Scheduling & SLA @oVirt

Shanghai 2013

Doron Fediuck
Red Hat
Overview: SLA

- **SLA**: Service Level Agreement
  - Ensures Quality of Service (QoS) based on parameters and a schema.

- **ISP**
  - Schema would be Internet access.
  - Parameters: Up/Down bandwidth, MTTR (Mean Time To Recover), etc.

- In cloud computing this is becoming crucial, as we're providing IaaS
Overview: Scheduling

- Placing a VM on a host
- Schedule various host tasks

Machine re-assignment problem\(^1\)

- Defined by Google; assign each process to a machine. All processes already have an original (unoptimized) assignment. Each process requires an amount of each resource (such as CPU, RAM, ...)
- A solution to this problem is a new process-machine assignment which satisfies all hard constraints and minimizes a given objective cost

Found to be mathematically NP-Complete (**can't be solved**)

\(^1\)http://challenge.roadef.org/2012/en/
Overview: Scheduling & SLA

So what CAN we do?

• Optimize scheduling scenarios
  • Scheduling improvements
  • Integration with external systems

• Gradually introduce SLA elements into oVirt
  • Add various features which will function as a toolbox
  • Prepare the infrastructure for advanced SLA concepts
Scenarios

What is it good for, anyway?
Scenarios

SLA Based

- Multi Tenancy / cloud models: capping, quotas
- VM HA

Scheduling based

- Memory over commitment
- Power saving policies
- KSM performance: positive affinity
- Advanced scheduling
Scenarios: SLA

Private-cloud / multi-tenancy models

- Limitations / Capping (CPU, RAM, TBD...)
  - Allow limiting a VM's resource consumption
  - Provide better control on VM behavior and prevent a VM from going wild.

- Quota
  - Management level limitations
Scenarios: SLA

VM High Availability

- **Host level:** Tagged hosts should be used when scheduling HA-VMs.

- **VM level:** allow auto-reset when guest fails (blue screen, etc.)

- **Application level:** monitor specific application(s) and act accordingly (reset, migrate, etc) when it stops responding
Scenarios

VM affinity (co-location, Positive / Negative)

• Negative affinity
  • One VM 'repels' the other
  • HA via separate host VM placements

• Positive affinity
  • One VM 'attracts' the other VM
  • Grouping all VMs with the same OS will get best KSM results.
  • Licensing pricing model in some OSs
  • Simple maintenance and power saving
  • Traffic monitoring for specific VMs
Scenarios: Scheduling

HW utilization: Memory Over Commitment

- Allow running more VMs than available physical memory

Power saving policies

- Shutdown idle VMs
- Gather all VMs to several hosts (load balancing, already exists) and shut down / suspend unused hosts.
Scenarios: Scheduling

Advanced VM scheduling

- Time based: turn on/off at a given time
- Various algorithms implementations
- Statistic-based scheduling
Scheduling considerations
Consider while scheduling...

Each VM and host has meta-data crucial for scheduling

- **Resources**
  - Connection to network RED
  - Storage usage (DB in a guest)
  - HA reservations

- **Topologies**
  - CPU pinning
  - NUMA
Consider while scheduling...

Resource mapping should be preserved after migration
  • What happens when destination host will not support it?

Avoid collisions
  • Host-Pinning / HA vs Power savings
  • CPU-pinning vs NUMA / KSM
  • Optional vs Mandatory VM network

Naive rule: specific settings will override the general policy
  • Host-Pinning overrides Power savings
Scheduling & SLA Today

What do we have so far?
Scheduling & SLA Today

Existing Algorithms
- Even distribution
- Power saving

Current scheduling
- Running a VM
  - Basic validations
  - HasMemoryToRunVM
  - Use the relevant selection algorithm to find the best host
Scheduling & SLA Today

Current scheduling

- Migrating a VM
  - Same validations as with running a VM
  - Avoid selecting current host
  - HasCpuToRunVM
  - Use the relevant selection algorithm to find the best host

Load balancing (cluster policy)

- Time based polling, using one of the current selection algorithms to migrate VMs as needed.
Scheduling & SLA Today

New features 3.1 and 3.2 introduced

- Enabling memory balloon by default\(^1\)
  - Deflated, may be used externally

- CPU pinning\(^2\)
  - Specific and range pinning topology
  - Migration allowed
    - No validation on destination host.


Scheduling & SLA Today

Quota

- Control resource allocation
- See it in YouTube!
- Storage quota
- Cluster (Memory+CPU) quota
- Disabled (default), audit and enforcing modes
- Search-queries (VMs, templates and disks)


Scheduling & SLA Today

Quota sample

New Quota

Name: SampleQuota
Description: Students quota year 2012
Data Center: Default

Memory & CPU

80% 120%

All Clusters
Specific Clusters

Cluster Name | Memory | vCPU
Default | 0 out of 2048 MB | 0 out of Unlimited vCPUs

Storage

75% 120%

All Storage Domains
Specific Storage Domains

Storage Name | Quota
All Storage Domains | 0 out of 1024 GB

Edit Quota

Memory:
- Unlimited
- Limit to: 2048 MB

CPU:
- Unlimited
- Limit to: vCPUs
Scheduling & SLA Today
Scheduling & SLA Today

- Better Hyperthreading support
- Native CPU flags support
- VDSM-MoM integration\(^1\)
  - Written and maintained by Adam Litke
  - Joined oVirt as an incubation project
  - Monitors and handles ksm and ballooning
  - Trying to prevent interaction mistakes
    - Ballooning VS KSM

\(^1\) [http://wiki.ovirt.org/wiki/SLA-mom](http://wiki.ovirt.org/wiki/SLA-mom)
Work in Progress

So what are we doing?
Work in Progress

Pluggable scheduling API

- Add to internal scheduler
- Allow users to write their own scheduling logic
- Simple API
- Community friendly
- Actually, needed by community...
Work in Progress
Smart Scheduler
Integrating BMC

Srinivas Gowda G
Surya Prabhakar
Dell India R&D

Presented
In Bangalore
oVirt workshop

Start VM / Migrate a VM
CPU + Memory
Host_A
Host_B
Host_C
Load Balancing Policy
Check for Hardware status
Host_A
Work in Progress

MoM integration\(^1\)

- MoM is becoming the enforcement agent
- VDSM integration done by Adam Litke and his colleagues (Mark Wu, Royce Lv)
  - Still gaps on engine side.
- Initial phase for basic integration while maintaining ksm functionalities, adding API support for memory balloon
  - Packaging and maintaining (added to Bugzilla)
- Now adding capping (limitations) API support to VDSM
  - CPU & Memory (guaranteed, hard and soft limits)

\(^1\) http://wiki.ovirt.org/wiki/SLA-mom
Work in Progress

- SLA features
  - VM Watchdog (VM HA)
  - Network QoS

- Extend MoM capabilities
  - Handle specific VMs
  - Policy resolution (allow policy parts)
  - Limitations for network & storage
Road-map to Infinity (affinity?) and Beyond!
Scheduling & SLA Road-map

- SLA features
  - HEAT integration (Application HA)
  - NUMA (numad, auto-numa)

- Scheduling: additional improvements

- Extend MoM capabilities
  - Handle specific VMs
  - Additional policies
  - Limitations for network & storage
and now is a good time for.... Questions?
THANK YOU!

http://wiki.ovirt.org/wiki/Category:SLA
engine-devel@ovirt.org
vdsm-devel@lists.fedorahosted.org

#ovirt irc.oftc.net