Lecture 13 (part 1)
Thread Level Parallelism (6)

EEC 171 Parallel Architectures

John Owens
UC Davis

Credits

- © John Owens / UC Davis 2007–17.
- Thanks to many sources for slide material: Computer Organization and Design (Patterson & Hennessy) © 2005, Computer Architecture (Hennessy & Patterson) © 2007, Inside the Machine (Jon Stokes) © 2007, © Dan Connors / University of Colorado 2007, © Kathy Yelick / UCB 2007, © Wen-Mei Hwu/David Kirk, University of Illinois 2007, © David Patterson / UCB 2003–7, © John Lazzaro / UCB 2006, © Mary Jane Irwin / Penn State 2005, © John Kubiatowicz / UCB 2002, © Krste Asinovic/Arvind / MIT 2002, © Morgan Kaufmann Publishers 1998.

Outline

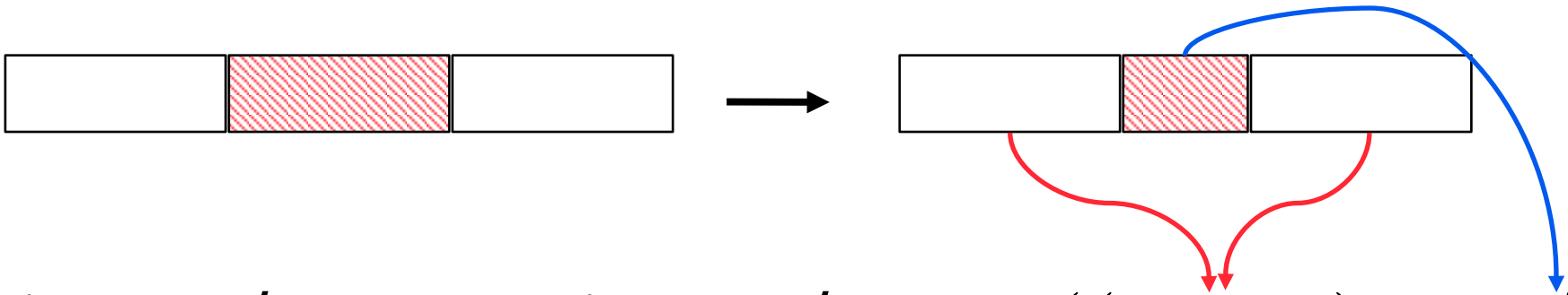
- Grab bag:
  - Amdahl's Law
  - Novices & Parallel Programming
  - Interconnect Technologies

Encountering Amdahl's Law

- Speedup due to enhancement E is

$$\text{Speedup with E} = \frac{\text{Exec time w/o E}}{\text{Exec time with E}}$$

- Suppose that enhancement E accelerates a fraction F (F ‹1) of the task by a factor S (S›1) and the remainder of the task is unaffected



$$\text{ExTime w/ E} = \text{ExTime w/o E} \times ((1 - F) + F/S)$$

$$\text{Speedup w/ E} = \frac{1}{(1 - F) + F/S}$$

Challenges of Parallel Processing

- Application parallelism ⇒ primarily via new algorithms that have better parallel performance
- Long remote latency impact ⇒ both by architect and by the programmer
- For example, reduce frequency of remote accesses either by
  - Caching shared data (HW)
  - Restructuring the data layout to make more accesses local (SW)

Examples: Amdahl's Law

$$\text{Speedup w/ E} = 1 / ((1\text{-}F) + F/S)$$

- Consider an enhancement which runs 20 times faster but which is only usable 25% of the time.
  - Speedup w/ E =
- What if it's usable only 15% of the time?
  - Speedup w/ E =
- Amdahl's Law tells us that to achieve linear speedup with 100 processors, none of the original computation can be scalar!
- To get a speedup of 99 from 100 processors, the percentage of the original program that could be scalar would have to be 0.01% or less

Challenges of Parallel Processing

- Second challenge is long latency to remote memory
- Suppose 32 CPU MP, 2 GHz, 200 ns remote memory, all local accesses hit memory hierarchy and base CPI is 0.5. (Remote access = 200/0.5 = 400 clock cycles.)
- What is performance impact if 0.2% instructions involve remote access?
  - 1.5X
  - 2.0X
  - 2.5X

Challenges of Parallel Processing

- Application parallelism ⇒ primarily via new algorithms that have better parallel performance
- Long remote latency impact ⇒ both by architect and by the programmer
- For example, reduce frequency of remote accesses either by
  - Caching shared data (HW)
  - Restructuring the data layout to make more accesses local (SW)

How hard is parallel programming anyway?

- Parallel Programmer Productivity: A Case Study of Novice Parallel Programmers
  - Lorin Hochstein, Jeff Carver, Forrest Shull, Sima Asgari, Victor Basili, Jeffrey K. Hollingsworth, Marvin V. Zelkowitz
  - Supercomputing 2005

Why use students for testing?

- First, multiple students are routinely given the same assignment to perform, and thus we are able to conduct experiments in a way to control for the skills of specific programmers.
- Second, graduate students in a HPC class are fairly typical of a large class of novice HPC programmers who may have years of experience in their application domain but very little in HPC-style programming.
- Finally, due to the relatively low costs, student studies are an excellent environment to debug protocols that might be later used on practicing HPC programmers.

Tests run

	Serial	MPI	OpenMP	Co-Array Fortran	StarP	XMT
Nearest-Neighbor Type Problems						
Game of Life	C3A3	C3A3 C0A1 C1A1	C3A3			
Grid of Resistors	C2A2	C2A2	C2A2		C2A2	
Sharks & Fishes		C6A2	C6A2	C6A2		
Laplace's Eq.		C2A3			P2A3	
SWIM			C0A2			
Broadcast Type Problems						
LU Decomposition			C4A1			
Parallel Mat-vec					C3A4	
Quantum Dynamics		C7A1				
Embarrassingly Parallel Type Problems						
Buffon-Laplace Nee-dle		C2A1 C3A1	C2A1 C3A1		C2A1 C3A1	
(Miscellaneous Problem Types)						
Parallel Sorting		C3A2	C3A2		C3A2	
Array Compaction						C5A1
Randomized Selection						C5A2

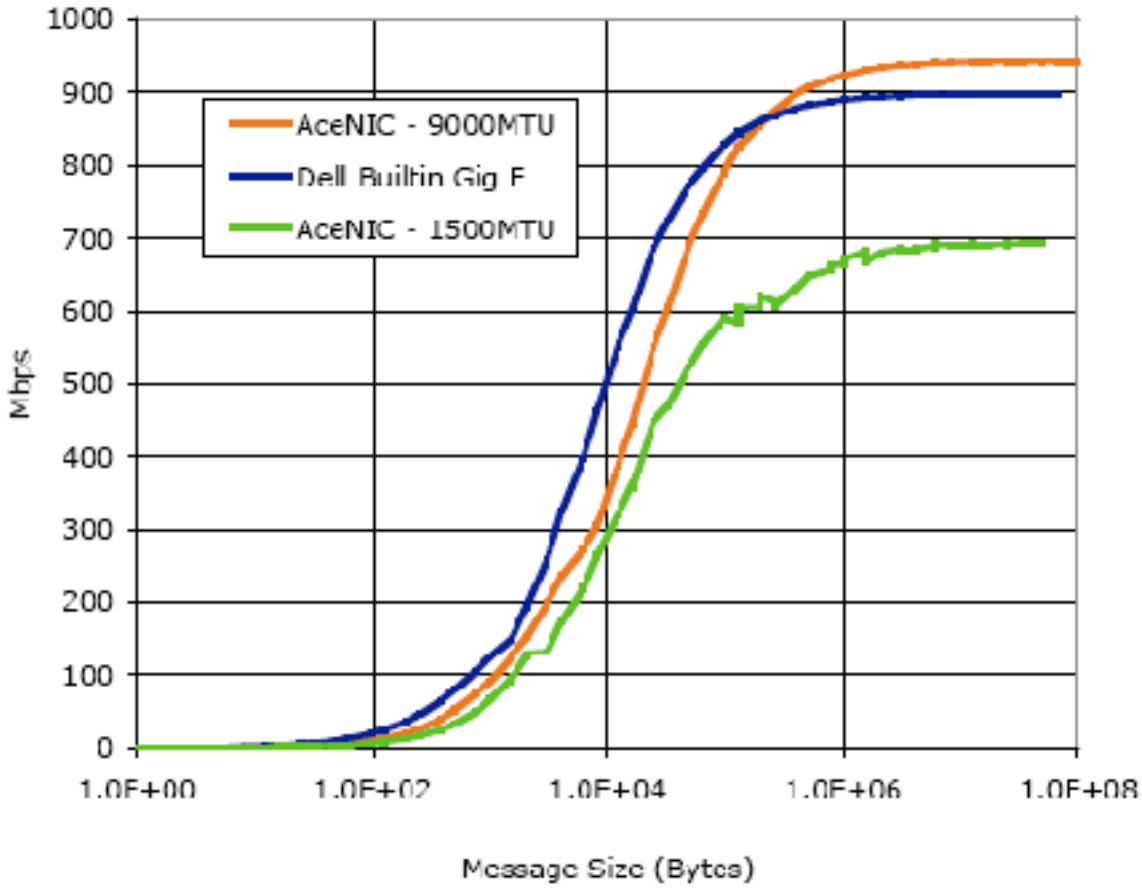
What They Learned

- Novices are able to achieve speedup on a parallel machine.
- MPI and OpenMP both require more {code, cost per line, effort} than serial implementations
  - MPI takes more effort than OpenMP

Data set	**Programming Model**	**Speedup on 8 processors**
Speedup w.r.t. serial version		
C1A1	MPI	mean 4.74, sd 1.97, n=2
C3A3	MPI	mean 2.8, sd 1.9, n=3
C3A3	OpenMP	mean 6.7, sd 9.1, n=2
Speedup w.r.t. parallel version run on 1 processor		
C0A1	MPI	mean 5.0, sd 2.1, n=13
C1A1	MPI	mean 4.8, sd 2.0, n=3
C3A3	MPI	mean 5.6, sd 2.5, n=5
C3A3	OpenMP	mean 5.7, sd 3.0, n=4

Ethernet Performance

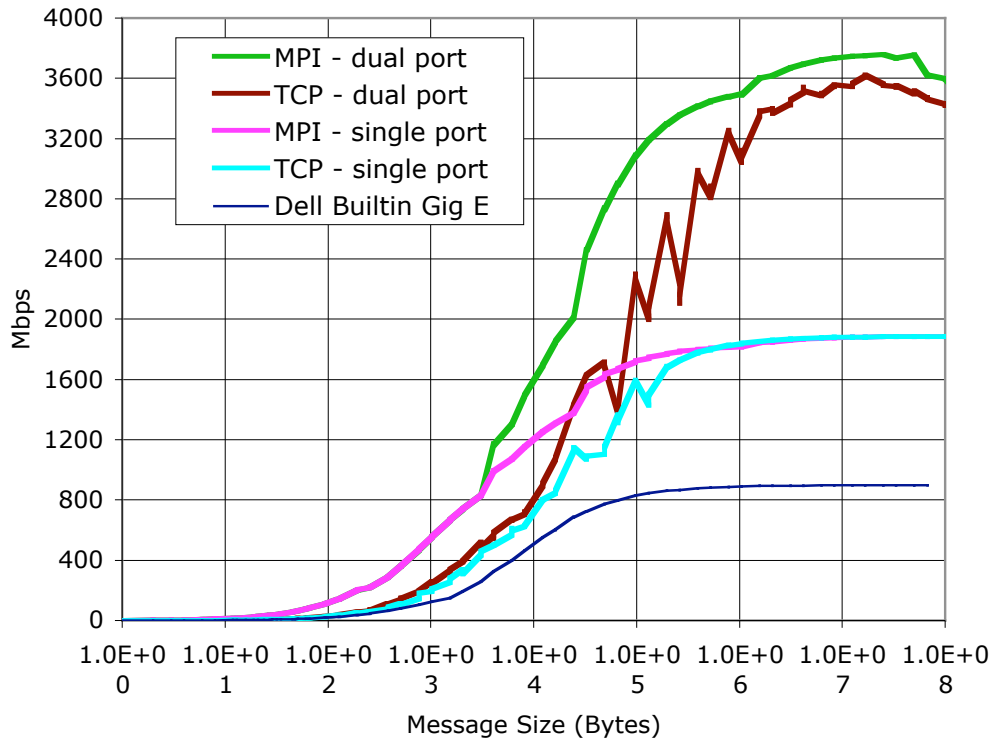
- Achieves close to theoretical bw even with default 1500 B/ message
- Broadcom part: 31 µs latency



Performance Characteristics of Dual-Processor HPC Cluster Nodes Based on 64-bit Commodity Processors, Purkayastha et al.

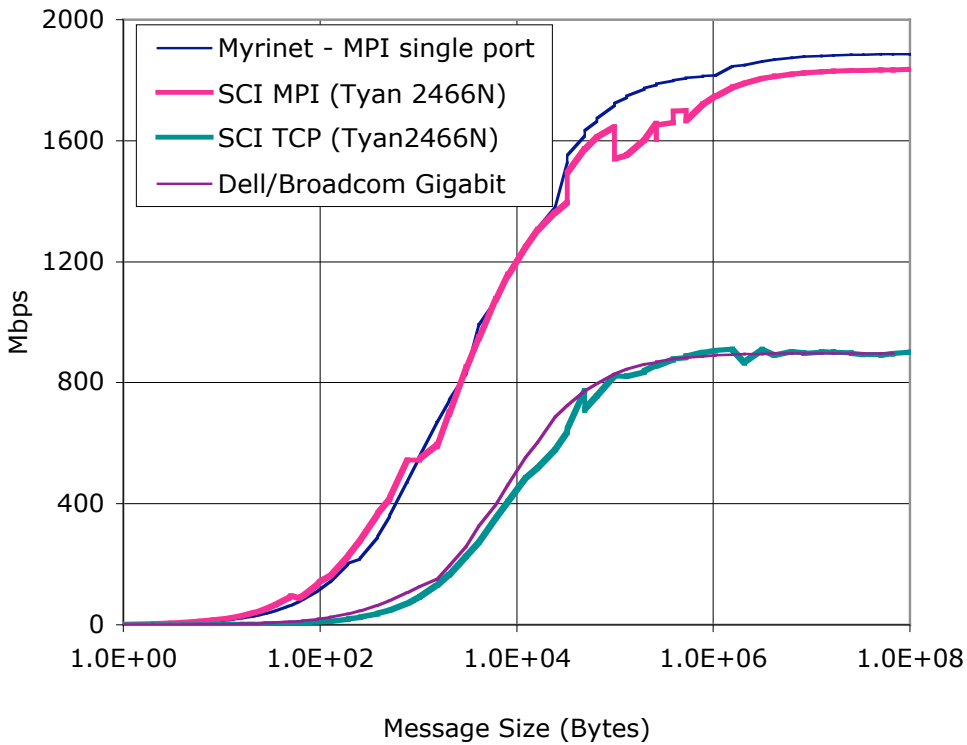
Myrinet

- Interconnect designed with clustering in mind
- 2 Gbps links, possibly 2 physical links (so 4 Gb/s)
- $850/node up to 128 nodes, $1737/node up to 1024 nodes
- MPI latency 6–7 µs
- TCP/IP latency 27–30 µs
  - Strong support!



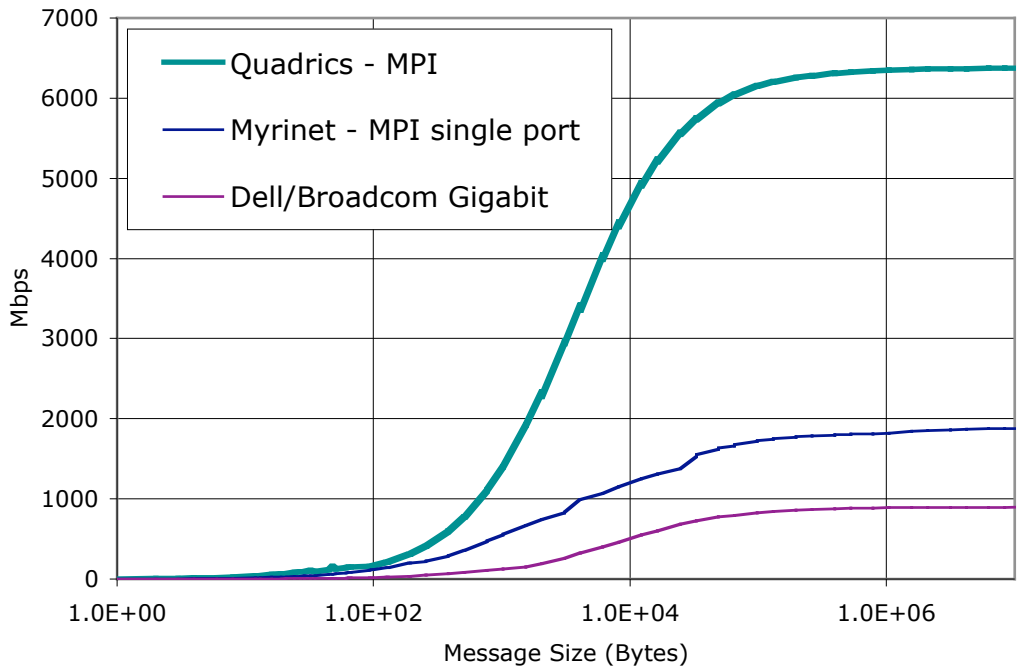
Scalable Coherent Interface

- Not a “switched” interconnect
- Max scalability: 64–100 nodes for 2D torus ($1095/node), 640–1000 for 3D torus ($1595/node)
- MPI: 4 µs latency, 1830 Mbps
- TCP/IP: 900 Mbps



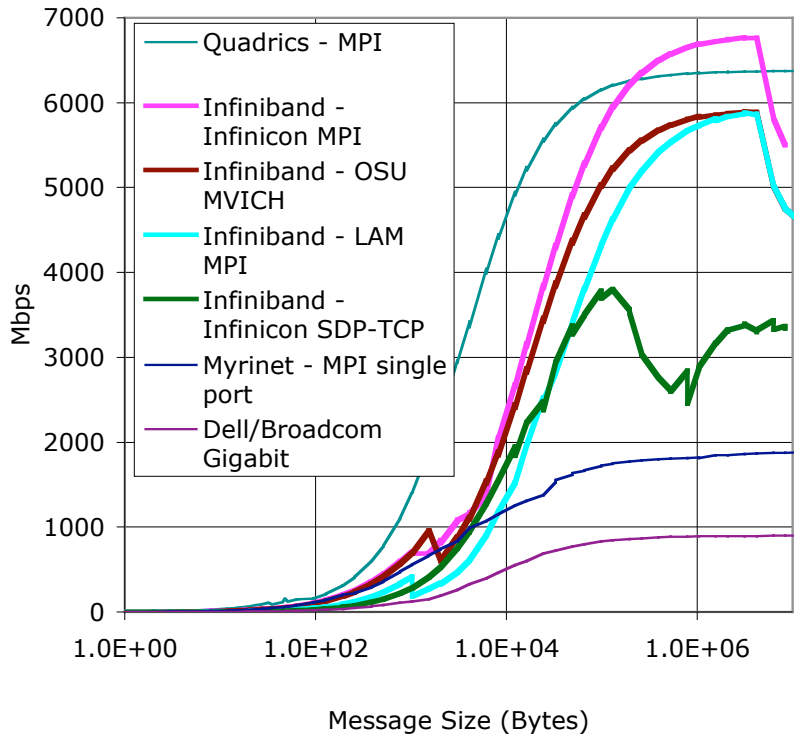
Quadrics

- “a premium interconnect choice on high-end systems such as the Compaq AlphaServer SC”
- Max 4096 ports
- $2400/port for small system up to $4078/port for 1024 node system
- MPI: 2–3 µs latency, 6370 Mpbs bw



Infiniband

- Defined by industry consortium
  - Scalable—base is 2.5 Gb/s link, scales to 30 Gb/s
  - Current parts are 4x (10 Gb/s)
- $1200–1600/node
- Also used as system interconnect
- 6750 Mb/s, 6–7 µs



10 Gb Ethernet

- Similar tradeoffs to previous Ethernets
- $10k per switch port
- Only 3700 Mb/s