# Linux PV on HVM

paravirtualized interfaces in HVM guests

**Stefano Stabellini**

# Linux as a guests: problems

Linux PV guests have limitations:

- ~~difficult~~ “different” to install
- some performance issue on 64 bit
- limited set of virtual hardware

Linux HVM guests:

- install the same way as native
- very slow

# Linux PV on HVM: the solution

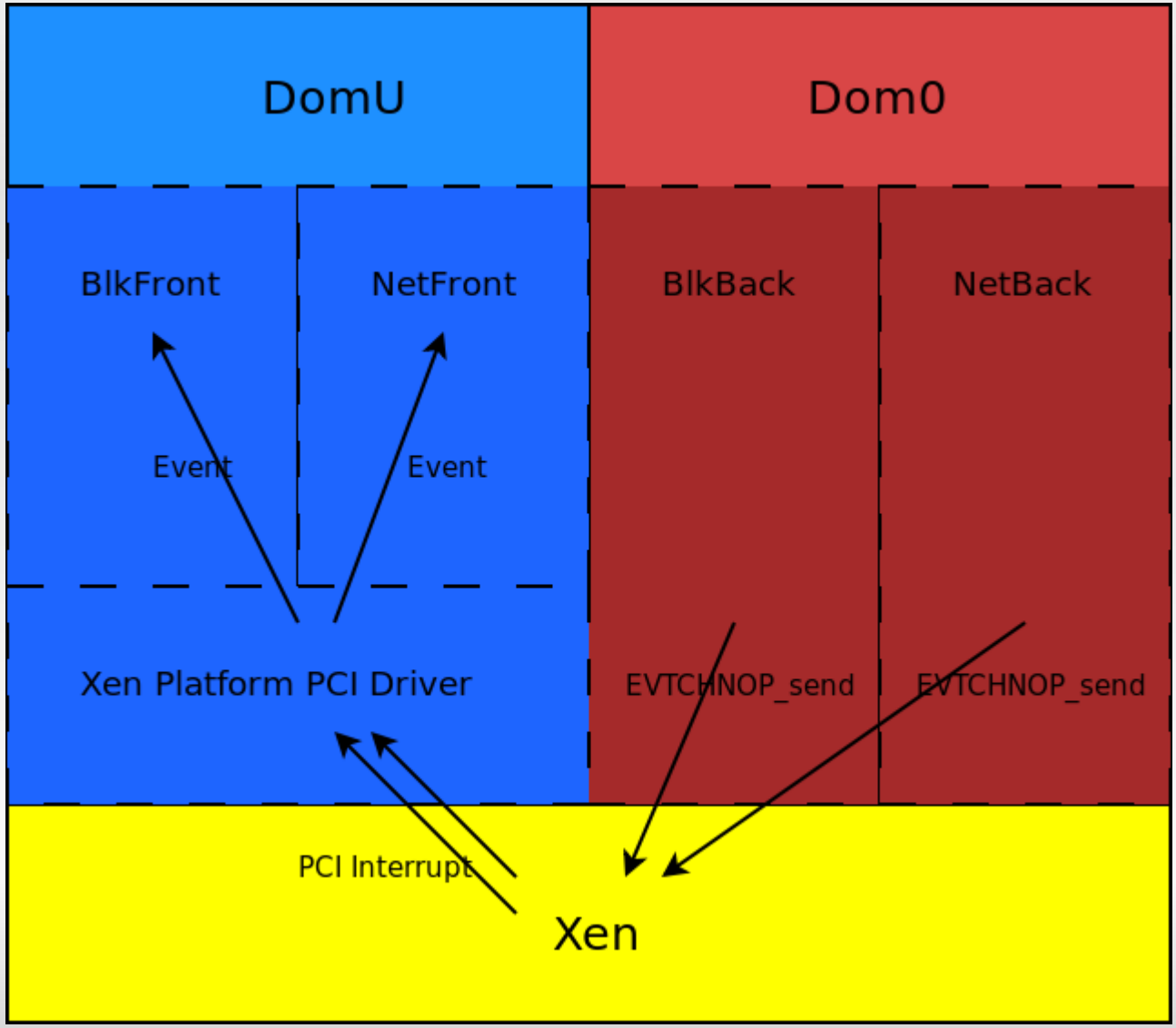
- install the same way as native
- PC-like hardware
- access to fast paravirtualized devices
- exploit nested paging

# Linux PV on HVM: initial feats

Initial version in Linux 2.6.36:

- introduce the xen platform device driver
- add support for HVM hypercalls, xenbus and grant table
- enables **blkfront**, **netfront** and **PV timers**
- add support to PV suspend/resume
- the **vector callback** mechanism

# Old style event injection



# Receiving an interrupt

do_IRQ

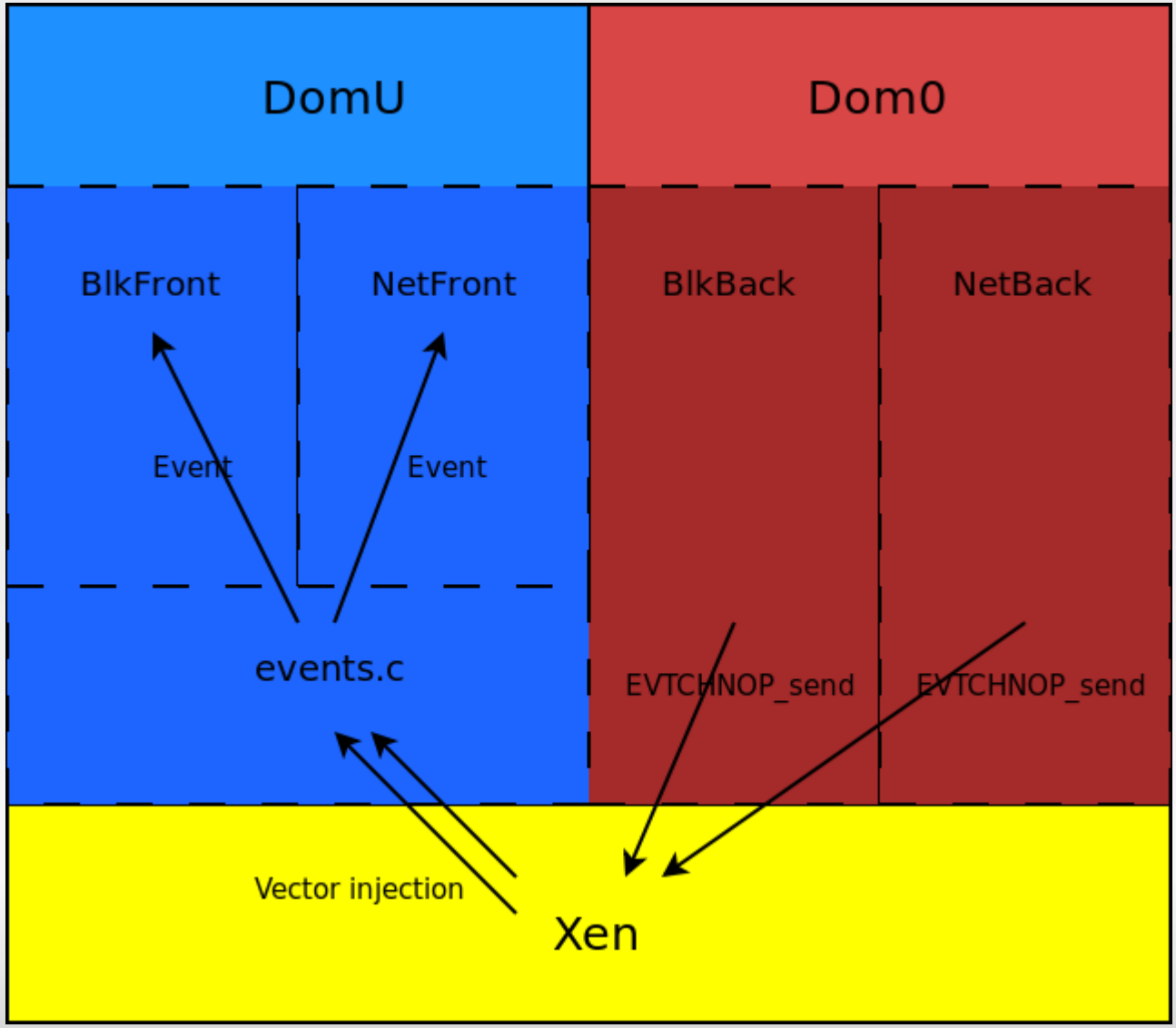
handle_fasteoi_irq

handle_irq_event

xen_evtchn_do_upcall

**ack_apic_level ← >=3 VMEXIT**

# The new vector callback



# Receiving a vector callback

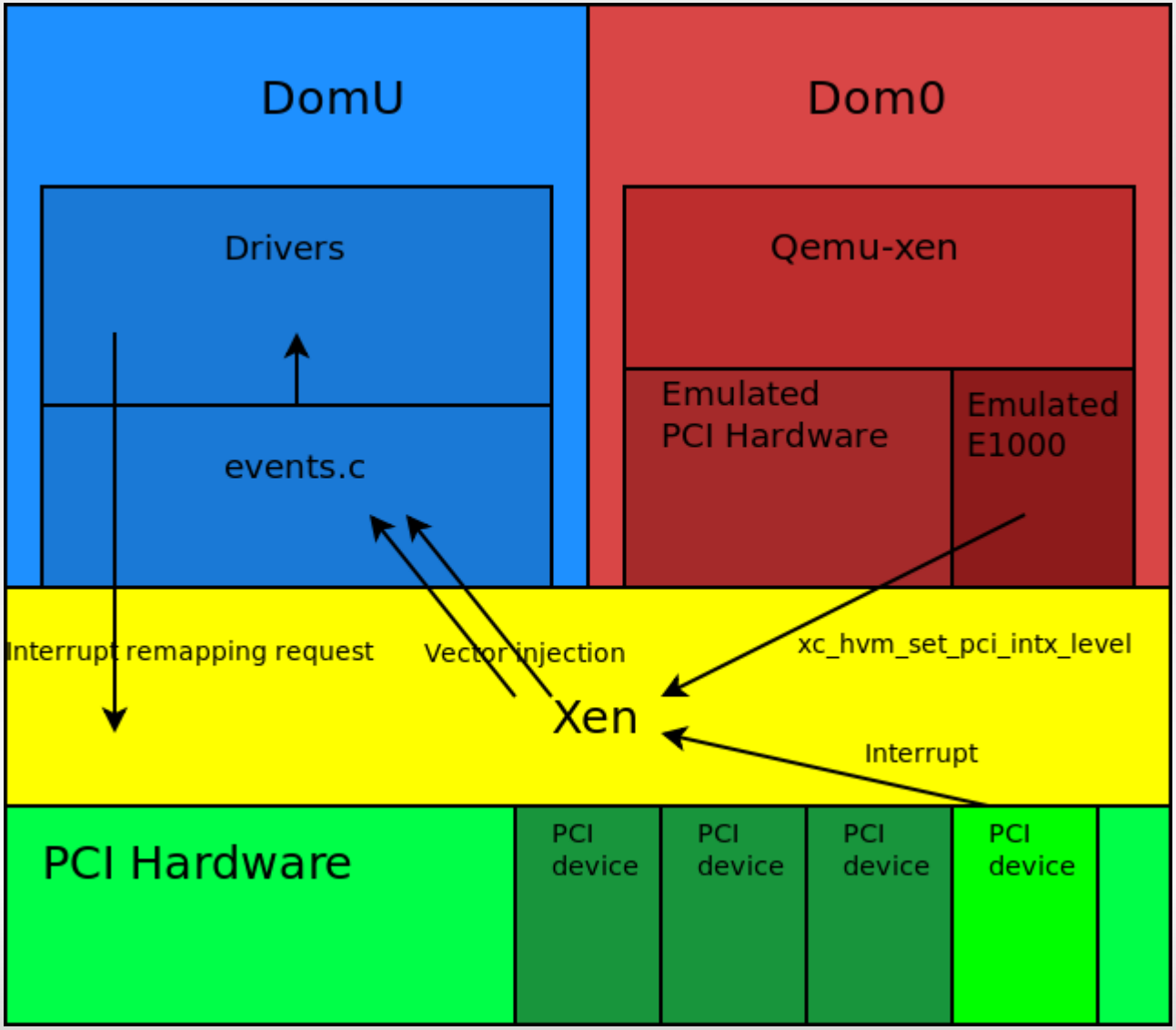
xen_evtchn_do_upcall

# Linux PV on HVM: newer feats

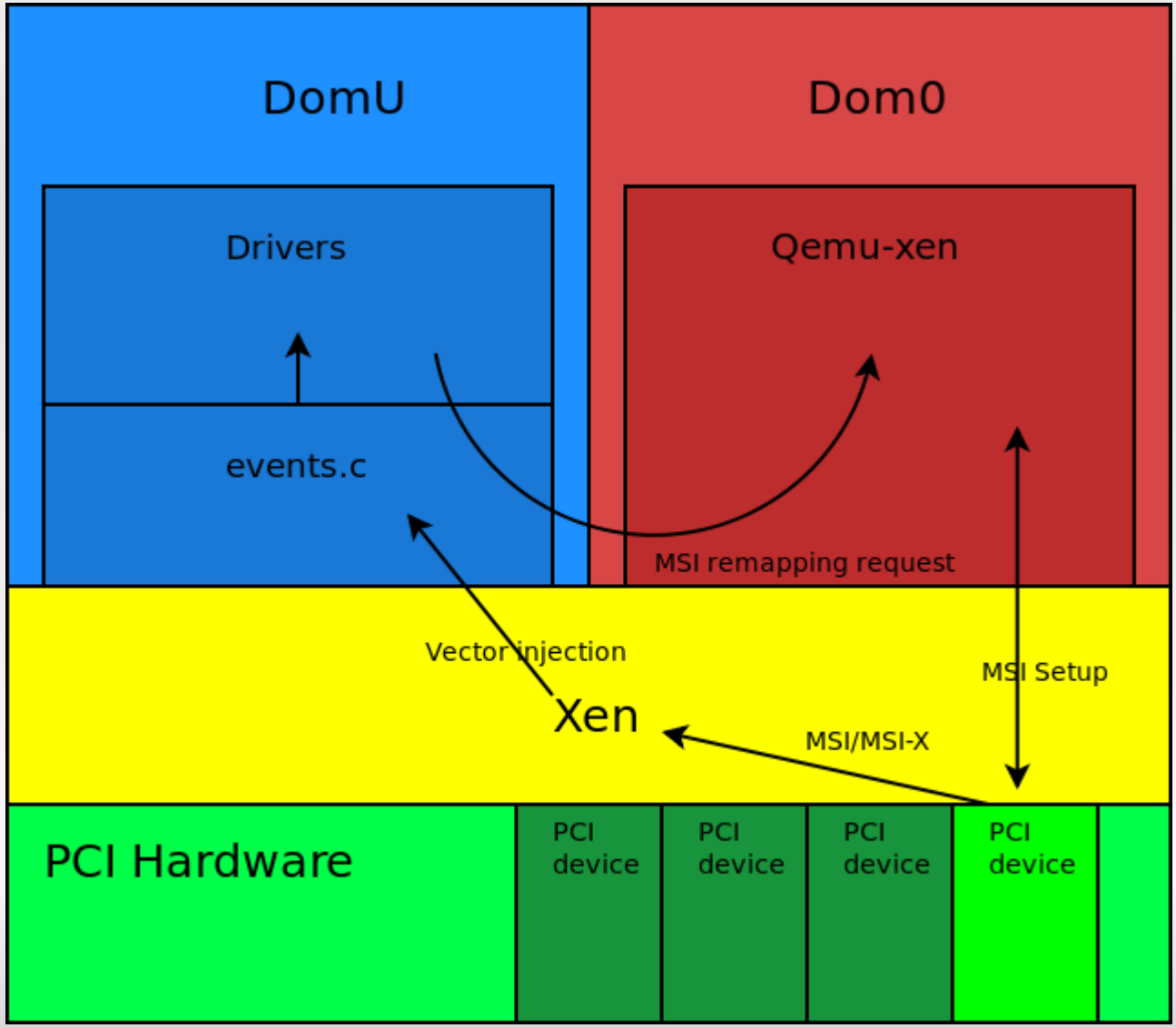
Later enhancements (2.6.37+):

- ballooning
- PV spinlocks
- PV IPIs
- Interrupt remapping onto event channels
- MSI remapping onto event channels

# Interrupt remapping



# MSI remapping



# PV spectrum

| | HVM guests | Classic PV on HVM | Enhanced PV on HVM | Hybrid PV on HVM | PV guests |
|---|---|---|---|---|---|
| Boot sequence | emulated | emulated | emulated | | paravirtualized |
| Memory | hardware | hardware | hardware | | paravirtualized |
| Interrupts | emulated | emulated | paravirtualized | | paravirtualized |
| Timers | emulated | emulated | paravirtualized | | paravirtualized |
| Spinlocks | emulated | emulated | paravirtualized | | paravirtualized |
| Disk | emulated | paravirtualized | paravirtualized | | paravirtualized |
| Network | emulated | paravirtualized | paravirtualized | | paravirtualized |
| Privileged operations | hardware | hardware | hardware | | paravirtualized |

# Benchmarks: the setup

## Hardware setup:

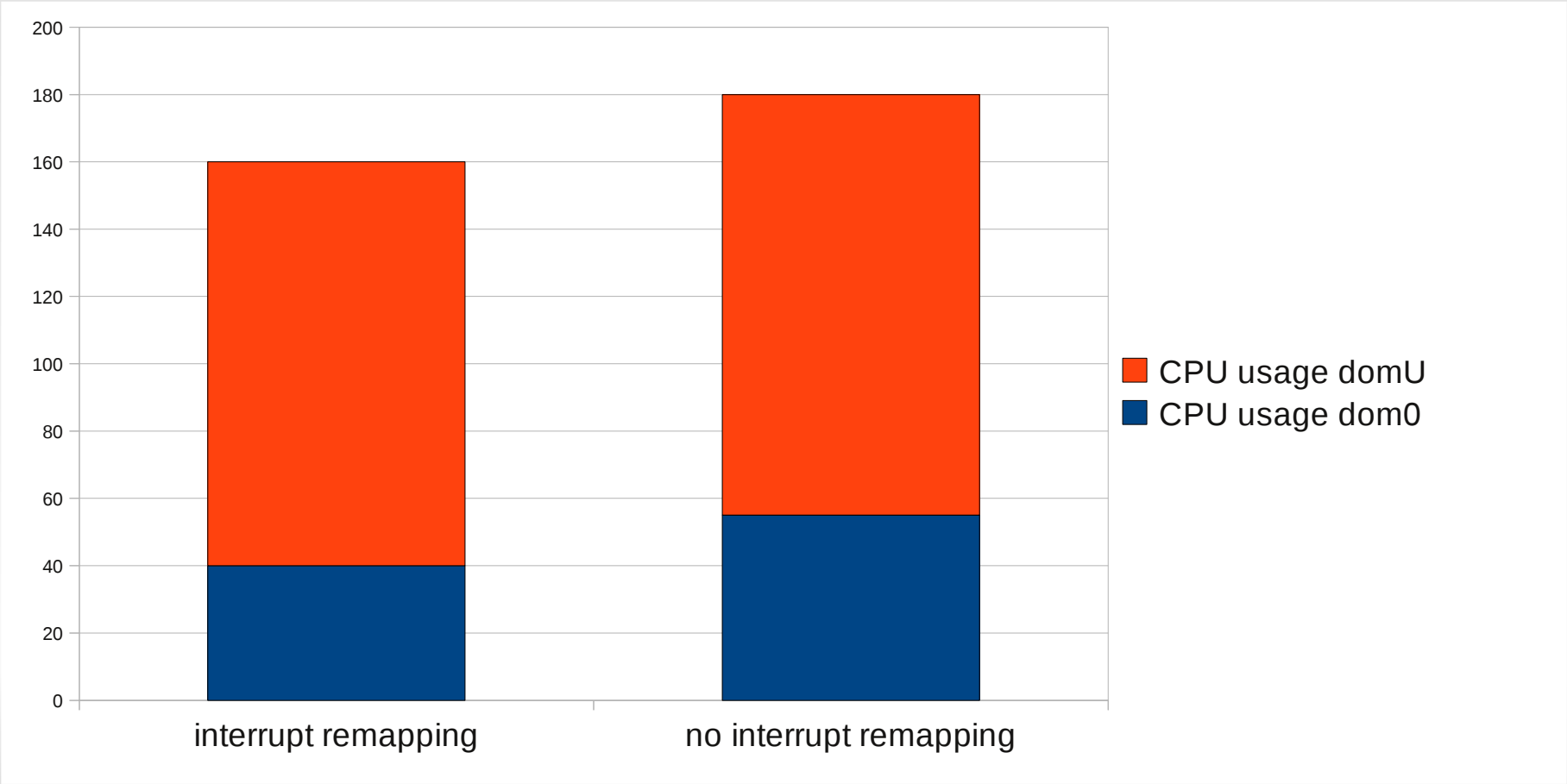
Dell PowerEdge R710
CPU: dual Intel Xeon E5520 quad core CPUs @ 2.27GHz
RAM: 22GB

## Software setup:

Xen 4.1, 64 bit
Dom0 Linux 2.6.32, 64 bit
DomU Linux 3.0 rc4, 8GB of memory, 8 vcpus

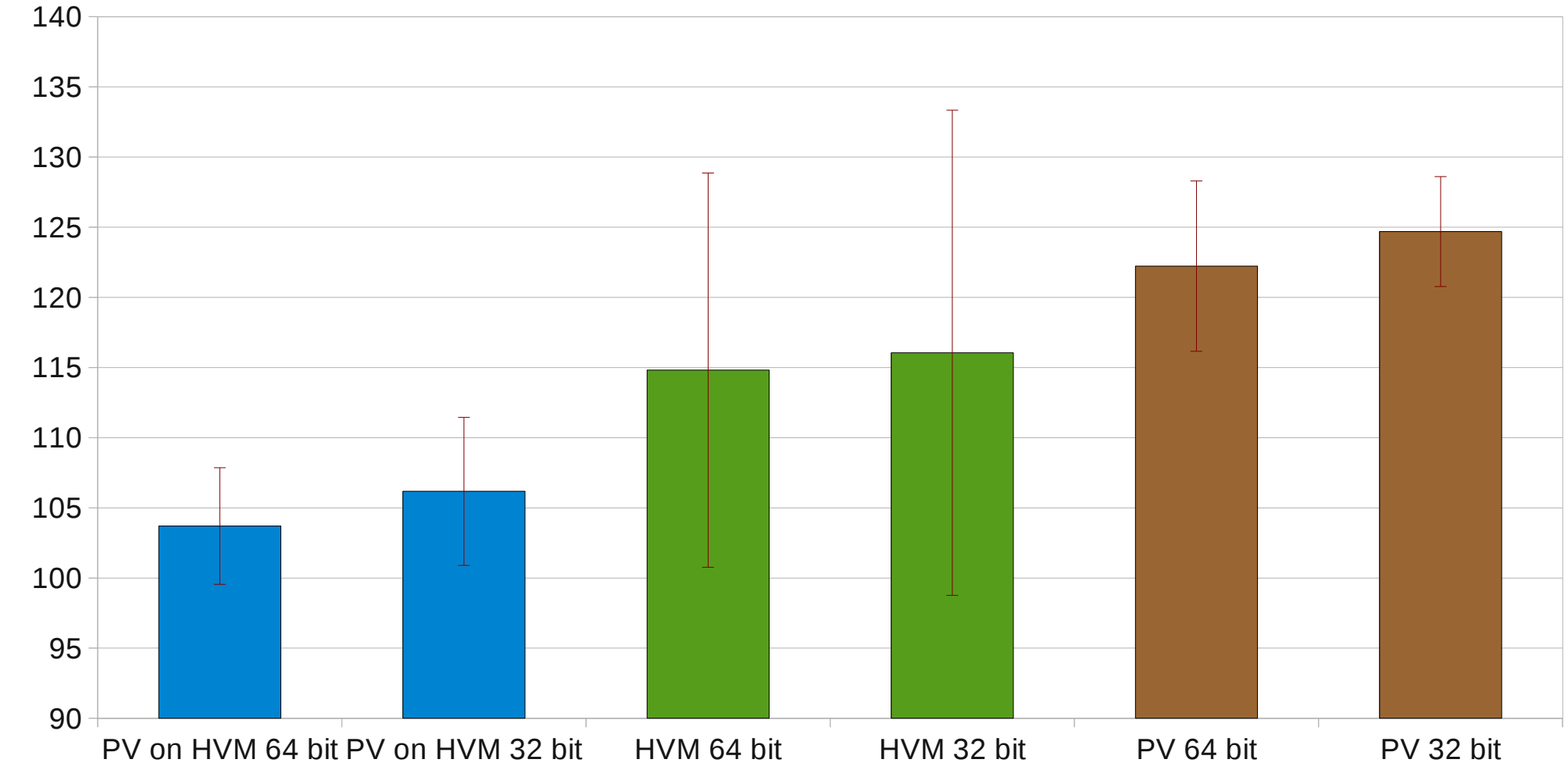
# PCI passthrough: benchmark

PCI passthrough of an Intel Gigabit NIC
CPU usage: the lower the better:



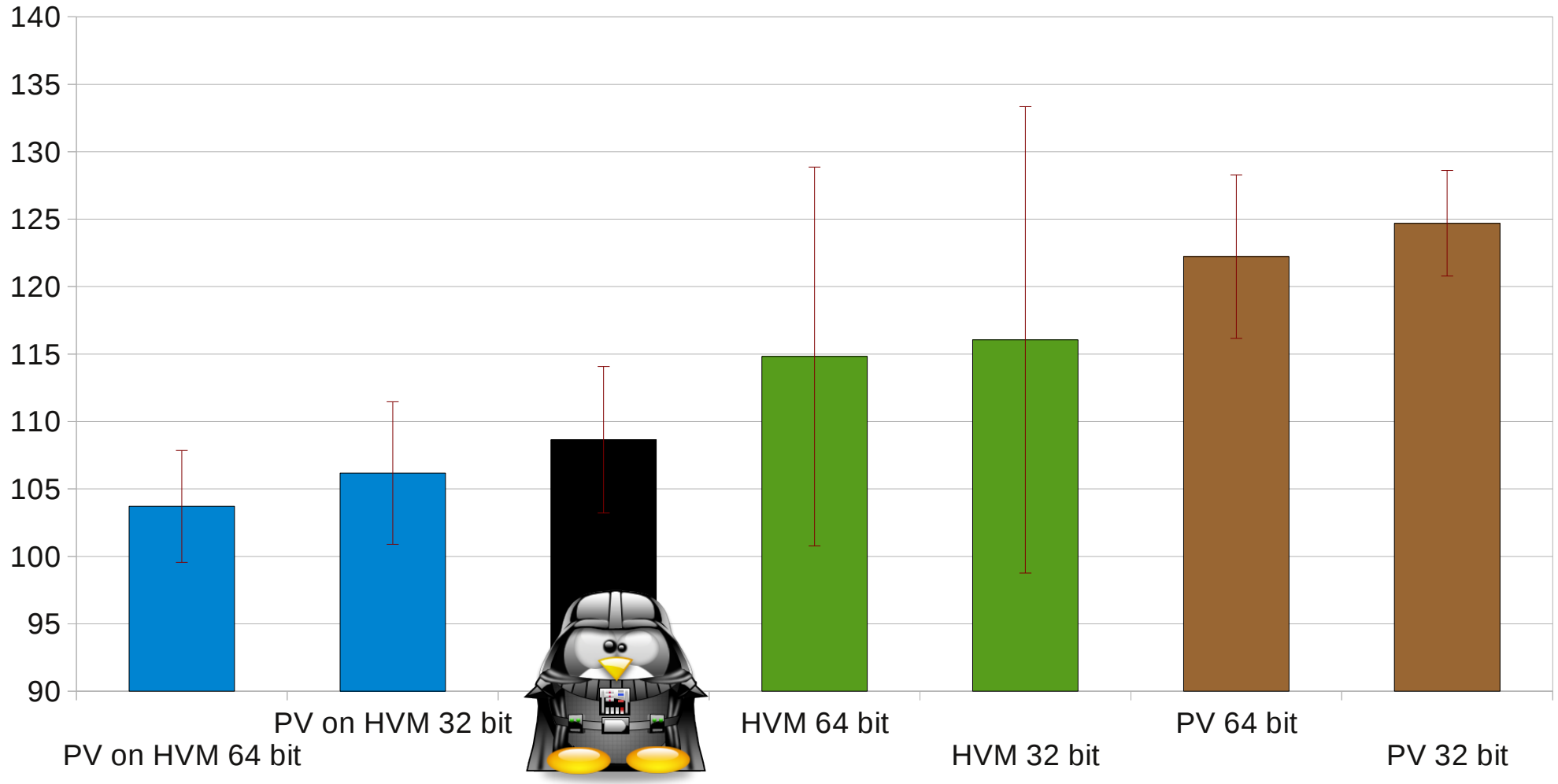
# Kernbench

Results: percentage of native, the lower the better



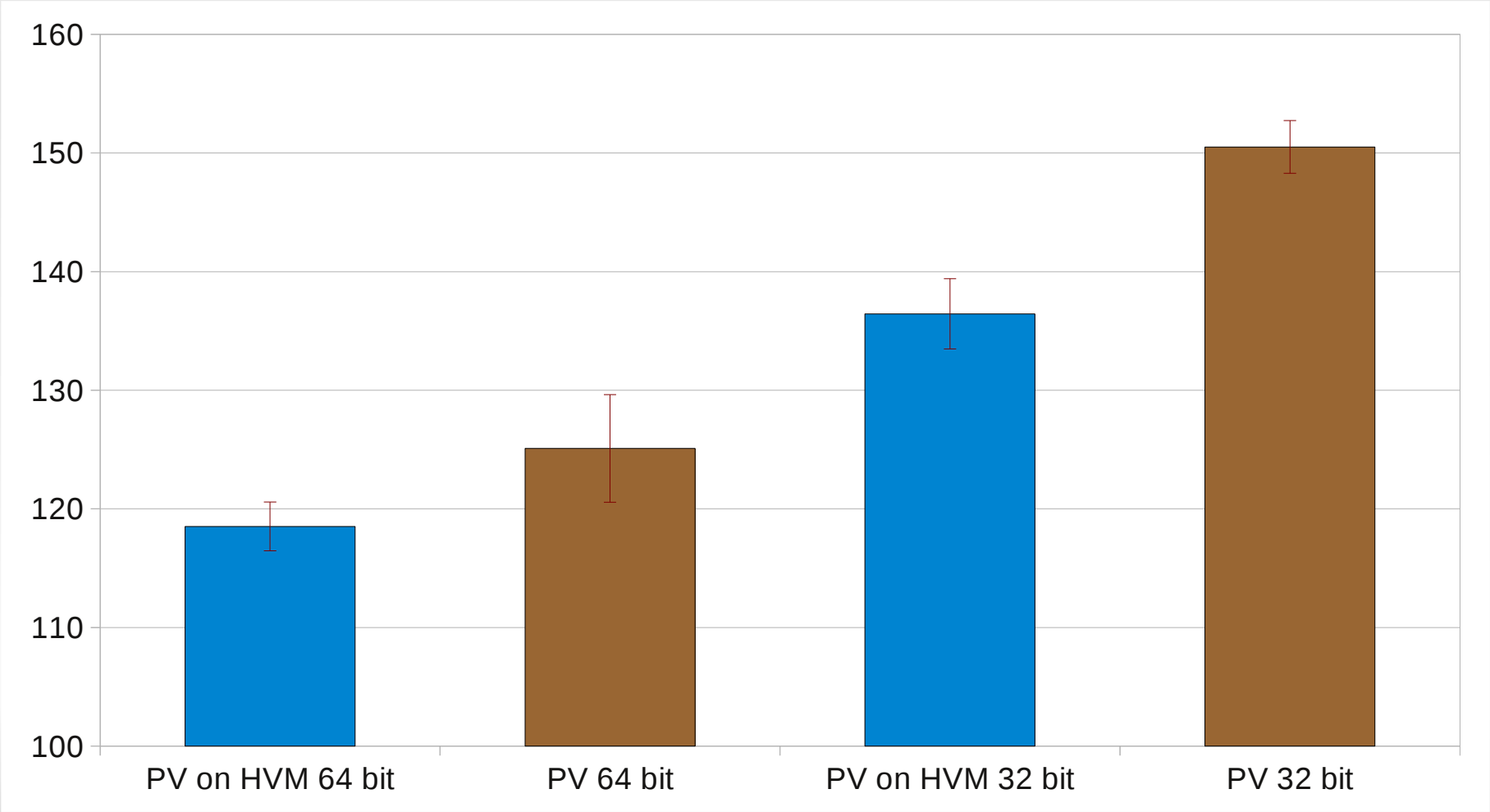
# Kernbench

Results: percentage of native, the lower the better



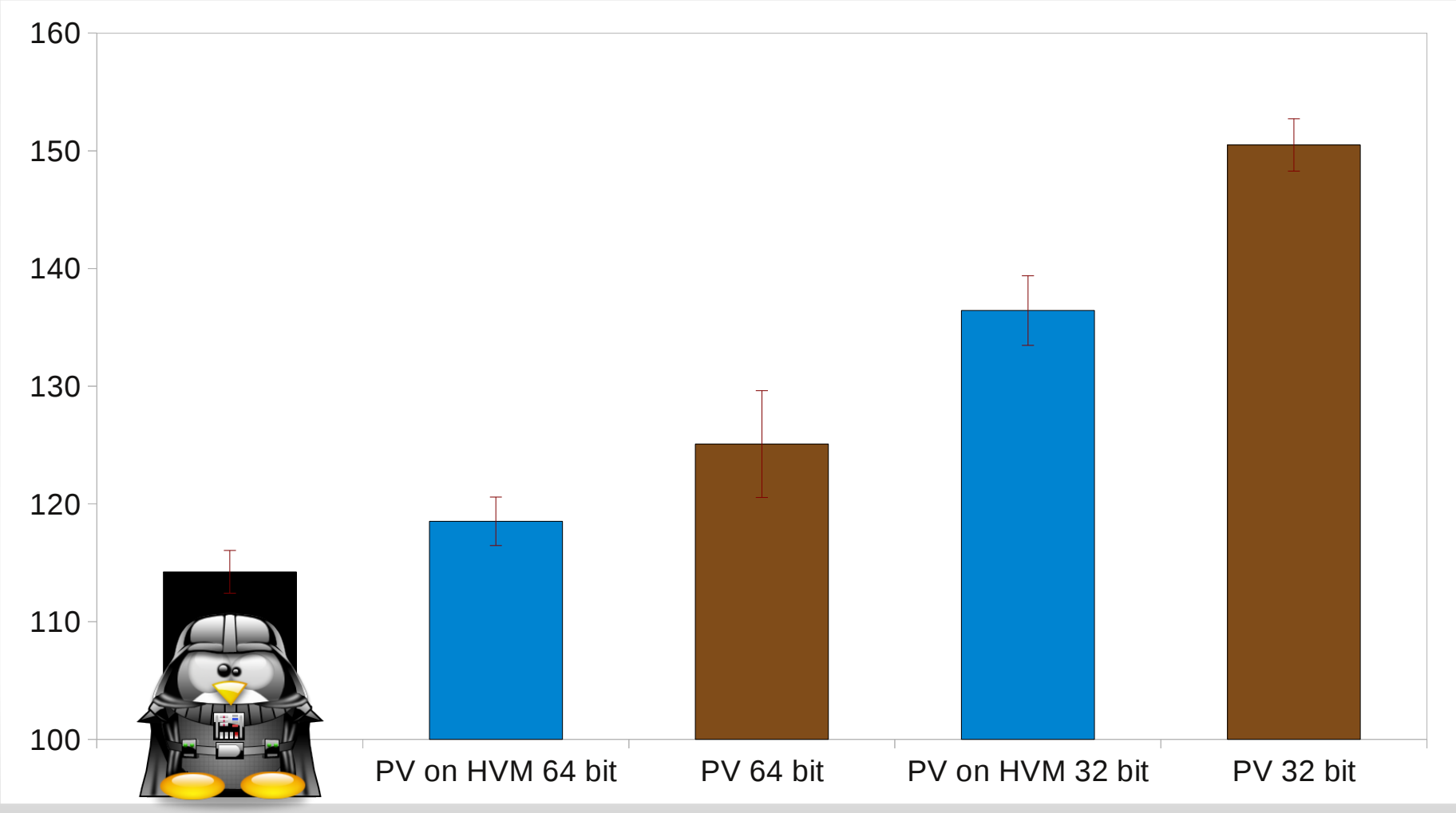
# PBZIP2

Results: percentage of native, the lower the better



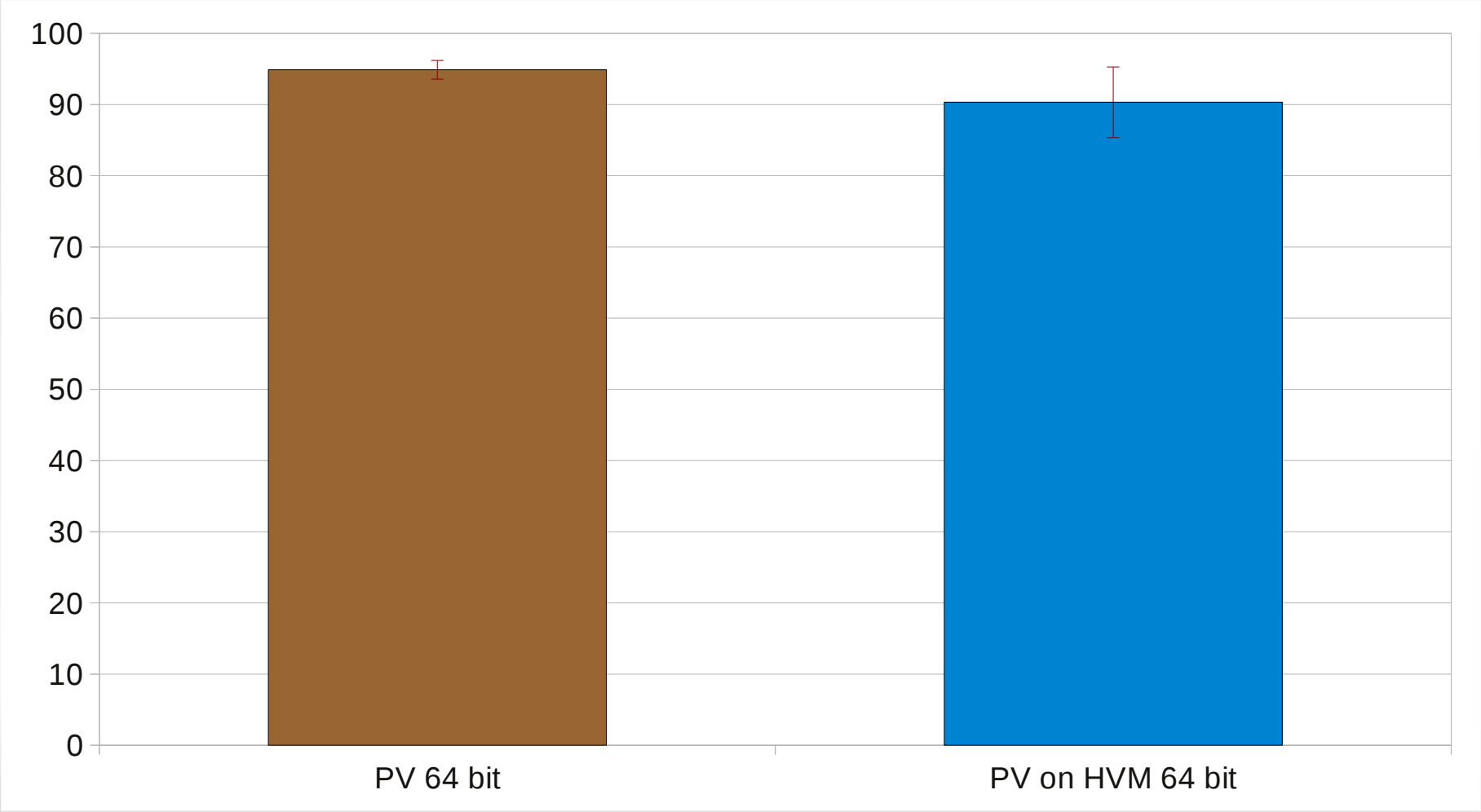
# PBZIP2

Results: percentage of native, the lower the better



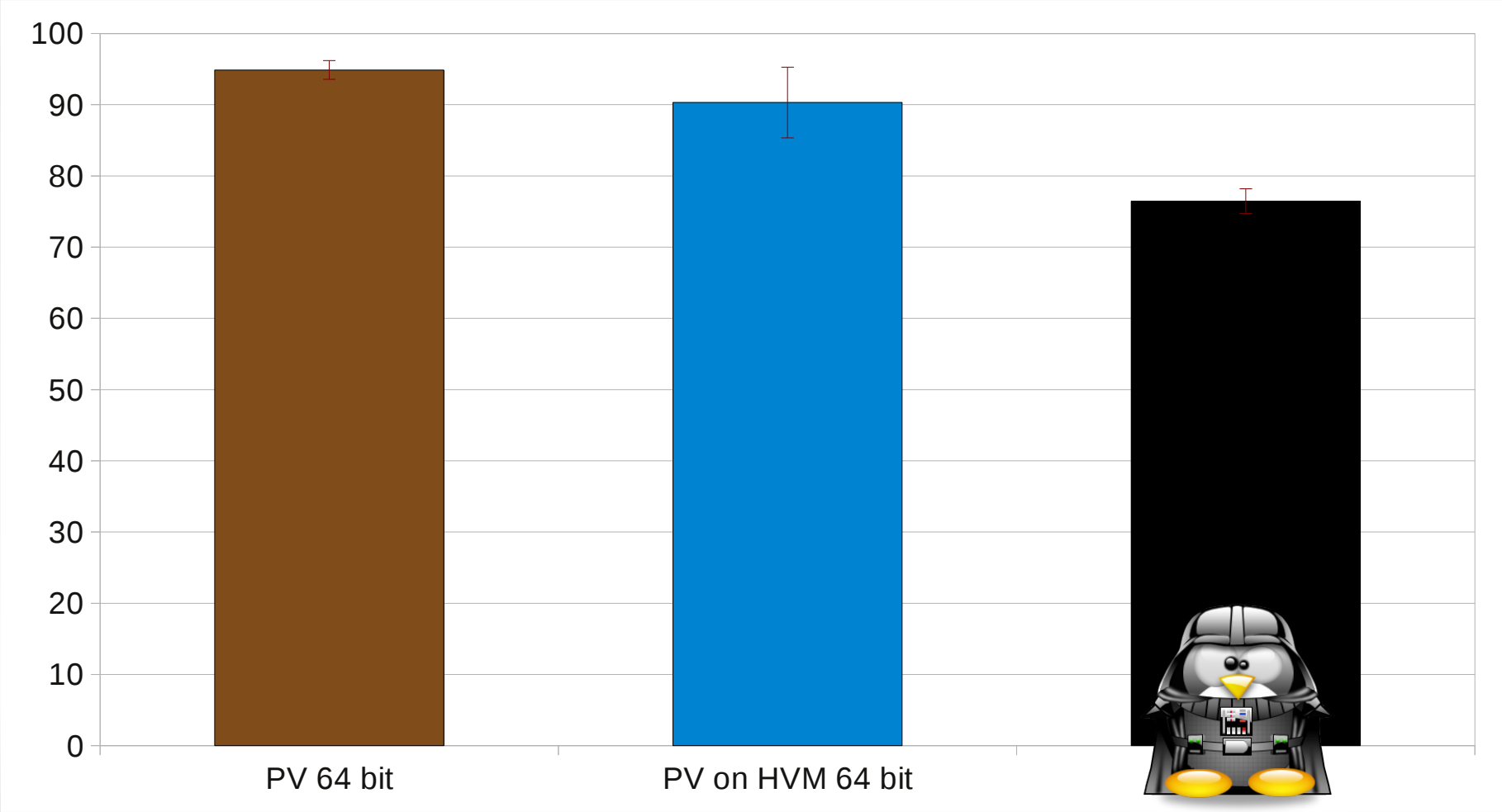
# SPECjbb2005

Results: percentage of native, the higher the better



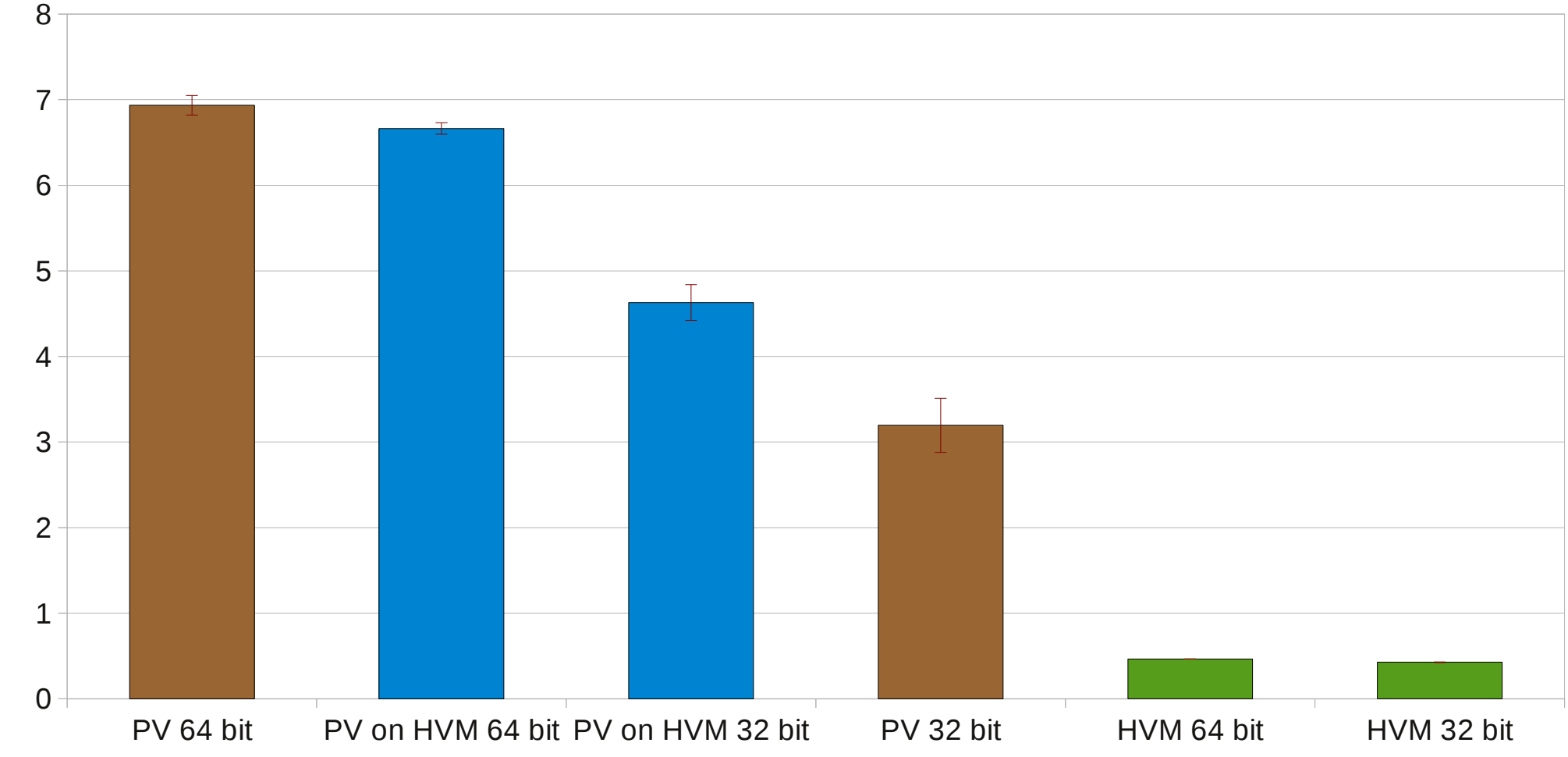
# SPECjbb2005

Results: percentage of native, the higher the better



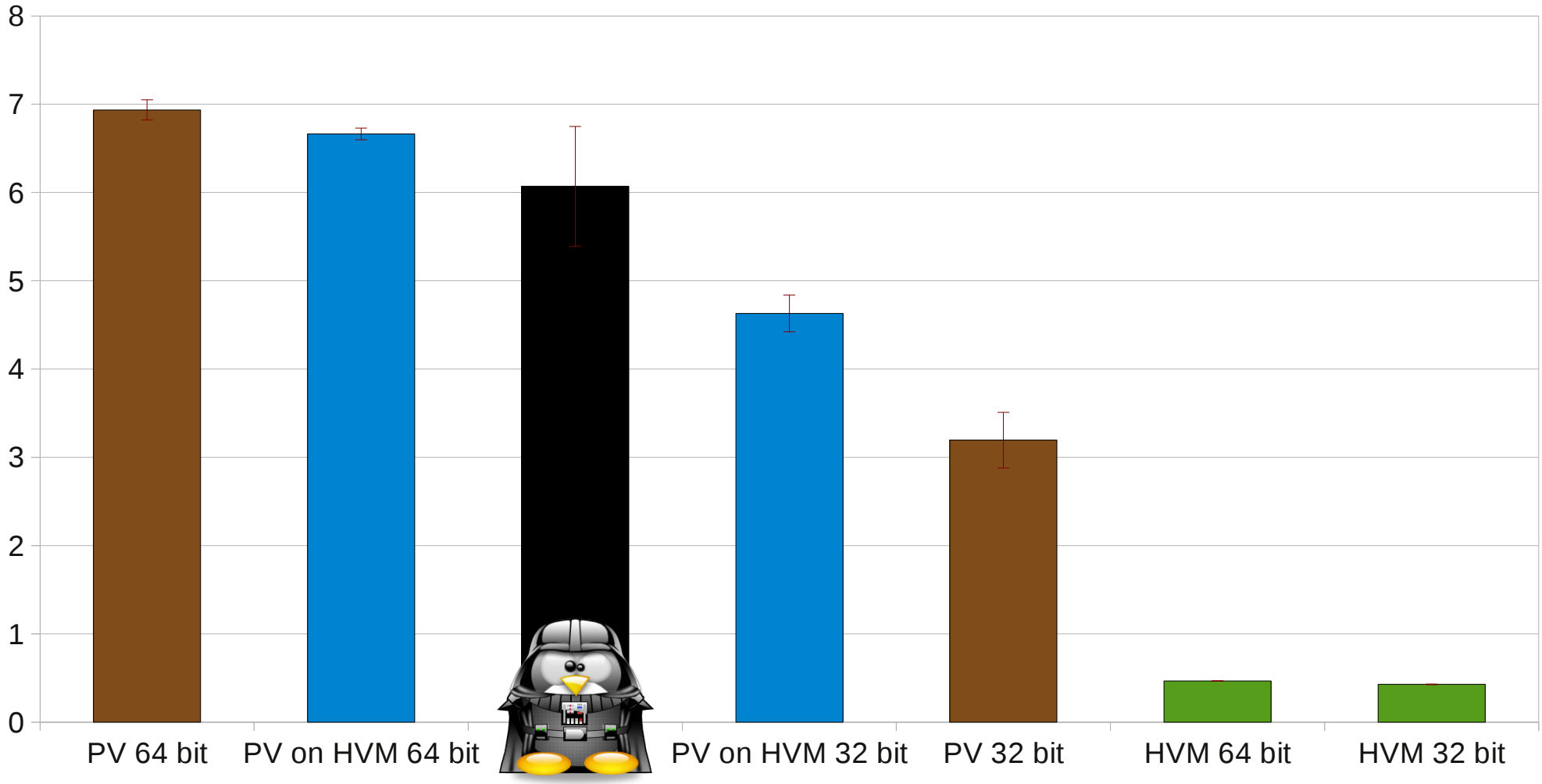
# Iperf tcp

Results: gbit/sec, the higher the better



# Iperf tcp

Results: gbit/sec, the higher the better



# Conclusions

PV on HVM guests are very close to PV guests in benchmarks that favor PV MMUs

PV on HVM guests are far ahead of PV guests in benchmarks that favor nested paging

# Questions?