

CPU Utilisation case study Configuration

Host:

OS: Ubuntu 16.04

Disable apparmor

```
$ /etc/init.d/apparmor stop
$ /etc/init.d/apparmor teardown
$ update-rc.d -f apparmor remove
```

Isolate 2 cores from the OS by modifying: /etc/default/grub

```
GRUB_CMDLINE_LINUX_DEFAULT="text isolcpus=10,11"
```

Update grub:

```
$ sudo update-grub
```

Reboot

Install barometer build-essentials:

```
$ git clone https://gerrit.opnfv.org/gerrit/barometer
$ ./src/install_build_deps.sh
```

Install virtualization packages and other required packages:

```
$ sudo apt install qemu-kvm libvirt-bin virtinst virt-manager screen
```

Add the user to the libvirtd group:

```
sudo adduser $USER libvirtd
```

Create a dir to store ISOs

```
$ sudo mkdir /usr/local/src/images/
$ cd /usr/local/src/images/
$ sudo -E wget http://releases.ubuntu.com/16.04/ubuntu-16.04.2-server-i386.iso
```

Create a Test VM

```
$ sudo virt-install -n test -r 256 --disk path=/var/lib/libvirt/images/test.img,bus=virtio,size=4 -c /usr/local
/src/images/ubuntu-16.04.2-server-i386.iso --network network=default,model=virtio --graphics vnc,listen=0.0.0.0
--noautoconsole -v
```

Connect to the console to complete the installation process.

Clone the latest libvirt code, build and install collectd:

```
$ git clone https://github.com/maryamtahhan/collectd.git mt_collectd
$ cd mt_collectd
$ ./build.sh
$ ./configure
```

Check that the virt plugin is configured:

```
virt . . . . . yes
```

Install collectd:

```
$ sudo make -j install
```

Install collectd as a service:

```
sudo mv systemd.collectd.service collectd.service
sudo chmod +x collectd.service
vi collectd.service
```

Edit the collectd.service script

```
[Service]#ExecStart=/usr/sbin/collectd
ExecStart=/opt/collectd/sbin/collectd -C /opt/collectd/etc/collectd.conf
#EnvironmentFile=-/etc/sysconfig/collectd
#EnvironmentFile=-/etc/default/collectd
ProtectSystem=full
ProtectHome=true
```

Modify the /opt/collectd/etc/collectd.conf file

```

<LoadPlugin cpu>
  Interval 1
</LoadPlugin>

<LoadPlugin virt>
  Interval 1
</LoadPlugin>

LoadPlugin csv

LoadPlugin network

<Plugin cpu>
  ReportByCpu true
  ReportByState false
  ValuesPercentage true
</Plugin>

<Plugin csv>
  DataDir "/tmp/collectd/csv"
  # StoreRates false
</Plugin>

<Plugin network>
  # server setup:
  Listen "<host ip address>" "28597"
  <Listen "<host ip address>" "28597">
  </Listen>
</Plugin>

<Plugin virt>
  Connection "qemu:///system"
  # RefreshInterval 60
  # Domain "name"
  # BlockDevice "name:device"
  # BlockDeviceFormat target
  # BlockDeviceFormatBasename false
  # InterfaceDevice "name:device"
  # IgnoreSelected false
  HostnameFormat uuid
  # InterfaceFormat name
  # PluginInstanceFormat name
  Instances 1
  ExtraStats "disk pcpu"
</Plugin>

```

Core 10 and 11 were isolated from the OS using the `isolcpus` argument to grub command line and using `virsh` to Pin the vCPUs to those cores (Requires KVM acceleration):

```

$ virsh vcpupin test 0 10
$ virsh vcpupin test 1 11

```

To check for KVM acceleration see that the response to the command below is `> 0`:

```

$ cat /proc/cpuinfo | egrep -c '(vmx|svm)'

```

without KVM acceleration you will need to taskset the qemu threads to the isolated cores:

```
$ grep pid /var/run/libvirt/qemu/test.xml
<domstatus state='running' pid='4907'>
$ grep Cpus_allowed_list /proc/4907/task/*/status
/proc/4907/task/4916/status:Cpus_allowed_list: 4
/proc/4907/task/4917/status:Cpus_allowed_list: 5
/proc/4907/task/4918/status:Cpus_allowed_list: 6
/proc/4907/task/4919/status:Cpus_allowed_list: 7
</section>
```

To taskset the process + tasks repeat the following for the PID and the subtasks:

```
$ taskset -pc 4 4907...
```

Guest

OS: Ubuntu 16.04.2 LTS

Install collectd and stress

```
$ sudo apt install collectd stress stress-ng
```

configure /etc/collectd/collectd.conf

```
Hostname "ubuntu_vm"

<LoadPlugin cpu>
    Interval 1
</LoadPlugin>

LoadPlugin network

<Plugin cpu>
    ReportByCpu true
    ReportByState false
    ValuesPercentage false
</Plugin>

<Plugin network>
#    # client setup:
    Server "<HOST IP>" "28597"
    <Server "<HOST IP>" "28597">
    </Server>
</Plugin>
```

Start collectd on the host and the guest

```
$ service collectd start
```

Run Stress in the VM on a single core

```
$ taskset 1 stress -c 2 -t 10
stress: info: [1422] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [1422] successful run completed in 10s
```