



Oracle Linux Virtualization Manager High Performance VMs

Por que usar o OLVM?

Marcos Sungaila

Sr Technical Product Manager

Oracle Linux & Virtualization

24.junho.2023



Patrocinadores DBA Brasil Data & Cloud 2023

A sponsorship grid for the DBA Brasil Data & Cloud 2023 event. The grid is organized into five horizontal tiers, each with a red header bar on the left. The tiers are: Diamante, Platina, Ouro, Prata, and Apoio. Each tier contains logos of various sponsors. The background is dark blue with a subtle grid pattern and a decorative white pattern in the top right corner.

Tier	Sponsors
Diamante	AGGRANDIZE, COMMAVAULT, TD SYNTEX
Platina	GOLDENGATEBR, DISCOVER
Ouro	scansource, VERTICA by opentext
Prata	TRACES, CAFÉ COM CLOUD, ROX
Apoio	FIAP, GRUPO POSEIDON DIGITAL





Marcos Sungaila

Sr Technical Product Manager
Oracle Linux & Virtualization

- Linux user desde 1994
- Trabalho profissionalmente com Linux desde 1998
- Comecei com Virtualização em 2005
- Virtualização com Open Source desde 2008



Agenda

- Oracle Linux & Oracle Linux Virtualization Manager Overview
- High-Performance VMs
- CPU pinning
- NUMA
- VirtIO, VirtIO-SCSI
- IO Threads
- KSM
- Huge pages



Oracle Linux & Oracle Linux Virtualization Manager

—
Overview

Oracle está comprometida com o Linux e open source

Platinum member da
Linux Foundation



Platinum member da
Cloud Native Computing Foundation



Parte do compromisso mais amplo da Oracle com o Código Aberto.



GraalVM™



VirtualBox



VERRAZANO

Oracle Linux – mais de 20 anos de contribuição e crescendo

Download e uso livres desde 2006

1998

- Primeiro RDBMS comercial para Linux
- Primeiro port x64 para Linux
- Lançamento do Unbreakable Linux

Oracle Linux livre para download e uso (2006)

- Anúncio do suporte a Oracle Linux em 2006
- Exadata redesenhado como membro fundador
- Unbreakable Enterprise Linux

Oracle Linux errata downloads aberto (2012)

- Ksplice zero-downtime patching
- DTrace dynamic tracing
- Oracle se associa à Open Container Initiative
- Oracle se associa à CNCF como Platinum member
- Oracle Linux for Arm
- Primeiro Linux com certificação NIAP
- Primeiro Autonomous Linux na cloud
- Oracle Linux Automation Manager

2023

Oracle Linux



Download gratuito

Incluindo updates. Sem taxas de licença ou subscrições



Compatível com RHEL e mais melhorias

Mais de 16 anos. Zero bugs de compatibilidade registrados



Desenhado para todos workloads

Oracle e aplicações de terceiros, DBs, Clouds, x86 & Arm



Virtualização com Oracle Linux KVM

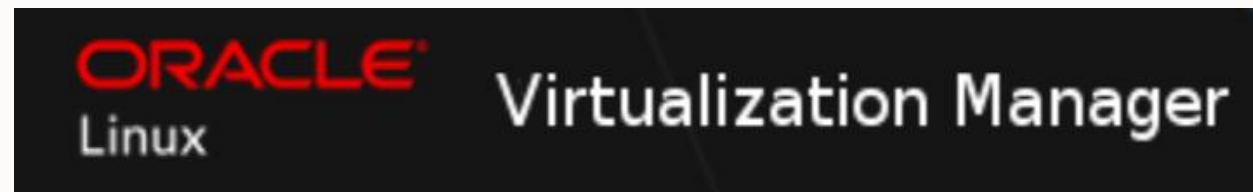
Baseada em Oracle Linux 8



- Oracle Linux Virtualization Manager (Gerenciamento)
 - Baseada no projeto oVirt 4.4
 - Oracle Linux 8 – Premier Support through July 2029
 - Novas features oferecem eficiência operacional e usabilidade
- Oracle Linux KVM (Compute)
 - Oracle Linux 8 KVM host (UEK ou RHCK)
 - Oracle Linux 7 KVM host
- Oracle VirtIO 2.0 drivers for Microsoft Windows

ORACLE
Linux
KVM

ORACLE
Linux
Virtualization Manager

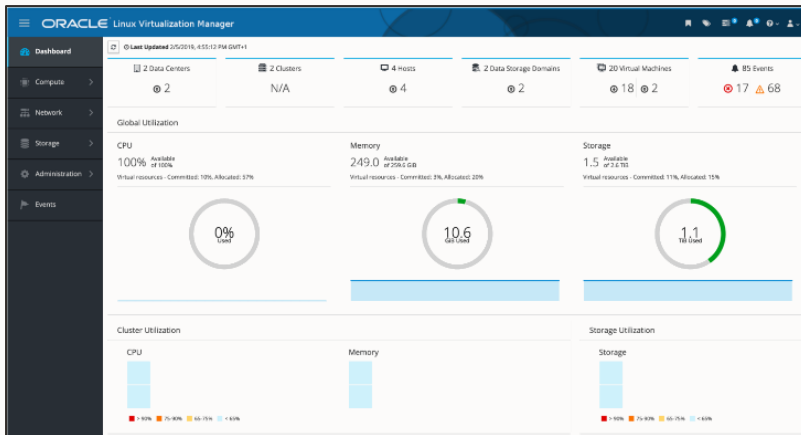


Oracle Linux Virtualization Manager

Overview da solução



Oracle Linux Virtualization Manager



yum.oracle.com/OL8/ovirt44

yum.oracle.com/OL8/ovirt44/extras



Oracle Linux KVM
Server - 7.9



Oracle Linux KVM
Server - 8.7



Oracle Linux KVM
Server - 8.8

Guest OS Support

- Oracle Linux 6/7/8/9
- RHEL 6/7/8/9
- CentOS 6/7/8
- SLES 12/15
- Ubuntu 18.04/20.04/22.04
- Solaris x86 11.4
- Windows Desktop 7, 8, 10, 11
- Windows Server 2008, 2012, 2016, 2019, 2022





Oracle Virtualization no Oracle Linux

Todos os benefícios da tecnologia KVM e oVirt, mais:

- + Permite aumentar a segurança e disponibilidade com Oracle Ksplice
- + Permite o rápido deploy de aplicações com templates para toda a pilha de softwares
- + Certificado para softwares Oracle
- + Desenhado para Multi-Cloud
- + Gerencie Licenças de software Oracle com hard partitioning
- + Incluído na subscrição Oracle Linux Premier
- + Sem vendor lock-in

VMs High-Performance

O que são High-Performance VMs e por que usar



- Perfis de otimização: Desktop, Server, e **High Performance**
 - Hardware virtual otimizado para melhor eficiência.
- Por que usar esse perfil específico?
 - Performance próxima de uma máquina física.
 - Melhor eficiência



Por que a otimização de performance importa em Virtualização

- No KVM, as VMs processos no host.
- Host virtualiza ou emula hardware virtual.
- Eficiência do hardware virtual.
- Recursos alocados x performance esperada.
- Overhead em hardwares virtuais.
- Impacto da configuração do hardware virtual.
- Uso eficiente dos recursos do host.

Atributos a considerar quando pensamos em Performance



- Virtual CPUs (vCPUs).
- Configurações de NUMA e Huge Pages.
- Impacto do I/O na performance.
- Emulação em hardwares específicos.

Recursos configurados automaticamente para uma VM do tipo High-Performance



Configurações definidas automaticamente para High-Performance VMs:

- Modo Headless e Serial Console habilitados
- Todos dispositivos USB desabilitados
- Placa de som desabilitada
- Smart Card desabilitado
- Host CPU Pass-Trough habilitado
- VM Migration desabilitada
- IO Threads habilitadas, com número de IO Threads = 1
- Memory Balloon desabilitada
- Habilita High-Availability apenas para os hosts com CPU pinning
- Watchdog desabilitado



Recursos configurados automaticamente para uma VM do tipo High-Performance – continuação



Configurações definidas automaticamente para High-Performance VMs:

- Paravirtualized Random Number Generator PCI (virtio-rng) habilitado
- Multi-queues para Interfaces Virtuais
- Configura a topologia de pinning de IO e Emulator Threads



Configurações adicionais que podem ser feitas manualmente



Configurações manuais:

- Configurar a topologia de CPU pinning
- Habilitar NUMA virtual e configurar a topologia de pinning NUMA
- Desabilitar Kernel SamePage Merging (KSM)
- Habilitar Huge Pages



CPU pinning

CPU pinning



O que é CPU pinning?

- Controlar em quais cores físicos um processo deve ser executado.
- Permite executar uma CPU virtual (vCPU) em um core específico da CPU física (pCPU) no host.

Por que CPU pinning é usado?

- Processos alternam entre cores da CPU.
- Recarregar cache, e memória do processo.

CPU pinning – um exemplo

CPU Allocation:

CPU Profile

CPU Shares

CPU Pinning topology

Memory Allocation:

Memory Balloon Device Enabled

I/O Threads:

I/O Threads Enabled

Queues:

Multi Queues enabled

VirtIO-SCSI Enabled



CPU pinning – um exemplo em linha de comando

```
<memory unit='KiB'>8290304</memory>
<currentMemory unit='KiB'>8290304</currentMemory>
<vcpu placement='static' current='2'>16</vcpu>
<iothreads>1</iothreads>
<b>cputune</b>
  <b>vcpupin vcpu='0' cpuset='16' />
  <b>vcpupin vcpu='1' cpuset='8' />
  <b>emulatorpin cpuset='0-1' />
  <b>iothreadpin iothread='1' cpuset='0-1' />
</cputune>
<numatune>
  <memnode cellid='0' mode='interleave' nodeset='0' />
</numatune>
```



NUMA



NUMA



Non-Uniform Memory Access – NUMA

- Máquinas com mais de uma CPU, banco de memória ligado diretamente a um socket.

Por que NUMA é importante?

- Acesso mais rápido a dados locais.
- Usar NUMA vai melhorar a performance.
- Preferível executar os processos em NUMA nodes.

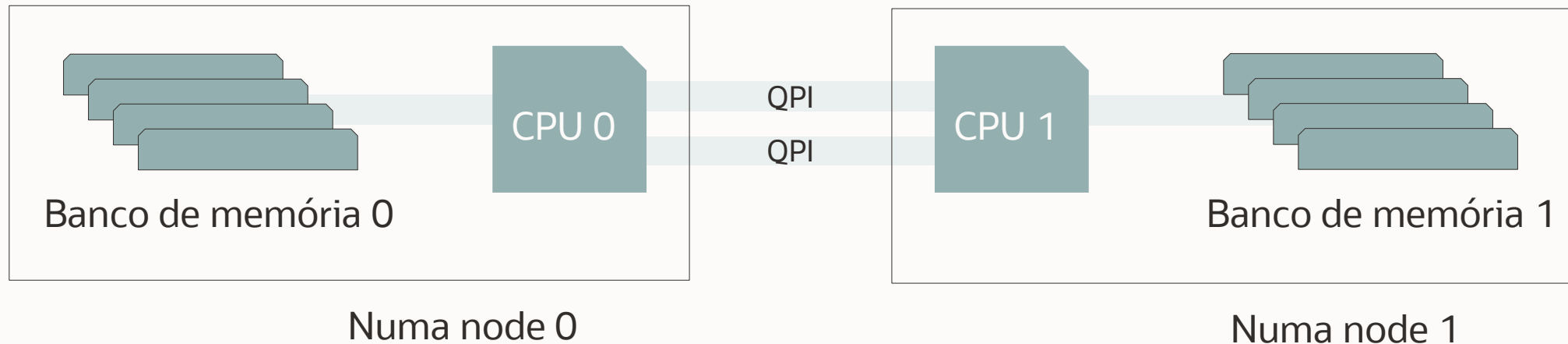
NUMA – continuação



O que é um Sistema Linux NUMA?

- Conjunto de CPU, memória local e/ou barramentos IO.
- Cada NUMA node é um subconjunto SMP.
- NUMA nodes estão interconectados – QPI

Visão básica de um arquitetura NUMA



vNUMA & NUMA Autobalance



O que é vNUMA?

- Arquitetura NUMA apresentada ao sistema operacional da VM.
- O guest agenda o processo considerando a arquitetura NUMA subjacente.
- Tech Preview no OLVM 4.4

NUMA Balancing

- Melhora o desempenho.
- Move tasks (threads ou processos) para mais perto da memória que estão acessando.
- Pode mover os dados da aplicação para a memória mais próxima das tasks que os usam.
- Feito automaticamente pelo kernel quando o balanceamento NUMA automático está ativo.



oVirt NUMA settings



oVirt NUMA settings

- Para definir os nós NUMA e a topologia de pinning, você precisa de um host com pinning habilitado para NUMA com pelo menos dois nós NUMA.
 - Pelo menos duas CPUs.
 - Um banco de memória para cada CPU.
- **Como fazer isso no OLVM:**
 - Na guia Host, selecione NUMA Node Count e Tune Mode na dropdown list.
 - Clique em NUMA Pinning.
 - Na janela NUMA Topology, clique e arraste os nós NUMA virtuais da caixa à direita para os nós NUMA físicos do host à esquerda, conforme necessário.

NUMA Node – Exemplo

Start Running On:

Any Host in Cluster

Specific Host(s) ovs245,ovs246

Migration Options:

Migration mode ⓘ Allow manual migration only

Use custom migration policy ⓘ Minimal downtime

Use custom migration downtime ⓘ

Auto Converge migrations Inherit from cluster setting

Enable migration compression Inherit from cluster setting

Pass-Through Host CPU ⓘ

Configure NUMA: ⓘ

NUMA Node Count 1

Tune Mode Interleave

NUMA Pinning



NUMA Node – Exemplo – continuação



KVM Host

```
# lscpu | grep NUMA
NUMA node(s) :          2
NUMA node0 CPU(s) :    0-7,16-23
NUMA node1 CPU(s) :    8-15,24-31
```

NUMA Node – Exemplo – continuação

The screenshot displays the 'NUMA Topology - ovs245' window. At the top, it shows a summary: 'Totals: 32 CPUs 31767 MB 2 NUMA' with '0% Used' and '30804 MB used'. Below this, a list of vNUMA nodes is shown, including 'ol7u6-myDC_NUMA0'. The interface is divided into two main sections: 'Socket 0' and 'Socket 1'.
- **Socket 0:** Contains 'NUMA 0' with '16 CPUs' and '15679 MB' (0% Used, 2488 MB used). It has one vNUMA node, 'ol7u6-myDC_NUMA0', which is currently pinned to this socket.
- **Socket 1:** Contains 'NUMA 1' with '16 CPUs' and '16087 MB' (0% Used, 841 MB used). It has zero vNUMA nodes.
On the right side, a panel titled 'Unassigned virtual nodes' shows a list with 'ol7u6-myDC' (ID 0) and a dashed box indicating where it can be dragged to be pinned to a NUMA node.



VirtIO, VirtIO-SCSI, e IO Threads

VirtIO



VirtIO

- Dispositivo para-virtualizado de alta performance para KVM.
- Design é simples, mas tem limitações:
 - SCSI Pass-Through limitado
 - sem acesso a recursos avançados
 - um dispositivo PCI por disco.

VirtIO-SCSI



VirtIO-SCSI

- Design eficiente do VirtIO-Blk.
- Tem acesso a paths múltiplos, SCSI Pass-Through efetivo.
- Escalabilidade quase ilimitada.
- Cada controlador PCI pode suportar centenas de dispositivos de disco e é a opção recomendada no OLVM.
- Principal vantagem:
 - Pode lidar com centenas de dispositivos.
- Impacto na performance quando tem vários discos na mesma controladora.



IO Threads



IO Threads

- Threads às quais os dispositivos de bloco podem ser pinados.
- Melhorar significativamente o desempenho das VMs.
- IO Threads são executados no Hypervisor.
- No OLVM, para cada VM, há 1 IO Thread independente do número de discos anexados.

VirtIO-SCSI e IO Threads

- Discos VirtIO-SCSI podem usar IO Threads.
- Configurados no controlador SCSI.
- Discos na mesma controladora compartilham o mesmo IO Thread.
- Pode conectar várias controladoras.



KSM



Kernel Same page Merging



KSM – Kernel Same page Merging



KSM

- De duplicação de memória.
- Páginas de memória em todos os NUMA nodes.
- Pode afetar a performance.
- Desativado em VMs do tipo High-Performance.



Huge Pages

Huge Pages



- Gerenciamento de memória virtual:
 - Uma tabela que mapeia o endereço da memória virtual para um endereço físico.
 - Páginas de memória pequenas, mais tabelas de mapeamento. Isso diminui o desempenho.
- Páginas grandes significa tabela menor.
- Mapeamento menor e mais eficiente.
- Aumenta o desempenho.
- No OLVM, os usuários podem definir uma propriedade personalizada chamada hugepages de acordo com suas necessidades.

Referências



- Oracle Linux Virtualization Manager Data Sheet
 - <https://www.oracle.com/a/ocom/docs/oracle-linux-virtualization-manager-ds-final.pdf>

Keep in touch



twitter.com/oraclelinux



facebook.com/oraclelinux



blogs.oracle.com/linux
blogs.oracle.com/virtualization



youtube.com/oraclelinuxchannel



linkedin.com/showcase/oracle-linux



oracle.com/linux
oracle.com/virtualization



ORACLE