Extension of the Data Lake to efficiently serve data to external compute resources providers

Introduction

The Data Lake Model developed in ESCAPE's WP2 federates different distributed storage systems via a high level Data Management layer: Rucio. The different storage endpoints that constitute the global Data Lake system are the Rucio Storage Elements (RSEs), each of them mapping to a storage endpoint offered by the resource provider.
Rucio enables file upload and download capabilities mapping the client-server interaction to the RSEs offering different transfer protocols (http/WebDAV, root, GridFTP, s3).
The Rucio system is a policy driven, rule based Data Management system enabling data lifecycle capabilities for the users and the experiments: file replication levels, placement rules and implementation of policies.
The goal of this study is to evaluate the integration of heterogeneous resources within the ESCAPE Data Lake and assess the flexibility of the Data Lake model and the system to address punctual resource integration using standard interfaces. In particular integration with commercial cloud resources from AWS and Google have been evaluated, including both storage & CPU via Swift/S3 protocol.
Part of this study has been performed with the support of the cloud bank project with fundings for AWS and Google Cloud Platform (GCP)

Proof Of Concept #1: Generation of CTA's monte carlo files & upload to the Data Lake

Introduction

The goal of this use case is to use cloud resources to simulate the generation of Cherenkov Telescope Array (CTA)'s monte carlo data, and upload generated data to ESCAPE's Data Lake.
One of the nice features of commercial cloud is the fact that you can use different hardware depending on your needs. For this test, we used an instance "t3.xlarge" from AWS. The t3 instance supports new Intel Advanced Vector Extensions 512 (AVX-512) instruction set which is heavily used to process CTA's data

Set up:

1. Install docker Daemon on the AWS t3 instance (not installed by default)
2. Download "cta-rucio-cmd" container in order to generate and upload CTA monteCarlo Data. The container can be downloaded from the following registry : **docker://gitlab-registry.in2p3.fr/cta-lapp/cta-rucio-cmd:latest**
3. Run the docker container to generate and upload files to the ESCAPE's Data Lake.

For example, using the following command, we generate and upload 25 CTA Monte Carlo files of 2GB each to the RSE “LAPP-DCACHE” hosted in LAPP :

docker run -it gitlab-registry.in2p3.fr/cta-lapp/cta-rucio-cmd:latest /home/user/cmd/CTA-Ingest-workFlow.sh 25 2G LAPP-DCACHE /srv/var/tmp cta.log DESY-DCACHE SARA-DCACHE

Results

Performance and Cost: 2TB of data has been generated and uploaded in 1 hour on 2021/03/03. Data egress cost refers to data transfer fees for moving data out. Commercial cloud providers do not publish clearly this cost (from AWS’s documentation : “Data transfer from AWS to the internet is charged per service, with rates specific to the originating Region "). According to our test the egress cost is 10 € / TB.

In conclusion, even if, technically speaking, this use case can be implemented using cloud resources, at the moment the egress costs measured during this exercise largely dominates the cost, making this solution not viable at scale"(see figure 1)

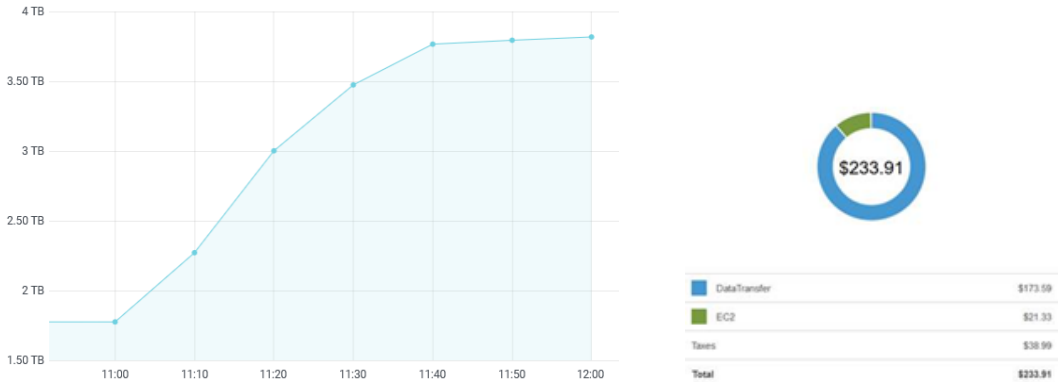


Fig.1: (left) 2 TB uploaded in 1 hour to the RSE “LAPP-DCACHE”, (right) egress cost for 2 TB

Proof Of Concept #2 : Use Commercial Cloud (CC) as an RSE for the ESCAPE Data Lake

Introduction

The goal of this use case is to federate storage resources hosted in commercial cloud with ESCAPE's Data Lake. This use case an be implemented in 2 ways :

- Object Cloud storage: Commercial cloud providers offer Object Storage capability (s3 for AWS, Google Cloud Storage for Google and Swift for openStack). Those storage offerings support different Quality of Service (QoS), From hot high IOPS SSDs to cold high latency, tape-like storage endpoints as "glacier" in AWS.Rucio has been capable of connecting to such endpoints since version 1.16.1. The QoS aspects have

not been tested in these exercises but this approach is considered interesting and worth exploring in future evaluations.

- Deploy a RSE in the cloud: This configuration is very similar to the configuration used in our data center. One slight difference is the fact that AWS or Google certificates are not trusted by the grid infrastructure. Therefore we need to generate a dedicated grid's certificate for the ssl communication with the RSE.

Moreover, in contrast to "Object Cloud Storage", there is only one QoS available for storage (SSD), this storage is named "Elastic Block Store" (EBS) in AWS.

Set up to "deploy an RSE in AWS":

1. Set up Domain Name
2. Activate "route53" service to map the domain name to the EC2 instance
3. Generated a grid certificate for the given Domain name
4. Install the software acting as a rse (apache with WebDav plugin) on the ec2 instance (in our case apache is package as a docker container using webDav as transfer protocol)
5. Create and map a storage capacity using the Elastic Block Store (EBS) service.
6. Configure the rse to use the EBS storage
7. additional configuration are needed to include this RSE in RUCIO (see rucio documentation)

Instance used : Elastic Compute Cloud (EC2) instance : t3.medium

Instance	**vCPU***	**CPU Credits/hour**	**Mem (GiB)**	**Storage**	**Network Performance (Gbps)*** **
t3.medium	2	24	4	EBS-Only	Up to 5

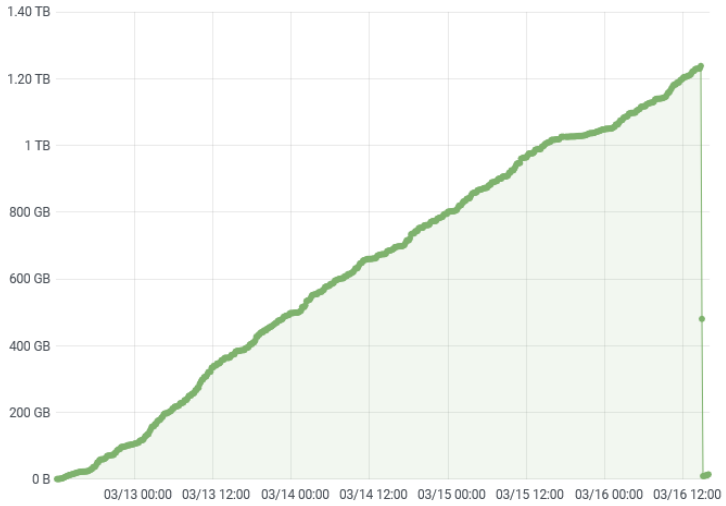
Result:

- Ingress cost is low and negligible when compared to EC2 cost. The main part of the cost is dedicated to the EBS storage (80 USD/TB-Month).
- In terms of performance, it really depends on the infrastructure set up in the cloud (CPU, network's bandwidth, ...).
- There is no limitation on RUCIO to integrate a RSE hosted in the cloud.

Conclusion

This study shows that we are able to seamlessly integrate commercial cloud storage resources in the ESCAPE Data Lake. Nevertheless the integration has associated costs, especially noticeable for egress operations. The overall cost calculation depends on several parameters and market fluctuations which makes it difficult to estimate in the long run, but

the proof of concept clearly indicated the integration is possible and the current Data Lake system is ready to provide this flexibility.
Moreover, the Root Certificate Authority from AWS and GCP (Amazon Root CA 1 & GTS Root R1) are not trusted by grid infrastructures. Therefore, it makes the integration more complex because we need to generate and install a grid certificate on the RSE's host.
As a final note, the integration with object cloud storage should be evaluated as there are several storage QoS available, and some of the QoS "cold/glacier" may decrease the storage cost. This might be of interest for some experiments in case this type of high latency archive-like storage cannot be hosted within the experiment's contributing sites.



1.2 TB of data replicated to the cloud's rse in 4 days