|  |  |  |
| --- | --- | --- |
|  | Ceph | GlusterFS |
| iSCSI? | Yes | Yes |
| NFS/CIFS | Through a gateway | Nfs-ganesha looks best |
| Easy to install/config | No (some say Yes) | Yes |
| POSIX compliant | Yes | Yes |
| Architecture | Distributed | Decentralized |
| Naming | CRUSH | EHA |
| API | FUSE, mount REST | FUSE, mount |
| Fault detection | Fully connect. | Detected |
| System availability | High | High |
| Data availability | Replication | RAID-like |
| Placement Strategy | Auto | Manual |
| Replication | Sync. | Sync. |
| Cache consistency | Lock | No |
| Load balancing | Manual | Manual |

***Summary:***

Ceph features a more mature block interface and better integration with OpenStack and similar cloud projects, whereas Gluster has a more mature file system interface and better traditional web storage integration. However, each project has its own strategies for addressing file, object, and block storage scenarios, developed with the needs of their separate communities in mind.

The best technology option in a given situation depends on many factors, but among community deployments, users tend to opt for Ceph for OpenStack services that require block storage, such as Cinder or Nova instance storage. For file-based sevices, such as Manila, or for user or guest application data, we see more users choosing Gluster.

***Ceph:***

Pools of storage can have a read-only or write-back caching tier.

The physical location of data can be managed using CRUSH maps whereas snapshots can be handled entirely by the storage backend.

Ceph features native integration with KVM+QEMU.

Ceph is a distributed object store , called RADOS, that interfaces with an object store gateway, a block device or a file system

Ceph is integrated into the Linux kernel since 2.6.34 release

Ceph can provide three different storage interfaces: Posix (both at kernel level and using fuse), Block and Object storage

Several IaaS cloud platforms (i.e.: OpenStack, CloudStack) officially supports CEPH to provide Block storage solution

***Gluster:***

The GlusterFS architecture aggregates compute, storage, and I/O resources into a global namespace.

Each server and its attached commodity storage are considered a node. Capacity is scaled by adding additional nodes or adding additional storage to each node.

Performance is increased by deploying storage among more nodes and high availability is achieved by replicating data between nodes.

The GlusterFS hashing algorithm is distributed to all of the servers in the cluster managing file placement on each of its building blocks. There is no single server that manages metadata or the cluster.

Gluster features a modular design, using what it calls translators, to give additional options beyond simple file distribution. Translators extend the base functionally by offering the ability to easily change redundancy or stripe the data across the cluster.

With the recent addition of native support for Gluster’s libgfapi into KVM+QEMU, gluster backed block devices and alpha stage native integration into Apache CloudStack Gluster is making a compelling offering for virtual machine storage.

References:

<http://blog.appcore.com/distributed-file-systems-brceph-vs-gluster-vs-nutanix>

<https://indico.cern.ch/event/214784/session/6/contribution/332/material/slides/0.pdf>

<http://blog.gluster.org/2014/09/glusterfs-and-nfs-ganesha-integration/>

<http://community.redhat.com/ceph-faq/>