OpenStack in Production

Hints and tips from the CERN OpenStack cloud team

Wednesday, 21 February 2018

Maximizing resource utilization with Preemptible Instances

Motivation

The CERN cloud consists of around 8,500 hypervisors providing over 36,000 virtual machines. These provide the compute resources for both the laboratory's physics program but also for the organisation's administrative operations such as paying bills and reserving rooms at the hostel.

The resources themselves are generally ordered once to twice a year with servers being kept for around 5 years. Within the CERN budget, the resource planning teams looks at:

- The needs of the physics program for the coming years under the review of the Computing Scrutiny Review Board and the LHC Experiments Committee.
- The resources required to run the computing services requirements for the CERN laboratory. These are projected using capacity planning trend data and upcoming projects such as video conferencing.

With the installation and commissioning of thousands of servers concurrently (along with their associated decommissioning 5 years later), there are scenarios to exploit underutilised servers. Programs such as LHC@Home are used but we have also been interested to expand the cloud to provide virtual machine instances which can be rapidly terminated in the event of

- Resources being required for IT services as they scale out for events such as a large scale web cast on a popular topic or to provision instances for a new version of an application.
- Partially full hypervisors where the last remaining cores are not being requested (the Tetris problem).
- Compute servers at the end of their lifetime which are used to the full before being removed from the computer centre to make room for new deliveries which are more efficient and in warranty.

The characteristics of this workload is that it should be possible to stop an instance within a short time (a few minutes) compared to a traditional physics job.

Resource Management In Openstack

Operators use project quotas for ensuring the fair sharing of their infrastructure. The problem with this, is that quotas pose as hard limits.This leads to actually dedicating resources for workloads even if they are not used all the time or to situations where resources are not available even though there is quota still to use.

At the same time, the demand for cloud resources is increasing rapidly. Since there is no cloud with infinite capabilities, operators need a way to optimize the resource utilization before proceeding to the expansion of their infrastructure.

Resources in idle state can occur, showing lower cloud utilization than the full potential of the acquired equipment while the users' requirements are growing.

The concept of Preemptible Instances can be the solution to this problem. These type of servers can be spawned on top of the project's quota, making use of the underutilised capabilities. When the resources are requested by tasks with higher priority (such as approved quota), the preemptible instances are terminated to make space for the new VM.

Blog Archive

- ▼ 2018 (4)
  - ► March (2)
  - ▼ February (1)
    - Maximizing resource utilization with Preemptible I...
  - ► January (1)
- ► 2017 (7)
- ► 2016 (6)
- ► 2015 (17)
- ► 2014 (6)
- ► 2013 (6)

Contributors

- Arne Wiebalck
- Belmiro Moreira
- Dan van der Ster
- Jan van Eldik
- Jose Castro Leon
- Konstantinos Samaras-Tsakiris
- Ricardo Rocha
- Theodoros Tsioutsias
- Thomas Oulevey
- Tim Bell

Preemptible Instances with Openstack

Supporting preemptible instances, would mirror the AWS Spot Market and the Google Preemptible Instances. There are multiple things to be addressed here as part of an implementation with OpenStack, but the most important can be reduced to these:

1. Tagging Servers as Preemptible

In order to be able to distinguish between preemptible and non-preemptible servers, there is the need to tag the instances at creation time. This property should be immutable for the lifetime of the servers.

2. Who gets to use preemptible instances

There is also the need to limit which user/project is allowed to use preemptible instances. An operator should be able to choose which users are allowed to spawn this type of VMs.

3. Selecting servers to be terminated

Considering that the preemptible instances can be scattered across the different cells/availability zones/aggregates, there has to be "someone" able to find the existing instances, decide the way to free up the requested resources according to the operator's needs and, finally, terminate the appropriate VMs.

4. Quota on top of project's quota

In order to avoid possible misuse, there could to be a way to control the amount of preemptible resources that each user/project can use. This means that apart from the quota for the standard resource classes, there could be a way to enforce quotas on the preemptible resources too.

OPIE : IFCA and Indigo Dataclouds

In 2014, there were the first investigations into approaches by Alvaro Lopez from IFCA (https://blueprints.launchpad.net/nova/+spec/preemptible-instances). As part of the EU Indigo Datacloud project, this led to the development of the OpenStack Pre-Emptible Instances package (https://github.com/indigo-dc/opie). This was written up in a paper to Journal of Physics: Conference Series (http://iopscience.iop.org/article/10.1088/1742-6596/898/9/092010/pdf) and presented at the OpenStack summit (https://www.youtube.com/watch?v=eo5tQ1s9ZxM)

Prototype Reaper Service

At the OpenStack Forum during a recent OpenStack summit, a detailed discussion took place on how spot instances could be implemented without significant changes to Nova. The ideas were then followed up with the OpenStack Scientific Special Interest Group.

Trying to address the different aspects of the problem, we are currently prototyping a "Reaper" service. This service acts as an orchestrator for preemptible instances. It's sole purpose is to decide the way to free up the preemptible resources when they are requested for another task.

The reason for implementing this prototype, is mainly to help us identify possible changes that are needed in Nova codebase to support Preemptible Instances.

More on this WIP can be found here:
https://gitlab.cern.ch/ttsiouts/ReaperServicePrototype

Summary

The concept of Preemptible Instances gives operators the ability to provide a more "elastic" capacity. At the same time, it enables the handling of increased demand for resources, with the same infrastructure, by maximizing the cloud utilization.

This type of servers is perfect for tasks/apps that can be terminated at any time, enabling the users to take advantage of extra cpu power on demand without the fixed limits that quotas enforce.

Finally, here in CERN, there is an ongoing effort to provide a prototype orchestrator for Preemptible Servers with Openstack, in order to pinpoint the changes needed in Nova to support this feature optimally. This could also be available in future for other OpenStack clouds in use by CERN such as the T-Systems Open Telekom Cloud through the Helix Nebula Open Science Cloud project.

Contributors

- Theodoros Tsioutsias (CERN openlab fellow working on Huawei collaboration)
- Spyridon Trigazis (CERN)
- Belmiro Moreira (CERN)

References

- CERN Huawei openlab collaboration at https://openlab.cern/members/huawei and https://openlab.cern/project/openstack-clouds
- Helix Nebula Science Cloud project at http://www.helix-nebula.eu
- Indigo Datacloud at https://www.indigo-datacloud.eu/
- CERN SKA collaboration plans at https://www.openstack.org/videos/sydney-2017/future-science-on-future-openstack-developing-next-generation-infrastructure-at-cern-and-ska
- OpenStack Scientific Special Interest Group at https://wiki.openstack.org/wiki/Scientific_SIG
- Nova patch to support Reaper - https://review.openstack.org/547450

Posted by Theodoros Tsioutsias at 04:37 G+

No comments yet

Add a comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.