Optimizing Network Virtualization in Xen

Jeff Shafer, Paul Willmann, Scott Rixner, Alan Cox
Rice University
Aravind Menon, Willy Zwaenepoel
EPFL

Virtual Machines

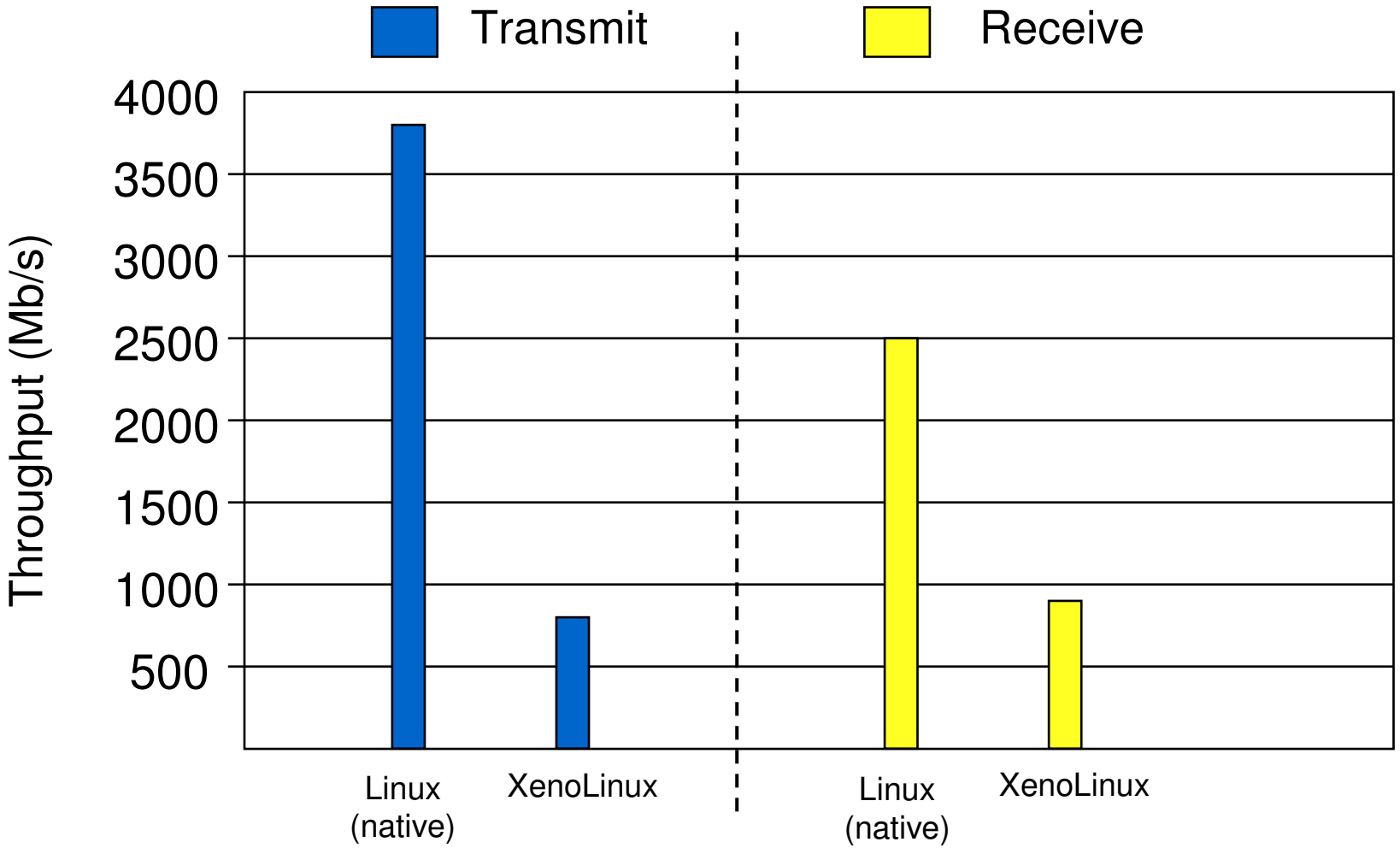
- Multiple VMs on a single machine
  - VMM/Hypervisor provides VM abstraction
  - Each VM runs its own “guest” OS
- Applications
  - Server consolidation
  - Migration
  - …
- Xen: open-source paravirtualization solution

Virtual Machine I/O

- Hardware devices shared between guests
- Software (de-)multiplexing
  - Data, interrupts
  - Xen: hypervisor, driver domain
- Overhead can be substantial
  - Important for networking on servers

Xen 2.0.6 Network Performance

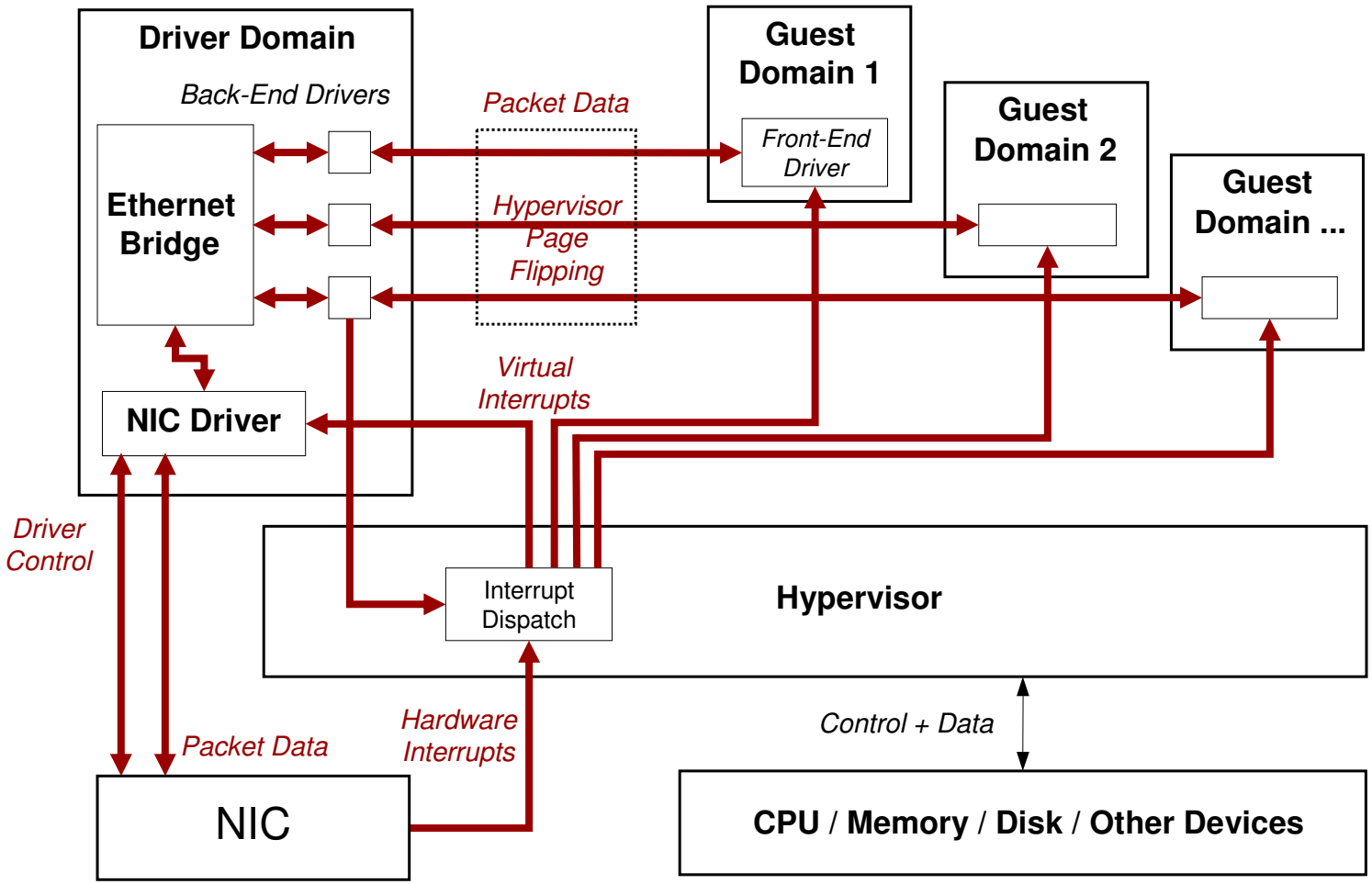
2.4GHz Xeon, Linux 2.6.11



Talk Outline

- Xen networking
- Software optimizations [Usenix 06]
- Hardware solution [HPCA 07]
  - Concurrent Direct Network Access (CDNA)

Networking in Xen



Xen Networking Overhead

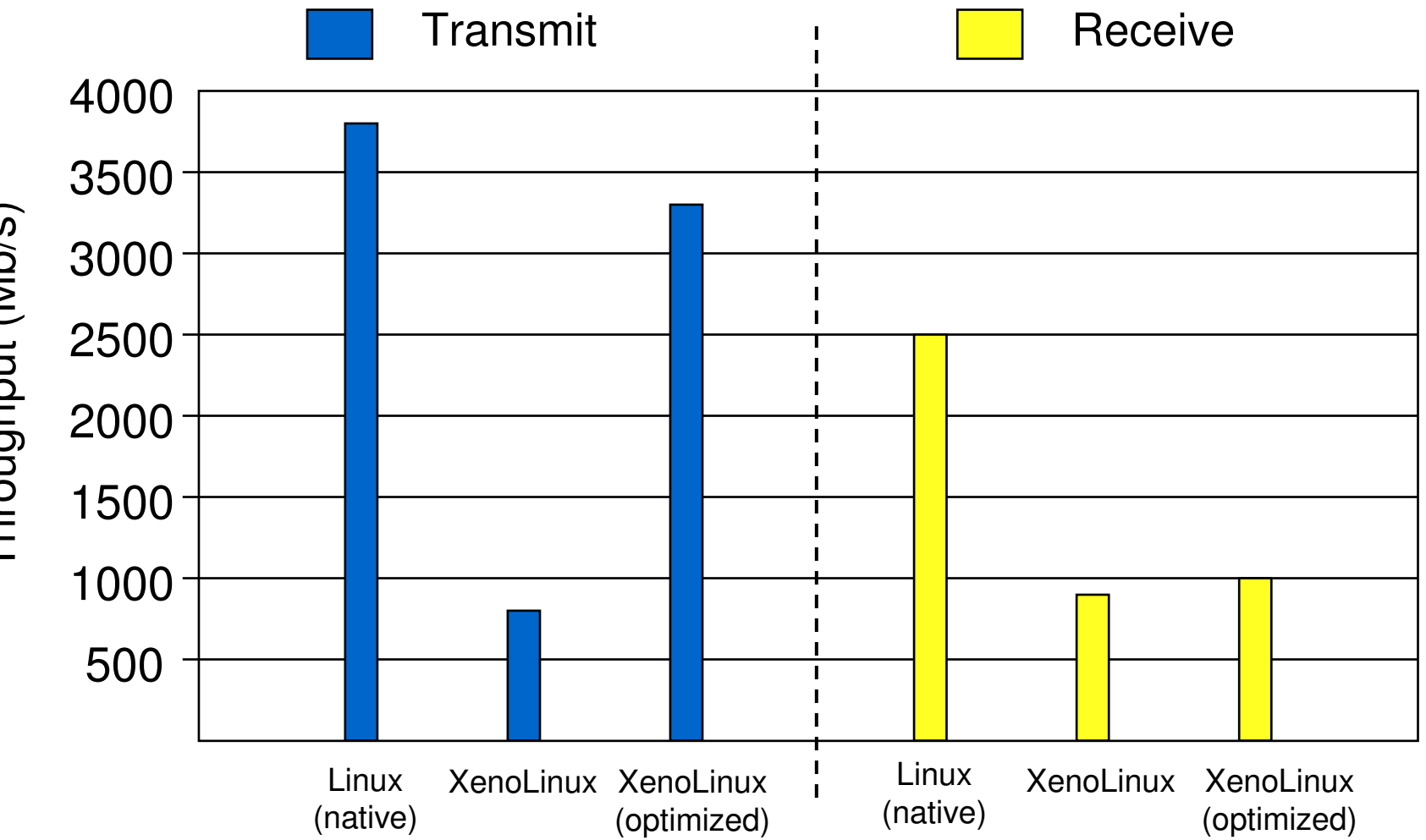
- Page remapping
- Context switching guest/driver
- Software bridge management
- Interrupt handling

Software Optimizations

- High-level virtual NIC
- I/O channel optimizations
- Virtual memory optimizations

- Basic Xen I/O architecture unchanged
  - Driver domain

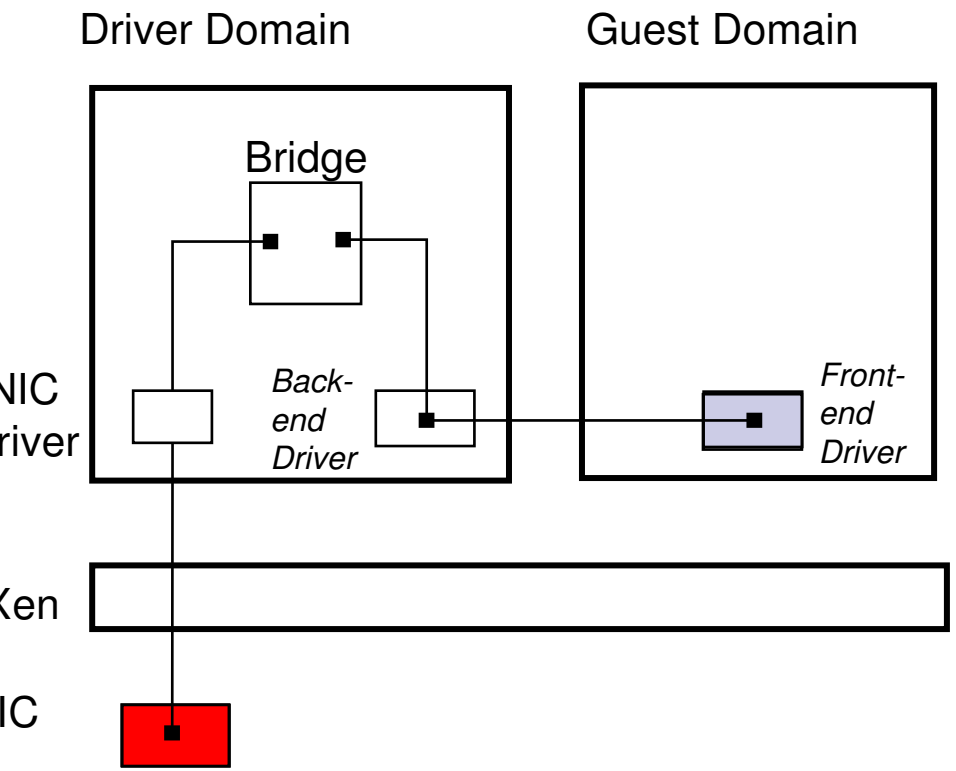
Preview: Overall Results



Our Optimizations

- Smart virtual NIC
- I/O channel optimizations
- Virtual Memory optimizations

Xen: Dumb Virtual NIC

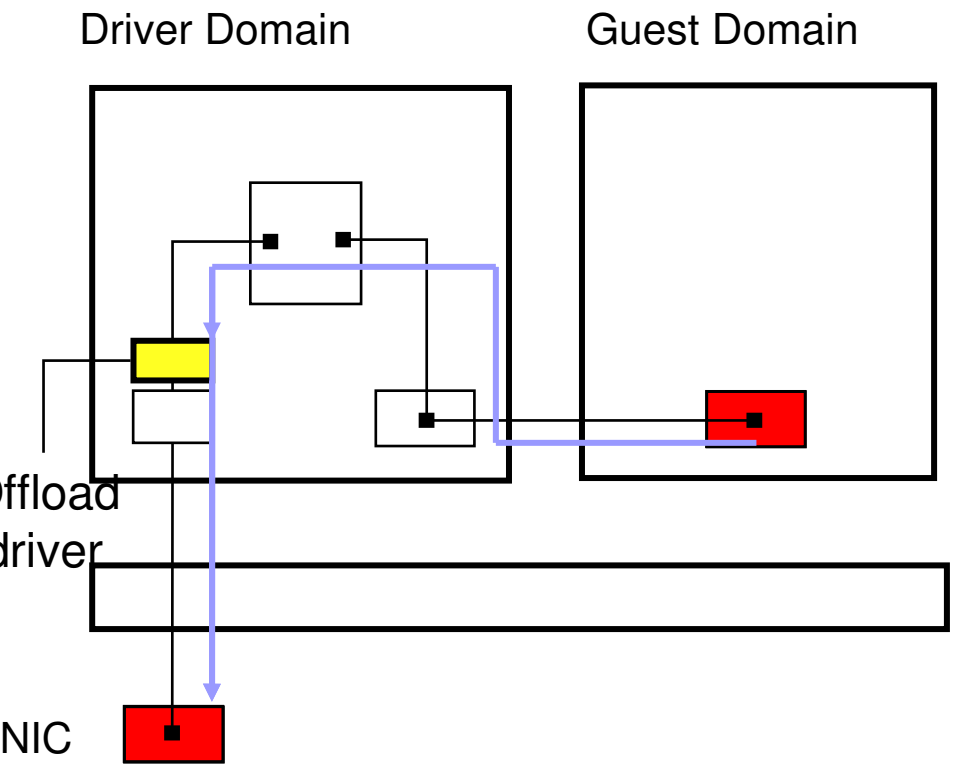


- Physical NIC
  - TSO
  - Scatter/gather I/O
  - Checksum offload
- Virtual NIC
  - None of the above
  - Map to any NIC

Smart Virtual NIC

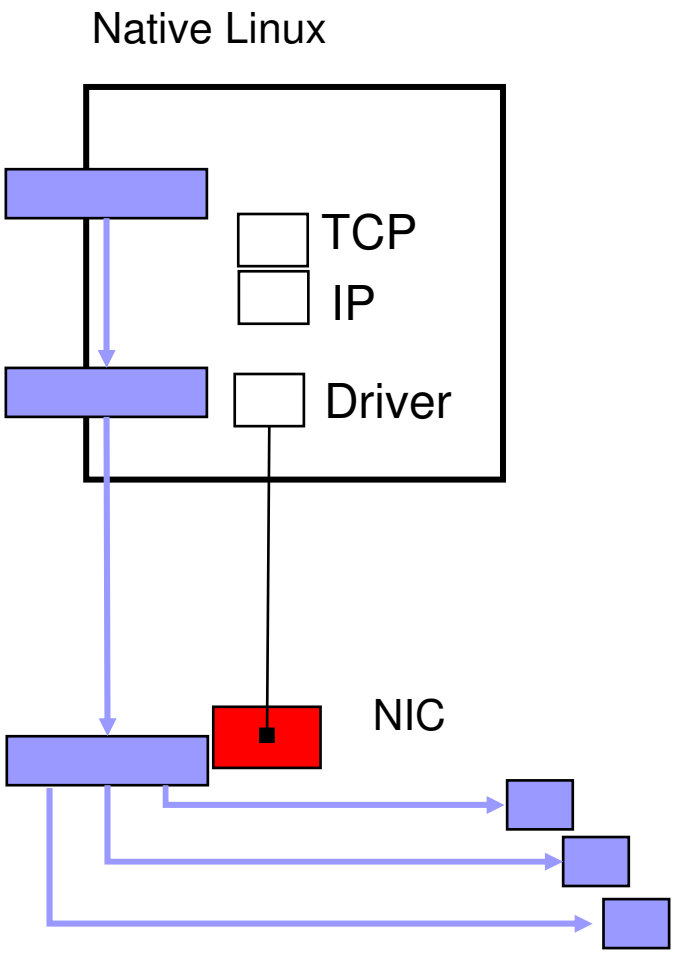
- Support offload in virtual NIC
- Independent of physical NIC
  - Emulate offload in driver domain if required

Smart Virtual NIC: Architecture



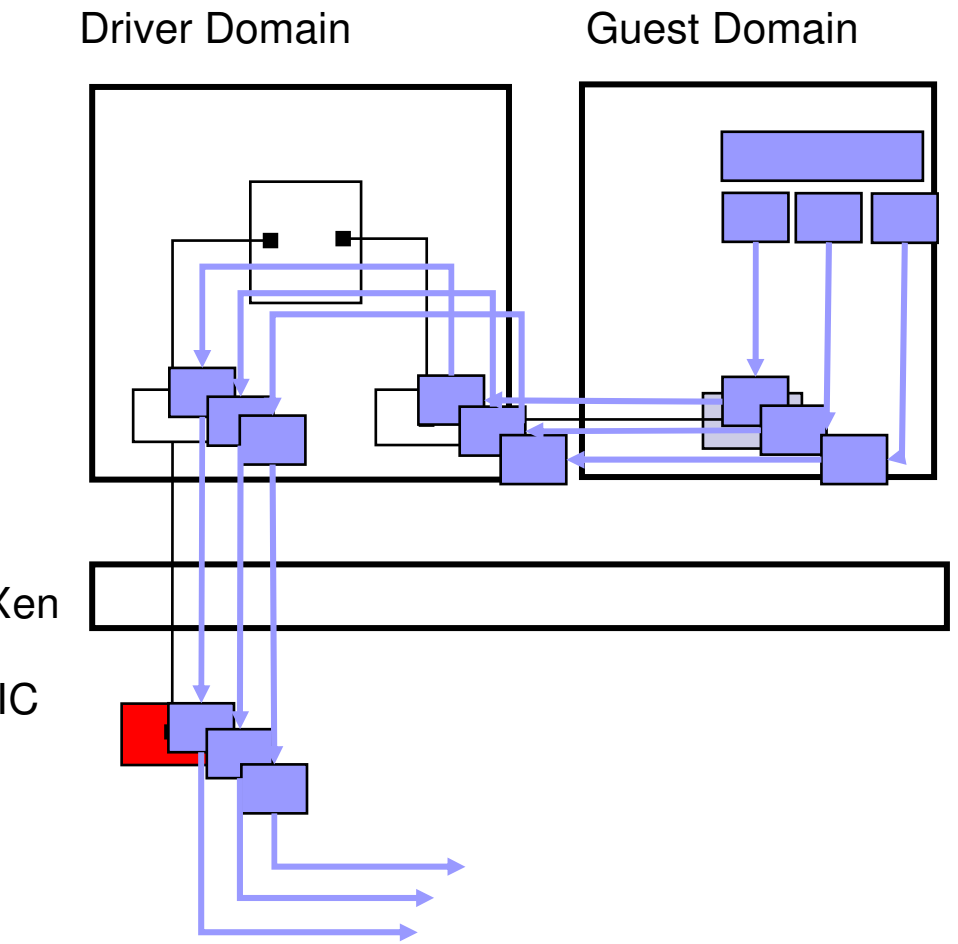
- Offload in virtual NIC
- Offload driver in driver domain

TCP Segmentation Offload (TSO)



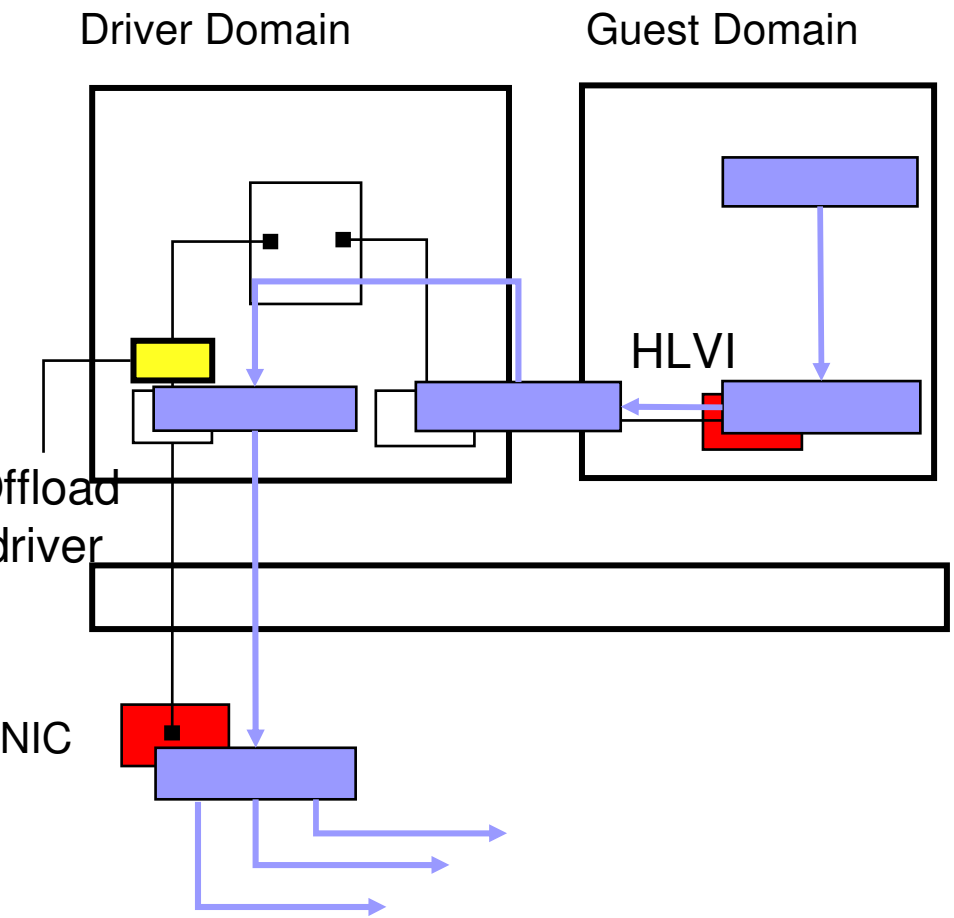
- Large effective MTU
- Fewer “packets”
- Lower per-byte cost

Xen Virtualization Overheads (TSO)



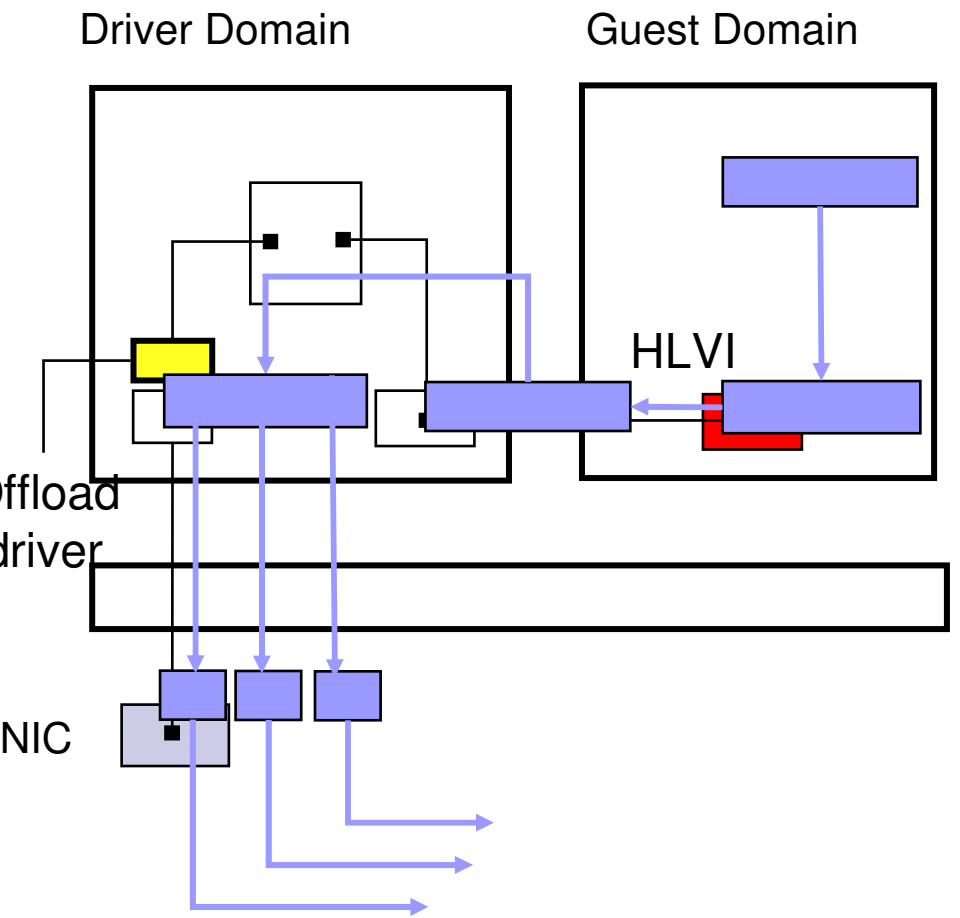
- No TSO support
- More “packets”
- Higher per-byte cost

Smart Virtual/Physical NIC



- TSO support
- Fewer “packets”
- Per-byte cost much reduced

Smart Virtual/Dumb Physical NIC

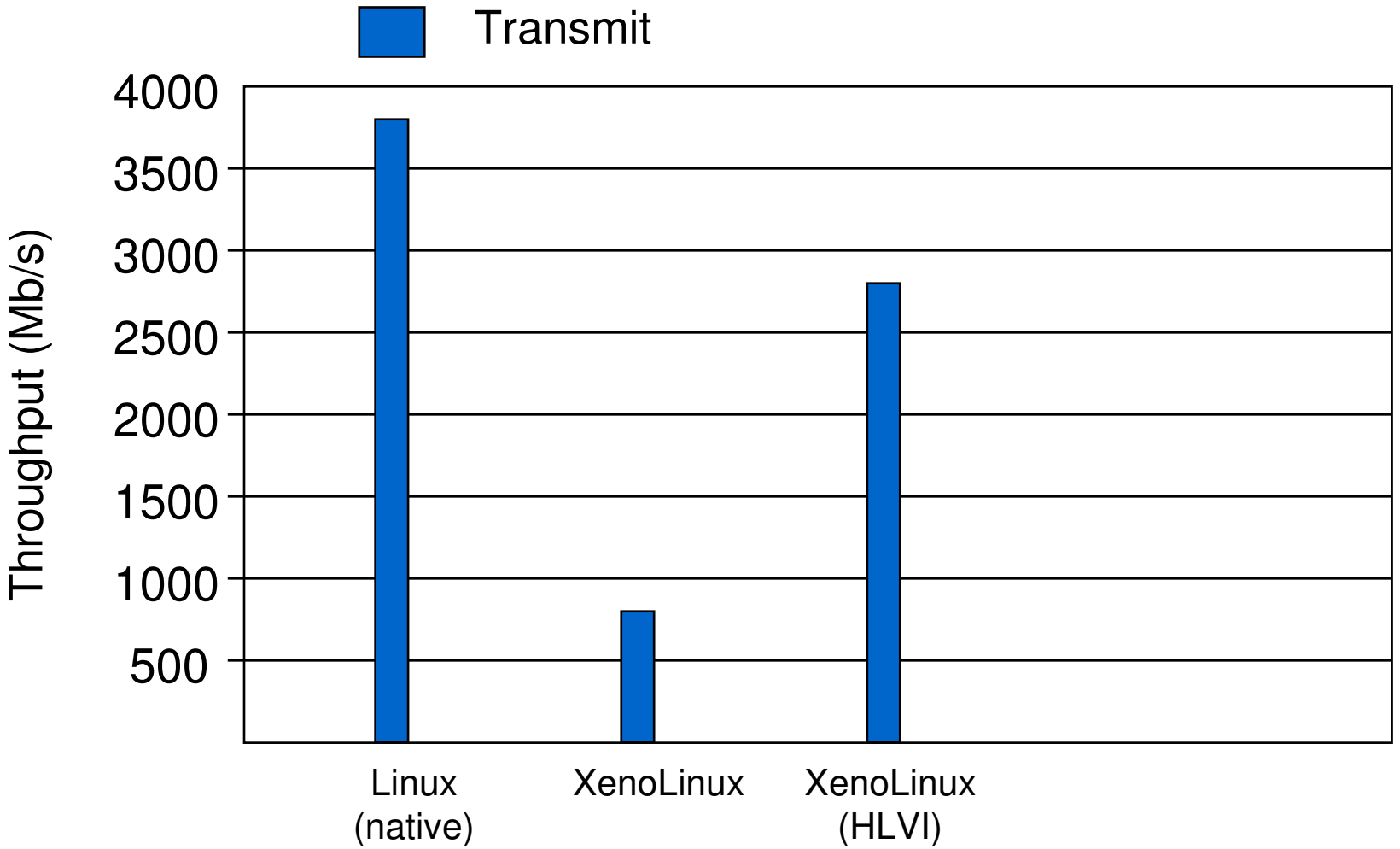


- Fewer “packets” above offload driver
- I/O virtualization cost still reduced

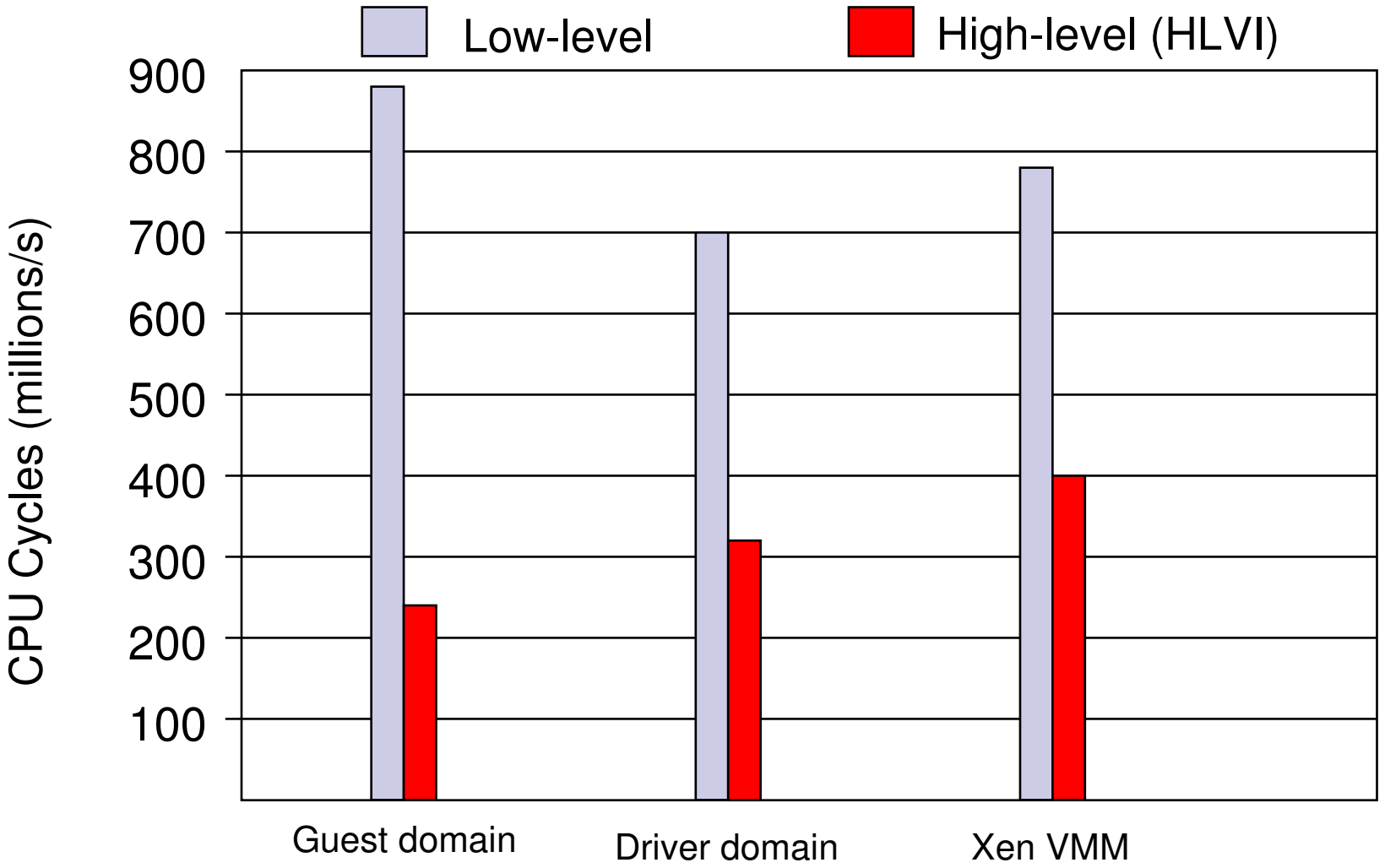
Evaluation

- 2.4 GHz Xeon
- 4 Intel Pro-1000 NICs
  - Support TSO, S/G, Checksum
- Xen 2.0.6, Linux 2.6.11
- Transmit workloads
  - Zero-copy sendfile

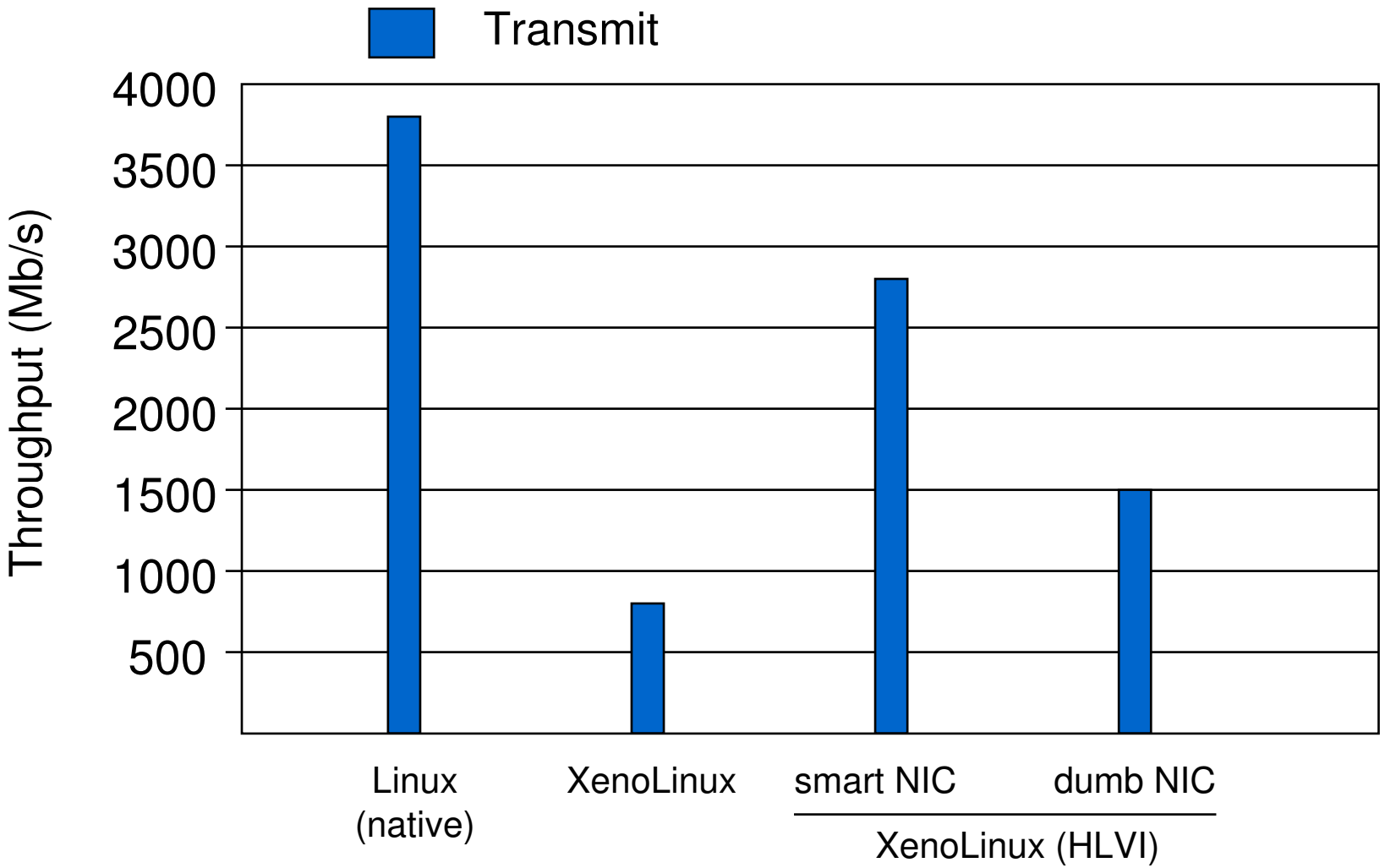
Evaluation: Sendfile



CPU Overhead



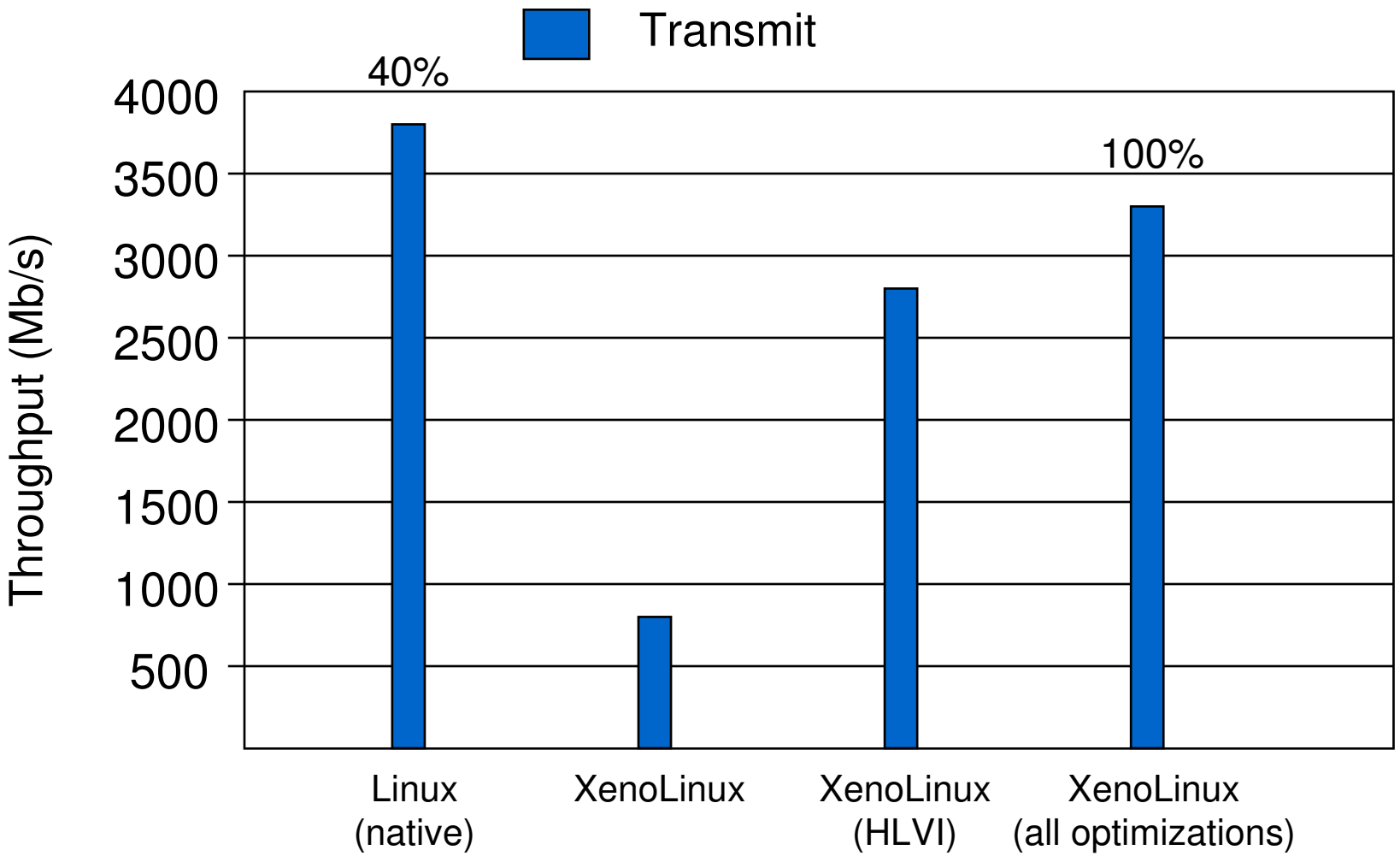
Evaluation: Offload Driver



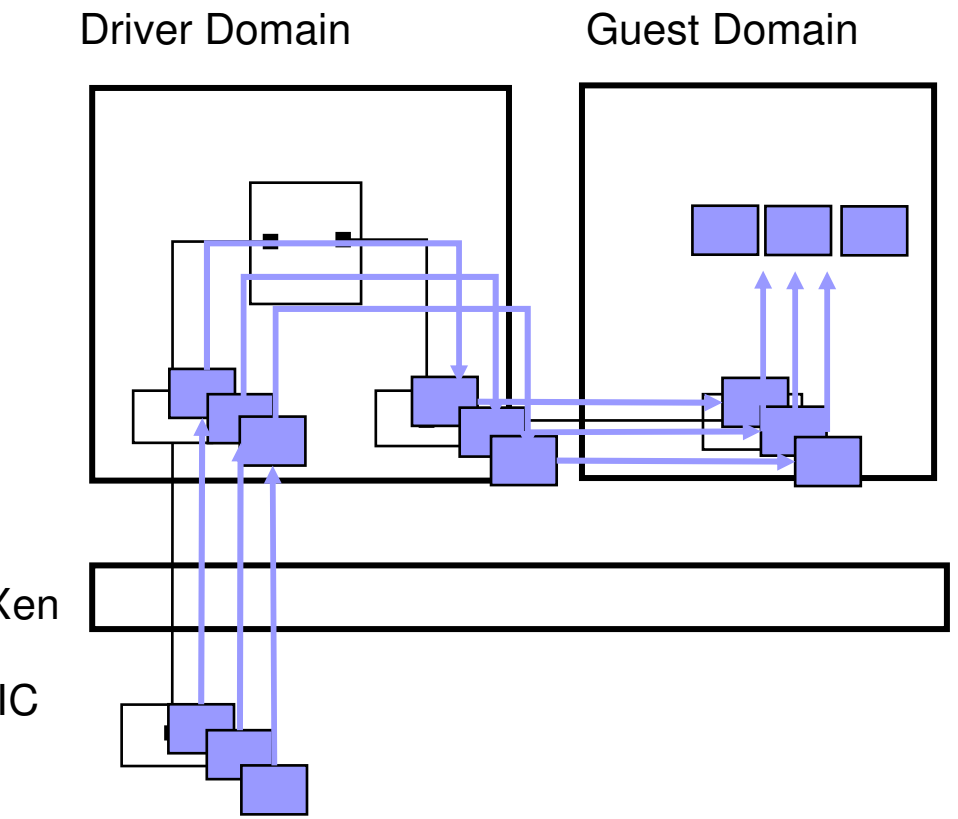
Other Optimizations

- I/O Channel optimizations
  - Avoid mapping on transmit path
  - Copy instead of remapping on receive path
- Virtual memory optimizations
  - Superpages
  - Global page mappings

Overall Results

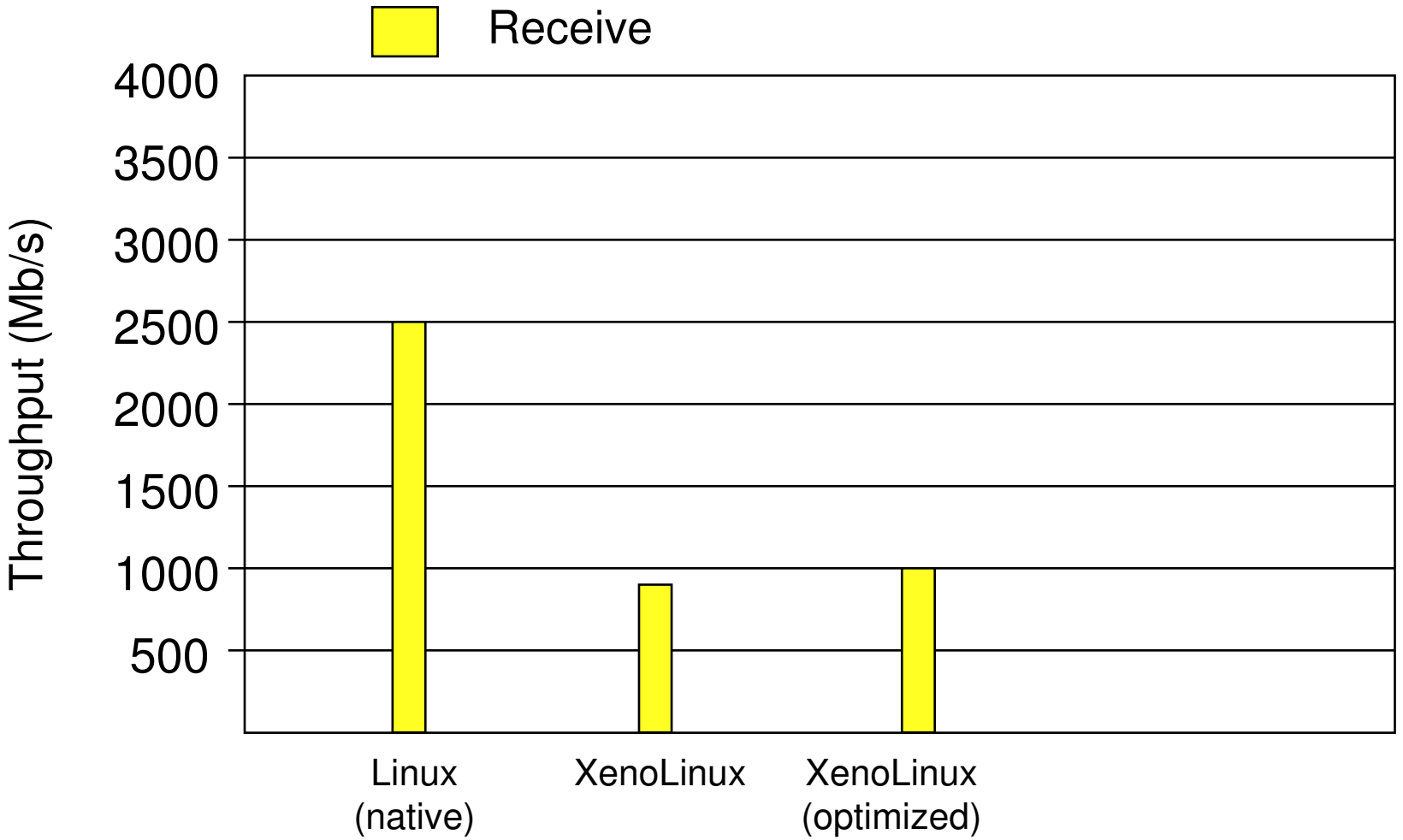


Receive Processing



- Packets handled individually
  - Bridge
  - Drivers
- High per-packet cost
- No obvious receive-side optimizations

Receive Performance



Software Optimizations

- Support a smart virtual NICs
- Reduce Xen virtualization overheads
  - “Smart” device better than “dumb” device
  - Transmit performance improvements of 4x
- Other optimizations yield further benefits

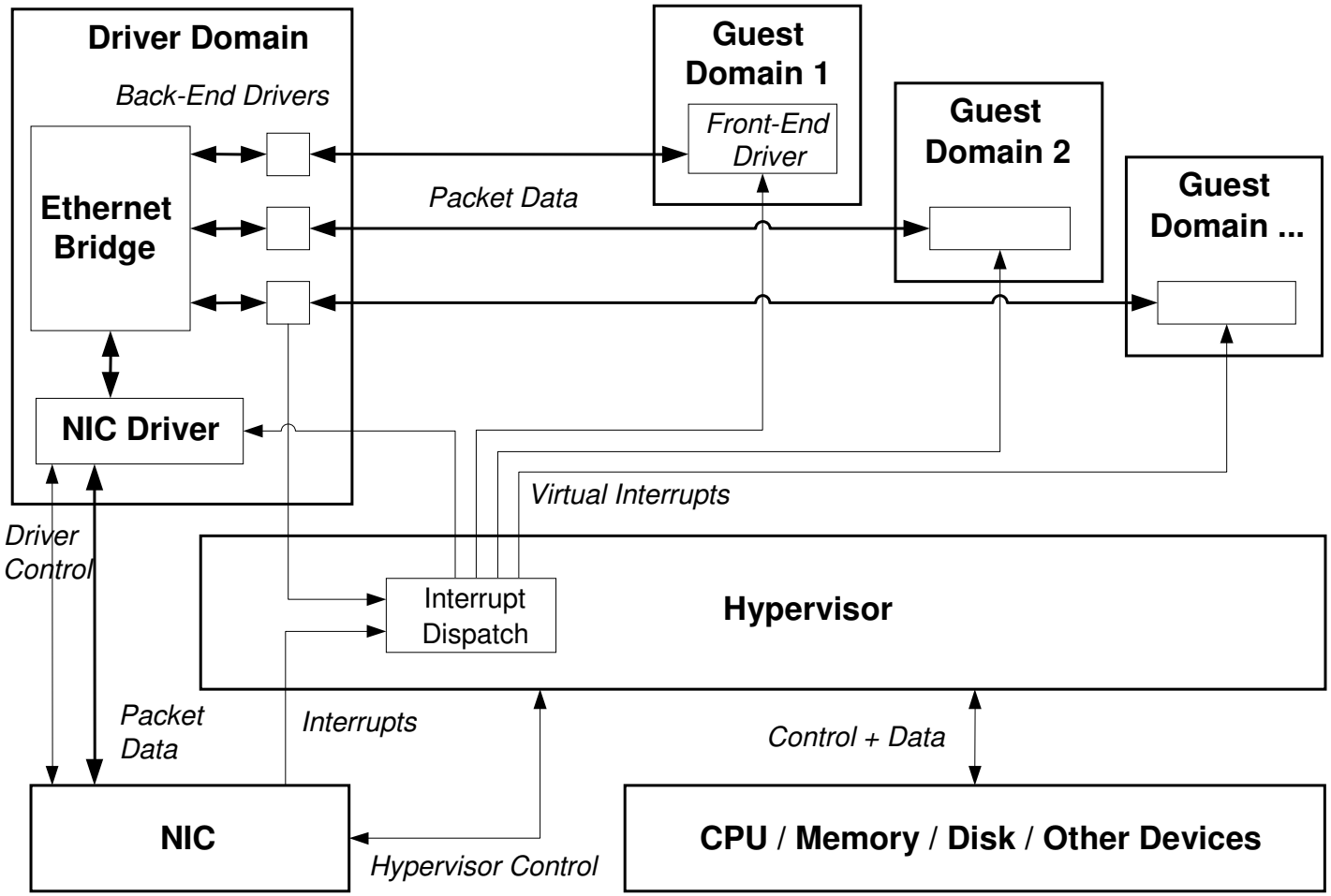
Limitations

- Receive performance limited
- Transmit performance scales poorly
  - Performance drops with increasing no of VMs
- Can new hardware give further benefits?

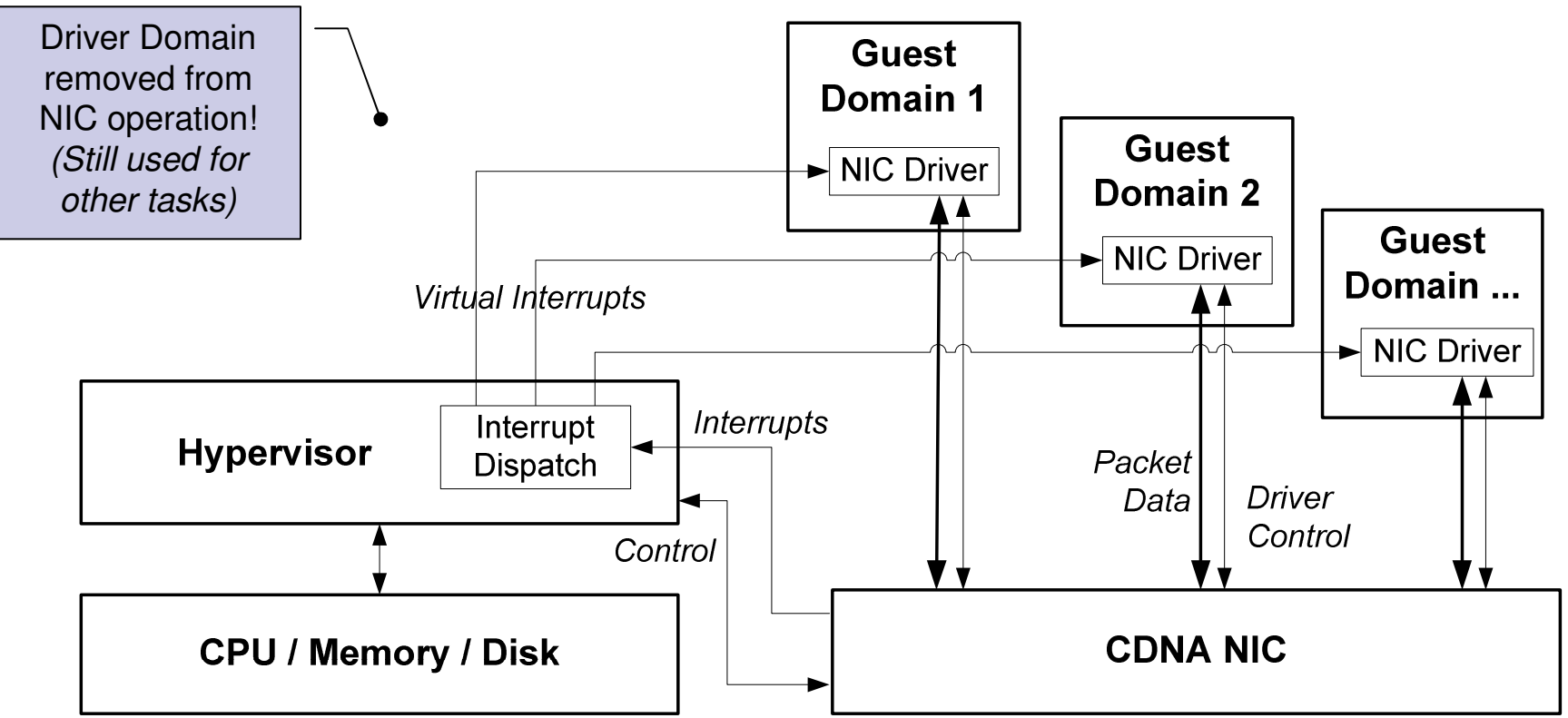
Concurrent Direct Network Access

- CDNA NIC exports up to 128 *contexts*
- Context ~= individual interface
- Each guest connects to a context
- CDNA NIC multiplexes network traffic
- *Not* the hypervisor or the driver domain

Networking in Xen (again)



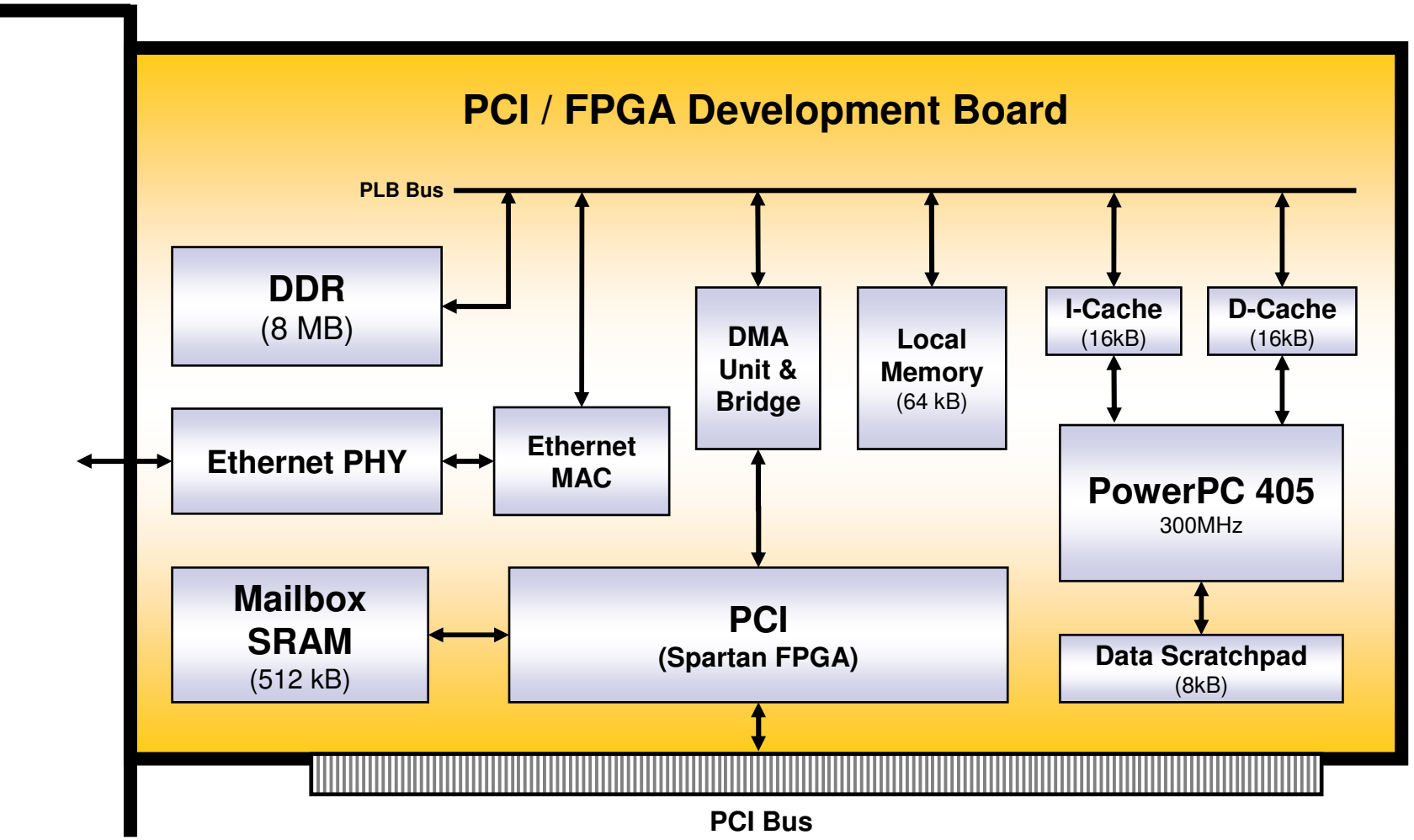
Xen + CDNA Networking



Performance Improvement

- Remove driver domain from
  - Data
  - Interrupts
- As before, remove hypervisor from
  - Data
- Hypervisor only responsible for
  - Virtual interrupts
  - Assigning context to guest OS

CDNA NIC Architecture



Host/NIC Communication

- Programmed I/O from host to NIC
  - “Mailboxes” in CDNA NIC
  - Typical use: Transfer control data and buffer indices
- DMA by NIC
  - Bidirectional
  - Typical use: Bulk data transfer
- Interrupts from NIC to Host
  - Typical use: Notification of sent/received packet(s)

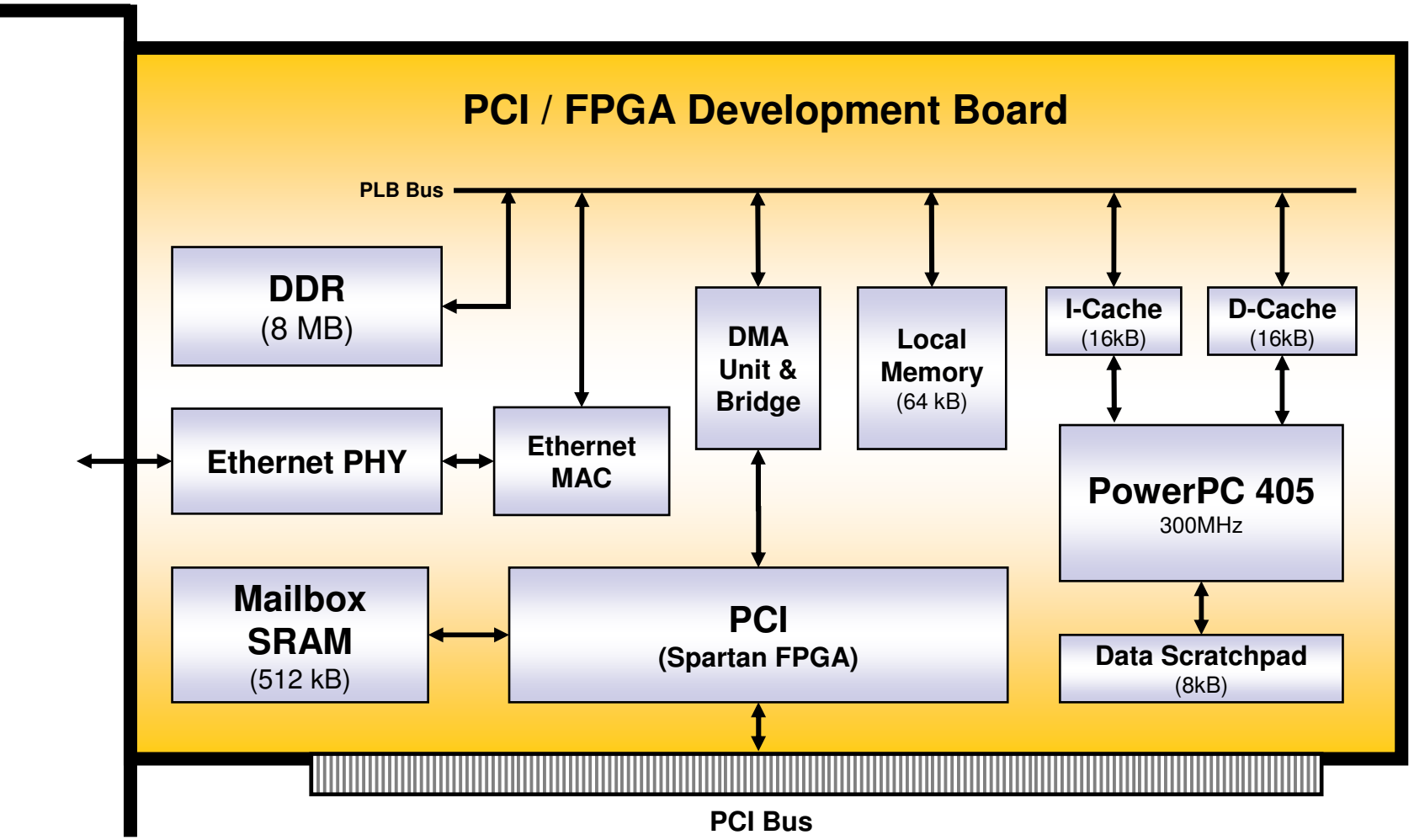
Protection Issues

- Access to mailboxes
- Deviating interrupts
- DMA into other guest

Protection (1)

- Mailboxes
  - Page per context mapped to one VM

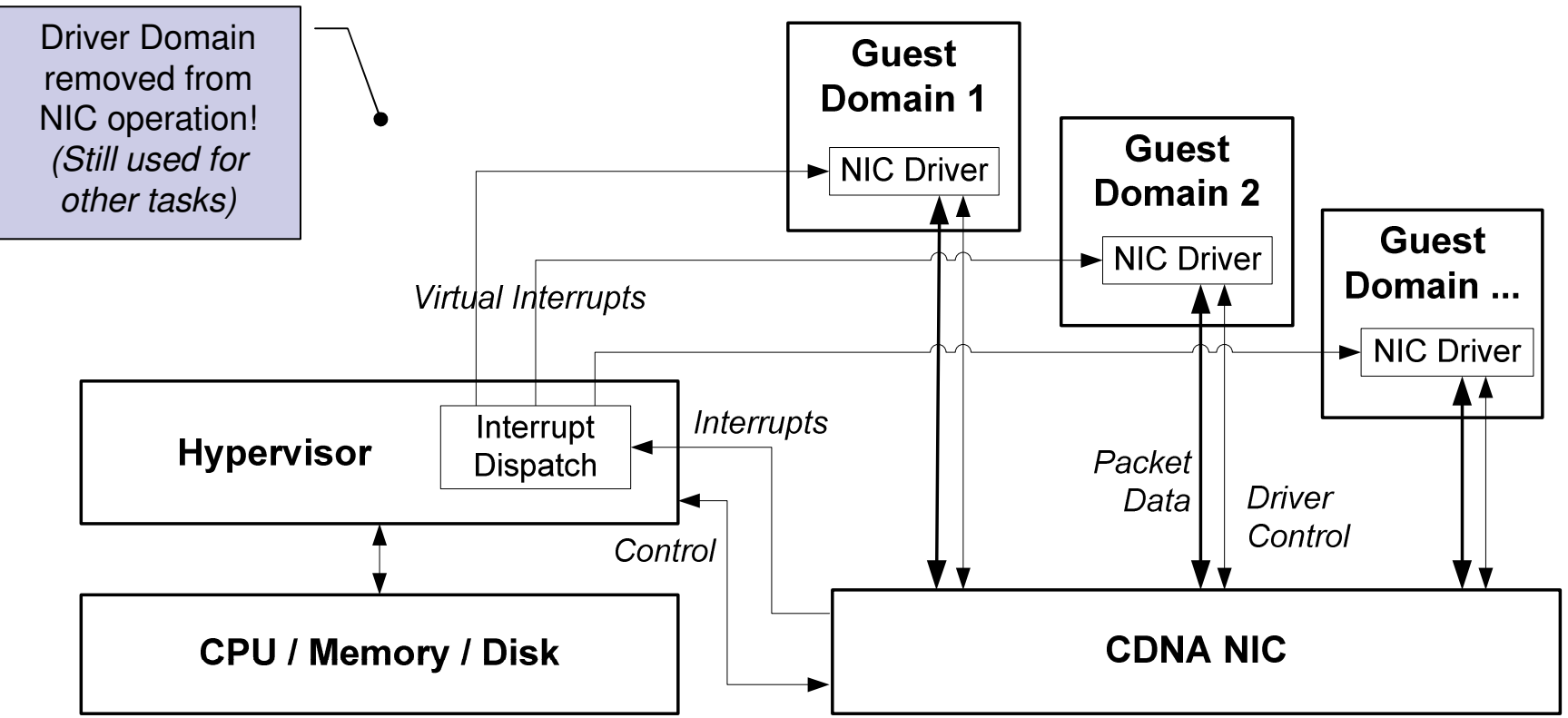
CDNA NIC Architecture



Protection (2)

- Mailboxes
  - Page per context mapped to one VM
- Interrupts
  - Mediated by hypervisor

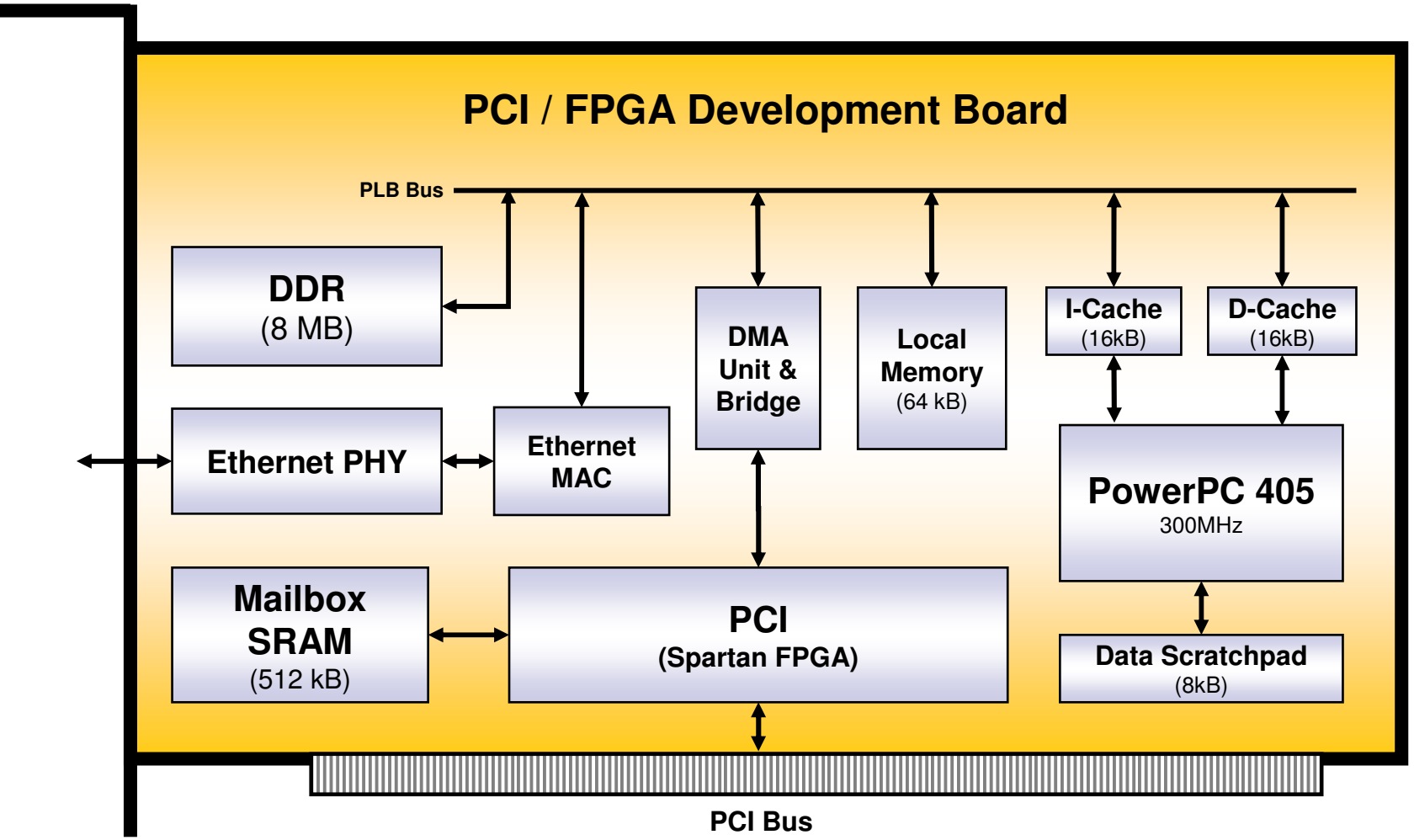
Xen + CDNA Networking



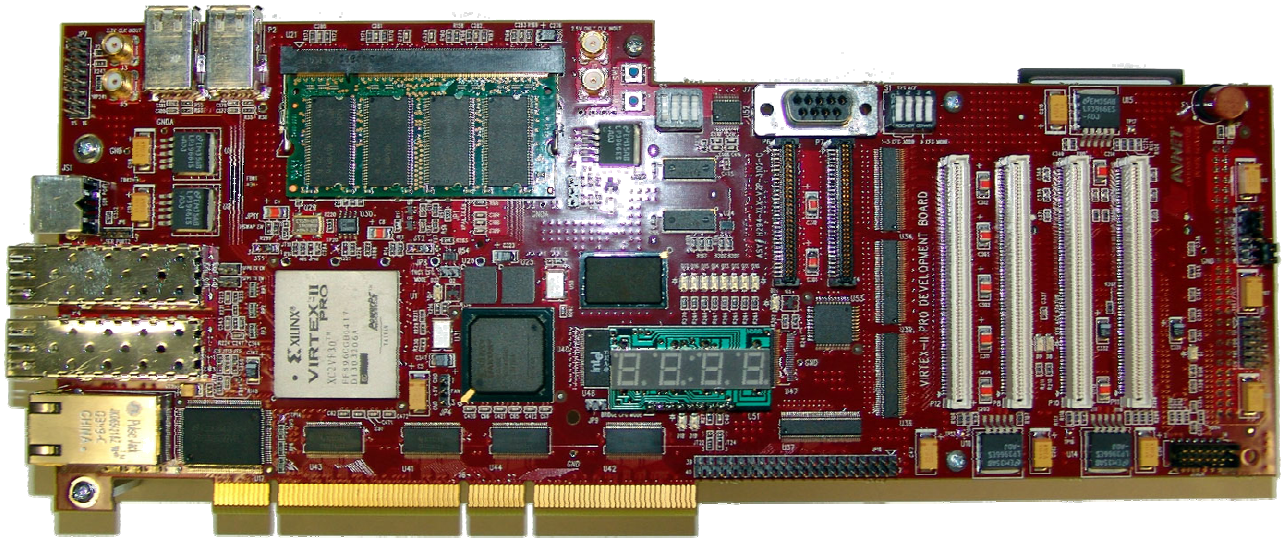
Protection (3)

- Mailboxes
  - Page per context mapped to one VM
- Interrupts
  - Mediated by hypervisor
- DMA
  - Buffers validated by hypervisor
  - No buffer deallocation before DMA complete

CDNA NIC Architecture



CDNA NIC Prototype

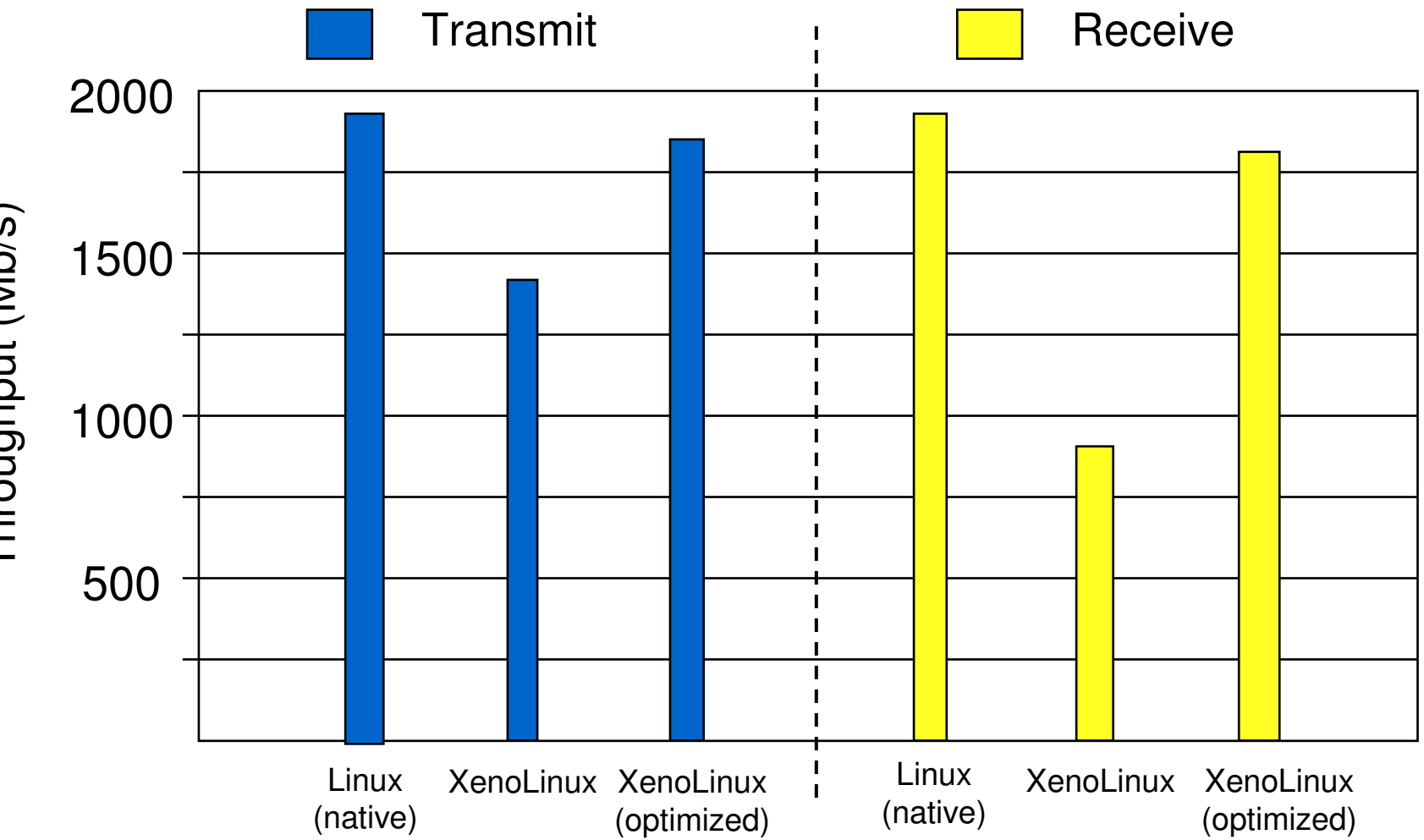


- FPGA Development Board
- Virtex and Spartan FPGAs
- 2 PowerPC 405 processors
  - Only 1 used for CDNA
- DDR and SRAM Memory
- Gigabit Ethernet PHY
- 64-bit / 66 MHz PCI bus

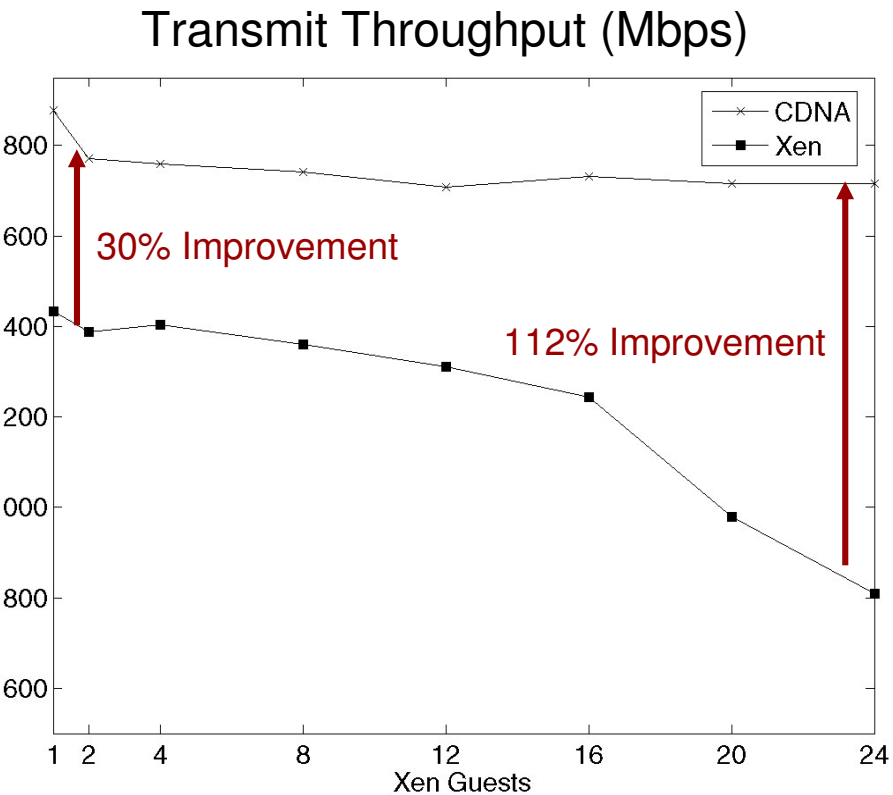
CDNA Evaluation

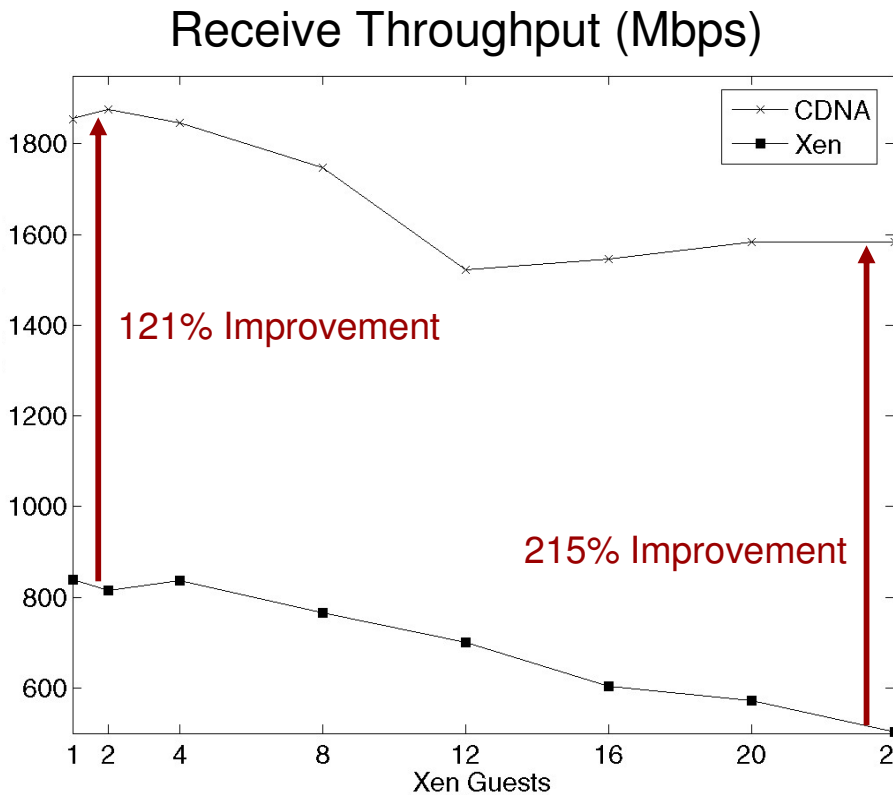
- Opteron 250 (2.4 GHz)
- 2 Broadcom NICs
- 2 CDNA NICs
- Xen 2.0.6, Linux 2.6.11
- Streaming transmit/receive workloads

Evaluation: Throughput



Evaluation: Scalability





Hardware Optimizations

- Remove driver domain from Xen networking
  - Improved throughput and scalability
- Minimal cost to have NIC multiplex data
  - 512kB SRAM for PIO mailboxes (128 guests)
  - 8MB of DDR memory for data buffers
  - Custom firmware to service all contexts

Conclusions

- Xen's low-level virtual NIC limits performance
  - Software optimizations can remove this bottleneck
  - For transmit traffic of a few guests
- Driver domain overhead limits scalability
  - CDNA removes driver domain from networking
  - Improved transmit and receive performance
  - Better scalability