappings. The users default group is the group with the smallest RID.

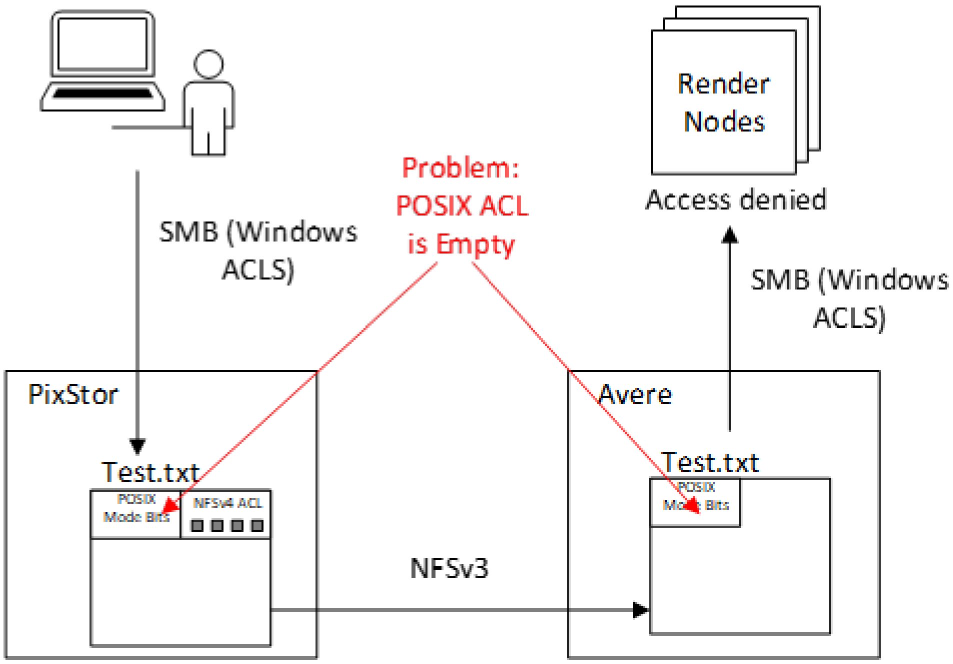
2. Store the user and group file on a web server that is reachable by the Avere. Next provide the two URLs of the user and group file to Avere Directory services. Alternatively, specify the two files to the Avere Terraform provider as shown in the following example: https://github.com/Azure/Avere/blob/main/src/terraform/examples/houdinienvironment/3.cache/main.tf#L66-L67.

The above manual step may be automated in the Windows AT task scheduler and run at a frequency similar to how often the user and group accounts change.

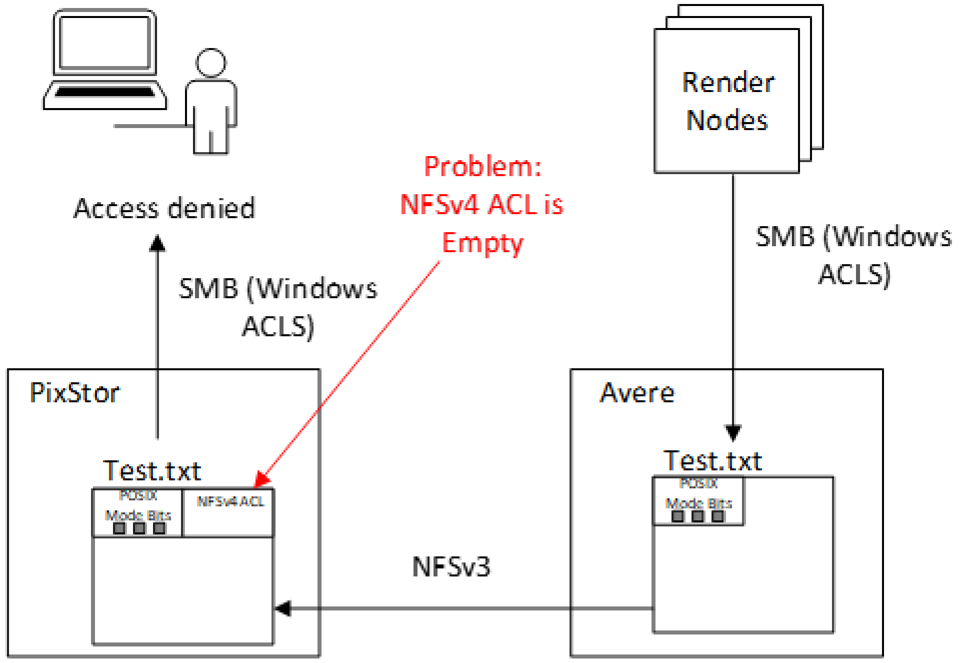
The ACLS Problem

Pixit Media PixStor stores the ACL of each file in a single extended NFSv4 ACL. The ACL interoperability problem impacts the compatibility between POSIX mode bits and the NFSv4 ACLS. This results in the permission denied problem if the files are coming from on-prem or from the cloud.

When the file gets written from an on-premises workstation it receives an NFSv4 ACL, and empty POSIX mode bit. Then when the file goes across the NFSv3 boundary to the Avere it loses its NFSv4 ACL as shown in the diagram below. Then when the render nodes go to read the file, they encounter access denied errors.



The reverse situation is not much better. In the reverse case, the POSIX mode bits are preserved, but the NFSv4 ACL is now empty. The result is that when the on-premises user reads the file they will encounter access denied:



The solution here is to leverage the ACL inheritence model of PixStor in combination with the create and directory masks of the Avere.

For PixStor, the solution is to configure each new share export with file and directory inheritance and the acls shown in the following table. The mode bits permissions and NFSv4 ACLS are essentially implementing the equivalent of 770 for mode bits.

Here are the steps to enable the correct permissions on the shares:

1. Configure PixStor with "no_root_squash" and "rw" to the avere range

2. Choose a primary user and group. All clients accessing the Avere cache must belong to the primary group. Otherwise, clients not belonging to the primary group will be blocked from accessing the Avere cluster. However, if additional groups are specified in the PixStor ACLs, they will still be allowed to access the PixStor share directly.

3. Create the folder to share, and "chown" the user and group to the primary user and group. The folder mode bits should be 770. Then cascade the user and group with the following command: `chown -R 10876601203:10876600513 someShareName.`

4. Cascade the following permissions:

ACL	Mode Bit Permissions and Inheritance	NFSv4 ACLS
special:owner@	rwxc:allow:FileInherit:DirInherit:Inherited	(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR (X)READ_NAMED (X)DELETE (X)DELETE_CHILD (X)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED
special:group@	rwxc:allow:FileInherit:DirInherit:Inherited	(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR (X)READ_NAMED (X)DELETE (X)DELETE_CHILD (X)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL

		(X)WRITE_ATTR (X)WRITE_NAMED
special:everyone@	`----:allow:FileInherit:DirInherit:Inherited`	(-)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL (-)READ_ATTR (-)READ_NAMED (-)DELETE (-)DELETE_CHILD (-)CHOWN (-)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED
user:<RID> (specify one or more users the share is working with)	`rwxc:allow:FileInherit:DirInherit:Inherited`	(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR (X)READ_NAMED (X)DELETE (X)DELETE_CHILD (X)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED
group:<RID> (specify one or more groups the share is working with)	`rwxc:allow:FileInherit:DirInherit:Inherited`	(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR (X)READ_NAMED (X)DELETE (X)DELETE_CHILD (X)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

For example, to prepare a share with groups `1087660513` and `1087660512`, you would take the following steps:

1. Create the share by creating the directory and updating PixStor configuration to expose the share as SMB and NFSv3.

2. Next copy the following contents to a file, and run the command “`mmputacl –i the_acl_filename top_dir_name`”

```
#NFSv4 ACL
#owner:1087661000
#group:1087660513
special:owner@:rwxc:allow:FileInherit:DirInherit:Inherited
(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
(X)DELETE   (X)DELETE_CHILD (X)CHOWN    (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

special:group@:rwxc:allow:FileInherit:DirInherit:Inherited
(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
(X)DELETE   (X)DELETE_CHILD (X)CHOWN    (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

special:everyone@:----:allow:FileInherit:DirInherit:Inherited
(-)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL  (-)READ_ATTR  (-)READ_NAMED
(-)DELETE   (-)DELETE_CHILD (-)CHOWN    (-)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED

group:1087660513:rwxc:allow:FileInherit:DirInherit:Inherited
(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
(X)DELETE   (X)DELETE_CHILD (X)CHOWN    (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

group:1087660512:rwxc:allow:FileInherit:DirInherit:Inherited
(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
(X)DELETE   (X)DELETE_CHILD (X)CHOWN    (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED
```

Notice that both of the groups above is constructed from the unique integer of `1087660000` and the RID of `513` and `512` respectively.

Next the Avere needs to be configured with CIFS enabled and configure each CIFS share with no ACES and an octal value of 0770 for the create and directory masks. The ACE needs the same group as defined above, and any other groups that are accessing the share.

Here is an example configuration to use for the Avere Terraform Provider corefiler block:

```
core_filer {
    ...
    cifs_share_name = "exampleshare"
    # remove all aces
    cifs_share_ace = "S-1-5-21-1372296732-3586249450-4126101624-513"
    cifs_create_mask = "0770"
    cifs_dir_mask = "0770"
    ...
}
```

To get the SID, you can run either of the following two powershell commands for user or group:

```
# find SID for user name "azureuser"
Get-ADUser -Filter * -Properties SamAccountName, SID | where-object
SamAccountName -eq "azureuser"
# find SID for "Domain Users"
Get-ADGroup -Filter * -Properties SamAccountName, SID | where-object
SamAccountName -eq "Domain Users"
```

Once you have configured the share on both the PixStor and the Avere vFXT, test creating, writing, and removing files both in the cloud and on-premises. Also test creating / removing directories in both environments and confirming the inherited permissions are correct.

Contact

For contact, please reach our team at azurerendering@service.microsoft.com.

References

Avere. Active Directory Administrator Guide to Avere FXT Deployment. 2014-07-16.

Avere. Cluster->Directory Services. 2020

Samba Wiki. Idmap Config Rid , 2 May 2020.

Terraform avere_vfxt Provider junction CIFS configuration. 2020

The ACL Interoperability Problem. 2020