Contribution ID: **18** Type: **not specified**

Storage solutions for a production-level cloud infrastructure

Monday 17 November 2014 15:30 (20 minutes)

We have set-up an OpenStack-based cloud infrastructure in the framework of a publicly funded project, PRISMA, aimed at the implementation of a fully integrated PaaS+IaaS platform to provide services in the field of smart-government (e-health, e-government, etc.). The IaaS testbed currently consists of 18 compute nodes providing in total almost 600 cores, 3550 GB of RAM, 400 TB of storage (disks). Connectivity is ensured through 2 NICs, 1Gbit/s and 10Gbit/s. Both the backend (MySQL database and RabbitMq message broker) and the core services (nova, keystone, glance, neutron, etc.) have been configured in high-availability using HA clustering techniques. The full capacity available by 2015 will provide 2000 cores and 8 TB of RAM.
In this work we present the storage solutions that we are currently using as backend for our production cloud services.
Storage is one of the key components of the cloud stack and can be used both to host the running VMs ("ephemeral"storage), and to host persistent data such as the block devices used by the VMs or users'archived unstructured data, backups, virtual images, etc..
The storage-as-service is implemented in Openstack by the Block Storage project, Cinder, and the Object Storage project, Swift. Selecting the right software to manage the underlying backend storage for these services is very important and decisions can depend on many factors, not only merely technical, but also economic: in most cases they result from a trade-off between performance and costs.

Many operators use separate compute and storage hosts. We decided not to follow this mainstream trend aiming at the best cost-performance scenario: for us it makes sense to run compute and storage on the same machines since we want to be able to dedicate as many of our hosts as possible to running instances. Therefore, each compute node is configured with a significant amount of disk space and a distributed file system (GlusterFS and/or Ceph) ties the disks from each compute node into a single file-system.
In this case, the reliability and stability of the shared file-system is critical and defines the effort to maintain the compute hosts: tests have been performed to asses the stability of the shared file-systems changing the replica factor. For example, we observed that GlusterFS in replica 2 cannot be used in production because highly unstable even at moderate storage sizes.

Our experience can be useful for all those organizations that have specific constraints in the procurement of a compute cluster or need to deploy on pre-existing servers for which they have little or no control over their specifications. Moreover, the solution we propose is flexible enough, since it is always possible to add external storage when additional storage is required.

We currently use GlusterFS distributed file system for:
- storage of the running VMs enabling the live migration,
- storage of the virtual images (as primary Glance image store),
- implementation of one of the Cinder backends for block devices.

In particular, we have been using Cinder with LVM-iSCSI driver since Grizzly release when the GlusterFS driver for Cinder did not support advanced features like snapshots and clones, fundamental for our use-cases. In order to exploit GlusterFS advantages even using LVM driver, we created the Cinder volume groups on GlusterFS loopback devices.
Upgrading our infrastructure to Havana, we decided to enable Ceph as additional backend of Cinder in order to compare features, reliability and performances of the two solutions.
Our interest for Ceph derives also from the possibility to consolidate the infrastructure overall backend storage into a unified solution. To this aim, currently we are testing Ceph to run the Virtual Machines, both using RBD and Ceph-FS protocols, and to implement the object storage.

In order to test the scalability and performance of the deployed system using test cases which are derived from the typical pattern of storage utilization. The tools used for testing are standard software widely used for this purpose such as: iozone and/or dd for block storage and specific benchmarking tools like Cosbench, swift-bench and ssbench for the object storage. Using different tools for testing the file-system and comparing their results with the observation of the real test case, is also a good possibility for testing the reliability of the benchmarking tools.

Throughput tests have been planned and conducted on the two system configurations in order to understand the performance of both storage solutions and its impacts to applications aiming at achieving the better SLA and end-users experience.

Implementing our cloud platform, we focused also on providing transparent access to data using standardized protocols (both de-iure and de-facto standards). In particular, Amazon-compliant S3 and the CDMI (Cloud Data Management Interface) interfaces have been installed on top of the Swift Object Storage in order to promote interoperability also at PaaS/SaaS levels.

Data is important for businesses of all sizes. Therefore, one of the most common user requirement is the possibility to backup data in order to minimize their loss, stay compliant and preserve data integrity. Implementing this feature is particularly challenging when the users come from the public administrations and the scientific communities that produce huge quantities of heterogeneous data and/or can have strict constraints.
An interesting feature of the Swift Object Storage is the geographic replica that can be used in order to add a disaster-recovery feature to the set of data and services exposed by our infrastructure.
Also Ceph provides a similar feature: the geo-replication through RADOS gateway.
Therefore, we have installed and configured both a Swift global cluster and a Ceph federated cluster, distributed on three different geographic sites. Results of the performance tests conducted on both clusters are presented along with a description of the parameters tuning that has been performed for optimization. The different replication methods implemented in the two middlewares, Swift and Ceph, are compared in terms of network traffic bandwidth, cpu and memory consumption.

Another important aspect we are taking care of is the QoS (Quality of Service) support, i.e. the capability of providing different levels of storage service optimized wrt the user application profile. This can be achieved defining different tiers of storage and setting parameters like how many I/Os the storage can handle, what limit it should have on latency, what availability levels it should offer and so on.
Our final goal is also to set-up a (semi-)automated system that is able of self-optimising. Therefore we are exploring the cache tiering feature of Ceph, that handles the migration of data between the cache tier and the backing storage tier automatically. Results of these testing activities will be shown too in this presentation.

Author: Dr DONVITO, Giacinto (INFN-Bari)

Co-authors: ANTONACCI, Marica (INFN-Bari); VINCENZO, Spinoso (INFN-Bari)

Presenters: Dr DONVITO, Giacinto (INFN-Bari); ANTONACCI, Marica (INFN-Bari); VINCENZO, Spinoso (INFN-Bari)

Session Classification: Technology and research

Track Classification: Technology and research