IBM Spectrum Scale and the cloud

v1.3 – 21/03/2017 – Simon Thompson, University of Birmingham

Context

This report makes comment on the use of IBM Spectrum Scale[1] with respect to its applicability to cloud and is drawn from the experience gained by several projects in additional to on-going discussion with members of IBM Research and the Spectrum Scale development teams. A workshop was also coordinated by the Spectrum Scale user group [2]in the UK in October 2016 and content from that workshop is drawn upon for this report. The workshop was attended by both users from the community (including The Francis Crick[3] institute) and from commercial suppliers (e.g. DDN, OCF).

Throughout this report, we refer to IBM Spectrum Scale, in the past, this product has been named GPFS or General Parallel File-System. In general, there is no significant difference, however comments in this report may only be applicable to recent releases on IBM Spectrum Scale. A number of products that use Spectrum Scale are marketed by other companies (for example DDN GridScaler), the content of this report should stand regardless of OEM supplier.

IBM Spectrum Scale is a highly scalable, parallel file-system designed to provide software-defined storage at huge scale, both in terms of storage capacity and numbers of clients accessing the file-system. It is extensively used in HPC system deployments and also increasingly for sites wishing to deploy scale-out storage systems to hold research data.

The contents of this report provide a point-in-time snapshot of the state of IBM Spectrum Scale and its cloud readiness in early 2017. Any comments related to IBM development relate to research work IBM have openly discussed but do not provide a commitment to deliver.

A note on the author:

Simon Thompson works for the University of Birmingham as Research Computing Infrastructure Architect. He has been involved in the development of BEARCloud and also the multi-site, MRC funded CLIMB project. He has a number of years experience in deploying Spectrum Scale and is the chair of the Spectrum Scale user group as well as being a member of the RCUK Working Group on Cloud. He has also chaired a panel on HPC file-systems [4]at the SC16 international conference in Salt Lake City. It is from these engagements and through discussion with other members of the community that the experience for this report is drawn.

---

[1] http://www-03.ibm.com/systems/uk/storage/spectrum/scale/ IBM Spectrum Scale

[2] www.spectrumscale.org - Independent user group focused on Spectrum Scale

[3] www.crick.ac.uk - member of eMedLab consortium, MRC funded OpenStack cloud

[4] http://sc16.supercomputing.org/presentation/?id=pan120&sess=sess185 SC16 programme

Operating Spectrum Scale in the Public Cloud

It is currently a non-trivial task to operate Spectrum Scale in the public cloud. There are a number of factors for this. The first of these is that traditionally deployments use twin-tailed storage, a feature generally not available in cloud environments. This is generally a pre-requisite for operating Spectrum Scale servers to allow multiple systems to directly access the same underlying storage infrastructure. In addition to this, fast networks (Infiniband, 40GbE) are generally not available, and the impact of shared connectivity between VMs running on shared hypervisors is difficult to define and benchmark consistently. There is also an abstraction layer in terms of selection of drive (SAS, NL-SAS, SATA) and IO controllers which limits the ability to select and effectively benchmark performance storage.

As many of the technologies used for storage virtualisation are new, there is also a question of testing and validation of the storage infrastructures in terms of both data validation and of performance as well as questions regarding data placement of storage in a virtualised world. For example, say two servers were attached to different EBS storage for resilience, is it possible to guarantee that the physical storage is different to protect against failures?

Taking the lack of twin-tailed storage as an example, it is possible to configure Spectrum Scale to use non-shared storage, however this typically requires a replication factor of 2 or more which has a performance impact from the client needing to write multiple copies of the data. Whilst it would technically be possible to run without replication, this would be highly unwise as the loss of any one storage node would result in loss of the entire file-system.

IBM have undertaken some experimentation to use their file-placement-optimiser (FPO) code to investigate the feasibility of running Spectrum Scale servers in the public cloud, however this requires great care in configuration and understanding of the workload being used to optimise the configuration for performance. This includes significant deployments in IBM's cloud offering SoftLayer which has been tested with up to 900 nodes on an HPC workload. Typically the testing work has disabled the FPO features for data affinity as the nodes and storage are virtualised, but significant effort has been invested to try to optimise the configuration. Current configuration recommendations are provided in an IBM RedBook [5] as well as tuning recommendations[6].

A further issue that requires consideration is the license models of Spectrum Scale. Using a socket based license requires that all sockets in the system are licensed and in the public cloud, this may be impossible to determine and hence license correctly. The new capacity based license may be more appropriate as it would be based on storage capacity available, however this does not lend itself well to dynamic or on-demand

---

[5] http://www.redbooks.ibm.com/redpapers/pdfs/redp5410.pdf

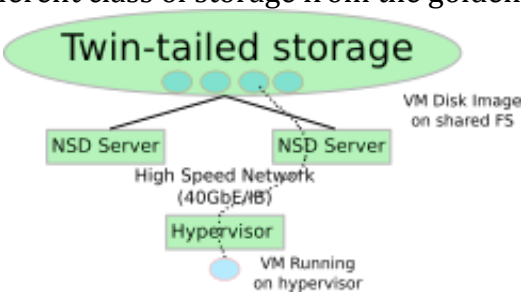
[6] https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20%28GPFS%29/page/IBM%20Spectrum%20Scale%20Tuning%20Recommendations%20for%20Shared%20Nothing%20Environments

deployments (e.g. sudden need for a cloud deployment). The IBM technical team reported it was investigating options for providing more dynamic models of licensing Spectrum Scale (e.g. providing an Amazon Machine Image which would allow customers a much more dynamic model of purchasing/packaging the software) but there is no official update on this work.

When operating Spectrum Scale in a private-cloud deployment, many of these issues go-away as there is a direct understanding of the systems in use and typically the Spectrum Scale server deployment would be undertaken in a bare-metal manner rather than attempting to run storage servers in the VM environment.

Spectrum Scale as an infrastructure layer for Private Cloud

OpenStack provides an open driver layer for storage integration and IBM have developed a Cinder driver (for block storage) which interacts with Glance (the image service). This allows certain types of disk image to be rapidly cloned using the internal mmclone command. For example a golden disk image would be used and VM instances would only track copy-on-write changes from the golden image. The driver is also configurable to place the cloned image onto a different storage pool which allows it to be used placed onto a different class of storage from the golden disk image.



Example architecture showing VM Disk Image on share infrastructure

The key advantages of using Spectrum Scale for the storage infrastructure layer are:

- many sites run Spectrum Scale for data services and VM hosting on exiting Spectrum Scale infrastructure gives a single data plane and reduces management complexity
- shared storage enabling rapid migration of VMs as images do not need to be transferred
- hypervisors can use Infiniband to enable high speed access to storage
- features such as LROC and HAWC can accelerate access to VM images
- use of file-system features (mmclone) enables rapid image deployment with a consistent deployment time regardless of VM disk image size

The OpenStack Spectrum Scale driver has been in the main-line code for some time and since its initial deployment, the QCOW2 driver has been developed to enable copy-on-write. Testing to determine the IO patterns of the different disk image formats needs significant investigation to determine if there are benefits to not using the Spectrum Scale driver over the QCOW2 drivers.

Cloud enabled access to Spectrum Scale

As part of protocol stack for Spectrum Scale, IBM provide packages and support traditional protocol access through NFS and SMB interfaces. In addition to this, there is also an object access stack. Internally this is based on providing a pre-defined set of OpenStack Swift packages, including the necessary packages for authentication (e.g.

Keystone). There is also a white paper from IBM detailing how to use external Swift packages to provide object access to Spectrum Scale file-systems.

It should be noted that one of the advantages of Swift on Spectrum Scale is that it is possible to provide multiple Swift servers to front the file-system without the need for Swift object replication (as the storage is shared between gateway systems). Internally the Swift gateway uses direct POSIX access in the same way as a non-Spectrum Scale enabled Swift server. In addition to this, IBM have developed middleware for Swift for deployments using HSM tape tiers[7] for long-term cold data storage.

One further feature of the Spectrum Scale Swift implementation is the ability to use "object-on-file unified access". This access method is designed for sites wishing to access both POSIX and object to the same storage, this required that a special objectising process is undertaken when files are added from the POSIX interface to ensure they are made available for Swift users. Note that prior to v4.2.2[8], it was not possible to convert an existing directory tree into a unified access system and that the POSIX access method to unified storage is prescriptive on the directory structures used.

Challenges for Spectrum Scale in a cloud-enabled world

The challenges outlined in the following section are applicable both to public and private cloud deployments and focus on the provision of access to Spectrum Scale data from cloud systems and assumes that a suitable storage infrastructure has already been deployed.

It is first important to understand why access to file-systems such as Spectrum Scale are important in a cloud world. It is of course possible to place data directly into VM disk images or onto ephemeral disks, however this solution is less than ideal. In properly integrated solution where bare-metal compute systems or other data access solutions are in place (e.g. Windows SMB access to file-systems), then a common storage sub-system is necessary. The use of shared file-systems also makes the protection of data easier, for example the backup and replication of key data is easier when it is outside of the control of users of VMs and is integrated into a centrally managed storage solution.

Finally, testing at Birmingham with users of the CLIMB project has shown significant limitations file-systems such as ext4 or XFS, commonly used to format directly attached ephemeral disks. For example when laying down significant amounts of data from sequencers, the ability of the file-system to perform in a timely manner has been limited. Working with one research group, it was identified that the only realistic way of reading the data in the VM ephemeral disk was to use inode traversal and not using traditional directory access mechanisms.

---

[7] https://wiki.openstack.org/wiki/Swift/HighLatencyMedia

[8] https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_enablingobjectaccessonexistingfilesets.htm Object enabling a "legacy" fileset

Trust and multi-tenancy

The Spectrum Scale client is inherently trusting. This is due to the design of it being part of a very fast access mechanism to files in trusted environments. This therefore limits its ability to operate in multi-tenant environments.

Where all access within a VM is trusted, then this does not apply, however there are significant challenges

OpenStack provides a file-sharing component called Manila. This in essence is designed to make NFS/CIFS servers and shares on the fly, however the Spectrum Scale driver is limited in the way it works. As it builds configuration of the CES protocol stack, it requires that the VMs are on a flat (single) network and there is no-way to use truly software defined networks with CES. This may be of use in some environments, but has many drawbacks.

The use of NFS does not in itself solve the multi-tenancy problem. NFSv4 with identity mapping is required which requires configuration on the client VM. It is also important to ensure that VMs have fast access to the NFS servers.

The University of Birmingham has undertaken some significant work in addressing some of these issues but still does not have an ideal solution. The solution currently being tested uses two approaches. Where VMs are attached to VXLAN software networks, the network gateway nodes have been configured with CES (not using the Manila driver). This allows the VMs running on the VXLAN layer 2 access (non routed) to the NFS server shares. Server shares are then manually configured to provide the appropriate shares into specific tenant networks/VMs, given projects are allocated their own network and storage space, this means that the shares made available are only those related to the project. Note that this approach requires some intervention when users define networks to ensure there is no IP address overlaps which would make it impossible to determine which network the share should be made available to. Where VMs are attached to VLAN networks, these use the centrally provided CES NFS servers (note that this requires layer 3 networking, but the network is specifically designed to reduce routing hops and provide maximum bandwidth with minimal latency).

"Elastic" scaling

Where a native Spectrum Scale client is used inside VMs, either as a separate cluster or as part of an existing cluster, it is currently difficult to scale the cluster on the fly. This is due to the way in which clients are added to the cluster which is accomplished by running commands on a server node in the cluster which copies configuration to the client node before Spectrum Scale can be started. Note that it is not possible to pre-seed the VM with the appropriate configuration/keys as all nodes in the cluster require some degree of configuration to be notified of a new node. This means that the VM must be running and contactable for the node can be added to the cluster.

Some experimentation has been undertaken at the University of Birmingham where clients are pre-created in the cluster, however this causes delays when using cluster management commands. This is because the cluster is aware of a number of nodes which are currently un-contactable and hence timeouts when running admin commands occur.

Further to this, there is a risk that in a cloud deployment of Spectrum Scale clients, that a user could very easily and quickly delete/poweroff a significant number of VMs. As a parallel file-system, any member of the cluster may become the meta-node for a file or directory. If another client requires access to a file/directory, it must contact the meta-node for the item, if the VM is destroyed when it is a meta node, this may cause the file-system to hang whilst the Spectrum Scale client is expelled and file-system recovery completes. This is the same with a bare-metal deployment, however it is unlikely that significant numbers of nodes would be "powered-off" on the fly. This hang is to be expected and should only be for a few seconds to minutes and is an effect of the user requirement to support POSIX semantics for locking of files. If the was no recover process, the whole locking process would be meaningless and is essential to maintain file access integrity and would affect all shared file-system implementations.

Given the difficulty with elastic scaling and the risk of clients being rapidly removed, it is essential that the operator understands that running Spectrum Scale in a virtualised environment is very similar, static manner to bare-metal deployments. It is perfectly feasibly to use Spectrum Scale in a private cloud, but the operator should do so with their "eyes wide open" to the potential issues.

High-speed access to data

Spectrum Scale is designed to operate over both Infiniband and Ethernet networks with Infiniband preferred for the highest speed, lowest latency access. Few public cloud providers have the ability to provide Infiniband interconnect into systems and only a handful of private cloud deployments have the necessary technology in place to make it work effectively. This means that access to Spectrum Scale is likely to fall-back to Ethernet only. This is likely to increase latency when accessing the storage, further to this, network configuration, for example MTU and contention/sharing of networks is likely to be of significant impact. For example, to access data at the highest speeds over Ethernet, an MTU approaching 9000 is required, however this requires jumbo frames to be enabled on hypervisors and all switches in the data path. In a private cloud deployment, this is within the control of the operator of the cloud but is an unknown quantity in public cloud deployments.

Additional speed can be gained by using SR-IOV, which enables a direct control path from a VM to a network device (either Ethernet or Infiniband) reducing the overhead of para-virtualisation. This technology is currently known to only be available in a limited number of private cloud deployments and its use impacts some of the flexibility of cloud deployment (e.g. restricting the ability to migrate VMs). Note that the Spectrum Scale is not the limiting factor, but the use of SR-IOV. SR-IOV has other potential uses in HPC cloud, for example it would enable the use of Infiniband for MPI applications, however this is beyond the scope of this report.

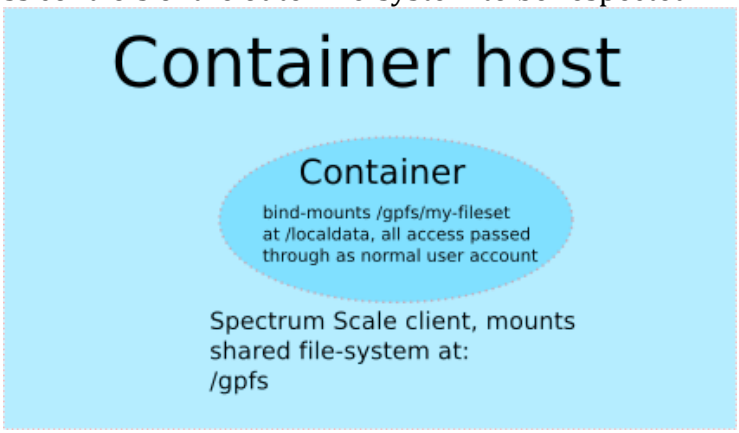
Containers and Spectrum Scale

There is currently limited direct support for the use of Spectrum Scale with containers, for example a direct Docker driver for Spectrum Scale file-systems. Using a file-path or NFS driver is possible with both Docker and K8s to bind-mount a file-path into a container. Where the hosting systems is running Spectrum Scale client to mount a file-system, it is fully supported to bind-mount all or part of the file-system into a container as the access mechanism is just normal POSIX semantics.

IBM are currently developing a volume service which will integrate with the Docker volume API and K8s dynamic provisioning which will allow users (where permitted by an admin) to manage data volumes in Spectrum Scale using the Docker/K8s command line tools.

Care needs to be taken when enabling users to manage their own containers and how access to file-path mounting is undertaken to restrict a user's ability to mount only their own files. The security and capabilities for managing this relate to the container solution used with different systems having capabilities to support this.

Many of the multi-tenancy questions regarding data access may be overcome where container orchestration is managed by an admin. For example if the container solution is able to effectively map the container access to outside the container as a specific user, it would allow users bringing containers and mapping data to have "full control" inside the container, but that all file-system access is mapped to their external user though allowing the access controls of the outer file-system to be respected.



Container created by user mounts Spectrum Scale file-system, in-container access is mapped back to normal user account to container host to provide data access authorisation

Data replication between sites

Spectrum Scale includes a feature know as Active File Management (AFM[9]), this feature is specifically designed to enable data access between Spectrum Scale clusters (typically between multiple-sites) which have a reliable network connection with adequate bandwidth.

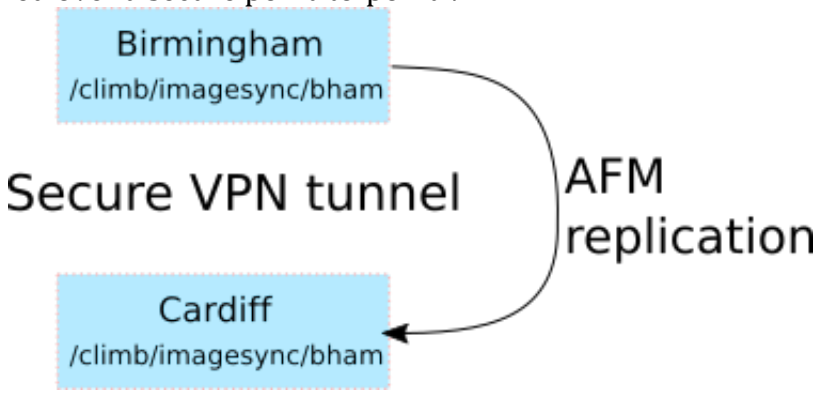
AFM provides a facility such that a local Spectrum Scale cluster is able to have a view and local cache of a remote file-system. This works most efficiently when the source is also a Spectrum Scale cluster, but can function when the source is an NFS server. AFM provides a view of the remote file-system (e.g. list files and directories) and when a file is accessed, it is automatically transferred using a reliable mechanism and cached at the local site. Files can also be pre-cached and policy settings are used to evict files from the cache. Depending on the mode of use, the cached view of the files can be either read-only or read-write when local updates are replicated back to the remote source.

---

9 https://www.ibm.com/support/knowledgecenter/en//STXKQY_4.2.0/com.ibm.spectrum.scale.v4r2.adv.doc/bl1adv_afm.htm AFM documentation

This feature may be of particular use in private cloud deployments as it would enable local caching of remote data, for example data held in a remote repository could be accessed and processed as a local cache with the updates being replicated back to the master at some point. This type of use case might also be useful for cloud-bursting of workloads where a Spectrum Scale deployment could be used in a public or private cloud to enable transparent data access and movement.

In addition to this, the CLIMB project[10] is currently piloting the use of AFM to synchronise data between their four sites to enable images to be automatically transferred between sites using AFM for reliable data transfer. This is being integrated directly into the CLIMB web UI "Bryn" which will allow a user to click a button that will mark the VM image such that it should then be replicated between the sites. Some additional middleware is also used to notify the local CLIMB OpenStack environment that a new image has completed transfer and is now available for import into the site. Currently this is functioning between the Birmingham and Cardiff sites, with AFM data being transferred over a secure point-to-point VPN link.



Shared images are copied to sync folder at Birmingham, code at Cardiff watches for new files and imports into the local Glance repository

Note that for sites that have fast local links, then the use of multi-cluster may be more appropriate which provides synchronous direct access between Spectrum Scale clusters.

AFM currently does not support an NFSv4 source and it would also be useful if it were possible to pre-fetch data before the file is listed in meta-data. i.e. A file would not be visible to user 'ls' commands until it had been fetched.

When operated with Advanced Spectrum Scale licensing, AFM can also be used with async-DR to provide remote site DR access to a file system.

Ongoing work and development

University of Birmingham is currently working with DDN to identify areas in which the use of the Spectrum Scale driver for volumes can be affected by the underlying storage configuration (for example block size, use of flash, LROC, HAWC features of Spectrum Scale).
On-going discussion with IBM is also continuing as the cloud integration is being developed by the IBM teams.

[10] www.climb.ac.uk, MRC funded OpenStack deployment over 4 geographically dispersed sites

As a closed-source, commercial product, the direction of development is directed by IBM's development priorities and business drivers. The RCUK cloud working group can continue to engage with IBM development. Specifically allocating funds to work collaboratively with IBM may also be an enabling solution to develop Spectrum Scale for cloud usage.

Development work to identify how VM images can be shared and referenced between sites should be undertaken. For example a repository of known scientific images. There is the potential to use features such as AFM or integration with metadata management tools is significant.

Acknowledgements

Thanks for their assistance in contributing to this report go to Dean Hildebrand, IBM Research, John Lewars, Bill Owen and Ulf Troppens, IBM for their attendance at the UK based cloud workshop and their open and ongoing discussion on cloud deployment. Thanks also to Radoslaw Poplawski, CLIMB/University of Birmingham for his contributions.