



Topic: A Deep Dive into Memory Access

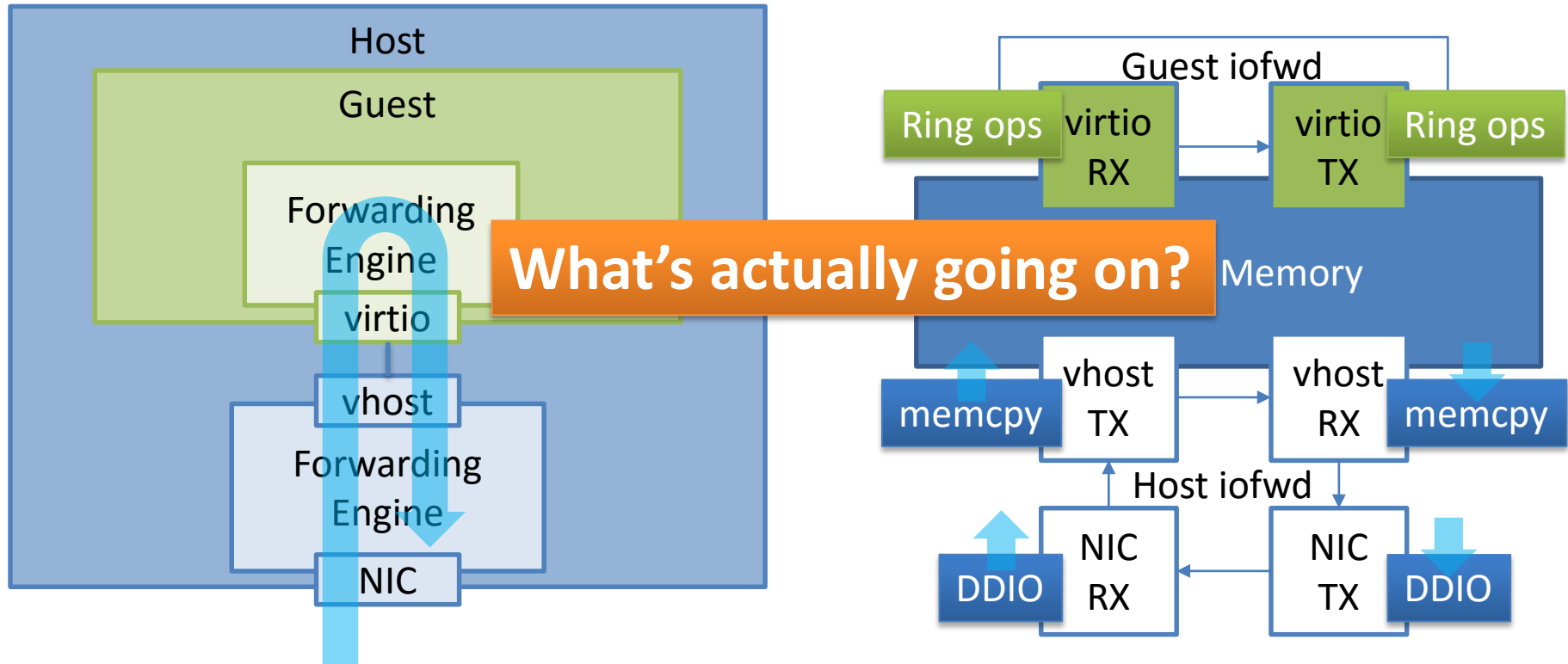
Company: Intel

Title: Software Engineer

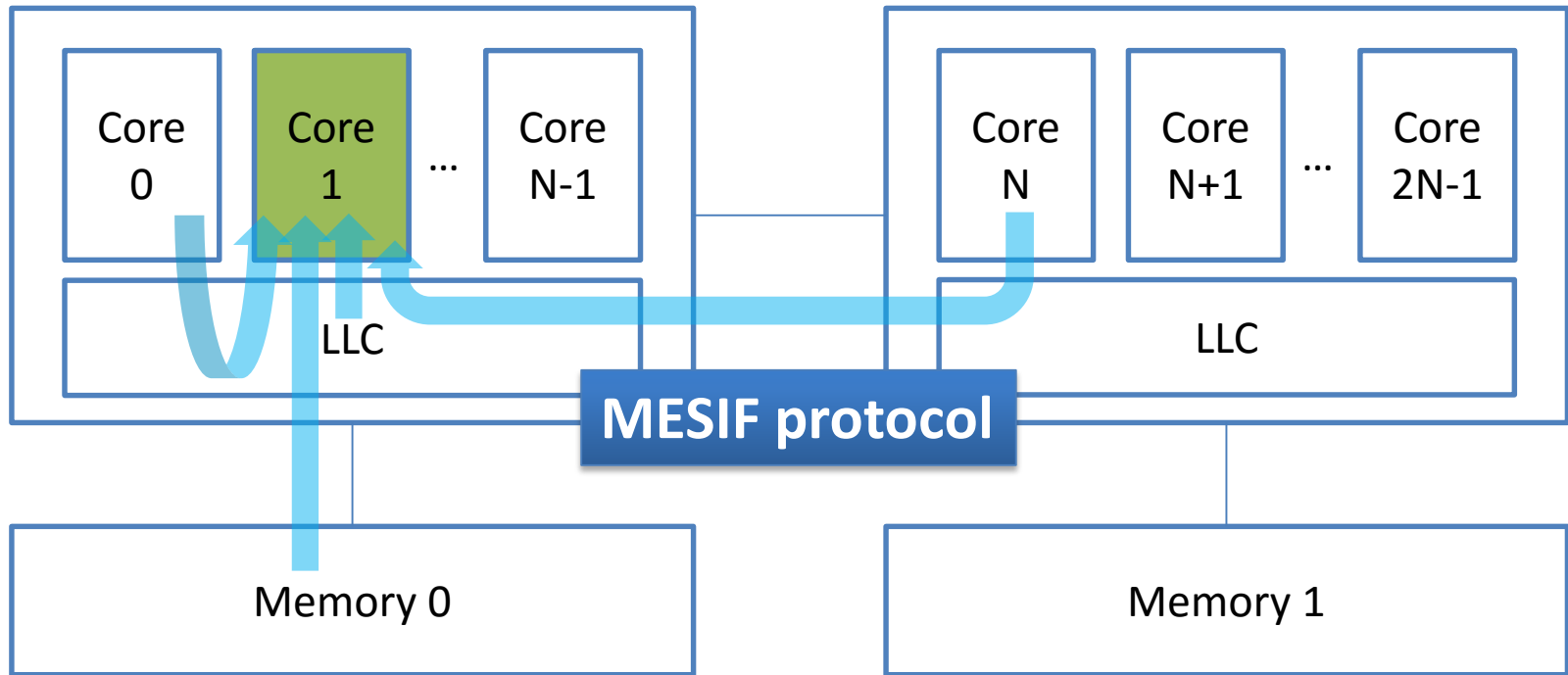
Name: Wang, Zhihong



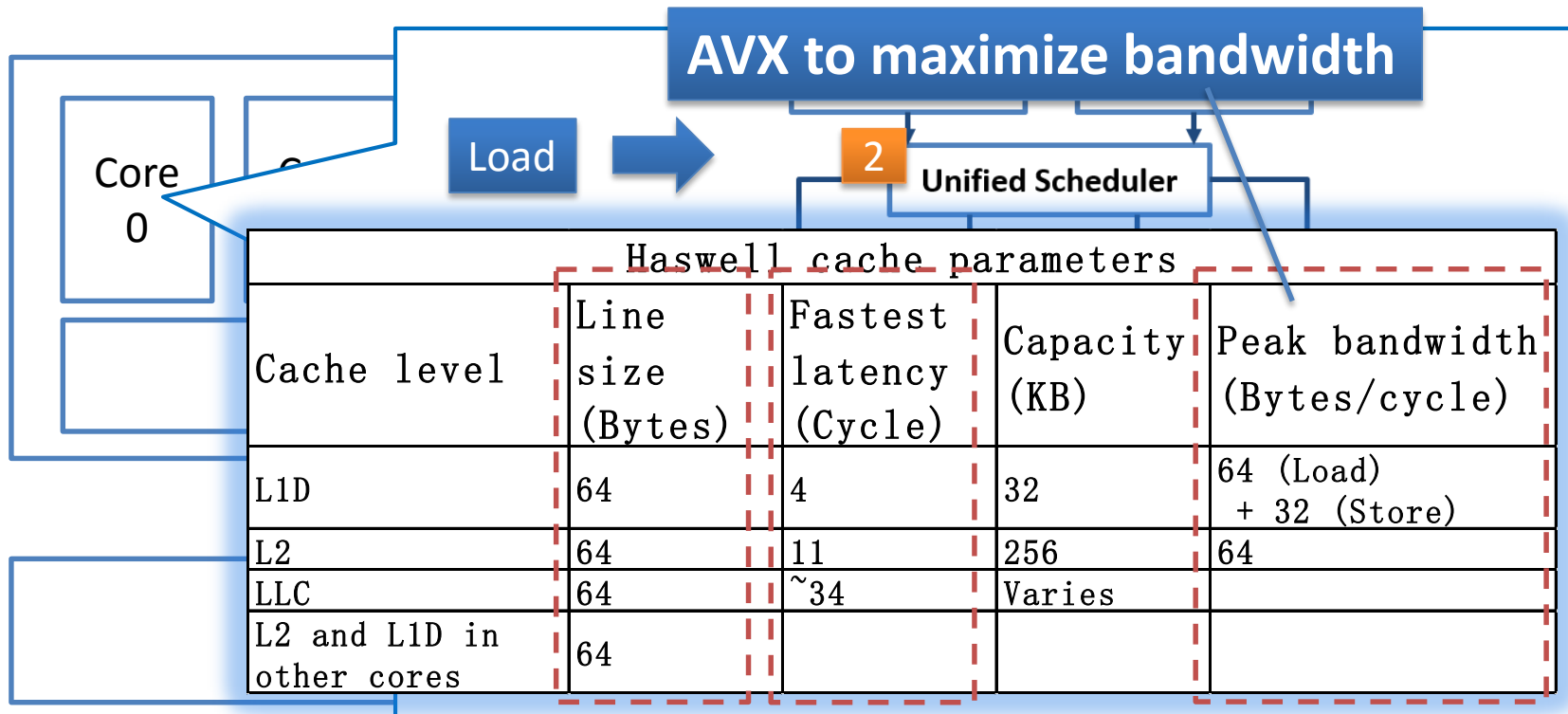
A Typical NFV Scenario: PVP



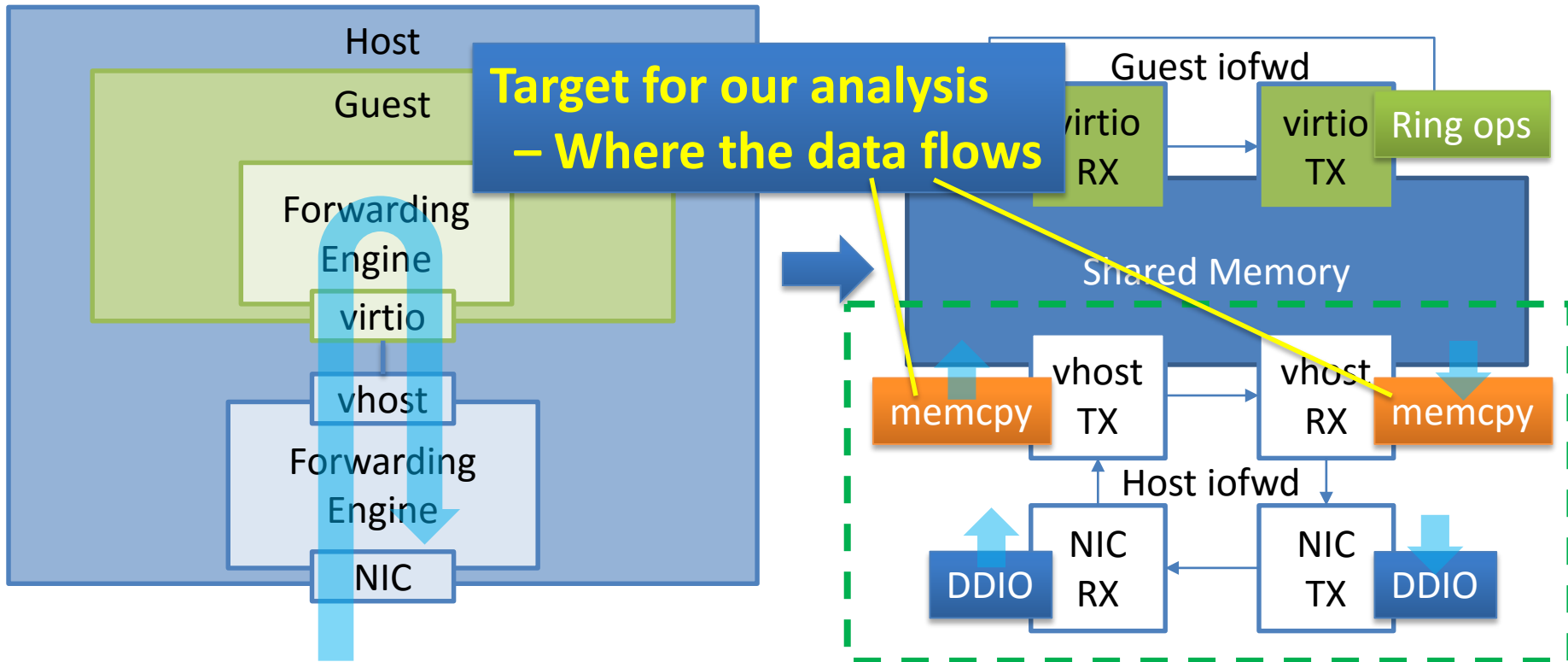
Overview of Memory System



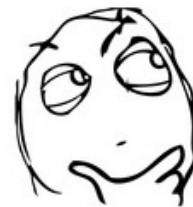
Overview of Memory System (cont'd)



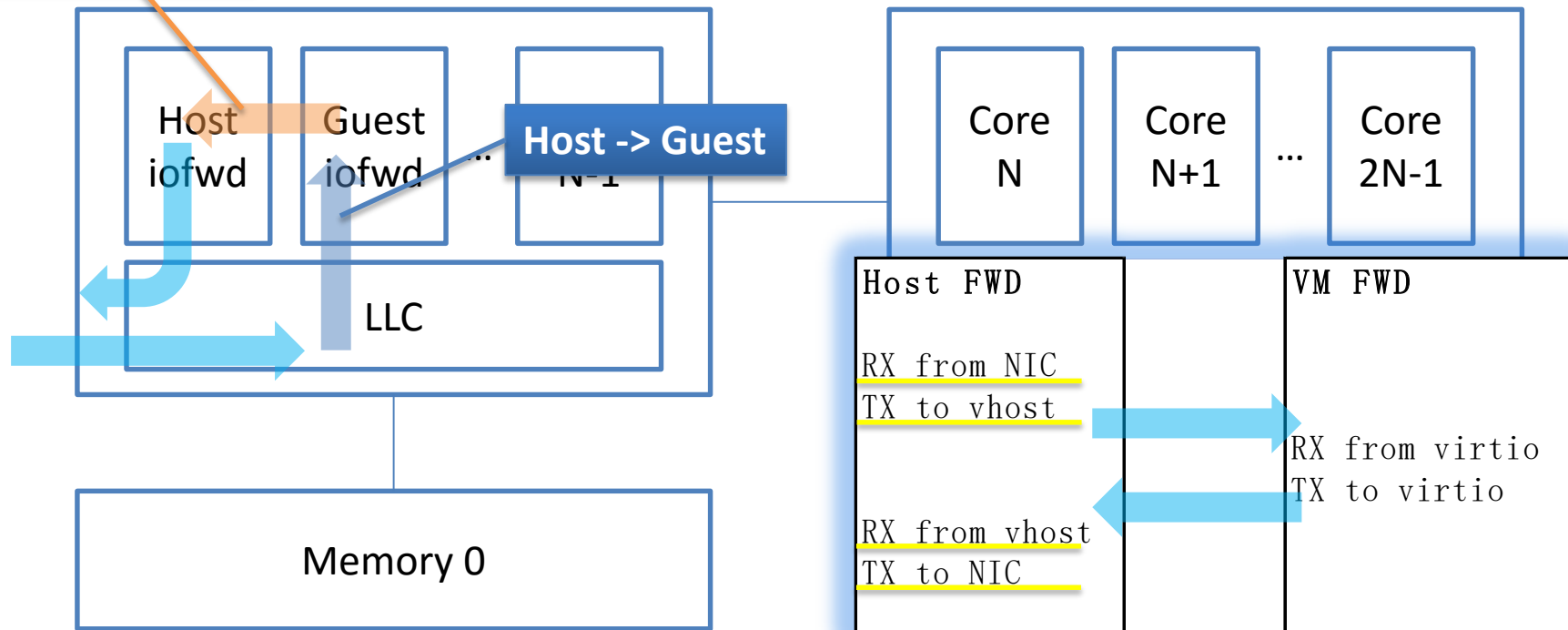
Let's Do It!



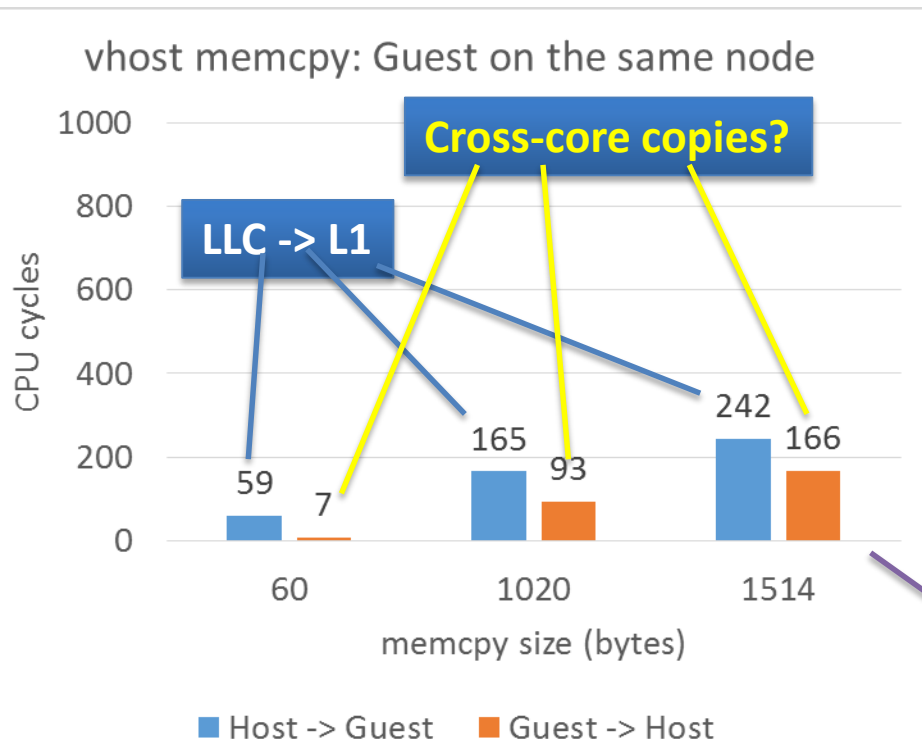
First Impression



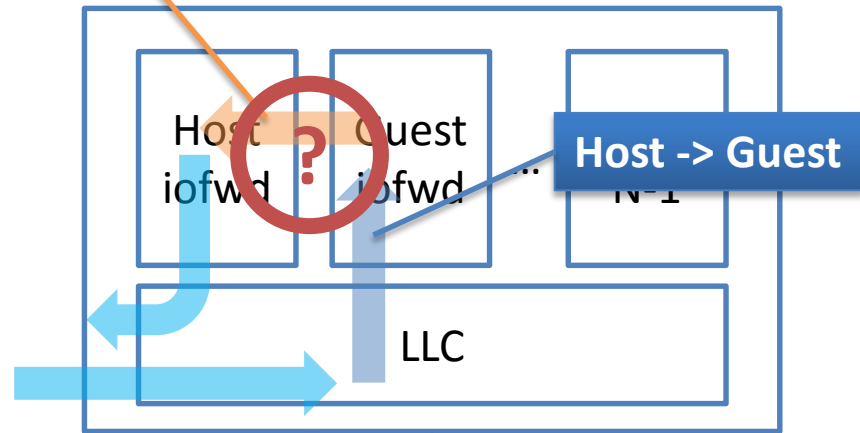
Guest -> Host



Unexpectedly...



Guest -> Host



Notice

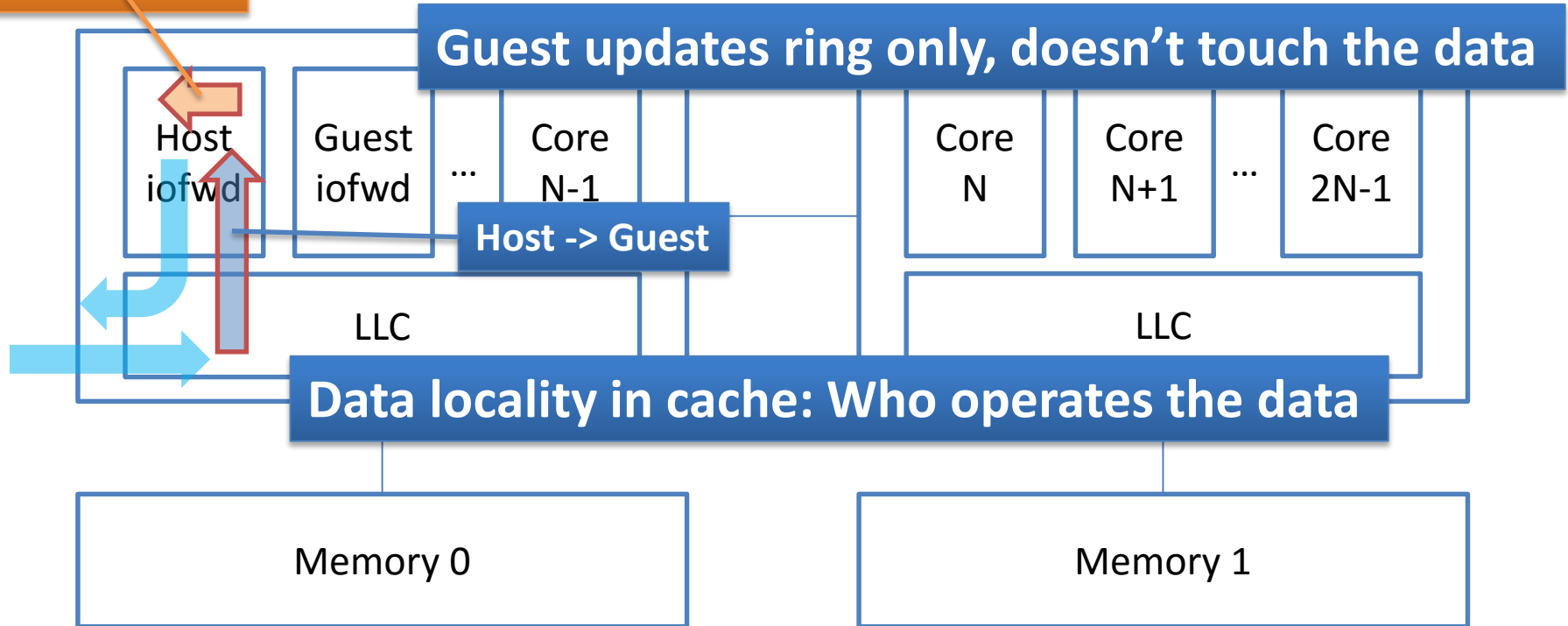
– CPU cycle measurement disturbs overall performance

First try

Under The Hood

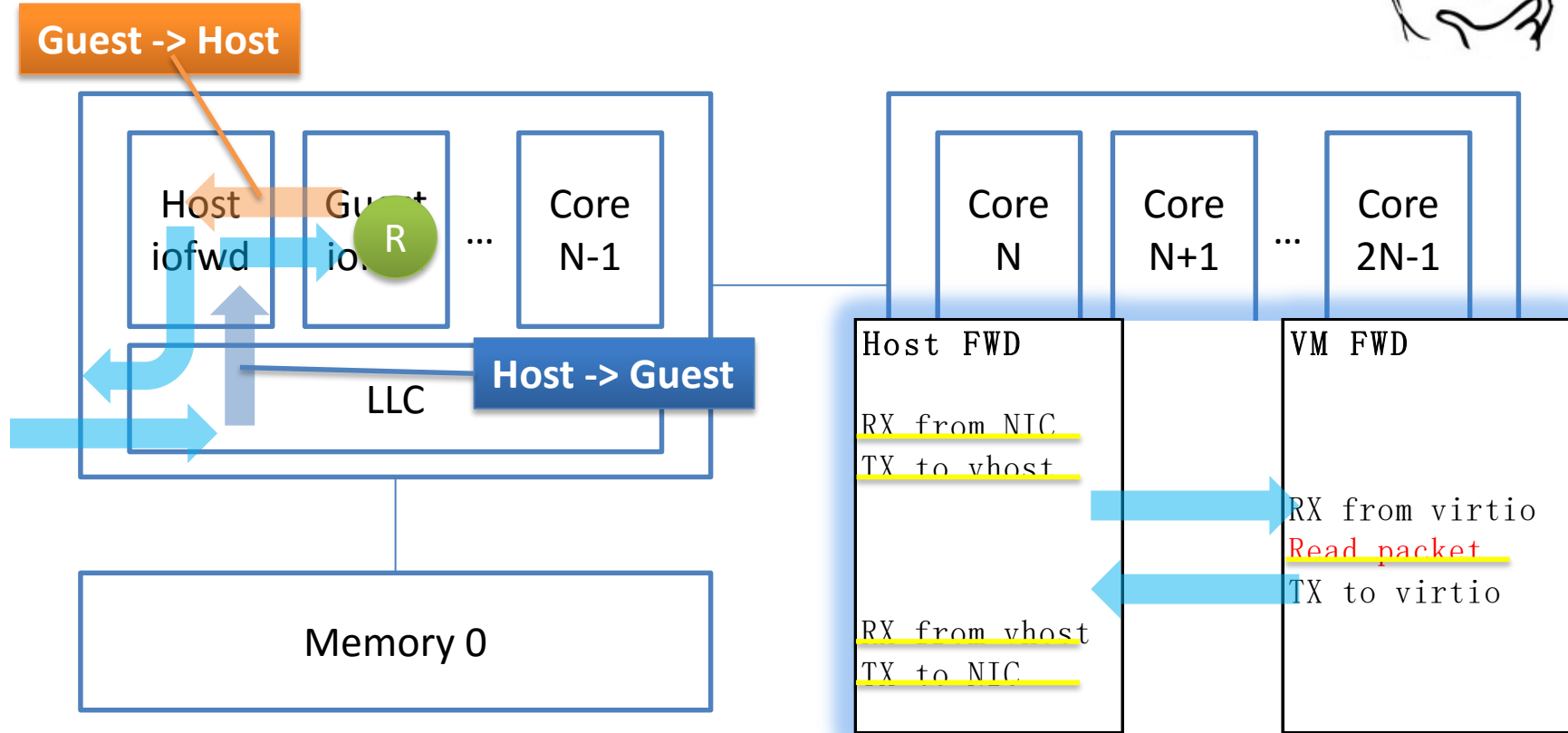
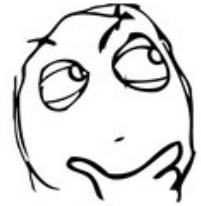
Guest -> Host

Guest updates ring only, doesn't touch the data

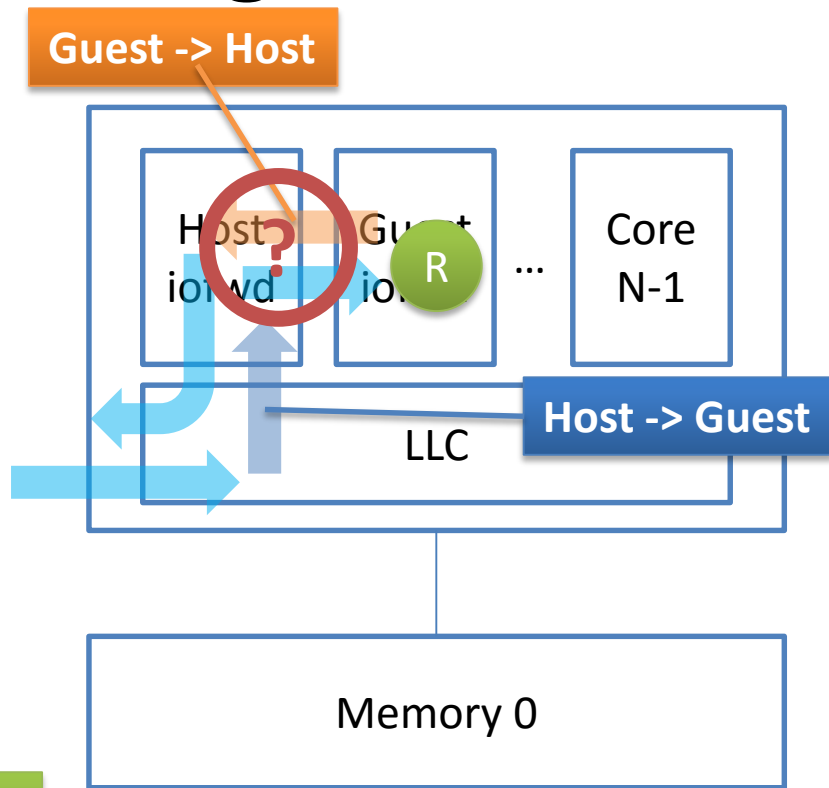
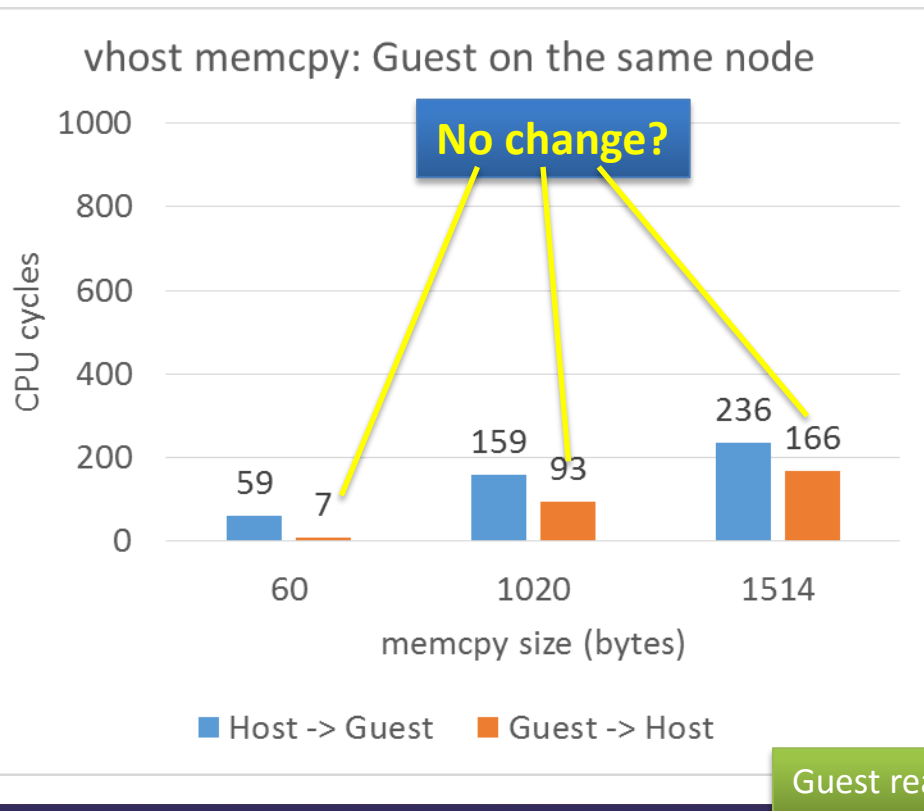


Data locality in cache: Who operates the data

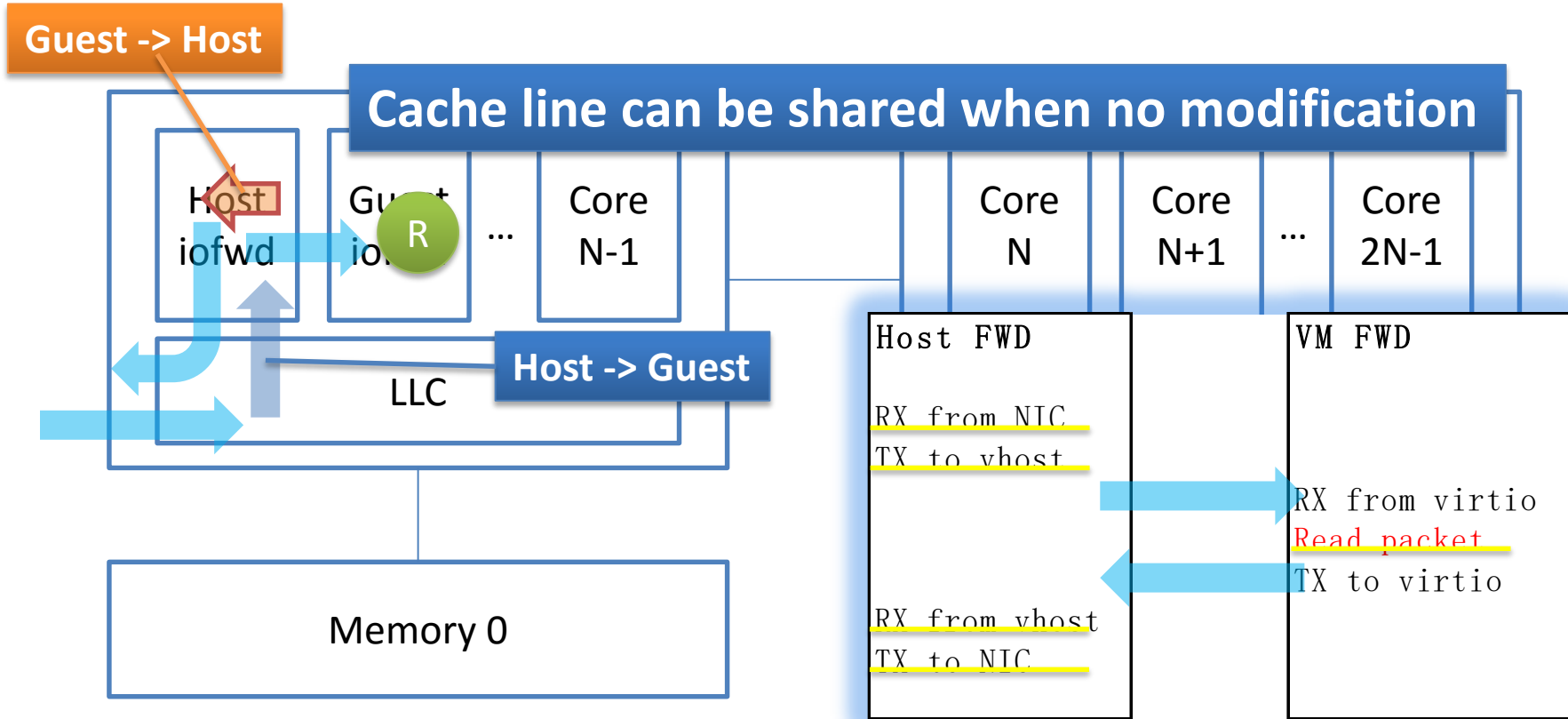
Guest Read The Packet



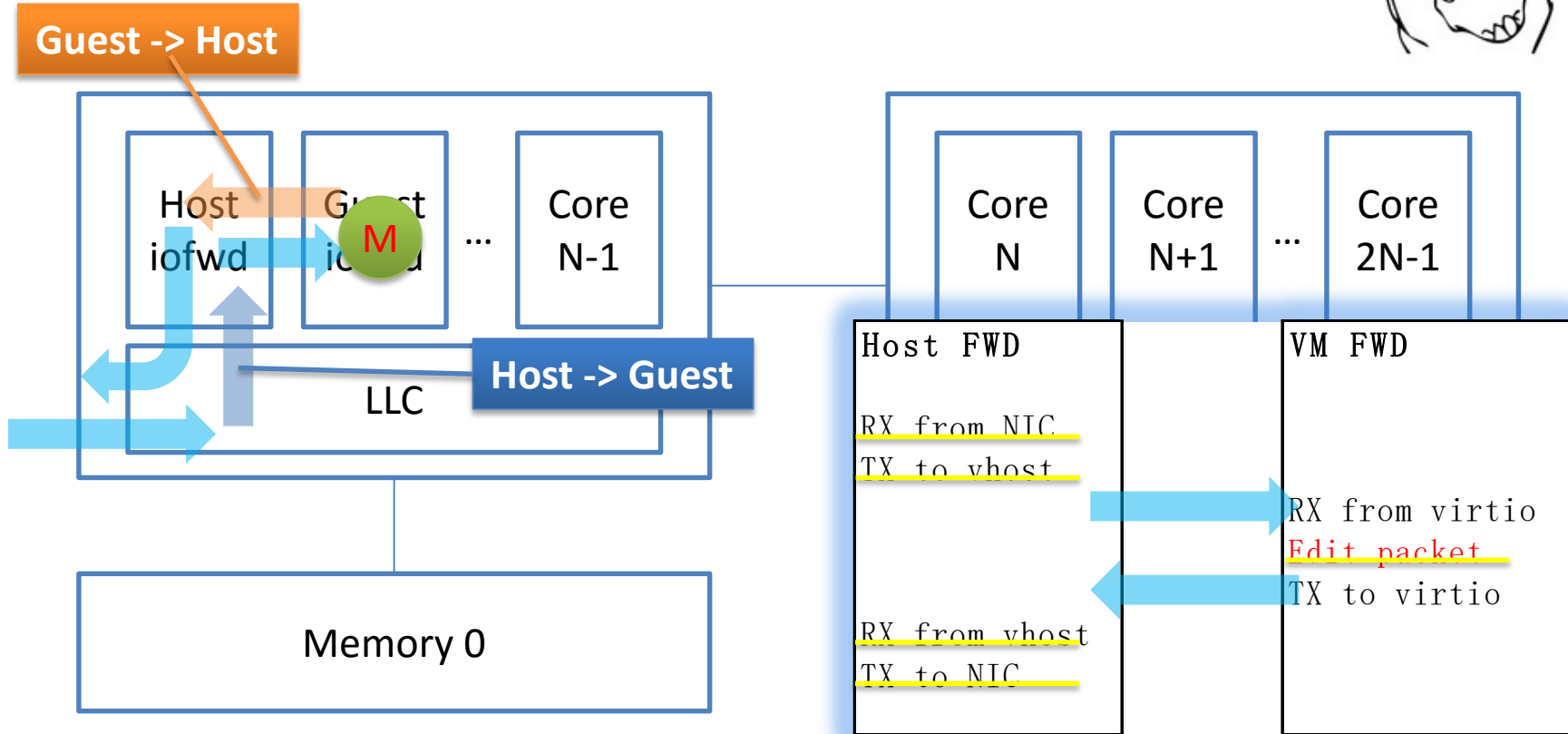
Still Doesn't Feel Right...



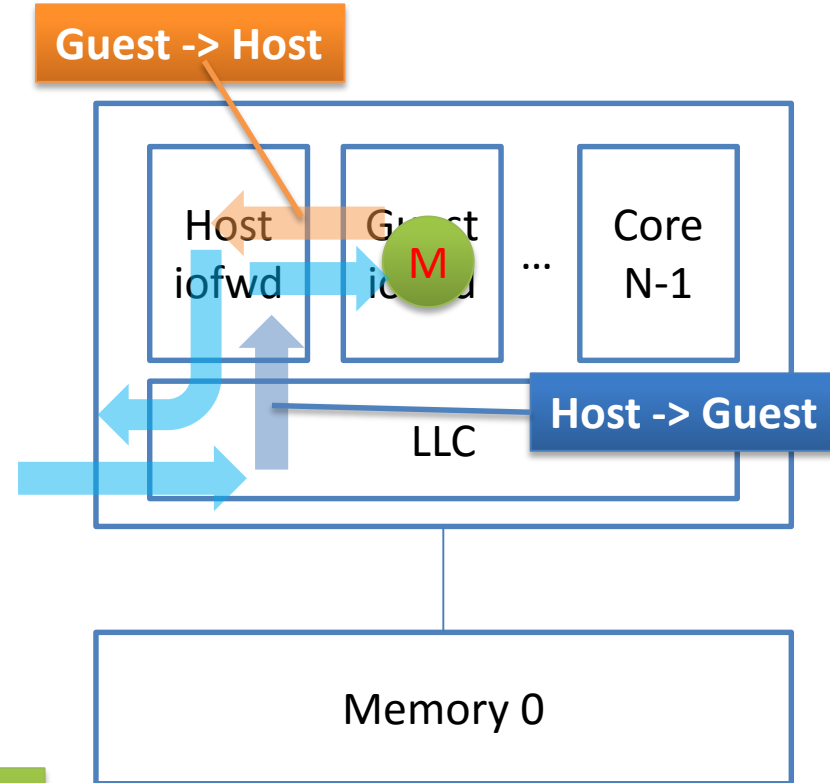
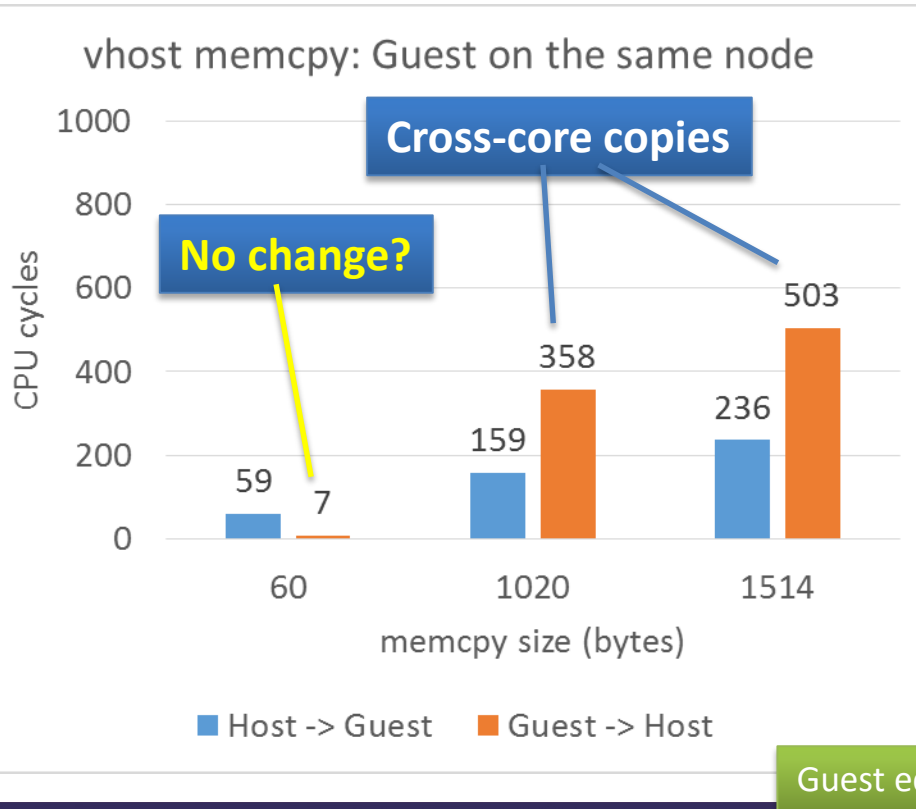
Under The Hood



Guest Edit The Packet



Write-back



Go See Some C Code

```
desc_addr = gpa_to_vva(dev, desc->addr);  
rte_prefetch0((void *)((uintptr_t)desc_addr);
```

Oh I see – S/W Prefetching to reduce latency

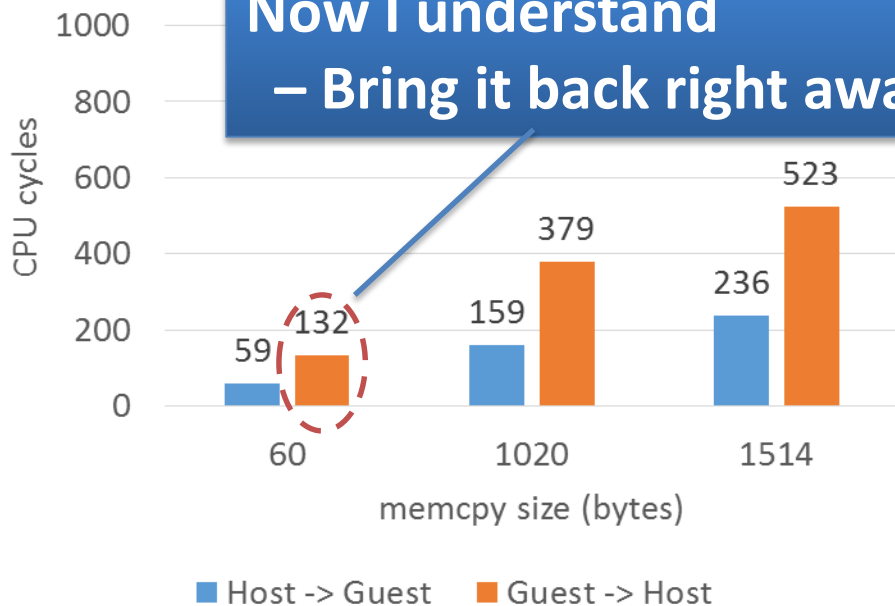
```
rte_memcpy(rte_pktmbuf_mtod_offset(cur, void *, mbuf_offset),  
           (void *)((uintptr_t)(desc_addr + desc_offset)),  
           copy_len);
```

```
desc_addr = gpa_to_vva(dev, desc->addr);  
rte_prefetch0((void *)((uintptr_t)desc_addr);  
  
/* Return the virtio net header  
hdr = (struct virtio_net_hdr *)((uintptr_t)desc_addr);  
desc_avail = desc->len - vq->whost_hlen;  
desc_offset = vq->whost_hlen;  
  
mbuf_offset = 0;  
mbuf_avail = m->buf_len - RTE_PKTMBUF_HEADROOM;  
while (desc_avail != 0 || (desc->flags & VRING_DESC_F_NEXT) != 0) {  
    /* This desc reaches to its end, get the next one */  
    if (desc_avail == 0) {  
        if (unlikely(desc->next >= vq->size ||  
                    ++nr_desc >= vq->size))  
            return -1;  
        desc = &vq->desc[desc->next];  
  
        desc_addr = gpa_to_vva(dev, desc->addr);  
        rte_prefetch0((void *)((uintptr_t)desc_addr);  
  
        desc_offset = 0;  
        desc_avail = desc->len;  
  
        PRINT_PACKET(dev, (uintptr_t)desc_addr, desc->len, 0);  
    }  
  
    /* This mbuf reaches to its end, get a new one  
    * to hold more data.  
    */  
    if (desc_avail == 0) {  
        rte_pktmbuf_alloc(mbuf_pool);  
        likely(cur == NULL) {  
            RTE_LOG(ERR, VHOST_DATA, "Failed to "  
                    "allocate memory for mbuf.\n");  
            return -1;  
        }  
  
        next = cur;  
        data_len = mbuf_offset;  
        mbuf_offset = 0;  
        m->pkt_len += mbuf_offset;  
        prev = cur;  
  
        mbuf_offset = 0;  
        mbuf_avail = cur->buf_len - RTE_PKTMBUF_HEADROOM;  
    }  
  
    copy_len = RTE_MIN(desc_avail, mbuf_avail);  
    rte_memcpy(rte_pktmbuf_mtod_offset(cur, void *, mbuf_offset),  
              (void *)((uintptr_t)(desc_addr + desc_offset)),  
              copy_len);  
    desc_avail -= copy_len;  
    mbuf_offset += copy_len;  
    if (desc->next >= vq->size || ++nr_desc >= vq->size)  
        return -1;  
    desc = &vq->desc[desc->next];  
    desc_addr = gpa_to_vva(dev, desc->addr);  
    rte_prefetch0((void *)((uintptr_t)desc_addr);  
    desc_offset = 0;  
    desc_avail = desc->len;  
    PRINT_PACKET(dev, (uintptr_t)desc_addr, desc->len, 0);  
}
```

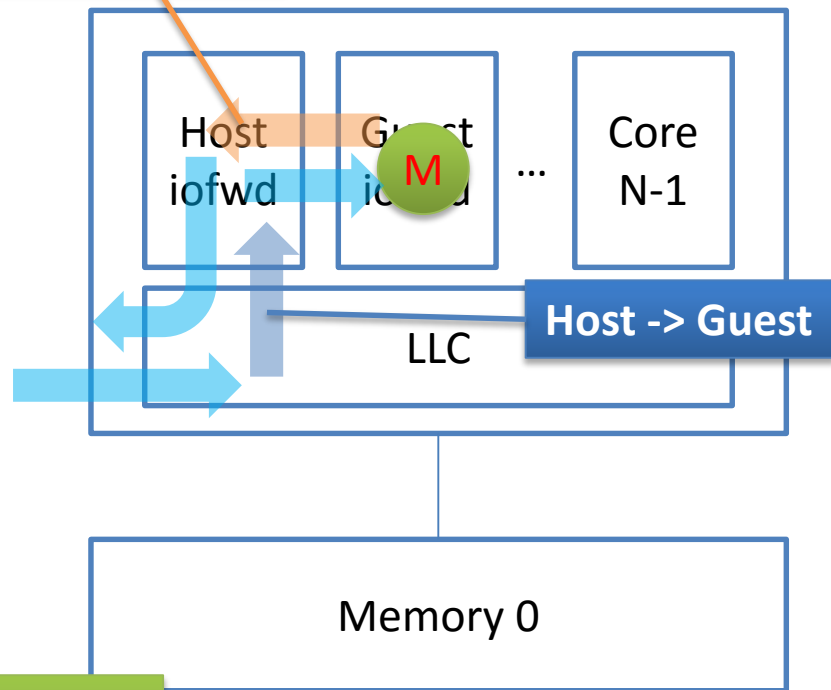
Without S/W Prefetching

vhost memcpy: Guest on the same node

**Now I understand
– Bring it back right away!**

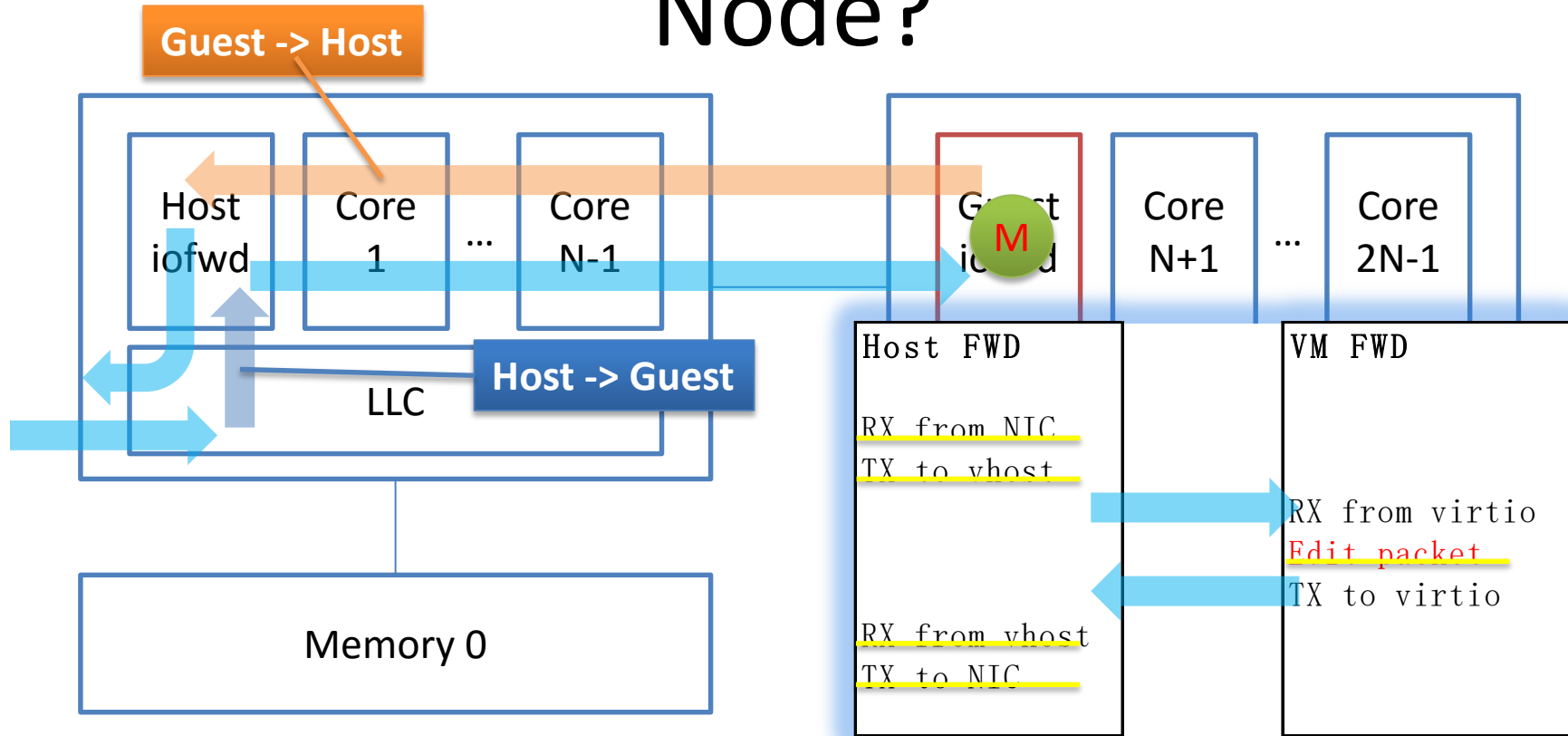


Guest -> Host

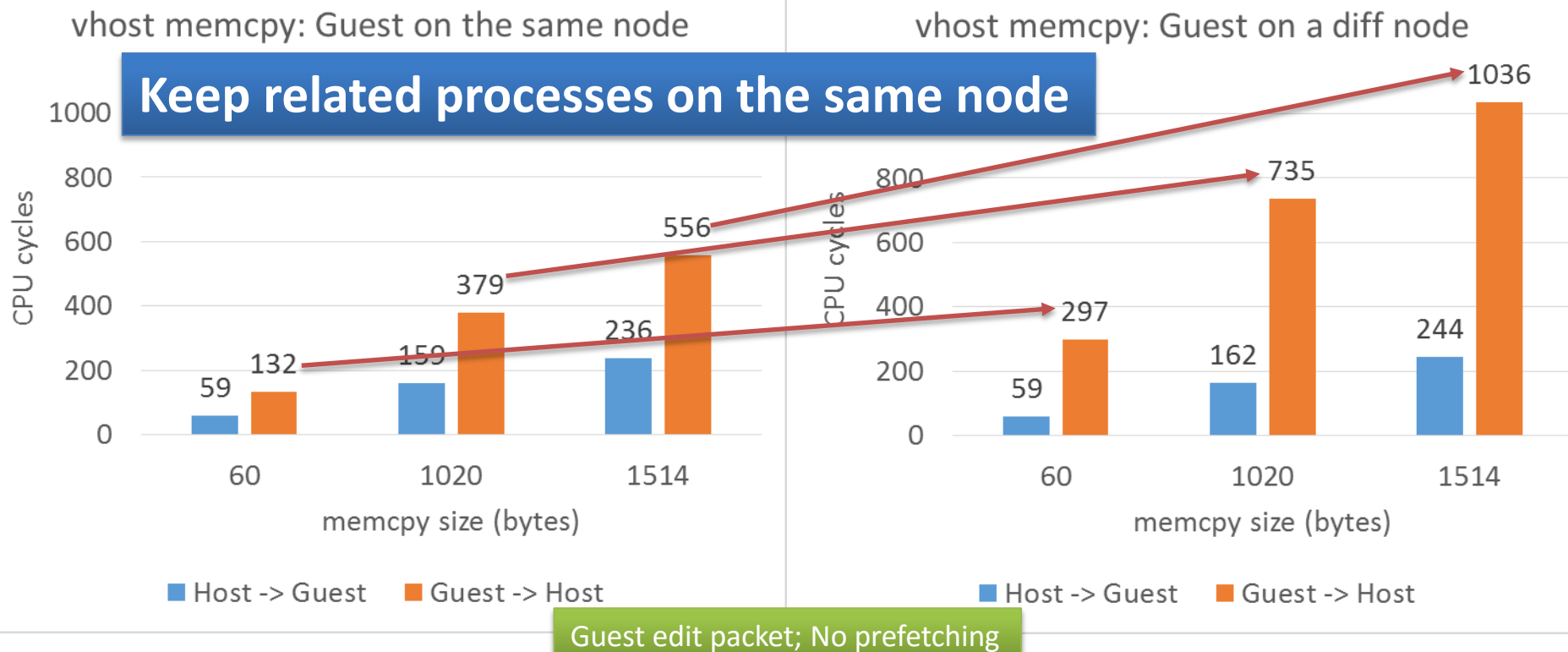


Guest edit packet; No prefetching

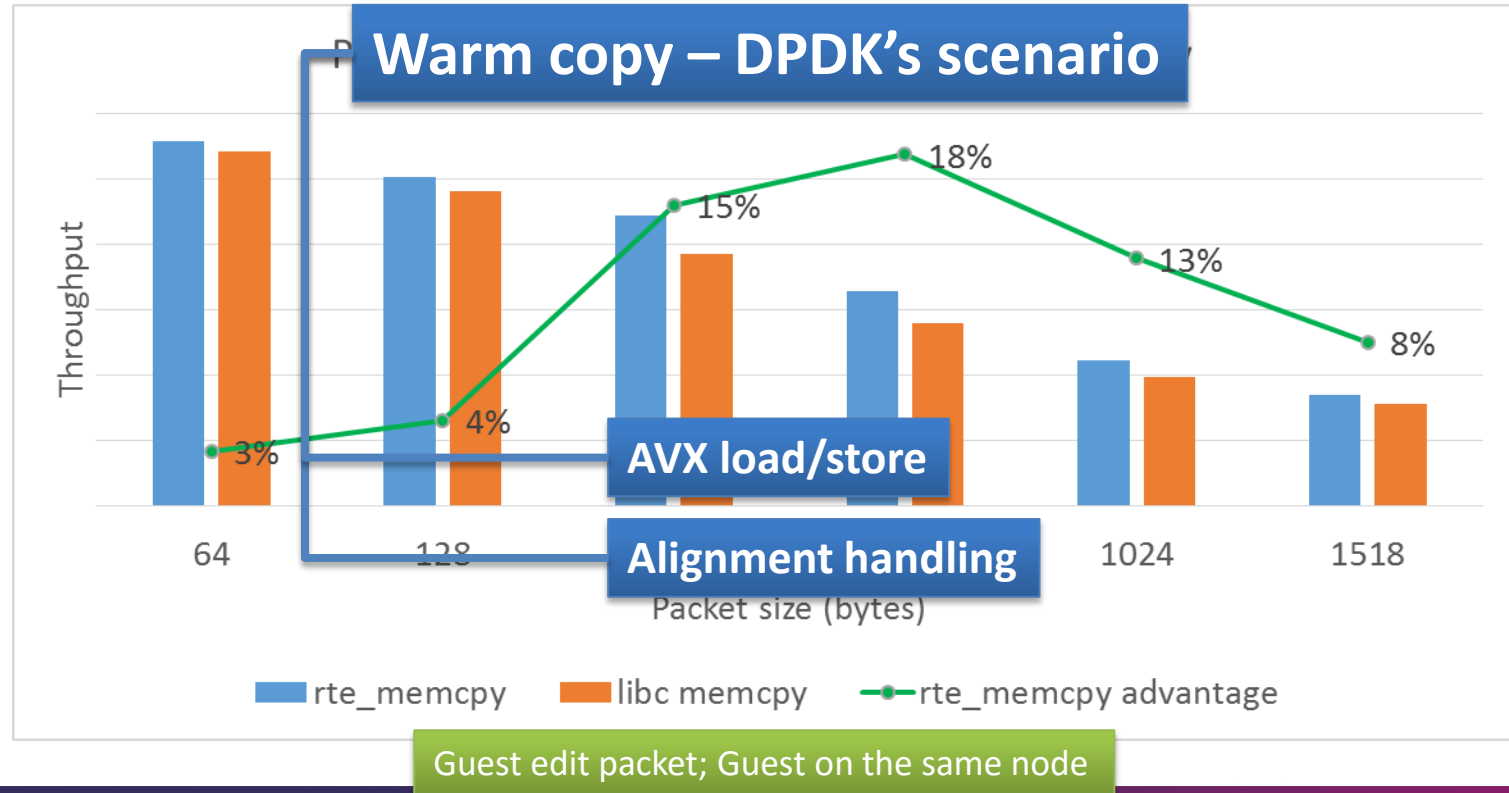
How About Guest In Another Node?



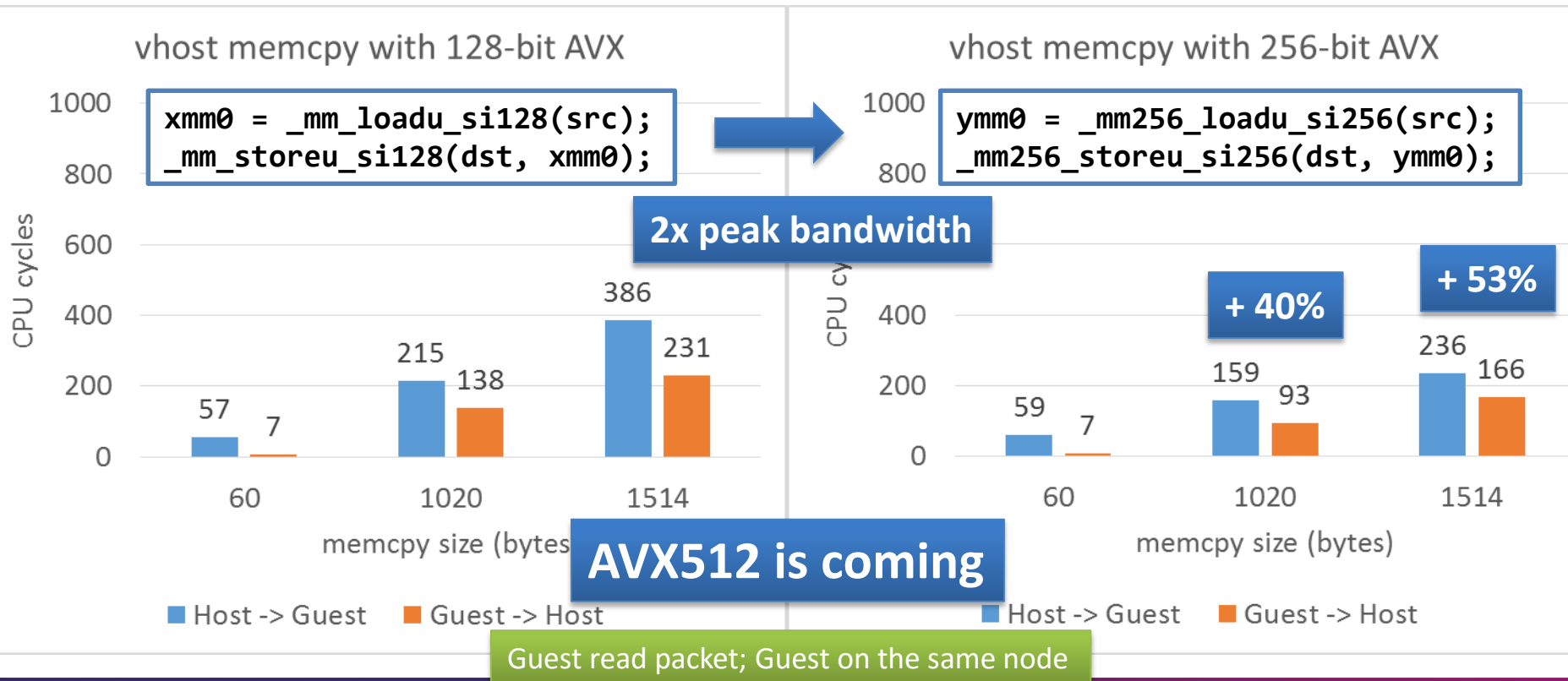
Better NOT...



rte_memcpy()? Why Even Bother?



AVX For Bandwidth



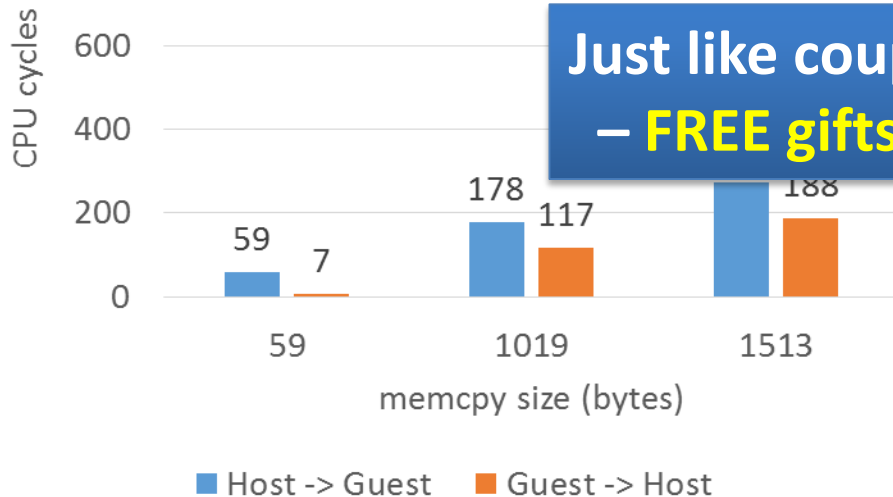
Alignment Matters

vhost unaligned memcpy

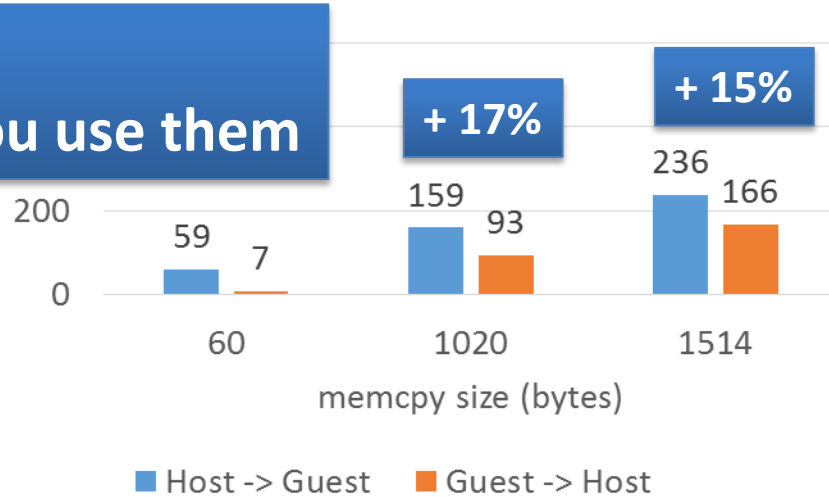
```
rte_memcpy((void *)((uint8_t *)dst + 1),  
src, len - 1);
```

vhost 64 bytes aligned memcpy

```
rte_memcpy(dst, src, len);
```



1000
800



Guest read packet; Guest on the same node

Takeaways

- See actual memory behaviors under the hood
 - Intel® 64 and IA-32 Architectures Optimization Reference Manual
- Benefit from new IA technologies
 - AVX, DDIO...



DPDK

DATA PLANE DEVELOPMENT KIT

Cache Allocation Technology

