



Netchannel 2: Optimizing Network Performance

J. Renato Santos⁺, G. (John) Janakiraman⁺
Yoshio Turner⁺, Ian Pratt^{*}

⁺HP Labs - ^{*}XenSource/Citrix



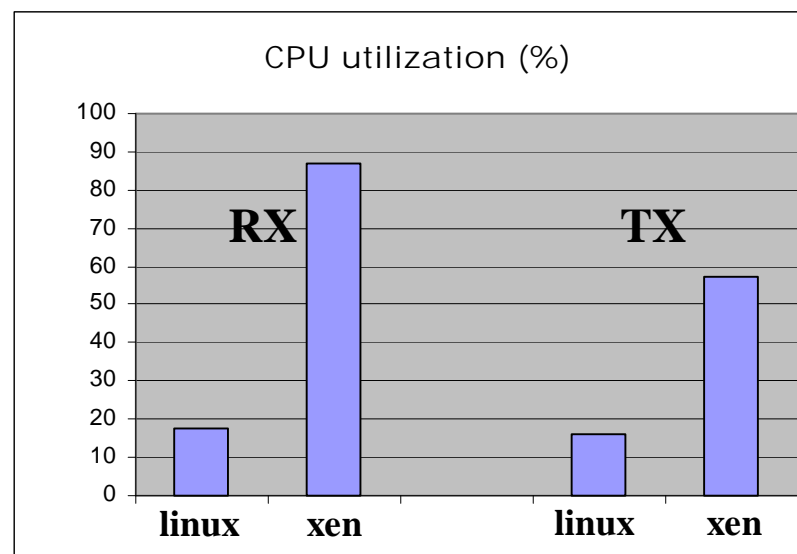
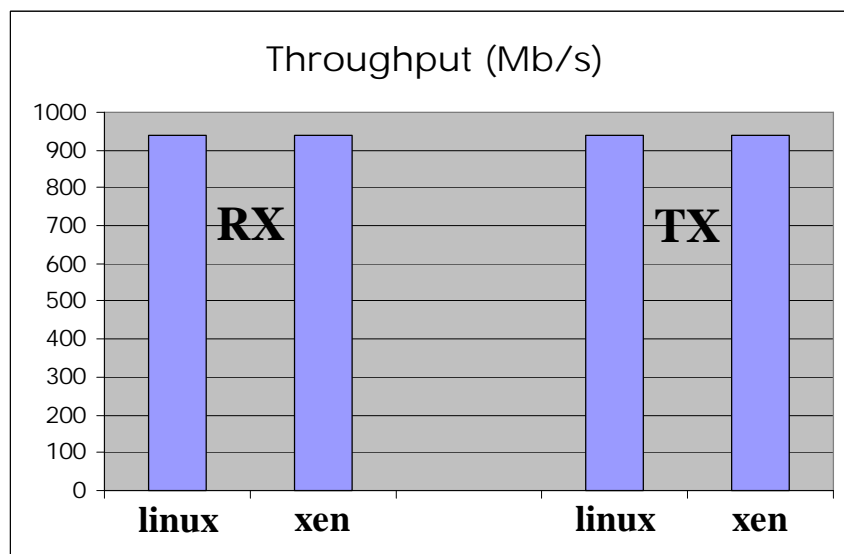
Xen Summit

Nov 14-16, 2007

Motivation



TCP performance for GigE today (PV drivers)



- Xen PV driver can sustain peak throughput on GigE
- But Xen uses significantly more CPU cycles than Linux
 - Less available cycles for application.
 - 10 Gig networks: CPU saturation prevents achieving line rate
- Need to reduce I/O virtualization overhead in Xen networking

Netchannel2: New I/O channel protocol to enable Xen networking design changes for improving performance

- Work in progress

- Software optimizations
 - Implementation optimizations
 - Software design changes

- Devices with direct guest access (Direct I/O, PCI-IOV)
(Device exposes multiple virtual interfaces accessed directly by guest)

- Multi-queue devices
 - Device has multiple RX queues
 - Avoids data copy on receive

Performance Analysis of Xen networking

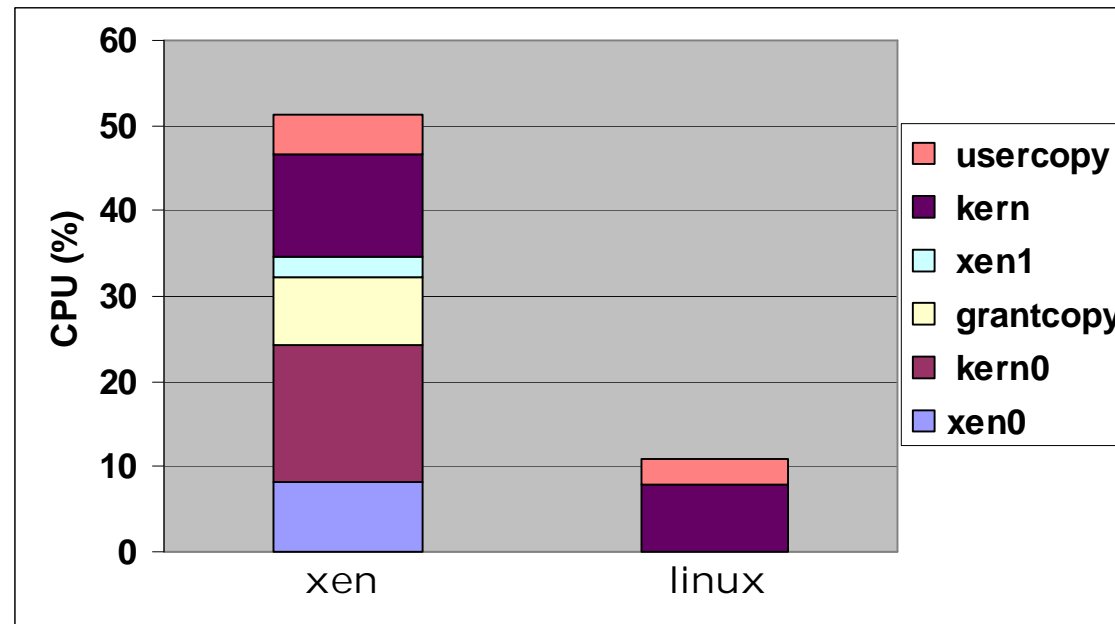


- Identified main sources of overhead
- Results guided design choices in Netchannel 2
- Emphasis on RX results
 - higher overhead than TX

Performance Analysis of Xen Networking (RX)



RX, UDP traffic, large msg (48KB) (Xen 64bit on Intel core2 duo)



usercopy: data copy from guest kernel to user buffer

kern: kernel code in guest

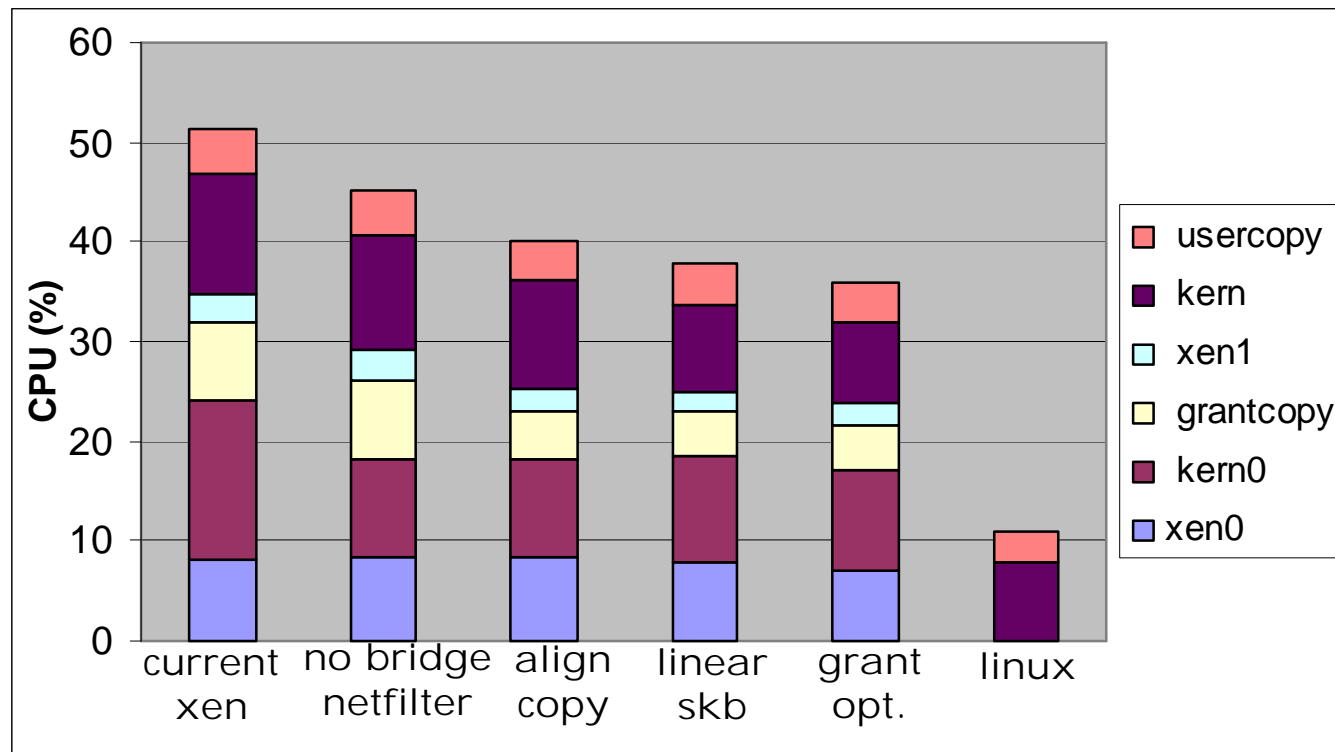
xen1: xen code executed in the context of the guest

grantcopy: data copy using Xen grant (only memory copy cost)

kern0: kernel code in domain 0

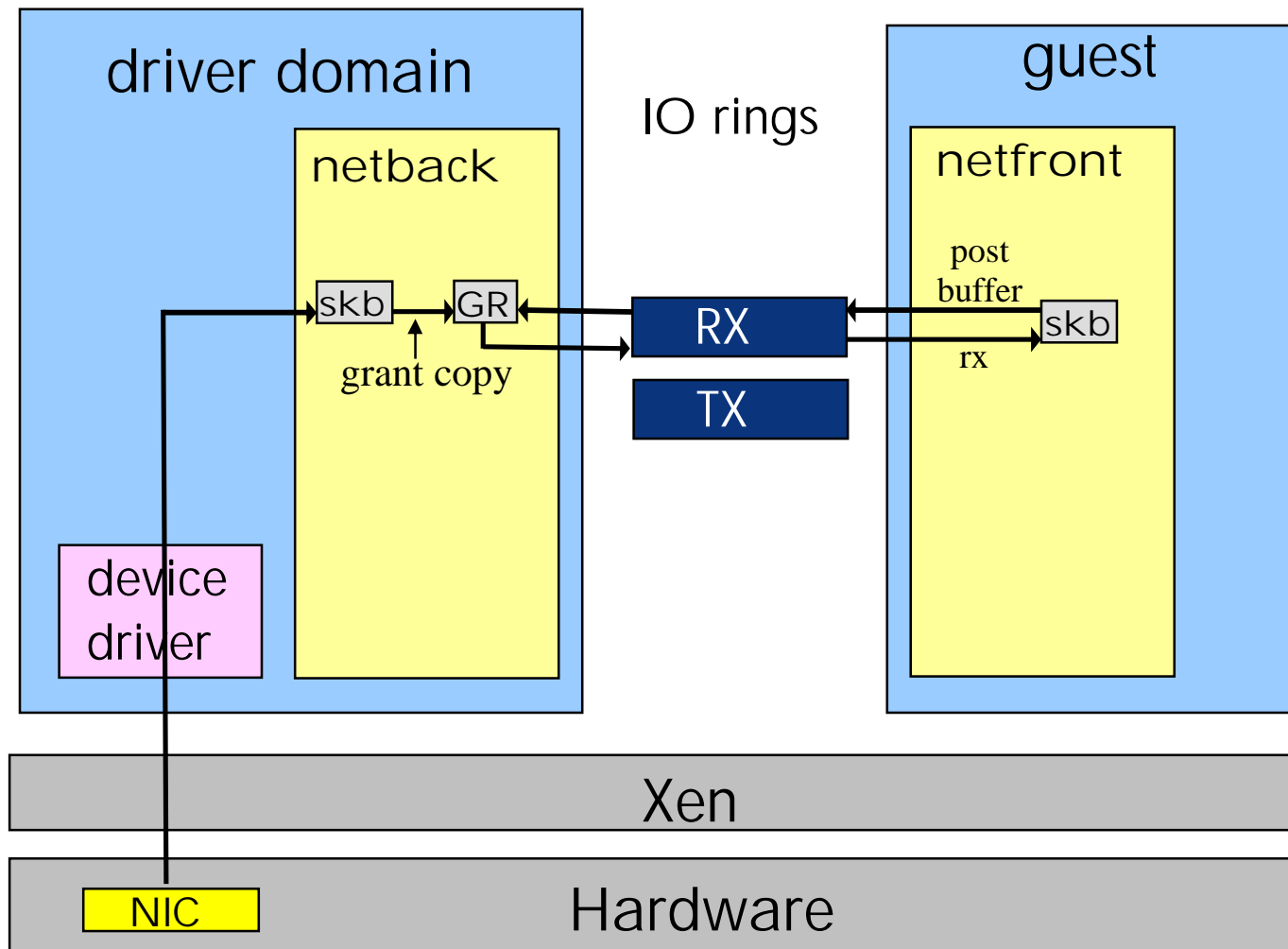
xen0: xen code executed in the context of domain 0

Implementation optimizations on RX

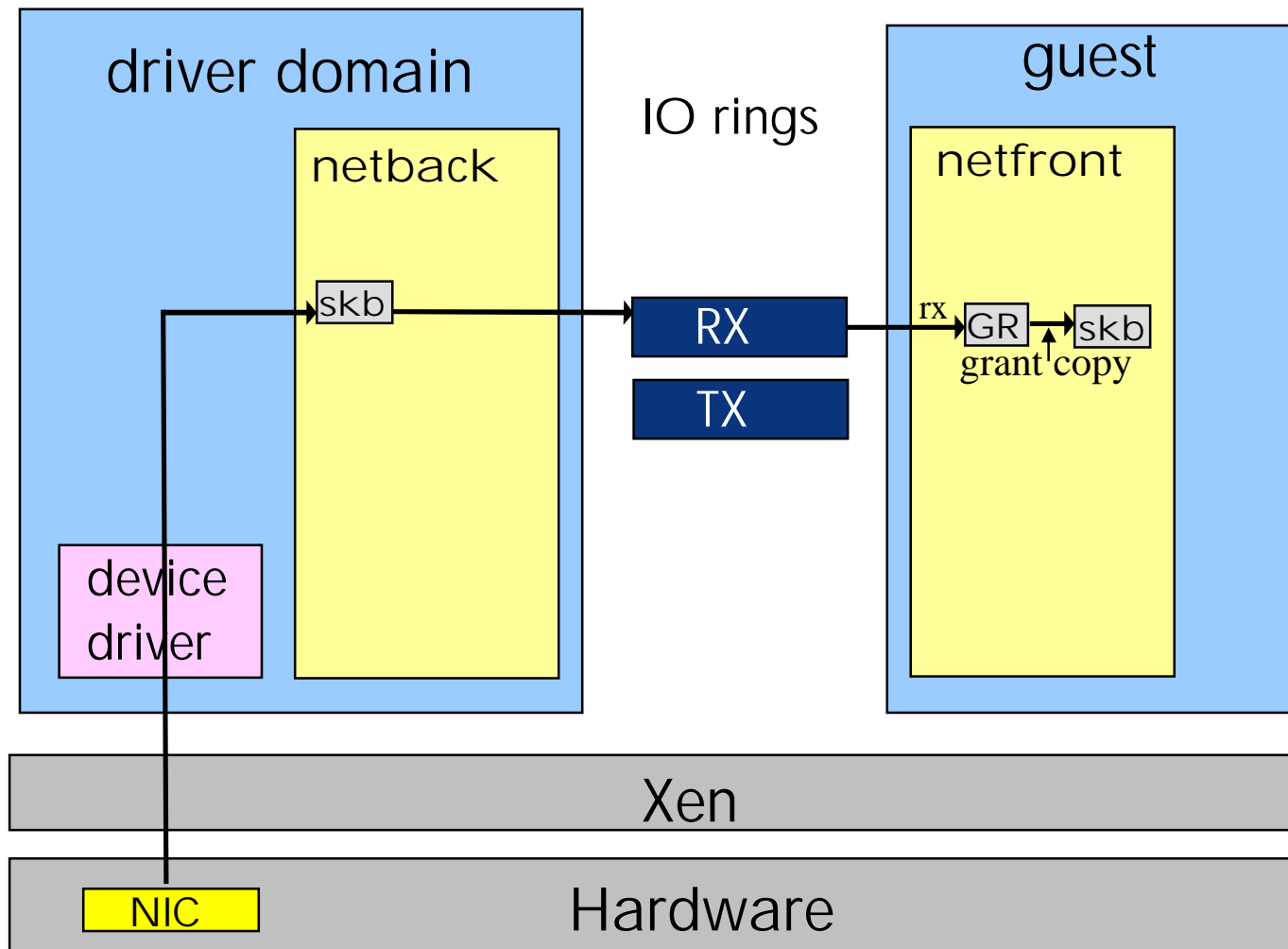


- Implementation optimizations (also possible in netchannel 1)
 - Disabling netfilter on bridge
 - Fix grant copy alignment problem
 - Avoid fragments on single page packets
 - A few optimizations in grant code

Background: receive path on Xen today



Netchannel 2: Moving grant copy to guest



Netchannel 2: Moving grant copy to guest



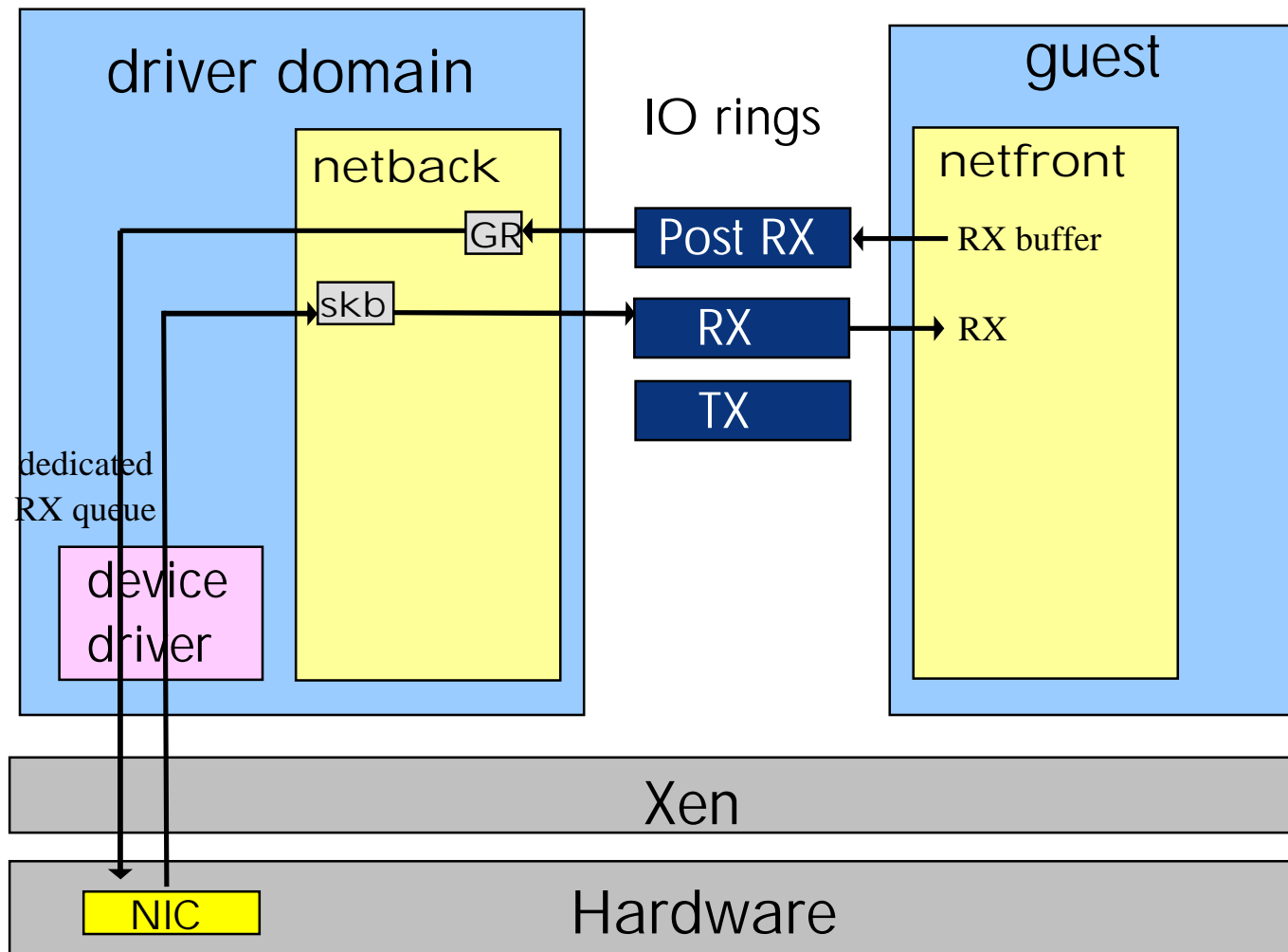
- Improve performance by placing packet in the guest CPU cache
 - Improve 2nd data copy from kernel to user
 - Avoid polluting dom0 CPU cache
- Improve resource management
 - Assigns CPU cost of data copy to the guest
- Better scalability
 - Eliminates dom0 as data copy bottleneck

Netchannel 2: Multi queue device support



- Device has multiple RX queues
- Dedicate one RX queue to a particular guest
- Program device to demultiplex incoming packets to the dedicated queue using guest MAC address
- Post RX descriptors pointing to guest memory
- Device places received packet directly into guest memory avoiding data copy

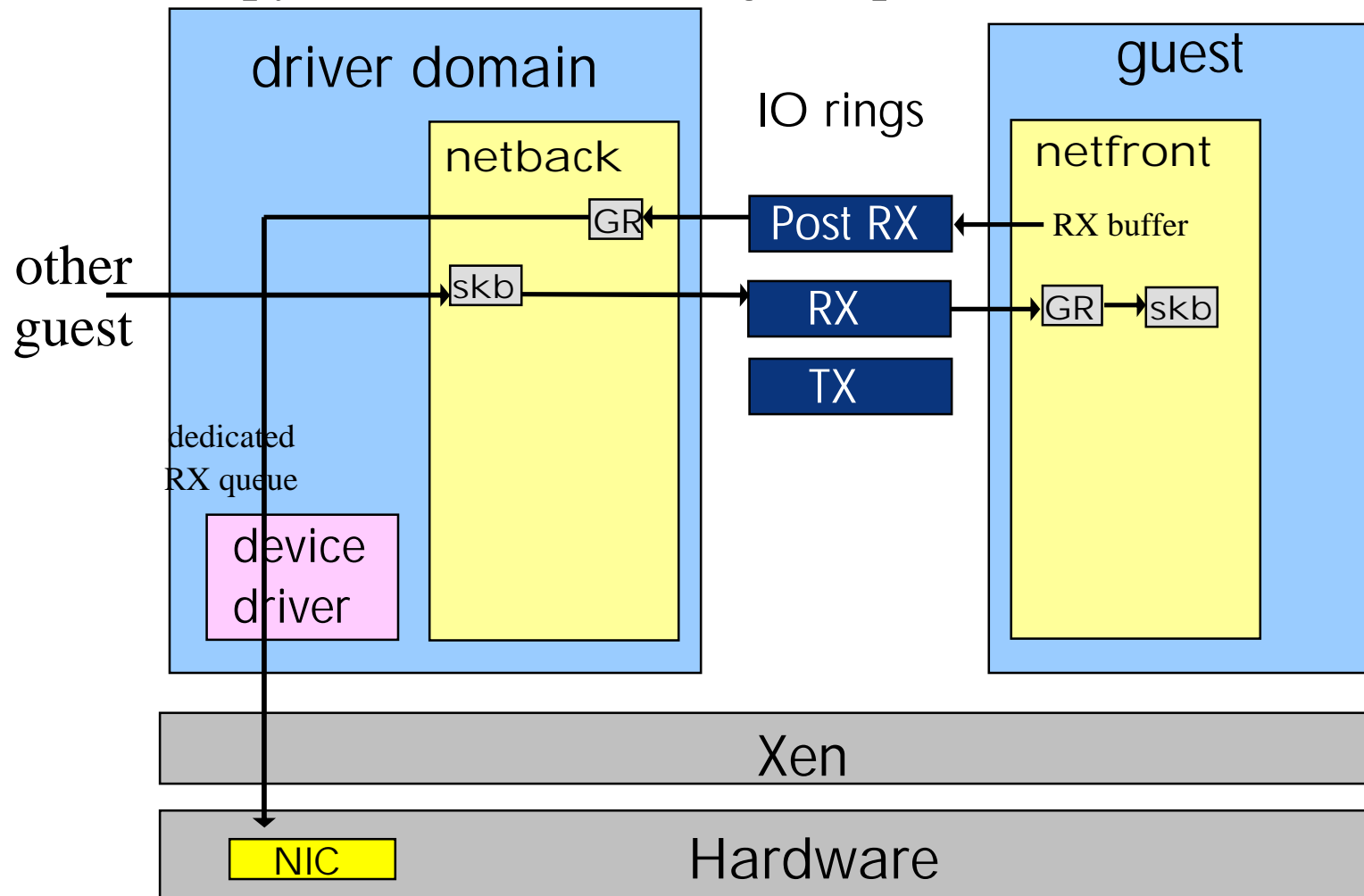
Netchannel 2: Multi-queue device support



Netchannel 2: Multi-queue device support



Grant copy still used for inter-guest packet (also multicast, broadcast)



Netchannel 2: Multi-queue device support



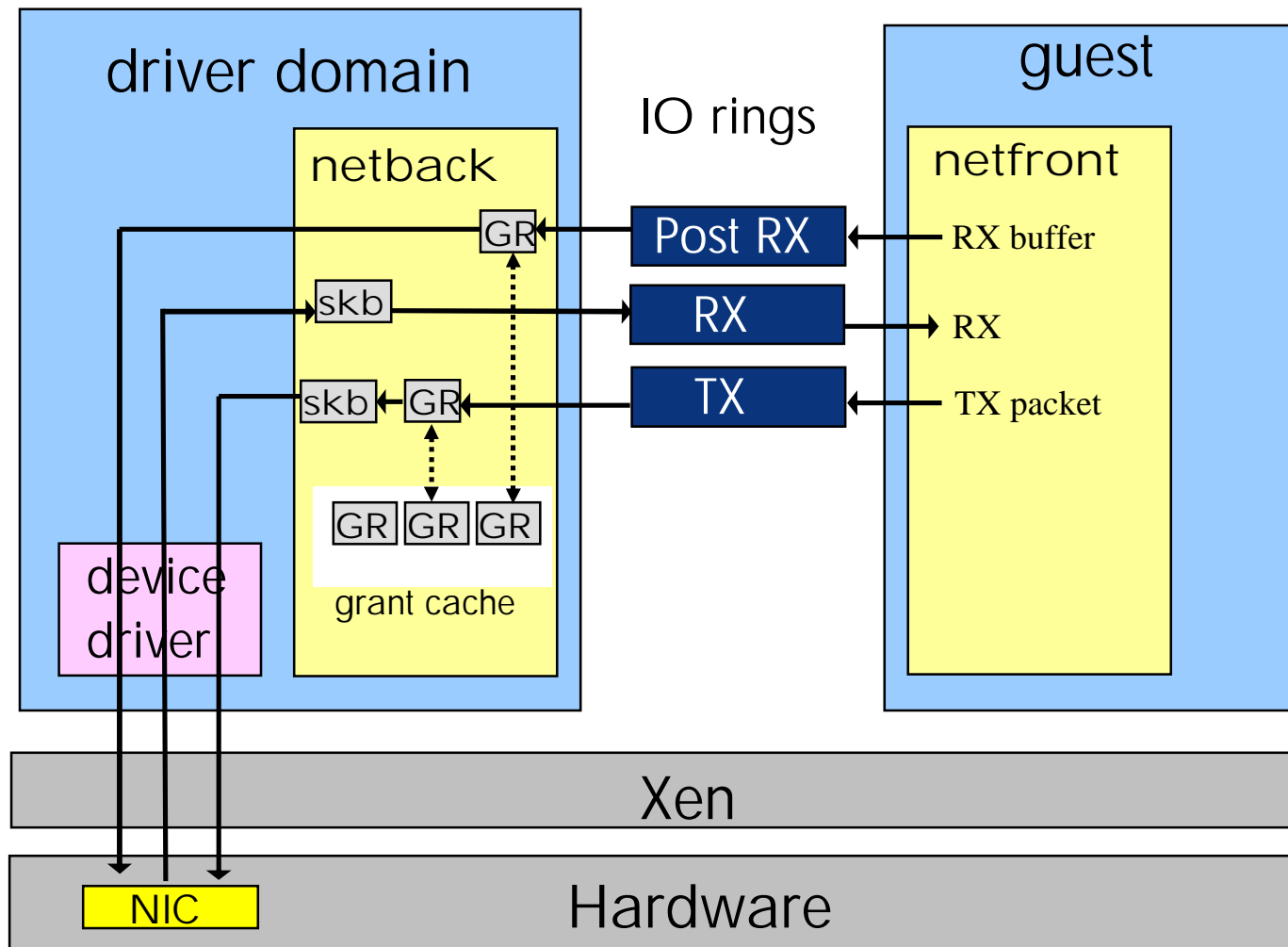
- Need to extend linux netdev interface (for native device driver)
 - Buffer posted on dedicated queue must be allocated from the respective guest pool
 - Need a new buffer allocation function that selects the memory pool based on the queue id
 - In Xen this will be mapped to a function in netback

Netchannel 2: Grant caching in driver domain

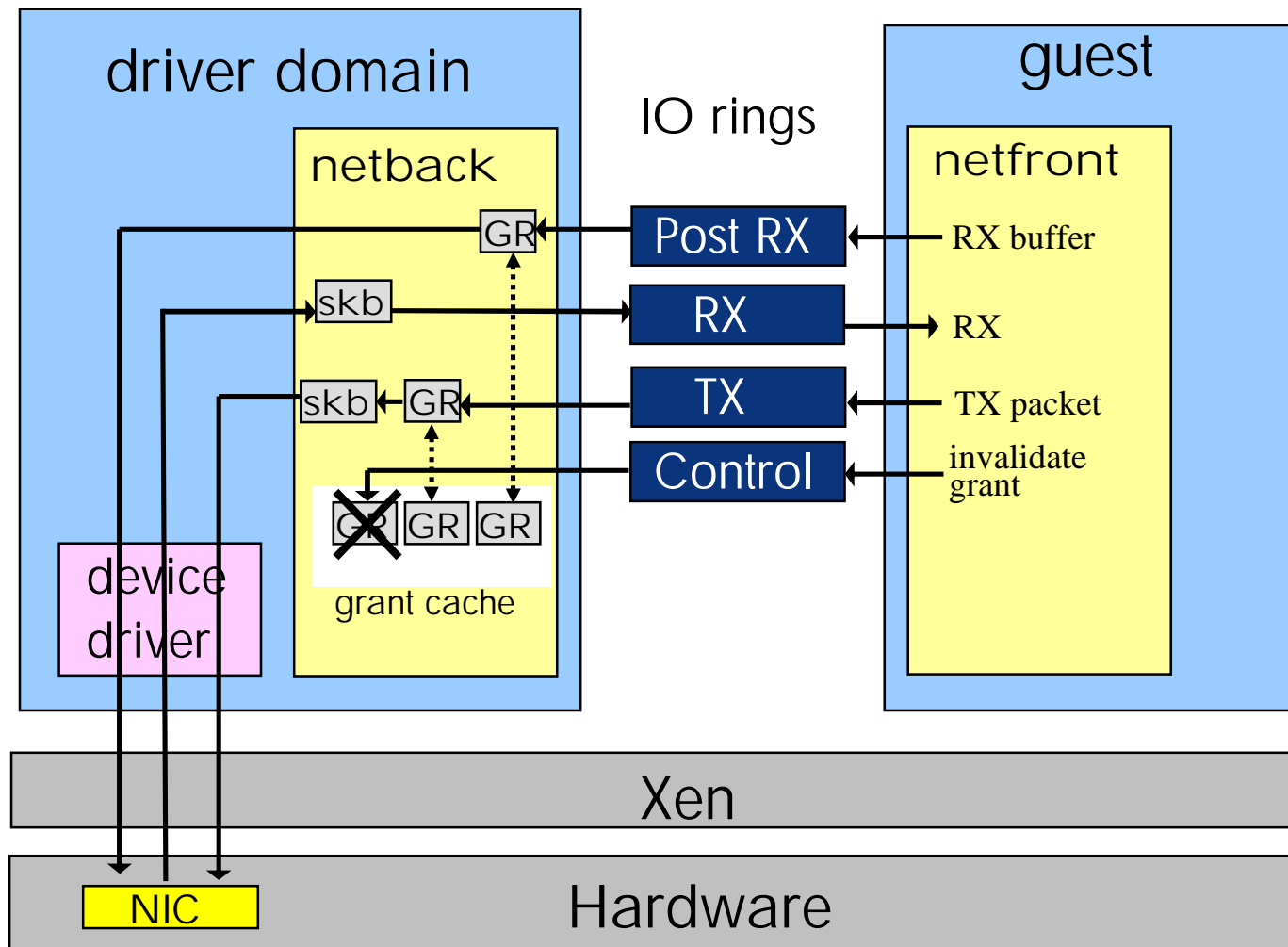


- Avoid unmapping grants, expecting that the same guest memory will be reused in the future
 - Reduces grant mapping/unmapping overhead
- RX buffers should have high locality
 - Windows and Linux tend to recycle RX buffers
- TX buffer recycling behavior is uncertain
 - Need to evaluate experimentally
 - In Linux we could promote buffer recycling if we can modify the skb allocator

Netchannel 2: Grant caching in driver domain



Netchannel 2: Grant caching in driver domain



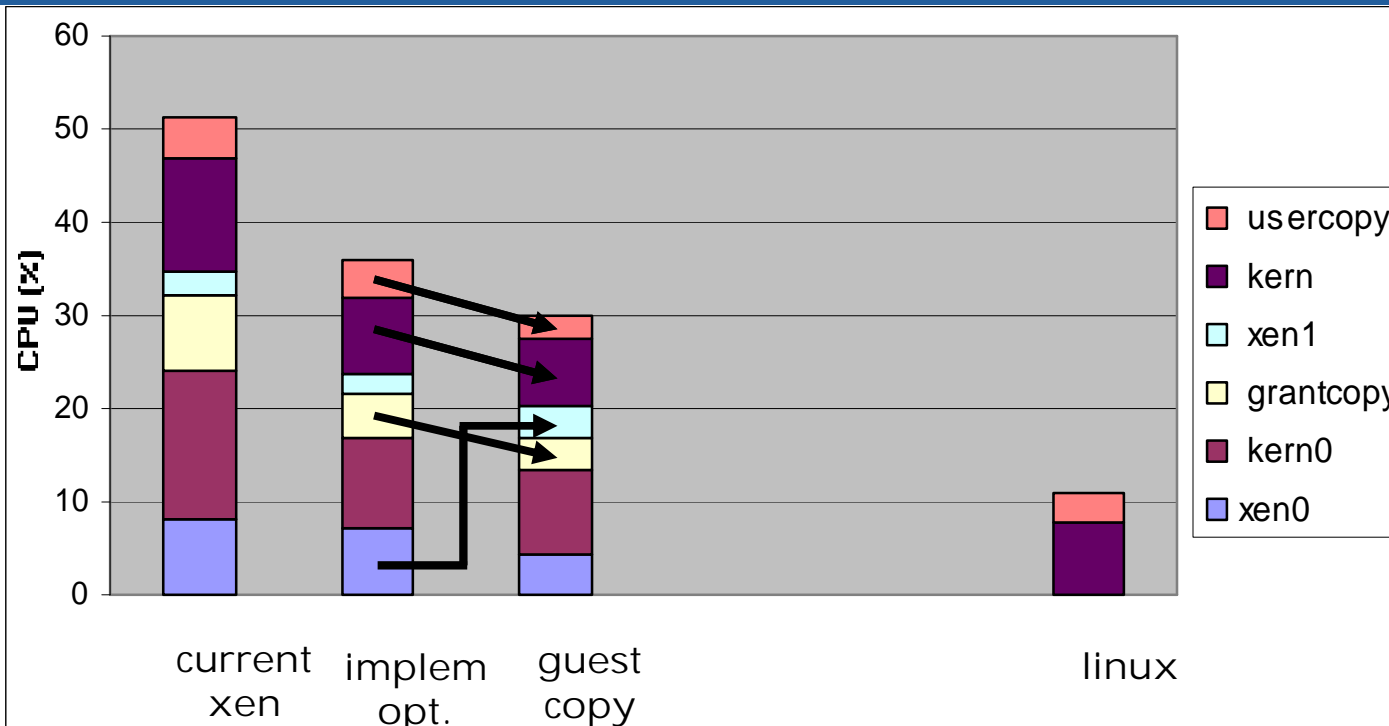
1. Grant transitivity

- Guest authorizes grant to be transferred to other domain
- For guest to guest communication
 - TX guest issues a grant for driver domain
 - Driver domain transfers grant rights to RX guest
 - RX guest uses grant to copy packet to local memory

2. New grant copy operation with range limit

- Grant does not give access to a full page
 - access limited to a range specified by (offset, size)
- Prevents guest from snooping on other guest packets previously received on the same page

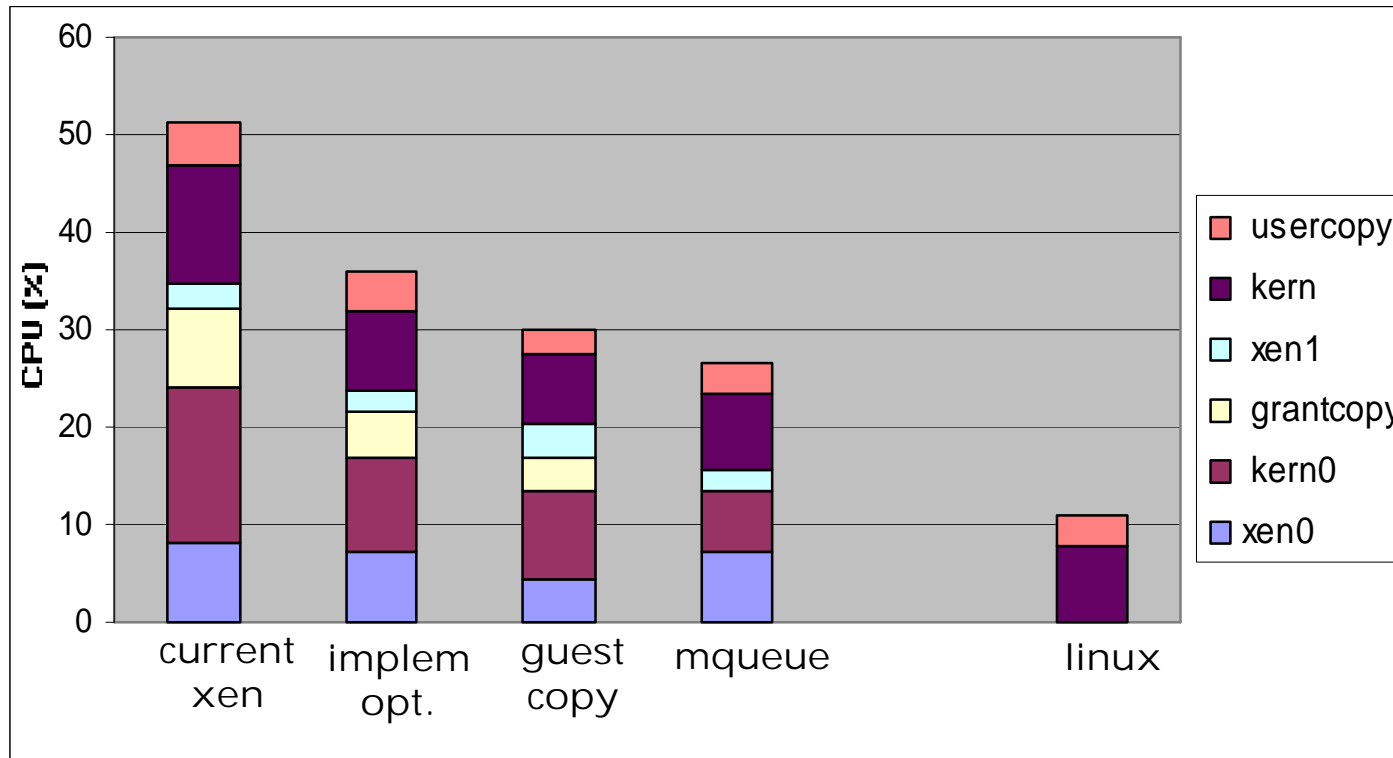
Netchannel 2: RX performance improvements



- Guest copy improvements

- usercopy: cached packet reduces copy overhead
- grant copy: lower overhead on copy (possibly because copy is cache aligned)
- kern: kernel also benefits from cached packet for packet header access
- xen: grant copy cost moved from dom0 (xen0) to guest (xen1); also grant optimizations are more effective on guest side (pin read-only page)

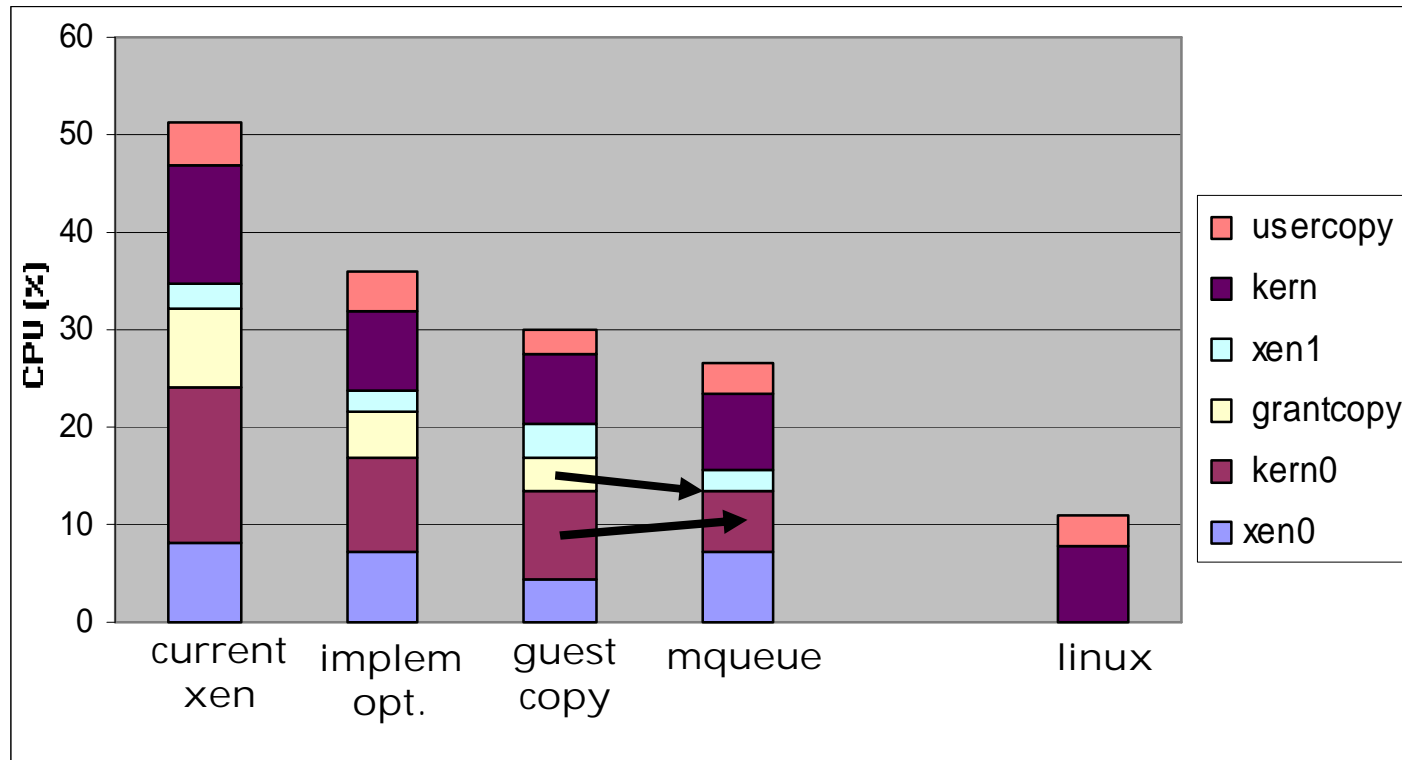
Netchannel 2: RX performance improvements



- multi queue

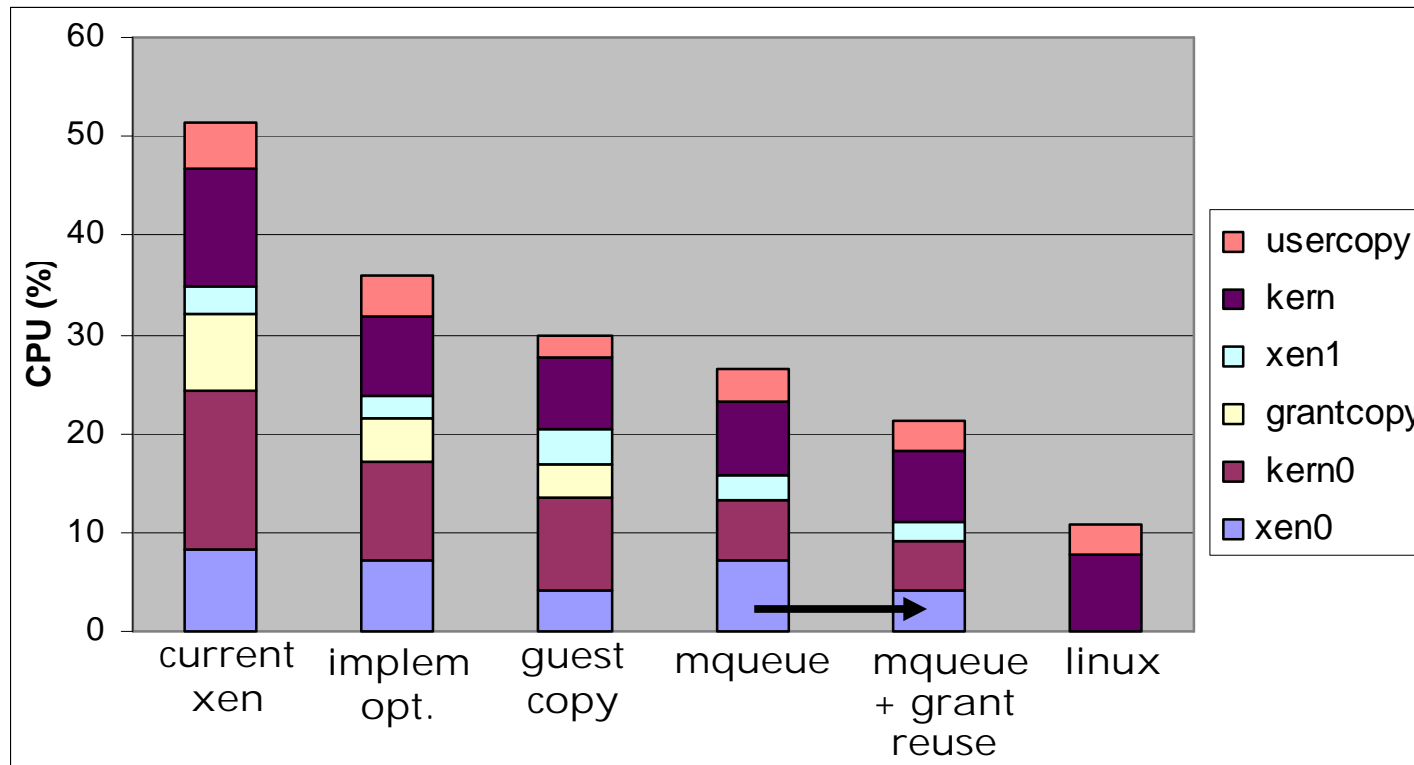
- multi-queue behavior emulated by modifying device driver of traditional NIC and dedicating device RX queue to guest

Netchannel 2: RX performance improvements



- multi queue improvements
 - grant copy eliminated
 - kern0: reduced network processing cost in dom0:
 - no bridge overhead
 - no socket buffer allocation/deallocation.overhead

Netchannel 2: RX performance improvements



- multi queue + grant cache

- grant caching emulated assuming 100% hit rate on cache
- Grant cache benefit: eliminates grant operation overhead in dom0 (xen0)

Interrupt throttling

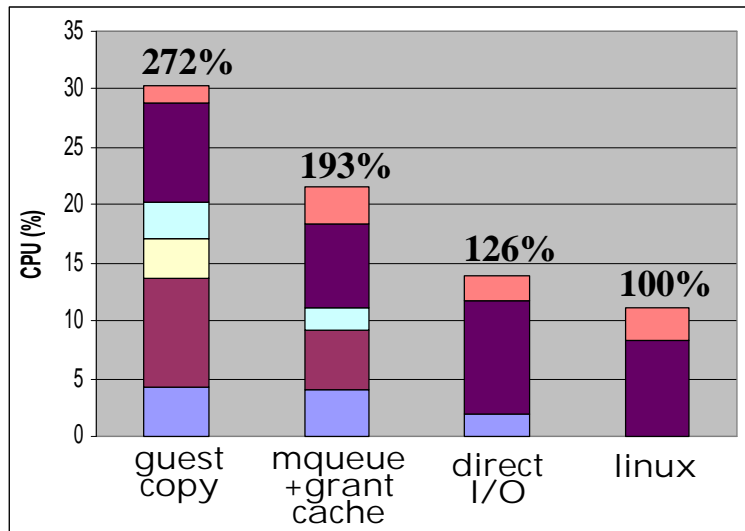


- Netback process RX packets in batches
 - Number of packets in each batch determined by the number of packets received in each hardware interrupt
- Most remaining Xen overhead proportional to number of batches
 - Interrupts, event notification/delivery, Xen scheduler runs
 - Increasing batch size should reduce Xen overhead
- NIC can be configured to throttle interrupt rate (coalescing)
 - RX interrupt delayed until
 - N packets received (limit interrupt rate at high throughput)
 - Or after a given timeout (limit latency at low throughput)
- Latency sensitive applications should not be significantly affected by larger batch size
 - Latency effect is limited by latency coalescing parameter

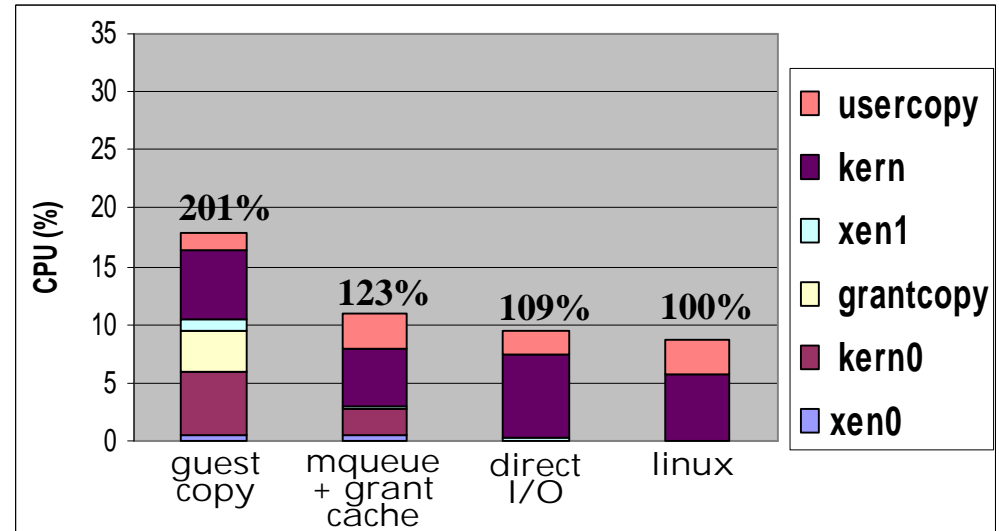
Performance impact of interrupt throttling



Default configuration (6 pkt/intr)



Interrupt throttling config (64 pkt/intr)



- Interrupt throttling significantly improves Xen performance
- Default configuration is good for Linux (but not optimal for Xen)
 - Xen users should use different device coalescing settings
- Large batches achieve almost native performance with multi-queue (23% overhead)

Conclusion



- Netchannel 2 will provide:
 - significant performance improvement for traditional NICs
 - Overhead over linux reduced by 3.7 times on RX (370% to 100%)
 - near native performance for RX on multi-queue devices
 - 23% overhead over linux
- Multi-queue devices can be a good alternative to direct I/O devices (direct guest access)
 - Slightly higher CPU cost
 - But no hardware dependency on guest
 - single device driver to maintain, test, debug, etc. Easier to migrate, easier to monitor/enforce traffic policies (firewall, rate control, etc)

