Tuning Your SUSE® Linux Enterprise Virtualization Stack

Jim Fehlig
Software Engineer
jfehlig@suse.com

Agenda

- General guidelines
- Network
- Disk
- CPU
- Memory
- NUMA

General Guidelines

- **Minimize software installed on the host**
  - Reduces resources
  - Reduces security risks/increases availability
- **Synchronize time**
  - Use NTP to synchronize time on the host AND virtual machines
- **Consider host resource requirements**
  - Host uses resources too!
- **Avoid over-allocating resources to virtual machines**
- **Remove unneeded devices from virtual machines**
- **Use paravirtual drivers for better performance**

General Guidelines

- **Xen**
  - Disable autoballooning of domain0
    - Xen parameter 'dom0_mem=xxG'
    - /etc/xen/xl.conf: autoballoon=”off”
  - Limit domain0 vcpus
    - Xen parameter 'dom0_max_vcpus=xx'
  - Use tmpfs for xenstore database
    - Default configuration in SLES12 and newer
  - pvops kernel in SLES12 SP2
    - Goodbye kernel-xen, hello kernel-default

Network

- **Use multiple networks to avoid congestion**
  - admin, storage, live migration, ...
  - May require using arp_filter to prevent ARP flux
    - http://linux-ip.net/html/ether-arp.html#ether-arp-flux

    echo 1 > /proc/sys/net/ipv4/conf/arp_filter
- **Same MTU in all devices to avoid fragmentation**

Network

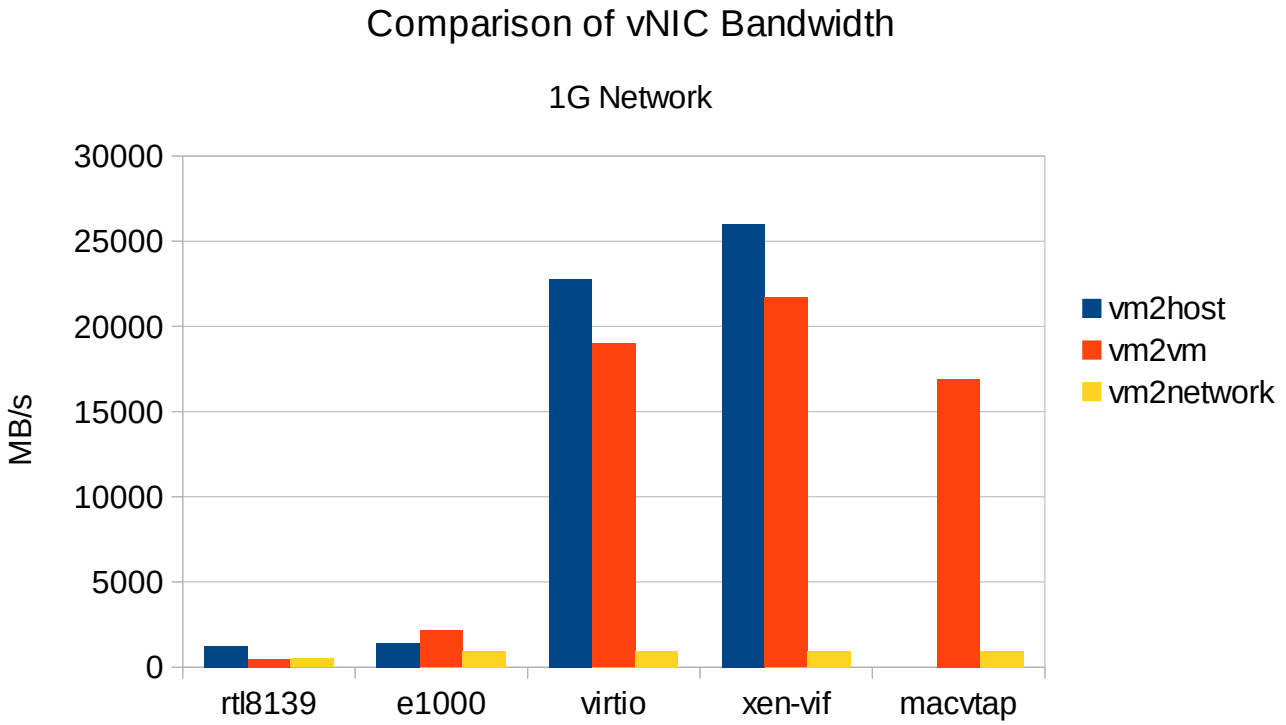
- **Multiqueue-enabled Virtual NICs**
  - virtio (KVM)
    - vhost_net backend
  - xen-vif (Xen)
    - netbk backend (kernel-xen)
    - xen_netback backend (pvops)
- **Emulated NICs**
  - e1000
    - Default and preferred emulated NIC
  - rtl8139

Network

- **Shared physical NICs**
  - SR-IOV
  - macvtap
    - VM ↔ Host communication not possible
- **Passthrough of physical NICs, aka PCI passthrough**
  - Not supported by Intel due to security concerns

Note: These approaches offer increased performance, but may complicate migration

Network


Comparison of vNIC Bandwidth
1G Network
MB/s
30000
25000
20000
15000
10000
5000
0
rtl8139
e1000
virtio
xen-vif
macvtap
vm2host
vm2vm
vm2network

Network

- **virtio**
  - Several tunables available via virtual interface configuration

```
<interface type='bridge'>
  <model type='virtio'/>
  < driver ioeventfd='off|on' queues='4'>
    <host csum='off|on' gso='off|on'/>
    <guest csum='off|on' gso='off|on'/>
  </driver>
  <bandwidth>
    <inbound average='102400' peak='1048576'/>
  </bandwidth>
  …
</interface>
```

Network

- **xen_vif**

```
<interface type='bridge'>
  <model type='netfront'/>
  <bandwidth>
    <inbound average='102400'/>
  </bandwidth>
  …
</interface>
```

  - netbk backend (kernel-xen)
    - netbk.tasklets, netbk.bind, and netbk.queue_length options
  - xen_netback backend (pvops)
    - xen_netback.max_queues option

Disk Devices and Double Vision

- **Two page caches**
  - Two copies of data in memory
- **Two IO schedulers**
  - Guest and host both reordering and delaying IO
  - kernel >= 3.13 has no IO scheduler for virtual devices

    # cat /sys/block/vd[x]/queue/scheduler

    none
- **Possibly two filesystems**
  - Guest filesystem
  - Host filesystem containing the image
- **Possibly two volume managers**
  - Guest and host both using LVM
- **Dr says**
  - Configure guest or host to bypass one of the redundant layers

Disk
Block devices vs Image Files

- **Block devices**
  - Historically better performance
  - Use “standard” tools for administration/disk modification
  - Accessible from host (pro and con)
  - Eliminates one of the file systems
- **Image Files**
  - Easier system management
    - Easier to move, clone, backup
  - Comprehensive toolkit (guestfs) for image manipulation
  - Fully allocated vs sparse
    - Performance vs resource consumption

Disk
Image Files

- **raw**
  - Most common format
  - Historically, best performance
- **qcow2**
  - Required for snapshot support in libvirt + tools
  - Improved performance and stability
    - Compared to older versions of qcow2
- **qed**
  - Next generation qcow
- **vhd/vhdx/vmdk/others**
  - Suggest using for import/export only

Disk

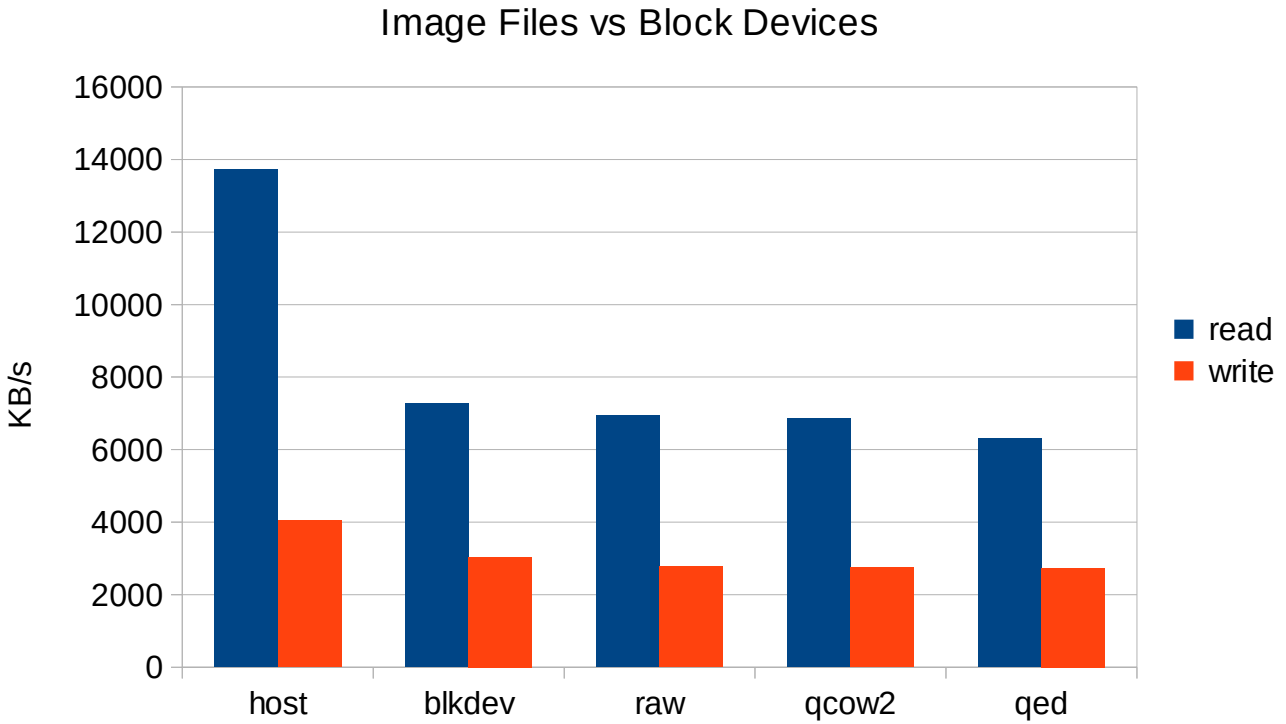


Image Files vs Block Devices
KB/s
16000
14000
12000
10000
8000
6000
4000
2000
0
host
blkdev
raw
qcow2
qed
read
write

Disk
Cache Modes

- **Write-back**
  - Host page cache enabled
  - Writes reported completed when data placed in host page cache
  - VM flush commands honored
  - Default mode in KVM and Xen
- **Write-through**
  - Host page cache enabled
  - Writes reported completed only when data has been committed to storage device
  - VM informed no writeback cache

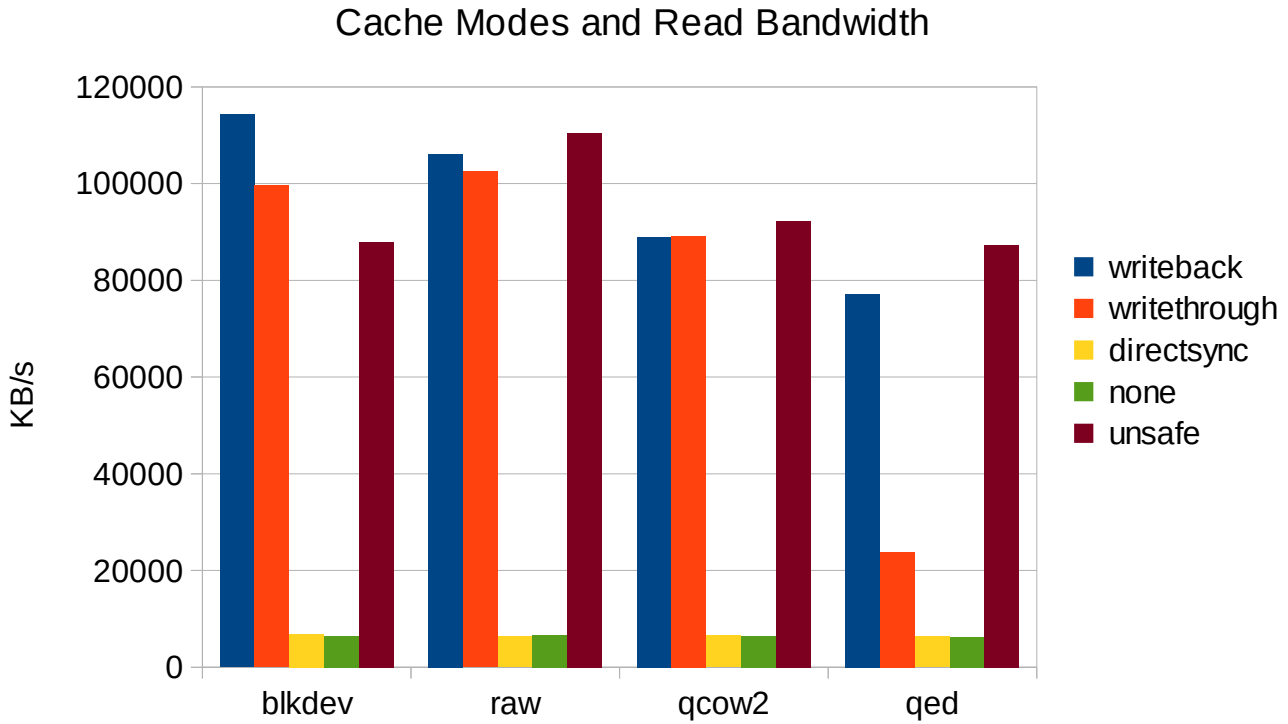
Disk
Cache Modes

- **Directsync**
  - Host page cache disabled
  - Writes reported completed only when data committed to storage device
  - Useful for guests that don't send flush commands
- **none**
  - Host page cache disabled
  - O_DIRECT semantics
  - Guest informed of writeback cache

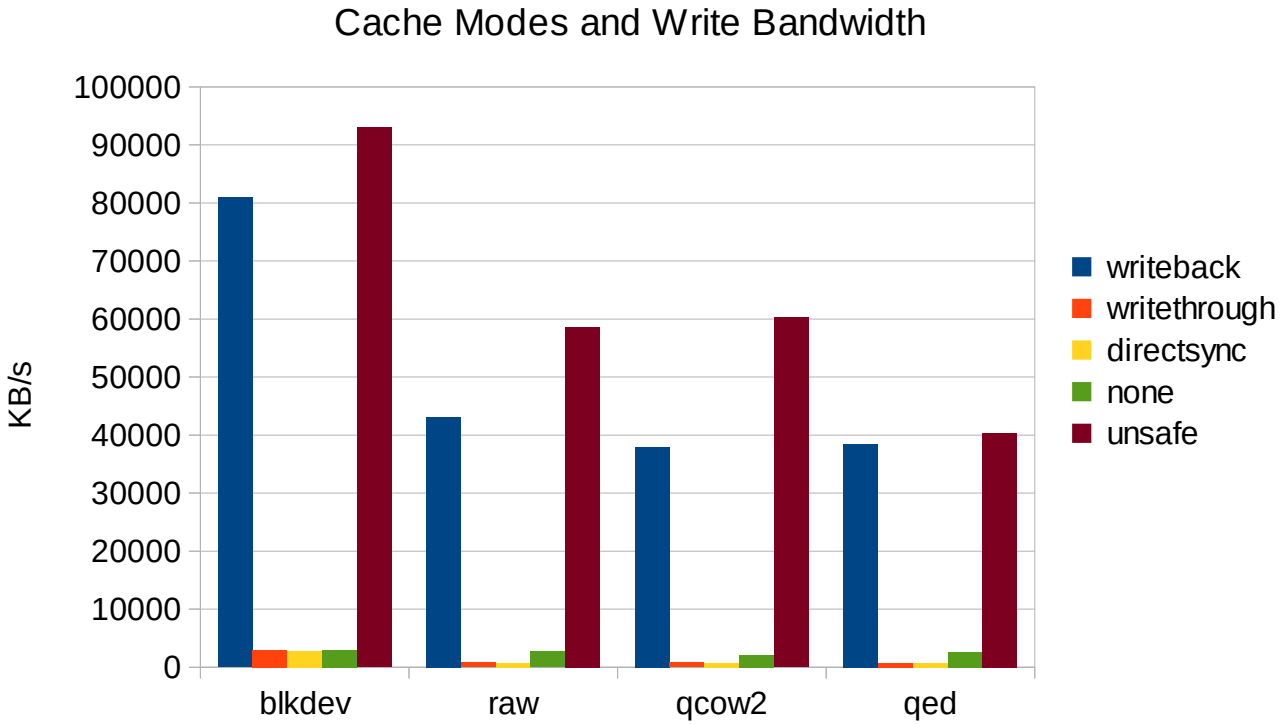
Disk
Cache Modes

- **Unsafe**
  - Host page cache enabled
  - Similar to writeback, except VM flush commands ignored

Disk
Cache Modes



Disk
Cache Modes

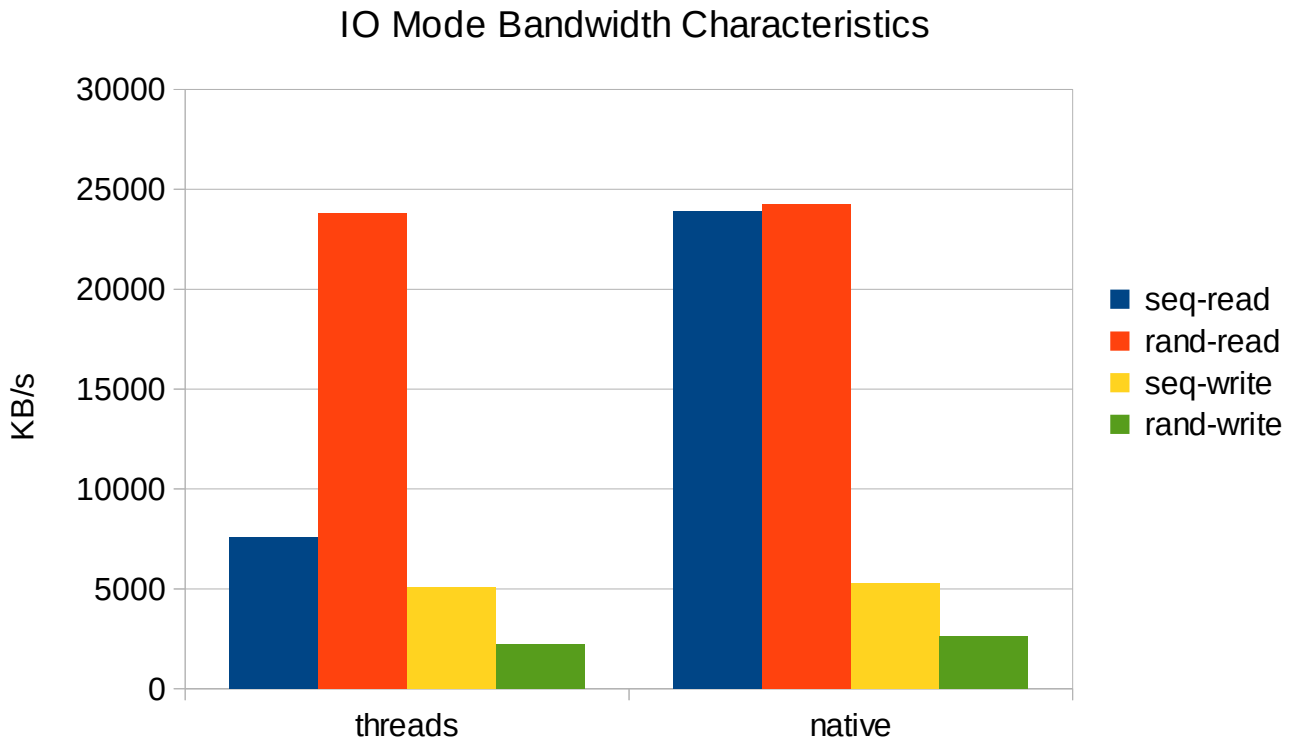


Disk - KVM Specific

- **IO modes**
  - native
    - Linux asynchronous IO
    - Lower CPU overhead
  - threads
    - POSIX asynchronous IO emulation using a pool of worker threads
    - Compatible with all disk types: LVM, block devices, images files
    - Default mode in SLES

```
<disk>
  <driver name='qemu' io='native|threads'/>
  …
</disk>
```

Disk
IO Modes

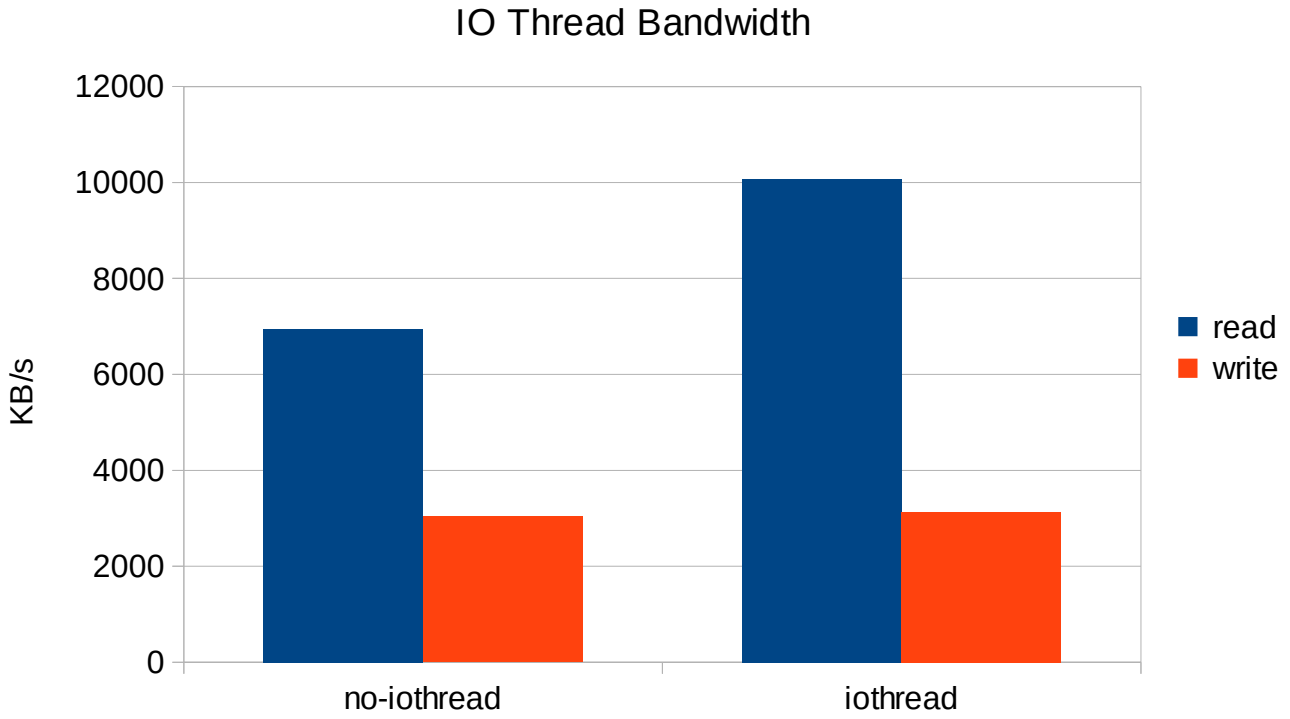


Disk – KVM Specific

- **IO Threads**
  - Dedicated threads for servicing IO requests

```
<iothreads>2</iothreads>
…
<devices>
  <disk>
    <driver name='qemu' iothread='1'/>
    …
    </disk>
    <disk>
      <driver name='qemu' iothread='2'/>
    </disk>
  </devices>
```

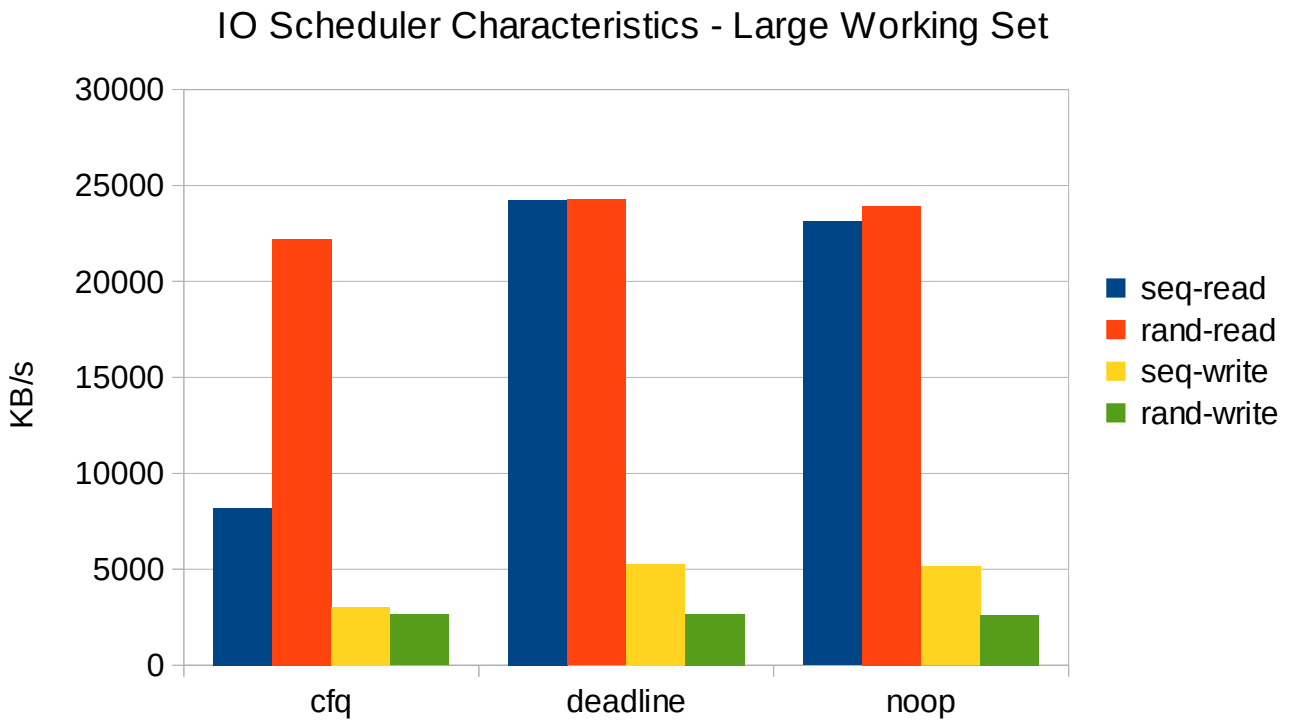
Disk – KVM Specific



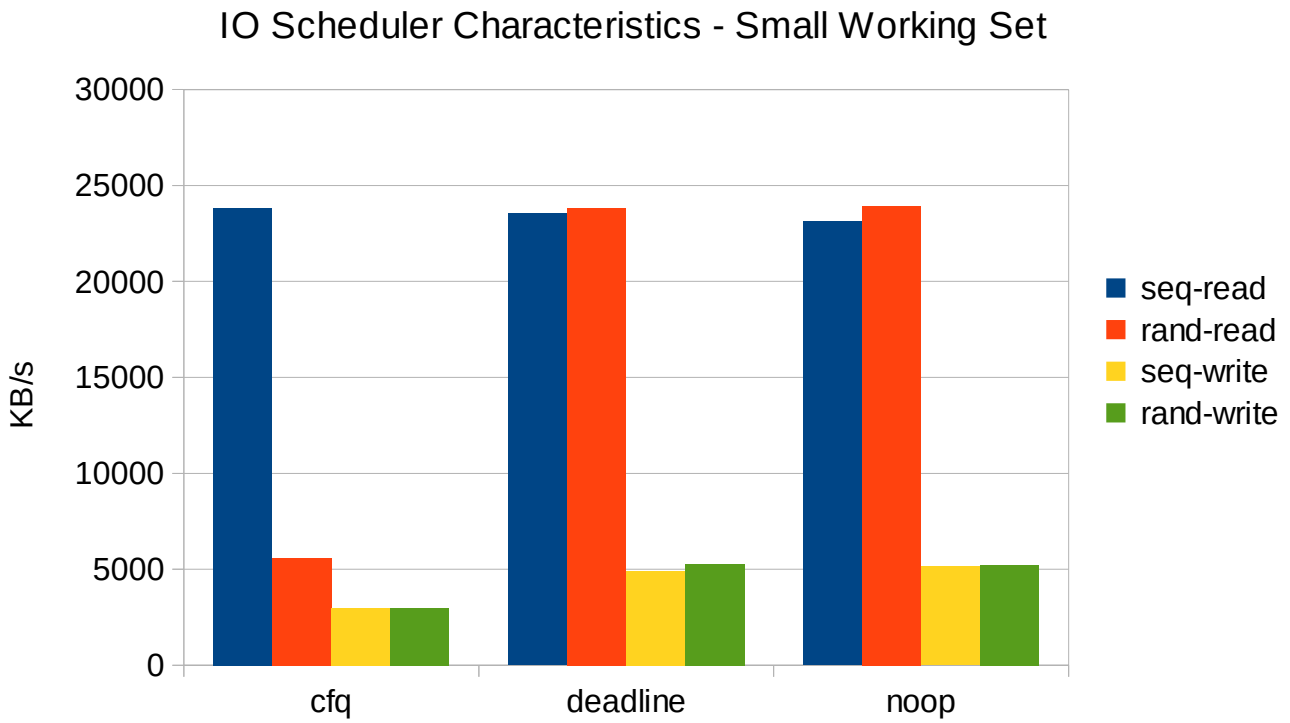
Disk

- **IO scheduler**
  - Completely Fair Queuing (CFQ), deadline, noop, none
  - In kernel >= 3.13
    - Virtual block devices only support 'none'
    - CFQ is default for others
  - In kernel < 3.13 default is CFQ for all block devices
  - Tunable per device
    - echo noop > /sys/block/<device>/queue/scheduler
  - Disable one of the schedulers
    - noop in the VM, deadline in the host
    - noop in the VM, CFQ in the host

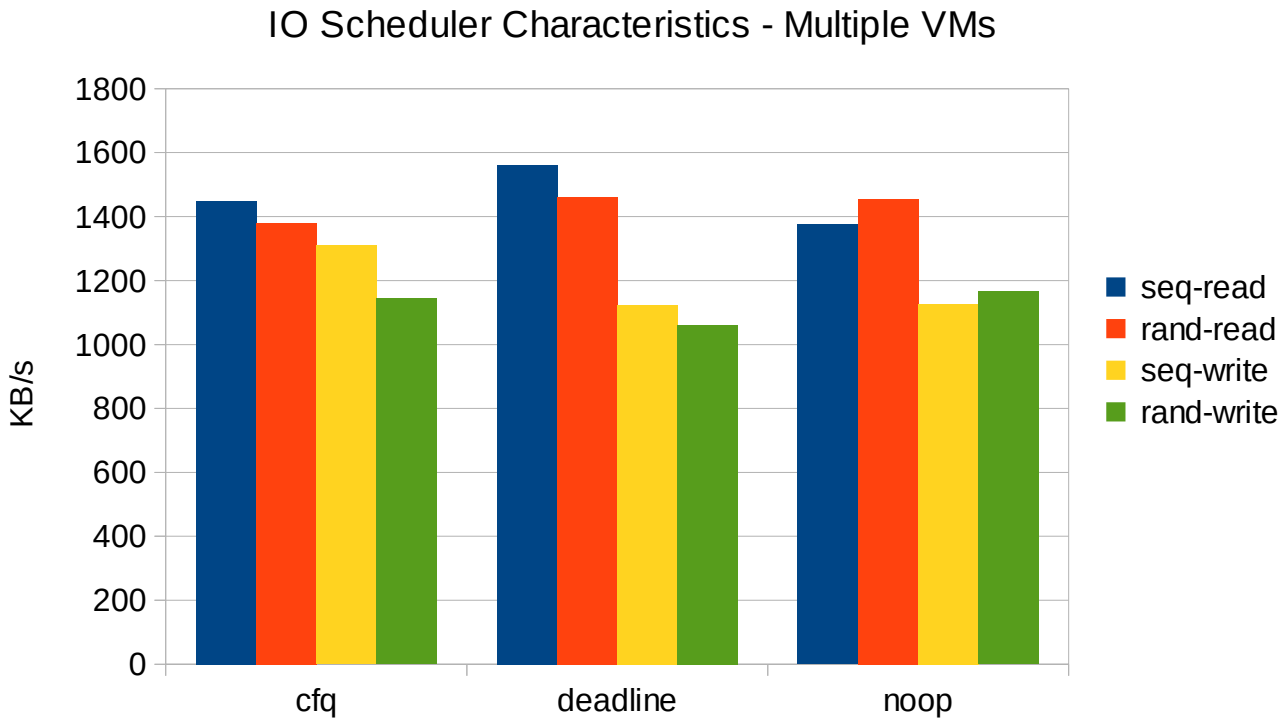
Disk
IO Scheduler



Disk
IO Scheduler



Disk
IO Scheduler



CPU - Host

- **Avoid excessive CPU contention**
  - Due to excessive CPU overcommit or incorrect vCPU pinning
- **Scheduler**
  - Performance vs latency
  - CFS tuned with kernel.sched_* parameters
- **CPU power states**
  - CPU frequency governor
    - cpupower frequency-set -g performance
  - Kernel parameters processor.max_cstate and intel_idle.max_cstate

- **SLES12 Tuning Guide**

https://www.suse.com/documentation/sles-12/book_sle_tuning/data/book_sle_tuning.html

CPU – Virtual Machine

- **vCPU model and features**
  - Normalize to allow migration among heterogeneous hosts
    - virsh capabilities | virsh cpu-baseline /dev/stdin >> all-hosts-cpu-caps.xml
    - virsh cpu-baseline all-hosts-cpu-caps.xml

```
<cpu mode='custom' match='exact'>
  <model fallback='allow'>Nahalem</model>
  <feature policy='require' name='cmt'/>
  ...
</cpu>
```

CPU – Virtual Machine

- **vCPU topology**
  - For smaller VMs (<= 8 vCPUs), multiple sockets with a single core and thread, on the same NUMA node, generally give best performance
  - For larger VMs, topologies that closely resemble the host topology generally give best performance

```
<cpu'>
  <topology sockets='8' cores='1' threads='1'/>
  ...
</cpu>
```

CPU – Virtual Machine

- **vCPU Pinning**
  - Constrain vCPU threads to physical CPUs

```
<cputune>
  <vcpupin vcpu='0' cpuset='0-15'/>
  ...
</cputune>

<cputune>
  <vcpupin vcpu='0' cpuset='0'/>
  <vcpupin vcpu=’1’ cpuset=’1’/>
  ...
</cputune>
```

CPU – Virtual Machine

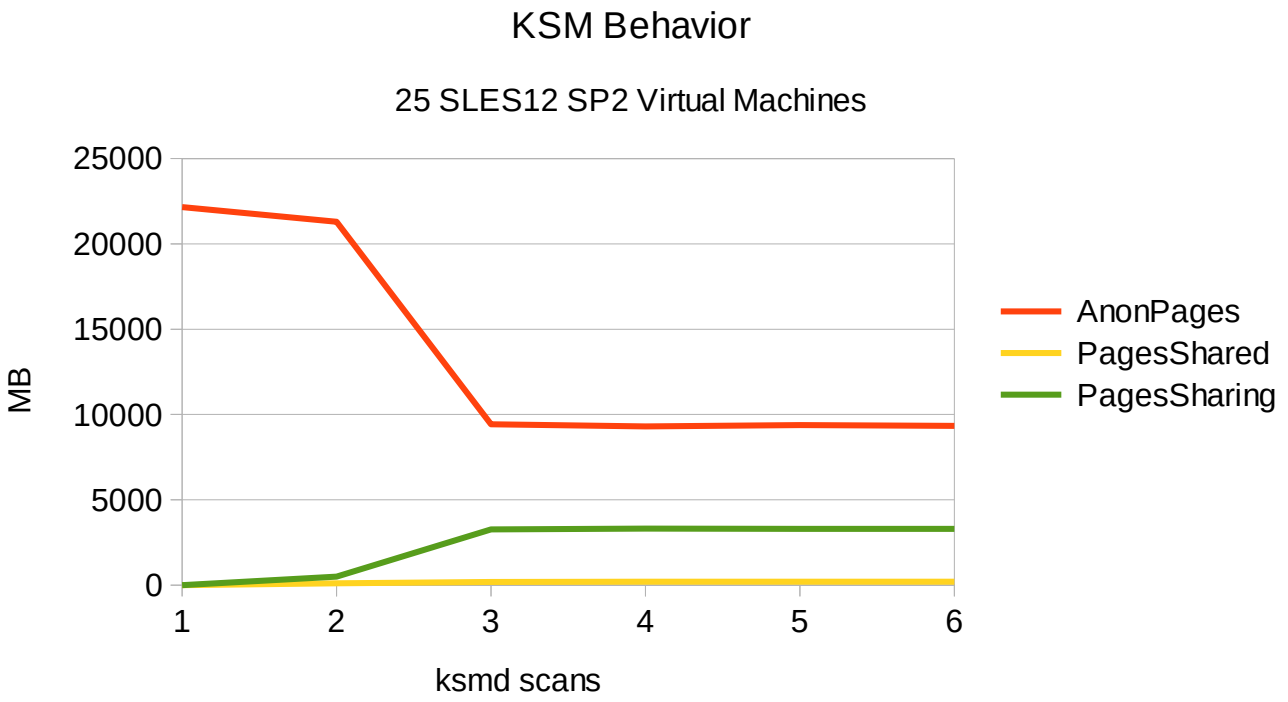
- **vCPU scheduling**
  - Fine tune scheduling of vCPUs

```
<cputune>
  <shares>2048</shares>
  <period>1000000</period>
  <quota>10000000</quota>
  ...
</cputune>
```

Memory - Host

- **Memory overcommit generally not recommended, but...**
- **Kernel Samepage Merging (KSM)**
  - Memory overcommit technique
  - Best results when running multiple instances of same image
  - ksmd thread consumes 5-10% of one core with default settings
  - echo 1 > /sys/kernel/mm/ksm/run
  - /sys/kernel/mm/ksm/pages_to_scan
  - /sys/kernel/mm/ksm/sleep_millisecs
- **Warning: By default, pages common across NUMA nodes are merged**
  - Increased memory access latencies may be observed in VM
  - echo 0 > /sys/kernel/mm/ksm/merge_across_nodes

Memory - KSM


KSM Behavior
25 SLES12 SP2 Virtual Machines
25000
20000
15000
10000
5000
0
MB
1
2
3
4
5
6
ksmd scans
AnonPages
PagesShared
PagesSharing

Memory - Host

- **Transparent Huge Pages (THP)**
  - Enabled by default
  - Anonymous memory and tmpfs/shmem only
  - Warning: May reduce performance of workloads with sparse memory access patterns
    - echo > never /sys/kernel/mm/transparent_hugepage/enabled
- **Huge Pages**
  - Manually control allocation and use of huge pages
    - At boot: hugepagesz=2M hugepages=8192
    - Runtime: echo 8192 > /proc/sys/vm/nr_hugepages
  - Virtual machine configuration

```
<memoryBacking>
  <hugepages/>
</memoryBacking>
```

Memory – Virtual Machine

- **Lock VM pages to prevent swapping**

```
<memoryBacking>
  <locked/>
</memoryBacking>
```

- **Prevent page sharing**

```
<memoryBacking>
  <nosharepages/>
</memoryBacking>
```

NUMA

- **Potentially huge impact on performance**
- **Consider host topology when sizing guests**
  - virsh {nodeinfo, capabilities, freecell}
- **Prevent vCPUs from floating across NUMA nodes**
  - vCPU pinning
- **Avoid allocating VM memory across NUMA nodes**

```
<numatune>
  <memory mode='strict' nodeset='1'/>
</numatune>
```

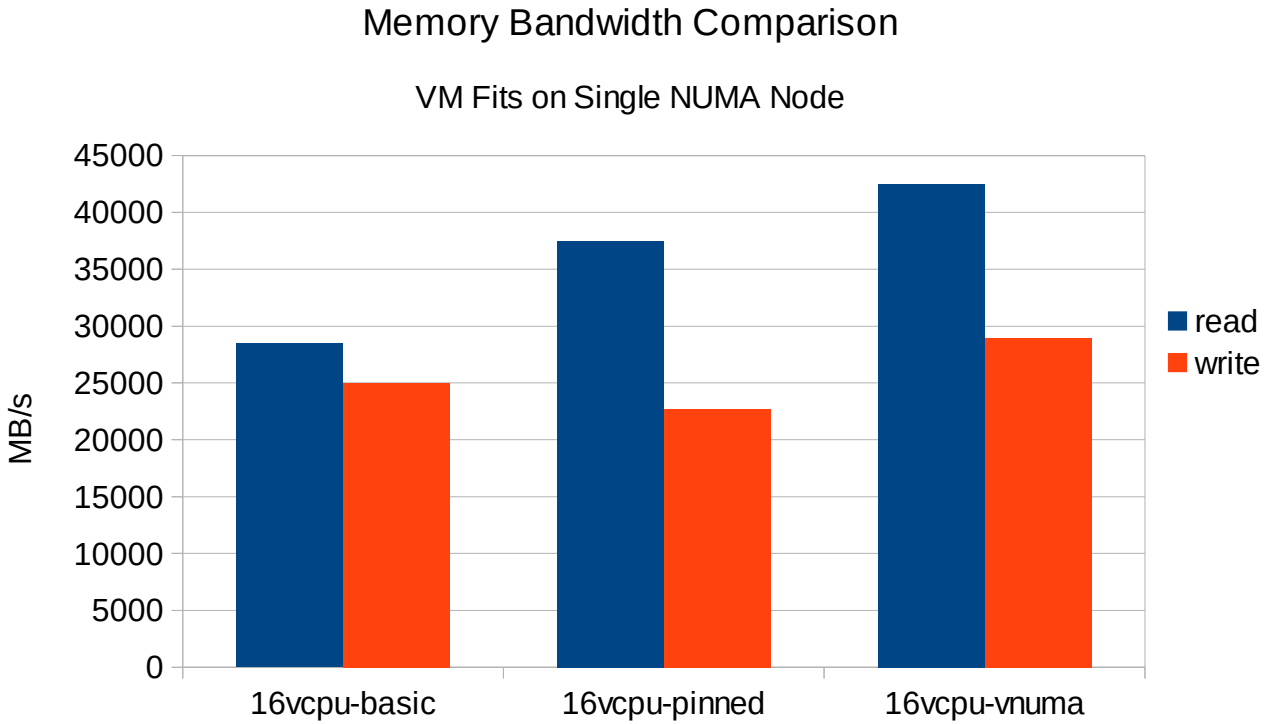
- **Disable NUMA autobalance in host if pinning VM resources**
  - echo 0 > /proc/sys/kernel/numa_autobalancing

NUMA

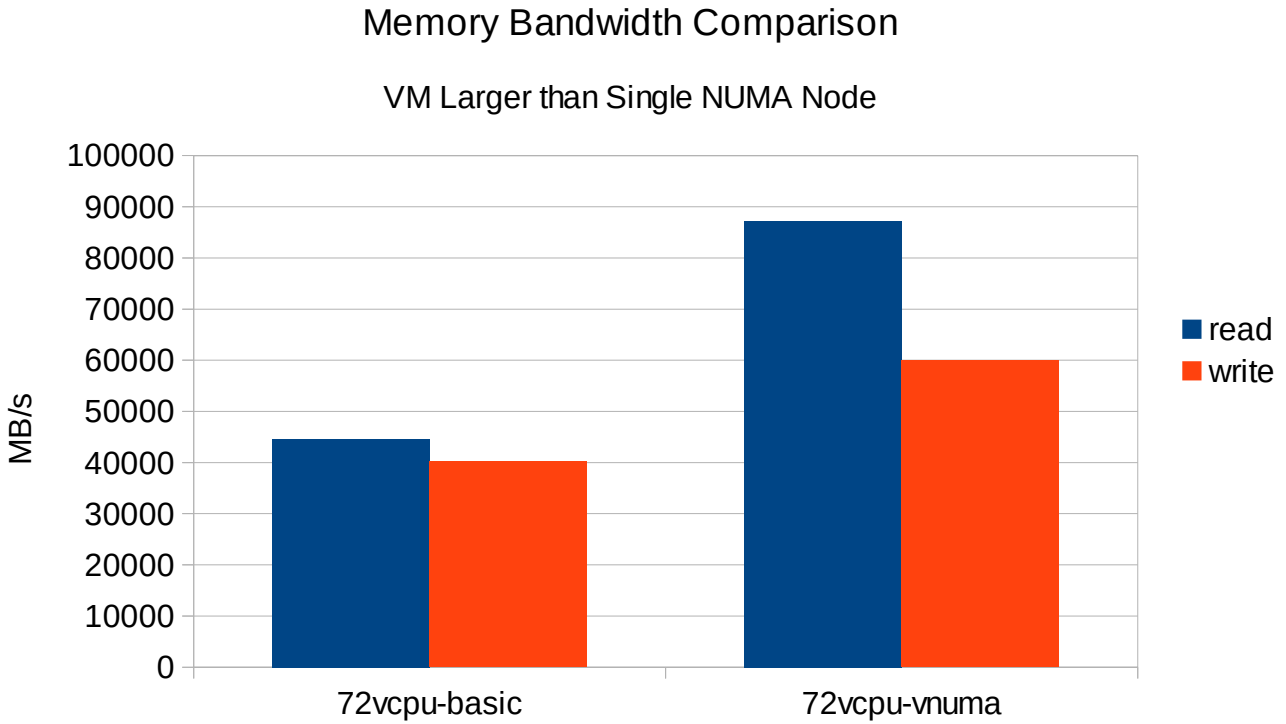
- **Consider vNUMA for large virtual machines**

```
<cpu>
  <numa>
    <cell id=’0’ cpus=’0-15’ memory=’16777216’ unit='KiB'/>
    <cell id=’1’ cpus=’16-31’ memory=’16777216’ unit='KiB'/>
  </numa>
</cpu>
```

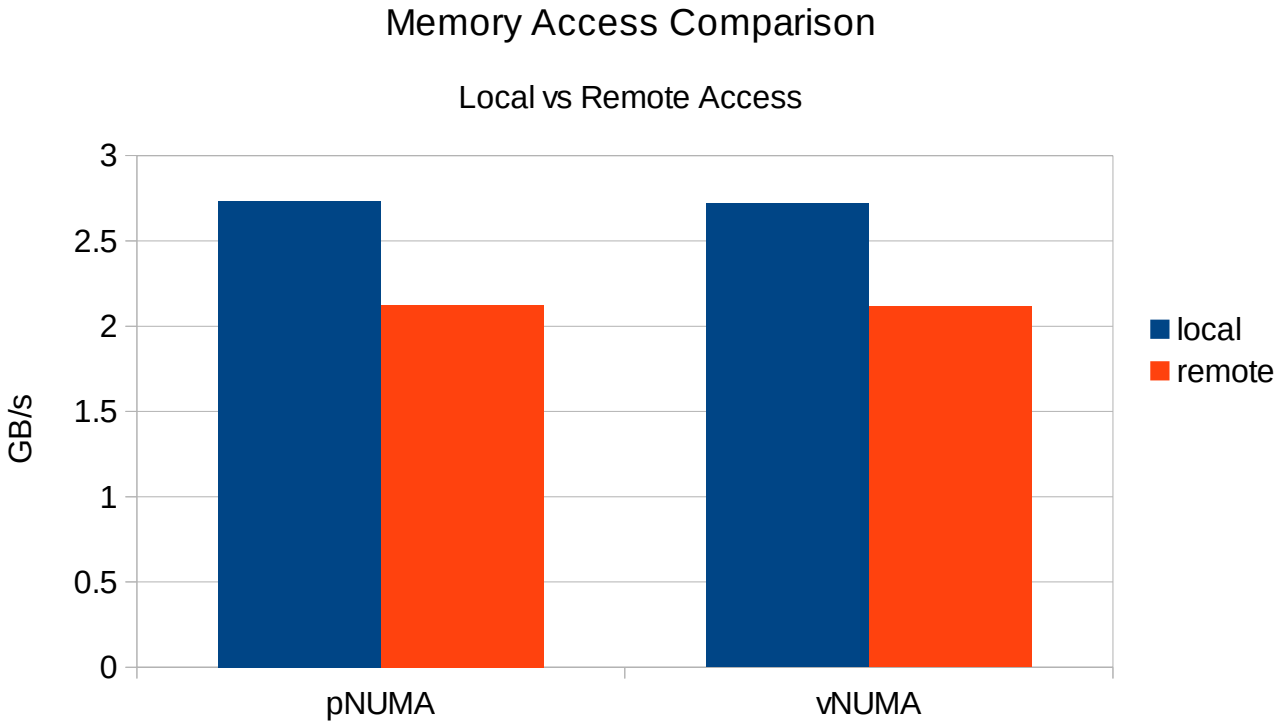
NUMA


Memory Bandwidth Comparison
VM Fits on Single NUMA Node
MB/s
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
16vcpu-basic
16vcpu-pinned
16vcpu-vnuma
read
write

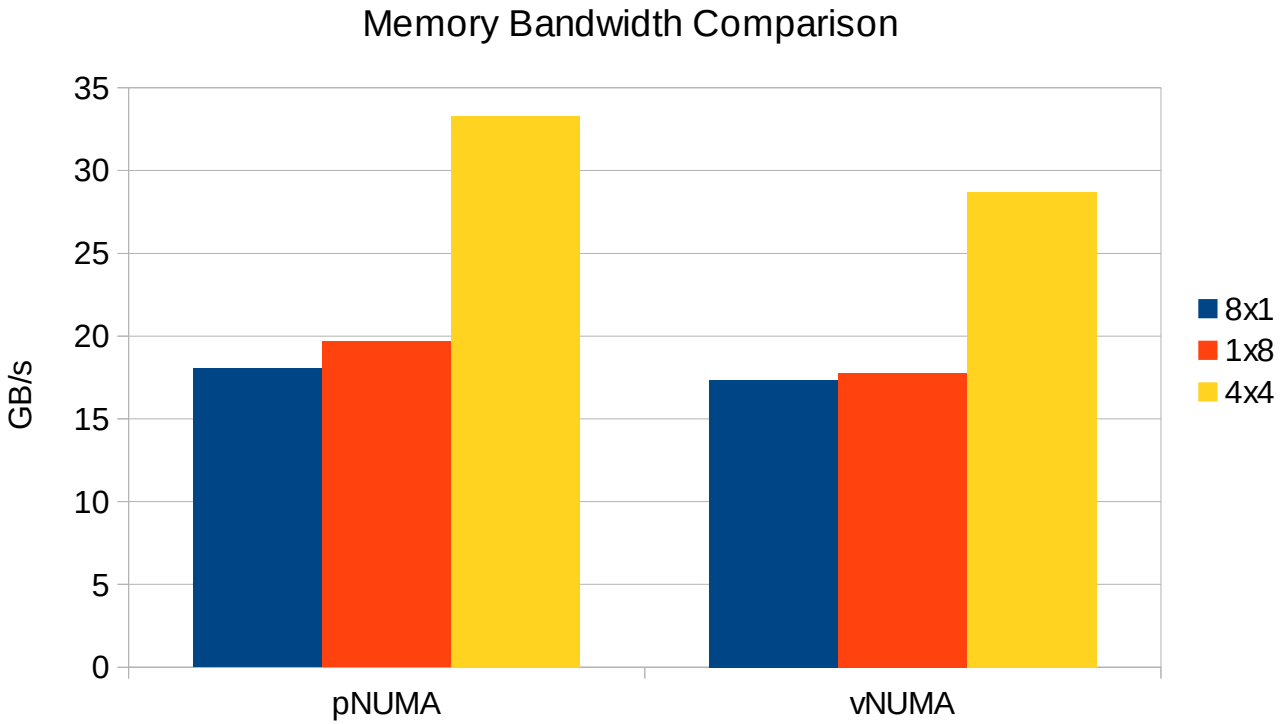
NUMA


Memory Bandwidth Comparison
VM Larger than Single NUMA Node
MB/s
100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
72vcpu-basic
72vcpu-vnuma
read
write

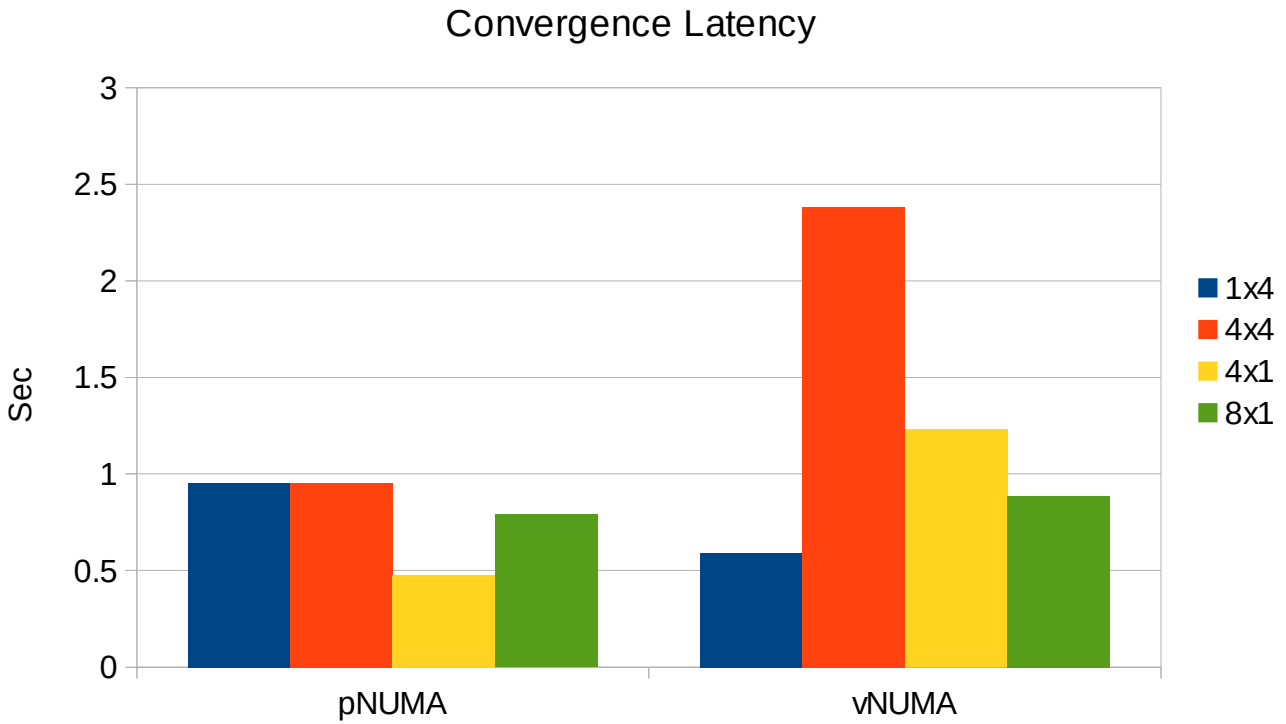
NUMA


Memory Access Comparison
Local vs Remote Access
GB/s
3
2.5
2
1.5
1
0.5
0
pNUMA
vNUMA
local
remote

NUMA


Memory Bandwidth Comparison
GB/s
35
30
25
20
15
10
5
0
pNUMA
vNUMA
8x1
1x8
4x4

NUMA


Convergence Latency
Sec
3
2.5
2
1.5
1
0.5
0
pNUMA
vNUMA
1x4
4x4
4x1
8x1


SUSE®
We adapt. You succeed.