

---

## NOW and the Killer Network

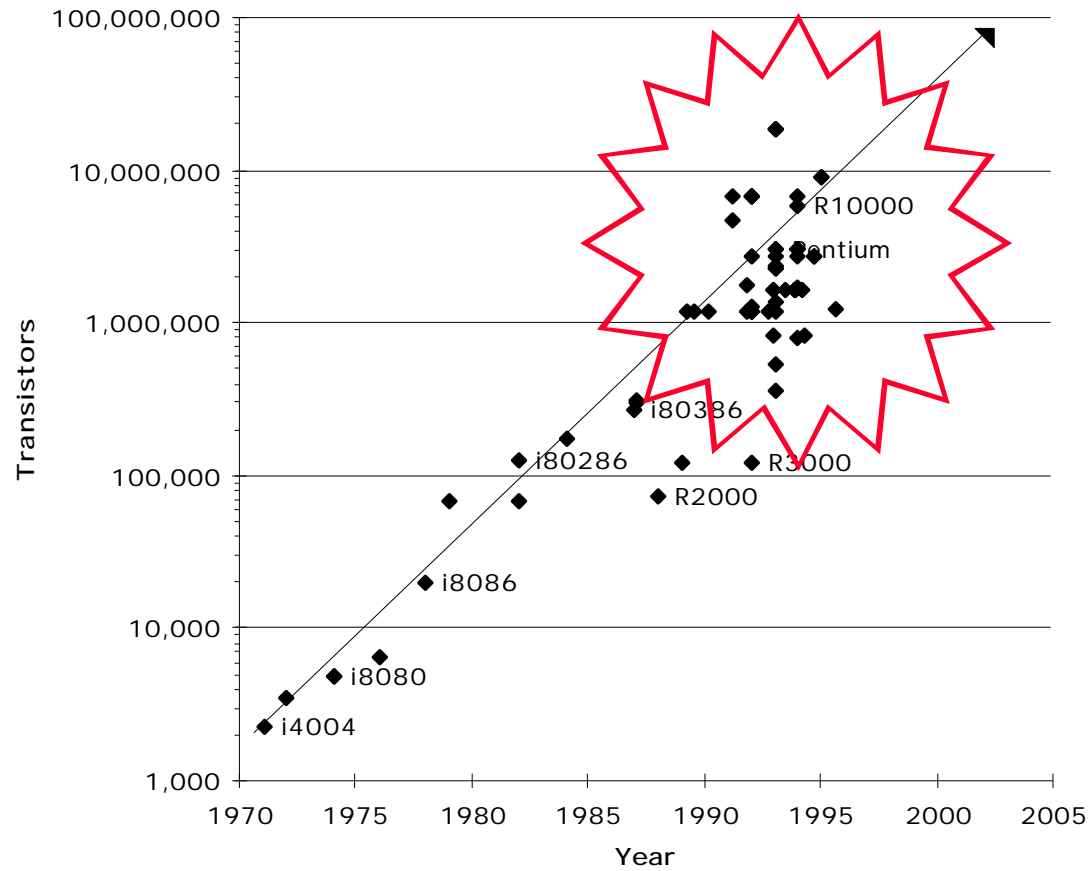


**David E. Culler**  
**culler@cs**

**<http://now.cs.berkeley.edu>**

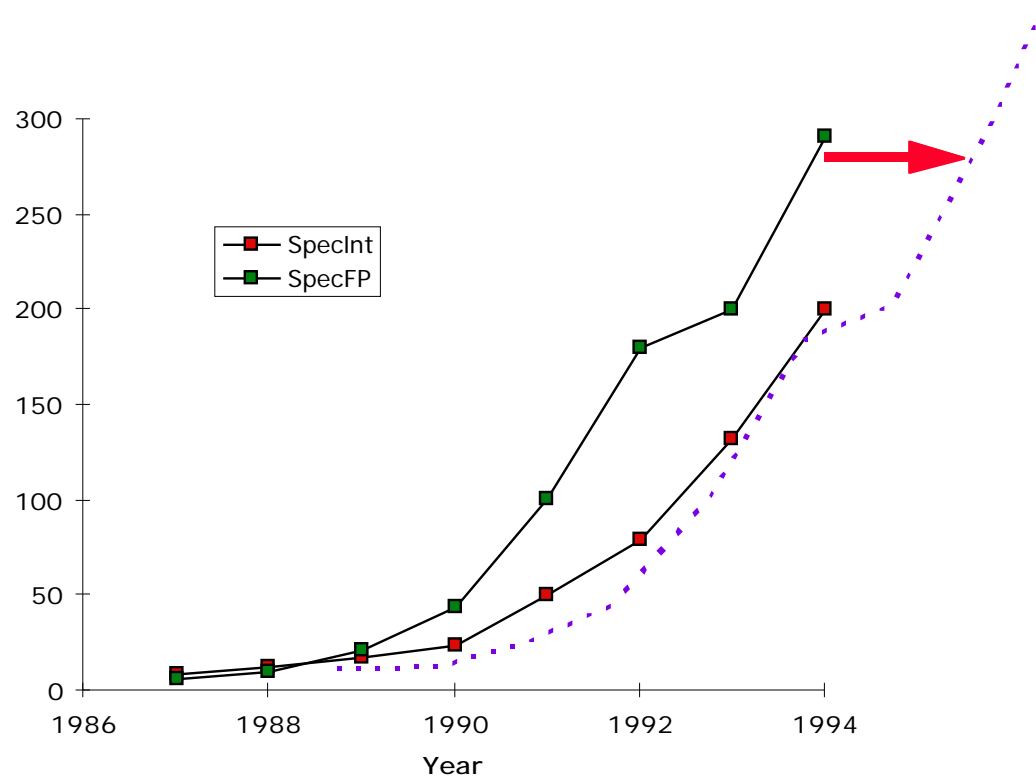
# Remember the Killer Micro

---



# Timely Engineering of Large Systems

---



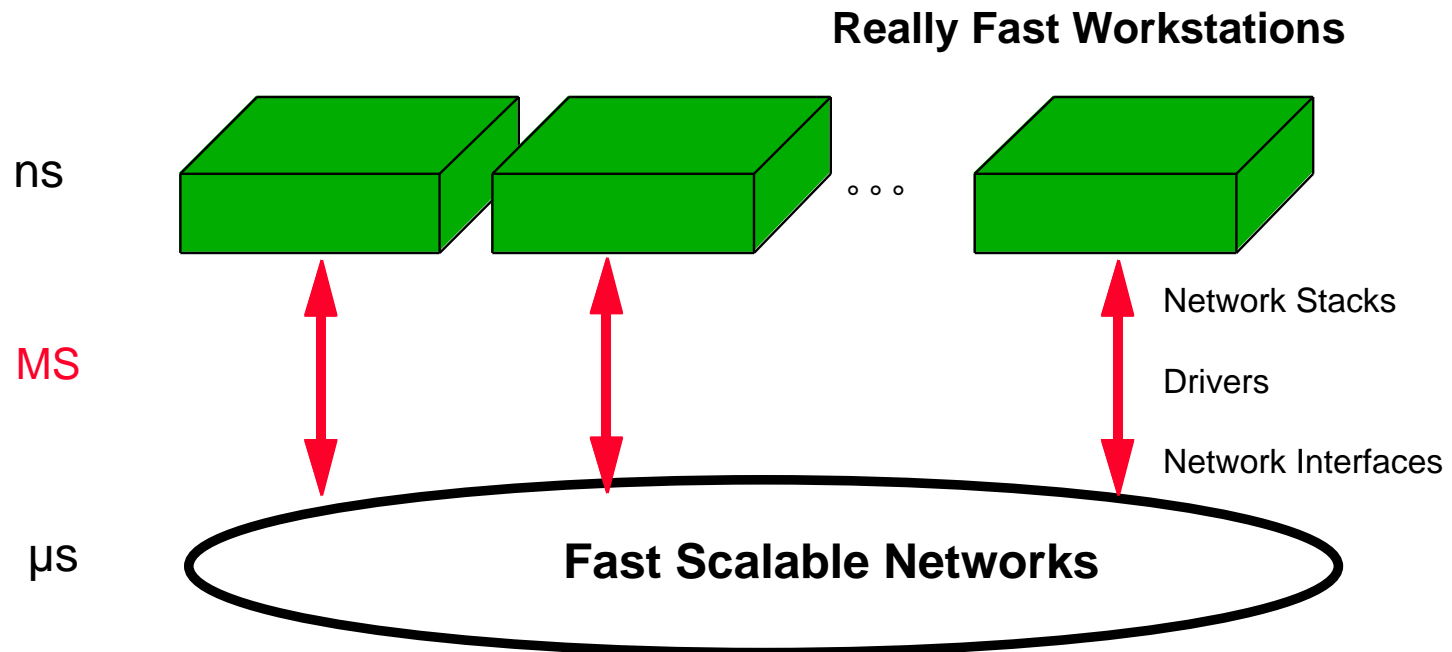
# 90's Technological Breakthrough

---

- **the killer network**
- **the single-chip, high bandwidth, low-latency, high reliability building block for scalable networks**
- **=> the new LAN ?**
- **at least SAN (System Area Network)**

# The “Last Two Inches” Problem

---



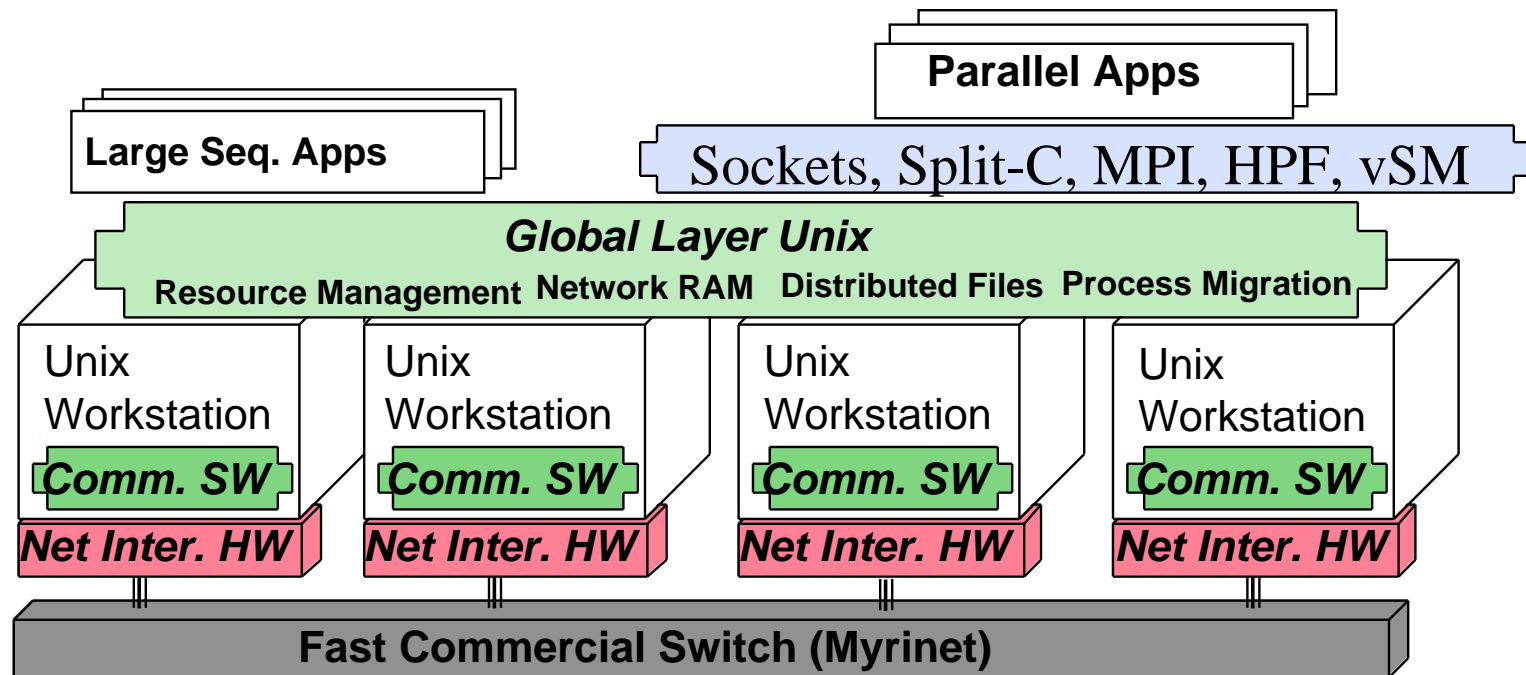
# Goals of the Research

---

- **Fundamental change in how we design large-scale computing systems**
  - snap together commodity components
  - self-managing, self-tuning, highly available
- **Make the “killer network” real**
  - realize the potential of emerging hardware technology
  - and push its effect through the rest of the system
- **Integrated system on a building-wide scale**
  - pool of resources (proc, disk mem)
  - remote processor and memory closer than local disk
  - federation of systems with local and global role
- **The right way to build internet services**

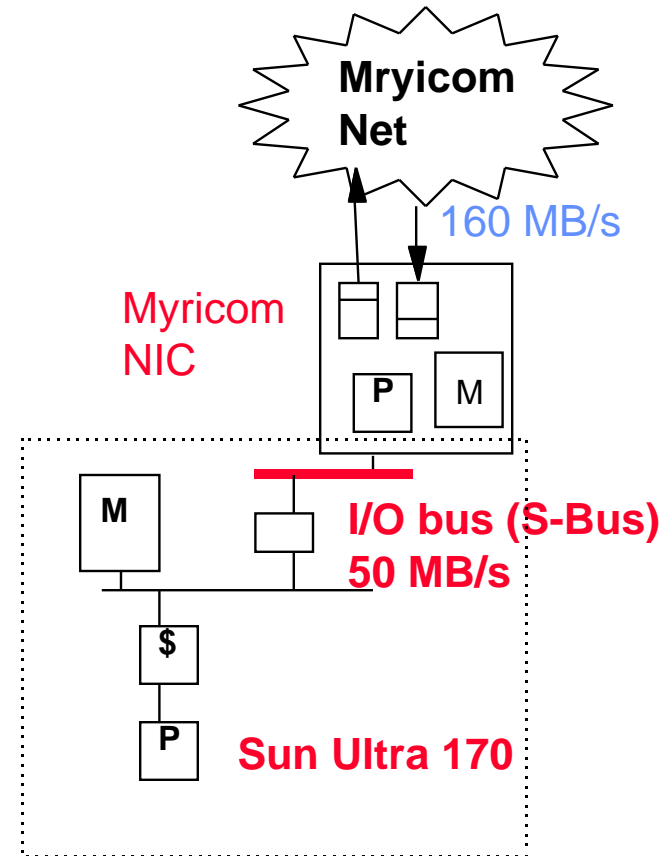
# NOW System Architecture

---



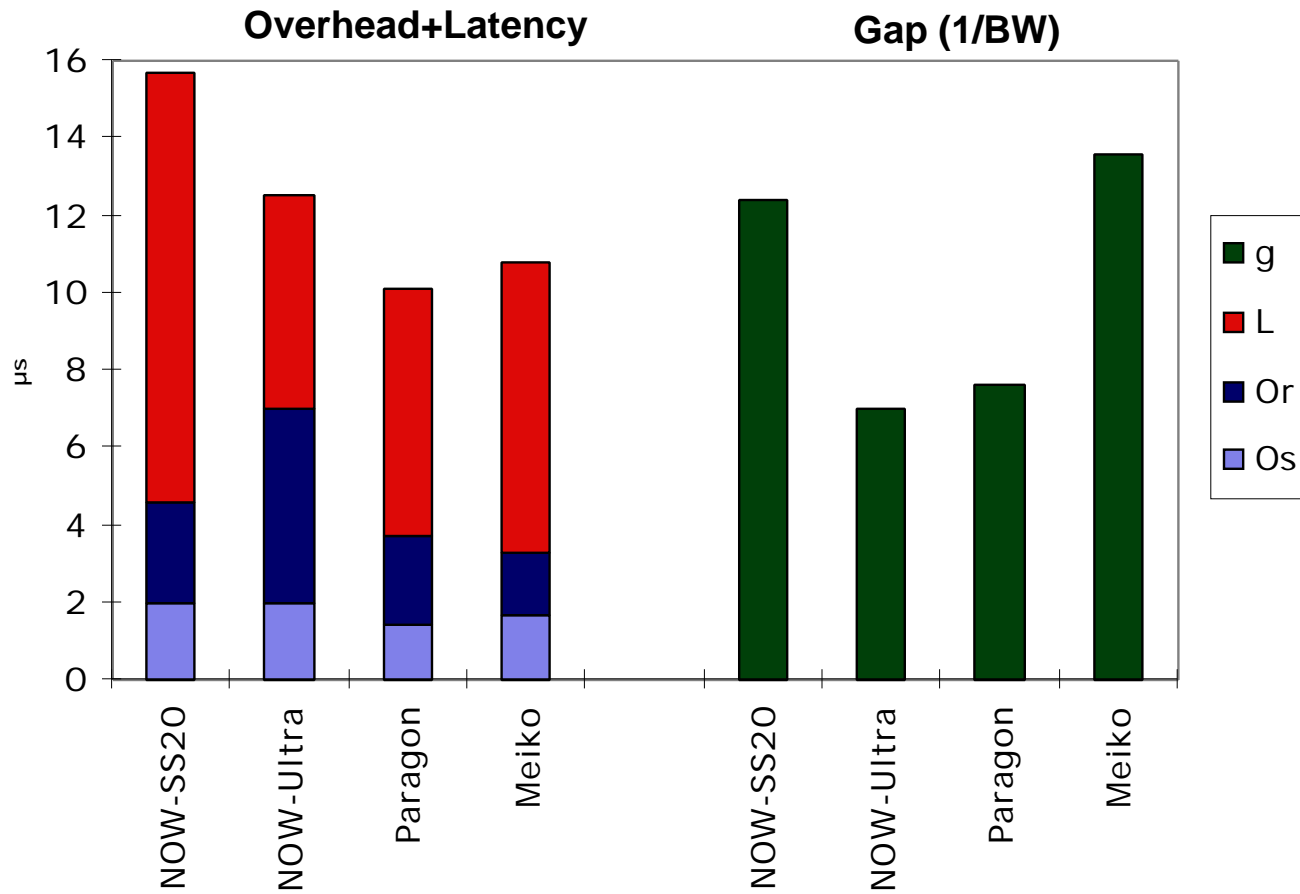
# Intelligent Network Interfaces

- Processing power and storage embedded in the NIC





# AM: Fast, Portable Communication

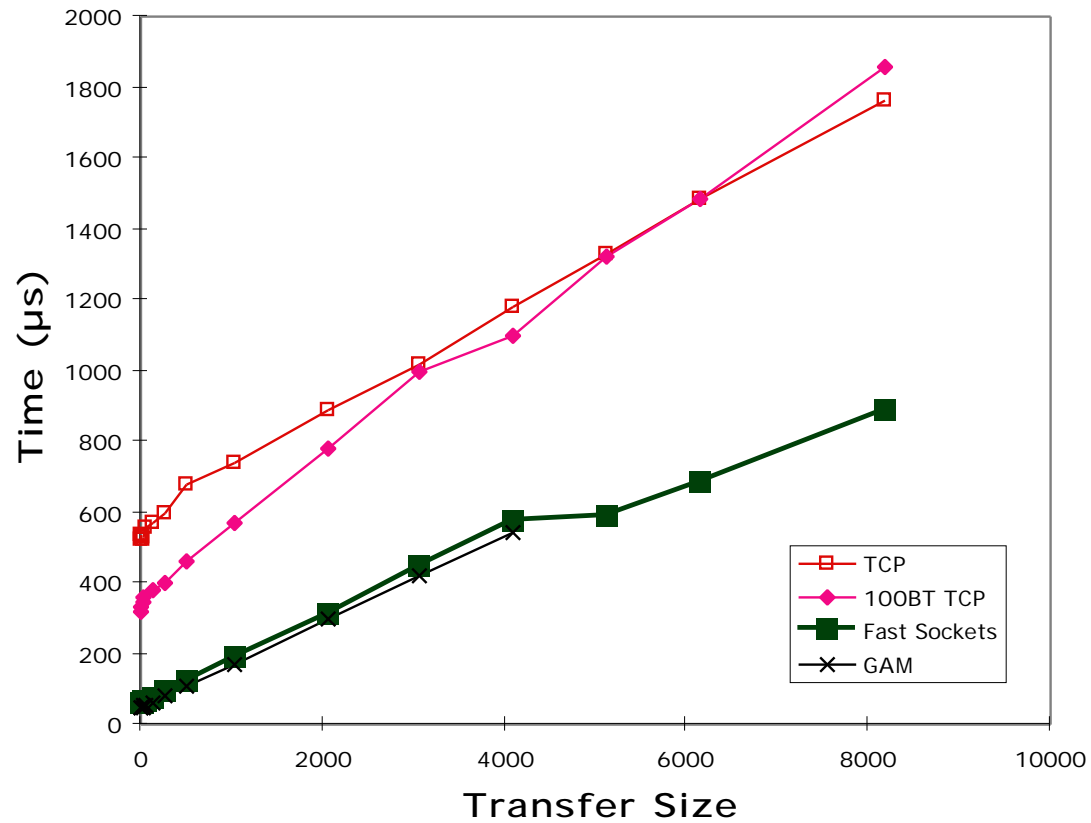


## MPI over AM

---

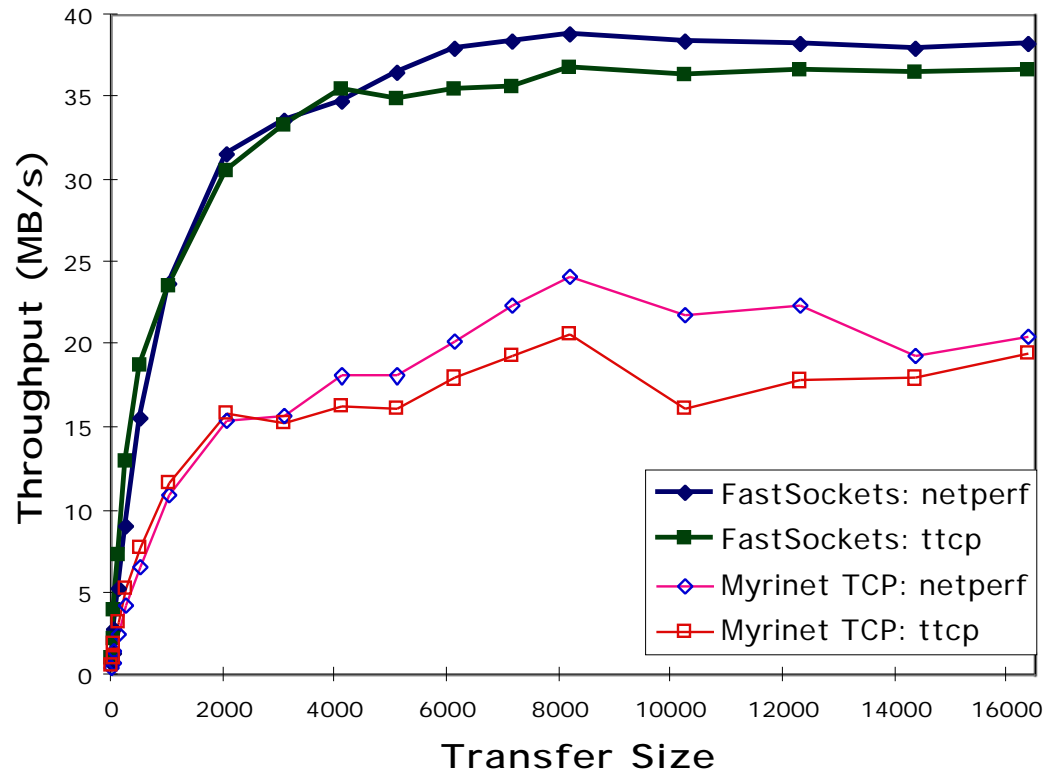
System	Start-up ( $\mu$ s)	Peak BW (MB/s)
NOW	17.5	37.7
Paragon	25.4	171.9
Meiko CS-2	82.5	43.3
IBM SP-2	38.0	34.2

# Sockets over AM: Latency

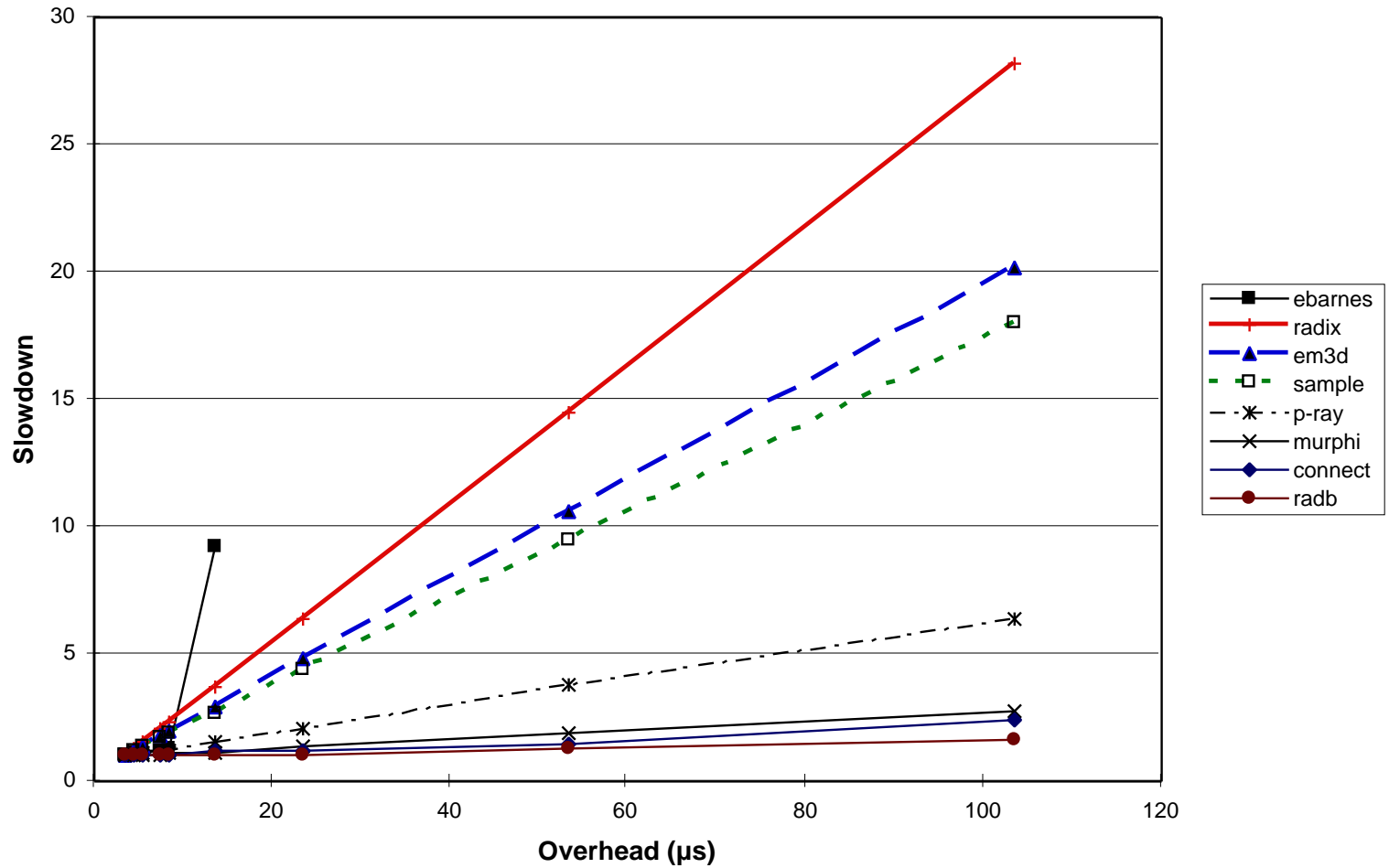


# netperf, ttcp bandwidth

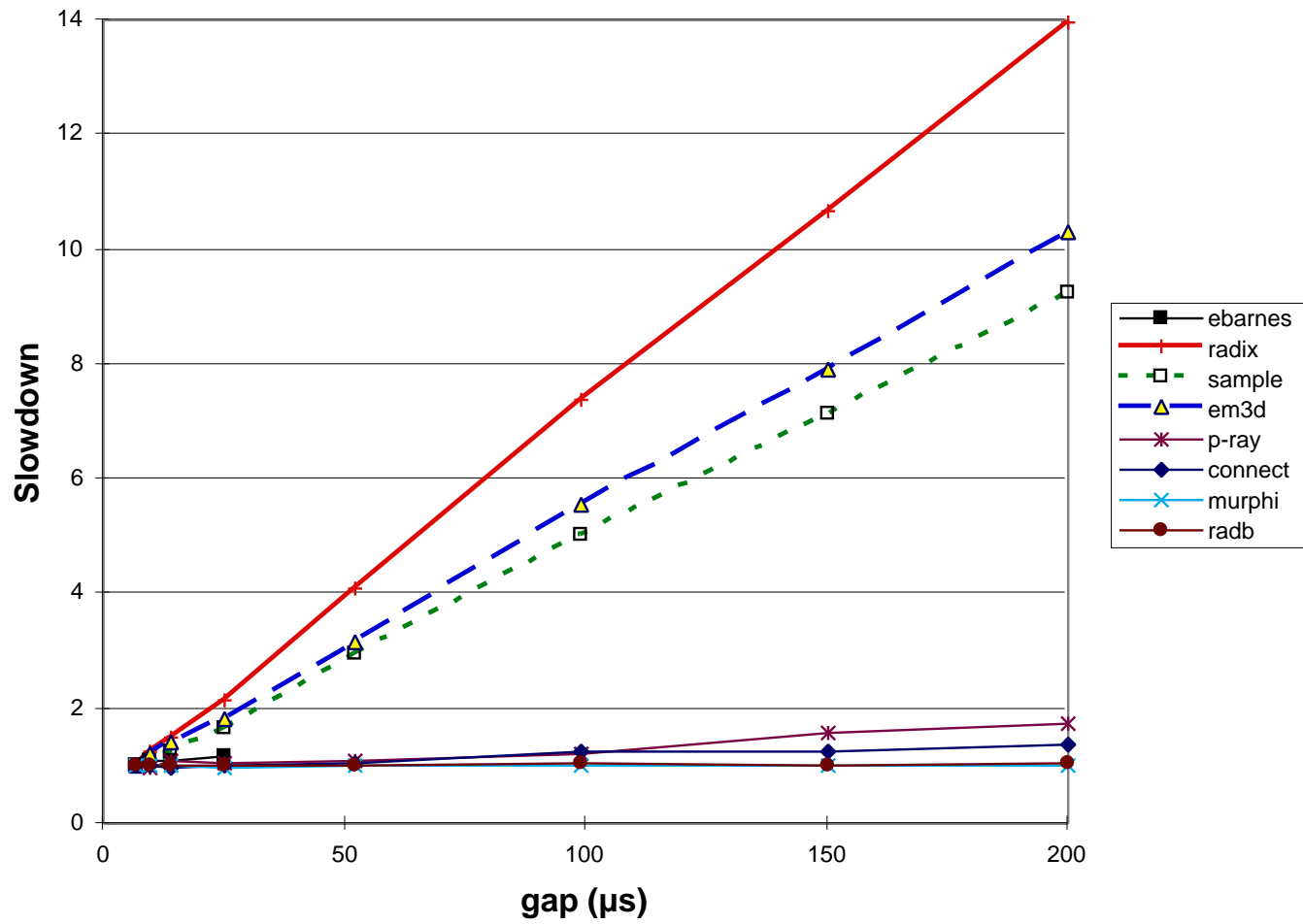
---



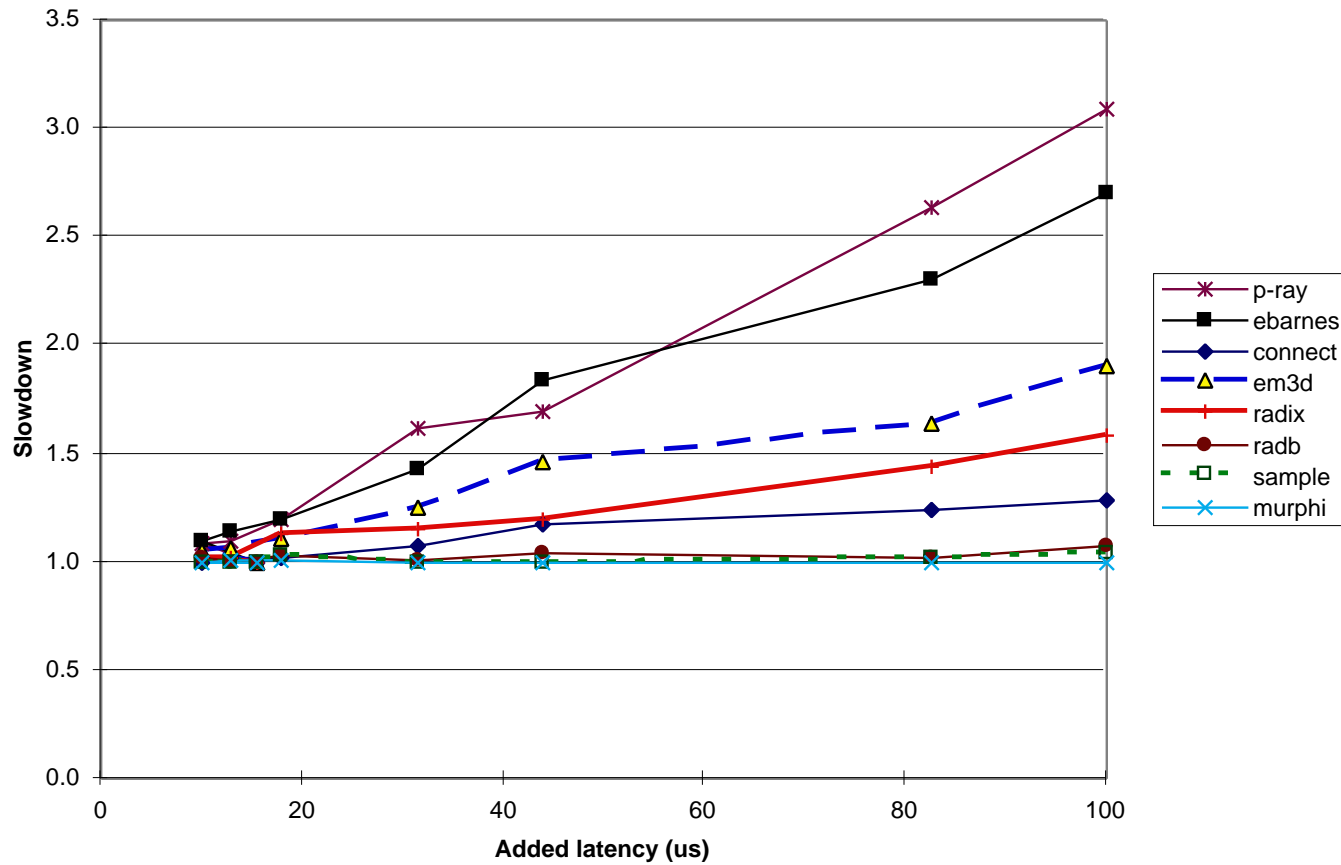
# Application Sensitivity to Overhead



# Sensitivity to gap (1/msg rate)

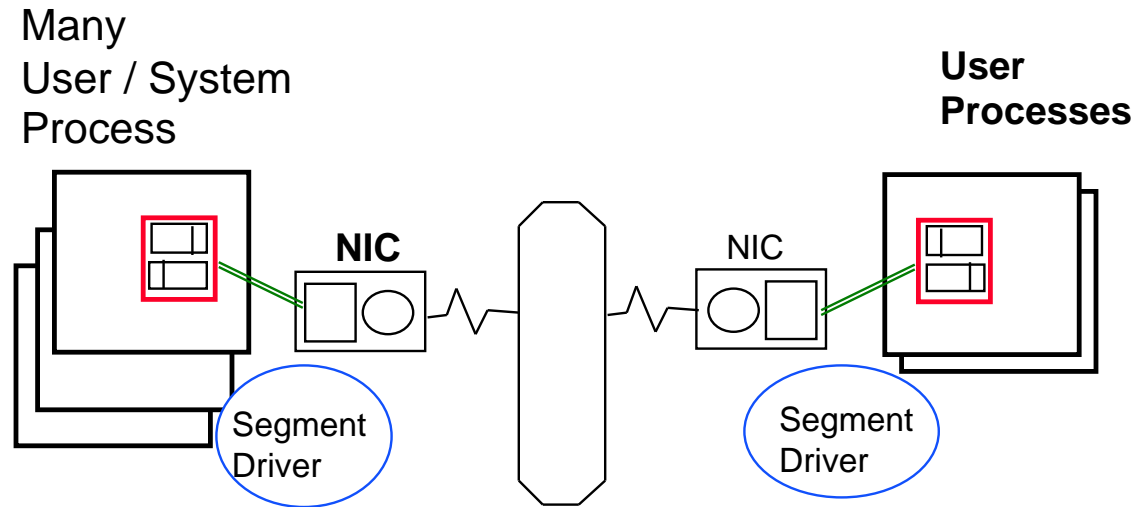


# Sensitivity to Latency



# General Purpose, fault-tolerant “Virtual Networks”

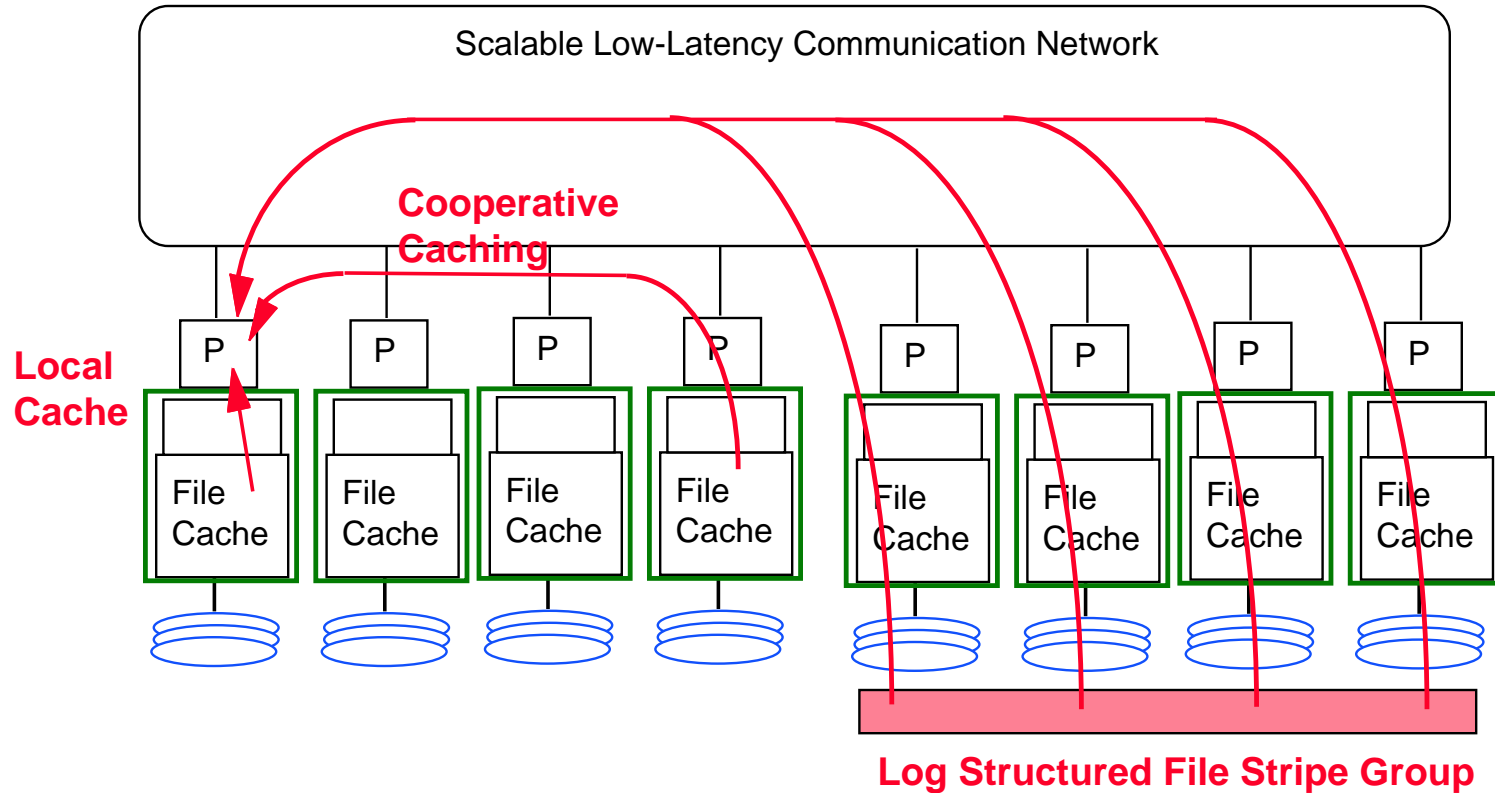
---



- **Dynamic binding of multiple “virtual” network end-points directly to physical NIC resources**
- **Deep integration with VM and threads**
- **Smart NIC mux-demux, errors, and flow-control**
- **Clean error model - return-to-sender**



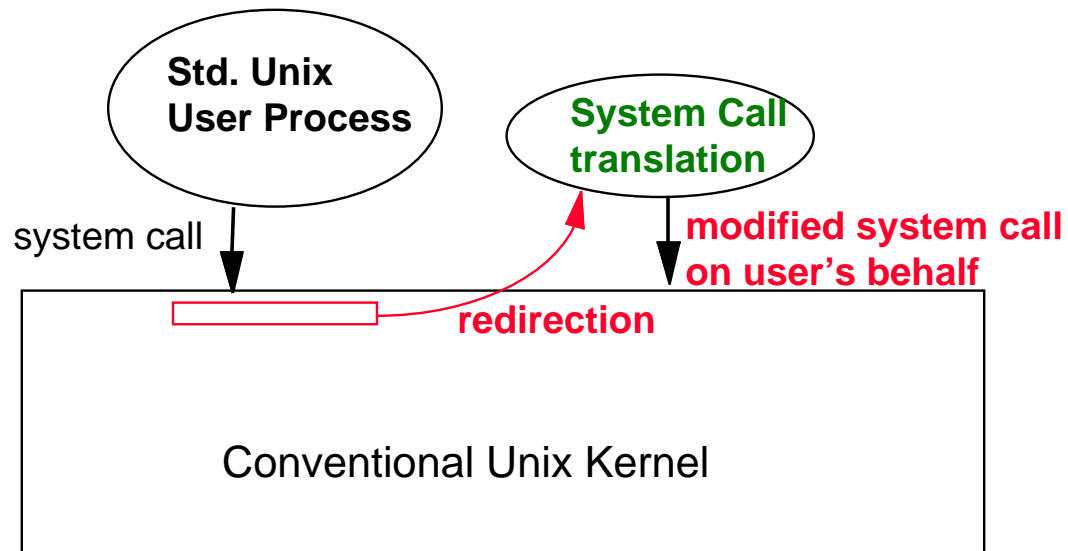
# Truely Distibuted File System – XFS



$$G = \text{Node Comm BW} / \text{Disk BW}$$

# Virtualization of O.S. Services

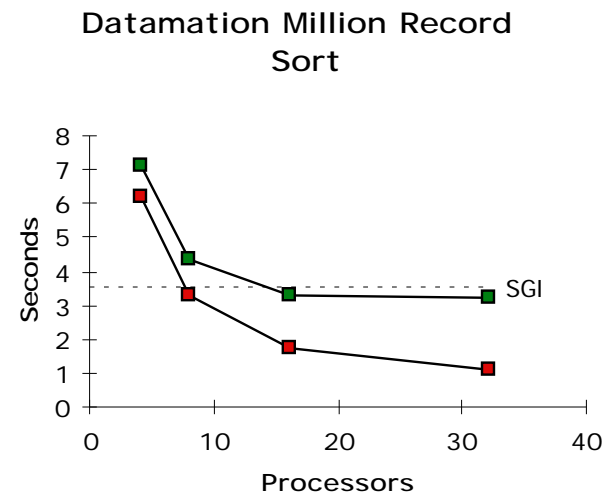
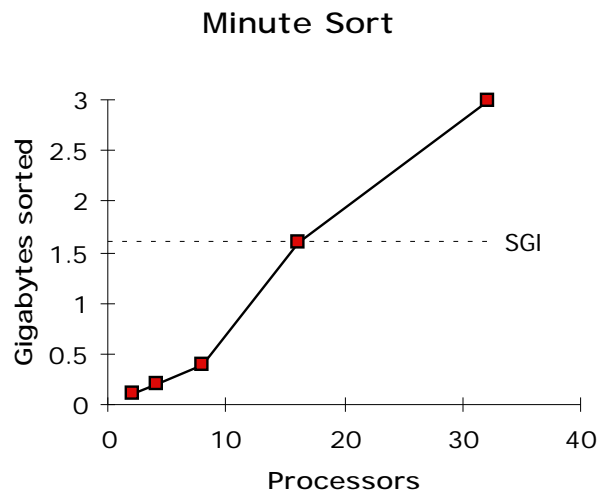
---



- **Small kernel insertion provides redirection to user-level translation facility**
- **Translation provide global-local mapping of system functions for std. binaries**

# World-Record Disk-to-Disk Sort

---



## 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**

# Toward Immense Disk Clusters

---



NOW 21

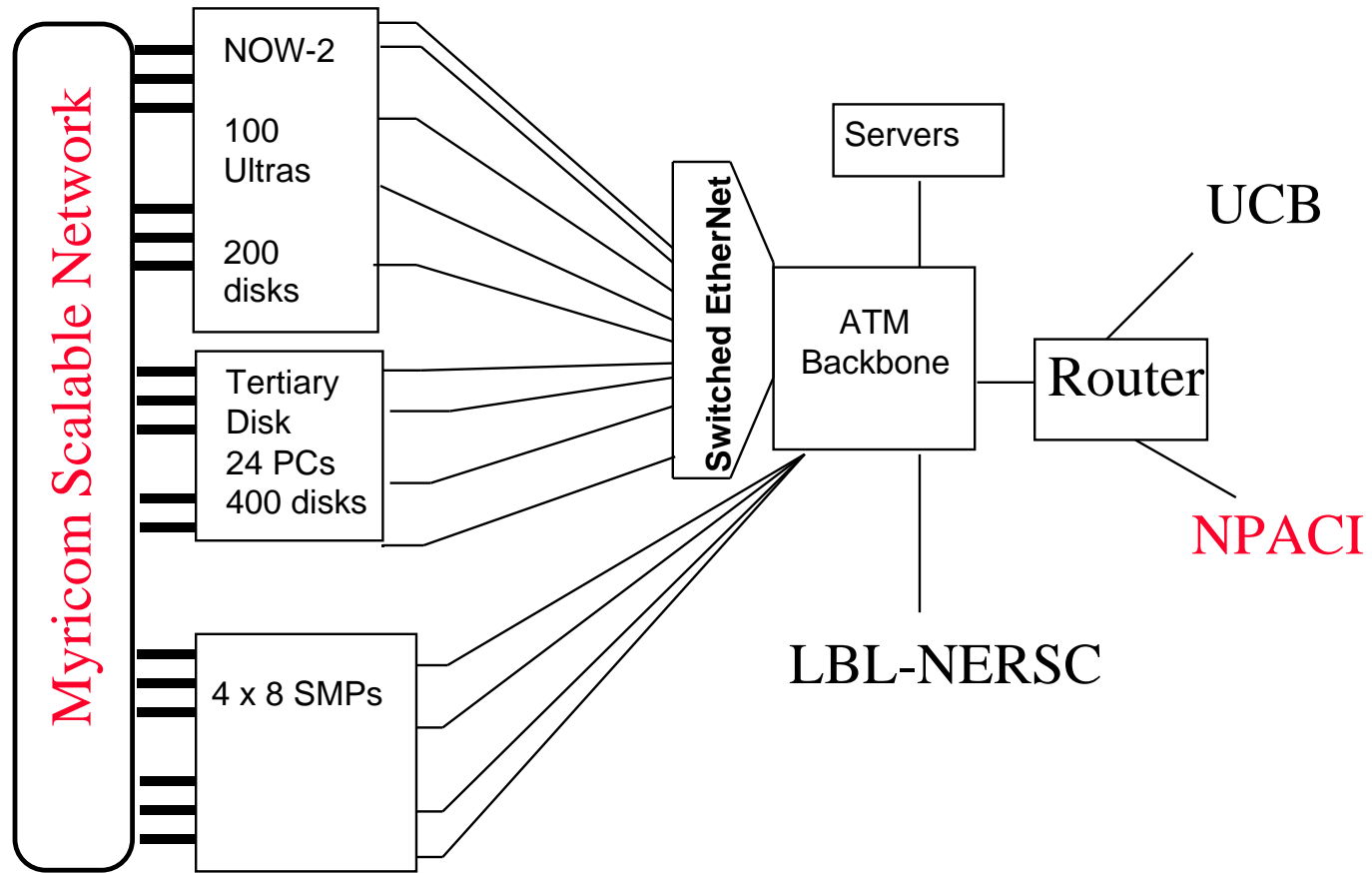
# Clusters of SMPs (CLUMPS)

---



# Overall System Configuration

---



# Invitation

---

- **System is operational enough for research**
- **CS267 is using it heavily**
- **Think about it for term projects**
  - CS252, CS262, CS286, ...
- **Ready to work with other research groups**
- 
- 
- **see: <http://now.cs.berkeley.edu/>**