OpenStack performance optimization
NUMA, Large pages & CPU pinning

Daniel P. Berrangé <berrange@redhat.com>
About me

- Contributor to multiple virt projects
- Libvirt Developer / Architect 8+ years
- OpenStack contributor 2 years
- Nova Core Team Reviewer
- Focused on Nova libvirt + KVM integration
Talk Structure

- Introduction to OpenStack
- NUMA config
- Large page config
- CPU pinning
- I/O devices
What is OpenStack?

- Public or private cloud
- Multiple projects (compute, network, block storage, image storage, messaging, ....)
- Self-service user API and dashboard
What is OpenStack Nova?

- Execution of compute workloads
- Virtualization agnostic
  - Libvirt (KVM, QEMU, Xen, LXC), XenAPI, Hyper-V, VMware ESX, Ironic (bare metal)
- Concepts
  - Flavours, instances, image storage, block storage, network ports
Nova approach

- Cloud infrastructure administrators
  - Flavours for VM instance policy
  - Minimal host provisioning / setup
  - No involvement in per-VM setup
- Guest instance users
  - Preferences via image metadata
  - No visibility of compute hosts / hardware
Nova architecture (simplified)
Current VM scheduling

• VM scheduler has multiple filters
• Filters applied to pick compute host
• Overcommit of RAM and CPUs
• VMs float across shared resources
• Assignment of I/O devices (PCI)
Scheduling goals

• Motivation: Network function virt (NFV)
  – Support “dedicated resource” guest
  – Support predictable / low latency

• Motivation: Maximise hardware utilization
  – Avoid inefficient memory access on NUMA
NUMA

- Factors for placement
  - Memory bandwidth & access latency
  - Cache efficiency
  - Locality of I/O devices
- Goal – small guests
  - Fit entirely within single host node
- Goal – large guests
  - Define virtual NUMA topology
  - Fit each guest node within single host node
libvirt host resource info

<capabilities>
  <host>
    <topology>
      <cells num='2'>
        <cell id='0'>
          <memory unit='KiB'>4047764</memory>
          <pages unit='KiB' size='4'>999141</pages>
          <pages unit='KiB' size='2048'>25</pages>
          <distances>
            <sibling id='0' value='10'/>
            <sibling id='1' value='20'/>
          </distances>
        </cell>
        <cell id='1'>....
      </cells>
    </topology>
  </host>
</capabilities>
Nova NUMA config

- Property for number of guest nodes
  - Default: 1 node
  - hw:numa_nodes=2

- Property to assign vCPUS/RAM to guest nodes
  - Assume symmetric by default
  - hw:numa_cpu.0=0,1
  - hw:numa_cpu.1=2,3,4,5
  - hw:numa_mem.0=500
  - hw:numa_mem.1=1500

- **NO** choice of host node assignment
NUMA impl

- Scheduling
  - Hosts NUMA topology recorded in DB
  - VM Instance placement recorded in DB
  - Filter checks host load to identify target
  - Scheduler records NUMA topology in DB
  - Compute node starts VM with NUMA config
libvirt NUMA config

- VCPUs pinned to specific host NUMA nodes
- VCPUs float within host NUMA nodes
- Emulator threads to union of vCPU threads

```xml
<vcpu placement='static'>6</vcpu>
<cputune>
  <vcpupin vcpu="0" cpuset="0-1"/>
  <vcpupin vcpu="1" cpuset="0-1"/>
  <vcpupin vcpu="2" cpuset="4-7"/>
  <vcpupin vcpu="3" cpuset="4-7"/>
  <vcpupin vcpu="4" cpuset="4-7"/>
  <vcpupin vcpu="5" cpuset="4-7"/>
  <emulatorpin cpuset="0-1,4-7"/>
</cputune>
```
Libvirt NUMA config

- VCPUS + RAM regions assigned to guest NUMA nodes
- RAM in guest NUMA nodes pinned to host NUMA nodes

```xml
<memory>2048000</memory>
<numatune>
    <memory mode='strict' nodeset='0-1'/>
    <memnode cellid='0' mode='strict' nodeset='0'/>
    <memnode cellid='1' mode='strict' nodeset='1'/>
</numatune>
<cpu>
    <numa>
        <cell id='0' cpus='0,1' memory='512000'/>
        <cell id='1' cpus='1,2,3,4' memory='1536000'/>
    </numa>
</cpu>
```
Large pages

• Factors for usage
  – Availability of pages on hosts
  – Page size vs RAM size
  – Lack of over commit

• Goals
  – Dedicated RAM resource
  – Maximise TLB efficiency
Large page config

• Property for page size config
  – Default to small pages (for over commit)
  – `hw:mem_page_size=large|small|any|2MB|1GB`
Large page impl

• Scheduling
  – Cloud admin sets up host group
  – NUMA record augmented with large page info
  – Filter refines NUMA decision for page size
libvirt large page config

- Page size set for each guest NUMA node

```xml
<memoryBacking>
  <hugepages>
    <page size='2' unit='MiB' nodeset='0-1'/>
    <page size='1' unit='GiB' nodeset='2'/>
  </hugepages>
</memoryBacking>
```
CPU pinning

Factors for usage
- Efficiency of cache sharing
- Contention for shared compute units

Goals
- Prefer hyperthread siblings for cache benefits
- Avoid hyperthread siblings for workload independence
- Dedicated CPU resource
CPU pinning config

- Property for dedicated resource
  - hw:cpu_policy=shared|dedicated
  - hw:cpu_threads_policy=avoid|separate|isolate|prefer
CPU pinning impl

• Scheduling
  – Cloud admin sets up host group
  – NUMA info augmented with CPU topology
  – Filter refines NUMA decision with topology
libvirt CPU pinning config

- Strict 1-to-1 pinning of vCPUs <-> pCPUs
- Emulator threads pinned to dedicated CPU

```xml
<cputune>
  <vcpupin vcpu="0" cpuset="0"/>
  <vcpupin vcpu="1" cpuset="1"/>
  <vcpupin vcpu="2" cpuset="4"/>
  <vcpupin vcpu="3" cpuset="5"/>
  <vcpupin vcpu="4" cpuset="6"/>
  <vcpupin vcpu="5" cpuset="7"/>
  <emulatorpin cpuset="2"/>
</cputune>
```
I/O devices

• Factors for usage
  – Locality of PCI device to NUMA node
  – Connectivity of PCI network interface

• Goals
  – Assign PCI device on local NUMA node
Libvirt device info

<device>
    <name>pci_0000_80_16_7</name>
    <path>/sys/devices/pci0000:80/0000:80:00.16.7</path>
    <capability type='pci'>
        <domain>0</domain>
        <bus>128</bus>
        <slot>22</slot>
        <function>7</function>
        <product id='0x342c'>5520/5500/X58 Chipset QuickData Technology</product>
        <vendor id='0x8086'>Intel Corporation</vendor>
        <iommuGroup number='25'>
            <address domain='0x0000' bus='0x80' slot='0x16' function='0x0'/>
        </iommuGroup>
        <numa node='1'/>
    </capability>
</device>
I/O device impl

• Scheduling
  – Hosts record locality of PCI devices in DB
  – Filter refines NUMA decision for device

• Guest config
  – TBD: Tell guest BIOS NUMA locality of PCI dev

https://wiki.openstack.org/wiki/VirtDriverGuestCPUMemoryPlacement
http://people.redhat.com/berrange/kvm-forum-2014/