Megasas and VFIO

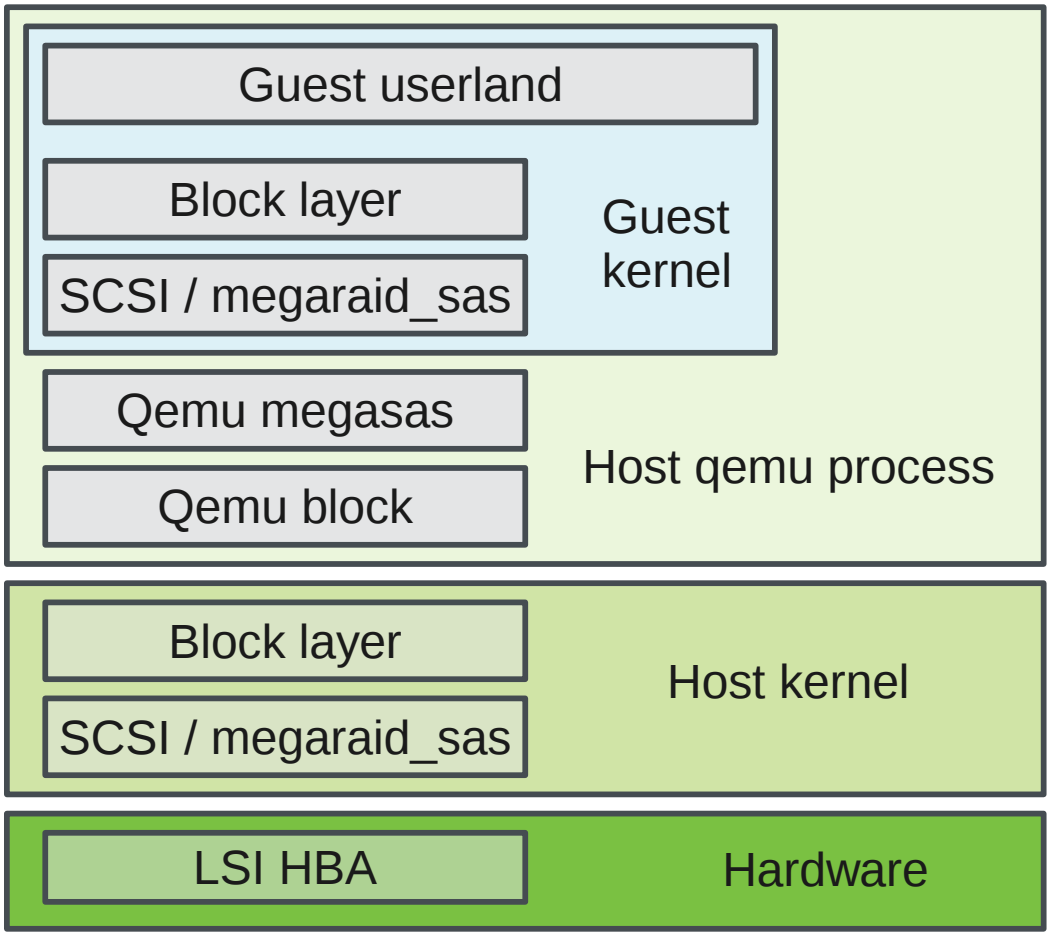
PCI device-assignment with qemu

Dr. Hannes Reinecke
SUSE Labs
hare@suse.de

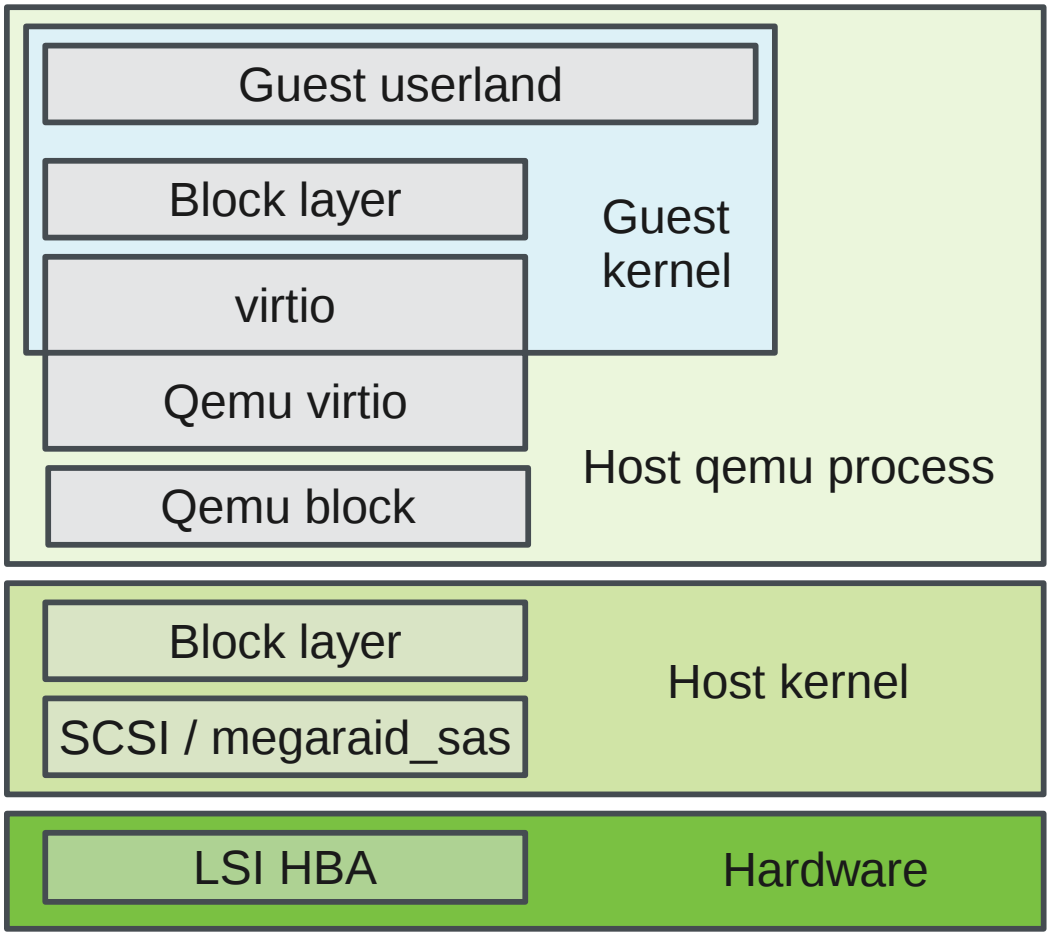
PCI device-assignment

- Various I/O methods:
  - Emulated devices (qemu)
  - Virtual devices (virtio)
  - Accelerated virtio (vhost)
  - Direct access to hardware

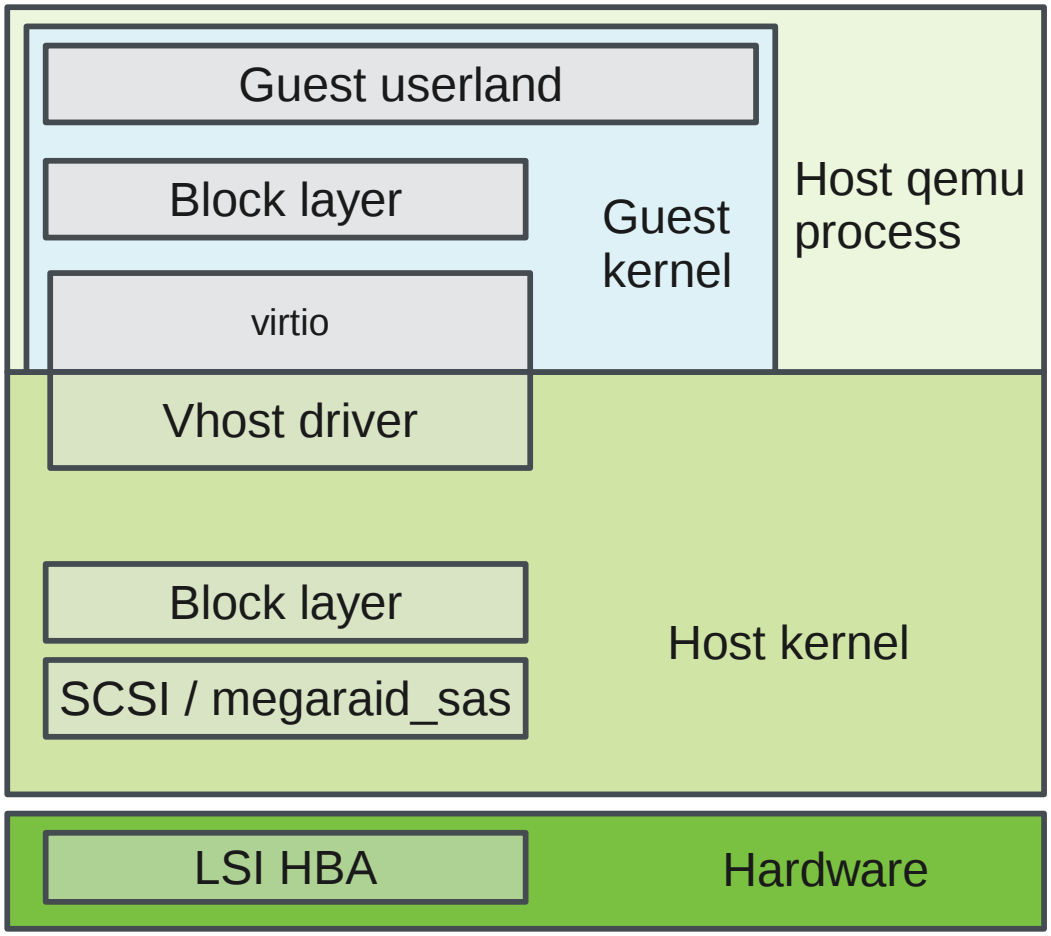
Emulated devices



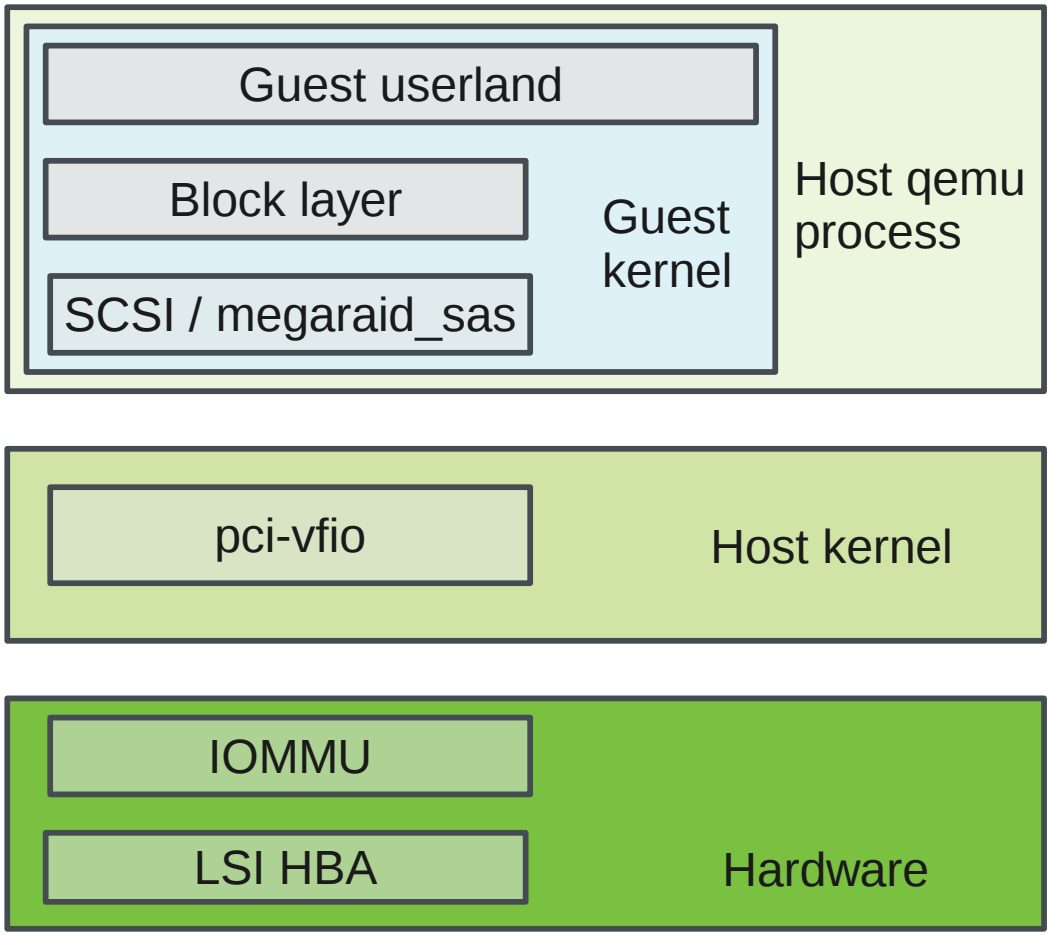
Virtio: Efficient device interface



Vhost: in-kernel I/O pass-through



SR-IOV: PCI device assignment



Logical partitioning with Qemu

PCI device assignment

- Direct access to hardware:
  - Individual PCI devices are assigned to a guest
  - Guest can use unmodified drivers
- Prevent host access to assigned devices
  - pci-stub

PCI device assigment

- Guest and host have a different memory mapping
- DMA addresses need to be translated
- Hardware support needed
- Interrupts might need to be remapped
- VFIO

LPAR guest

- Create a guest with just VFIO devices
- No emulation
- Simple commandline:

```
# qemu-system-x86_64 -enable-kvm -net none \
-device vfio-pci,host=01:10.0,id=igbvf0 \
-device vfio-pci,host=07:00.0,id=megasas0 -m 1024
```

LPAR guest


QEMU
RAID Controller BIOS Version 3.06.00 (Build March 25, 2009)
HA -0 (Bus 0 Dev 4) Intel (R) RAID Controller RS2BL080
Battery Status: Not present
PCI SLOT ID LUN VENDOR PRODUCT REVISION CAPACITY
3 INTEL Intel (R) RAID Controller 512MB
3 10 0 SEAGATE ST9146802SS 0003 140014MB
3 11 0 SEAGATE ST9146802SS 0003 140014MB
3 0 INTEL Virtual Drive RAID0 278472MB
1 Virtual Drive(s) found on the host adapter.
1 Virtual Drive(s) handled by BIOS
Press <Ctrl><Y> for Preboot CLI
<Ctrl><G> to enter the RAID BIOS Console <<<<<<<
2009 LSI Corporation. All rights reserved !

Performance measurement

Measurement goals

- Compare different emulation methods
- Measure emulation overhead
- Identify possible bottlenecks
- Identify areas of improvement

Testcases

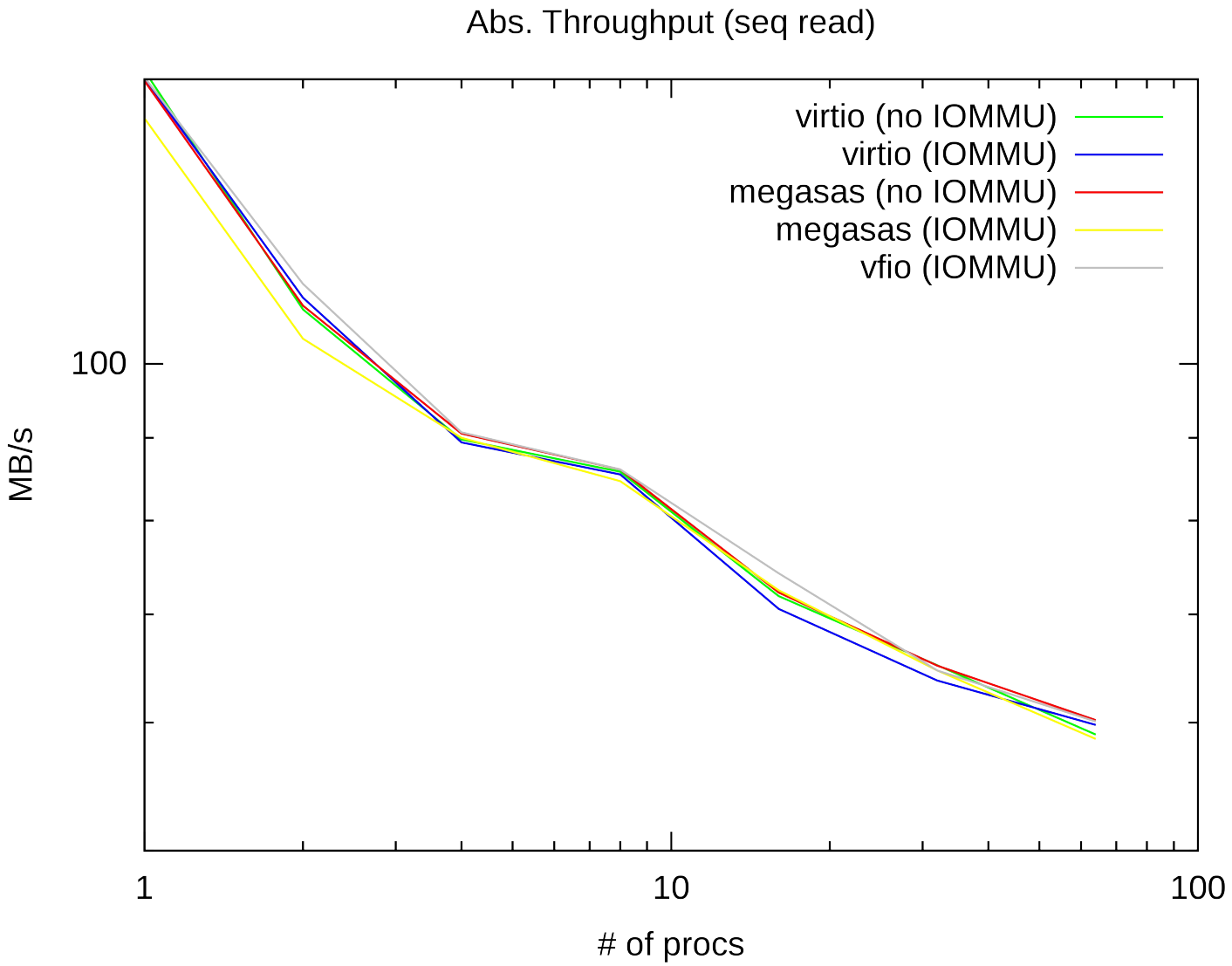
- Test 5 different cases:
  - Megasas (IOMMU enabled)
  - Megasas (IOMMU disabled)
  - Virtio-scsi (IOMMU enabled)
  - Virtio-scsi (IOMMU disabled)
  - VFIO
- Using same hardware for each test
- No modifications to host or guest installation

Test platform

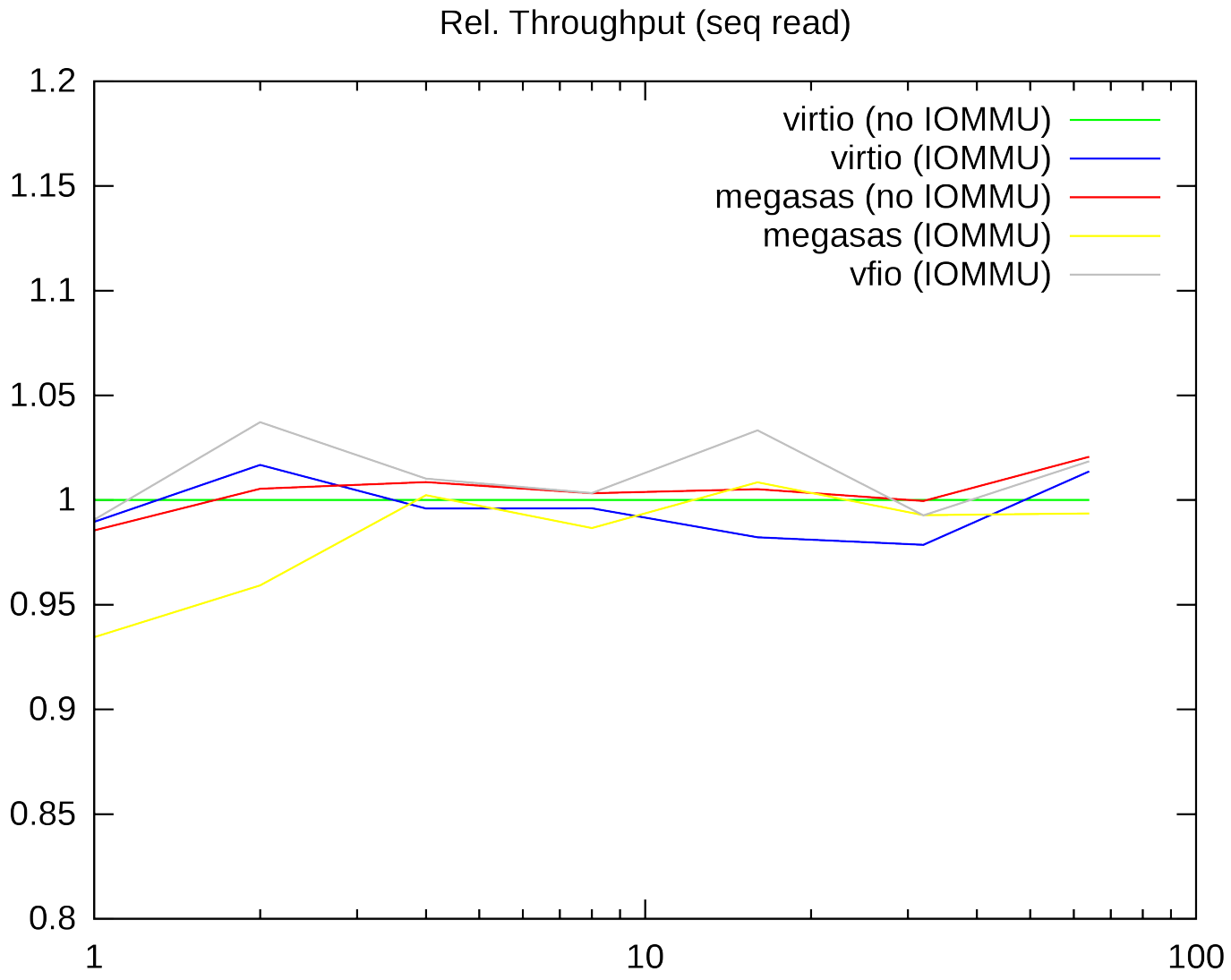
- Intel development platform
- 4-socket 10-core Xeon
- Integrated LSI megaraid HBA
- 128 GB RAM
- Using mmtest / tiobench to generate test results

I/O Throughput results

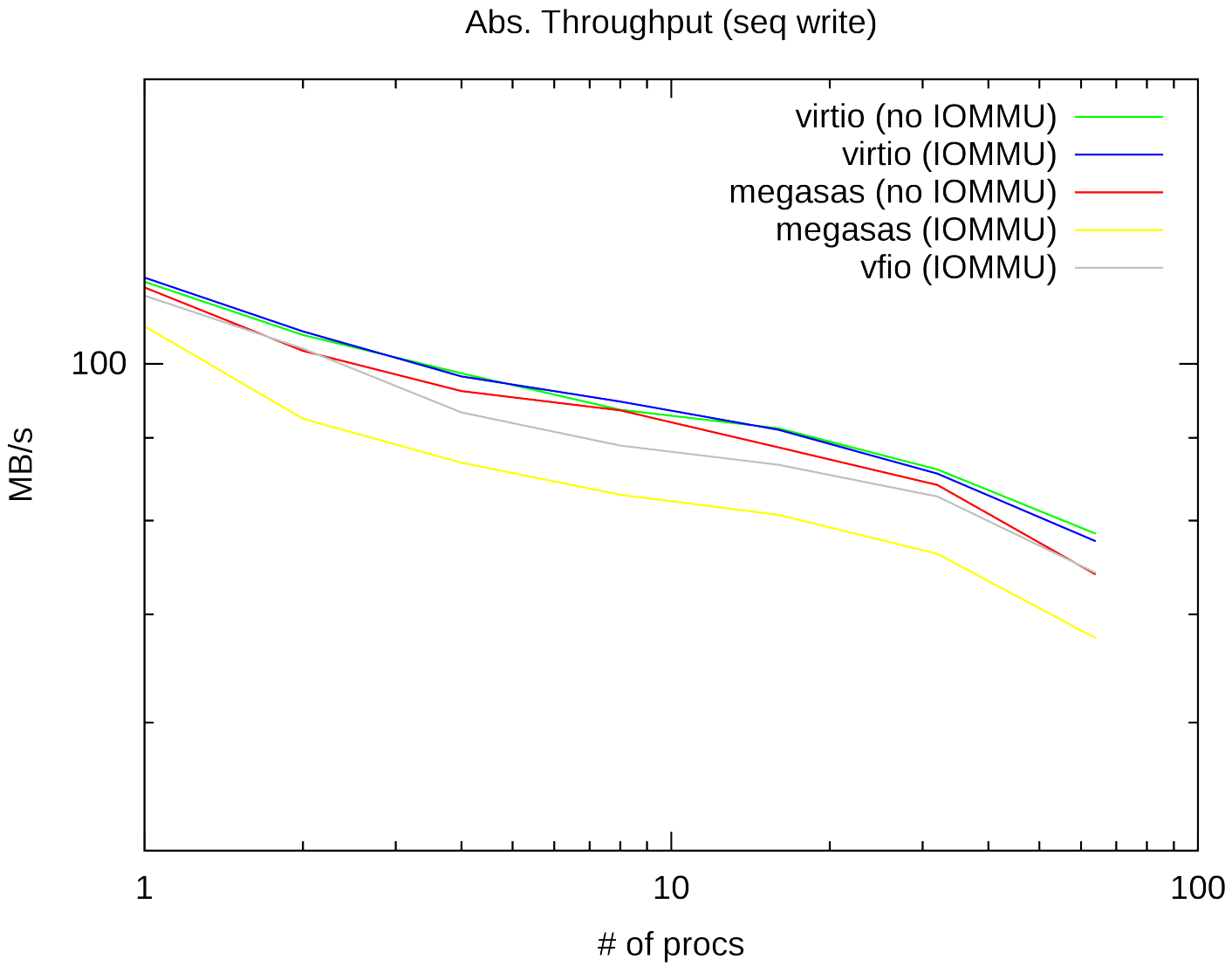
I/O throughput (seq read)



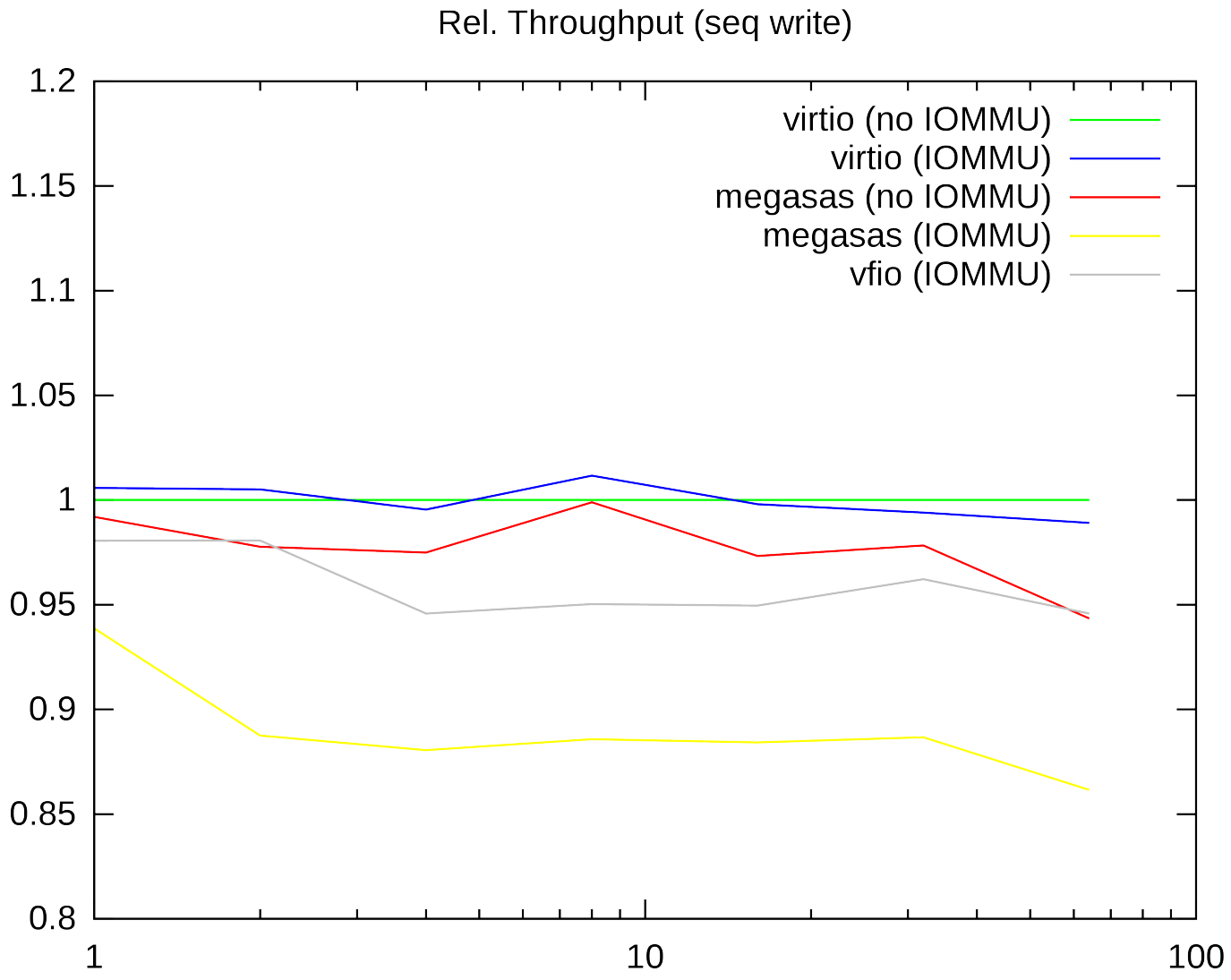
I/O throughput (seq read)



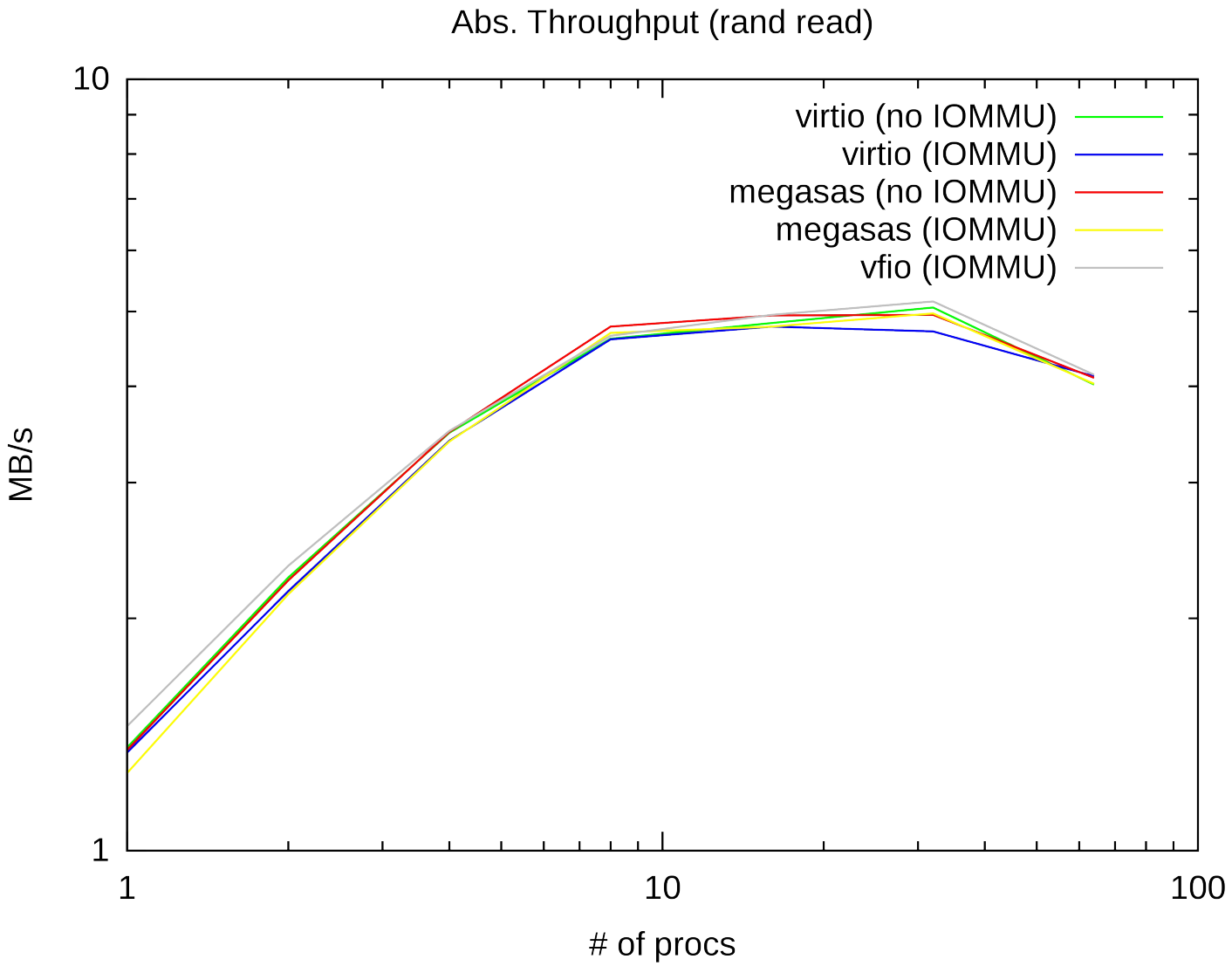
I/O throughput (seq write)



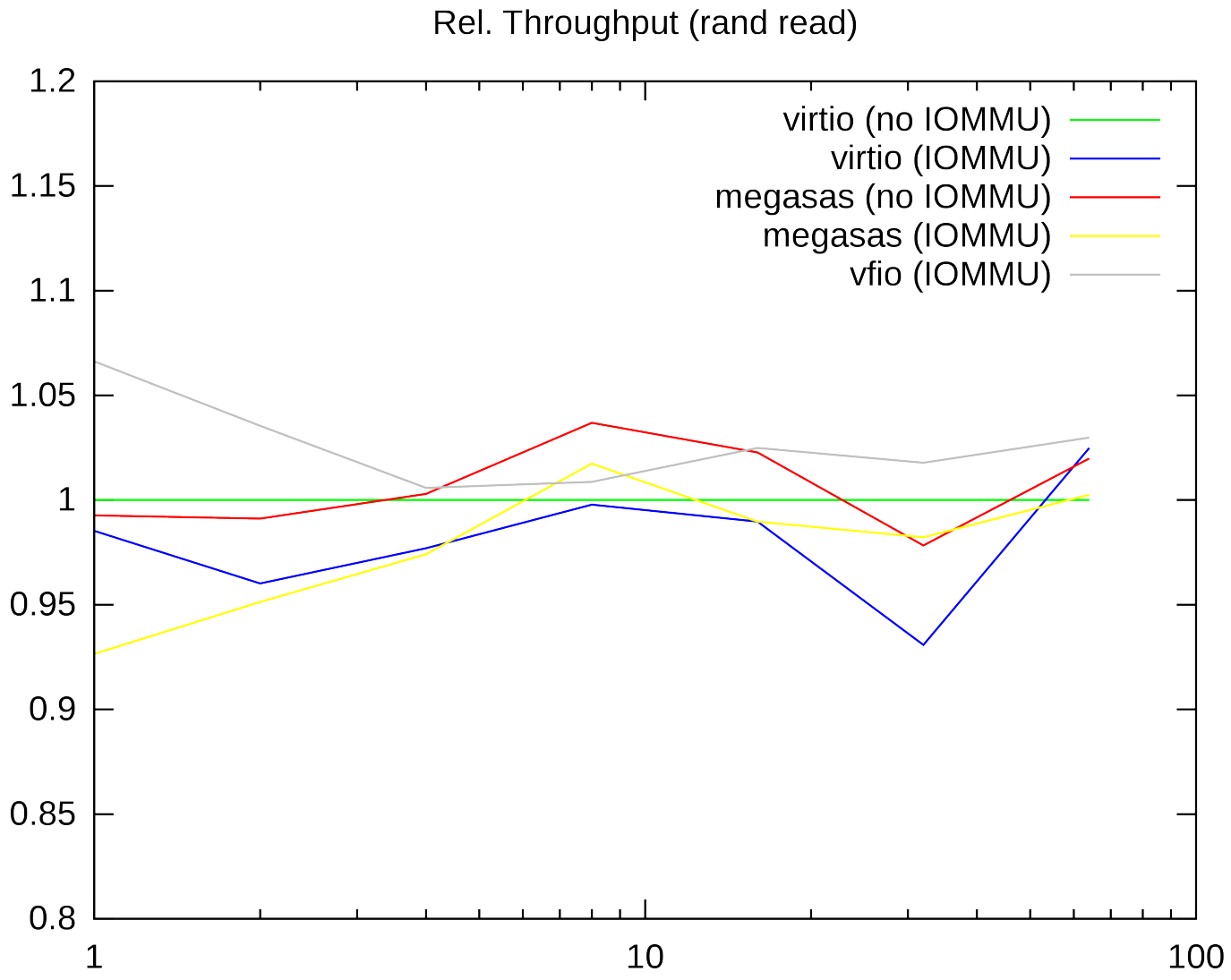
I/O throughput (seq write)



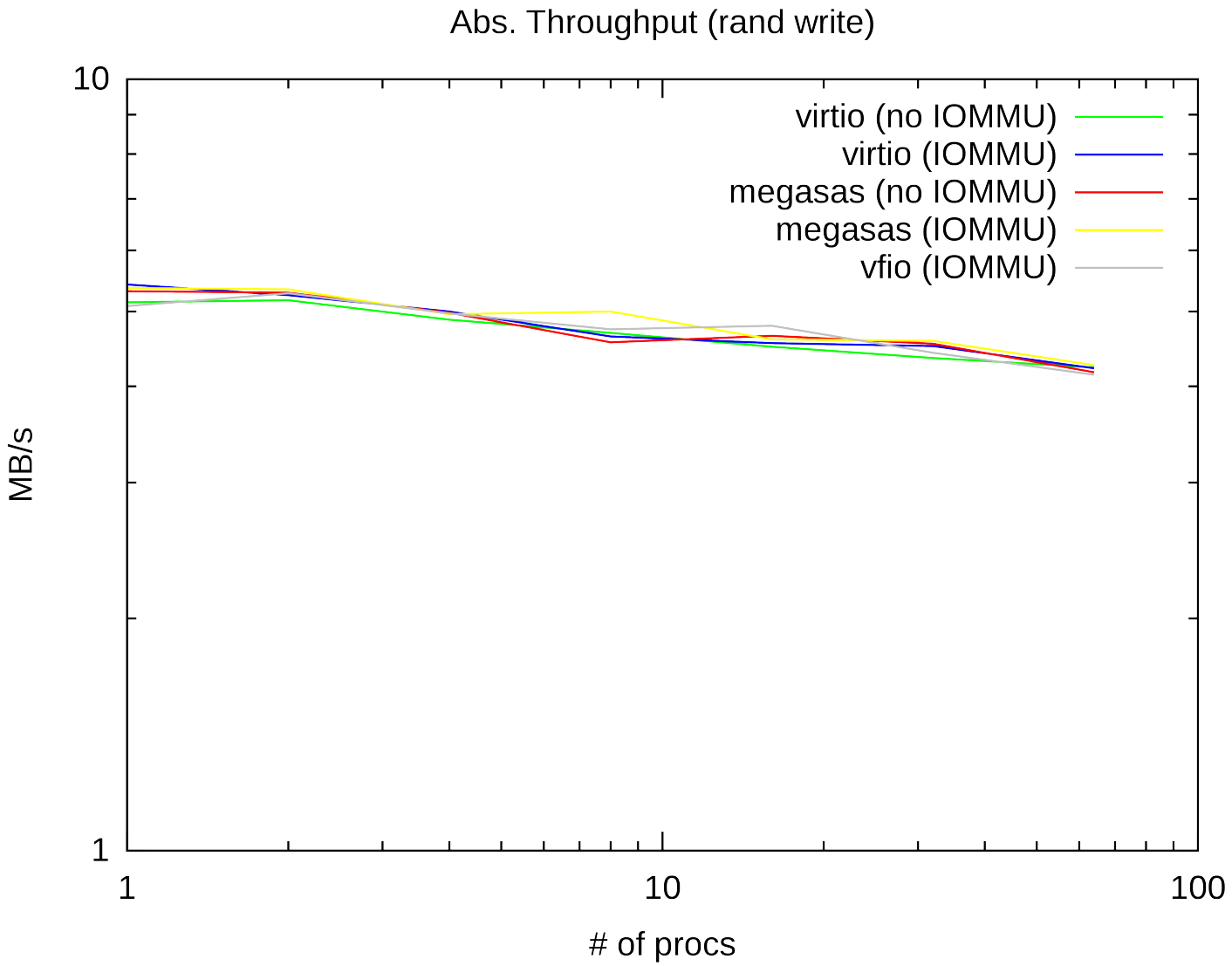
I/O throughput (rand read)



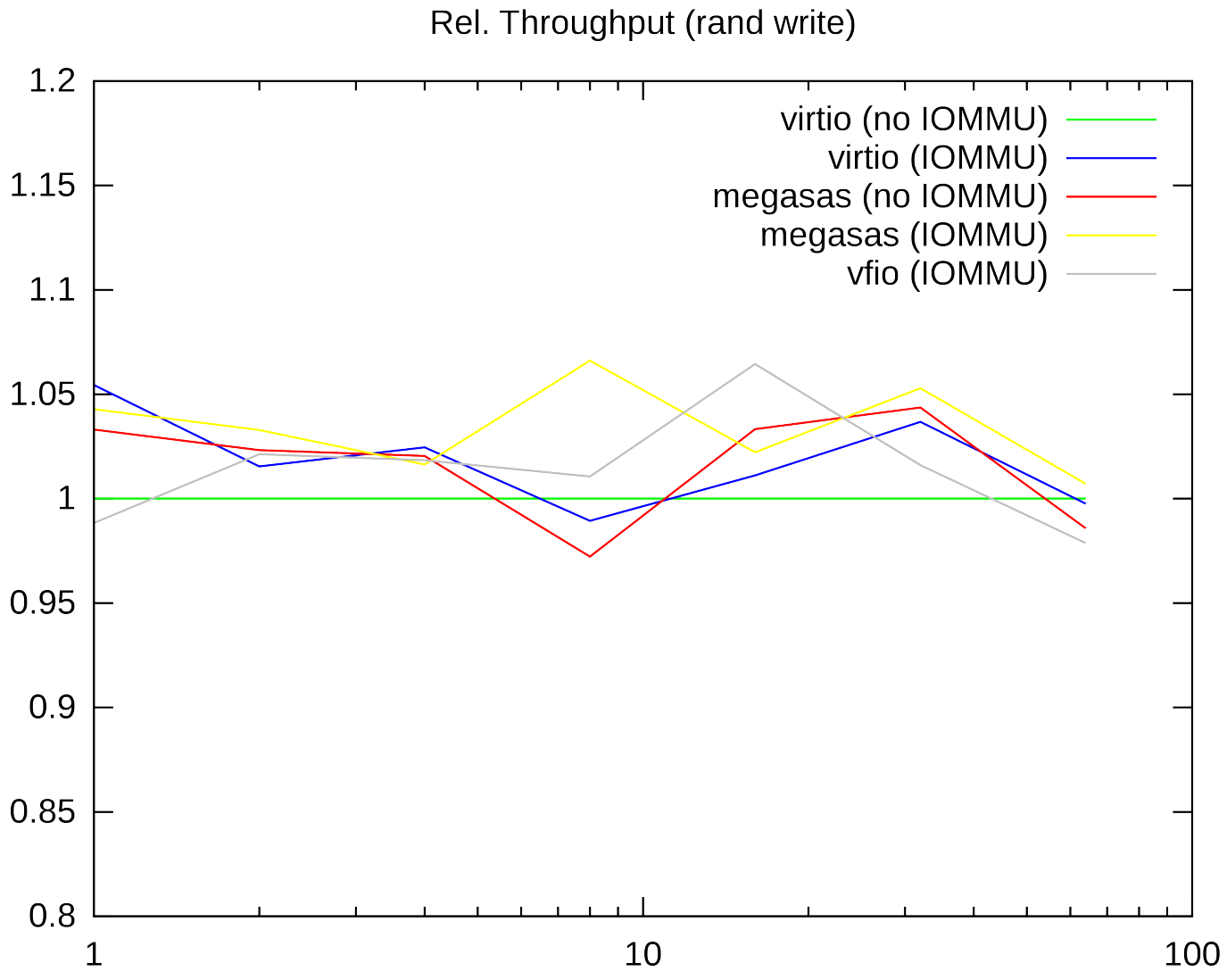
I/O throughput (rand read)



I/O throughput (rand write)

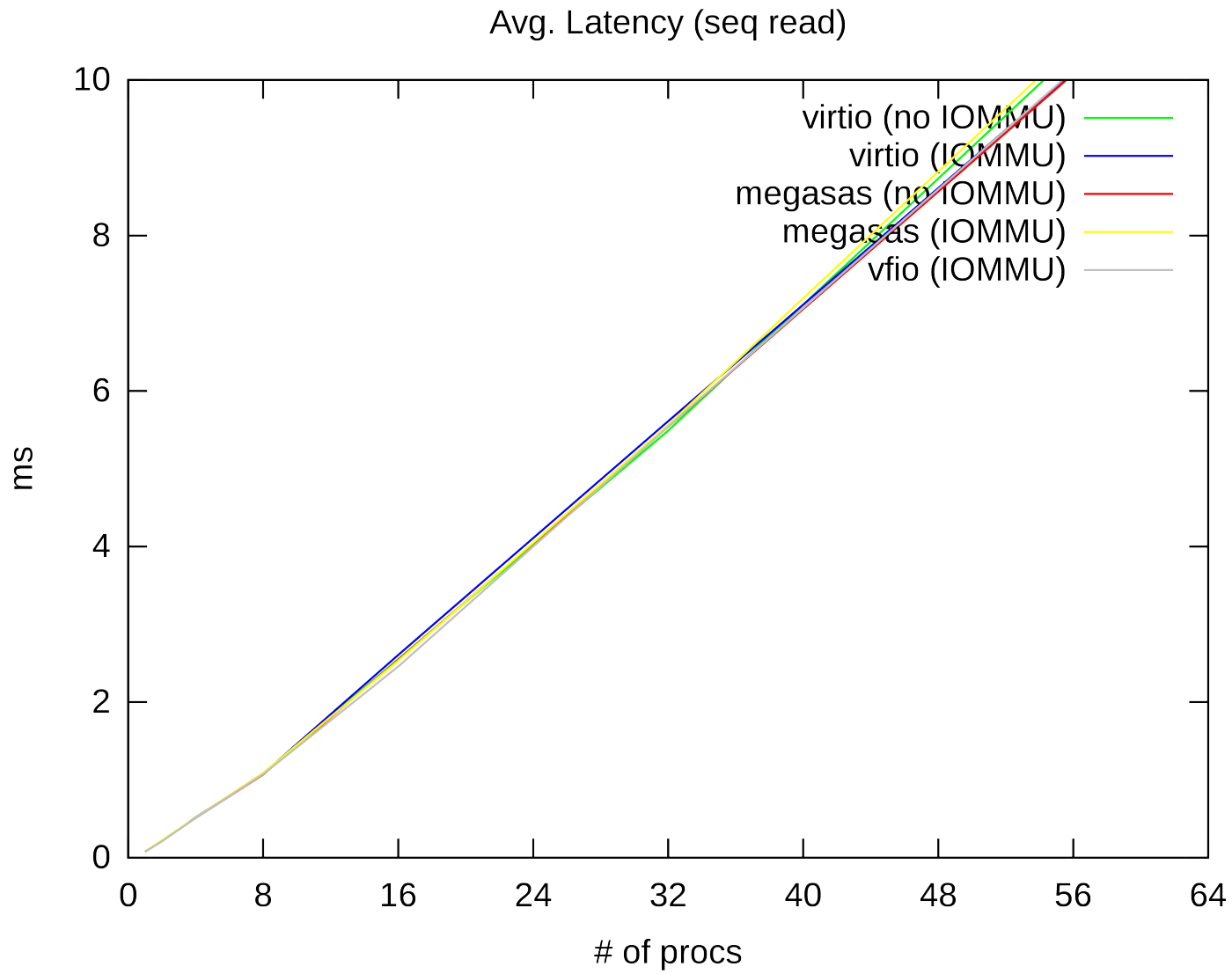


I/O throughput (rand write)

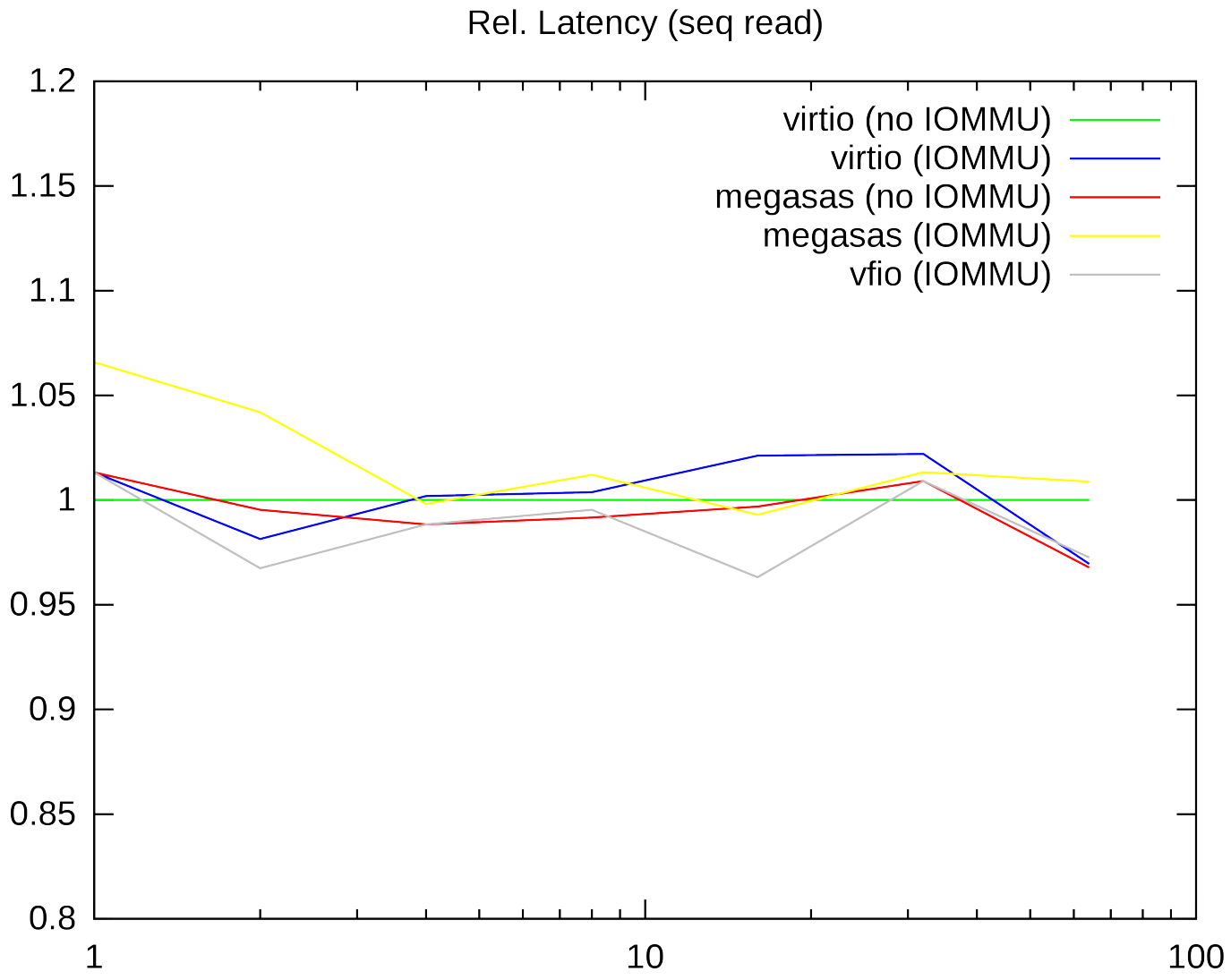


I/O Latency results

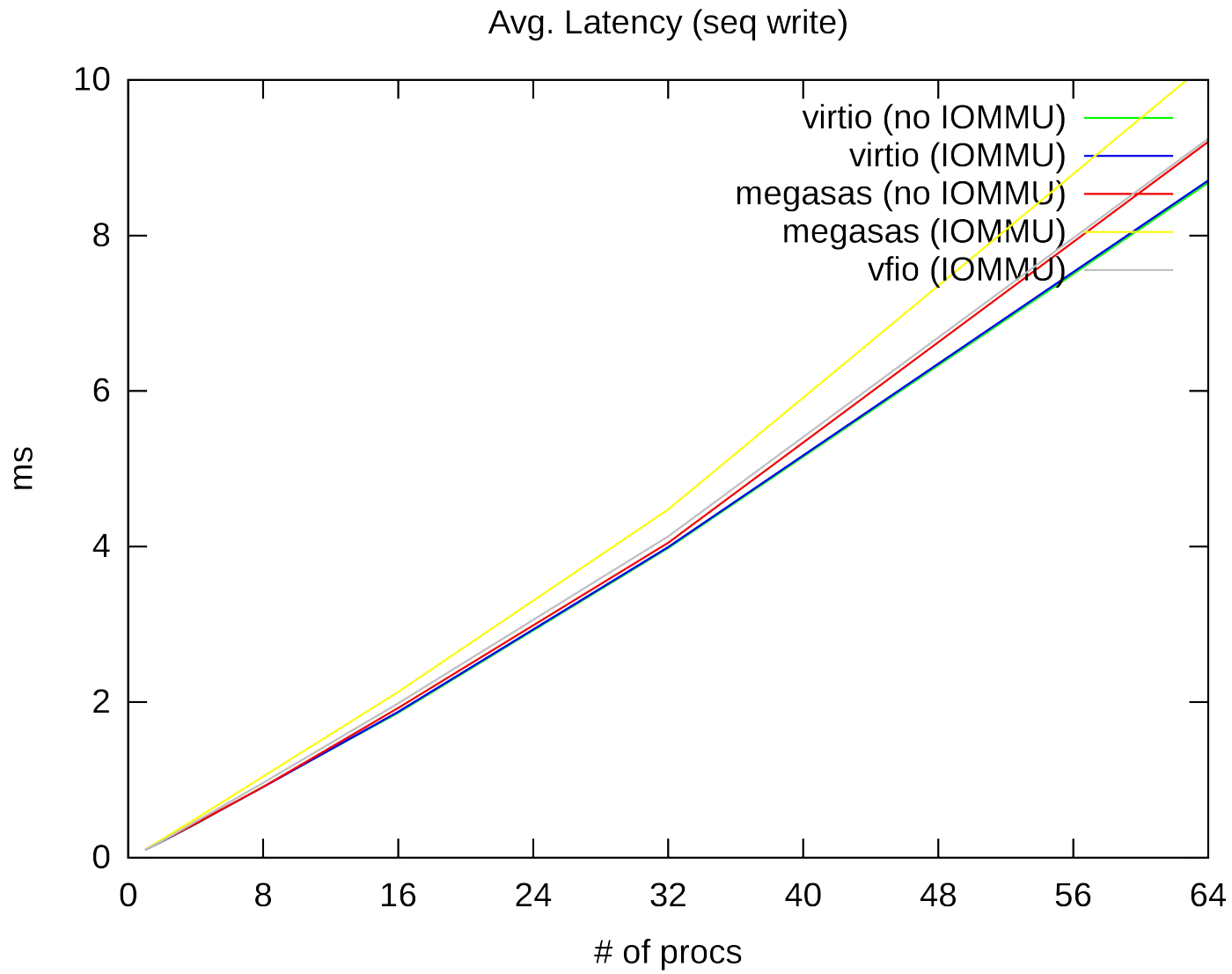
I/O latency (seq read)



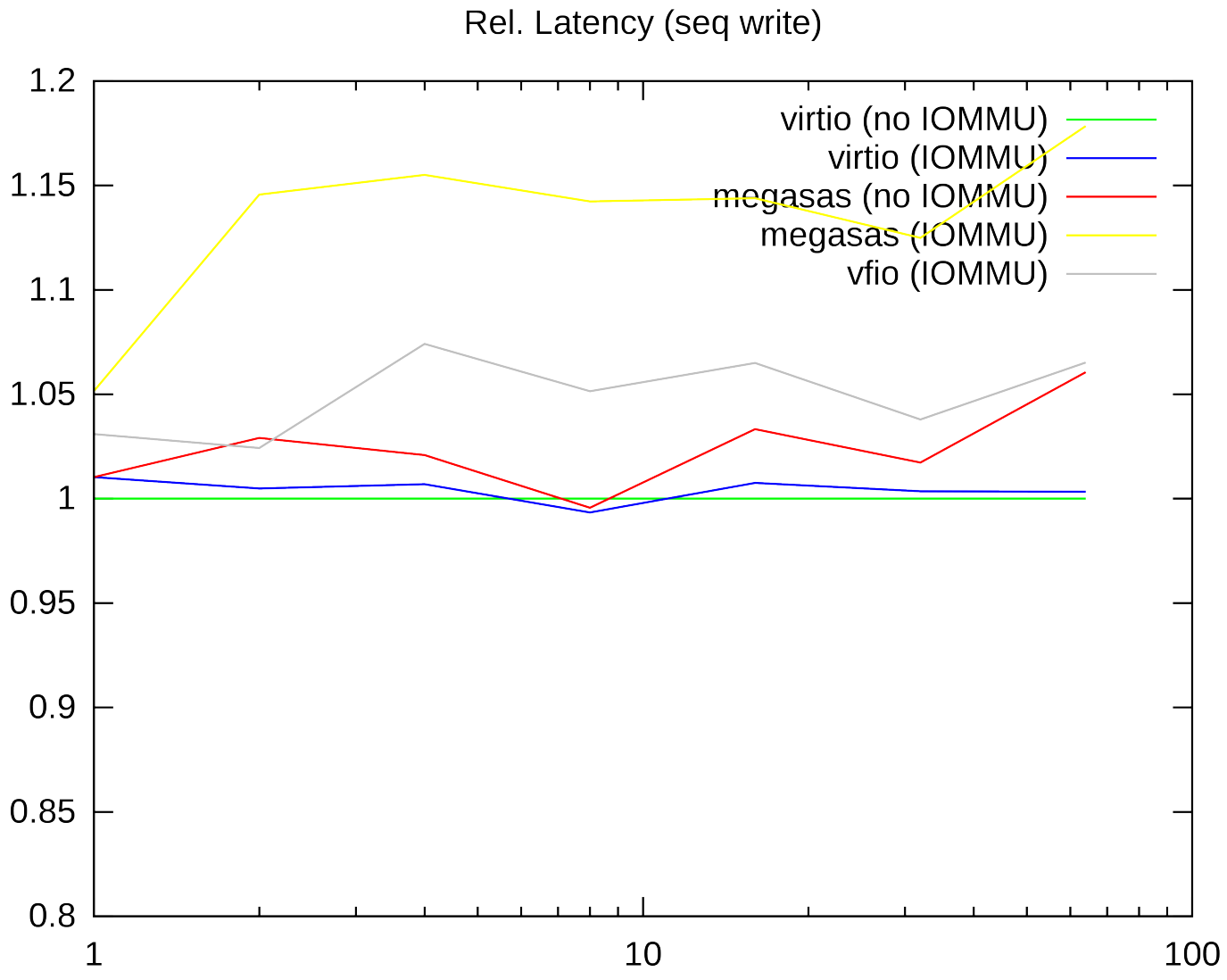
I/O latency (seq read)



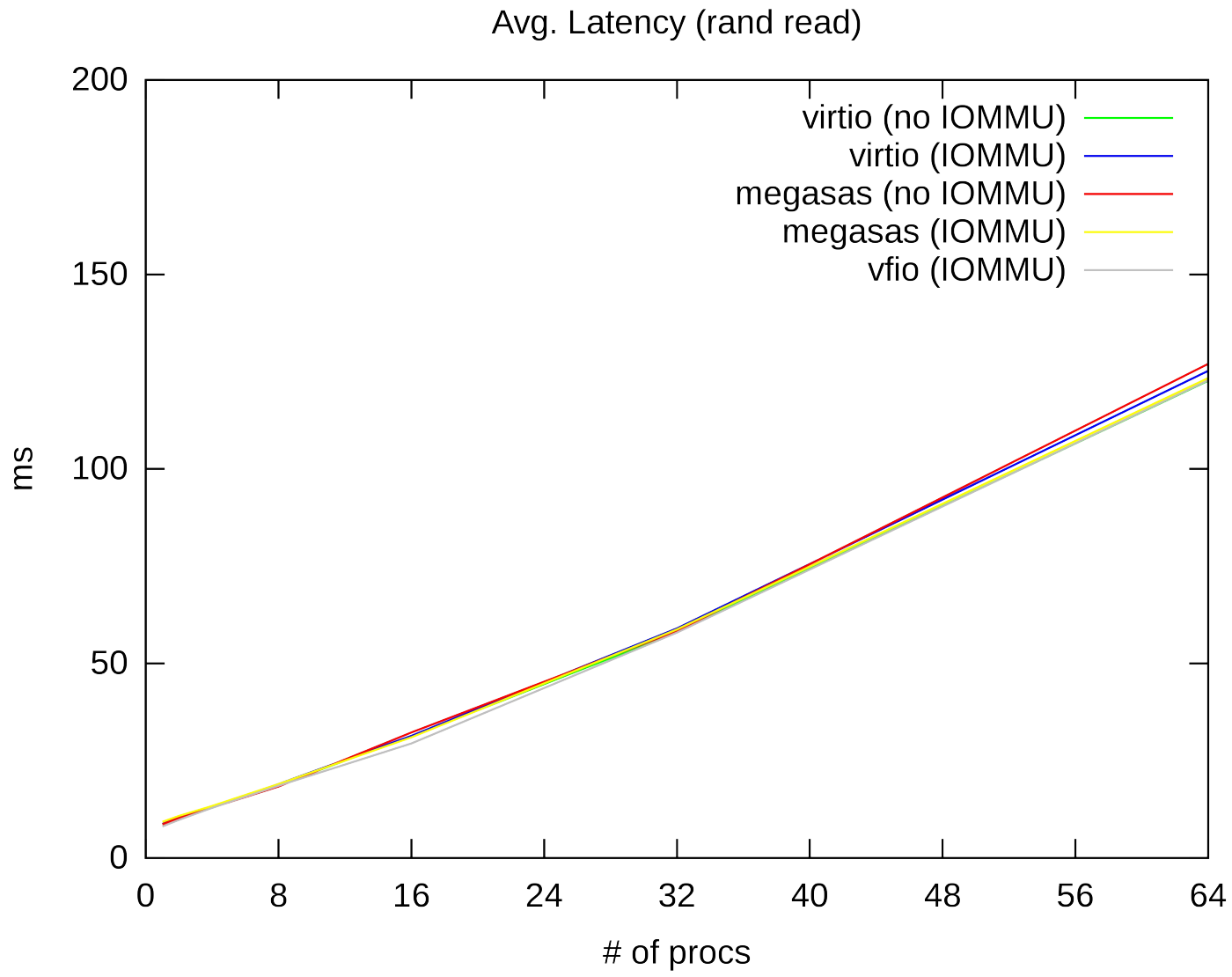
I/O latency (seq write)



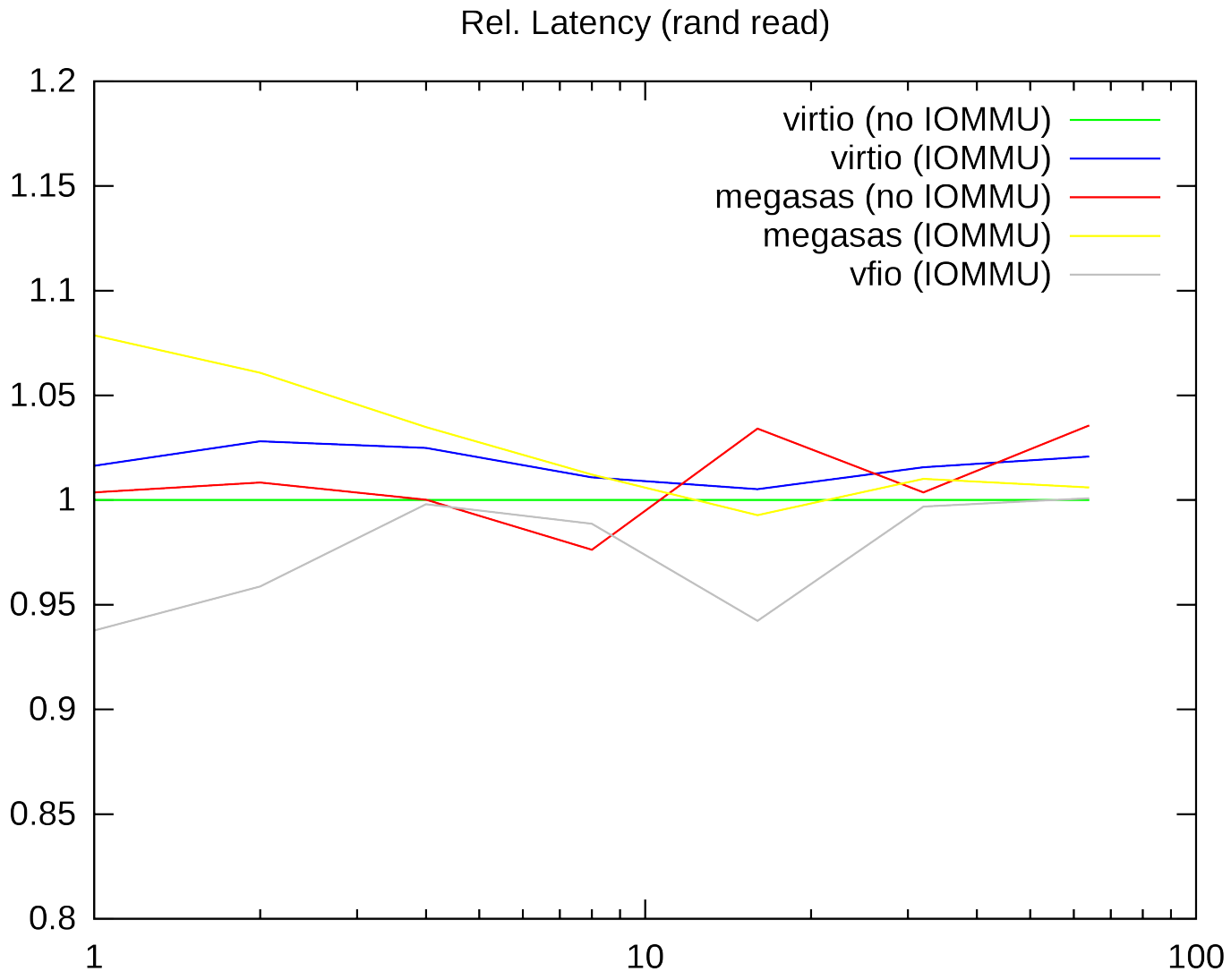
I/O latency (seq write)



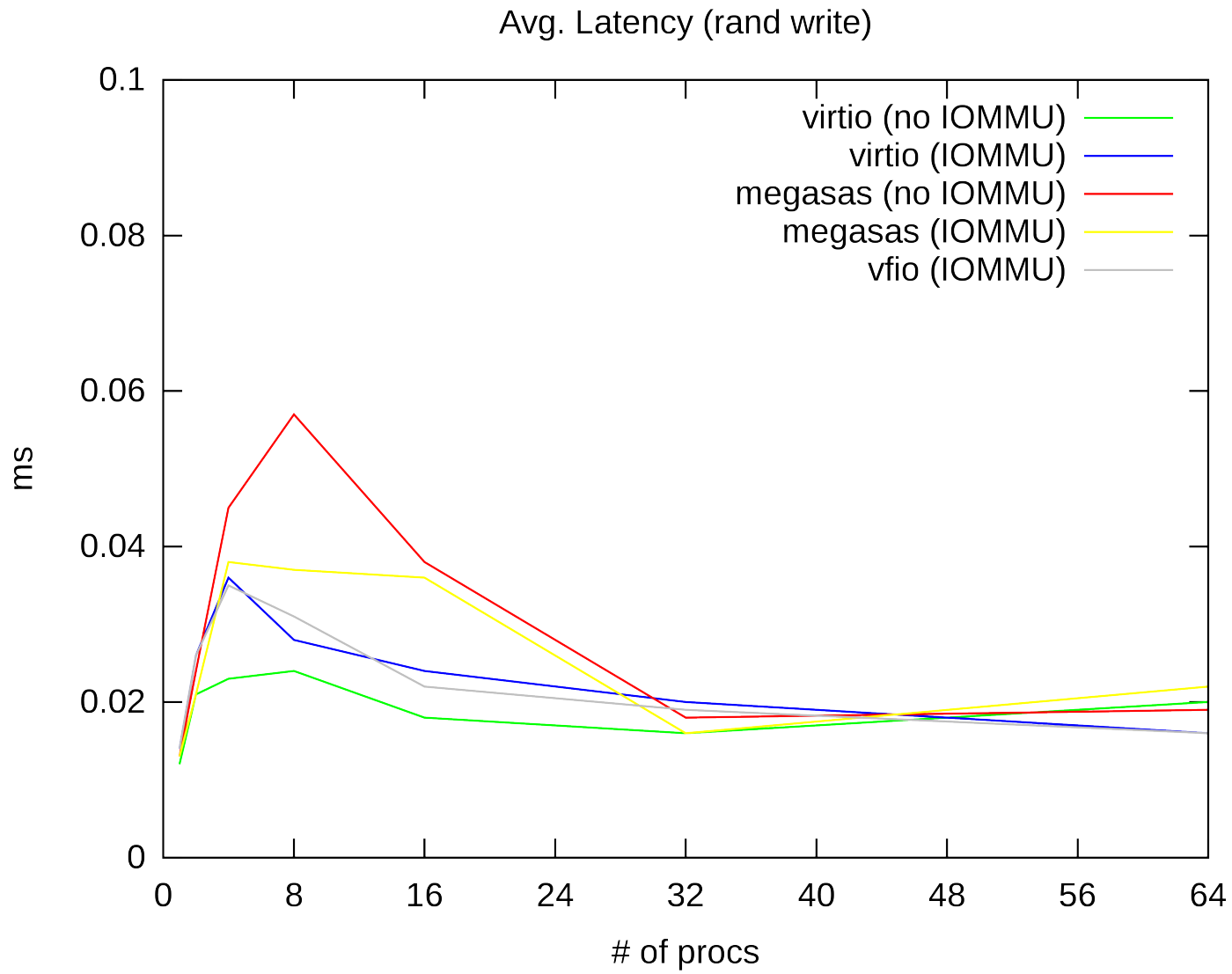
I/O latency (rand read)



I/O latency (rand read)



I/O latency (rand write)



Results

- All methods yield nearly identical results (+/- 5%)
- VFIO is not the fastest I/O path
- Random I/O significantly slower than sequential
- 'sweet spot' at 8 concurrent processes; most likely hardware related (8-core processors)
- Random write latency significantly lower than random read; caching?


suse®

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. SUSE may make improvements in or changes to the software described in this document at any time.