

Dynamic scaling of CPU and RAM

- [Bug Reference](#)
- [Branch](#)
- [Introduction](#)
- [Feature Specifications](#)
- [Hypervisor support](#)
- [Use cases](#)
- [Architecture and Design description](#)
- [Web Services APIs](#)
- [Xenserver /Vmware Tools](#)
- [Hypervisor Changes](#)
 - [Xenserver](#)
 - [Vmware](#)
- [UI flow](#)
- [Open Issues](#)

Bug Reference

<https://issues.apache.org/jira/browse/CLOUDSTACK-658>

Branch

4.2.0 onwards

Introduction

It is not always possible to exactly identify the CPU and RAM requirements at the time of deploying a VM. But for various reasons it may be required to scale up these resources later on. At that time there is no other way but to restart the VM with increased resources. Dynamic scaling for CPU and RAM feature would allow to change these resources for a running VM avoiding any downtime.

Currently CS allows updating CPU/RAM by changing to a different compute offering for stopped VMs. This feature will enable the same for running VMs.

Feature Specifications

- Ability to scale up CPU and/or RAM for running user VMs based on predefined compute offerings
- Ability to scale up CPU and/or RAM for running system VMs based on predefined system compute offerings
- If scaling up requires a migration it will be limited within the cluster only - if the current host where VM is running has capacity then simply the resources will be scaled up, if not live migration will be done within the cluster and then resources will be scaled up. If no suitable host is found operation will fail. In future if live migration across clusters is possible then this constraint can be relaxed.
- Ability to mark the VM for scale up at creation time (see open issue#1)
- Ability to scale down CPU and/or RAM (see open issue#2)
- No check for guest OS compatibility (can be considered only if there is a concise document/API listing for supported guest OS's for each hypervisor)
- Only supported for newly created VMs, existing VMs (from prior CS releases) won't have the capability to scale up. To make it work for existing VMs we need to update VM using "**updateVirtualMachine**" API as explained in "Web services APIs" section.
- Ability to scale system vms (only for VMware). We do not support System vm scaling for xenserver as the system template for xenserver does not have XS tools to support hot add memory/cpu inside OS.

Hypervisor support

Currently planning to do it for VMware and Xenserver. Support for other HVs can also be added based on HV capabilities.

For KVM - Marcus Sorrensen has sponsored to do this.

Use cases

- End users can scale up CPU and/or RAM for running VMs

Architecture and Design description

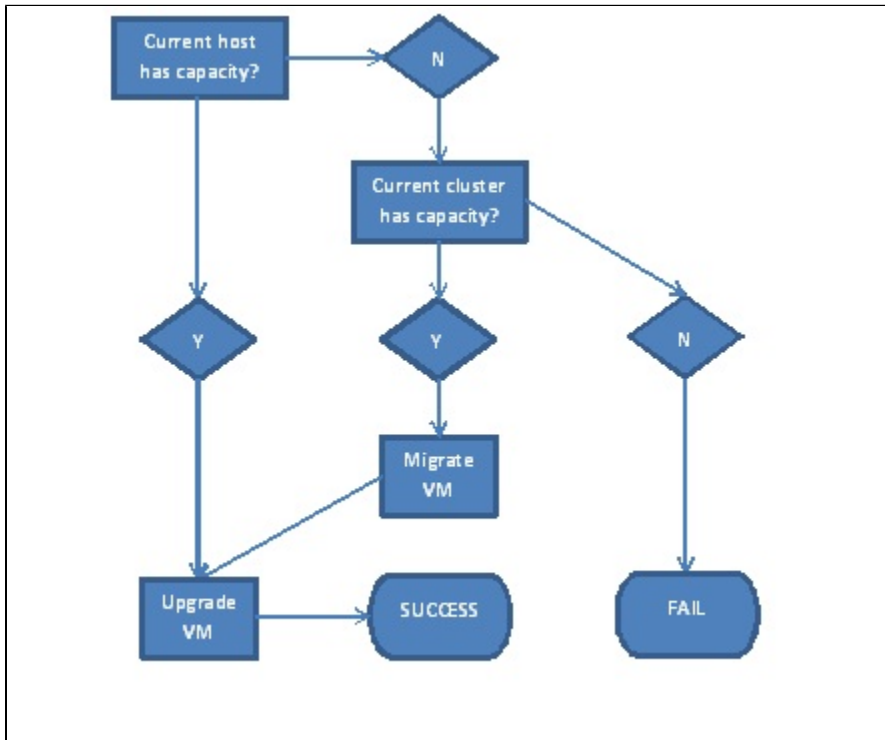
A new command class needs to be introduced for actually changing CPU/RAM at agent layer. This needs to be handled for the supported HVs.

ReconfigureVMCommand: This will have the updated CPU and RAM values as members

Allocation logic: Refer to flow chart below. There are 3 primary use cases:-

- **VM's current host has capacity** - If VM's current host has capacity to scale up the vm then we put the vm in Reconfiguring state and lock the delta capacity. We then send the ReconfigureVMCommand to the HV to reconfigure the vm and scale it to the new values. Whether success / failure we put the vm back into running state, but release the delta capacity in case of failure.

- **VM's current host doesn't have capacity but the vm's cluster has** - If VM's current host doesn't have capacity, then we call up the planners to find a suitable host that can take the scaled up vm in the cluster. Once the host is found out we lock the new required capacity on the new host and migrate the vm to that host. Once migrated we send ReconfigureVMCommand to the HV. If there is failure in reconfiguring here then we release the delta capacity on the new host.
- **Cluster in which vm is running doesn't have capacity** - we simply return failure to the end user saying that we don't have enough capacity to scale up the vm.



Web Services APIs

Following APIs need to be changed:

scaleVirtualMachine - There is an existing API `upgradeVirtualMachine` and takes `vm_id` and `compute_offering_id` as inputs. This is a sync call currently and will not be used anymore since we need it to be async. I plan to deprecate this api in the next big release (5.0).

With this feature I will introduce another API named `scaleVirtualMachine` which will be similar to `upgradeVirtualMachine` in every aspect except that it would be async. In addition it will be callable when the vm is in running state as well to dynamically scale the vm.

In case of a migration this will internally use the `migrateVirtualMachine` API logic.

scaleSystemVm - There is an existing API `upgradeSystemvm` and takes `system vm_id` and `compute_offering_id` as inputs. This is a sync call currently and will not be used anymore since we need it to be async. I plan to deprecate this api in the next big release (5.0).

With this feature I will introduce another API named `scaleSystemvm` which will be similar to `upgradeSystemvm` in every aspect except that it would be async. In addition it will be callable when the system vm is in running state as well to dynamically scale the vm.

registerTemplate/updateTemplate - An additional optional boolean parameter "isdynamicallyscalable" is introduced, the value is true if template contains XS/VMWare tools in order to support dynamic scaling of VM cpu/memory

updateVirtualMachine - An additional optional boolean parameter "isdynamicallyscalable" is introduced, the value is true if VM contains XS/VMWare tools in order to support dynamic scaling of VM cpu/memory. Stop and start the VM in order to dynamic scaling to work.

Xenserver /Vmware Tools

For dynamic scaling to work virtual machine should have XS tools / VMware tools installed on it. To ensure this Admin/User can do it in two ways :-

- Admin/User while registering the template provides an input whether tools are installed on the template(or can be done using update template API).
- If the user deploys a virtual machine with a template that does not have XS tools / VMware tools and later if he/she installs the tools on the VM then he can inform Cloudstack using `updatevirtualmachine` API. After installation of tools and updating the virtual machine, user needs to stop and start the vm from cloudstack in order for dynamic scaling of CPU and RAM for that VM. The reason why we need to stop start the VM after updating is we need to set static max memory to some higher value before VM starts so that we can dynamically scale the VM upto static max.

Hypervisor Changes

Xenserver

To facilitate scaling up RAM for the VMs in xenserver utilizes Dynamic Memory Control(DMC) to change the amount of host physical memory assigned to any running virtual machine without rebooting it.

During the deployment of VM, if we set memory-dynamic-min = memory-dynamic-max = requested RAM(say K) then the guest VM gets K amount of RAM, later we can increase the RAM by setting a higher value than K within the memory static range.

Note : From XS following are the constraints (read xenserver admin guide)

- 0 memory-static-min memory-dynamic-min memory-dynamic-max memory-static-max
- Dynamic Minimum 14 Static Maximum for all supported operating systems

Following are the settings made during the CS API calls.

DeployVmCmd -

During the initial deployment of the VM we set memory variable as follows :-

- static min = service_offering / memory_overprovisioning_of_cluster.
- dynamic min = service_offering / memory_overprovisioning_of_cluster.
- dynamic max = service_offering
- static max = f_min(4 * service_offering) / memory_overprovisioning_of_cluster , Xenserver recommended max for guestOS)

ScaleVmCmd -

During scaling up of vm, the new requested RAM is set to mem-dynamic-min and mem-dynamic-max. Static min and max do not change unless the vm is stopped and started.

While scaling "up" the vm say from service offering "x" to "y" we will change

- dynamic min = new service_offering / memory_overprovisioning_of_cluster.
- dynamic max = new service_offering

and ensure that

- new dynamic max (y) <= static max and
- new dynamic min (y / memory_overprovisioning_of_cluster) >= static min.

[Further reading](#)

Vmware

DeployVmCmd -

During the initial deployment of the VM we enable the flags hotAddCpu and hotAdd memory true. These flags will be turned on if the guest OS supports it. Read http://partnerweb.vmware.com/comp_guide2/pdf/VMware_GOS_Compatibility_Guide.pdf

ScaleVmCmd -

During scaling up of vm, the new requested RAM and CPU are set if the flags are turned on.

Limitation - 1 - After dynamically scaling memory user "might" need to run a couple of commands on Linux OS for new memory to take affect --> [Further Reading](#)

Limitation - 2 - If a VM is initially assigned a RAM of less than 3gb then it cannot be dynamically scaled beyond 3gb. Holds true for Linux 64 bit and windows 7 32 bit guest os. [Further Reading](#)

Limitation - 3 - Hot add vcpu will fail If the number of cores per socket is not 1 and virtual machine hardware version=7. [Further Reading](#)

DB Changes

- Introducing a global config - enable.dynamic.scale.vm. Enables/Diables dynamically scaling a vm. This can be used to turn off the feature and is available at the zone level. By default the feature is turned off for upgrade reasons.
- Introducing a global config - scale.retry. By default the value = 2. This is for number of tries before failing the scaling.
- A new column in "vm_template" table named "dynamically_scalable" of type tinyint(1) and the value is 1 if template contains XS/VMWare tools inorder to support dynamic scaling of VM cpu/memory otherwise 0
- "user_vm_details" table holds key value pair with key "enable.dynamic.scaling" and value is true if VM contains XS/VMWare tools inorder to support dynamic scaling of VM cpu/memory otherwise false

UI flow

- UI needs to give an action for upgrading vm (when vm is in running / stopped state) just like we give the same option when vm is stopped.
- UI needs to call the new api scaleVirtualMachine for this and also keep in mind that this api is async in nature unlike the previous one which was sync. Other than that all the parameters remain the same.
- They show that for systemvms "which action label", "which api should be called " and "which HV" information

System Vms

State \ HV	XS	VMware	KVM
Stopped	Scale System VM API	Scale System VM API	Scale System VM API
Running	X	"	X

Change or offering

- They show that for uservms "which action label", "which api should be called " and "which HV" information

User Vm

State \ HV	XS	VMware	KVM
Stopped	Scale VM API	Scale VM API	Scale VM API
Running	"	"	X

Change or offering

Scale Up

- Created a subtask for this <https://issues.apache.org/jira/browse/CLOUDSTACK-1041>
- During template registration/updation there should be a check box to know whether tools are installed on the template or not. Same check box should be there for updateVirtualMachine API. <https://issues.apache.org/jira/browse/CLOUDSTACK-2988>

Open Issues

- Should scale down be allowed? It can be explicitly prevented since none of the HVs/guest OS supports it.
- There is also an option of having a custom compute offering where user can specify values for CPU and RAM during deployment or scaling up. But am not sure if this option can be misused since this is a user level API. Another complexity is to capture usage. Currently it is done based on compute offering.
- Scaling flag should be put in service offering to enable/disable scaling