Science & Technology Facilities Council

www.stfc.ac.uk

A Quantative Evaluation of Different Methods For Instantiating Private Cloud Virtual Machines

Alexander Dibbo, Ian Collier, George Ryall, Andrew Lahiff, Frazer Barnsley
Alexander.dibbo@stfc.ac.uk

Background

At STFC RAL we have been developing a Cloud for a number of different use cases.

These include a self service VM on Demand service for staff in the Scientific Computing Department to use for accelerating development and testing, an Infrastructure-as-a-Service offering for a number of larger projects within the STFC and the larger community, and we are also testing bursting our Batch Farm into the cloud.

The cloud is based around OpenNebula with a Ceph RBD storage back-end.

Currently it is composed of 28 KVM Hypervisors (32 cores, 128GB RAM) with a total of 892 cores and 3.4TB of RAM, and 30 storage nodes (8 cores, 8GB RAM and 8x 4TB drives per node, one of which is used for the OS) for a raw capacity of ~750TB.

The storage is currently configured to hold 3 replicas of data.

We also have a VM for the OpenNebula Headnode, a 3 VM Galera/MariaDB cluster for the OpenNebula database and 3 VMs for Ceph Monitors.

What Are We Testing?

We are testing the performance of different configurations of virtual machines.

How easily we can perform management tasks (such as draining a hypervisor) against them.

And how quickly the machines can be deployed.

Why Are We Testing?

We want to know the performance characteristics of the various different configurations of virtual machines.

This is so that we can know if certain configurations give advantages to certain work-loads or use cases i.e. will a virtual worker node on local storage perform better.

We also want to know how agile we can be with the Virtual Machines in each configuration i.e. how quickly a hypervisor can be drained for patching.

Test Setup

We have two images being used for testing, our standard Quattor (Aquilon) managed image (10GB OS disk and 50GB Data disk) and the uCernVM images (20MB OS disk and 50GB Data disk).

All Virtual machines are configured with 1 core and 4GB of RAM. 4 storage configurations will be tested with each Virtual Machine image:

1. OS disk on Ceph, Data disk on Ceph
2. OS disk on Local Disk, Data disk on Local Disk
3. OS disk on Ceph, Data disk on Local Disk
4. OS Disk on Local Disk, Data disk on Ceph

3 Virtual Machines of each configuration will be tested spread throughout the cluster.

The cluster is currently very lightly used as it is still in the development stage.

How Are We Testing?

There are two stages to the testing performed here

Stage 1 – Read and Write Performance

These are only run against the Virtual Machines with our Image and are run 20 times each and the results averaged

- IOZone Single Threaded Test (Read, ReRead, Write, ReWrite)
  - Test file size of 6GB
  - Test file size of 24GB
- IOZone Agregate Test – 12 Threads of Mixed Read and Write equally split (Read, ReRead, Write, ReWrite)
  - Test file size of 0.5GB per Thread for 6GB total size
  - Test file size of 2GB per Thread for 24GB total size

Stage 2 – Usability and Agility

These are run against both the VMs with our image and the VMs based on the uCernVM:

1. Pending to Running – Once instantiation is initiated how long before the VM enters the RUNNING state i.e. it is deployed to a hypervisor
2. Running to Useable – Once the VM is RUNNING, how long until it is useable.
3. Pending to Useable – The aggregate of 1 and 2
4. Live Migration Time – How long does it take for the VM to migrate to another hypervisor.

Results

Stage 1 – Read and Write Performance

Single Stream Throughput *(Figure 1)*

- For these tasks Ceph has an advantage in Write performance
- Local disk has an advantage in Read performance

Multiple Stream (log scaled) *(Figure 2)*

- For small files (around 0.5 GB) there is significant caching of ReReads and ReWrites on both Ceph and Local disk.
- We can see that Ceph has a slight advantage here in terms of both read and write performance

Stage 2 - Usability and Agility

Our VM Image (monolithic) *(Figure 3)*

- Having the OS on a Ceph disk gives an advantage to deployment
- Once the image is deployed it then boots quicker from local disk. This makes sense as booting is mostly a single stream operation.
- The advantage goes to Ceph for Live Migration performance.

uCernVM Image *(Figure 4)*

The uCernVM is significantly faster both to deploy and boot. However the more important is the difference between the performance of each configuration.

Here we see an advantage to deployment time and live migration time by having both on Ceph however the best time to useable seems to come by having the OS on local disk and the Data in Ceph. This makes sense as you then get the high single stream read performance of local disk and the high single stream write performance of Ceph

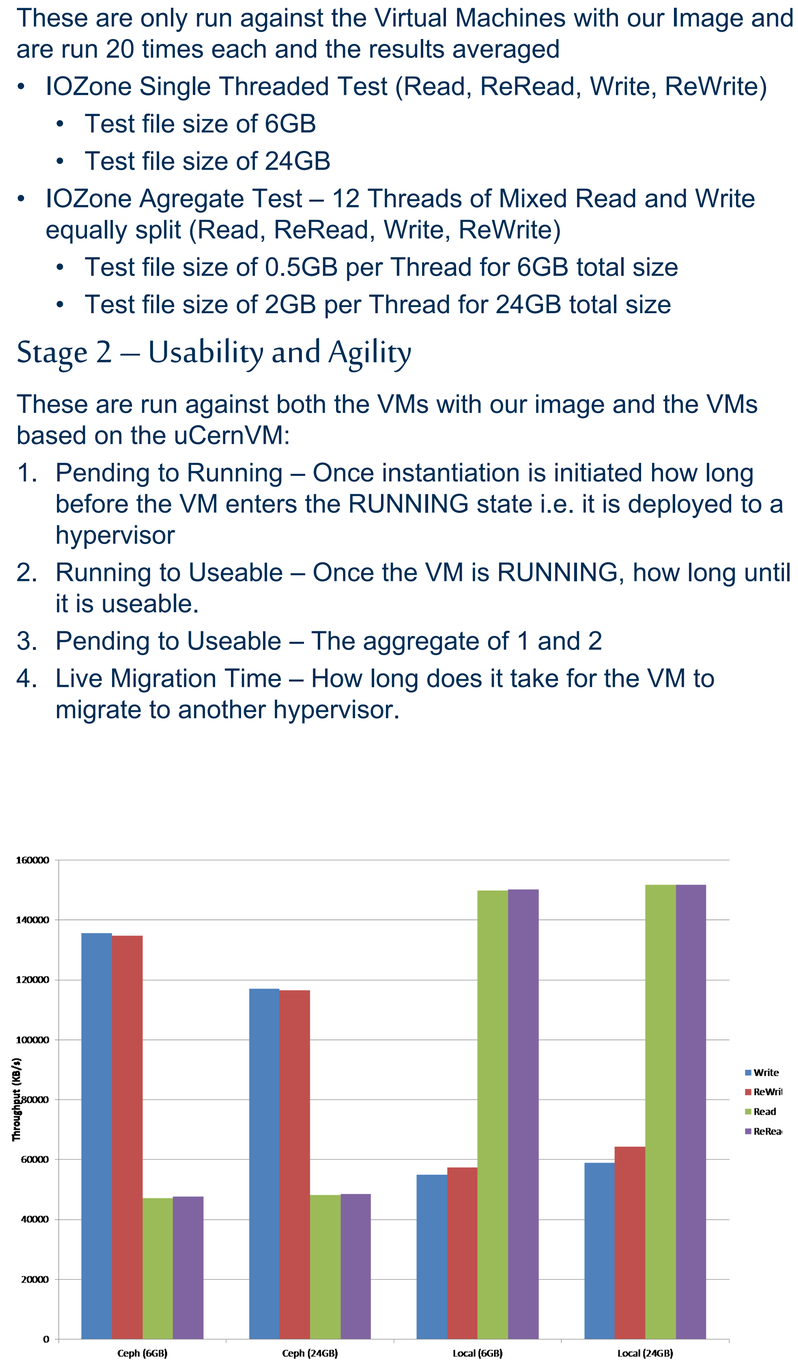

Figure 1: Single Stream Throughput (each column is an average of 120 test runs)

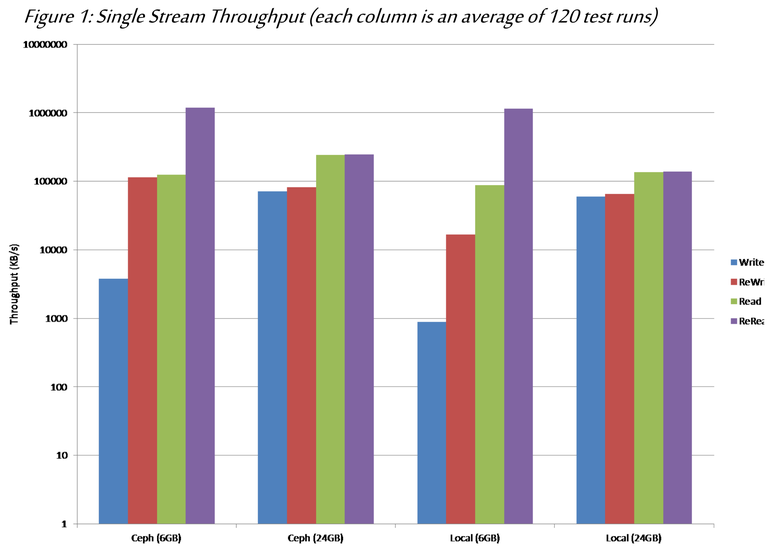

Figure 2: Multi Stream Throughput (each column is an average of 120 test runs)

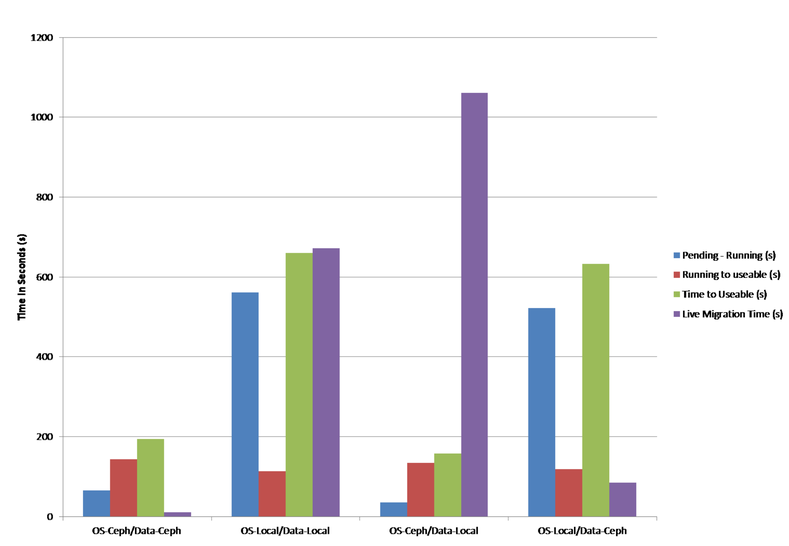

Figure 3: Monolithic image stats – Boot and Live Migration Times

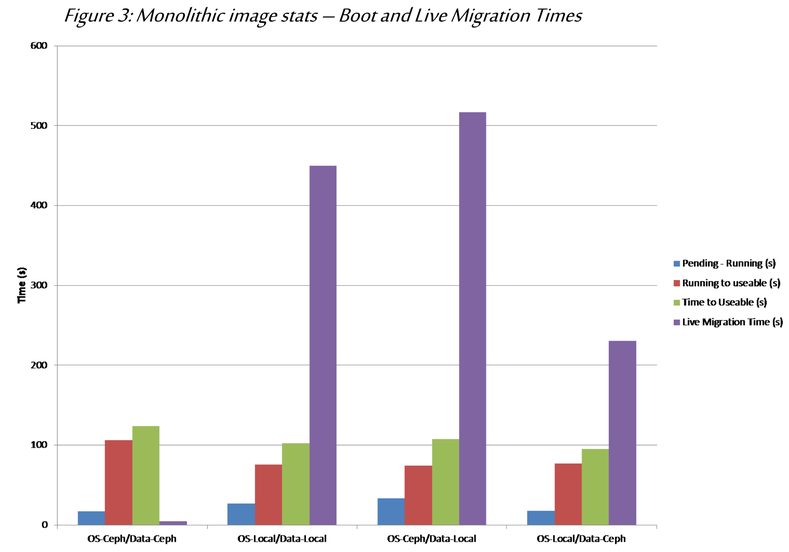

Figure 4: uCernVM image stats – Boot and Live Migration Times

Conclusions

Generally Ceph has significant advantages both in terms of performance and in how agile it allows you to be, the exception to this is of course the single stream read performance. This becomes quite an advantage when using micro kernel VMs if paired with a Ceph data disk.

For scenarios where the fastest boot time and good performance is required, such as a virtual worker node, a micro kernel VM with a Ceph data disk may be the best option.

Of course we should bear in mind that there are additional cost implications for running the Ceph cluster and that is constitutes a potential single point of failure. This is even taking into consideration the distributed nature of Ceph, there are some things which can bring a cluster to it's knees pretty rapidly.

Further Work

In order to further the work done here and to gain more insight into how to use the different configurations we would like to:

- Perform latency testing on all of the configurations here
- Examine the effects of KVM tuning on performance
- Look at pre-staging the image to hypervisors to see what kind of difference that makes.
- Test with different size data sets.
- Test when the hypervisors and/or shared storage system is heavily loaded.

GridPP UK Computing for Particle Physics

GridPP TIER 1