# Keep a limit on it
## IO Throttling in QEMU

**Ryan Harper – ryanh@us.ibm.com**
**Open Virtualization**
**IBM Linux Technology Center**

***August 15, 2011***

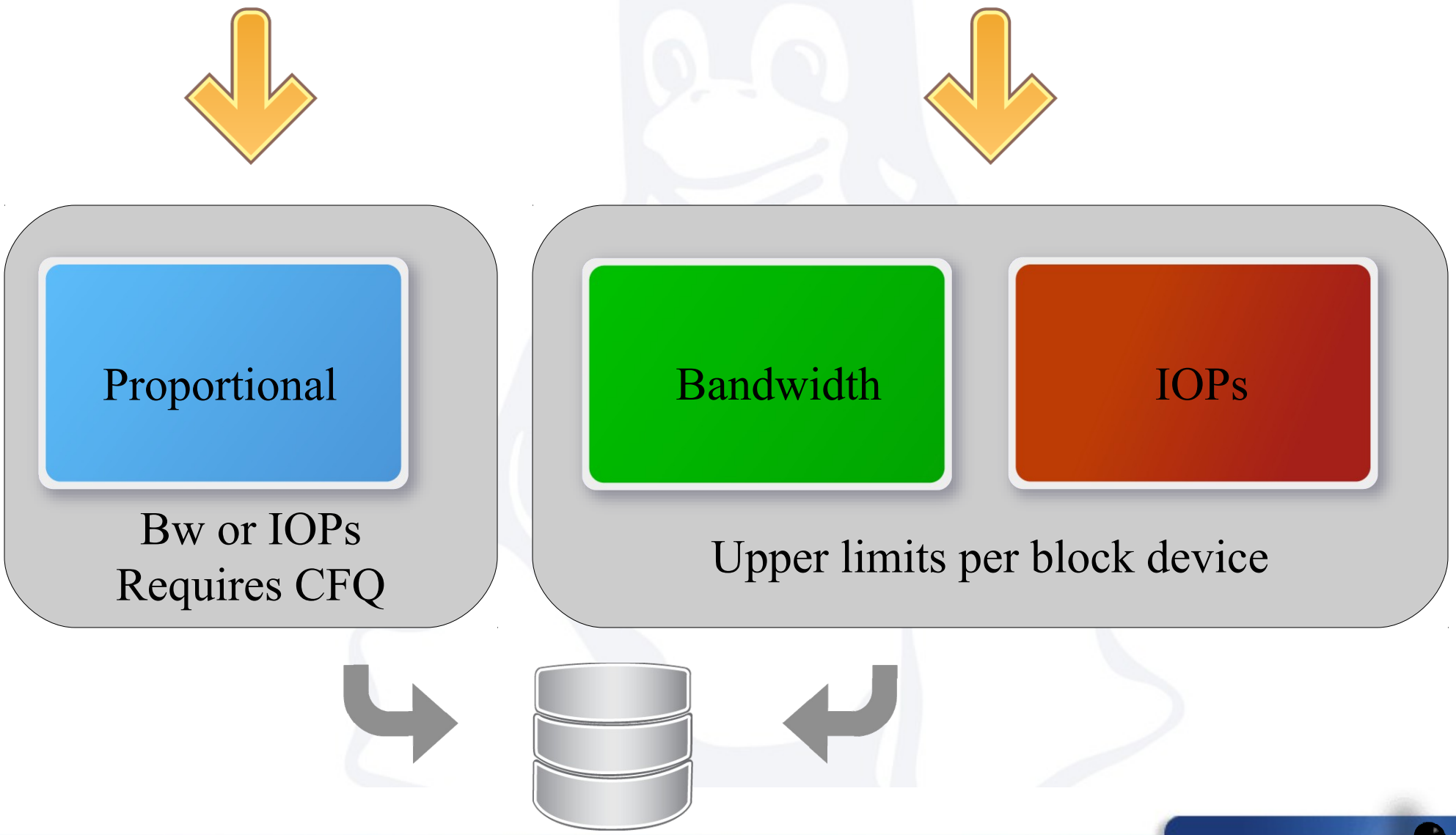


# Contributors

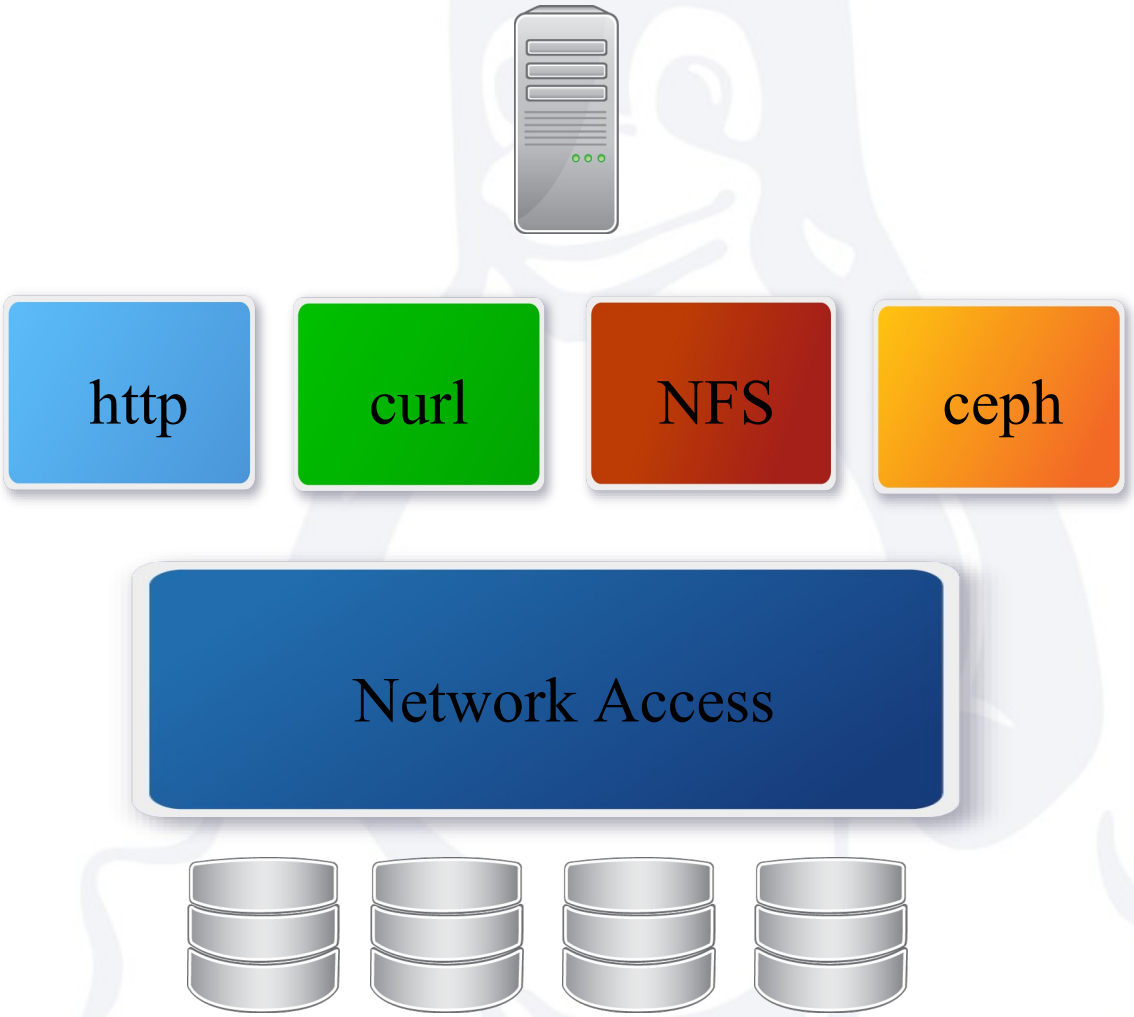
Zhi Yong Wu – wuzhy@linux.vnet.ibm.com
Stefan Hajnoczi – stefanha@linux.vnet.ibm.com
Karl Rister – krister@us.ibm.com
Khoa Huynh, Ph.D. – khoa@us.ibm.com
Steve Pratt – spratt@us.ibm.com

# Limited Resources

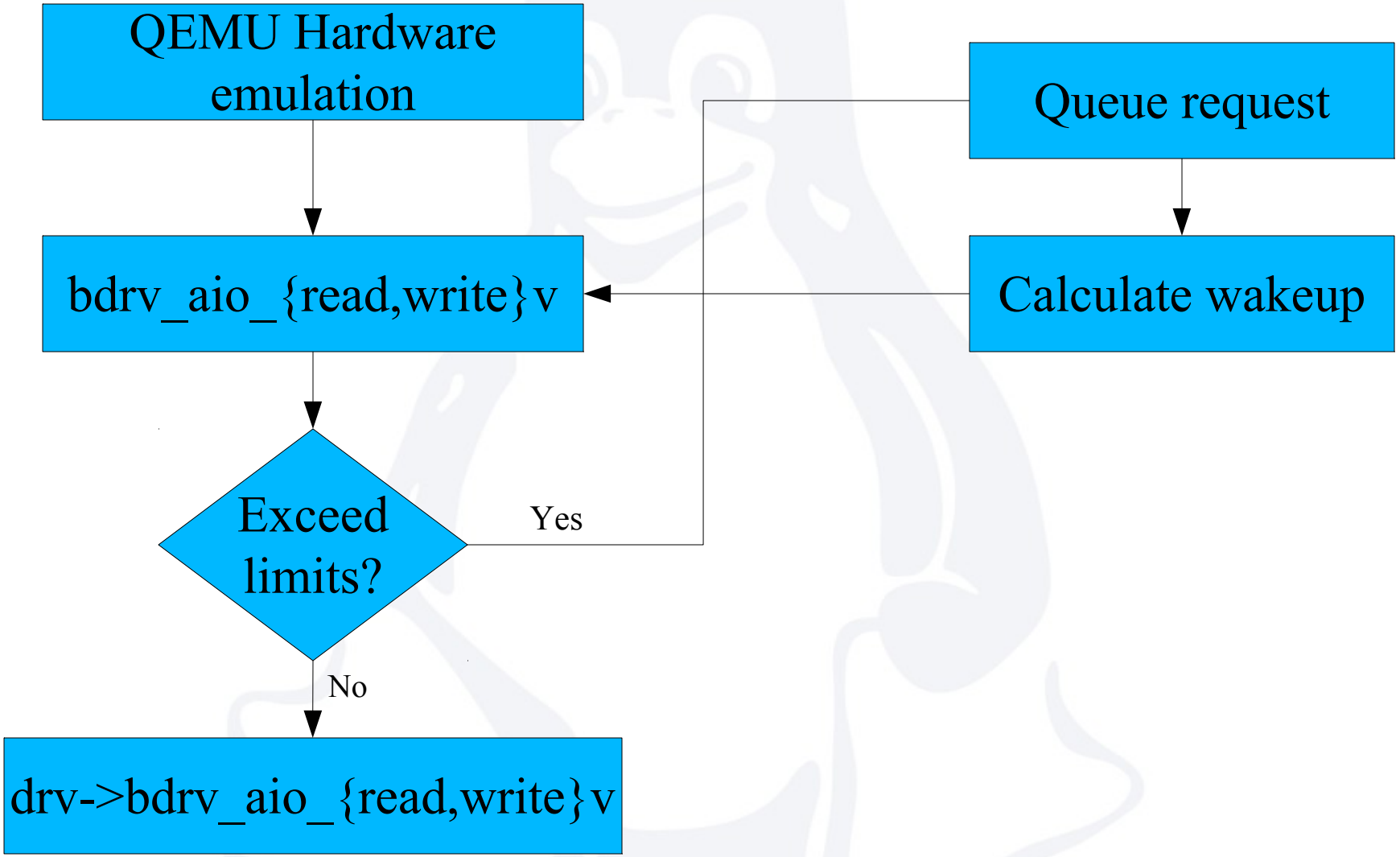
# cgroup blkio controller



# Non host block access



# QEMU Block Layer Limits



# Block IO Throttle Comparison

- Effectiveness
  - Can your configuration be throttled?
  - Is the cap ever exceeded?
  - What amount of IO does the guest observe?
- Cost
  - Is there a substantial cost to implement throttling?
    - If so, where is that cost incurred?
  -
  -

# Block IO Throttle Configuration

- Storage backends
  - LVM over SATA disk
  - EXT4 over SATA disk
  - NFS (IBM n3600)
- Image Formats
  - RAW
  - QCOW2
- Host Cache mode
  - ,cache=none
  - ,cache=writethrough
- Block Limiting
  - cgroup blkio throttling
  - QEMU blk-throttle

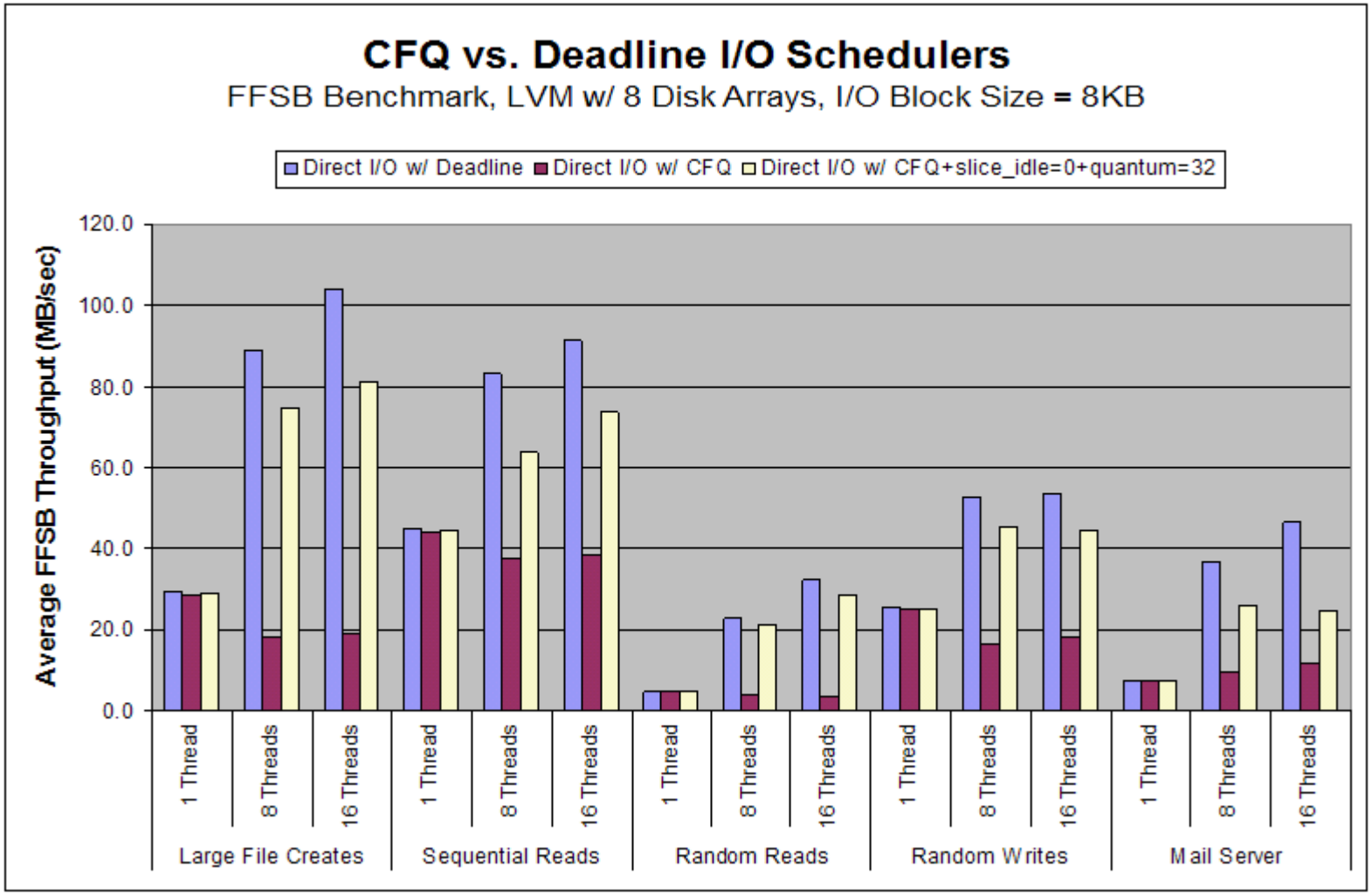
# Workloads

- 5 different workloads
  - streaming writes
    - mkfs.ext4
  - random reads and writes
    - fio iometer with randrw mix
  - random reads
    - fio aio-read
  - random writes
    - fio aio-write
  - streaming reads
    - fio disk-surface-scan
- 1 and 5VM instances, isolated and mixed
- VMs have 50G virtio-blk device
  -

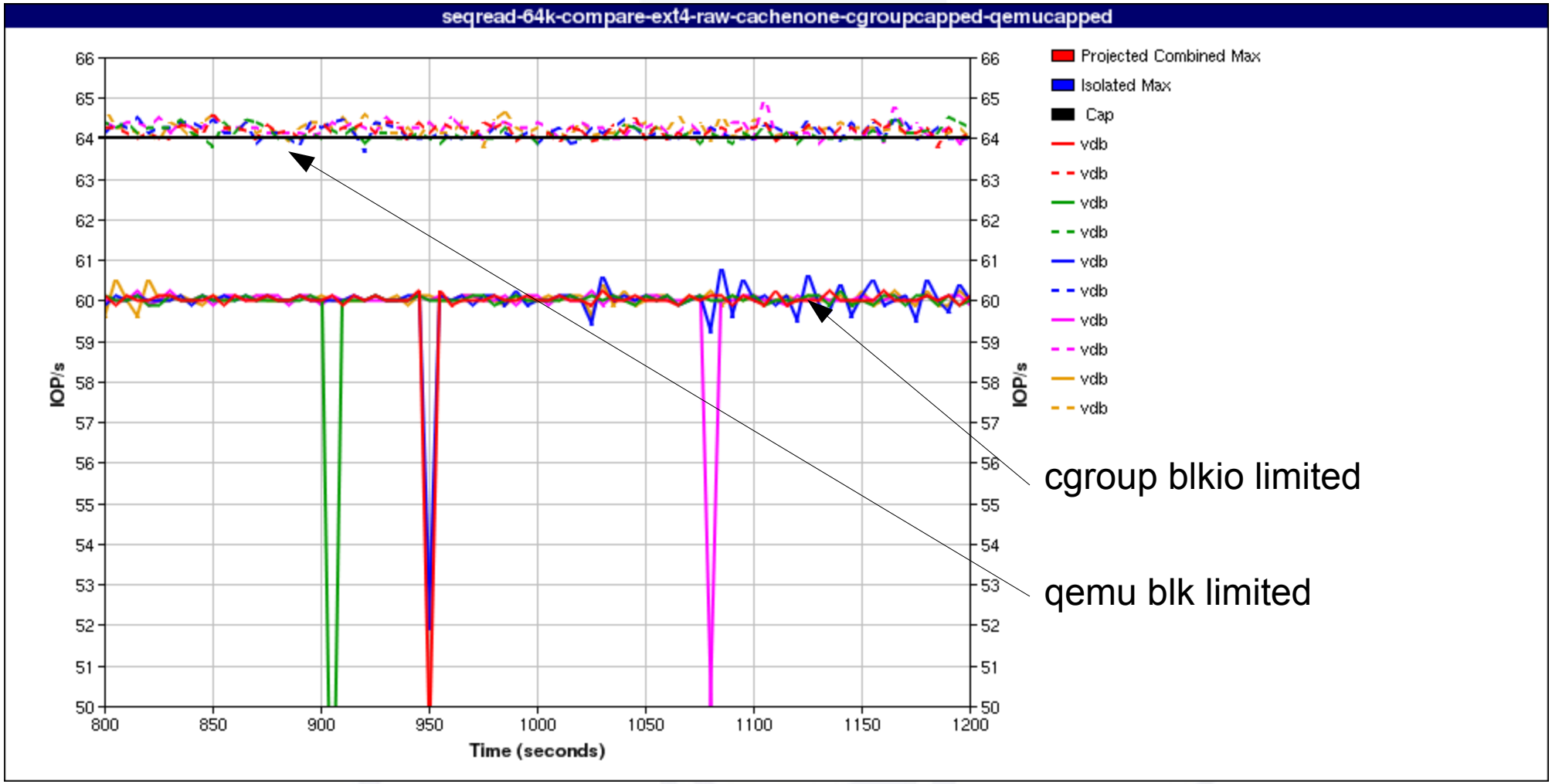
# Host Config

- IBM System x iDataPlex dx360 M3
  - 2x Intel X5670 @ 2.93GHz
  - 128G RAM
  - 5 2TB SATA
  - 2 1G Intel NIC
  - 1 10G Emulex NIC
- RHEL 6.1
- ioscheduler=deadline

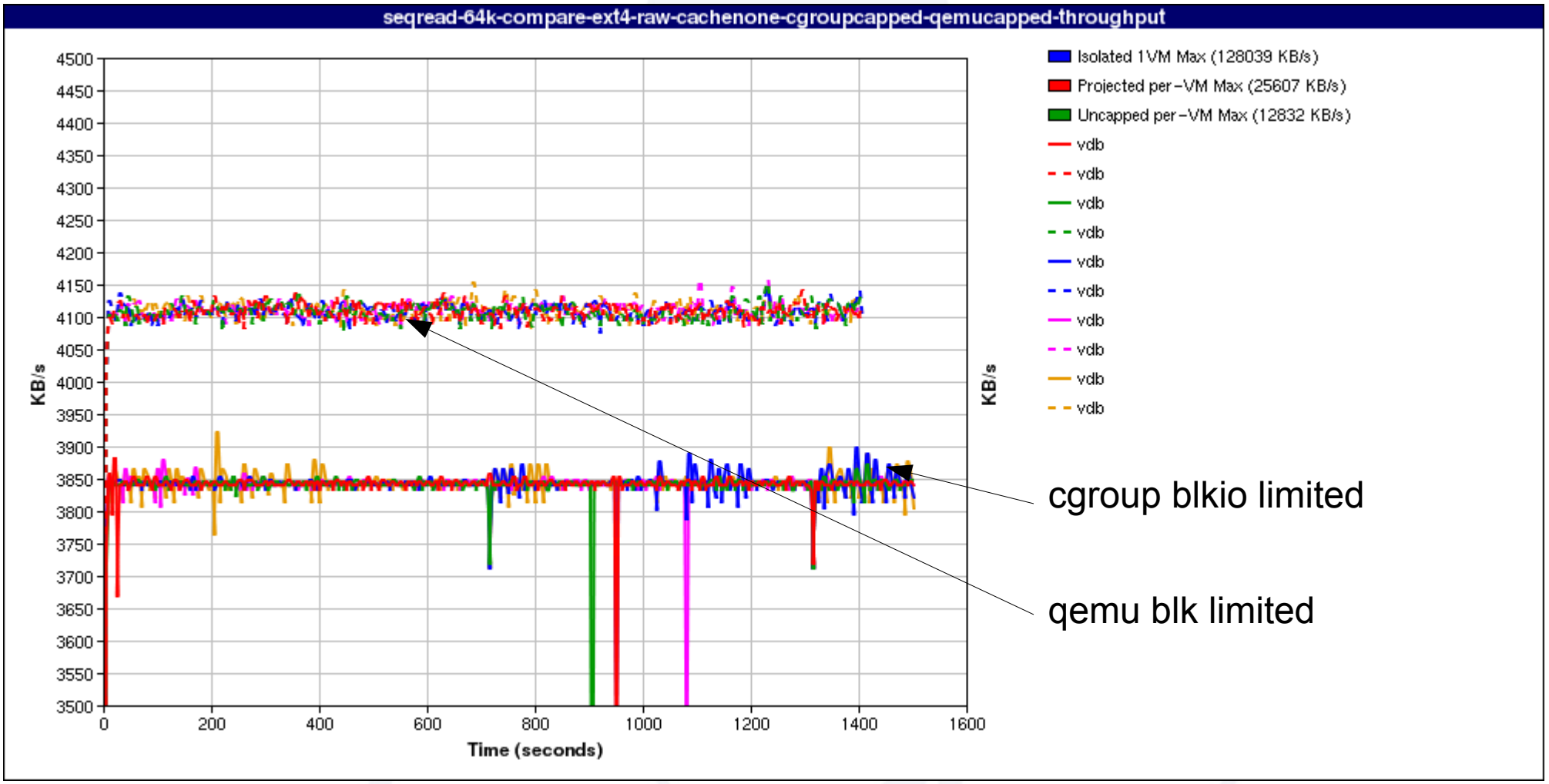

# CFQ vs Deadline



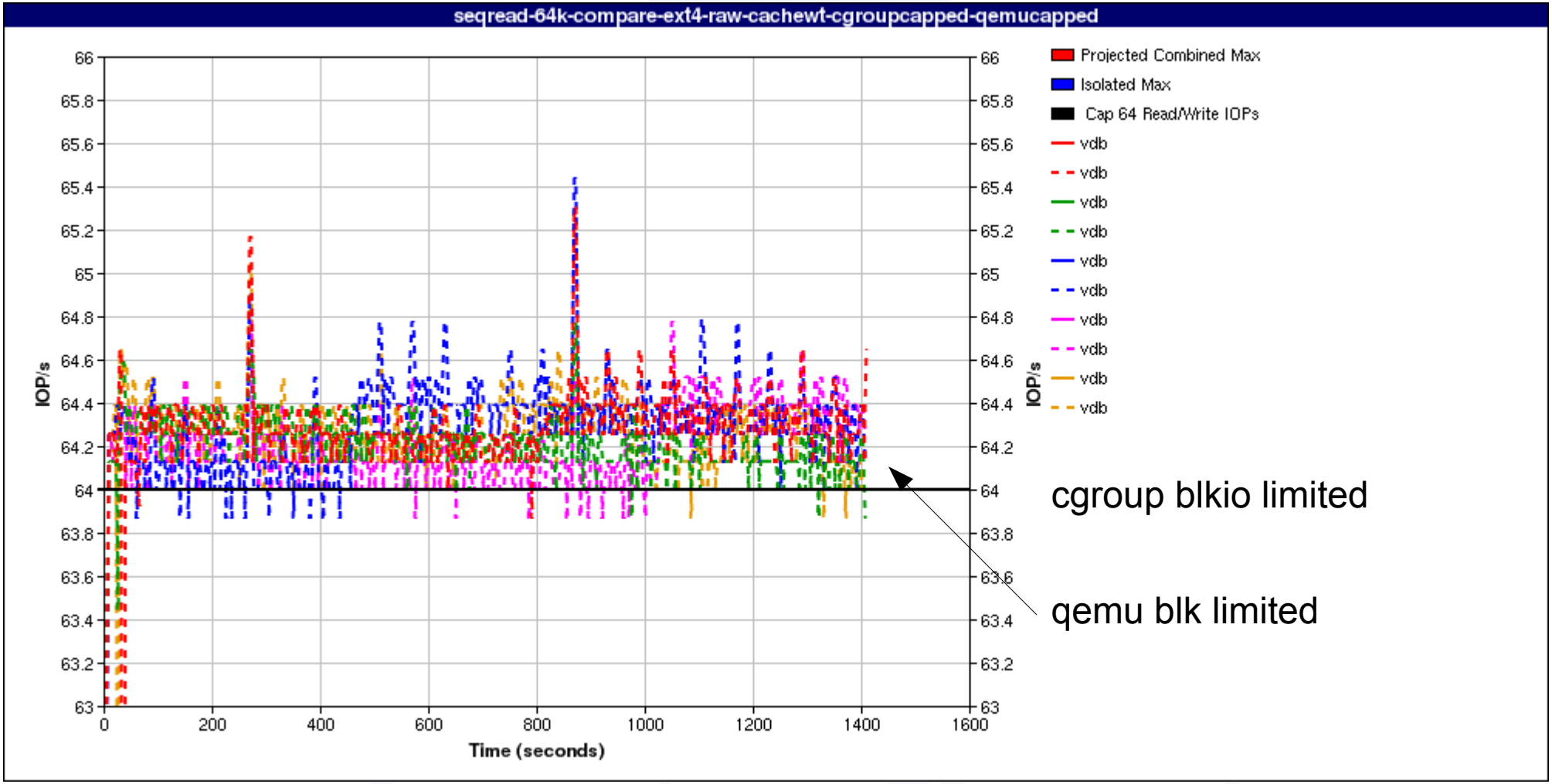
# Cgroup vs QEMU – IOPs cache=none



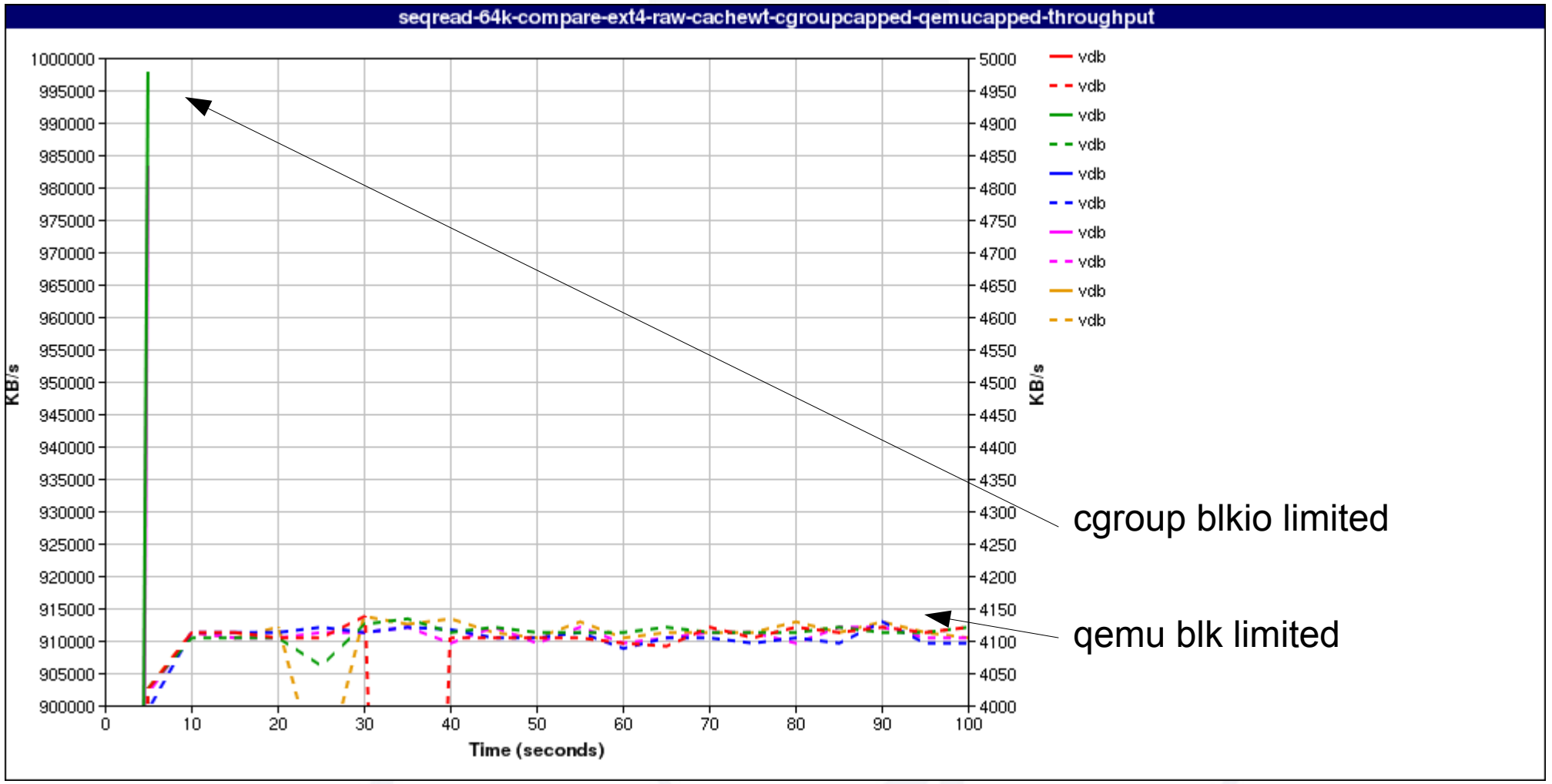
# Cgroup vs QEMU – Throughput cache=none



# Cgroup vs QEMU – IOPs cache=writethrough



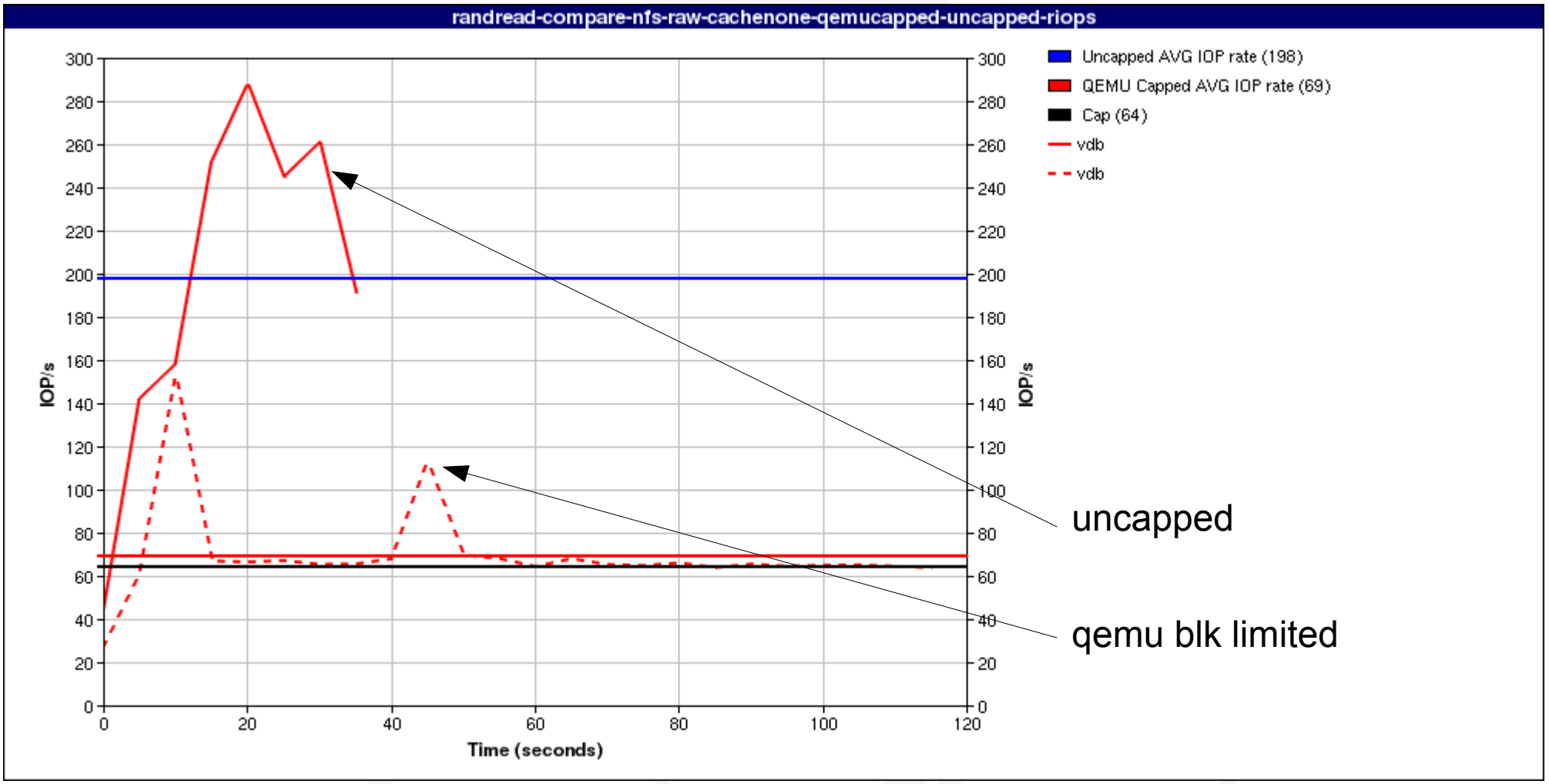
# Cgroup vs QEMU – Throughput cache=writethrough



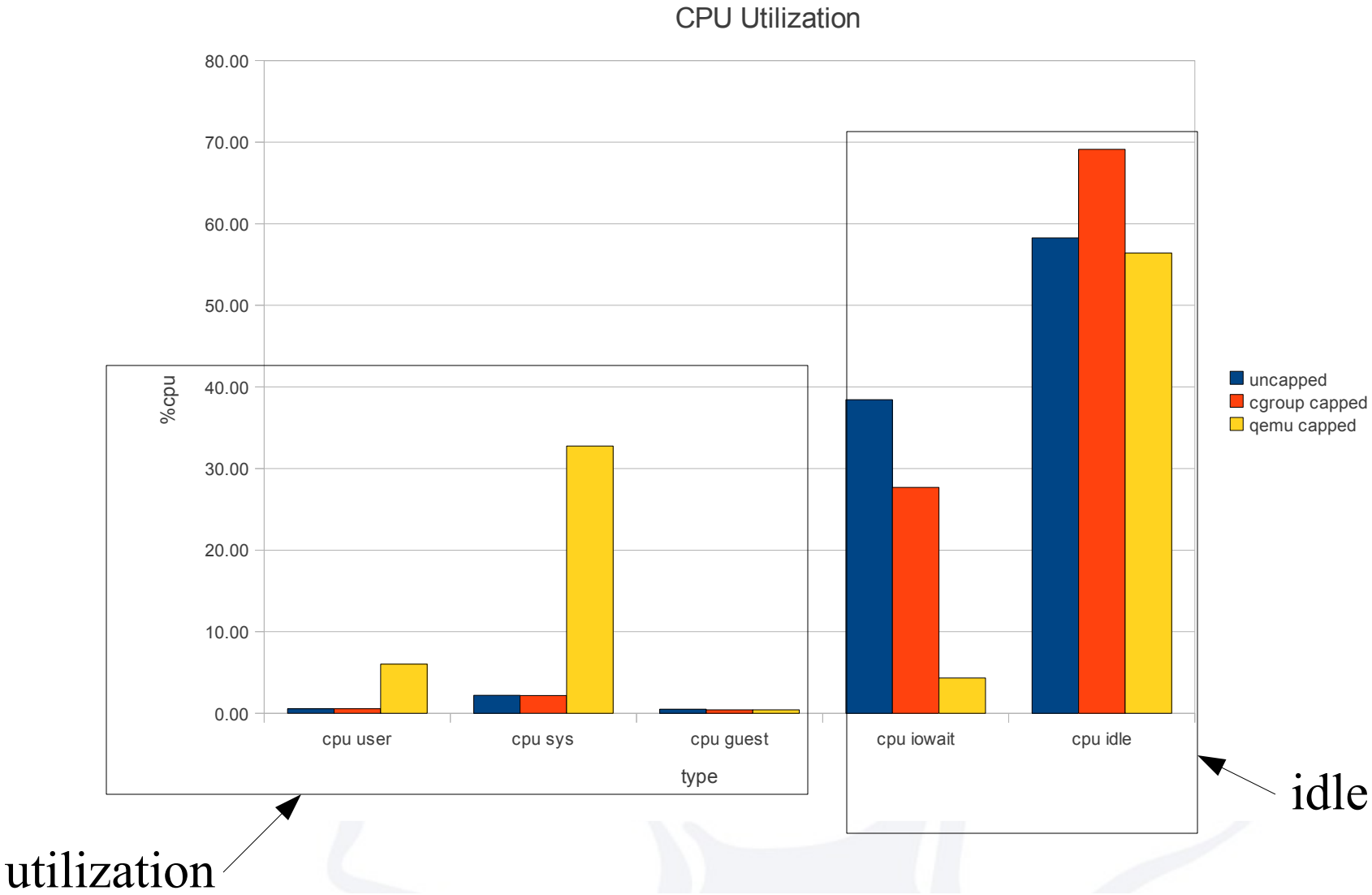
# QEMU Capped vs Uncapped cache=none, nfs-backed

# QEMU Capped vs Uncapped -- Throughput cache=none, nfs-backed

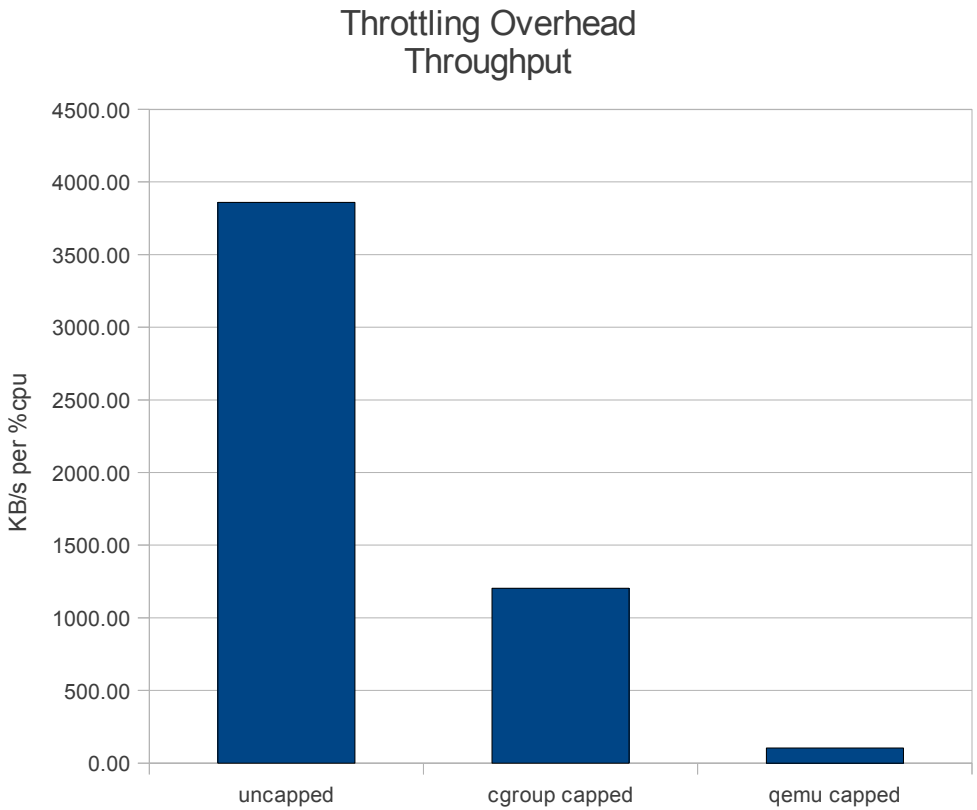
# QEMU Capped vs Uncapped -- IOPs cache=none, nfs-backed

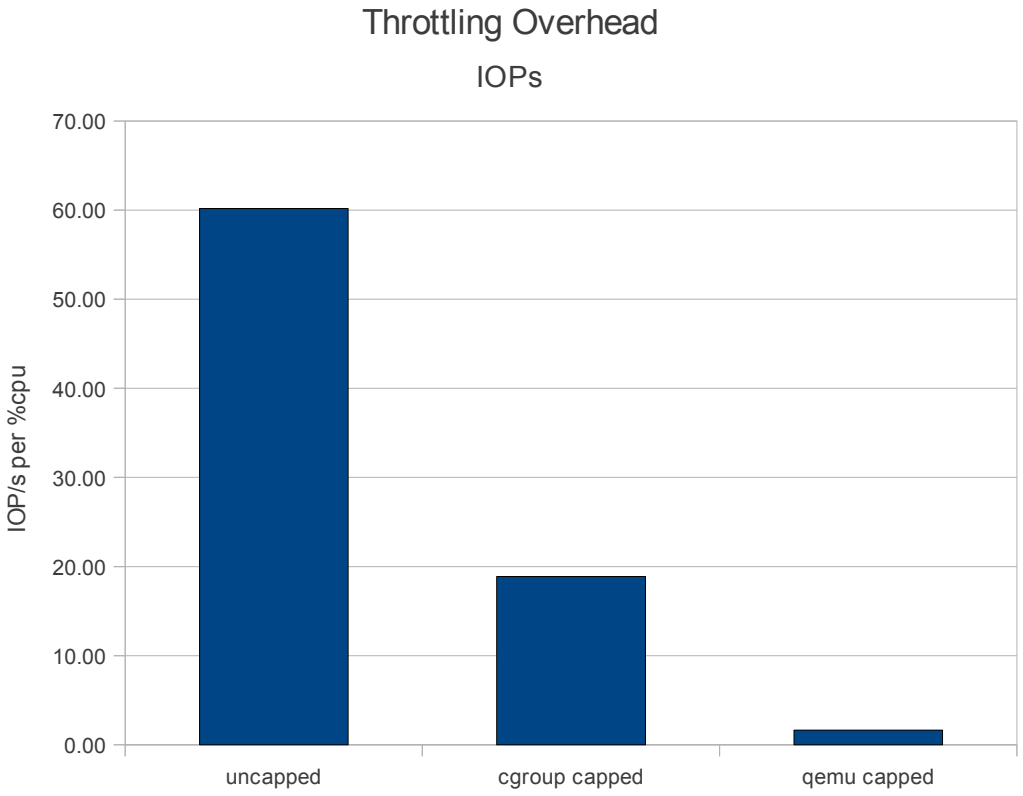


# Throttling Cost -- utlization



# Work per %cpu


Throttling Overhead
Throughput
KB/s per %cpu
4500.00
4000.00
3500.00
3000.00
2500.00
2000.00
1500.00
1000.00
500.00
0.00
uncapped
cgroup capped
qemu capped


Throttling Overhead
IOPs
IOP/s per %cpu
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
uncapped
cgroup capped
qemu capped

# Next Steps

- Algorithm improvements
  - Focus on preventing spikes
- Reduce CPU consumption
  - Data are incomplete but suggests there is room for improvement
-
  -

# Questions?

- http://wiki.qemu.org/Features/DiskIOLimits