Outrunning Moore's Law

Can IP-SANs close the host-network gap?

Jeff Chase
Duke University

But first….

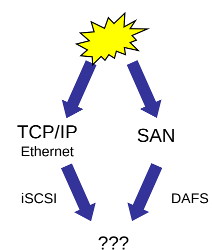
- This work addresses questions that are important in the industry right now.
- It is an outgrowth of Trapeze project: 1996-2000.
- It is tangential to my primary research agenda.
  - Resource management for large-scale shared service infrastructure.
  - Self-managing computing/storage utilities
  - Internet service economy
  - Federated distributed systems
  - Amin Vahdat will speak about our work on Secure Highly Available Resource Peering (SHARP) in a few weeks.

A brief history

- Much research on fast communication and end-system TCP/IP performance through 1980s and early 1990s.
- Common theme: advanced NIC features and host/NIC boundary.
  - TCP/IP offload controversial: early efforts failed
  - User-level messaging and Remote Direct Memory Access or RDMA (e.g., unet)
- SAN market grows enormously in mid-1990s
  - VI Architecture standardizes SAN messaging host interface in 1997-1998.
  - FibreChannel (FC) creates market for network block storage.
- Then came Gigabit Ethernet…

A brief history, part 2

- "Zero-copy" TCP/IP
- "First" gigabit TCP [1999]
- Consensus that zero-copy sockets are not general [2001]
- IETF RDMA working group [2002]
- Direct Access File System [2002]
- iSCSI block storage for TCP/IP
- Revival of TCP/IP offload
- 10+GE
- NFS/RDMA, offload chips, etc.
- Uncalibrated marketing claims



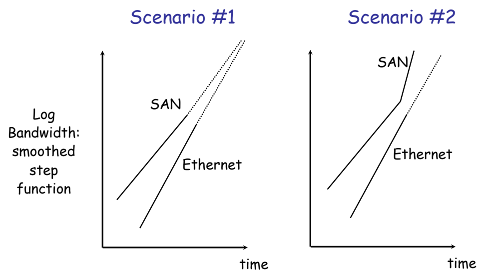
Ethernet/IP in the data center

- 10+Gb/s Ethernet continues the trend of Ethernet speeds outrunning Moore's Law.
- Ethernet runs IP.
- This trend increasingly enables IP to compete in "high performance" domains.
  - Data centers and other "SAN" markets
    - {System, Storage, Server, Small} Area Network
    - Specialized/proprietary/nonstandard
  - Network storage: iSCSI vs. FC
  - Infiniband vs. IP over 10+GE

Ethernet/IP vs. "Real" SANs

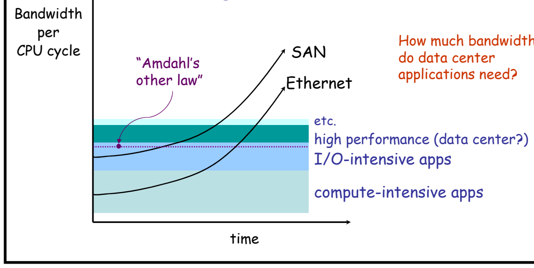
- IP offers many advantages
  - One network
  - Global standard
  - Unified management, etc.
- But can IP really compete?
- What do "real" SANs really offer?
  - Fatter wires?
  - Lower latency?
  - Lower host overhead

SAN vs. Ethernet Wire Speeds



Outrunning Moore's Law?

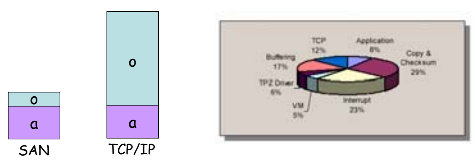
Whichever scenario comes to pass, both SANs and Ethernet are advancing ahead of Moore's Law.



The problem: overhead

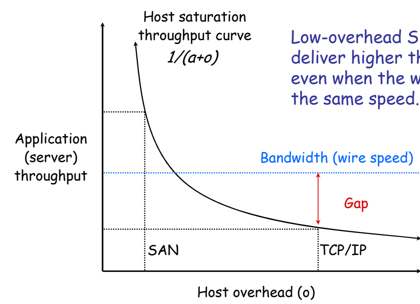
Ethernet is cheap, and cheap NICs are dumb.

Although TCP/IP family protocol processing itself is reasonably efficient, managing a dumb NIC steals CPU/memory cycles away from the application.



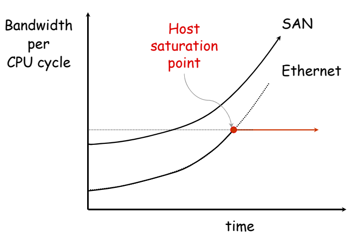
a = application processing per unit of bandwidth
o = host communication overhead per unit of bandwidth

The host/network gap



Low-overhead SANs can deliver higher throughput, even when the wires are the same speed.

Hitting the wall



Throughput improves as hosts advance, but bandwidth per cycle is constant once the host saturation point is reached.

"IP SANs"

- If you believe in the problem, then the solution is to attach hosts to the faster wires with smarter NICs.
  - Hardware checksums, interrupt suppression
  - Transport offload (TOE)
  - Connection-aware w/ early demultiplexing
  - ULP offload (e.g., iSCSI)
  - Direct data placement/RDMA
- Since these NICs take on the key characteristics of SANs, let's use the generic term "IP-SAN".
  - or just "offload"

How much can IP-SANs help?

- IP-SAN is a difficult engineering challenge.
  - It takes time and money to get it right.
- LAWS [Shivam&Chase03] is a "back of napkin" analysis to explore potential benefits and limitations.
- Figure of merit: marginal improvement in peak application throughput ("speedup")
- Premise: Internet servers are fully pipelined
  - Ignore latency (your mileage may vary)
  - IP-SANs can improve throughput if host saturates.

What you need to know (about)

- Importance of overhead and effect on performance
- Distinct from latency, bandwidth
- Sources of overhead in TCP/IP communication
  - Per segment vs. per byte (copy and checksum)
- MSS/MTU size, jumbo frames, path MTU discovery
- Data movement from NIC through kernel to app
- RFC 793 (copy semantics) and its impact on the socket model and data copying overhead.
- Approaches exist to reduce it, and they raise critical architectural issues (app vs. OS vs. NIC)
- RDMA+offload and the layer controversy
- Skepticism of marketing claims for proposed fixes.
- Amdahl's Law
- LFNs

Focusing on the Issue

- The key issue IS NOT:
  - *The pipes:* Ethernet has come a long way since 1981.
    - Add another zero every three years?
  - *Transport architecture:* generality of IP is worth the cost.
  - *Protocol overhead*: run better code on a faster CPU.
  - *Interrupts, checksums, etc*: the NIC vendors can innovate here without us.

All of these are part of the bigger picture, but we don't need an IETF working group to "fix" them.

The Copy Problem

- The key issue IS *data movement within the host.*
  - Combined with other overheads, copying sucks up resources needed for application processing.
- The problem won't go away with better technology.
  - Faster CPUs don't help: it's the memory.
- General solutions are elusive…on the receive side.
- The problem exposes basic structural issues:
  - interactions among NIC, OS, APIs, protocols.

"Zero-Copy" Alternatives

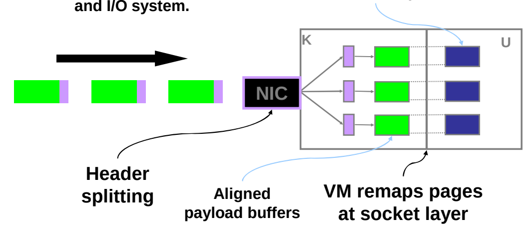
- Option 1: page flipping
  - NIC places payloads in aligned memory; OS uses virtual memory to map it where the app wants it.
- Option 2: scatter/gather API
  - NIC puts the data wherever it want; app accepts the data wherever it lands.
- Option 3: direct data placement
  - NIC puts data where the headers say it should go.

Each solution involves the OS, application, and NIC to some degree.

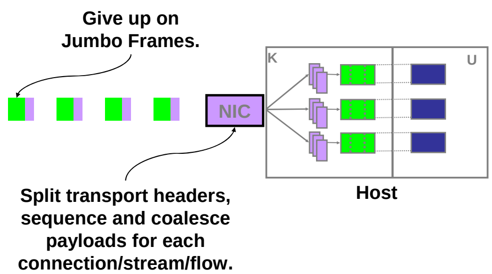
Page Flipping: the Basics

Goal: deposit payloads in aligned buffer blocks suitable for the OS VM and I/O system.

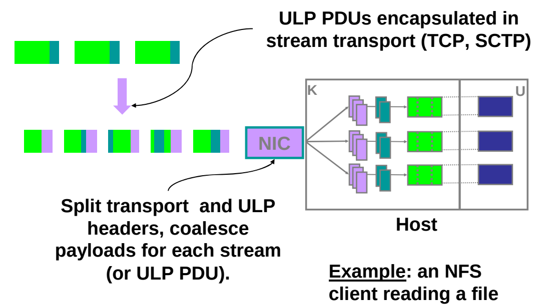
Receiving app specifies buffers (per RFC 793 copy semantics).



Page Flipping with Small MTUs



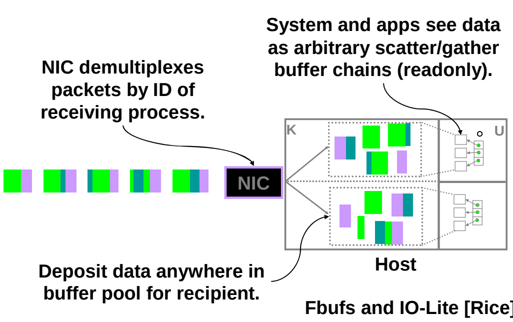
Page Flipping with a ULP



Page Flipping: Pros and Cons

- Pro: sometimes works.
  - Application buffers must match transport alignment.
- NIC must split headers and coalesce payloads to fill aligned buffer pages.
- NIC must recognize and separate ULP headers as well as transport headers.
- Page remap requires TLB shootdown for SMPs.
  - Cost/overhead scales with number of processors.

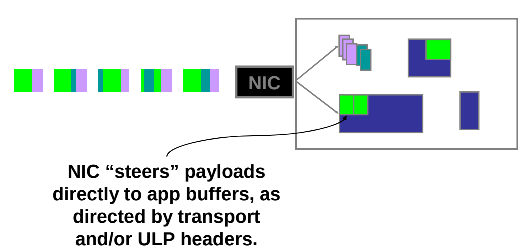
Option 2: Scatter/Gather



Scatter/Gather: Pros and Cons

- Pro: just might work.
- New APIs
- New applications
- New NICs
- New OS
- May not meet app alignment constraints.

Option 3: Direct Data Placement



DDP: Pros and Cons

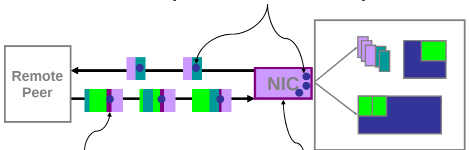
- Effective: deposits payloads directly in designated receive buffers, without copying or flipping.
- General: works independent of MTU, page size, buffer alignment, presence of ULP headers, etc.
- Low-impact: if the NIC is "magic", DDP is compatible with existing apps, APIs, ULPs, and OS.
- Of course, there are no magic NICs...

DDP: Examples

- TCP Offload Engines (TOE) can steer payloads directly to preposted buffers.
  - Similar to page flipping ("pack" each flow into buffers)
  - Relies on preposting, doesn't work for ULPs
- ULP-specific NICs (e.g., iSCSI)
  - Proliferation of special-purpose NICs
  - Expensive for future ULPs
- RDMA on non-IP networks
  - VIA, Infiniband, ServerNet, etc.

Remote Direct Memory Access

Register buffer steering tags with NIC, pass them to remote peer.



RDMA-like transport shim carries directives and steering tags in data stream.

Directives and steering tags guide NIC data placement.

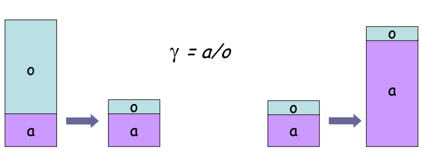
LAWS ratios

α	Ratio of Host CPU speed to NIC processing speed (Lag ratio)
γ	CPU intensity (compute/communication) of the application (Application ratio)
σ	Percentage of wire speed the host can deliver for *raw* communication without offload (Wire ratio)
β	Portion of network work not eliminated by offload (Structural ratio)

"On the Elusive Benefits of Protocol Offload", Shivam and Chase, NICELI 2003.

Application ratio (γ)

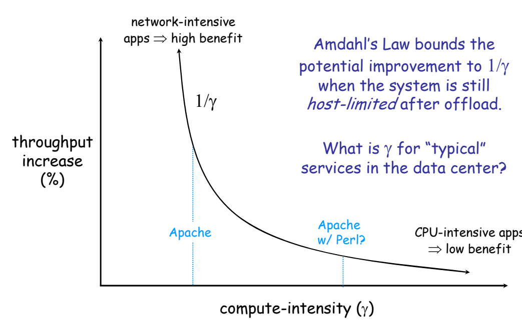
Application ratio (γ) captures "compute-intensity".



For a given application, lower overhead increases γ.

For a given communication system, γ is a property of the application: it captures processing per unit of bandwidth.

γ and Amdahl's Law



Amdahl's Law bounds the potential improvement to 1/γ when the system is still *host-limited* after offload.

What is γ for "typical" services in the data center?

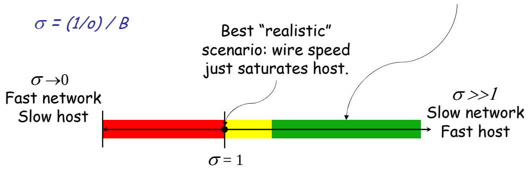
Wire ratio (σ)

Wire ratio (σ) captures host speed relative to network.

B = network bandwidth

Host saturation throughput for raw communication = *1/o*

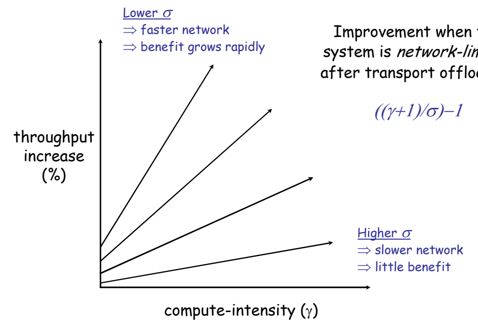
$\sigma = (1/o) / B$



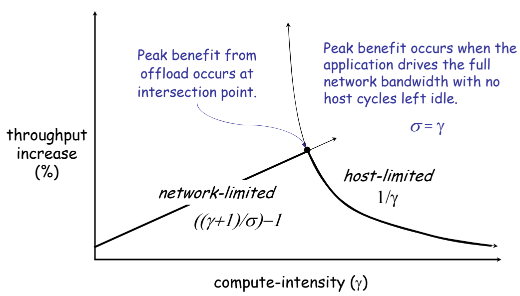
Effect of wire ratio (σ)

Improvement when the system is *network-limited* after transport offload is:

$((\gamma+1)/\sigma)-1$



Putting it all together

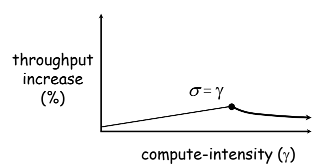


Offload for fast hosts

Faster hosts, better protocol implementations, and slower networks all push σ higher.

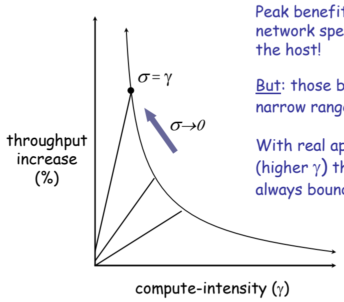
E.g., a 100 Mb/s net on a "gigabit-ready" host gives σ=10.

The throughput improvement is bounded by 1/σ (e.g., 10%).



Key question: Will network advances continue to outrun Moore's Law and push σ lower over the long term?

Offload for fast networks



Peak benefit is unbounded as the network speed advances relative to the host!

But: those benefits apply only to a narrow range of low-γ applications.

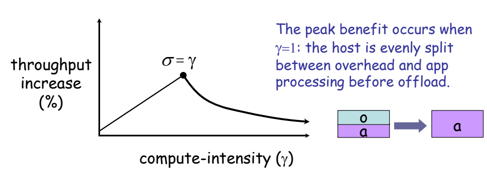
With real application processing (higher γ) the potential benefit is always bounded by 1/γ.

Offload for a "realistic" network

The network is *realistic* if the host can handle raw communication at wire speed (σ≥1).

The "best realistic scenario" is σ=1: raw communication just saturates the host.

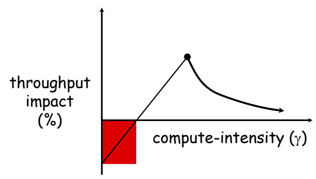
In this case, offload improves throughput by up to a factor of two (100%), but no more.



Pitfall: offload to a slow NIC

If the NIC is too slow, it may limit throughput when γ is low.

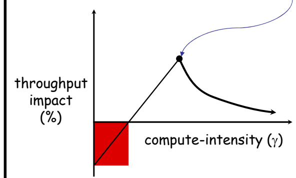
The slow NIC has no impact on throughput unless it saturates, but offload may do more harm than good for low-γ applications.



Quantifying impact of a slow NIC

The *lag ratio* (α) captures the relative speed of the host and NIC for communication processing.

When the NIC lags behind the host (α>1) then the peak benefit occurs when α=γ, and is bounded by 1/α.



We can think of the lag ratio in terms of Moore's Law.

E.g., α=2 when NIC technology lags the host by 18 months.

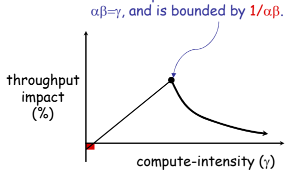
Then the peak benefit from offload is 50%, and it occurs for an application that wasted 33% of system CPU cycles on overhead.

IP transport offload: "a dumb idea whose time has come"?

Offload enables structural improvements such as direct data placement (RDMA) that eliminate some overhead from the system rather than merely shifting it to the NIC.

If a share β of the overhead remains, then the peak benefit occurs when $\alpha\beta$=γ, and is bounded by 1/$\alpha\beta$.

Jeff Mogul, "TCP offload is a dumb idea whose time has come", HotOS 2003.



If β = 50%, then we can get the full benefit from offload with 18 month-old NIC technology.

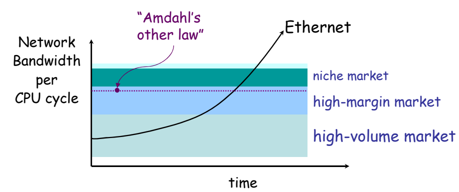
DDP/RDMA eases time-to-market pressure for offload NICs.

Outrunning Moore's Law, revisited

IP-SANs will free IP/Ethernet technology to advance along the curve to higher bandwidth per CPU cycle.

But how far up the curve do we need to go?

If we get ahead of our applications, then the benefits fall off quickly. What if Amdahl was right?



Conclusion

- To understand the role of 10+GE and IP-SAN in the data center, we must understand the applications (γ).
- "Lies, damn lies, and point studies."
  - Careful selection of γ and σ can yield arbitrarily large benefits from SAN technology, but those benefits may be elusive in practice.
- LAWS analysis exposes fundamental opportunities and limitations of IP-SANs and other approaches to low-overhead I/O (including non-IP SANs).
- Helps guide development, evaluation, and deployment.