

# Hard Partitioning with Oracle Linux KVM

Hard Partitioning Implementation with Oracle Linux KVM and Oracle  
Linux Virtualization Manager

WHITE PAPER / OCTOBER 11, 2019

## INTRODUCTION

This document describes hard partitioning with Oracle Linux KVM in conjunction with Oracle Linux Virtualization Manager, and how to use it to conform to the [Oracle Licensing policies for partitioned environments](#).

## CPU CORES AND CPU THREADS

On an x86-based system, a CPU core (no hyperthreading enabled) or a CPU thread (hyperthreading enabled) within a core is presented as a physical CPU by the hypervisor or the bare metal operating system. vCPUs (virtual CPUs) are exposed to the guest virtual machine as CPUs: the guest schedules applications on these vCPUs, and the hypervisor schedules these vCPUs over the physical CPU cores or threads. All vCPUs from a guest are symmetrical. Oracle Linux KVM treats these equally, as long as scheduling parameters such as using CPU pinning have not changed.

Oracle Linux KVM offers an advanced feature for hard partitioning, also known as CPU pinning. Hard partitioning means binding vCPUs to physical CPU threads or cores, and preventing these vCPUs from being scheduled on physical CPUs - threads or cores other than the ones specified.

## ORACLE HARD PARTITIONING LICENSING

To conform to the Oracle hard partition licensing requirement, you must follow the instructions described in this white paper to bind vCPUs to physical CPU threads or cores.

Live migration of CPU pinned virtual machines to another Oracle Linux KVM node is not permitted under the terms of the hard partitioning license. Consequently, the cluster that is a pool of Oracle Linux KVM nodes with shared storage must not be configured with any scheduling policy available on Oracle Linux Virtualization Manager.

When live migration is used with pinned virtual machines running Oracle software in an Oracle Linux KVM cluster, hard partition licensing for Oracle software is not applicable. You must determine the number of virtual machines running the Oracle software and then license the same number of physical servers (starting with the largest servers based on the CPU core count) up to the total number of the physical servers in the cluster. For example, if a customer has a cluster with 32 servers and 20 virtual machines running Oracle software within the cluster, the customer must license the 20 largest physical servers in the cluster. If the customer is running 50 virtual machines with Oracle software in a cluster of 32 physical servers, they need only to license the 32 physical servers in the cluster.

Live migration of other virtual machines with non-Oracle software within the server pool is not relevant to Oracle software hard partitioning or has no impact to how Oracle software license is calculated.

“Trusted Partitions” allow subset licensing without limitation on live migration, but only available on the approved Oracle Engineered Systems listed on [Oracle licensing policies for partitioned environments](#).

## UNDERSTANDING CPU TOPOLOGY IN ORACLE LINUX KVM

On a server running Oracle Linux KVM, you can run the "**lscpu**" command to print out the basic CPU configuration of the server hardware. Look for the lines below in the output for detail on your system's CPUs.

```
# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 104
On-line CPU(s) list:   0-103
Thread(s) per core:    2
Core(s) per socket:    26
Socket(s):              2
NUMA node(s):          2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  85
Model name:             Intel(R) Xeon(R) Platinum 8167M CPU @ 2.00GHz
Stepping:               4
CPU MHz:                1820.519
CPU max MHz:            2400.0000
CPU min MHz:            1000.0000
BogoMIPS:               4000.00
Virtualization:         VT-x
L1d cache:              32K
L1i cache:              32K
L2 cache:               1024K
L3 cache:               36608K
NUMA node0 CPU(s):     0-25,52-77
NUMA node1 CPU(s):     26-51,78-103
```

Some considerations on the output above:

- This server has 2 sockets with 26 cores and 2 threads per core. Total of 104 "CPUs"
- CPU 0..103 is really thread 0..103
- NUMA node0 includes threads 0-25 and 52-77
- NUMA node1 includes threads 26-51 and 78-103

### Get the CPU Topology for vCPU Bindings to Physical CPUs

The command "**virsh**", available on an Oracle Linux KVM Server managed by Oracle Linux Virtualization Manager, can be executed in read-only mode. With this command we can first generate the list of active virtual machines running locally on the server:

```
# virsh --readonly list
 Id Name State
-----
 5 win10-b1803 running
 7 ol17-uek5-it running
 8 ol16u9-kvm-it running
 9 debian-9.3 running
```

With the same “**virsh**” command we can also easily check if the CPUs for a virtual machines are pinned to a physical thread/core:

```
# virsh --readonly vcpuinfo ol7-uek5-it --pretty
VCPU:      0
CPU:       1
State:     running
CPU time:  7108.2s
CPU Affinity: 0-7 (out of 8)

VCPU:      1
CPU:       3
State:     running
CPU time:  7081.8s
CPU Affinity: 0-7 (out of 8)
```

On this example the virtual machine is configured with 2 vCPUs and both can span from thread/core 0 until thread/core 7 (out of 8); so, the virtual CPUs for this virtual machines are not pinned to physical threads/cores.

Virtual machines with vCPUs pinned to physical threads/cores, can be easily identified with the limited “CPU Affinity” compared to the total number of physical threads/cores (out of number):

```
# virsh --readonly vcpuinfo ol8-rc1 --pretty
VCPU:      0
CPU:       4
State:     running
CPU time:  17.5s
CPU Affinity: 3-4 (out of 8)

VCPU:      1
CPU:       4
State:     running
CPU time:  1.6s
CPU Affinity: 3-4 (out of 8)
```

## CONFIGURING HARD PARTITIONING

### Setting Hard Partitioning using the Oracle Linux Virtualization Manager Utility

The server running Oracle Linux KVM must meet the requirements as Oracle Linux KVM compute host defined in the Oracle Linux Virtualization Manager Installation Guide. You must use the **olvm\_vmcontrol** utility to set and get the CPU/vCPU bindings for a virtual machine running on Oracle Linux KVM through the Oracle Linux Virtualization Manager.

The **olvm\_vmcontrol** utility is available in Oracle Linux 7 channel from [Oracle Linux Yum Server](#) or [Unbreakable Linux Network \(ULN\)](#). Please review the man page file in the RPM for additional information on the utility and the commands that can be executed and the [Oracle Linux Virtualization Manager Documentation](#).

The **olvm\_vmcontrol** utility can run on the host running Oracle Linux Virtualization Manager or a separate Oracle Linux host that has connectivity to the Oracle Linux Virtualization Manager. Before running the utility, make sure that the utility version matches the version of the Oracle Linux Virtualization Manager.

To install the utility, you first need to install the required RPM containing them:

```
# yum install olvm-vmcontrol
```

Once installed the “olvm-vmcontrol” utility is available on the Linux system path:

```
# which olvm-vmcontrol
/usr/bin/olvm-vmcontrol
```

The olvm-vmcontrol utility requires authentication to the Oracle Linux Virtualization Manager for each operation executed. If a large number of operations have to be executed and if you are running the olvm-vmcontrol utility on the same Oracle Linux Virtualization Manager host, it's suggested to setup an environment variable dedicated to the password:

```
# export OLVMUTIL PASS=<password>
```

By specifying the “-e” option and with this environment setting in-place, each operation executed will not require authentication.

In the example below, a virtual machine named **ol76** accepts the action **getvcpu** to show that virtual CPUs of this guest are not bound to any specific thread or core.

```
# olvm-vmcontrol -m ovirt.it.oracle.com -u admin@internal -v ol76 -c
getvcpu -e
Oracle Linux Virtualization Manager VM Control Utility 4.2.1-8
Connected to Oracle Linux Virtualization Manager 4.2.8.2-1.0.8.el7
Error: VM migration mode must be set to PINNED ("Do not allow migration")
and the VM needs to be pinned to a single host.
Migration mode is MIGRATABLE
Number of hosts VM is pinned to = 0
See man page for more details.
```

Let's bind the vCPUs to core 0 by running the following command:

```
# olvm-vmcontrol -m ovirt.it.oracle.com -u admin@internal -v ol76 -c
setvcpu -s 0 -e
Oracle Linux Virtualization Manager VM Control Utility 4.2.1-8
Connected to Oracle Linux Virtualization Manager 4.2.8.2-1.0.8.el7
Setting vcpu pinning ...
Trying to pin virtual cpu # 0
Trying to pin virtual cpu # 1
Retrieving vcpu pinning to confirm it has been set...
vcpu 0 pinned to cpuSet[0]
vcpu 1 pinned to cpuSet[0]

NOTE: if the VM is running you must now stop and then start the VM from the
Oracle Linux Virtualization Manager in order for CPU pinning changes to
take effect.
NOTE: a restart or a reboot of the VM is not sufficient to put CPU pinning
changes into effect.
```

Once you have configured a virtual machine for CPU pinning, you need to stop the virtual machine and then start it again before the CPU pinning can take effect; With the virtual machine properly restarted we can verify that the proper vCPU pinning is in place:

```
# olvm-vmcontrol -m ovirt.it.oracle.com -u admin@internal -v ol76 -c
getvcpu -e
Oracle Linux Virtualization Manager VM Control Utility 4.2.1-8
Connected to Oracle Linux Virtualization Manager 4.2.8.2-1.0.8.el7
Getting vcpu pinning ...
vcpu 0 pinned to cpuSet[0]
vcpu 1 pinned to cpuSet[0]
```

On the server running Oracle Linux KVM, we can also execute the “virsh” command to find out the CPU pinning status:

```
# virsh --readonly vcpuinfo ol76 --pretty
VCPU:      0
CPU:       0
State:     running
CPU time:  2.7s
CPU Affinity: 0 (out of 8)

VCPU:      1
CPU:       0
State:     running
CPU time:  0.0s
CPU Affinity: 0 (out of 8)
```

The virtual machine now has **CPU Affinity 0** for both virtual CPUs.

## CONCLUSION

With Oracle Linux Virtualization Manager and Oracle Linux KVM, to conform to the Oracle hard partition licensing requirement, you must bind a virtual machine to physical CPUs or cores. This prevents the software from running on physical cores other than the ones specified. In such a case, virtual machines are configured with dedicated CPU resources instead of the default of resource scheduling, which is to use all available CPUs of the server. Using hard partitioning to limit Oracle product software licensing also adds some restrictions such as live migration and scheduling policies available on Oracle Linux Virtualization Manager.

For more information about Oracle's virtualization solutions, visit [oracle.com/virtualization](http://oracle.com/virtualization).

## ORACLE CORPORATION

### Worldwide Headquarters

500 Oracle Parkway, Redwood Shores, CA 94065 USA

### Worldwide Inquiries

TELE + 1.650.506.7000 + 1.800.ORACLE1

FAX + 1.650.506.7200

[oracle.com](http://oracle.com)

## CONNECT WITH US

Call +1.800.ORACLE1 or visit [oracle.com](http://oracle.com). Outside North America, find your local office at [oracle.com/contact](http://oracle.com/contact).

 [blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)

 [facebook.com/oracle](http://facebook.com/oracle)

 [twitter.com/oracle](http://twitter.com/oracle)

## Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1019

White Paper Hard Partitioning with Oracle Linux KVM  
October 2019