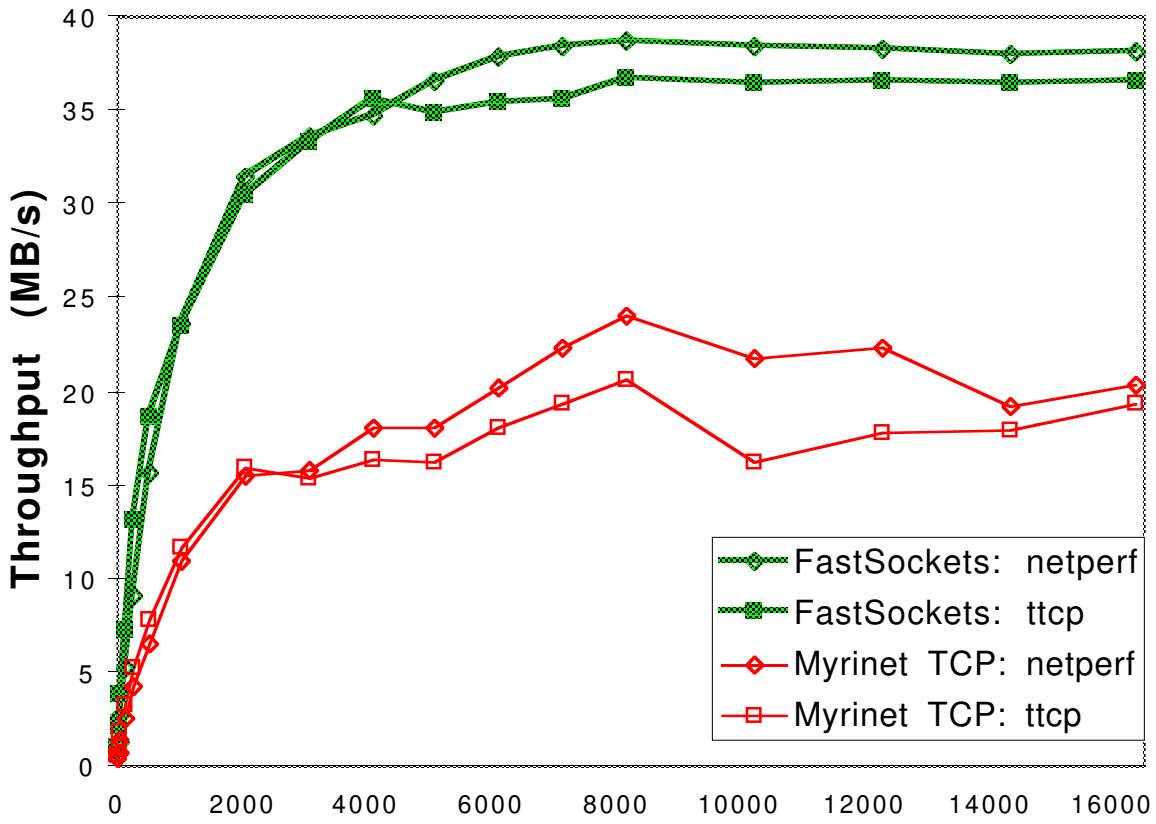
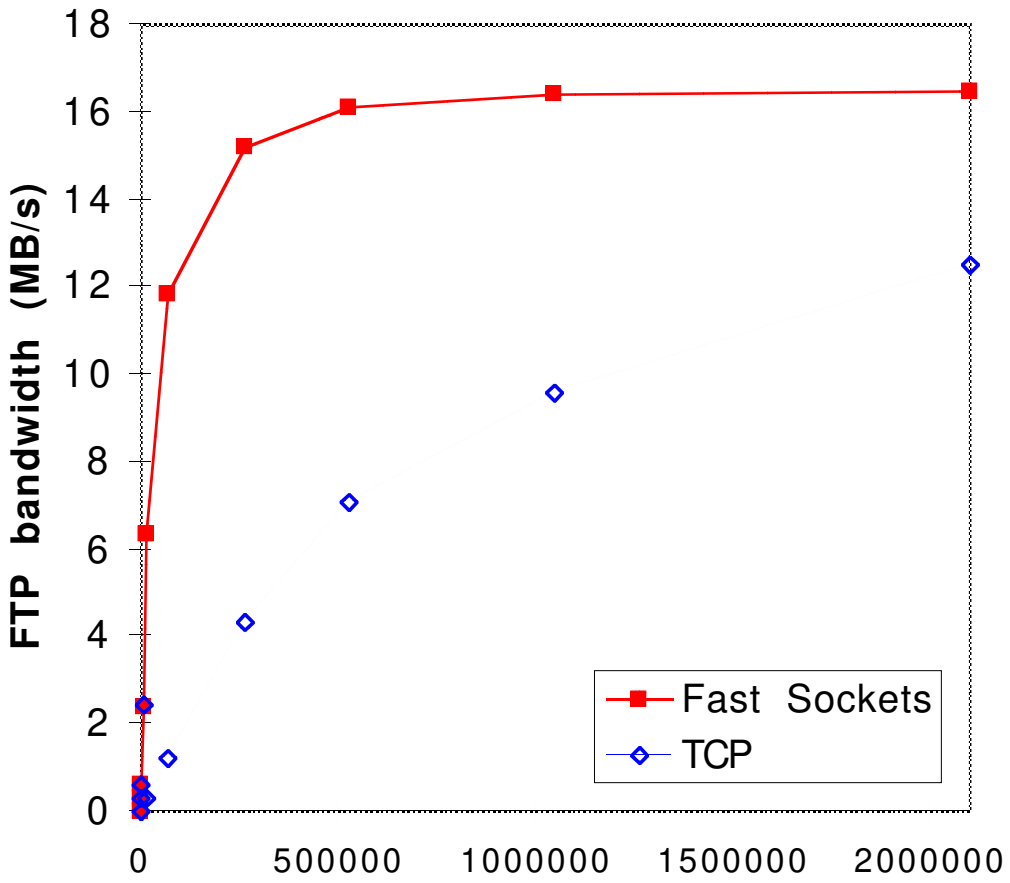
What does it mean for bandwidth?



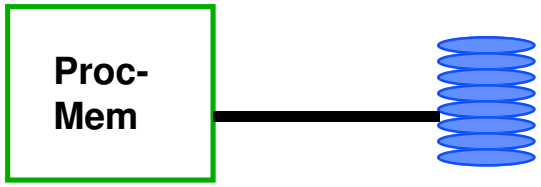
An application: ftp



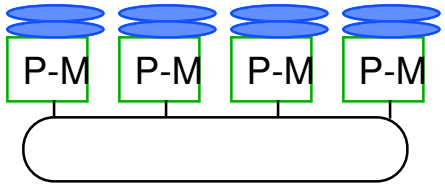
- **NcFTP with in-memory file system**

A look at Serious File I/O

- **Traditional I/O system**

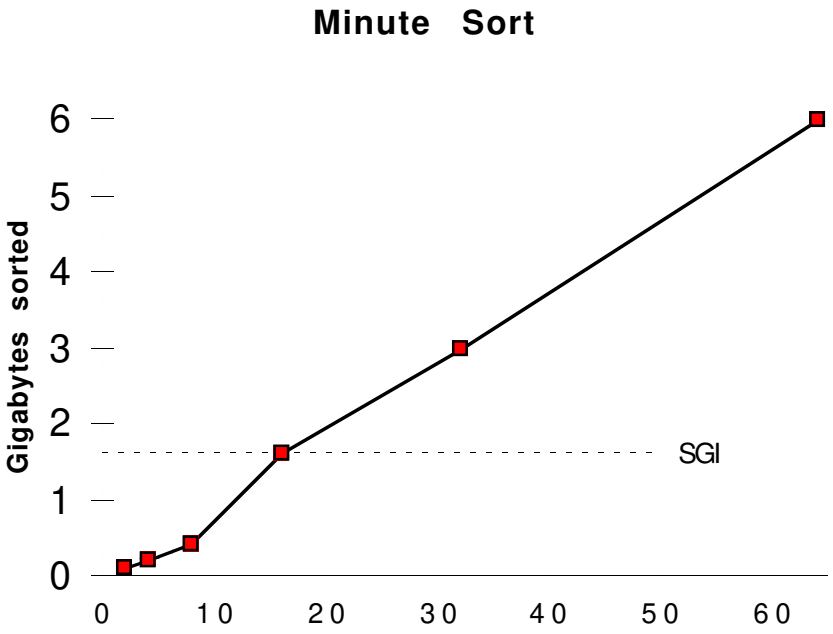


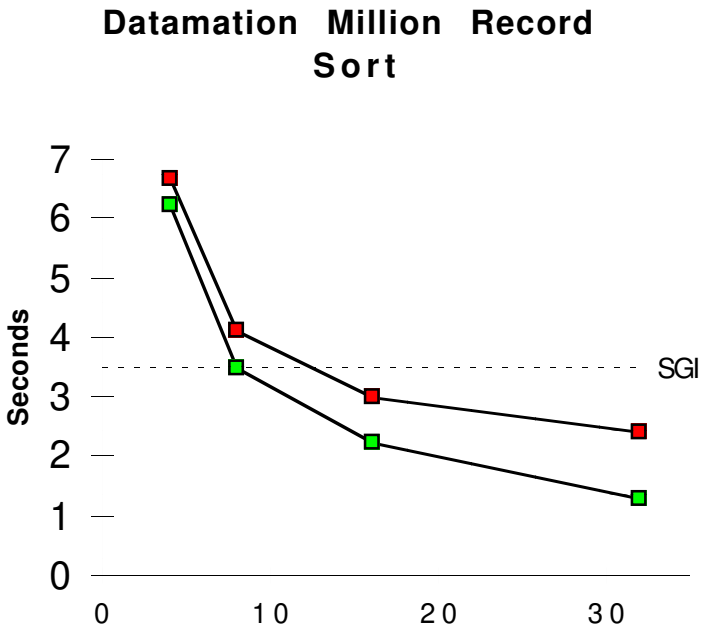
- **NOW I/O system**



- **Benchmark Problem: sort large number of 100 byte records with 10 byte keys**
  - **start on disk, end on disk**
  - **accessible as files (use the file system)**
  - **Datamation sort: 1 million records**
  - **Minute sort: quantity in a minute**

World-Record Disk-to-Disk Sort





Sort: Keys to Success

- **Performance Isolation: highly tuned local disk-to-disk sort**
  - **manage local memory**
  - **manage disk striping**
  - **memory mapped I/O with m-advise, buffering, threads**
  - **manage overlap**
- **Efficient Communication**
  - **completely hidden under disk I/O**
  - **competes for I/O bus bandwidth**
- **Self-tuning Software**
  - **probe available memory, disk bandwidth, trade-offs**

New Era of Distributed/Parallel Services

- Many services distributed over the cluster
  - including O/S
- When awake and interacting with clients need low latency and high bandwidth
- When not, others should have the network
  - timeshare the resource
- Remain available across machine failures, upgrades, expansions, etc.
  - error model

Example: www.hotbot.com @ inktomi.com

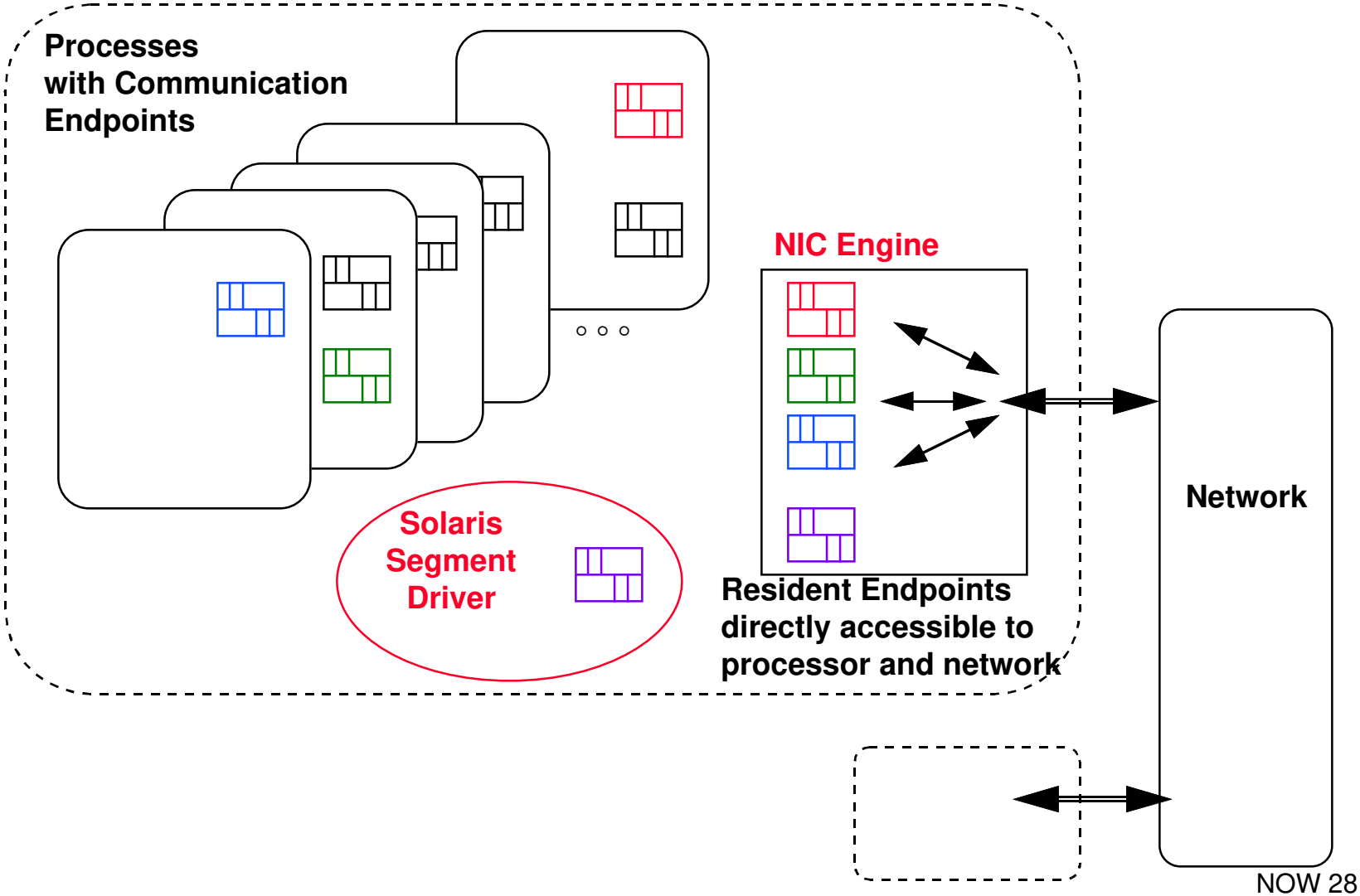
Rethinking Communication: Virtual Networks

- Low latency, high bandwidth => direct hardware access to the network
- At any time, only a small fraction of the potential applications are really communicating.
- Communication within an application is very bursty

=> Provide each application with its own virtual network (s)

Bind virtual networks on demand to physical communication resources

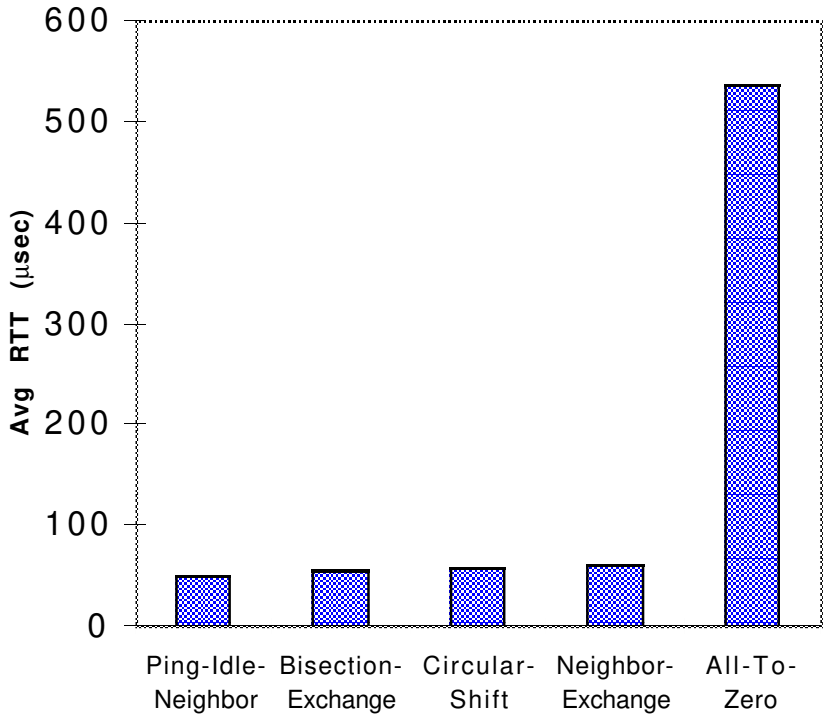
Virtual Networks meet Virtual Memory

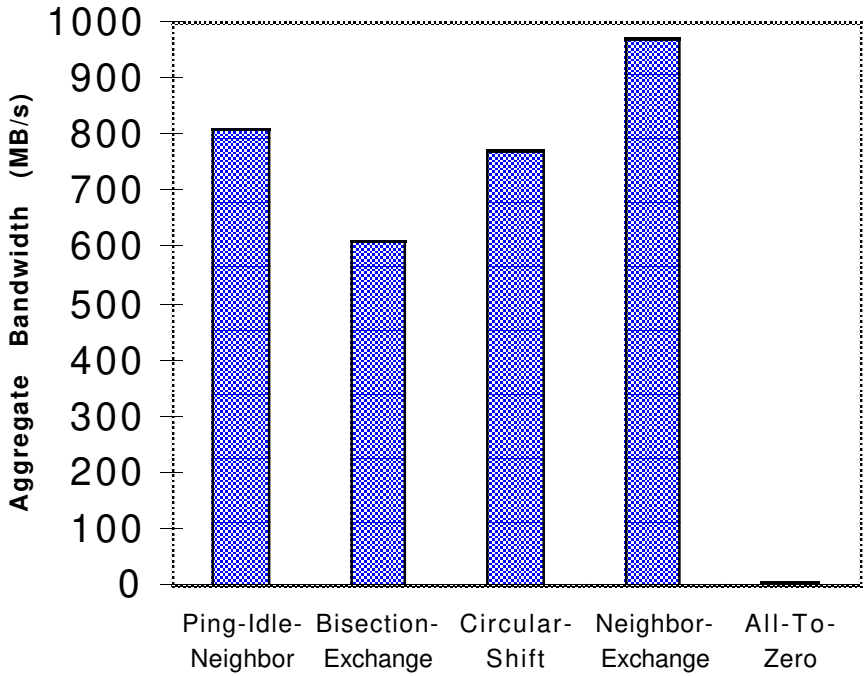


Basic Components

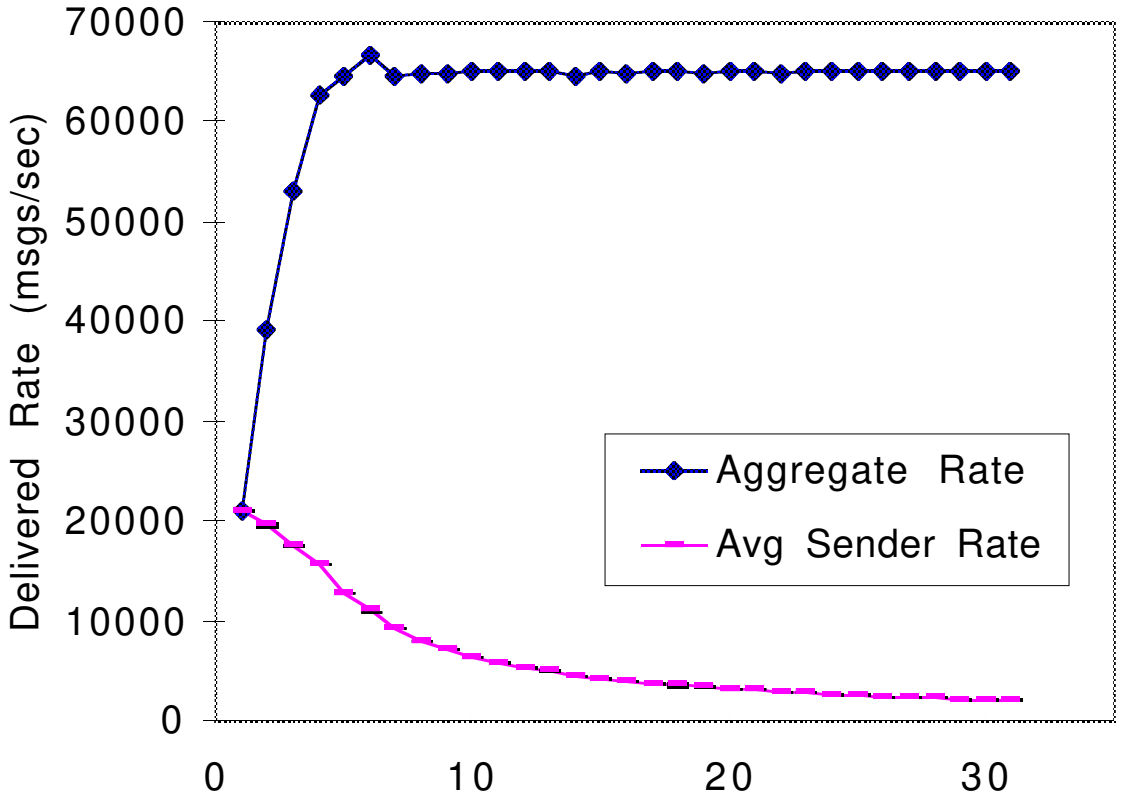
- **NIC firmware**
  - MUXes and DEMUXes traffic between network and resident end-points
  - Implements protection checks
  - Detects errors, end-point faults
  - NIC-to-NIC flow control and sequencing
  - Notifies segment driver of events
- **Segment Driver**
  - manages end-point residency
  - established protection domains
  - allocates / deallocates resources
  - forwards events to scheduler
  - establishes end-point policy

Performance of Current Version

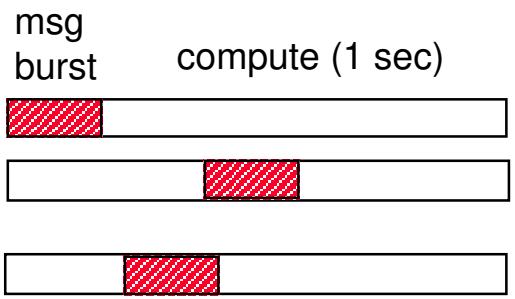


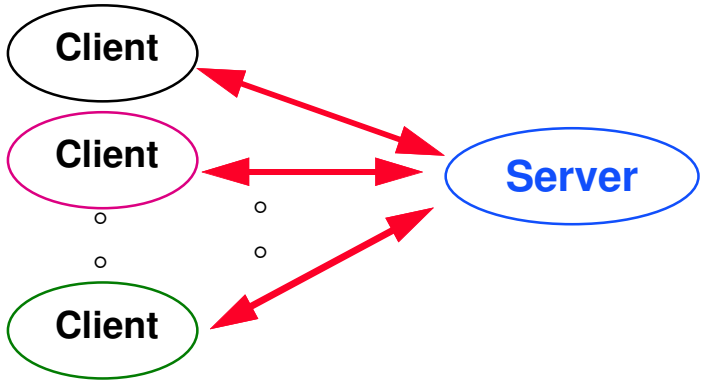


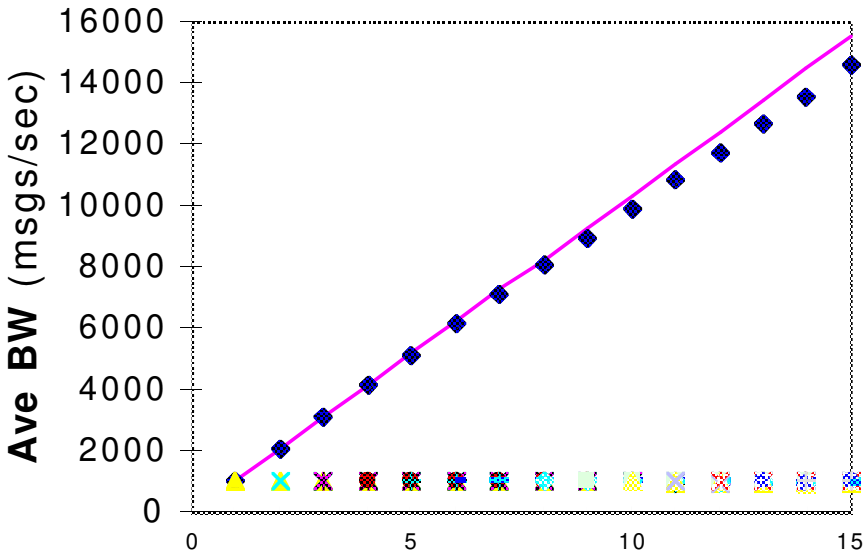
Fairness under heavy load

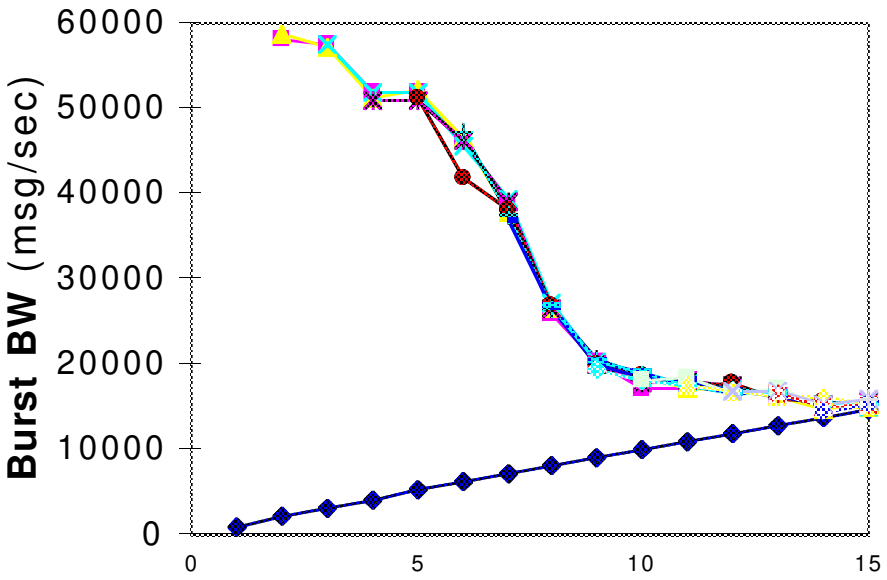


Service to Multiple Virtual Networks









What does this mean for core OS?

- What is the communication working set like?
- When should end-points be made resident?
  - msg send, msg use?
  - msg arrival?
- How does this interact with process scheduling?
  - within a node
  - across the cluster?

Example: implicit co-scheduling

- Conventional Local Schedulers do not understand how to run parallel programs in parallel.
- Can result in 10 - 100 fold slowdown compared to (ideal) co-scheduling
- Co-scheduling (gang scheduling) is bad for:
  - mixed workloads
  - poorly balanced, infrequently communicating pgms

=> use operations naturally occurring in the program to coordinate local scheduling

– communication

– synchronization

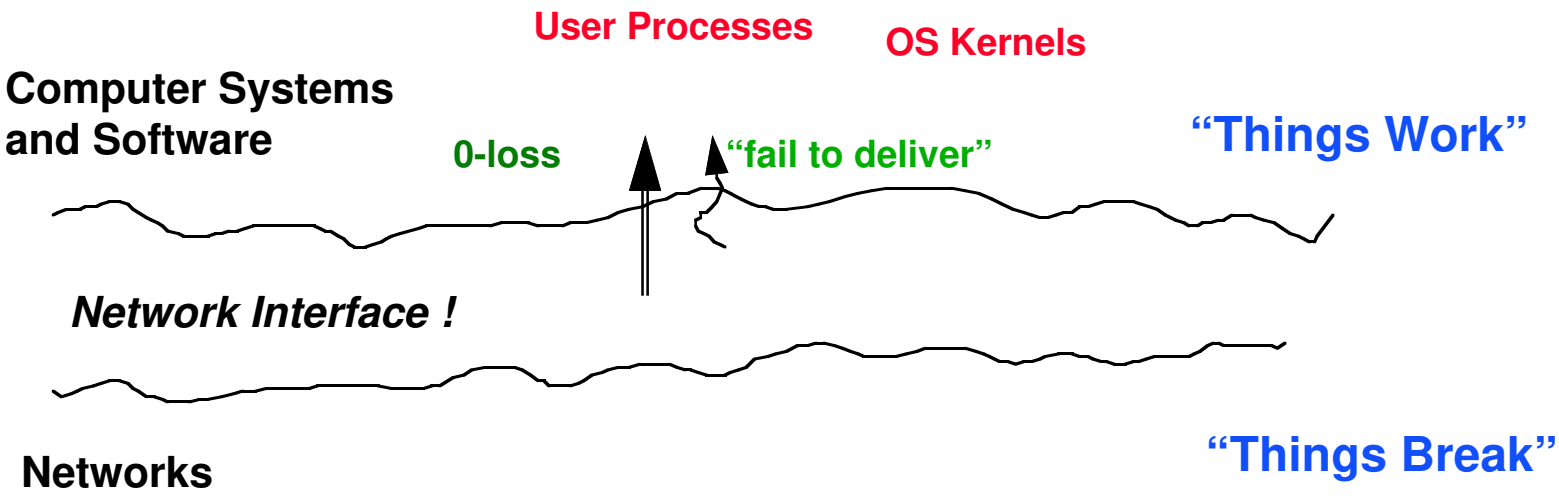
Studies show within 10% of ideal co-scheduling

Example: privileged communication daemons

- Virtualization provides protection boundary
- Privileged programs to have special access to hardware resources

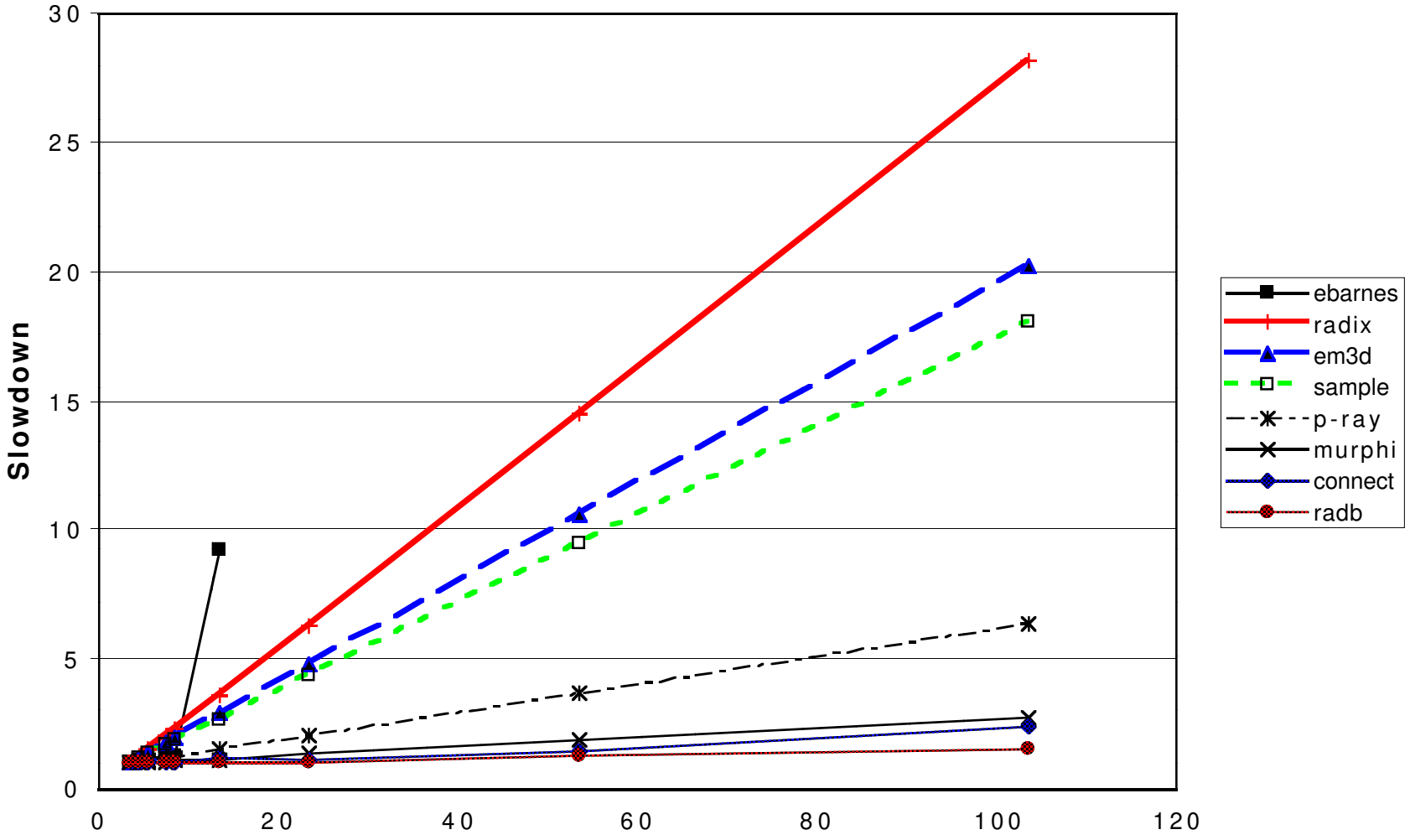
- Example: network mapper daemon
  - probes the network using absolute routes to discover physical topology
  - derives set of deadlock-free routes and distributes them to NIC
- Most of its messages go nowhere
  - stress the Active Message error model
  - return-to-sender

Perspective on errors



- Storage is essential to go "Breaks" –> "Works"
- Low latency x BW => limit on amount
- Net Interface needs dedicated proc. and storage
- Can project 0-loss up, (but pessimistic)
- AM: return-to-sender RPC

Does the "killer network" really matter to applications?



Summary of Concepts

- Plug-and-play scalable, high-BW, low-latency single-chip “killer network” switch is here.
- Dedicated processing and storage in Net Interface is critical
  - last “two inches”, error model, virtualization
- Layering communication for performance
  - like RISC instruction set architectures
- Performance Isolation and Fault Isolation
- Self-tuning Systems
- Virtualize the network
  - performance, protection, and generality
- Clusters are a new class of architecture
  - Push Technology Change throughout the system design

Clusters of SMPs (CLUMPS)

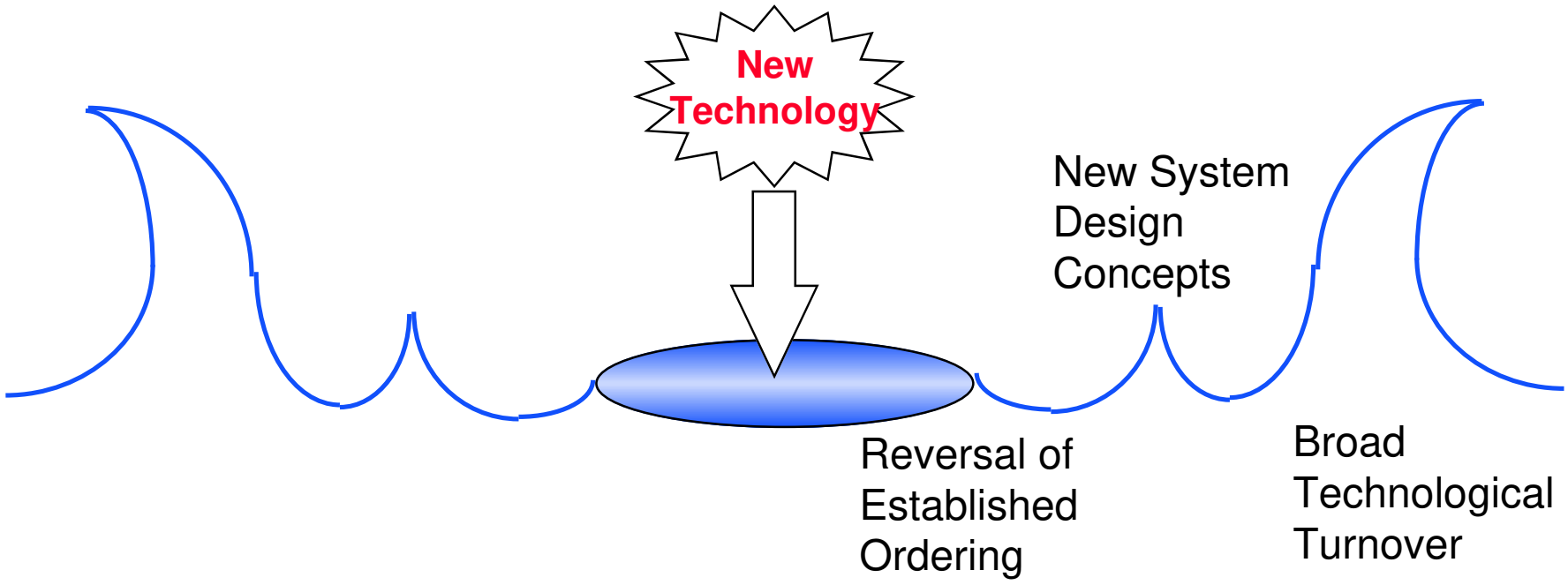
Towards Immense Disk Clusters

Toward a Web O.S.

- Build on basic technology developed for NOW to provide a powerful operating environment for advanced web applications
- Global (URL-based) file system
  - imports home environment to NOW and vice versa
  - build services on a cache-coherent global file system
- OS "sandbox" isolates foreign entity
- Smart (java-based) browsers provide scalable, interactive front-end

- Rent-a-server when you're "too hot"
- Cooperative web caching around the planet
- Interactive services

Technology Ripples



To find out more

- http://now.cs.Berkeley.edu/
  - Active Message Spec. + papers
- "A Case for Networks of Workstations", Anderson, Culler, Patterson, *IEEE Micro*, Feb. 1995.
- "Assessing Fast Network Interfaces", Culler, et al., IEEE Micro, Feb. 1996
- "High-Perf. Sorting on Networks of Workstations", Arpaci-Dusseau, et al, SIGMOD 1997
- "Application Sensitivity to ...", Martin et. al, ISCA97
- "High-Perf. Local Area Communication with Fast Sockets", Rodrigues, et al., USINEX 1996
- Dusseau, et al, "*Effective Distributed Scheduling of Parallel Workloads*", Sigmetrics 1996
- Kronenberg, Levy, Strecker "Vax Clusters: A closely-coupled distributed system," *ACM Transactions on Computer Systems*, May 1986 v 4(3):130-146.