ADRIAN PERRIG & TORSTEN HOEFLER

Networks and Operating Systems (252-0062-00)
Chapter 12: Reliable Storage & The Future

Our Small Quiz

- **True or false (raise hand)**
  - Receiver side scaling randomizes on a per-packet basis
  - Virtual machines can be used to improve application performance
  - Virtual machines can be used to consolidate servers
  - A hypervisor implements functions similar to a normal OS
  - If a CPU is strictly virtualizable, then OS code execution causes nearly no overheads
  - x86 is not strictly virtualizable because some instructions fail when executed in ring 1
  - x86 can be virtualized by binary rewriting
  - A virtualized host operating system can set the hardware PTBR
  - Paravirtualization does not require changes to the guest OS
  - A page fault with shadow page tables is faster than nested page tables
  - A page fault with writeable page tables is faster than shadow page tables
  - Shadow page tables are safer than writable page tables
  - Shadow page tables require paravirtualization

Memory allocation

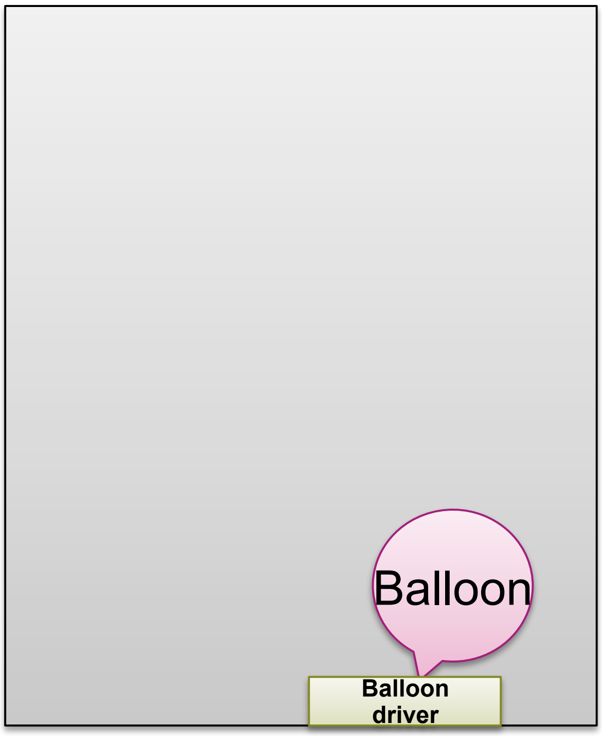
- **Guest OS is not expecting physical memory to change in size!**
- **Two problems:**
  - Hypervisor wants to overcommit RAM
  - How to reallocate (machine) memory between VMs
- **Phenomenon: Double Paging**
  - Hypervisor pages out memory
  - GuestOS decides to page out physical frame
  - (Unwittingly) faults it in via the Hypervisor, only to write it out again

Ballooning

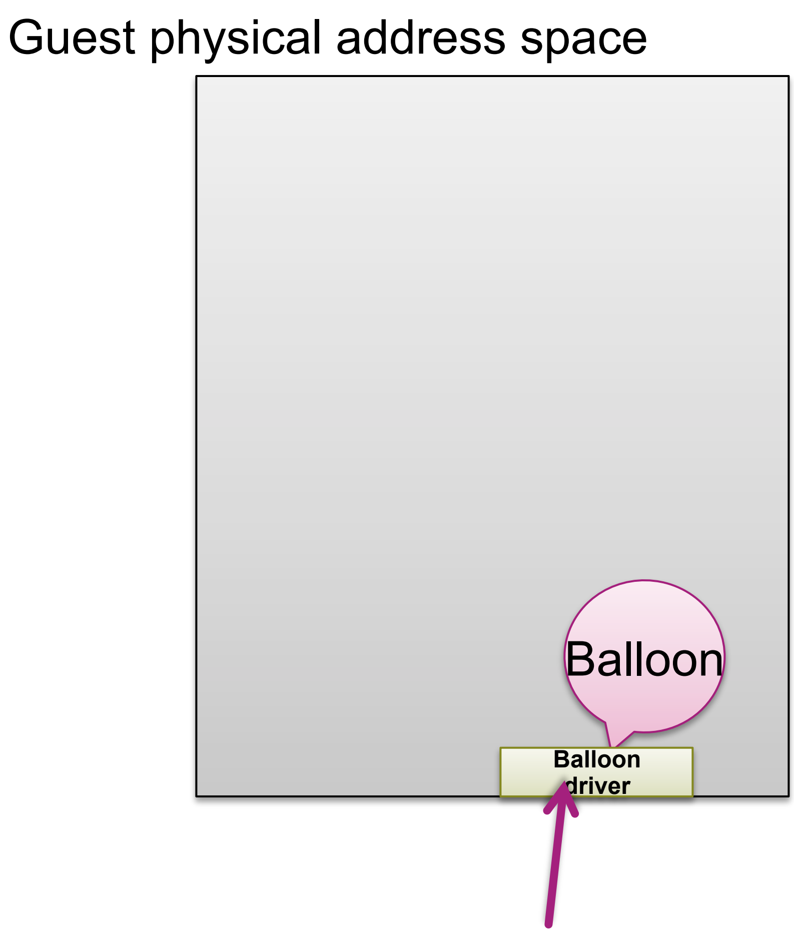
- **Technique to reclaim memory from a Guest**
- **Install a "balloon driver" in Guest kernel**
  - Can allocate and free kernel physical memory
    Just like any other part of the kernel
  - Uses HyperCalls to return frames to the Hypervisor, and have them returned
    Guest OS is unware, simply allocates physical memory

Ballooning: taking RAM away from a VM

Guest physical address space



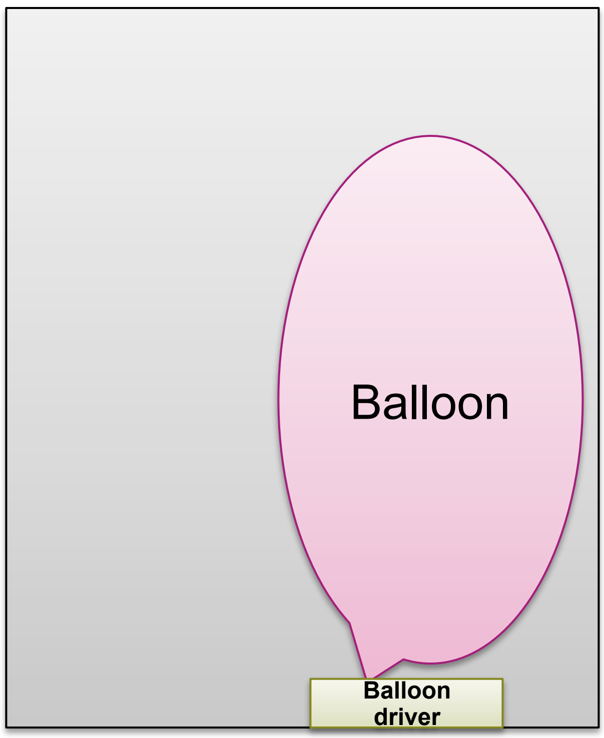
Ballooning: taking RAM away from a VM



1. **VMM asks balloon driver for memory**
2.
3.
4.

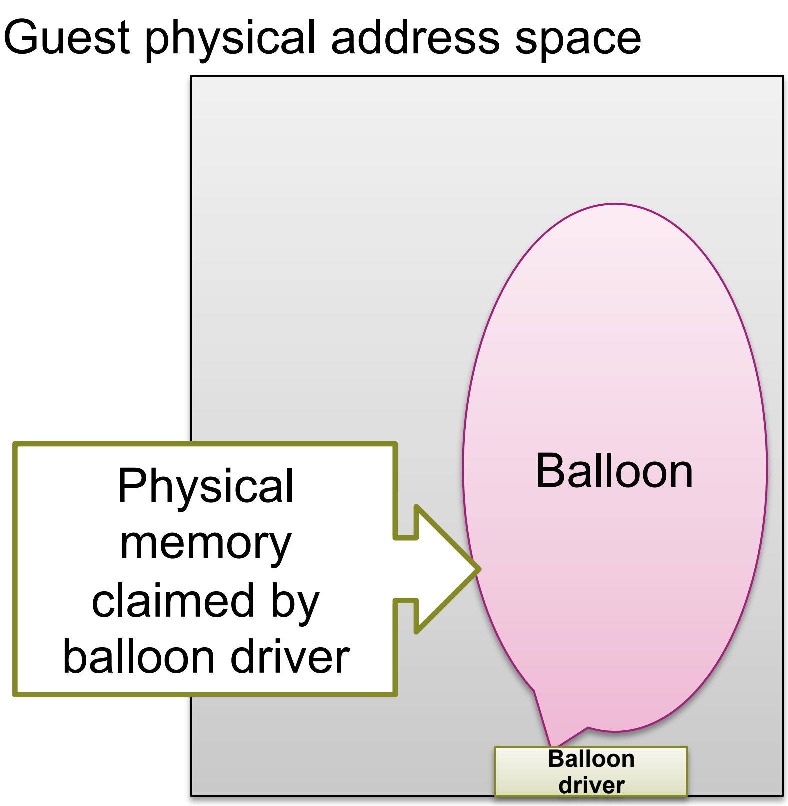
Ballooning: taking RAM away from a VM

Guest physical address space



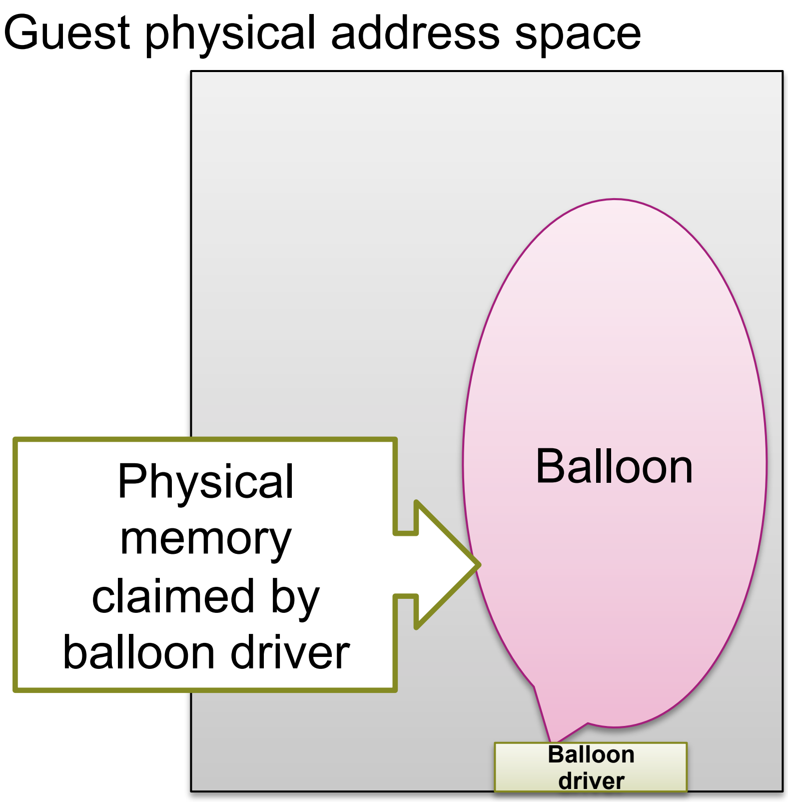
1. **VMM asks balloon driver for memory**
2. **Balloon driver asks Guest OS kernel for more frames**
   - "inflates the balloon"
3.
4.

Ballooning: taking RAM away from a VM



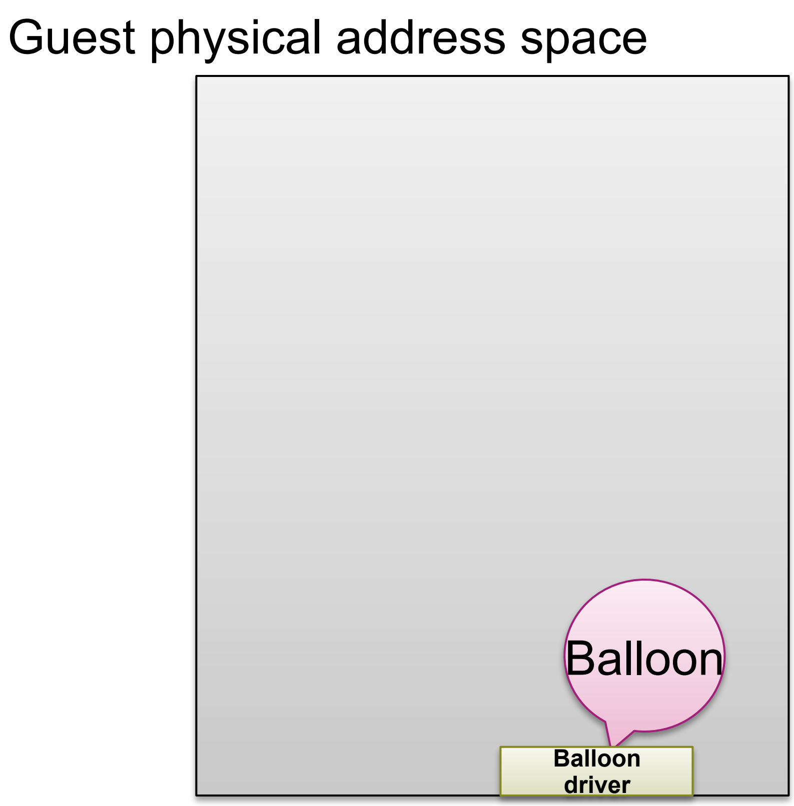
1. **VMM asks balloon driver for memory**
2. **Balloon driver asks Guest OS kernel for more frames**
   - "inflates the balloon"
3. **Balloon driver sends physical frame numbers to VMM**
4.

Ballooning: taking RAM away from a VM



1. **VMM asks balloon driver for memory**
2. **Balloon driver asks Guest OS kernel for more frames**
   - "inflates the balloon"
3. **Balloon driver sends physical frame numbers to VMM**
4. **VMM translates into machine addresses and claims the frames**

Returning RAM to a VM



1. **VMM converts machine address into a physical address previously allocated by the balloon driver**
2. **VMM hands PFN to balloon driver**
3. **Balloon driver frees physical frame back to Guest OS kernel**
   - "deflates the balloon"

Virtualizing Devices

- **Familiar by now: trap-and-emulate**
  - I/O space traps
  - Protect memory and trap
  - "Device model": software model of device in VMM
- **Interrupts → upcalls to Guest OS**
  - Emulate interrupt controller (APIC) in Guest
  - Emulate DMA with copy into Guest PAS
- **Significant performance overhead!**

Paravirtualized devices

- **“Fake” device drivers which communicate efficiently with VMM via hypercalls**
  - Used for block devices like disk controllers
  - Network interfaces
  - “VMware tools” is mostly about these

- **Dramatically better performance!**

Networking

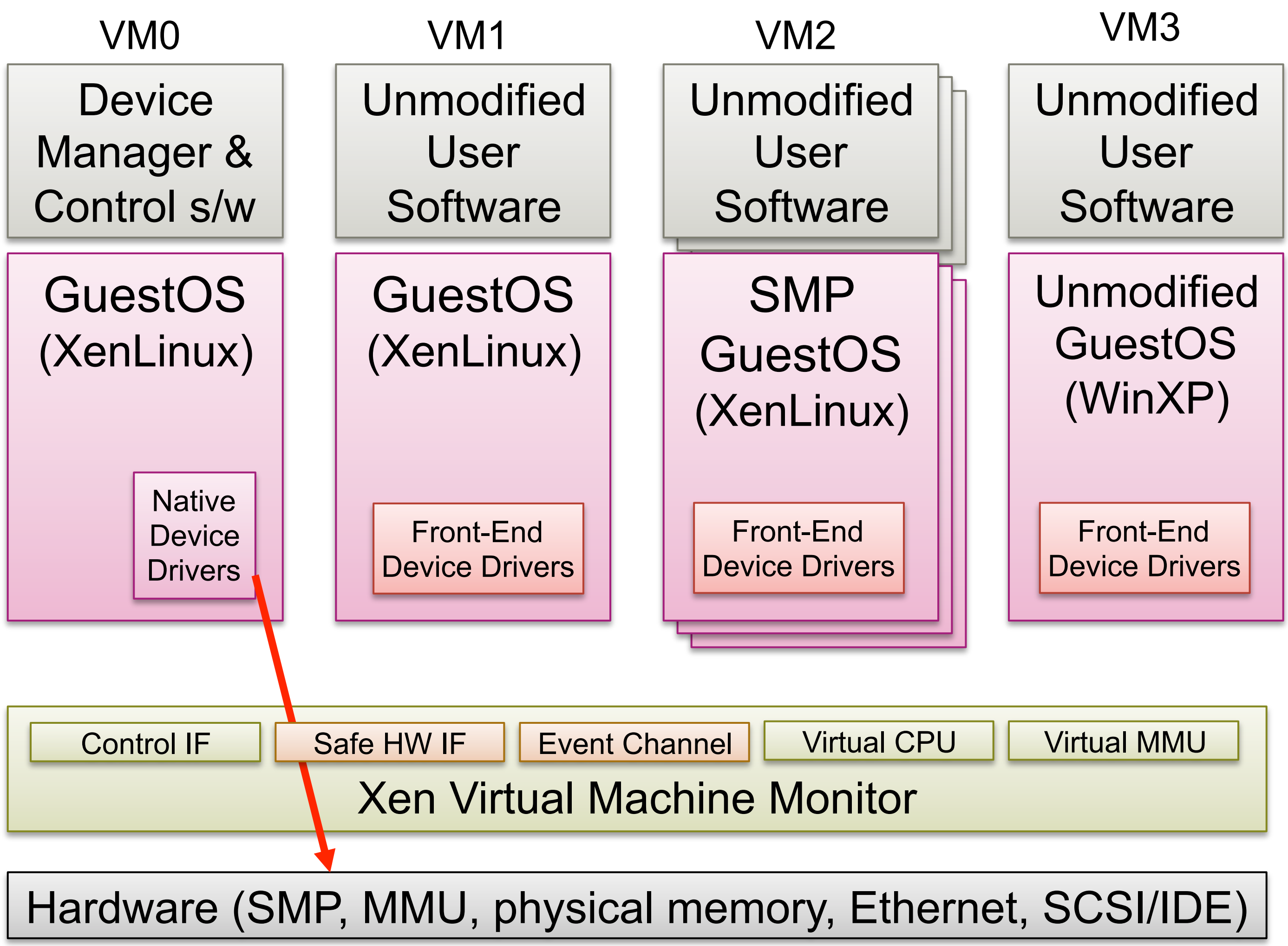
- **Virtual network device in the Guest VM**
- **Hypervisor implements a “soft switch”**
  - Entire virtual IP/Ethernet network on a machine
- **Many different addressing options**
  - Separate IP addresses
  - Separate MAC addresses
  - NAT
- **Etc.**

Where are the real drivers?

1. **In the Hypervisor**
   - E.g. VMware ESX
   - Problem: need to rewrite device drivers (new OS)
2. **In the console OS**
   - Export virtual devices to other VMs
3. **In "driver domains"**
   - Map hardware directly into a "trusted" VM

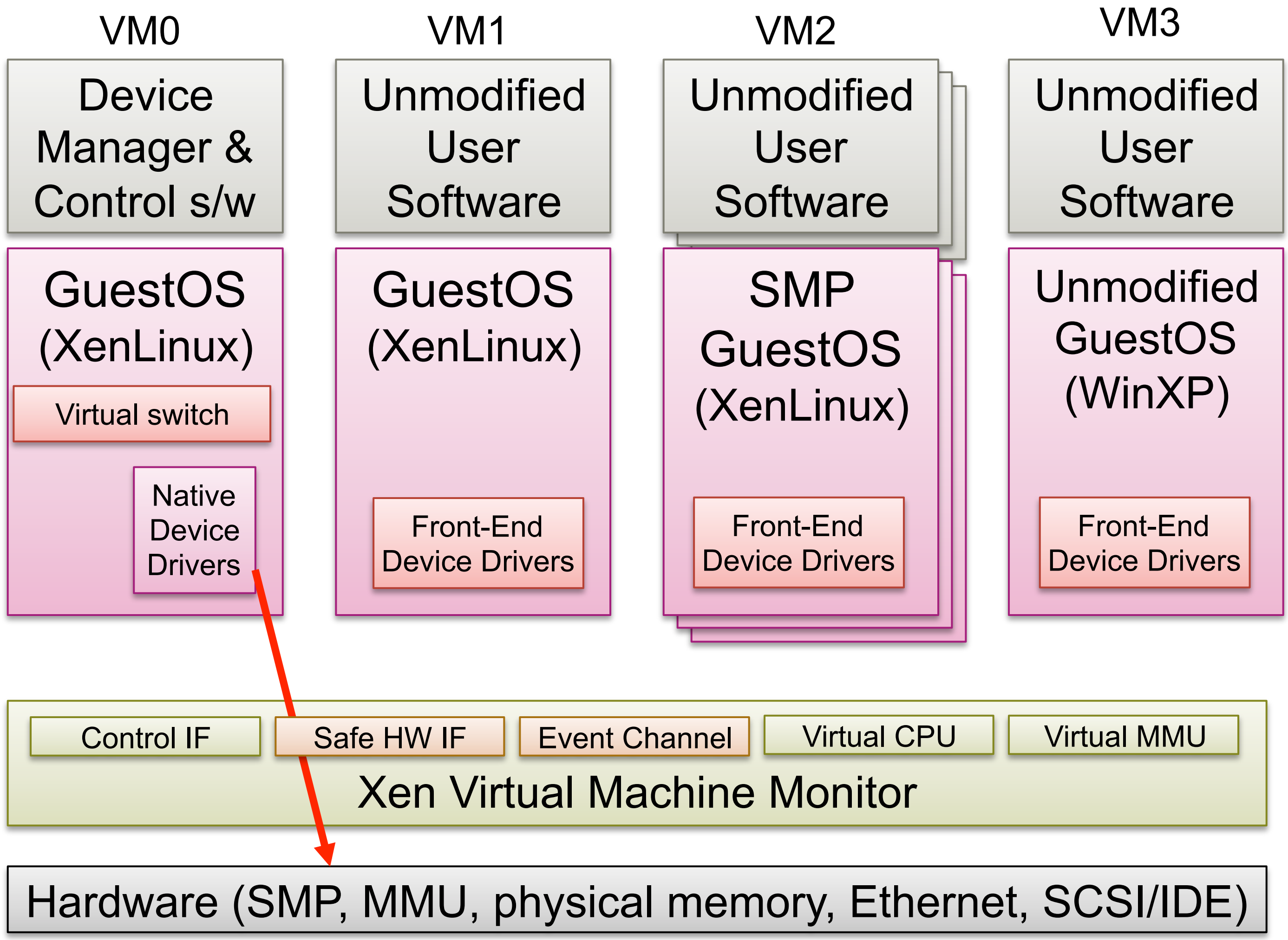
     Device Passthrough
   - Run your favorite OS just for the device driver
   - Use IOMMU hardware to protect other memory from driver VM
4. **Use "self-virtualizing devices"**

Xen 3.x Architecture



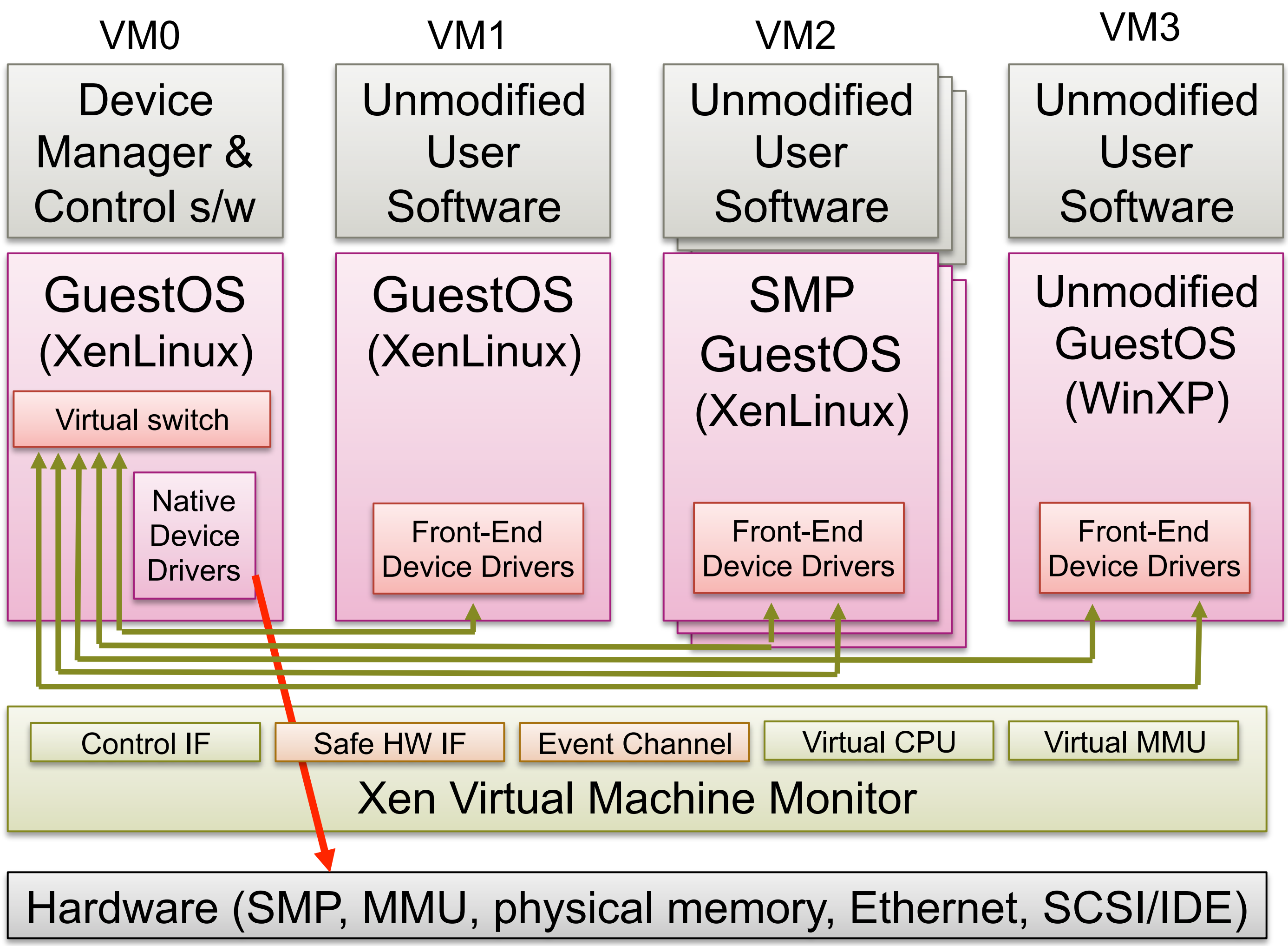
Thanks to Steve Hand for some of these diagra

Xen 3.x Architecture



Thanks to Steve Hand for some of these diagrams

Xen 3.x Architecture



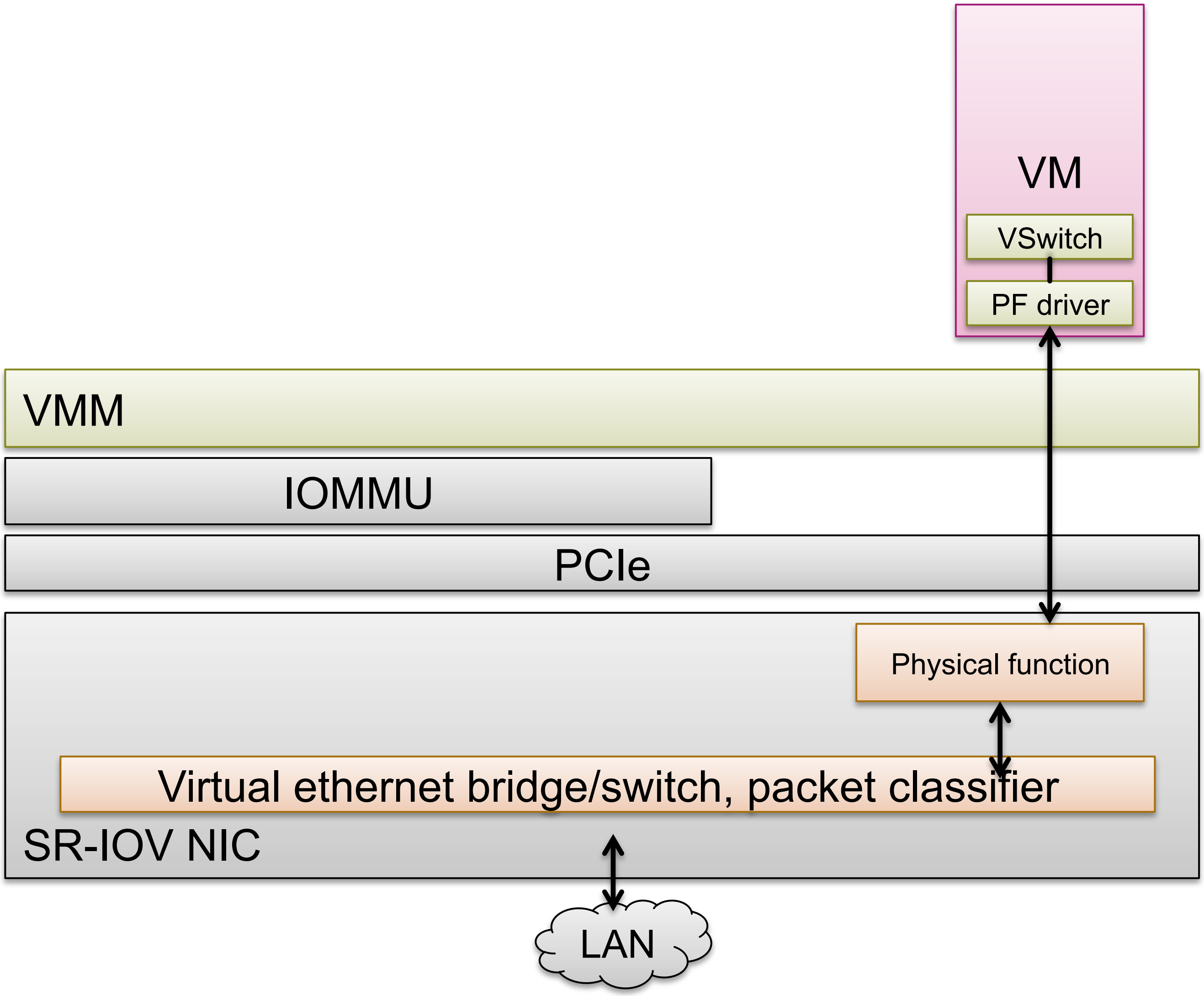
Thanks to Steve Hand for some of these diagrams

Remember this card?

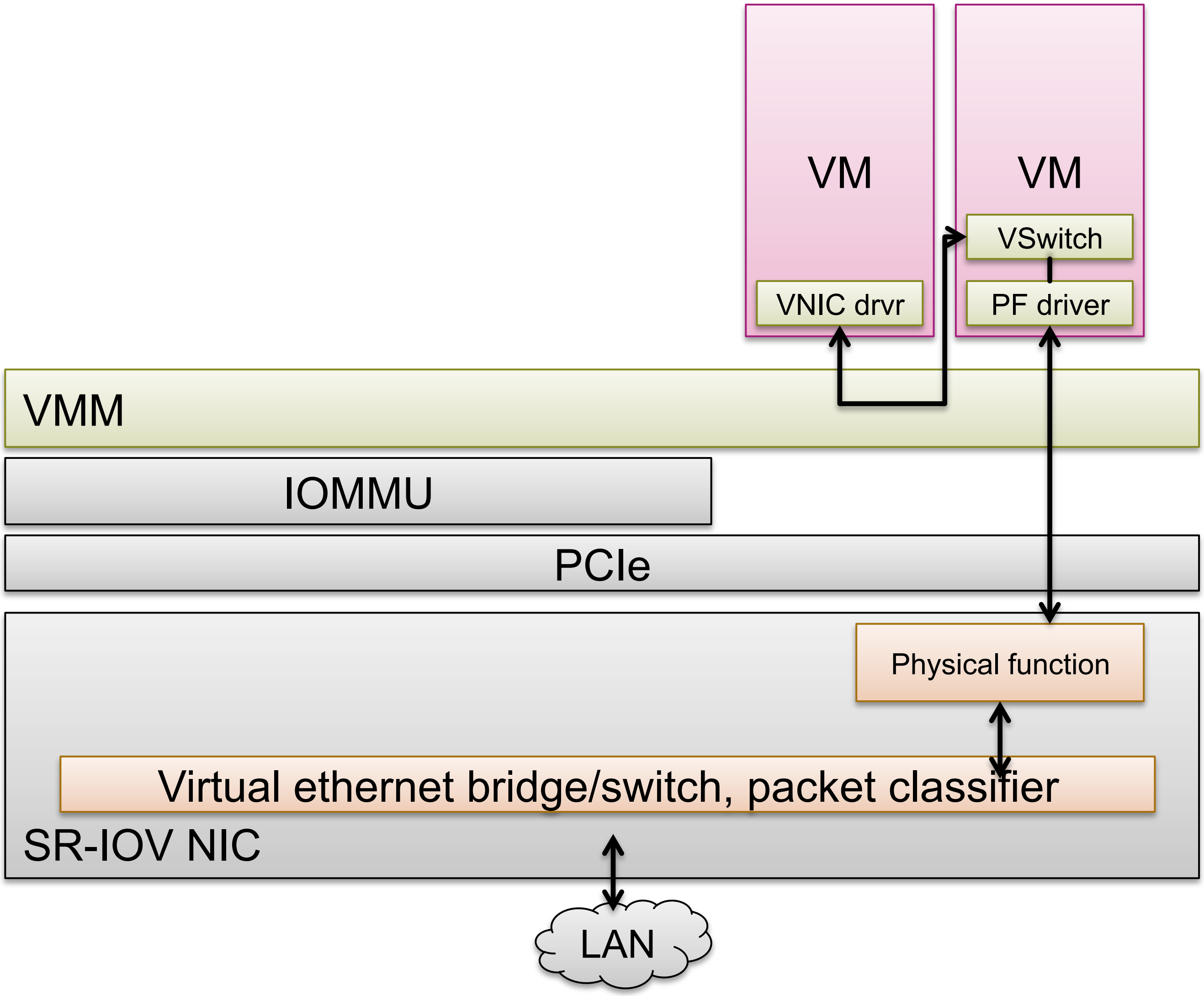
SR-IOV

- **Single-Root I/O Virtualization**
- **Key idea: dynamically create new "PCIe devices"**
  - Physical Function (PF): original device, full functionality
  - Virtual Function (VF): extra "device", limited funtionality
  - VFs created/destroyed via PF registers
- **For networking:**
  - Partitions a network card's resources
  - With direct assignment can implement passthrough

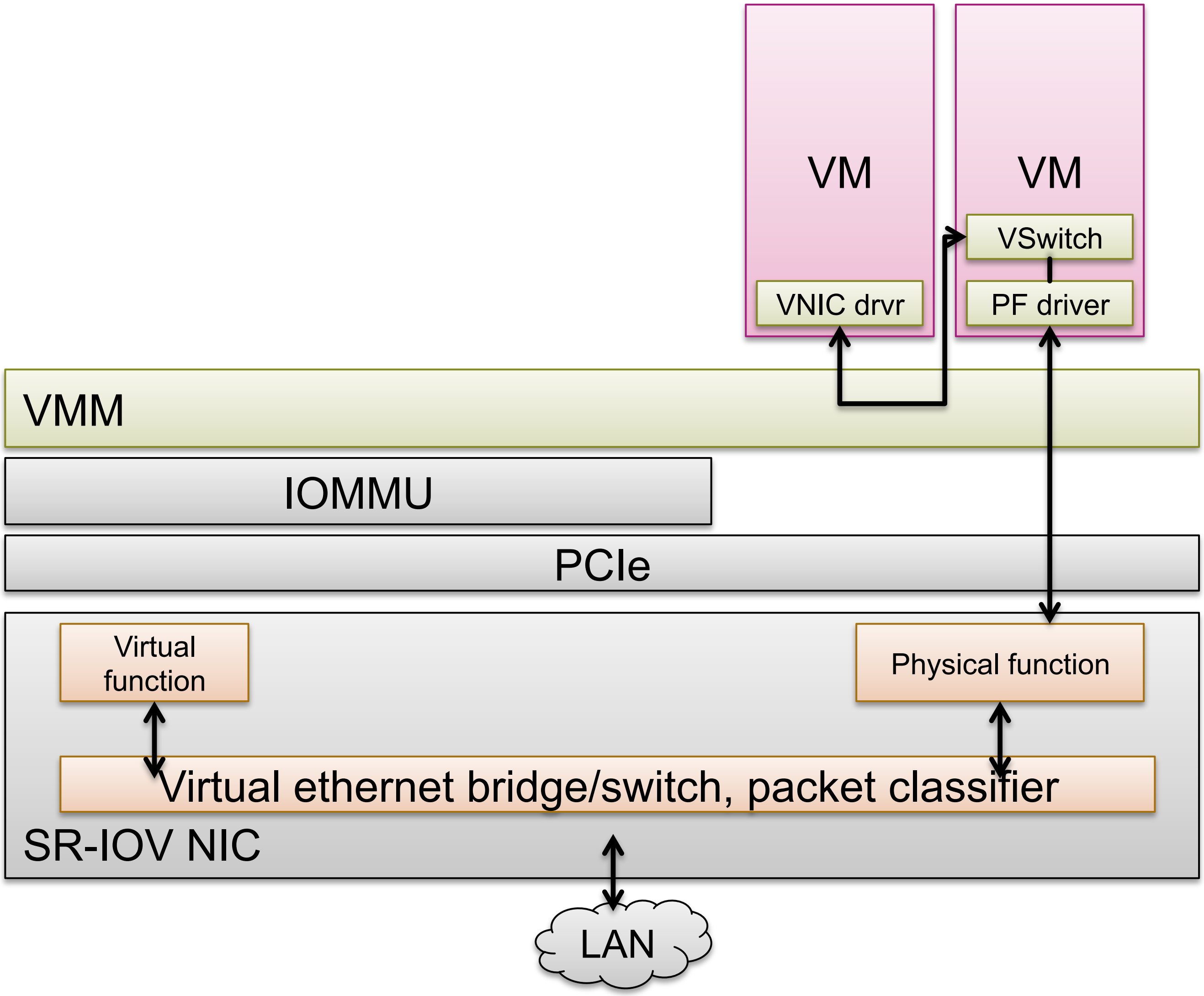
SR-IOV in action



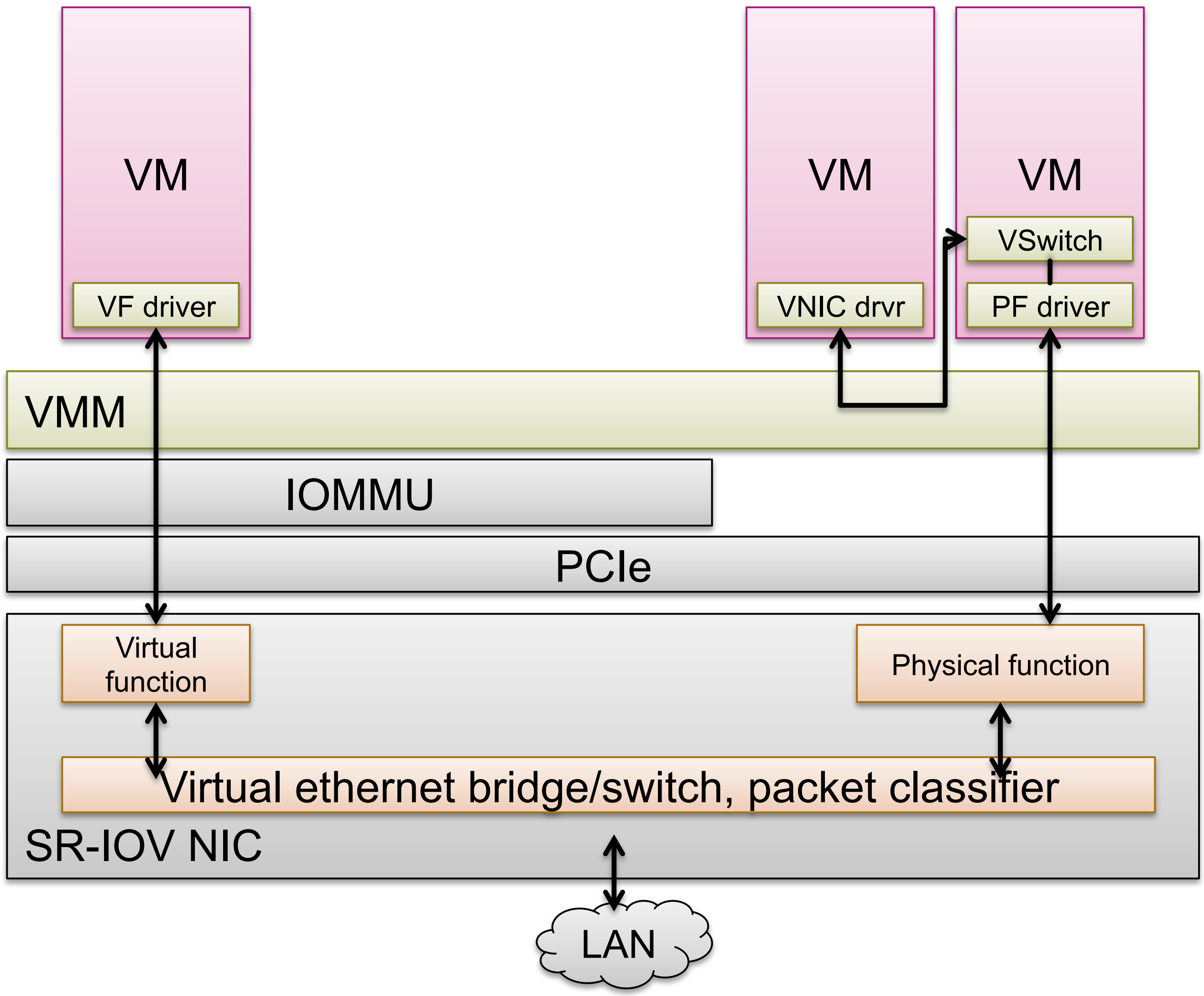
SR-IOV in action



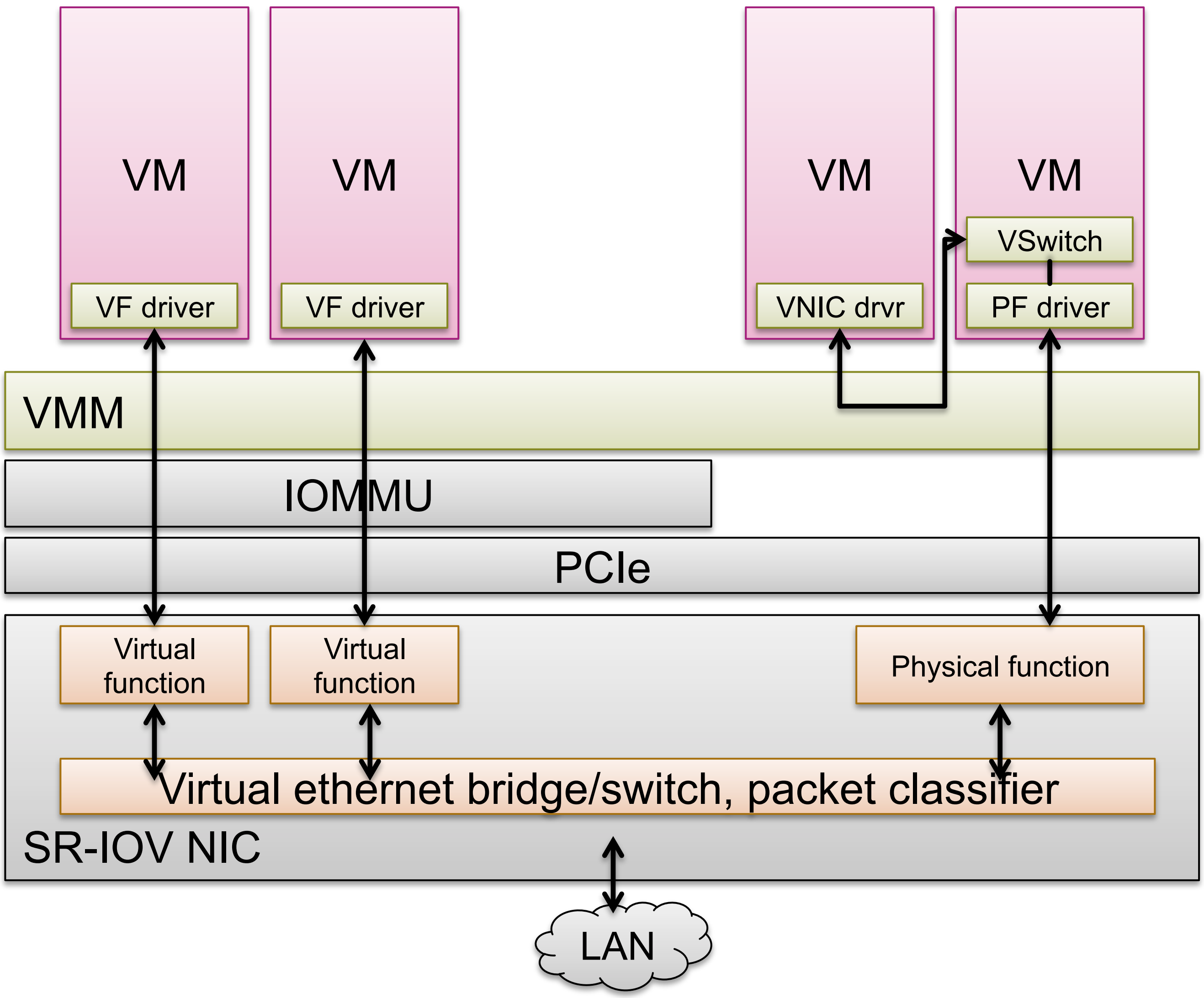
SR-IOV in action



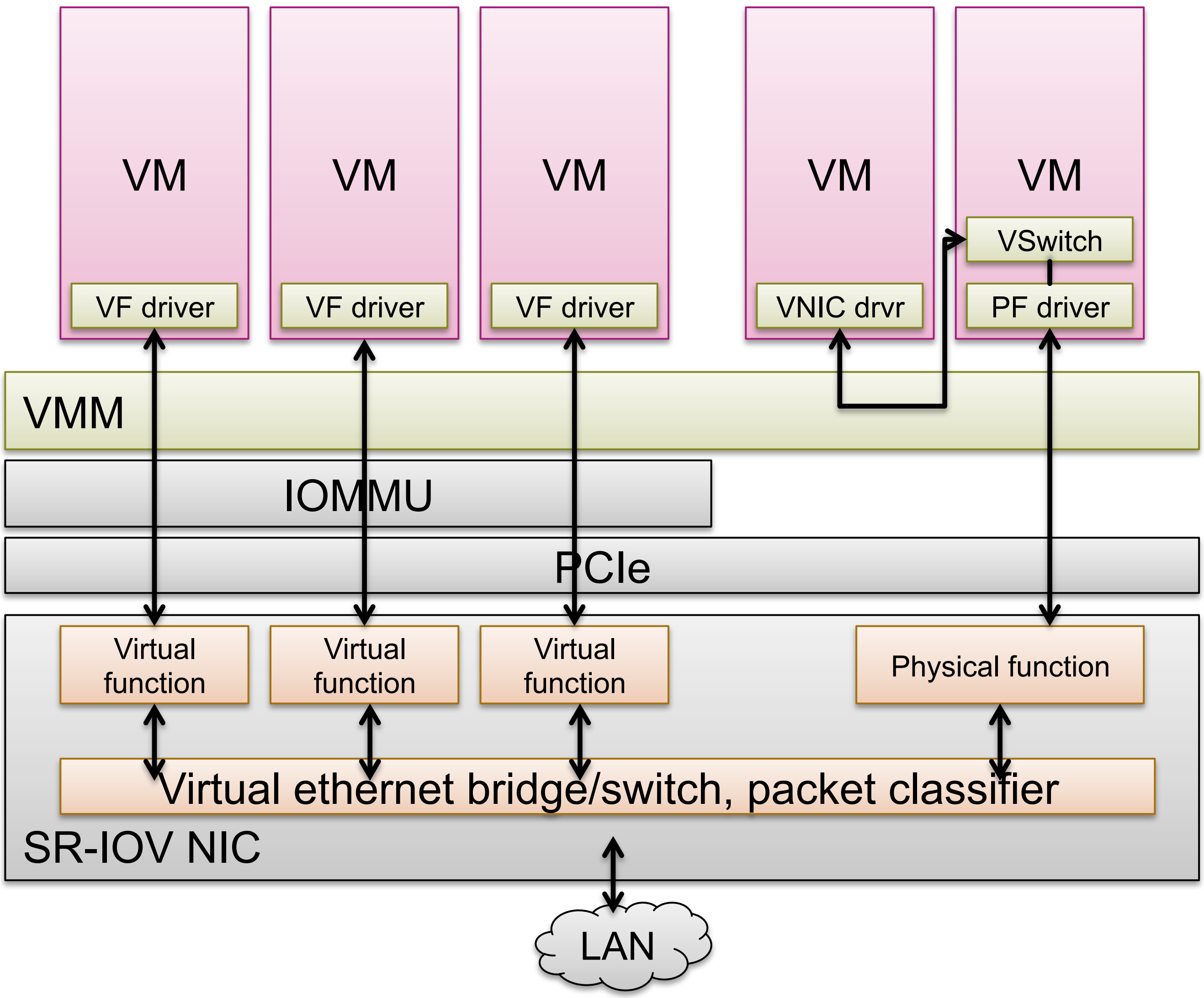
SR-IOV in action



SR-IOV in action



SR-IOV in action



Self-virtualizing devices

- **Can dynamically create up to 2048 distinct *PCI devices* on demand!**
  - Hypervisor can create a virtual NIC for each VM
  - Softswitch driver programs "master" NIC to demux packets to each virtual NIC
  - PCI bus is virtualized in each VM
  - Each Guest OS appears to have "real" NIC, talks direct to the real hardware

Reliable Storage

Reliability and Availabilty

A storage system is:

- ***Reliable*** **if it continues to store data and can read and write it.**
  ⇒ Reliability: probability it will be reliable for some period of time
- ***Available*** **if it responds to requests**
  ⇒ Availability: probability it is available at any given time

What goes wrong?

1. **Operating interruption: Crash, power failure**
   - Approach: use **transactions** to ensure data is consistent
   - Covered in the databases course
   - See book for additional material
2.

File system transactions

- **Not widely supported**
- **Only one atomic operation in POSIX:**
  - Rename
- **Careful design of file system data structures**
- **Recovery using fsck**
- **Superceded by transactions**
  - Internal to the file system
  - Exposed to applications

What goes wrong?

1. **Operating interruption: Crash, power failure**
   - Approach: use **transactions** to ensure data is consistent
   - Covered in the databases course
   - See book for additional material
2. **Loss of data: Media failure**
   - Approach: use **redundancy** to tolerate loss of media
   - E.g. RAID storage
   - Topic for today

Media failures 1: Sector and page failures

Disk keeps working, but a sector doesn't

- Sector writes don't work, reads are corrupted
- Page failure: the same for Flash memory

Approaches:

1. **Error correcting codes:**
   - Encode data with redundancy to recover from errors
   - Internally in the drive
2. **Remapping: identify bad sectors and avoid them**
   - Internally in the disk drive
   - Externally in the OS / file system

Caveats

- **Nonrecoverable error rates are significant**
  - And getting more so!
- **Nonrecoverable error rates are not constant**
  - Affected by age, workload, etc.
- **Failures are not independent**
  - Correlation in time and space
- **Error rates are not uniform**
  - Different models of disk have different behavior over time

A well-respected disk available now from pcp.ch

Seagate Barracuda 3TB,
7200rpm, 64MB, 3TB, SATA-3

Price this weekend: CHF 105.-
(last year CHF 150,-)

Specifications (from manufacturer's website)

Seagate

Persistent errors that are *not* masked by coding inside the drive

Specifications	3TB[1]	2TB[1]
Model Number	ST33000651AS	ST32000641AS
Interface Options	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ
Performance		
Transfer Rate, Max Ext (MB/s)	600	600
Max Sustained Data Rate OD (MB/s)	149	138
Cache (MB)	64	64
Average Latency (ms)	4.16	4.16
Spindle Speed (RPM)	7200	7200
Configuration/Organization		
Heads/Disks	10/5	8/4
Bytes per Sector	512	512
Reliability/Data Integrity		
Load/Unload Cycles	300K	300K
Nonrecoverable Read Errors per Bits Read, Max	1 per 10E14	1 per 10E14
Annualized Failure Rate (AFR)	0.34%	0.34%
Mean Time Between Failures (hours)	750,000	750,000
Limited Warranty (years)	5	5
Power Management		
Startup Current +12 Peak (A, ±10%)	2.0	2.8

Unrecoverable read errors

- What's the chance we could read a *full* 3TB disk without errors?
- For each bit:

$$\Pr(success) = 1 - 10^{-14}$$

- Whole disk:

$$\Pr(success) = (1 - 10^{-14})^{8 \times 3 \times 10^{12}}$$
$$\approx \mathbf{0.7868}$$

- Feeling lucky?

Lots of assumptions: Independent errors, etc.

Media failures 2: Device failure

- **Entire disk (or SSD) just stops working**
  - Note: always detected by the OS
  - Explicit failure ⇒ less redundancy required
- **Expressed as:**
  - Mean Time to Failure (MTTF)
    (expected time before disk fails)
  - Annual Failure Rate = 1/MTTF
    (fraction of disks failing in a year)

Specifications (from manufacturer's website)

Seagate®

Specifications	3TB[1]	2TB[1]
Model Number	ST33000651AS	ST32000641AS
Interface Options	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ
Performance		
Transfer Rate, Max Ext (MB/s)	600	600
Max Sustained Data Rate OD (MB/s)	149	138
Cache (MB)	64	64
Average Latency (ms)	4.16	4.16
Spindle Speed (RPM)	7200	7200
Configuration/Organization		
Heads/Disks	10/5	8/4
Bytes per Sector	512	512
Reliability/Data Integrity		
Load/Unload Cycles	300K	300K
Nonrecoverable Read Errors per Bits Read, Max	1 per 10E14	1 per 10E14
Annualized Failure Rate (AFR)	0.34%	0.34%
Mean Time Between Failures (hours)	750,000	750,000
Limited Warranty (years)	5	5
Power Management		
Startup Current +12 Peak (A, ±10%)	2.0	2.8

Caveats

- **Advertised failure rates can be misleading**
  - Depend on conditions, tests, definitions of failure…
- **Failures are not uncorrelated**
  - Disks of similar age, close together in a rack, etc.
- **MTTF is not useful life!**
  - Annual failure rate only applies during design life!
- **Failure rates are not constant**
  - Devices fail very quickly or last a long time

And Reality?

Appears in the Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST'07), February 2007

Failure Trends in a Large Disk Drive Population

(S.M.A.R.T – Self-Monitoring, Analysis, and Reporting Technology)

Eduardo Pinheiro, Wolf-Dietrich Weber and Luiz André Barroso
Google Inc.
1600 Amphitheatre Pkwy
Mountain View, CA 94043
{edpin,wolf,luiz}@google.com

Abstract

It is estimated that over 90% of all new information produced in the world is being stored on magnetic media, most of it on hard disk drives. Despite their importance, there is relatively little published work on the failure patterns of disk drives, and the key factors that affect their lifetime. Most available data are either based on extrapolation from accelerated aging experiments or from relatively modest sized field studies. Moreover, larger population studies rarely have the infrastructure in place to collect health signals from components in operation, which is critical information for detailed failure analysis.

We present data collected from detailed observations of a large disk drive population in a production Internet services deployment. The population observed is many times larger than that of previous studies. In addition to presenting failure statistics, we analyze the correlation between failures and several parameters generally believed to impact longevity.

for guiding the design of
vising deployment and m

Despite the importanc
few published studies on
drives. Most of the avai
the disk manufacturers t
typically based on extra
test data of small popu
databases. Accelerated li
viding insight into how s
affect disk drive lifetime
predictors of actual failu
in the field [7]. Statistic
cally based on much larger populations, but since there is little or no visibility into the deployment characteristics, the analysis lacks valuable insight into what actually happened to the drive during operation. In addition,

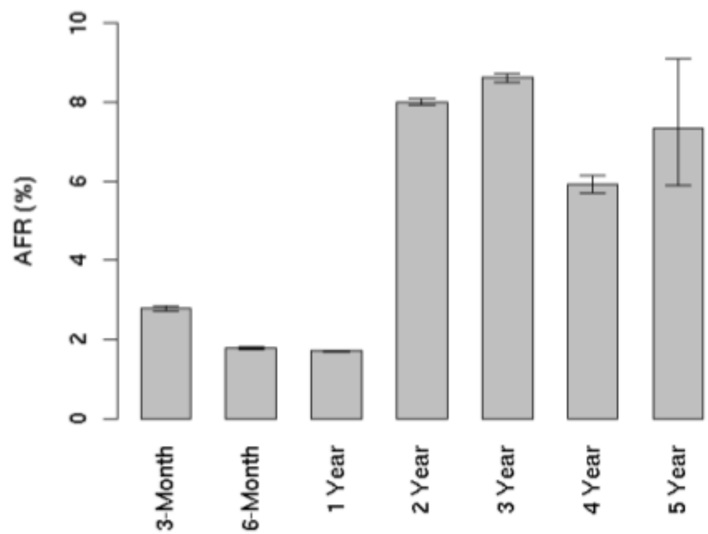
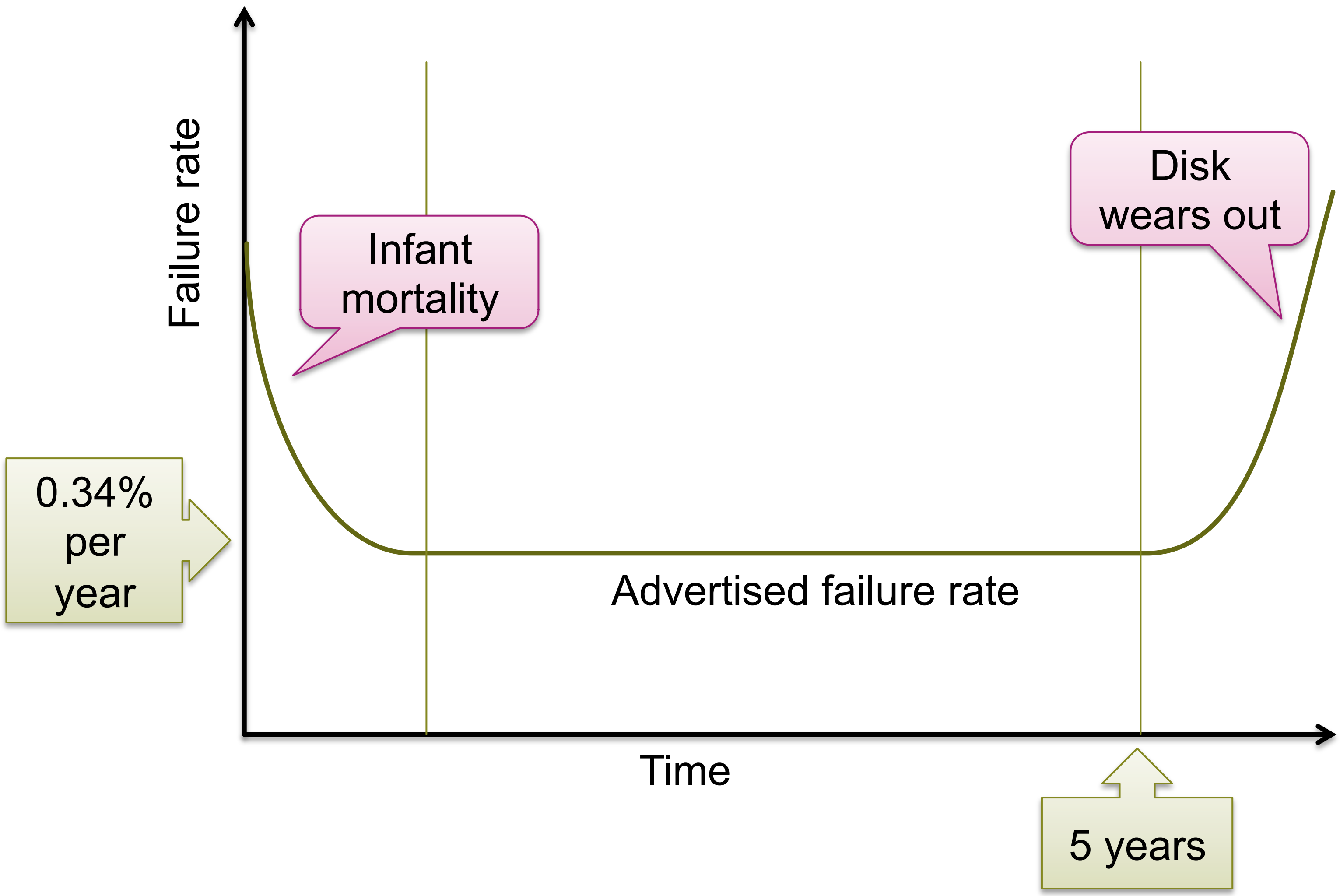
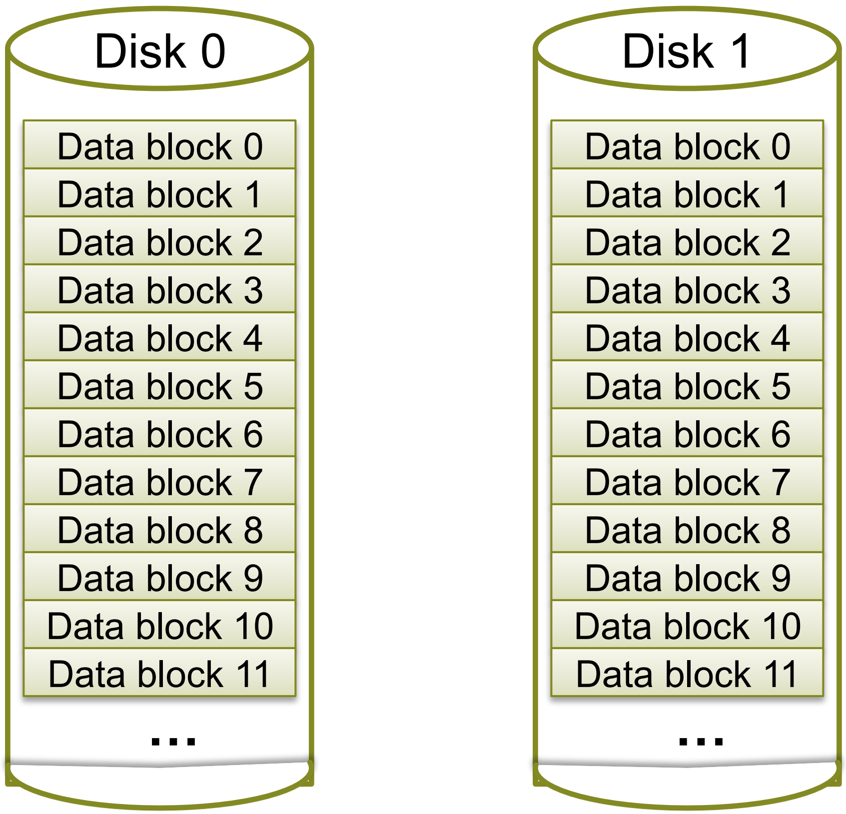


Figure 2: Annualized failure rates broken down by age groups

Bathtub curve



RAID 1: simple mirroring

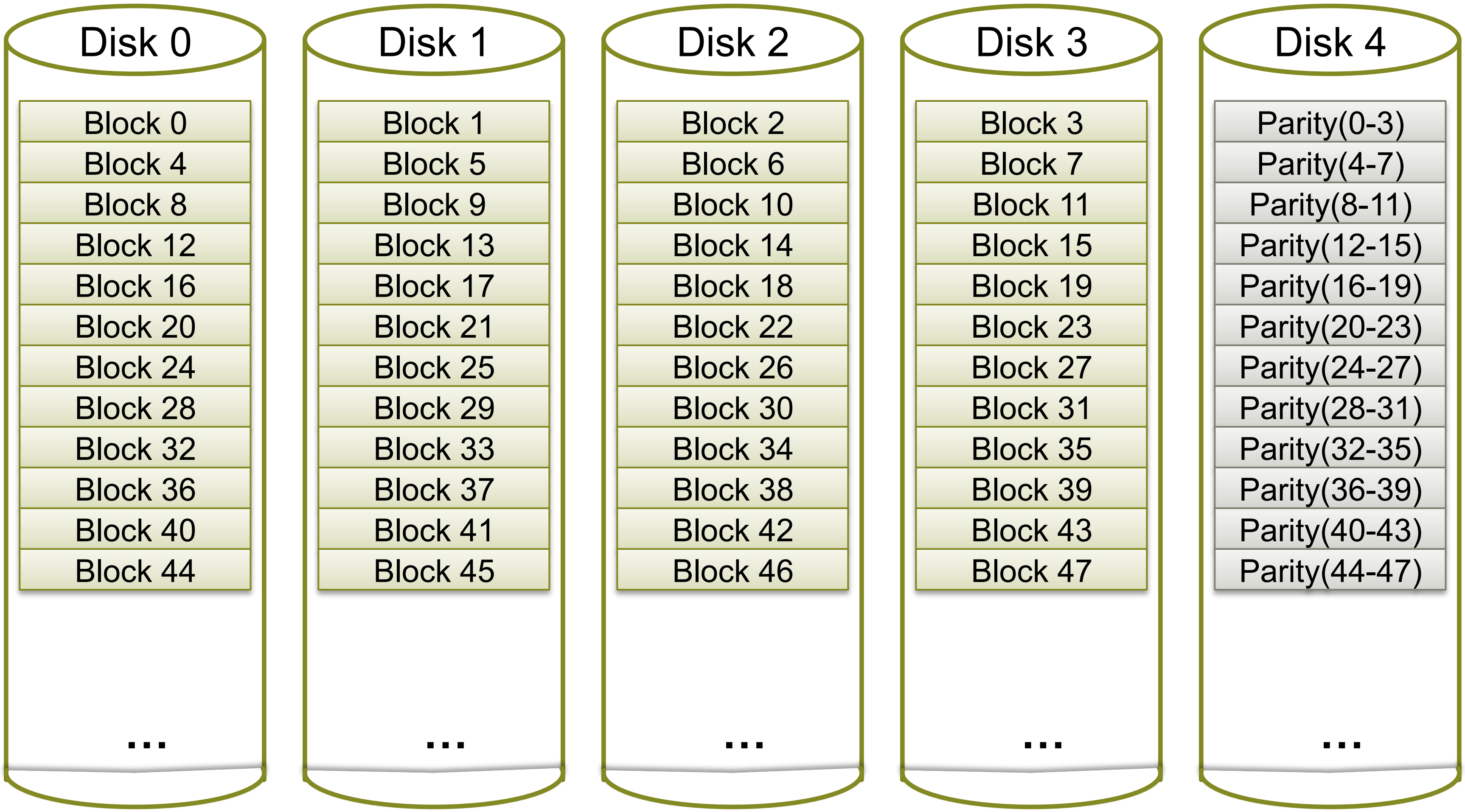


Writes go to both disks

Reads from either disk (may be faster)

Sector or whole disk failure ⇒ data can still be recovered

Parity disks and striping



Parity disks

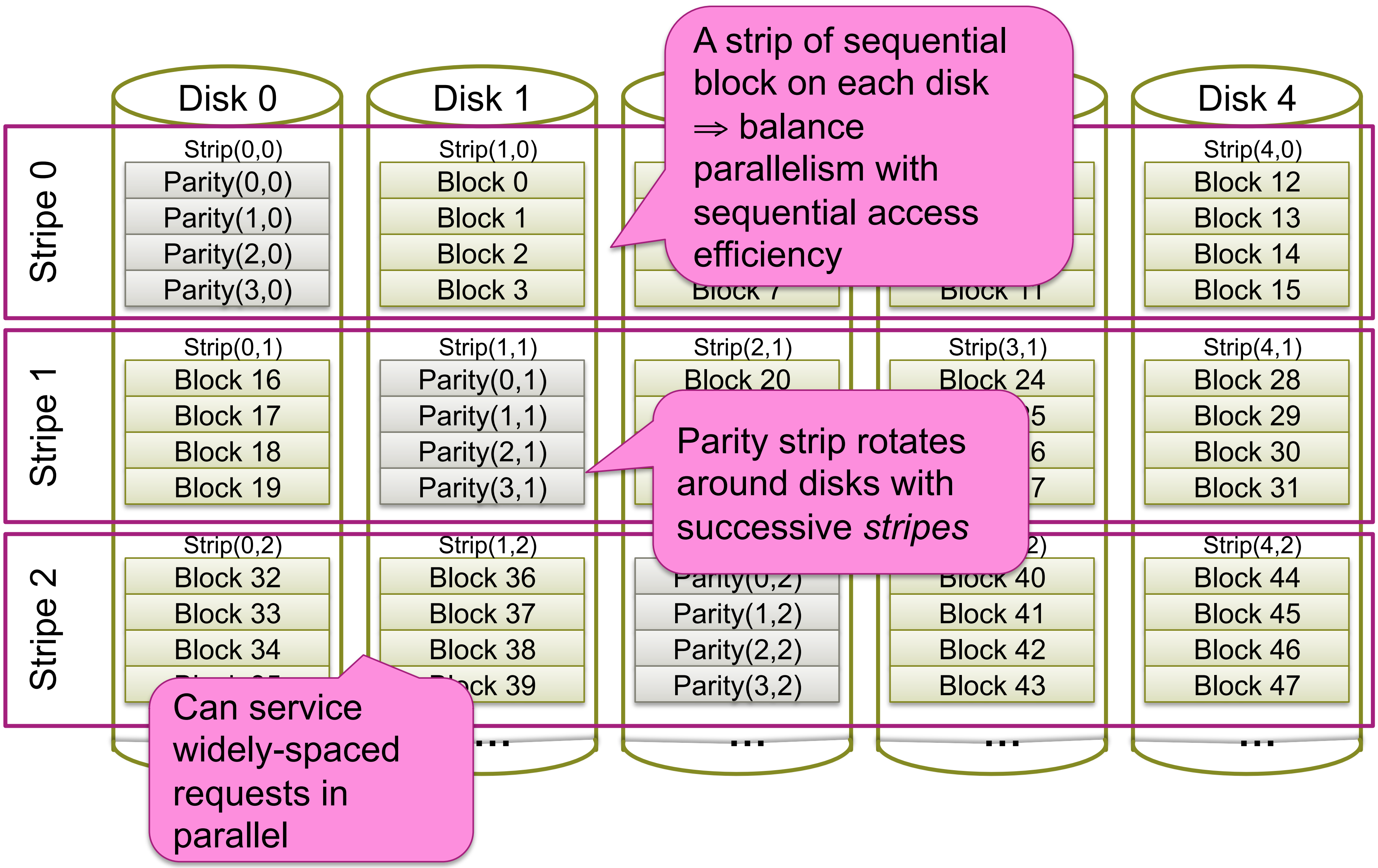
- Note: errors are always detected
$\Rightarrow$ Parity allows errors to be corrected

- Write *d'* to block $\Rightarrow$ must also update parity, e.g.
  - Read *d* from block, parity block, then:
$$parity' = parity \oplus n' \oplus n$$
  - Write *d'* to block *n, parity'* to parity block

High overhead for small writes

- Problem: with 5 disks, parity disk is accessed 4 times as often on average!

RAID5: Rotating parity



Atomic update of data and parity

What if system crashes in the middle?

1. **Use non-volatile write buffer**
2. **Transactional update to blocks**
3. **Recovery scan**
   - And hope nothing goes wrong during the scan
4. **Do nothing (seriously)**

Recovery

- **Unrecoverable read error on a sector:**
  - Remap bad sector
  - Reconstruct contents from stripe and parity
- **Whole disk failure:**
  - Replace disk
  - Reconstruct data from the other disks
  - Hope nothing else goes wrong…

Mean time to repair (MTTR)

RAID-5 can lose data in three ways:

1. **Two full disk failures**
   (second while the first is recovering)
2. **Full disk failure and sector failure on another disk**
3. **Overlapping sector failures on two disks**

- **MTTR: Mean time to repair**
  - Expected time from disk failure to when new disk is fully rewritten, often hours
- **MTTDL: Mean time to data loss**
  - Expected time until 1, 2 or 3 happens

Analysis

See the book for *independent* failures

- **Key result: most likely scenario is #2.**

Solutions:

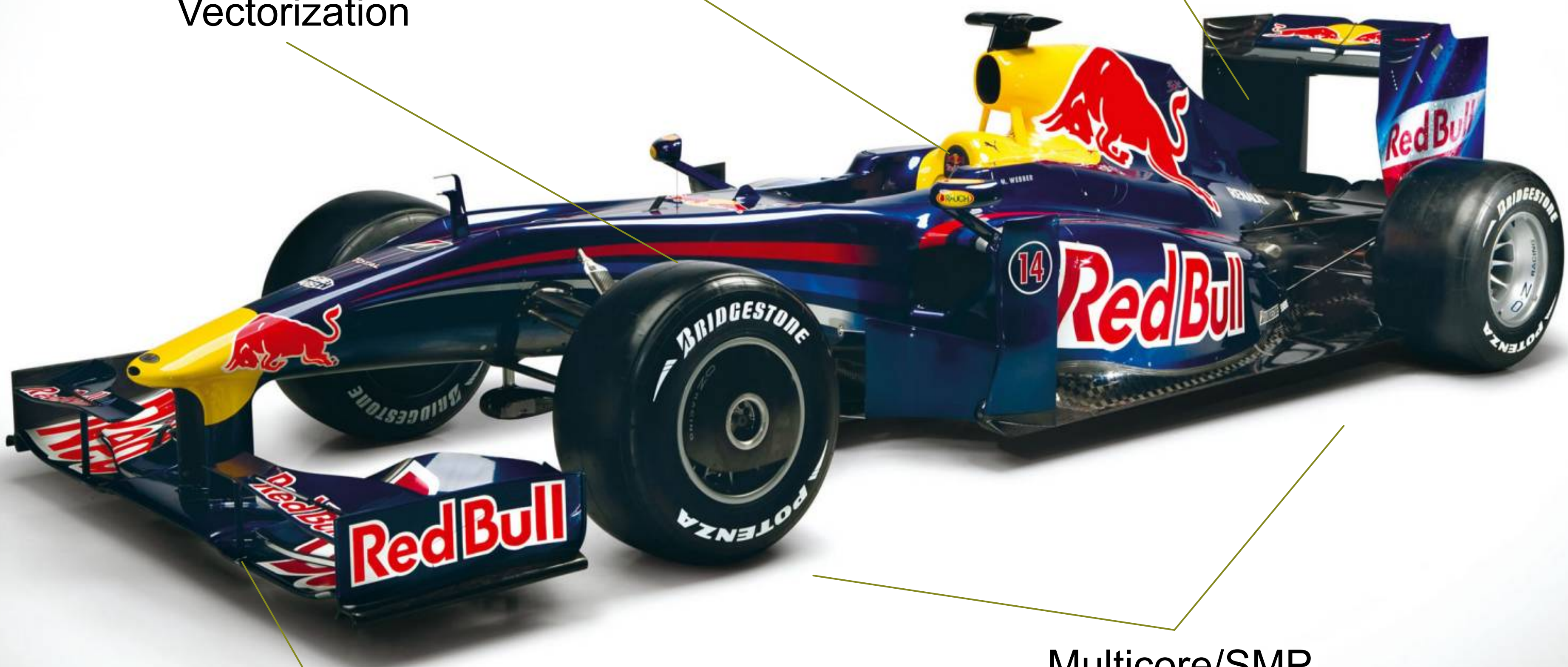
1. **More redundant disks, erasure coding**
2. **Scrubbing**
   - Regularly read the whole disk to catch UREs early
3. **Buy more expensive disks.**
   - I.e. disks with much lower error rates
4. **Hot spares**
   - Reduce time to plug/unplug disk

The Future™

What's happening to hardware?

- **Lots of cores (scaling, parallelism)**
- **Lots of different cores**
- **Complex memory hierarchies and interconnects**
- **Increasing diversity of machines**
  - Hardware is changing faster than system software can
- **Faster devices (especially networks)**
- **…**

Supercomputing



….

Top 500

- **A benchmark, solve Ax=b**
  - As fast as possible! → as big as possible ☺
  - Reflects **some** applications, not all, not even many
  - Very good historic data!
- **Speed comparison for computing centers, states, countries, nations, continents ☹**
  - Politicized (sometimes good, sometimes bad)
  - Yet, fun to watch

The November 2013 List

Rank	Site	System	Cores	(TFlop/s)	(TFlop/s)	(kW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3120000	33862.7	54902.4	17808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560640	17590.0	27112.5	8209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1572864	17173.2	20132.7	7890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705024	10510.0	11280.4	12660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786432	8586.6	10066.3	3945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115984	6271.0	7788.9	2325
7	Texas Advanced Computing Center/Univ. of Texas	**Stampede** - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz,	462462	5168.1	8520.1	4510

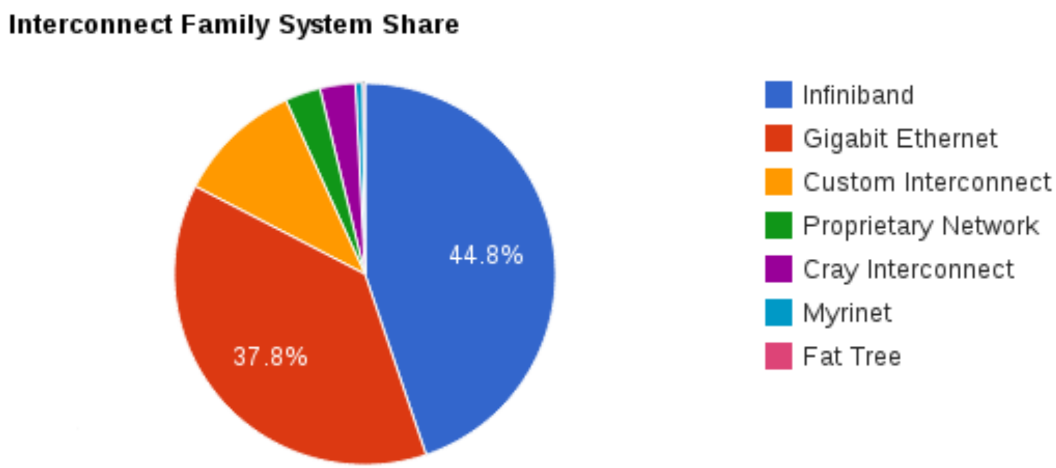
IDC, 2009: *"expects the HPC technical server market to grow at a healthy 7% to 8% yearly rate to reach revenues of $13.4 billion by 2015."*

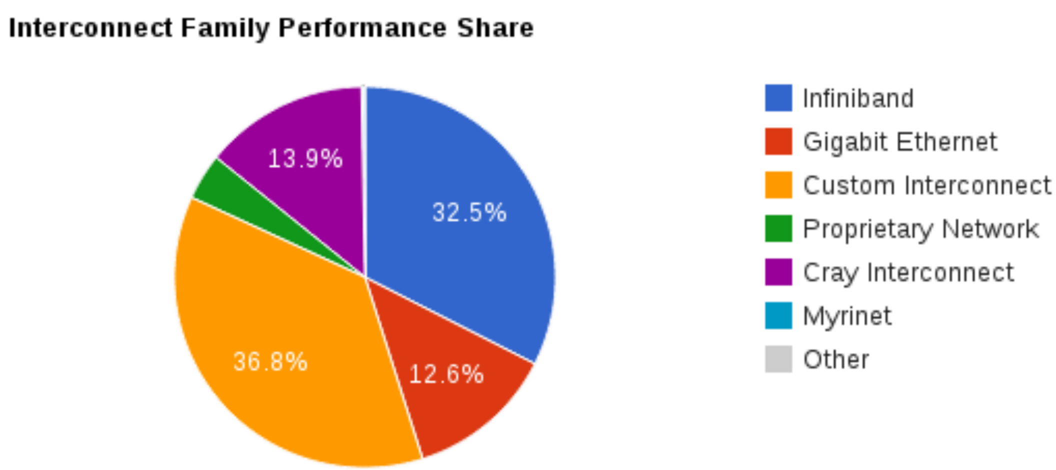
"The non-HPC portion of the server market was actually down 20.5 per cent, to $34.6bn"

www.top500.org

How to communicate?

- **Communication is key in problem solving ☺**
  - Not just relationships!
  - Also scientific computations





Source: top500.org

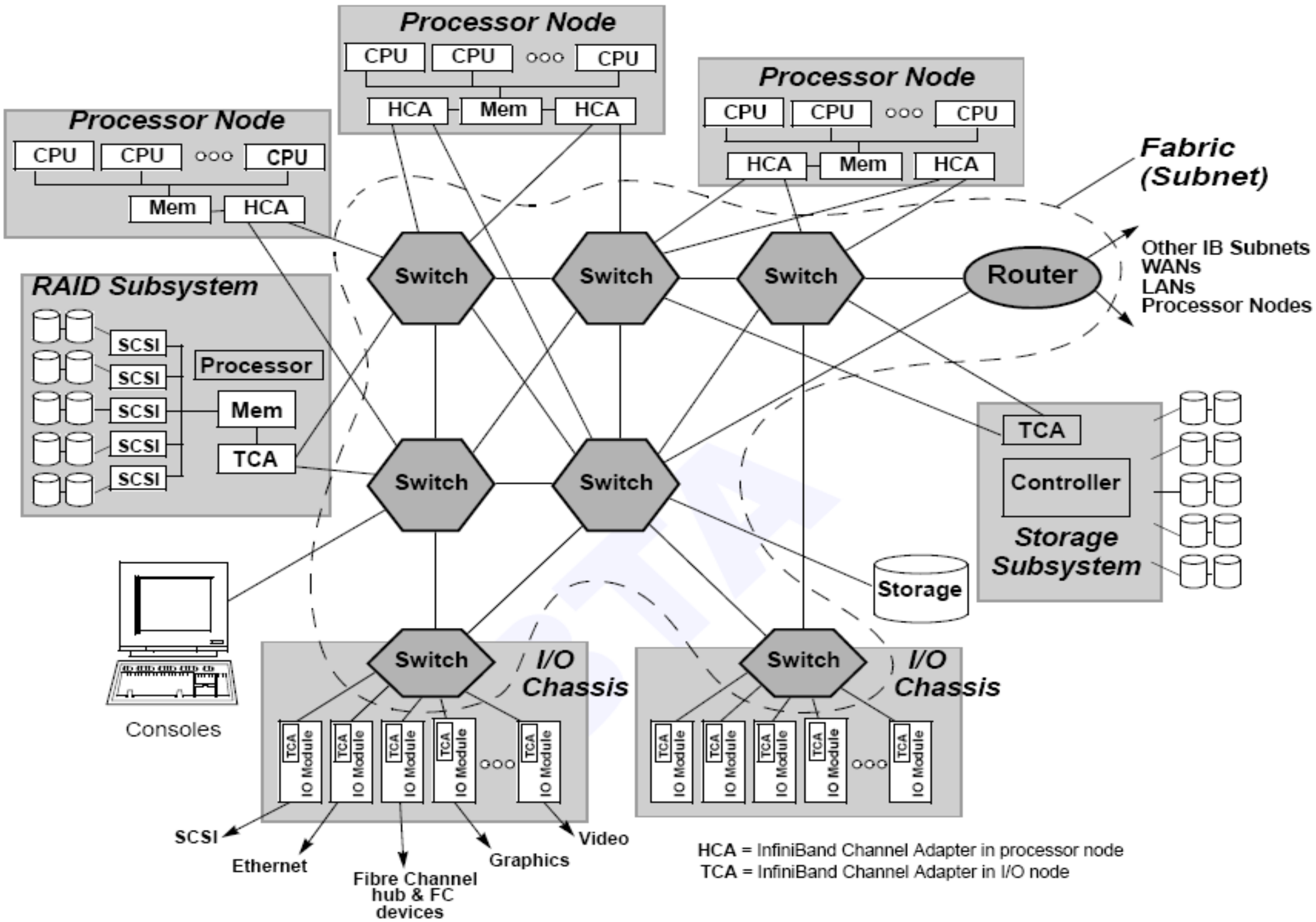
Remote Direct Memory Access

- **Remember that guy?**
  - 2x2x40 Gb/s → ~20 GB/s
  - Memory bandwidth: ~60 GB/s
  - 1.5 copies ☹
- **Solution:**
  - RDMA, similar to DMA
  - OS too expensive, bypass
  - Communication offloading

InfiniBand Overview

- **Components:**
  - Links/Channel adaptors
  - Switches/Routers
- **Routing is supported but rarely used, most IB networks are "LANs"**
- **Supports arbitrary topologies**
  - "Typical" topologies: fat tree, torus, islands
- **Link speed (all 4x):**
  - Single data rate (SDR): 10 Gb/s
  - Double data rate (DDR): 20 Gb/s
  - Quad data rate (QDR): 40 Gb/s
  - Fourteen data rate (FDR): 56 Gb/s
  - Enhanced data rate (EDR): 102 Gb/s

InfiniBand Network Structure



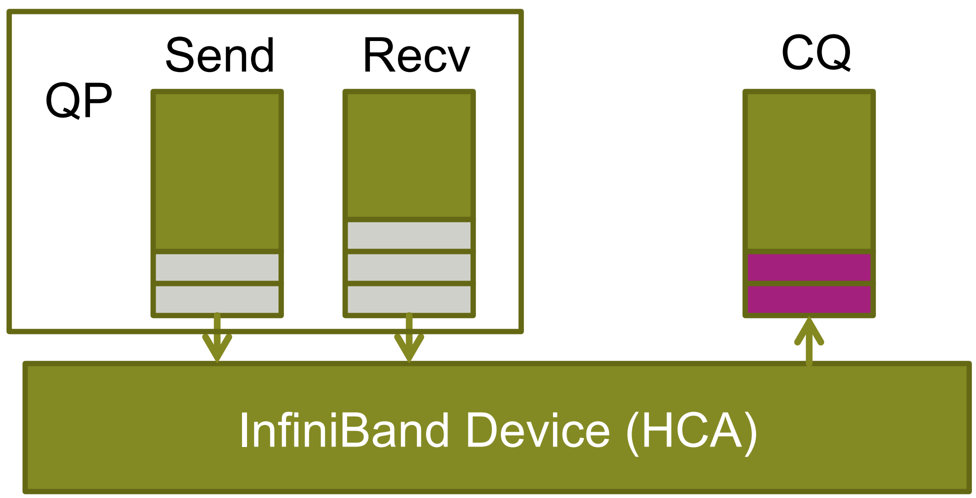
Source: IBA Spe

InfiniBand Subnet Routing

- ***No spanning tree protocol, allows parallel links&loops, initialization phases:***
  - *Topology discovery:* discovery MADs
  - *Path computation*: MinHop, …, DFSSSP
  - *Path distribution phase*: Configure routing tables
- **Problem: how to generate paths?**
  - MinHop == OSPF
  - Potentially bad bandwidth allocation!

Interaction with IB HCAs

- **Systems calls only for setup:**
  - Establish connection, register memory
- **Communication (send/recv, put, get, atomics) all in user-level!**
  - Through “verbs” interface



Open Fabrics Stack

- **OFED offers a unified programming interface**
  - Cf. Sockets
  - Originated in IB verbs
  - Direct interaction with device
  - Direct memory exposure
    Requires page pinning (avoid OS interference)
- **Device offers**
  - User-level driver interface
  - Memory-mapped registers

iWARP and RoCE

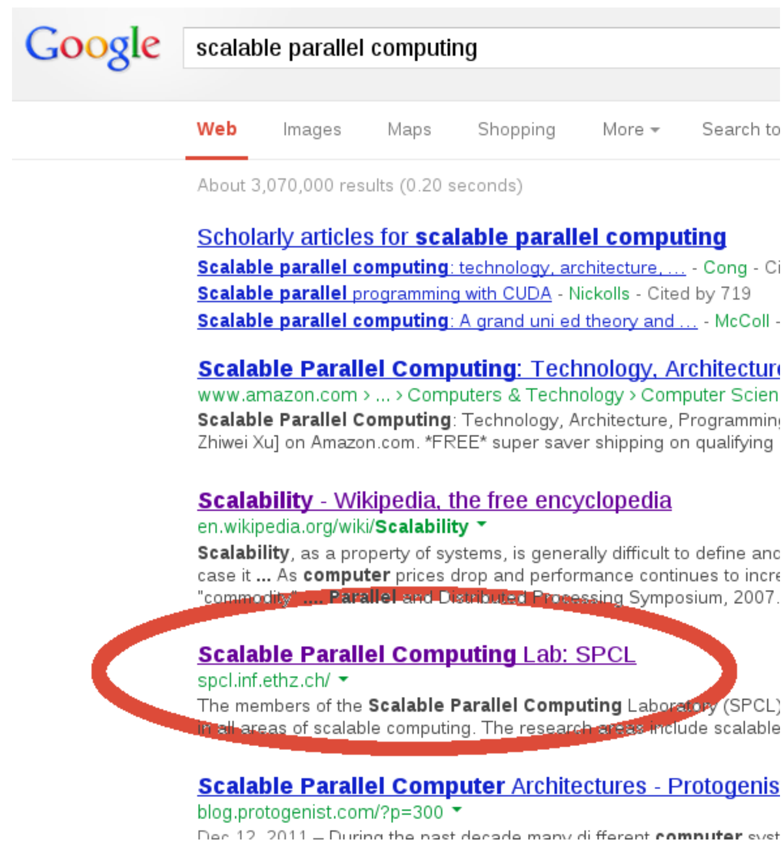
- **iWARP: RDMA over TCP/IP**
  - Ups:
    Routable with existing infrastructure
    Easily portable (filtering, etc.)
  - Downs:
    Higher latency (complex TOE)
    Higher complexity in NIC
    TCP/IP is not designed for datacenter networks
- **RoCE: RDMA over Converged Ethernet**
  - Data-center Ethernet!

Student Cluster Competition

- **5 undergrads, 1 advisor, 1 cluster, 2x13 amps**
  - 8 teams, 4 continents @SC13
  - 48 hours, five applications, non-stop!
  - top-class conference
- **Lots of fun**
  - Even more experience!
- **A Swiss team 2015?**
  - Search for "Student Cluster Challenge"
  - HPC-CH may help

Finito

- **Thanks for being such fun to teach ☺**
  - Comments (also anonymous) are always appreciated!
- **If you are interested in parallel computing research, talk to me!**
  - Large-scale (datacenter) systems
  - Parallel computing (SMP and MPI)
  - GPUs (CUDA and stuff)
  - … on twitter: @spcl_eth ☺



Thanks to Timothy Roscoe for many slides!