# OpenStack in Production

Hints and tips from the CERN OpenStack cloud team

Saturday, 30 April 2016

## Resource management at CERN

As part of the recent OpenStack summit in Austin, the Scientific Working group was established looking into how scientific organisations can best make use of OpenStack clouds.

During our discussions with more than 70 people (etherpad), we concluded on 4 top areas to look at and started to analyse the approaches and common needs. The areas were

1. Parallel file system support in Manila. There are a number of file systems supported by Manila but many High Performance Computing sites (HPC) use Lustre which is focussed on the needs of the HPC user community.
2. Bare metal management looking at how to deploy bare metal for the maximum performance within the OpenStack frameworks for identity, quota and networking. This team will work on understanding additional needs with the OpenStack Ironic project.
3. Accounting covering the wide range of needs to track usage of resources and showback/chargeback to the appropriate user communities.
4. Stories is addressing how we collect requirements from the scientific use cases and work with the OpenStack community teams, such as the Product working group, to include these into the development roadmaps along with defining reference architectures on how to cover common use cases such as high performance or high throughput computing clouds in the scientific domain.

Most of the applications run at CERN are high throughput, embarrassingly parallel applications. Simulation and analysis of the collisions such as in the LHC can be farmed to different compute resources with each event being handled independently and no need for fast interconnects. While the working group will cover all the areas (and some outside this list), our focus is on accounting (3).

Given limited time available, it was not possible for each of the interested members of the accounting team to explain their environment. This blog is intended to provide the details of the CERN cloud usage, the approach to resource management and some areas where OpenStack could provide additional function to improve the way we manage the accounting process. Within the Scientific Working group, these stories will be refined and reviewed to produce specifications and identify the potential communities who could start on the development.

### CERN Pledges

The CERN cloud provides computing resources for the Large Hadron Collider and other experiments. The cloud is currently around 160,000 cores in total spread across two data centres in Geneva and Budapest. Resources are managed world wide with the World Wide Computing LHC Grid which executes over 2 million jobs per day. Compute resources in the WLCG are allocated via a pledge model. Rather than providing direct funds from the sites, the sites commit to provide compute capacity and storage for a period of time as a pledge and these are recorded in the REBUS system. These are then made available using a variety of middleware technologies.

Given the allocation of resources across 100s of sites, the experiments then select the appropriate models to place their workloads at each site according to compute/storage/networking capabilities. Some sites will be suitable for simulation of collisions (high CPU, low storage and network). Others would provide archival storage and significant storage IOPS for more data intensive applications. For storage, the pledges are made in capacity on disk and tape. The compute resource capacity is pledges in Kilo-HepSpec06 units, abbreviated to kHS06 (based on a subset of the Spec 2006 benchmark) that allows faster processors to be given a higher weight in the pledge compared to slower ones (as High Energy Physics computing is an embarrassingly parallel high throughput computing problem).

The pledges are reviewed on a regular basis to check the requests are consistent with the



Contributors

- Ricardo Rocha
- Thomas Oulevey
- Jan van Eldik
- Jose Castro Leon
- Arne Wiebalck
- Marcos Lobo
- Tim Bell
- Daniel van der Ster
- Belmiro Moreira

experiments' computing models, the allocated resources are being used efficiently and the pledges are compatible with the requests.

Within the WLCG, CERN provides the Tier-0 resources for the safe keeping of the raw data and performs the first pass at reconstructing the raw data into meaningful information. The Tier-0 distributes the raw data and the reconstructed output to Tier 1s, and reprocesses data when the LHC is not running.

### Procurement Process

The purchases for the Tier-0 pledge for compute is translated into a formal procurement process. Given the annual orders exceed 750 KCHF, the process requires a formal procedure:

- A market survey to determine which companies in the CERN member states could reply to requests in general areas such as compute servers or disk storage. Typical criteria would be the size of the company, the level of certification with component vendors and offering products in the relevant area (such as industry standard servers)
- A tender which specifies the technical specifications and quantity for which an offer is required. These are adjudicated on the lowest cost compliant with specifications criteria. Cost in this case is defined as the cost of the material over 3 years including warranty, power, rack and network infrastructure needed. The quantity is specified in terms of kHS06 with 2GB/core and 20GB storage/core which means that the suppliers are free to try different combinations of top bin processors which may be a little more expensive or lower performing ones which would then require more total memory and storage. Equally, the choice of motherboard components has significant flexibility within the required features such as 19" rack compatible and enterprise quality drives. The typical configurations are white box manufacturers.
- Following testing of the proposed systems to ensure compliance, an order is placed with several suppliers, the machines manufactured, delivered, racked-up and burnt it using a set of high load stress tests to identify issues such as cooling or firmware problems.

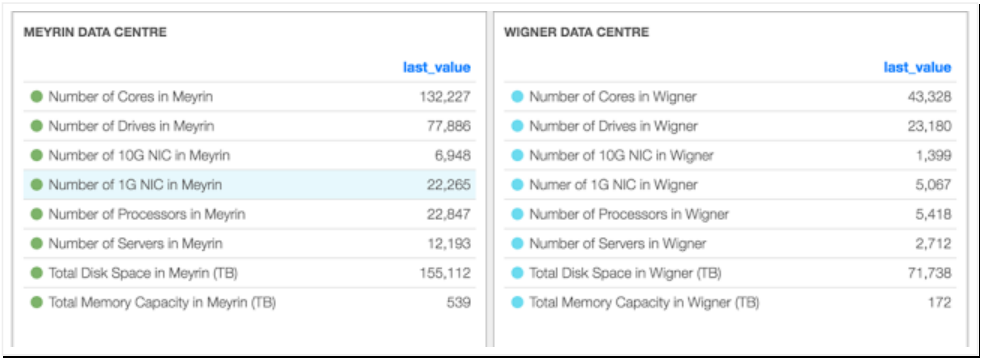
Typical volumes are around 2,000 servers a year in one or two rounds of procurement. The process from start to delivered capacity takes around 280 days so bulk purchases are needed followed by allocation to users rather than ordering on request. If there are issues found, this process can take significantly longer.

| Step | Time (Days) | Elapsed (Days) |
|---|---|---|
| User expresses requirement | | 0 |
| Market Survey prepared | 15 | 15 |
| Market Survey for possible vendors | 30 | 45 |
| Specifications prepared | 15 | 60 |
| Vendor responses | 30 | 90 |
| Test systems evaluated | 30 | 120 |
| Offers adjudicated | 10 | 130 |
| Finance committee | 30 | 160 |
| Hardware delivered | 90 | 250 |
| Burn in and acceptance | 30 days typical with 380 worst case | 280 |
| **Total** | | **280+ Days** |

Given the time the process takes, there are only one to two procurement processes run per year. This means that a continuous delivery model cannot be used and therefore there is a need for capacity planning on an annual basis and to find approaches to use the resources before they are allocated out to their final purpose.

### Physical Infrastructure

CERN manages two data centres in Meyrin, Geneva and Wigner, Budapest. The full data is available at the CERN data centre overview page. When hardware is procured, the final destination is defined as part of the order according to rack space, cooling and electrical availability.



MEYRIN DATA CENTRE

| | last_value |
|---|---|
| Number of Cores in Meyrin | 132,227 |
| Number of Drives in Meyrin | 77,886 |
| Number of 10G NIC in Meyrin | 6,948 |
| Number of 1G NIC in Meyrin | 22,265 |
| Number of Processors in Meyrin | 22,847 |
| Number of Servers in Meyrin | 12,193 |
| Total Disk Space in Meyrin (TB) | 155,112 |
| Total Memory Capacity in Meyrin (TB) | 539 |

WIGNER DATA CENTRE

| | last_value |
|---|---|
| Number of Cores in Wigner | 43,328 |
| Number of Drives in Wigner | 23,180 |
| Number of 10G NIC in Wigner | 1,399 |
| Numer of 1G NIC in Wigner | 5,067 |
| Number of Processors in Wigner | 5,418 |
| Number of Servers in Wigner | 2,712 |
| Total Disk Space in Wigner (TB) | 71,738 |
| Total Memory Capacity in Wigner (TB) | 172 |

While the installations in Budapest are new installations, some of the Geneva installations involve replacing old hardware. We typically retire hardware between 4 and 5 years old when the CPU power/watt is significantly better with new purchases and the hardware repair costs for new equipment is more predictable and sustainable.

Within the Geneva centre, there are two significant areas, physics and redundant power. Physics power has a single power source which is expected to fail in the event of an electricity cut lasting beyond the few minutes supported by the battery units. The redundant power area is backed by diesels. The Wigner centre is entirely redundant.

## Lifecycle

With an annual procurement cycle with 2-3 vendors per cycle, each one with their own optimisations to arrive at the lowest cost for the specifications, the hardware is highly heterogeneous. This has a significant benefit when there are issues, such as disk firmware or BMC controllers, that lead to delays in one of the deliveries being accepted, so the remaining hardware can be made available to experiments.

However, we run the machines for the 3 year warranty and then some additional years on minimal repairs (i.e. simple parts are replaced with components from servers of the same series), we have around 15-20 different hardware configurations for compute servers active in the centre at any time. There are variations in the specifications (as technologies such as SSDs and 10Gb Ethernet became commodity, the new tenders needed these) and those between vendor responses for the same specifications (e.g. slower memory or different processor models).

These combinations do mean that offering standard flavors for each hardware complication would be very confusing for the users, given that there is no easy way for a user to know if resources are available in a particular flavor except to try to create a VM with that flavor.

Given new hardware deliveries and limited space, there are equivalent retirement campaigns. The aim is to replace the older hardware by more efficient newer boxes that can deliver more HS06 within the same power/cooling envelope. The process to empty machines depends on the workloads running on the servers. Batch workloads generally finish within a couple of weeks so setting the servers to no longer accept new work just before the retirements is sufficient. For servers and personal build/test machines, we aim to migrate the workloads to capacity on new servers. This operation is increasingly being performed using live migration and MPLS to extend the broadcast domains for networks to the new capacity.

## Projects and Quota

All new users are allocated a project, “Personal XXXX” where XXXX is their CERN account when they subscribe to the CERN cloud service through the CERN resource portal. The CERN resource portal is the entry point to subscribe to the many services available from the central IT department and for users to list their currently active subscriptions and allocations. The personal projects have a minimal quota for a few cores and GBs of block storage so that users can easily follow the tutorial steps on using the cloud and create simple VMs for their own needs. The default image set is available on personal projects along with the standard ‘m’ flavors which are similar to the ones on AWS.

Shared projects can also be requested for activities which are related to an experiment or department. For these resources, a list of people can be defined as administrators (through CERN’s group management system e-groups) and a quota for cores, memory and disk space requested. Additional flavors can also be asked for according to particular needs such as the CERNVM flavors with a small system disk and a large ephemeral one.

The project requests go through a manual approval process, being reviewed by the IT resource management to check the request against the pledge and the available resources. An adjustment of the share of the central batch farm is also made so that the sum of resources for an experiment continues to be within the pledge.

Once the resource request has been manually approved, the ticket is then passed to the cloud team for execution. Rundeck provides us with a simple tool for performing high privilege operations with good logging and recovery. This is used in many of our workflows such as hardware repair. The Rundeck procedure reads the quota settings from the ticket and executes the appropriate project creation and role allocation requests to Keystone, Nova and Cinder.

## Need #1 : CPU performance based allocation and scheduling

(As with all requests, there is a mixture of requirements and implementation. The Needs are stated in an absolute fashion according to our current understanding. There may be alternative approaches or compromises which would address these needs in common with other user requirements).

The resource manager for the cloud in a high throughput/performance computing environment allocates resources based on performance rather than pure core count.

A user wants to request a processor of a particular minimum performance and is willing to have a larger reduction in his remaining quota.

A user wants to make a request for resources and then iterate according to how much performance related quota they have left to fill the available quota, e.g. give me a VM with less than a certain performance rating.

A resource manager would like to encourage the use of older resources which are often idle.

A quotas for slower and faster cores is currently the same (thus users create a VM, delete it if it is one of the slower type) so there is no incentive to use the slower cores.

As a resource manager preparing an accounting report, the faster cores should have a higher weight against the pledge to ensure continued treatment of slower cores.

The proposal is therefore to have an additional, optional quota on CPU units so that resource managers can allocate out total throughput rather than per core capacities.

The alternative approach of defining a number of flavors for each of the hardware types and quota. However, there are a number of drawbacks with this approach:

- The number of flavors to define would be significant (in CERN's case, around 15-20 different hardware configurations multiplied by 4-5 sizes for small, medium, large, xlarge, ...)
- The user experience impact would be significant as the user would have to iterate over the available flavors to find free capacity. For example, trying out m4.large first and finding the capacity was all used, then trying m3.large etc.

There is, as far as I know, no per-flavor quota. The specs have been discussed in some operator feedback sessions but the specification did not reach consensus.

Extendible resource tracking seems to be approaching the direction with 'compute units' (such as here) as defined in the specification. Many parts of this are not yet implemented so it is not easy to see if it addresses the requirements.

## Need #2 : Nested Quotas

As an experiment resource co-ordinator, it should be possible to re-allocate resources according to the priorities of the experiment without needing action by the administrators. Thus, moving quotas between projects which are managed by the experiment resource co-ordinators within the pledge allocated by the WLCG.

Nested keystone projects have been in the production release since Kilo. This gives the possibility for role definitions within the nested project structure.

The implementation of the nested quota function has been discussed within various summits for the past 3 years. The first implementation proposal, Boson, was for a dedicated service for quota management. However, there were concerns raised by the PTLs on the impacts for performance and maintainability of this approach. The alternative of enhancing the quotas in each of the projects has been followed (such as Nova). These implementations though have not advanced due to other concerns with quota management which are leading towards a common library, delimiter, which is being discussed for Newton.

## Need #3 : Spot Market

As a cloud provider, uncommitted resources should be made available at a lower cost but at a lower service level, such as pre-emption and termination at short notice. This mirrors the AWS spot market or the Google Pre-emptible instances. The benefits would be higher utilization of the resources and ability to provide elastic capacity for reserved instances by

reducing the spot resources.

A draft specification for this functionality has been submitted along with the proposal which is currently being reviewed. An initial implementation of this functionality (called OpenStack Preemptible Instances Extension, or opie) will be made available soon on github.

### Need #4 : Reducing quota below utilization

As an experiment resource co-ordinator, quotas are under regular adjustment to meet the chosen priorities. Where a project has a lower priority but high current utilization, further resource usage should be blocked but existing resources not deleted. The resource co-ordinator can then contact the user to encourage the appropriate resources to be deleted.

To achieve this function, one approach would be to allow the quota to be set below the current utilization in order to give the project administrator the time to identify the resources which would be best to be deleted in view of the reduced capacity.

### Need #5 : Components without quota

As a cloud provider, some inventive users are storing significant quantities of data in the Glance image service. There is only a maximum size limit with no accumulated capacity leaves this service open to non-planned storage.

It is proposed to add quota functionality inside Glance for

- The total capacity of images stored in Glance
- The total capacity of snapshots stored in Glance

The number of images and snapshots would be an lower priority enhancement request since the service risk comes from the total capacity although the numbers could potentially also be abused.

Given that this is new functionality, it could also be a candidate for the first usage of the new delimiter library.

#### Acknowledgements

There are too many people who have been involved in the resource management activities to list here. The teams contributing to the description above are:

- CERN IT teams supporting cloud, batch, accounting and quota
- BARC, Mumbai for collaborating around the implementation of nested quota
- Indigo Datacloud team for work on the spot market in OpenStack
- Other labs in the WLCG and the Scientific Working Group

Posted by Tim Bell at 06:15 1 comment



Thursday, 21 April 2016

## Containers and the CERN cloud

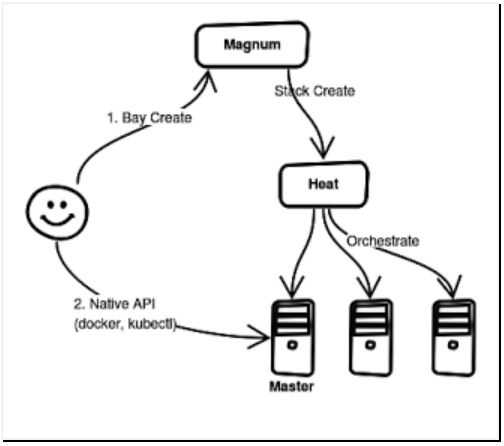
In recent years, different groups at CERN started looking at using containers for different purposes, covering infrastructure services but also end user applications. These efforts have been mostly done independently, resulting in a lot of repeated work especially for the parts which are CERN specific: integration with the identity service, networking and storage systems. In many cases, the projects could not complete before reaching a usable state, as some of these tasks require significant expertise and time to be done right. Alternatively, they found different solutions to the same problem which led to further complexity for the supporting infrastructure services. However, the use cases were real, and a lot of knowledge had been built on the available tools and their capabilities.

Based on this, we started a project with the following goals:

- integrate containers into the CERN OpenStack cloud, building on top of already available tools such as resource lifecycle, quotas, identity and authorization
- stay container orchestration agnostic, allowing users to select any of the most common solutions (Docker Swarm, Kubernetes, Mesos)
- allow fast cluster deployment and rebuild

We had done a prototype using the Nova LXC driver in the past but the long term support was not clear and we wanted access to the native container functions using the standard tools.

Looking for other possibilities, OpenStack Magnum seemed to be offering a lot of what we needed, and we decided to try it out. At around the same time we were also heading to the OpenStack Tokyo summit, which was a great opportunity to follow the Magnum sessions and learn more of what it provides.




Magnum relies heavily on Heat for the orchestration part of the container clusters - called **bays**. Bays are instantiated based on pre-defined **bay models**, which set how the master and the other nodes should look like (flavor, image, etc) and which container orchestration engine (COE) should be used - among other possible configuration options. Current choices include Docker Swarm, Kubernetes and Mesos. The Magnum homepage gives a lot more details.

At the beginning of November 2015, we started investigating the Magnum project in depth. At that time the project was functional but some of its requirements posed problems in our deployment:

- Dependency on OpenStack Neutron, something we had not yet deployed (we have nova-network since we started the cloud in 2012). Luckily we were working on it in parallel, and we got a functional control plane just in time. And as we use Nova Cells, we could enable Neutron in a dedicated cell where we would also enable Magnum, reusing the rest of the production infrastructure
- Requirement on Neutron LBaaS, which we don't have. This is something we plan to try, but it is not obvious how to implement this currently due to the way the CERN network is structured. We made some changes to the Heat templates to remove this requirement

The other pre-requisite projects, such as Keystone, Glance and Heat were already in production in the CERN cloud.

But no real show stoppers and very quickly we got a prototype deployment. For a more detailed evaluation we initially chose 3 internal projects that cover the most common use cases:

- GitLab CI, a continuous integration service we use internally - it has integration with Docker, making it a perfect example of how to use a Docker Swarm cluster as a drop in replacement for a local Docker daemon
- Infrastructure services, namely one of the critical services for the data movement between the multiple sites of the LHC Computing Grid (WLCG) - for a nice example of scaling a service by scaling its individual components
- Jupyter Notebooks - a growing trend for end user analysis in different scientific communities, providing a browser based interactive session running in a remote container

In addition, we are also working with the European Union Horizon 2020 project Indigo Datacloud which is developing an open source data and computing platform targeted at scientific communities, deployable on multiple hardware and provisioned over hybrid, private or public, e-infrastructures. Using Magnum, we can provide the test resources for this project to the partners.

For our users and resource managers, there are significant advantages of the Magnum approach:

- Native tools - anything that works with Docker will work talking to a Docker

Swarm COE or kubectl with a Kubernetes cluster. This allows smaller physics sites to provide native Docker or Kubernetes while the larger sites provide Containers-as-a-Service on-demand. User applications written to work with Docker or Kubernetes can be used without modification against the provisioned resources.

- Container engine agnostic - with our user community, there is a strong need for flexibility to allow different avenues to be explored. Magnum allows the IT department to offer Kubernetes, Docker Swarm and Mesos at a low cost within the same service at an affordable load for the support team. The users can then prototype different application approaches and select the best combination for them. Enforcing a central IT service decision on the end user community is never easy, especially where there are diverse user requirements being covered within a central cloud.
- Accounting, quota and permissions remains within the existing framework. Thus, whether resources are used for containers or VMs is a choice for the project user. Capacity planning can be done by cores/RAM rather than segmentation of resources for container or VM resources. Access controls follow the existing admin/member structures for projects.
- Elasticity - within the quota limits, containers can scale, with new bays as needed within the quota. This allows the resources to be allocated where there is a user need (and as importantly, shrunk when things are quiet)
- Repairs - failures in the infrastructure (software or hardware) are looked after by the cloud support team. For the user, the workloads can be scheduled elsewhere. For the hardware repair teams, the operations can be performed in a consistent fashion in bulk rather than on a one-by-one basis. Infrastructure monitoring procedures are the same for VMs and containers.
- The operating system support teams can provide reference images and follow up issues with the upstream providers. They can be confident that the image is based on supported configurations rather than ad-hoc builds. Rebuilding base images with the appropriate security patches can sometimes be delayed, raising the risk of incidents.

By the end of March, we had the use cases covered, and the few hick-ups covered in blueprints or patches upstream, and had contributed for the missing bits in puppet and documentation. And with a service running on our production resources and thanks to keystone endpoint filtering, we could increase service usage by enabling it for individual projects. Today we have around 15 different projects using Magnum as a pilot service and the number keeps growing.

In just a few months, we got Magnum up and running and it has proved to be a significant addition to the OpenStack cloud. Which makes us excited about what is coming next, including:

- Integration with Cinder - ready upstream, and we'll be trying it very soon
- Magnum benchmarks in Rally - we rely on Rally to make sure our cloud is performing as expected
- Further integration with our local storage systems such as CVMFS and EOS - relying on the ability to add site specific configurations to the bay templates
- Integration with Barbican - the recommended way to handle the required TLS certificates to talk to the native APIs of the orchestration engines, and the only option today to get Magnum in HA (though that's about to change)
- Integration with Horizon - this will help as we expand the service into production to communities who are used to using the web interfaces

If you're interested in more details on the available container orchestration technologies or our usage of OpenStack Magnum, or simply want to see some fancy demos, check our recent presentation at a CERN Technical Forum.

### Acknowledgments


- Mathieu Velten for his work on testing and adapting Magnum at CERN and contributions to Indigo DataCloud
- Bertrand Noel for all his time spent researching existing container technologies
- Spyros Trigazis, a fellow in the CERN OpenLab collaboration with Rackspace, for all his work upstream both for features and documentation improvements
- Jarek Polok for the CERN docker repository
- The OpenStack Magnum team for their support and collaboration
- All CERN users that helped us debug and set the service requirements


### References


- Presentation on containers at the CERN Technical Forum - https://cds.cern.ch/record/2144886
- End user documentation at http://clouddocs.web.cern.ch/clouddocs/containers/index.html
- OpenStack superuser article at http://superuser.openstack.org/articles/openstack-magnum-on-the-cern-production-cloud

Posted by Ricardo Rocha at 10:29 0 comments

G+1 Recommend this on Google

Tuesday, 19 April 2016

# Deploying the new OpenStack EC2 API project

OpenStack has supported a subset of the EC2 API since the start of the project. This was originally built in to Nova directly. At CERN, we use this for a number of use cases where the experiments are running across both the on-premise and AWS clouds and would like a consistent API. A typical example of this is the HTCondor batch system which can instantiate new workers according to demand in the queue on the target cloud.

With the Kilo release, this function was deprecated and has been removed in Mitaka. The functionality is now provided by the new ec2-api project which uses the public Nova APIs to provide an EC2 compatible interface.

Given that CERN has the goal to upgrade to the latest OpenStack release in the production cloud before the next release is available, a migration to the ec2-api project was required before the deployment of Mitaka, due to be deployed at CERN in 2H 2016.

The EC2 API project was easy to set up using the underlying information from Nova and a small database which is used to store some EC2 specific information such as tags.

As described in Subbu's blog, there are many parts needed before for an OpenStack API to become a service. By deploying using the CERN cloud, many aspects on identity, capacity planning, log handling, onboarding are covered by the existing infrastructure.


From the CERN perspective, the key functions we need in addition to the code are

- Packaging - we work with the RDO distribution and the OpenStack RPM-Packaging project to produce a package for installation on our CentOS 7 controllers.
- Configuration - Puppet provides us the configuration management for the CERN cloud. We are currently merging the CERN Puppet EC2 API modules to the puppet-ec2api project. The initial patch is now in review.
- Monitoring - each new project has a set of daemons to make sure are running smoothly. These have to be integrated into the site monitoring system.
- Performance - we use the OpenStack Rally project to continuously run functionality and performance tests, simulating a user. The EC2 support has been added in this review.

The current steps are the end user testing and migration from the current service. Given that the ec2-api project can be run on a different port, the two services can be run in parallel for testing. Horizon would need to be modified to change the EC2 endpoint in the ec2rc.sh (which is downloaded from Compute->Account & Security->API Access).

So far, the tests have been positive and further validation will be performed over the next few months to make sure that the migration has completed so there is no impact on the Mitaka upgrade.

### Acknowledgements

- Wataru Takase (KEK) for his work on Rally
- Marcos Fermin Lobo (CERN/Oviedo) for the packaging and configuration

- Belmiro Moreira (CERN) for the necessary local CERN customisations in Nova
- The folks from Cloudscaling/EMC for their implementation and support of the OpenStack EC2 API project

### References

- CERN enduser documentation for EC2 at http://clouddocs.web.cern.ch/clouddocs/ec2tutorial/index.html

Posted by Tim Bell at 03:35 2 comments



Sunday, 29 November 2015

## Our cloud in Kilo

Following on from previous upgrades, CERN migrated the OpenStack cloud to Kilo during September to November. Along with the bug fixes, we are planning on exploiting the significant number of new features, especially as related to performance tuning. The overall cloud architecture was covered at the Tokyo OpenStack summit video https://www.openstack.org/summit/tokyo-2015/videos/presentation/unveiling-cern-cloud-architecture.

As the LHC continues to run 24x7, these upgrades were done while the cloud was running and virtual machines were untouched.

Previous upgrades have been described as below

- Juno - http://openstack-in-production.blogspot.fr/2015/05/our-cloud-in-juno.html
- Icehouse - http://openstack-in-production.blogspot.fr/2014/11/our-cloud-in-icehouse.html
- Havana - http://openstack-in-production.blogspot.fr/2014/02/our-cloud-in-havana.html

The staged approach was used again. While most of the steps went smoothly, a few problems were encountered.

- Cinder - we encountered the bug https://bugs.launchpad.net/cinder/+bug/1455726 which led to a foreign key error. The cause appears to be related to UTF8. The patch (https://review.openstack.org/#/c/183814/) was not completed so did not get included into the release. More details at the thread at http://lists.openstack.org/pipermail/openstack/2015-August/013601.html.
- Keystone - one of the configuration parameters for caches had changed syntax and this was not reflected in the configuration generated by Puppet. The symptoms were high load on the Keystone servers since caching was not enabled.
- Glance - given the rolling upgrade on Glance, we took advantage of having virtualised the majority of the Glance server pool. This allows new resources to be brought online with a Juno configuration and the old ones deleted.
- Nova - we upgraded the control plane services along with the QA compute nodes. With the versioned objects, we could stage the migration of the thousands of compute nodes so that we did not need to do all the updates at once. Puppet looked after the appropriate deployments of the RPMs.
    - Following the upgrade, we had an outage of the metadata service for the OpenStack specific metadata. The EC2 metadata works fine. This is a cells related issue and we'll create a bug/blueprint for the fix.
    - The VM resize functions are giving errors during the execution. We're tracking this with the upstream developers.
        - https://bugs.launchpad.net/nova/+bug/1459758
        - https://bugs.launchpad.net/nova/+bug/1446082
    - We wanted to use the latest Nova NUMA features. We encountered a problem with cells and this feature, although it worked well in a non-cells cloud. This is being tracked in https://bugs.launchpad.net/nova/+bug/1517006. We will use the new features for performance optimisation once these problems are resolved.
    - The dynamic migration of flavors was only partially successful. With the cells database having the flavors data in two places, the

migration needed to be done simultaneously. We resolved this by forcing the migration of the flavors to the new endpoint,

- The handling of ephemeral drives in Kilo seems to be different from Juno. The option `default_ephemeral_format` now defaults to vfat, rather than ext3. The aim seems to have been to give vfat to Windows and ext4 to Linux but our environment does not follow this. This was reported by Nectar but we could not find any migration advice in the Kilo release notes. We have set the default to ext3 while we are working out the migration implications.
- We're also working through a scaling problem for our most dynamic cells at https://bugs.launchpad.net/nova/+bug/1524114. Here all VMs are being queried by the scheduler, not just the active ones. Since we create/delete hundreds of VMs an hour, there are large volumes of deleted VMs which made one query take longer than expected.

Catching these cases with cells early is part of the work for the scope of the the Cell V2 project at https://wiki.openstack.org/wiki/Nova-Cells-v2 to which we are contributing along with the BARC centre in Mumbai so that the cells configuration becomes the default (with only a single cell) and the upstream test cases are enhanced to validate the multi cell configuration.

As some of the hypervisors are still running Scientific Linux 6, we used the approach from GoDaddy to package the components using software collections. Details are available at https://github.com/krislindgren/openstack-venv-cent6. We used this for nova and ceilometer which are the agents installed on the hypervisors. The controllers were upgraded to CentOS 7 as part of the upgrade to Kilo.

Overall, getting to Kilo enables new features and includes bug fixes to reduce administration effort. Keeping up with new releases requires careful planning and sharing upstream activities such as the Puppet modules but has proven to be the best approach. With many of the CERN OpenStack team in the summit in Tokyo, we did not complete the upgrade before Liberty was released but this has been completed soon afterwards.

With the Kilo base in production, we are now ready to start work on the Nova network to Neutron migration, deployment of the new EC2 API project and enabling Magnum for container native applications.

Posted by Tim Bell at 07:46 6 comments



Thursday, 8 October 2015

## Scheduling and disabling Cells

In order to scale OpenStack Cloud Infrastructure at CERN, we were early to embrace an architecture that uses Cells. Cells is a Nova functionality that allows the partition a Cloud Infrastructure into smaller groups with independent control planes.

For large deployments Cells have several advantages like:

- single endpoint to users;
- increase the availability and resilience of the Infrastructure;
- avoid that Nova and external components (DBs, message brokers) reach their limits;
- isolate different user cases;

However, cells also have some limitations. There are some nova features that don't work when running cells:

- Security Groups;
- Manage aggregates on Top Cell;
- Availability Zone support;
- Server groups;
- Cell scheduler limited functionality;

There has been many changes since we deployed our initial cells configuration two years ago. During the past months ,there have been a lot of work involving Cells, especially make sure that they are properly tested and developing CellsV2 that should be the default way to deploy Nova in the future.

However, today when using Cells we continue to receive following welcome message :)

"The cells feature of Nova is considered experimental by the OpenStack project because it receives much less testing than the rest of Nova. This may change in the future, but current deployers should be aware that the use of it in production right now may be risky."

At CERN, we now have 26 children cells supporting the 130,000 cores across two data centres in a single cloud. Some cells are dedicated for the general use cases and others that are dedicated only to specific projects.

In order to map projects to cells we developed a scheduler filter for the cell scheduler.

https://github.com/cernops/nova/blob/cern-2014.2.2-2/nova/cells/filters/target_cell_project.py

The filter relies in two new values defined in nova.conf: "cells_default" and "cells_projects".

- "cells_default" contains the set of available cells to schedule instances if the project is not mapped to any specific cell;
- "cells_projects" contains the mapping cell -> project for the specific use cases;

```
"nova.conf"
cells_default=cellA,cellB,cellC,cellD
cells_projects=cellE:<project_uuid1>;<project_uuid2>,cellF:
<project_uuid3>
```

For example, when an instance belonging to "project_uuid2" is created, it's schedule to "cellE". But, if the instance belongs to "project_uuid4" it's schedule to one of the default cells ("cellA", "cellB", "cellC", "cellD").

One of the problems when using cells is that is not possible to disable them from the scheduler.

With this scheduler filter we can achieve this. To disable a cell we just need to remove it from the "cells_default" or" cells_projects" list. Disabling a cell means that it will not be possible to create new instances on it, however it is still available to perform operations like restart, resize, delete, ...

These experiences will be discussed in the upcoming summit in Tokyo with the deep dive into the CERN OpenStack deployment (https://mitakadesignsummit.sched.org/event/f929ea7ee625dadcc16888cb33984dad#.VhbHu3pCrWI), at the Ops meetup (https://etherpad.openstack.org/p/TYO-ops-meetup) and Nova design sessions (https://mitakadesignsummit.sched.org/overview/type/nova#.VhbHaXpCrWI)

Posted by Belmiro Moreira at 13:38 2 comments



Saturday, 26 September 2015

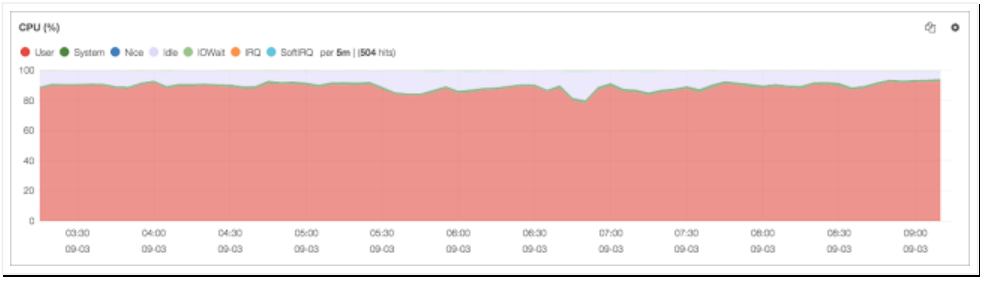
## EPT, Huge Pages and Benchmarking

Having reported that EPT has a negative influence on the High Energy Physics standard benchmark HepSpec06, we have started the deployment of those settings across the CERN OpenStack cloud,

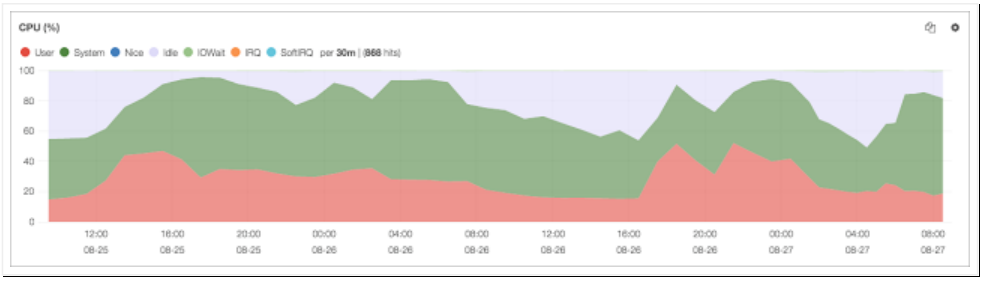
- Setting the flag in /etc/modprobe.d/kvm_intel.conf to off
- Waiting for the work on each guest to finish after stopping new VMs on the hypervisor
- Changing the flag and reloading the module
- Enabling new work for the hypervisor

According to the HS06 tests, this should lead to a reasonable performance improvement based on the results of the benchmark and tuning. However, certain users reported significantly worse performance than previously. In particular, some workloads showed significant differences in the following before and after characteristics.

Before the workload was primarily CPU bound, spending most of its time in user space. CERN applications have to process significant amounts of data so it is not always possible to ensure 100% utilisation but the aim is to provide the workload with user space CPU.




When EPT was turned off. some selected hypervisors showed a very difference performance profile. A major increase in non-user load and a reduction in the throughput for the experiment workloads. However, this effect was not observed on the servers with AMD processors.




With tools such as perf, we were able to trace the time down to handling the TLB misses. Perf gives

```
78.75% [kernel] [k] _raw_spin_lock
6.76% [kernel] [k] set_spte
1.97% [kernel] [k] memcmp
0.58% [kernel] [k] vmx_vcpu_run
0.46% [kernel] [k] ksm_docan
0.44% [kernel] [k] vcpu_enter_guest
```

The process behind the _raw_spin_lock is qemu-kvm.

Using systemtap kernel backtraces, we see mostly page faults and spte_* commands (shadow page table updates)
Both of these should not be necessary if you have hardware support for address translation: aka EPT.

There may be specific application workloads where the EPT setting was non optimal. In the worst case, the performance was several times slower. EPT/NPT increases the cost of doing page table walks when the page is not cached in the TLB. This document shows how processors can speed up page walks
- http://www.cs.rochester.edu/~sandhya/csc256/seminars/vm_yuxin_yanwei.pdf and AMD includes a page walk cache in their processor which speeds up the walking of pages as described in this paper http://vglab.cse.iitd.ac.in/~sbansal/csl862-virt/readings/p26-bhargava.pdf

In other words, EPT slows down HS06 results when there are small pages involved because the HS06 benchmarks miss the TLB a lot. NPT doesn't slow it down because AMD has a page walk cache to help speed up finding the pages when it's not in the TLB. EPT comes good again when we have large pages because it rarely results in a TLB miss. So, HS06 is probably representative of most of the job types, but the is a small share of jobs which are different and triggered the above-mentioned problem.

However, we have 6% overhead compared to previous runs due to EPT on for the benchmark as mentioned in the previous blog. Mitigating the EPT overheads following the comments on the previous blog, we looked into using dedicated Huge Pages. Our hypervisors run CentOS 7 and thus support both transparent huge pages and huge pages. Transparent huge pages performs a useful job under normal circumstances but are opportunistic in nature. They are also limited to 2MB and cannot use the 1GB maximum size.

We tried setting the default huge page to 1G using the Grub cmdline configuration.

```
$ cat /sys/kernel/mm/transparent_hugepage/enabled
[always] madvise never
$ cat /boot/grub2/grub.cfg | grep hugepage
linux16 /vmlinuz-3.10.0-229.11.1.el7.x86_64 root=UUID=7d5e2f2e-463a-
4842-8e11-d6fac3568cf4 ro
rd.md.uuid=3ff29900:0eab9bfa:ea2a674d:f8b33550
rd.md.uuid=5789f86e:02137e41:05147621:b634ff66 console=tty0 nodmraid
```

```
crashkernel=auto crashkernel=auto
rd.md.uuid=f6b88a6b:263fd352:c0c2d7e6:2fe442ac
vconsole.font=latarcyrheb-sun16 vconsole.keymap=us LANG=en_US.UTF-8
default_hugepagesz=1G hugepagesz=1G hugepages=55
transparent_hugepage=never
$ cat /sys/module/kvm_intel/parameters/ept
Y
```

It may also be advisable to disable tuned for the moment until the bug #1189868 is resolved.

We also configured the XML manually to include the necessary huge pages. This will be available as a flavor or image option when we upgrade to Kilo in a few weeks.

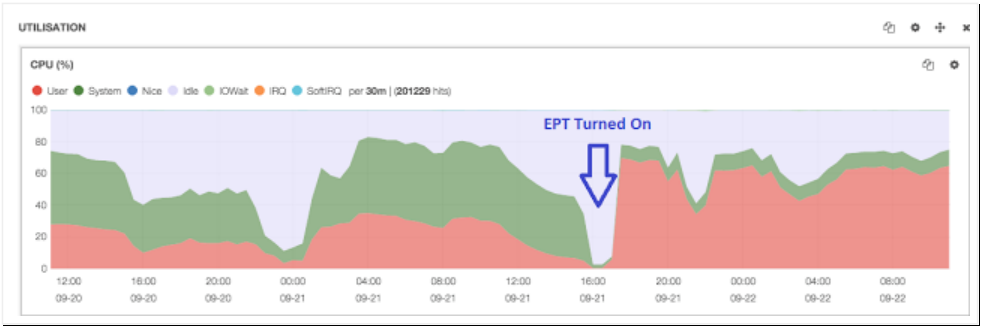
```
  <memoryBacking>
        <hugepages>
          <page size="1" unit="G" nodeset="0-1"/>
        </hugepages>
  </memoryBacking>
```

The hypervisor was configured with huge pages enabled. However, we saw a problem with the distribution of huge pages across the NUMA nodes.

```
$ cat /sys/devices/system/node/node*/meminfo | fgrep Huge
Node 0 AnonHugePages: 311296 kB
Node 0 HugePages_Total: 29
Node 0 HugePages_Free: 0
Node 0 HugePages_Surp: 0
Node 1 AnonHugePages: 4096 kB
Node 1 HugePages_Total: 31
Node 1 HugePages_Free: 2

Node 1 HugePages_Surp: 0
```

This shows that the pages were not evenly distributed across the NUMA nodes., which would lead to subsequent performance issues. The suspicion is that the Linux boot up sequence led to some pages being used and this made it difficult to find contiguous blocks of 1GB for the huge pages. This led us to deploy 2MB pages rather than 1GB for the moment, while may not be the optimum setting allows better optimisations than the 4K settings and still gives some potential for KSM to benefit. These changes had a positive effect as the monitoring below shows when the reduction in system time.




At the OpenStack summit in Tokyo, we'll be having a session on Hypervisor Tuning so people are welcome to bring their experiences along and share the various options. Details of the session will appear at https://etherpad.openstack.org/p/TYO-ops-meetup.

Contributions from Ulrich Schwickerath and Arne Wiebalck (CERN) and Sean Crosby (University of Melbourne) have been included in this article along with the help of the LHC experiments to validate the configuration.

## References


- Previous analysis for EPT at http://openstack-in-production.blogspot.fr/2015/08/ept-and-ksm-for-high-throughput.html
- Red Hat blog on Huge Pages at http://redhatstackblog.redhat.com/2015/09/15/driving-in-the-fast-lane-huge-page-support-in-openstack-compute/
- Mirantis blog on Huge Pages at https://www.mirantis.com/blog/mirantis-openstack-7-0-nfvi-deployment-guide-huge-pages/
- VMWare paper on EPT at https://www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf

- Academic studies of the overheads and algorithms of EPT and NPT (AMD's technology)
  at http://www.cs.rochester.edu/~sandhya/csc256/seminars/vm_yuxin_yanwei.pdf and http://vglab.cse.iitd.ac.in/~sbansal/csl862-virt/readings/p26-bhargava.pdf

Posted by Tim Bell at 07:23 3 comments



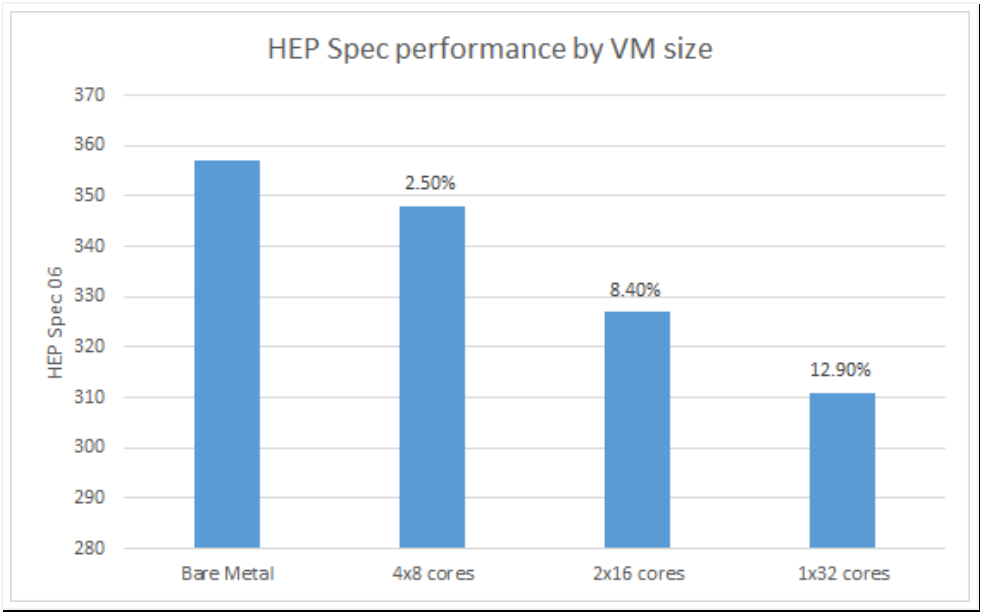
Wednesday, 5 August 2015

## Tuning hypervisors for High Throughput Computing

Over the past set of blogs, we've looked at a number of different options for tuning High Energy Physics workloads in a KVM environment such as the CERN OpenStack cloud.

This is a summary of the findings using the HEPSpec 06 benchmark on KVM and a comparison with Hyper-V for the same workload.

For KVM on this workload, we saw a degradation in performance on large VMs.




Results for other applications may vary so each option should be verified for the target environment. The percentages from our optimisations are not necessarily additive but give an indication of the performance improvements to be expected. After tuning, we saw around 5% overhead from the following improvements.

| Option | Improvement | Comments |
|---|---|---|
| CPU topology | ~0 | The primary focus for this function was not for performance so result is as expected |
| Host Model | 4.1-5.2% | Some impacts on operations such as live migration |
| Turn EPT off | 6% | Open bug report for CentOS 7 guest on CentOS 7 hypervisor |
| Turn KSM off | 0.9% | May lead to an increase in memory usage |
| NUMA in guest | ~9% | Needs Kilo or later to generate this with OpenStack |
| CPU Pinning | ~3% | Needs Kilo or later (cumulative on top of NUMA) |

Different applications will see a different range of improvements (or even that some of these options degrade performance). Experiences from other workload tuning would be welcome.

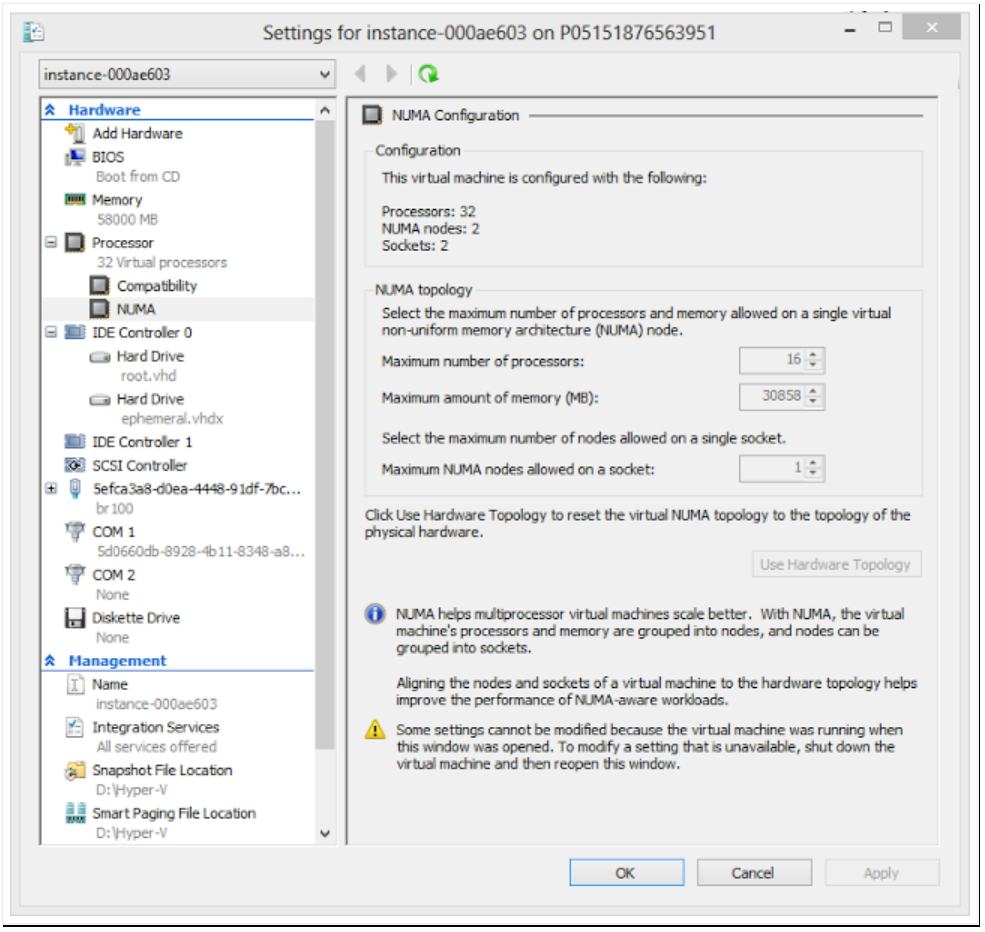
One of the things that led us to focus on KVM tuning was the comparison with Hyper-V. At CERN, we made an early decision to run a multi-hypervisor cloud building on the work by cloudbase.it and Puppet on Windows to share the deployment scripts for both CentOS and Windows hypervisors. This allows us to direct appropriate workloads to the best hypervisor for the job.

One of the tests when we saw a significant overhead on the default KVM configuration was to compare the performance overheads for a Linux configuration on Hyper-V. Interestingly, Hyper-V achieved better performance without tuning compared to the configurations with KVM. Equivalent tests on Hyper-V showed

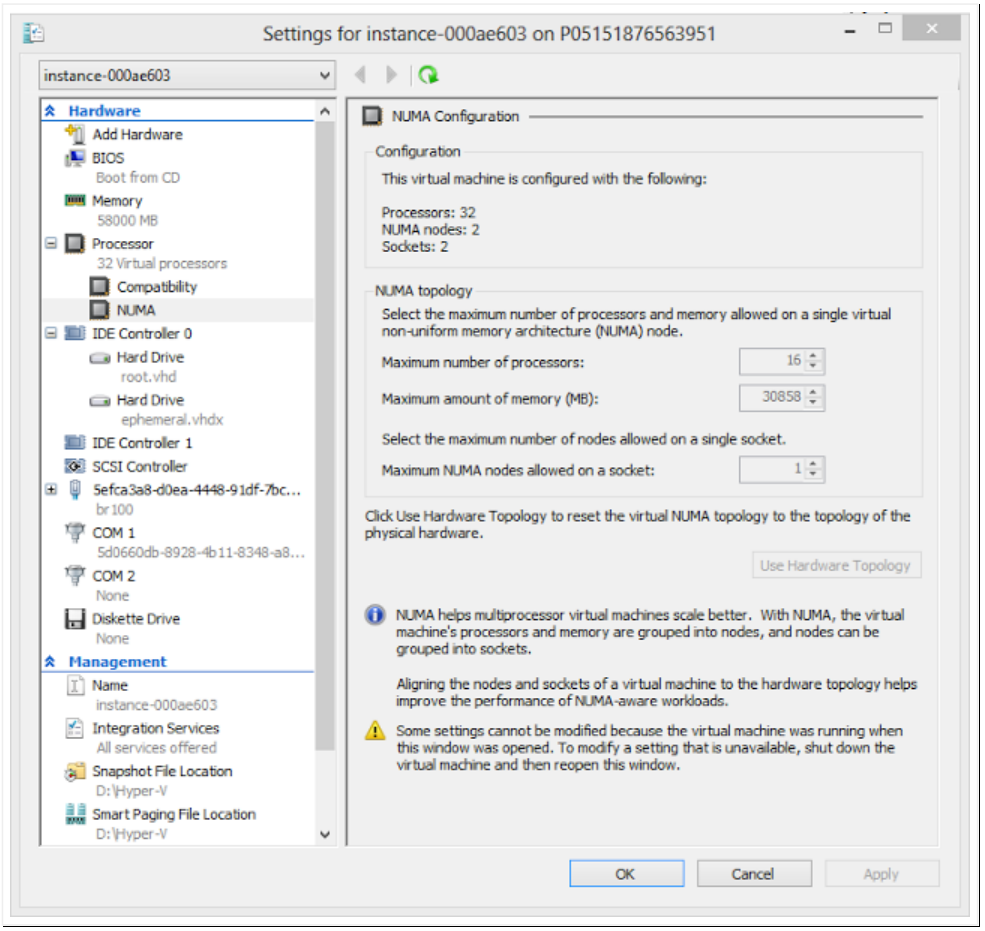
- 4 VMs 8 cores: 0.8% overhead compared to bare metal
- 1 VM 32 cores: 3.3% overhead compared to bare metal

These performance results allowed us to focus on the potential areas for optimisation, that we needed to tune the hypervisor rather than a fundamental problem with virtualisation (with the results above for NUMA and CPU pinning)

The Hyper-V configuration pins each core to the underlying NUMA socket which is similar to how the Kilo NUMA tuning sets KVM up.




and




This gives the Linux guest configuration as seen from the guest running on a Hyper-V hypervisor

```
# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
node 0 size: 28999 MB
node 0 free: 27902 MB
node 1 cpus: 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
node 1 size: 29000 MB
node 1 free: 28027 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
```

Thanks to the QEMU discuss mailing list and to the other team members who helped understand the issue (Sean Crosby (University of Melbourne) and Arne Wiebalck, Sebastian Bukowiec and Ulrich Schwickerath (CERN))

Posted by Tim Bell at 08:58 2 comments

G+1 +2 Recommend this on Google

Labels: cern openstack hyperv kvm numa

Home Older Posts

Subscribe to: Posts (Atom)