oVirt QoS

Martin Sivák
Red Hat Czech

KVM Forum
October 2013
Agenda

• Why is QoS important?
• Scalability and management challenges
• Managing resources
  • CPU
  • Network
  • Memory
• Future plans
How do we define QoS? SLA?

"Quality of service comprises requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, [...]."[1]

How do we define QoS? SLA?

"A service-level agreement is a negotiated agreement between two or more parties, where one is the customer and the others are service providers."\[1\]

\[1\] http://en.wikipedia.org/wiki/Service-level_agreement
Not Just Hypothetical...

- Alter Way Hosting[^1]
  - Hundreds of VMs on oVirt for its clients

- Resource Allocation Challenges
  - Media streaming – bandwidth requirements
  - Database server – heavy I/O
  - Scientific computing – CPU and memory
  - Power savings vs QoS
  - More efficient hardware utilization

- As infrastructure admin, how do you meet the SLA requirements of your customers and users?

[^1]: http://www.ovirt.org/Alter_Way_case_study
SLA / QoS Scaling

How do we apply SLA/QoS without the management application becoming a bottleneck?

1,000... 10,000... 100,000 hosts?!

And how to manage that without the admins going crazy?
Ovirt delegates QoS tasks to Hosts

- Management-Level Policy
- Host-Level Enforcement
- Reporting
And to make admins happy...

- Policy “documents” with QoS parameters for devices
  - Number of vNICs (each with own profile)
  - Disks, memory, cpu limits
  - Hard limits, soft limits, memory ballooning

- Admin will assign a policy to each VM
  - Setting a policy vs. entering bunch of numbers
  - Consistency between VMs of the same type
  - Changes to the policy
CPU

- Capacity/load: GHz, Shares, ...
- Different architectures and capabilities

```
# cat /proc/cpuinfo
[...]
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe
syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts
rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni
pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 fma cx16 xtpx
pdcm pcid sse4_1 sse4_2 movbe popcnt tsc_deadline_timer aes xsave
avx f16c rdrand lahf_lm abm ida arat epb xsaveopt pln pts dtherm
trp_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2
smep bmi2 erms invpcid
[...]
```

- How fast is 2GHz, anyway?
CPU Shares

- Just introduced (oVirt 3.3)
- Priorities
- Relative weights

```xml
<domain>
  ...
  <cputune>
    ...
    <shares>2048</shares>
    ...
  </cputune>
  ...
</domain>
```

- cgroup usage cap?
Network

- Capacity/load: bandwidth
- Consider latency and possible packet loss

- 3 levels:
  - VM (vNIC)
  - Host (physical NIC)
  - DC (switches, SDN)

- May have multiple vNICs to manage
QoS Objects and vNIC Profiles

- QoS objects
  - Inbound/outbound
  - Avg/peak/burst

- VNIC profile
  - QoS + permissions + etc.
  - Attach to network profile

- Access to profiles depends on user's permissions
  - E.g. professor vs student
Memory

- Capacity: amount available
- Load: amount used

... not that simple
• Kernel SamePage Merging

• 52 virtual instances of Windows XP with 1GB of memory, could run on a hypervisor that had only 16GB of RAM
VirtIO Memory Balloon

- The balloon driver is a special process
  - Non-swappable and un-killable
  - May be inflated or deflated

- Inflate => take more RAM from the guest OS
- Deflate => return RAM to the guest OS
VirtIO Memory Balloon

- Memory pages in the balloon are unmapped
- Then, reclaimed by the host

And now we can do memory over-commitment!

- 2 GB physical server runs 2x1GB VMs
- Using the balloon we can run 3x1GB VMs
  - Each VM's balloon will free 512MB back to the host
Host-Level Considerations

- Guest balloon drivers select pages to balloon without considering whether the host page might be shared.

- Ballooning a shared page is a mistake because it deprives the guest of resources without actually saving any host memory.
MoM to the rescue!

- Written and maintained by Adam Litke (IBM)
- Joined oVirt as an incubation project, now fully merged
- Monitors and handles KSM and ballooning
- Goal to prevent interaction mistakes
  - Ballooning VS KSM
MoM architecture

- Guest tracking
- Stats collection
- Fully extensible

- Dynamic policy engine
- Support for KSM and ballooning
- Stand-alone or integrated
MoM-VDSM Integration: under the hood

- MoM threads run within vdsm
- Stats collected via the vdsm API
- KSM / ballooning operations via vdsm API
- VDSM installs a default MoM policy

MoM<->Engine Communication

- Engine sends policy attributes
- VDSM converts them to MoM policy (constants only)
- MoM merges constants with the actual policy logic files
- Policy is enforced, results are collected and sent back to engine
Future of memory QoS management

- Policy “documents” for all resources (CPU, memory, IO)
- NUMA-aware SLA policies
- Upper hard limit for CPU (cgroups)

- Full control: min guarantee soft limit upper hard limit
Storage

- Capacity/load: IOPs
- Storage size considered for quotas

- Storage profile?
  - Min/Max IOPs
  - Not in oVirt today
- Mostly NFS, iSCSI – network based
  - Network SLA
- Lean on MoM
Summary

- Management defines the values
- Policy documents – SLA plans
- Enforcing delegated to host agent
- CPU, Memory, Network, Storage
Questions?
THANK YOU!

http://www.ovirt.org
http://www.ovirt.org/Category:SLA

http://lists.ovirt.org/mailman/listinfo
vdsm-devel@lists.fedorahosted.org

#ovirt irc.oftc.net

msivak@redhat.com