Virtualizing NUMA

Andre Przywara, AMD OSRC, Dresden

Virtualization developer (KVM and Xen)

Work areas:

- NUMA
- CPUID
- Cross vendor migration

NUMA architecture

- Driven by integrated memory controllers
- Performance optimization
- ACPI based
- Smaller guests scale well
- Guests may exceed one node's resources
  - They should know!
- Scheduling should be restricted
  - (or be very clever)

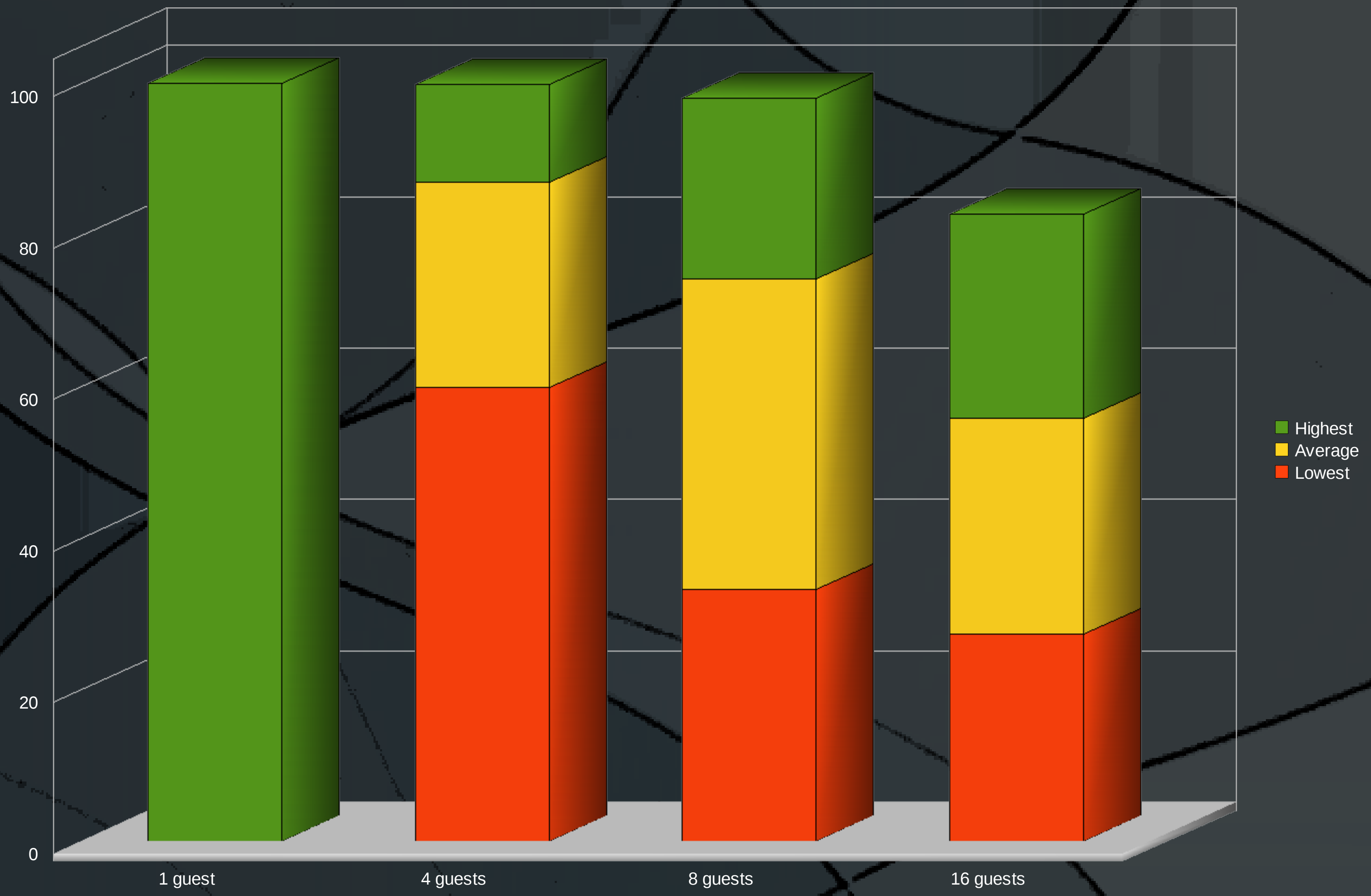
State of integration

- QEMU: can emulate in guest
  - KVM host binding patches pending
- Xen: patches posted, but need more work
  - Proper topology emulation required
    - No. of Cores must match NUMA topology
  - Both HVM and PV targetted

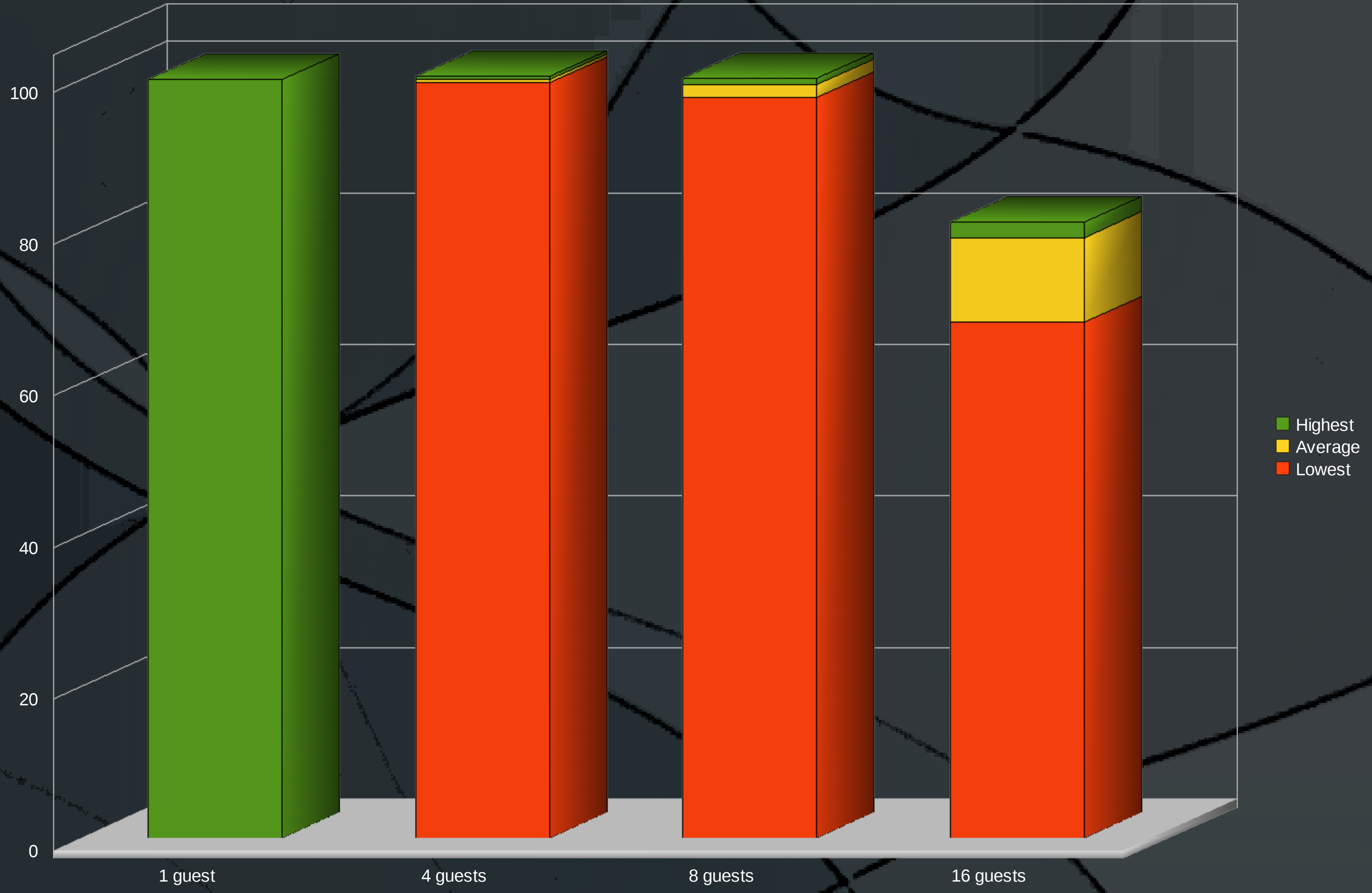
Numbers

- Four-way AMD Opteron 6164
  - Contains 8 nodes, 6 cores each
  - Each node has 8 or 16GB of RAM
- Kernbench:
  - different no. of VCPUs and RAM
  - Numactl'ed or not
- Lmbench
  - Starting many instances in parallel
  - Helping scheduler or not

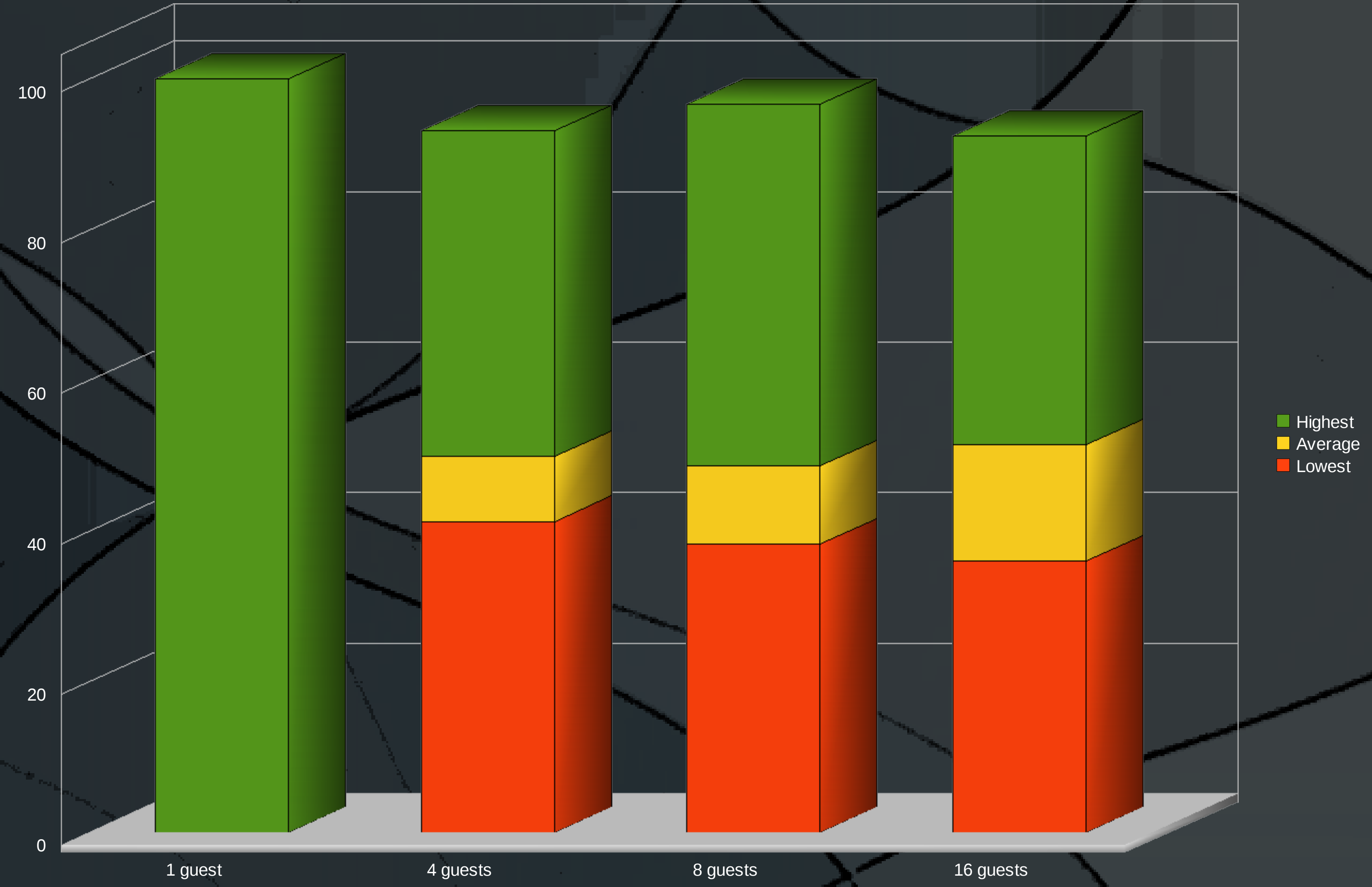
Lmbench (rd) KVM unpinned



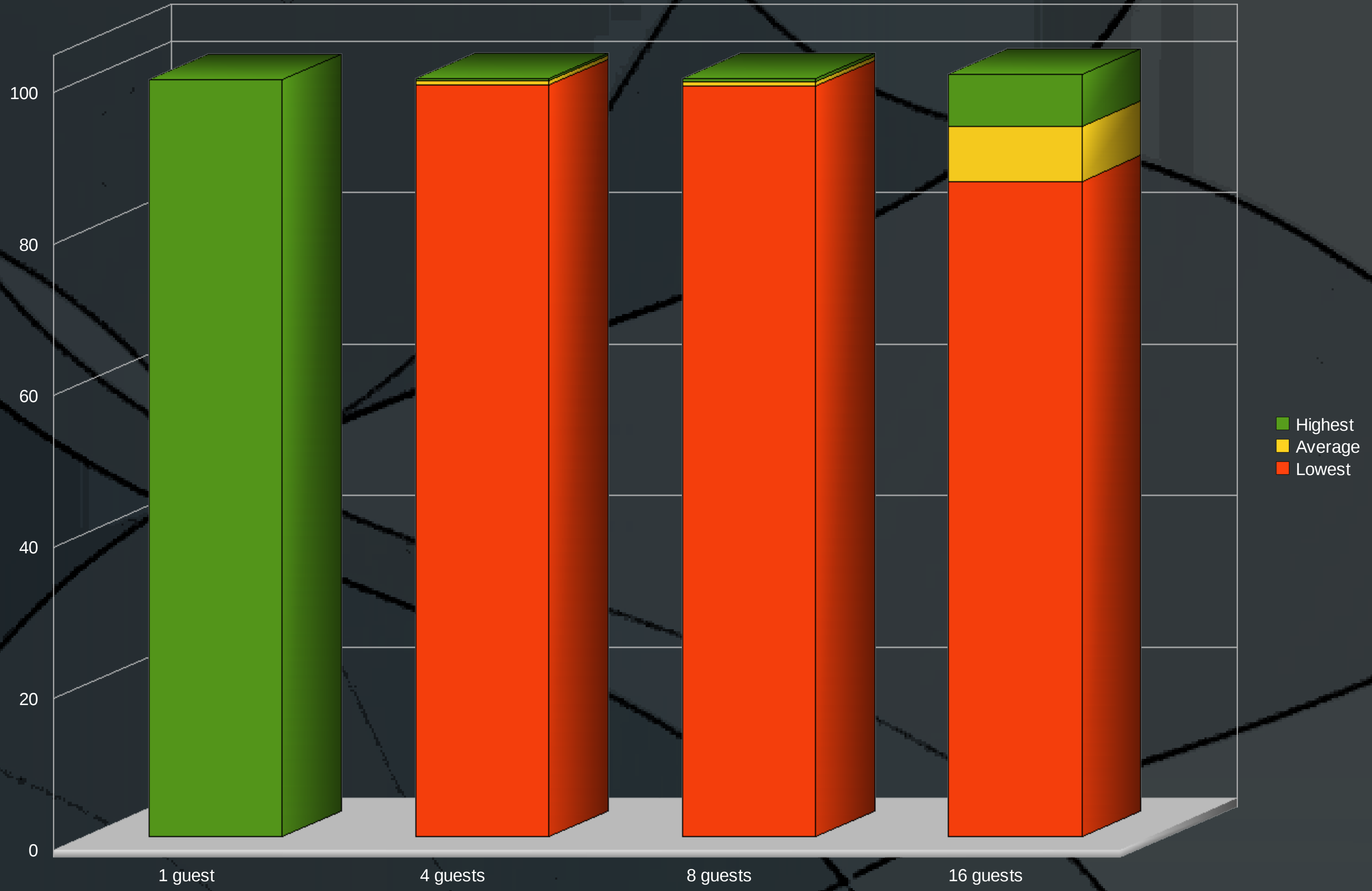
Lmbench (rd) KVM numactl



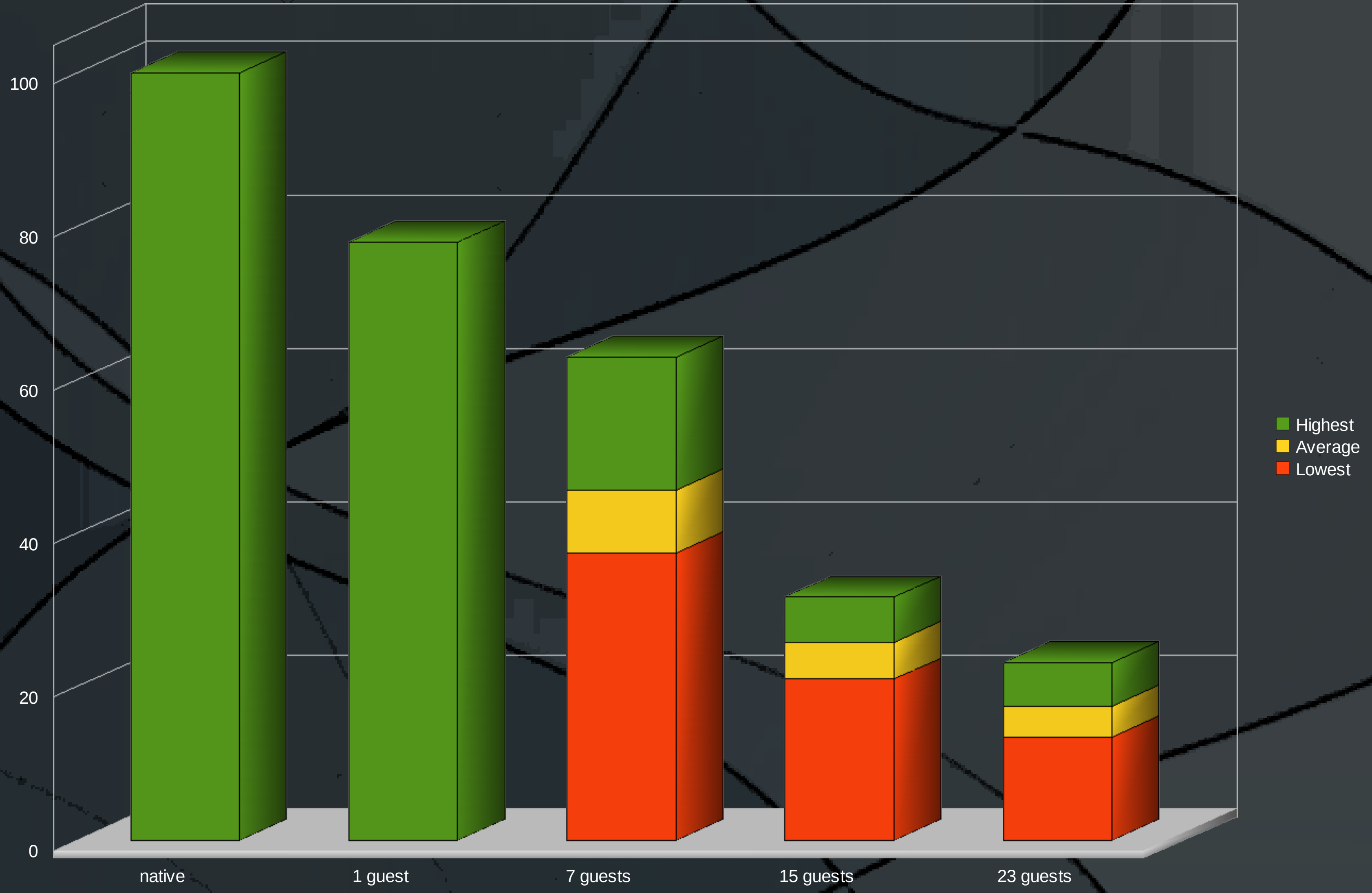
Lmbench (lat) KVM unpinned



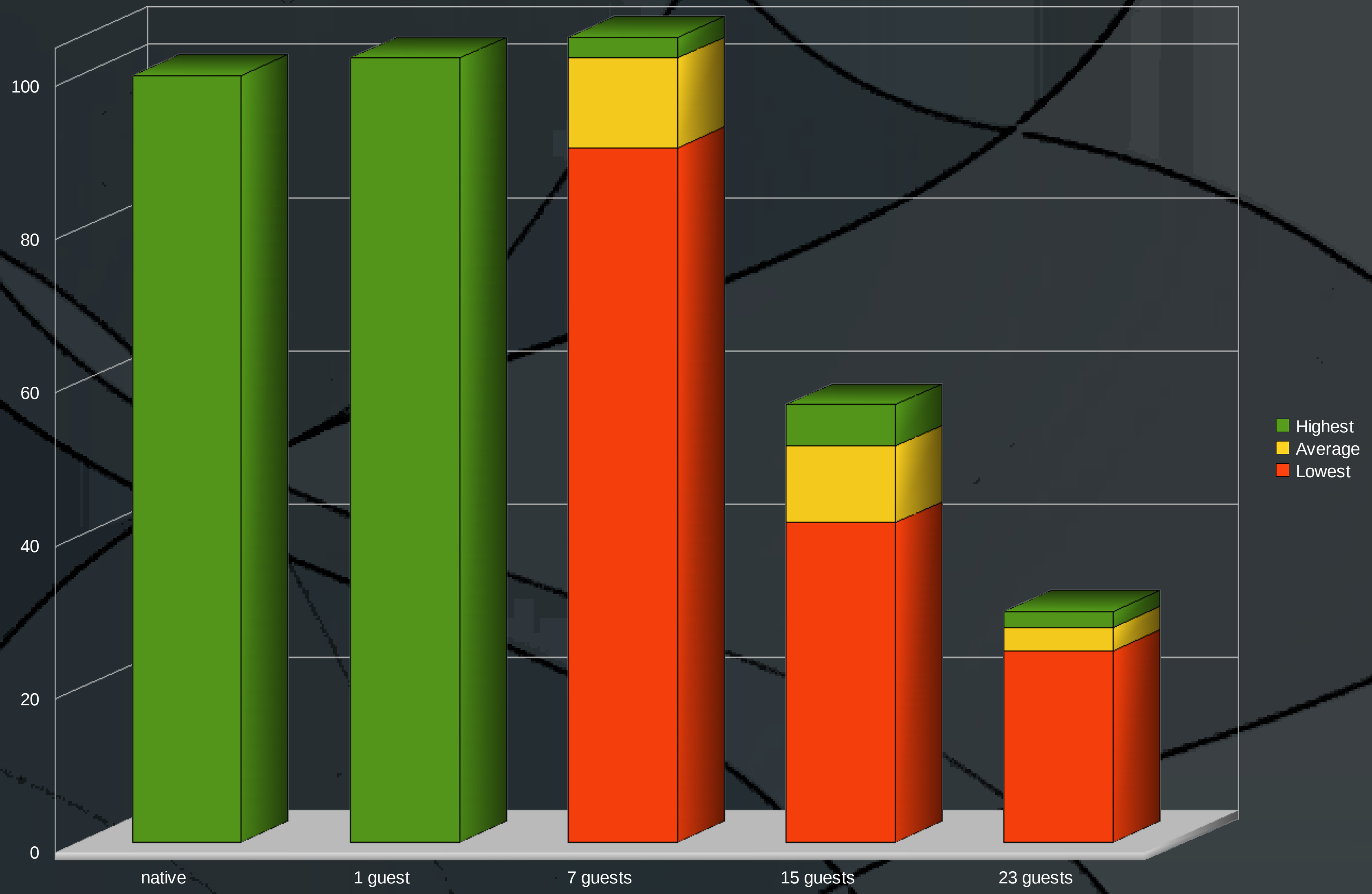
Lmbench (lat) KVM numactl



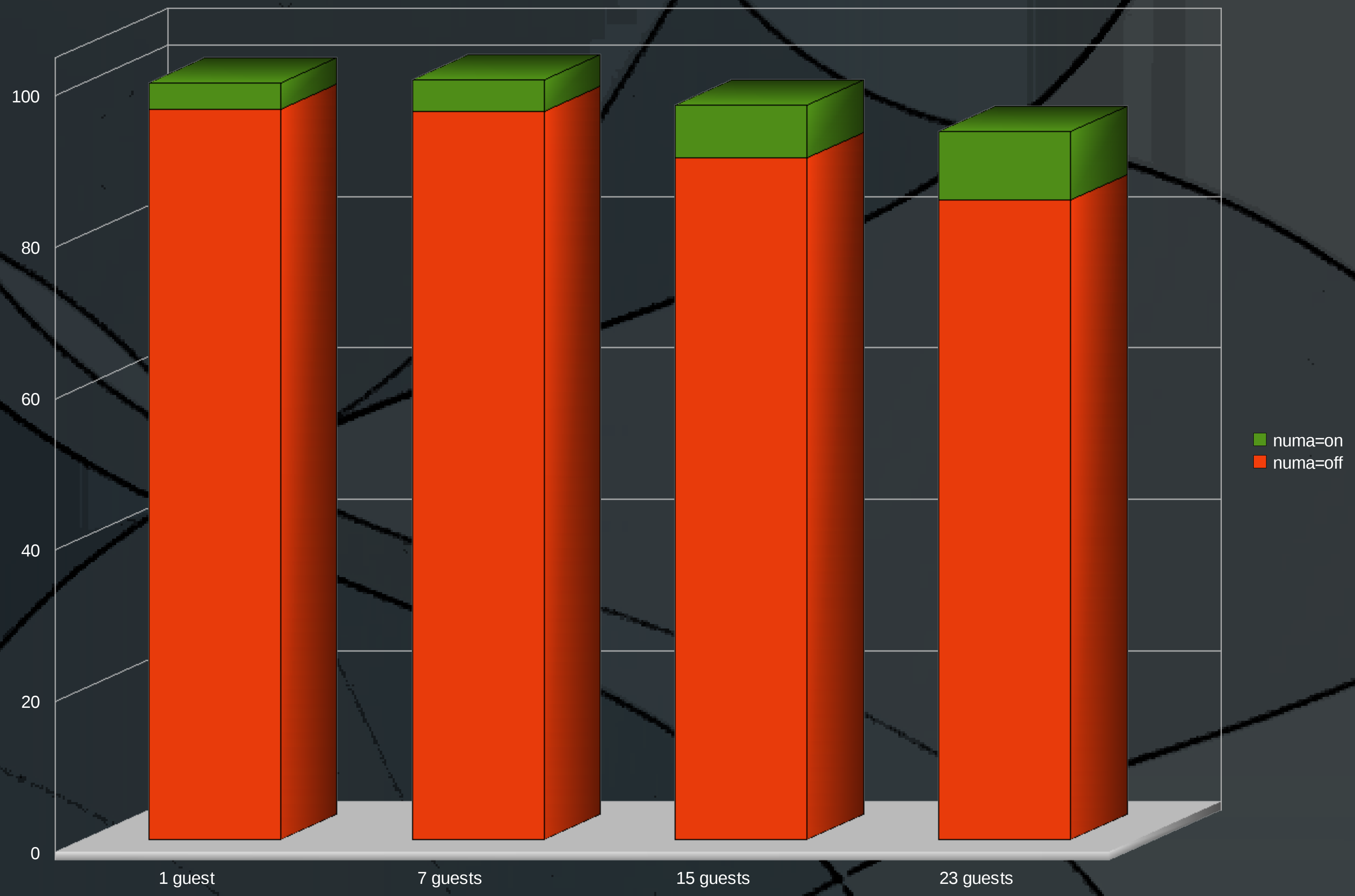
Lmbench (rd) Xen numa=off



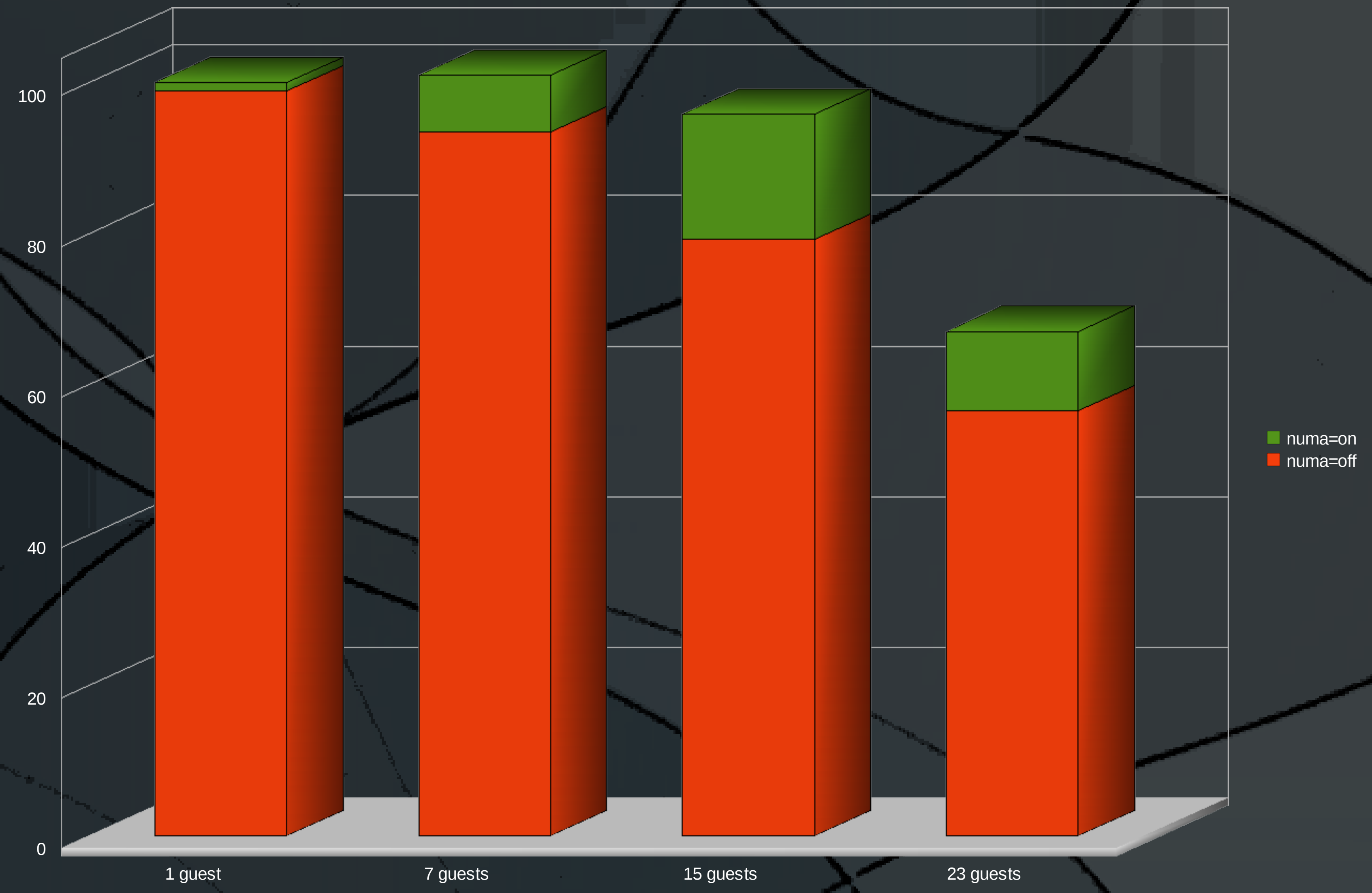
Lmbench (rd) Xen numa=on



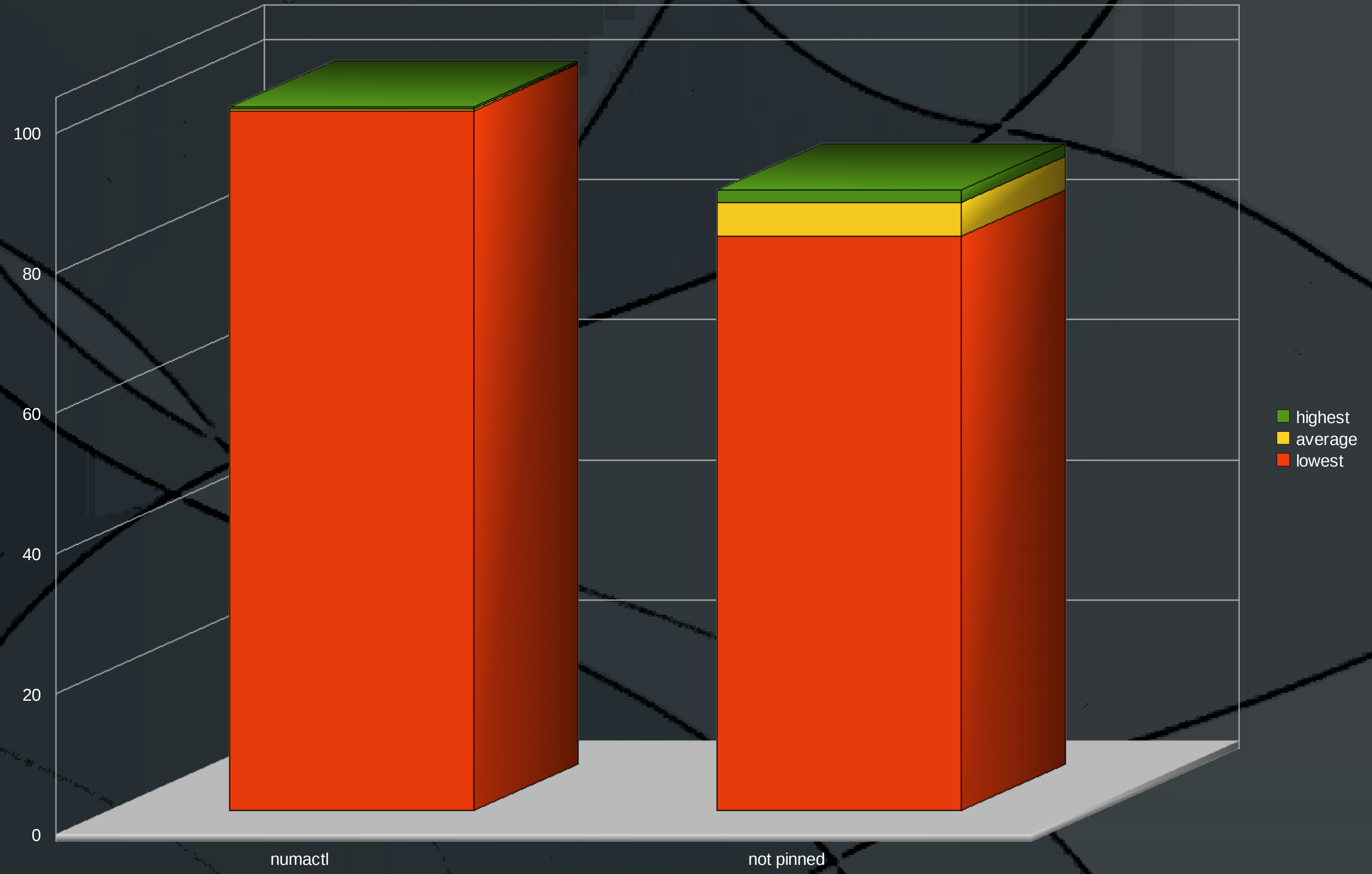
Kernbench Xen, 1 vCPU



Kernbench Xen, 2 vCPUs



Kernbench KVM, 7 guest



Discussion items

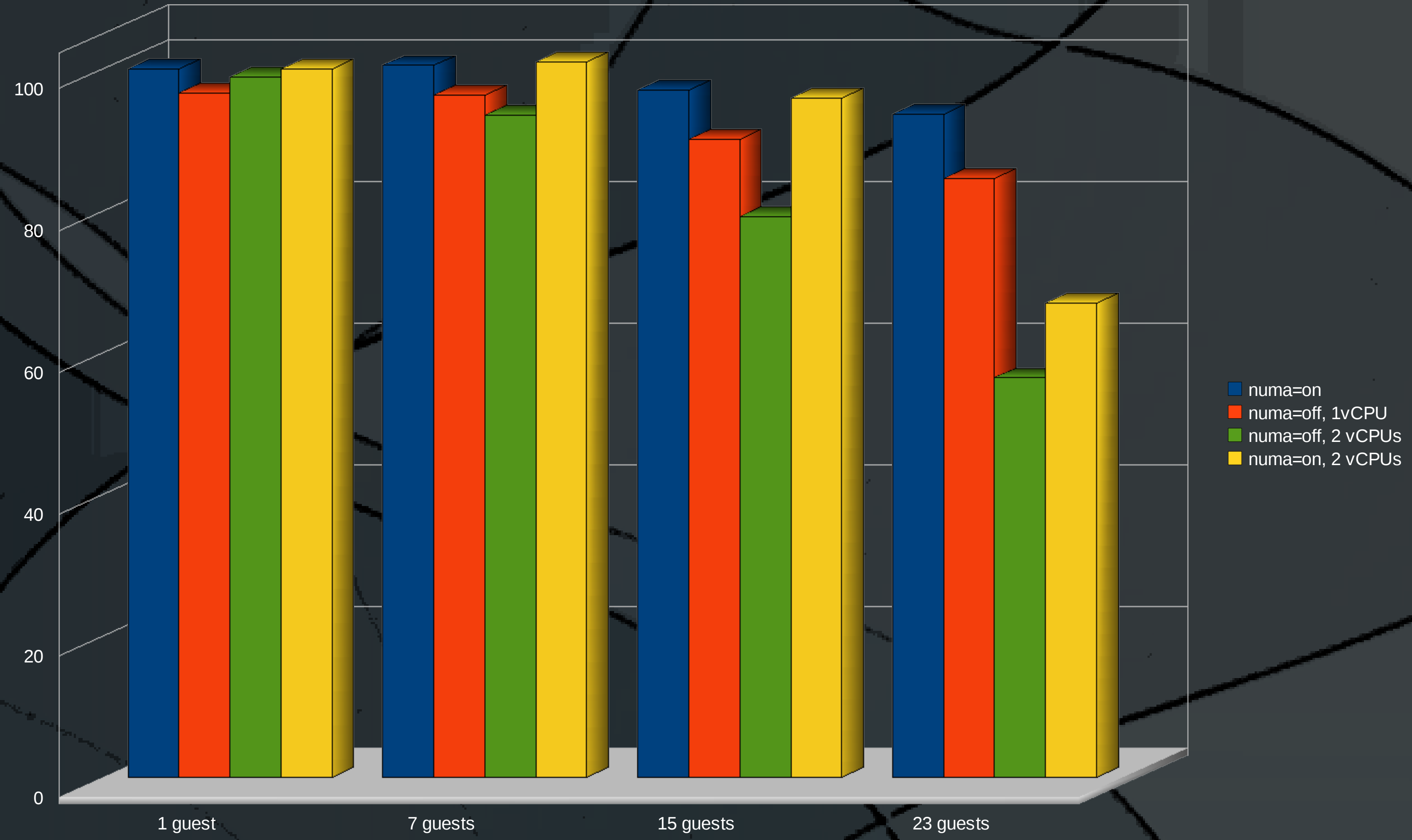
- Realization of KVM host NUMA binding
  - libnuma in QEMU
  - Externally by numactl or hugetlbfs
- Marry topology and NUMA?
- QEMU cmdline syntax for NUMA
  - Currently flexible, but hard to comprehend
  - Does it matter? (libvirt)
  - Unfortunate limits with comma

Scheduler items

- Avoid pinning (denies load balancing)
- But avoid node migration
- Schedule guests apart
  - Like Xen, but without pinning
- Rebalancing with page migration?
  - Hot pages first, maybe temporary?

Backup

Kernbench Xen


1 guest
7 guests
15 guests
23 guests
🟦 numa=on
🟥 numa=off, 1vCPU
🟩 numa=off, 2 vCPUs
🟨 numa=on, 2 vCPUs