A Crash course to (The) Bighouse

Brock Palen
brockp@umich.edu

CAEN Brown Bag, Oct 10th

Outline

1. Resources
   - Configuration
   - Hardware
2. Architecture
   - ccNUMA
   - Altix 4700 Brick
   - Dual Fat Tree
   - cpu sets
   - NUMA Effects
3. Software Performance
   - MPI Code
   - OpenMP Code

Hardware: bighouse

Bighouse

- bighouse is our Itanium SMP machine;
- Login: `bighouse.engin.umich.edu`
- Shares nyx's 6TB NFS file system
- Running SUsE Linux Enterprise Server 10
- ProPack 5 from SGI

Bighouse Hardware

Current Hardware

- Cache Coherency NonUniform Memory Access (ccNUMA)
- 16 CPU, 32 core Intel Itanium II's
- Measured 5.5 Gflop/cpu running 4 way
- 171.9 Gflop running 32 way
- 96 GB Ram
- Max 41 GB/s Aggregate Memory bandwidth
- NUMAlink4 3.2GByte/s, 1μ Second Latency

Bighouse Hardware

Current Hardware

- Cache Coherency NonUniform Memory Access (ccNUMA)
- 16 CPU, 32 core Intel Itanium II's
- Measured 5.5 Gflop/cpu running 4 way
- 171.9 Gflop running 32 way
- 96 GB Ram
- Max 41 GB/s Aggregate Memory bandwidth
- NUMAlink4 3.2GByte/s, 1μ Second Latency

Bighouse Hardware

Current Hardware

- Cache Coherency NonUniform Memory Access (ccNUMA)
- 16 CPU, 32 core Intel Itanium II's
- Measured 5.5 Gflop/cpu running 4 way
- 171.9 Gflop running 32 way
- 96 GB Ram
- Max 41 GB/s Aggregate Memory bandwidth
- NUMAlink4 3.2GByte/s, 1μ Second Latency

Bighouse Hardware

Current Hardware

- Cache Coherency NonUniform Memory Access (ccNUMA)
- 16 CPU, 32 core Intel Itanium II's
- Measured 5.5 Gflop/cpu running 4 way
- 171.9 Gflop running 32 way
- 96 GB Ram
- Max 41 GB/s Aggregate Memory bandwidth
- NUMAlink4 3.2GByte/s, 1μ Second Latency

Bighouse Hardware

Current Hardware

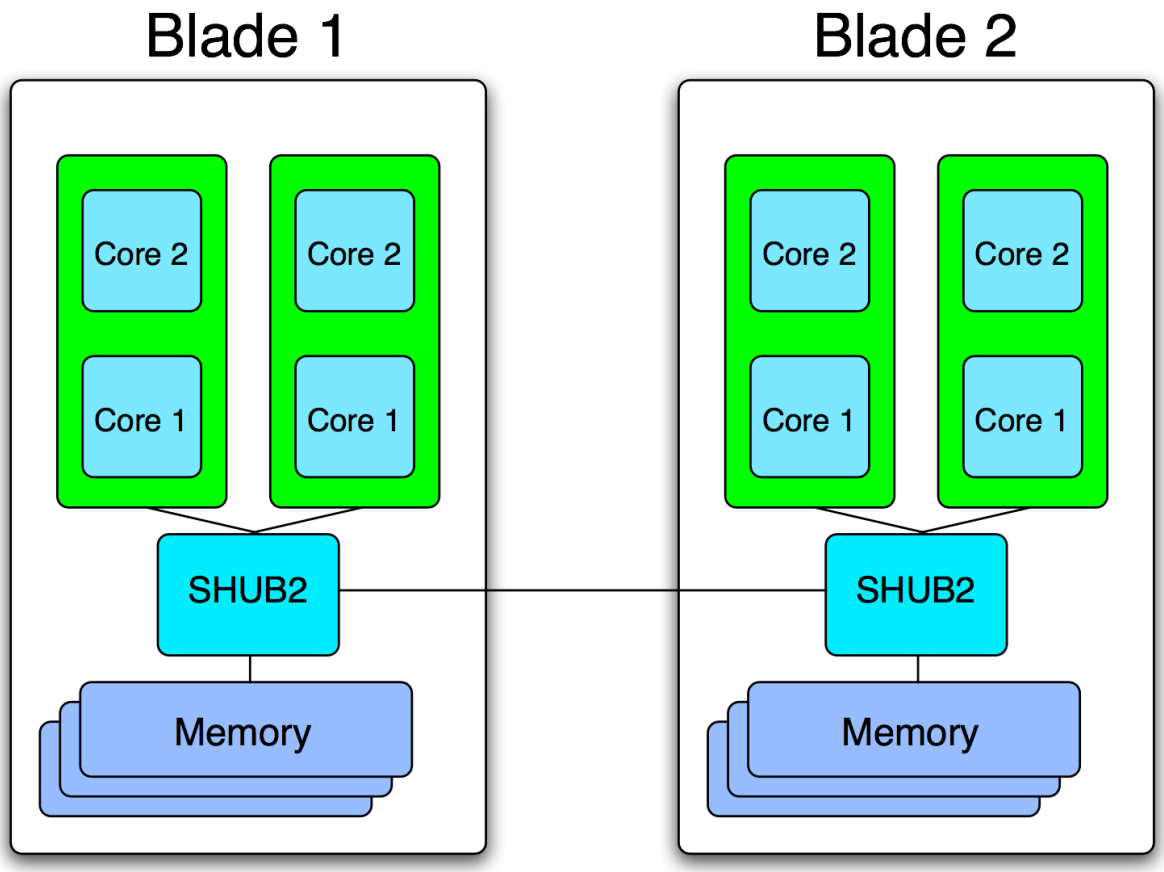
- Cache Coherency NonUniform Memory Access (ccNUMA)
- 16 CPU, 32 core Intel Itanium II's
- Measured 5.5 Gflop/cpu running 4 way
- 171.9 Gflop running 32 way
- 96 GB Ram
- Max 41 GB/s Aggregate Memory bandwidth
- NUMAlink4 3.2GByte/s, 1μ Second Latency

Bighouse Hardware

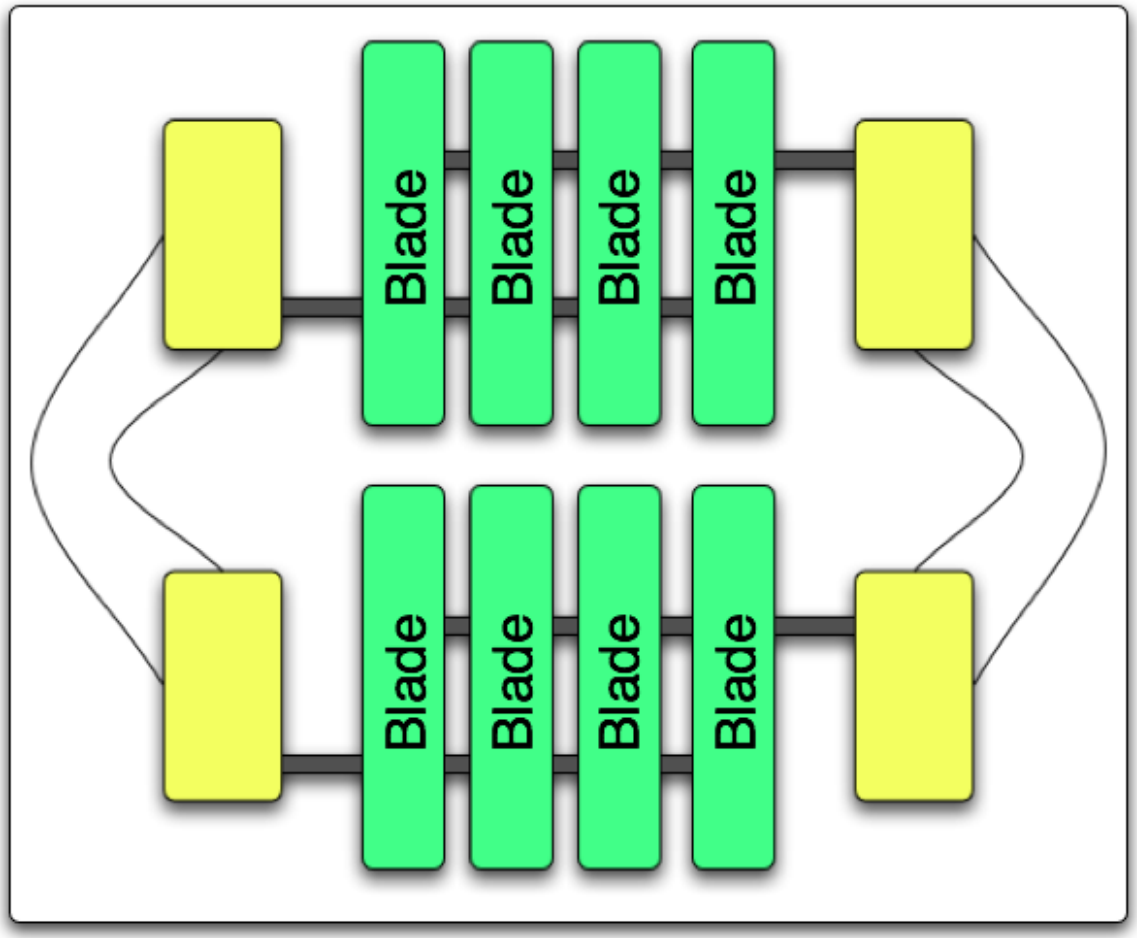
Current Hardware

- Cache Coherency NonUniform Memory Access (ccNUMA)
- 16 CPU, 32 core Intel Itanium II's
- Measured 5.5 Gflop/cpu running 4 way
- 171.9 Gflop running 32 way
- 96 GB Ram
- Max 41 GB/s Aggregate Memory bandwidth
- NUMAlink4 3.2GByte/s, 1μ Second Latency

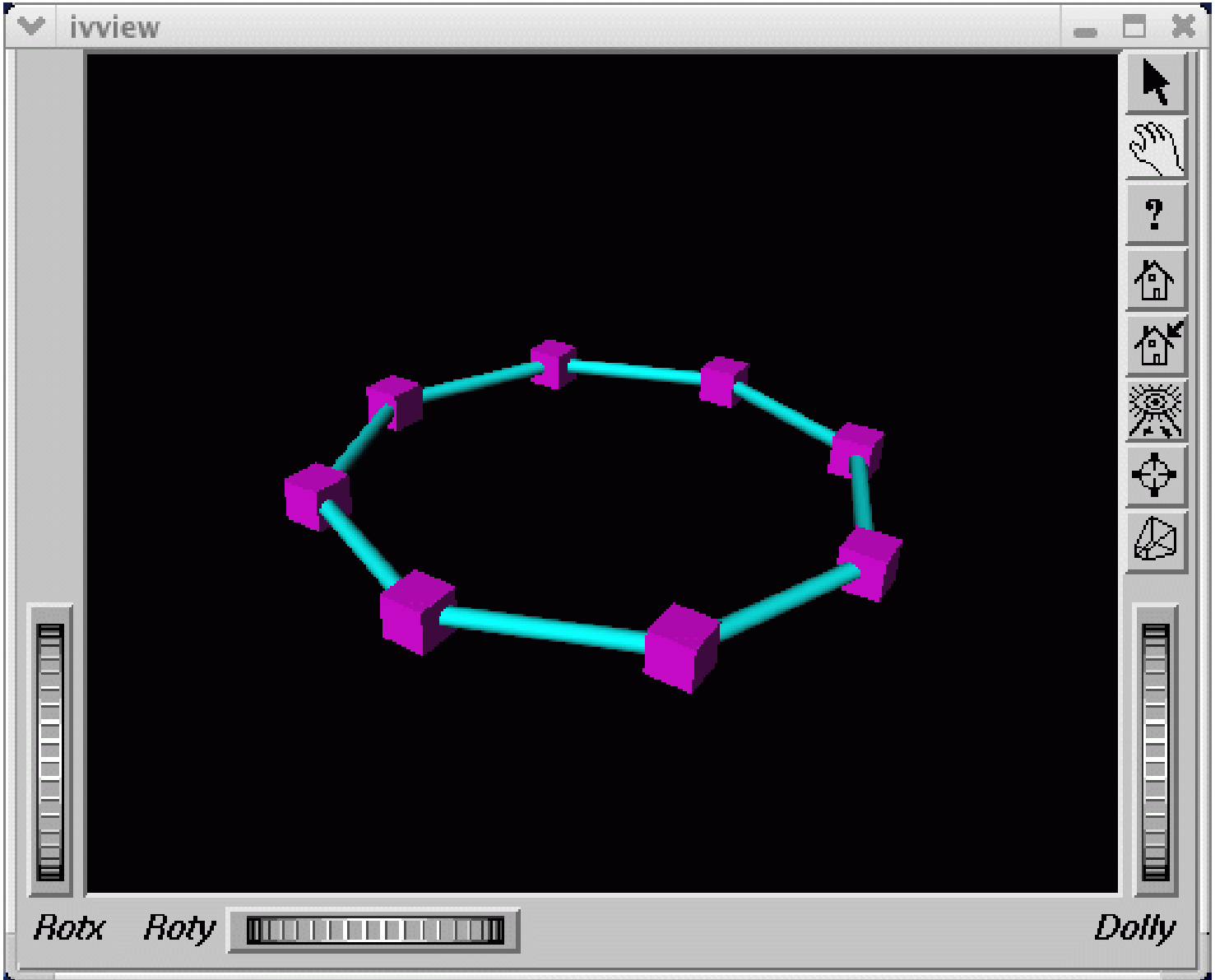
ccNUMA


Blade 1
Blade 2
Core 2
Core 2
Core 1
Core 1
Core 2
Core 2
Core 1
Core 1
SHUB2
SHUB2
Memory
Memory
NUMAlink 3.2GB/s

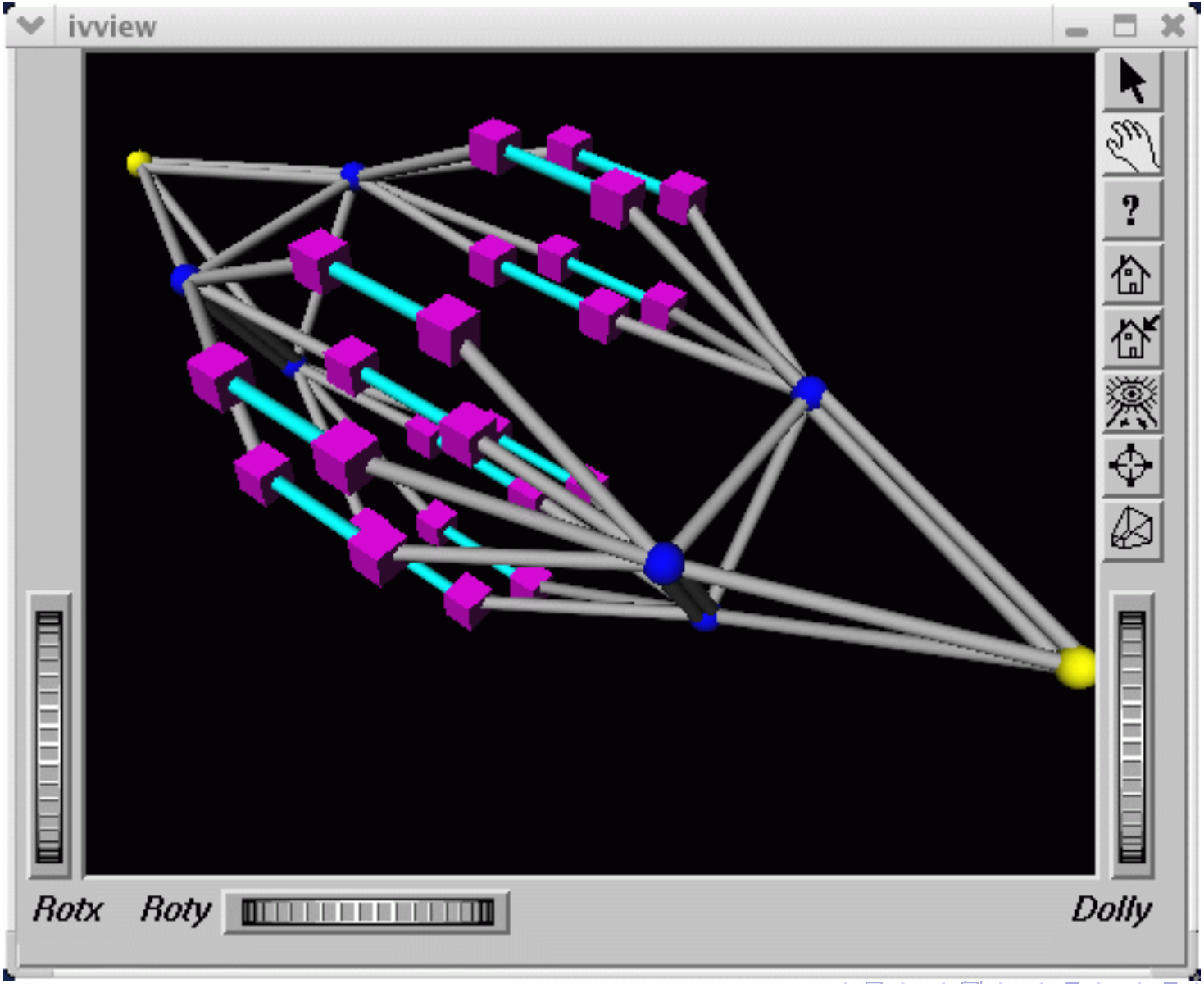
Altix 4700 Brick



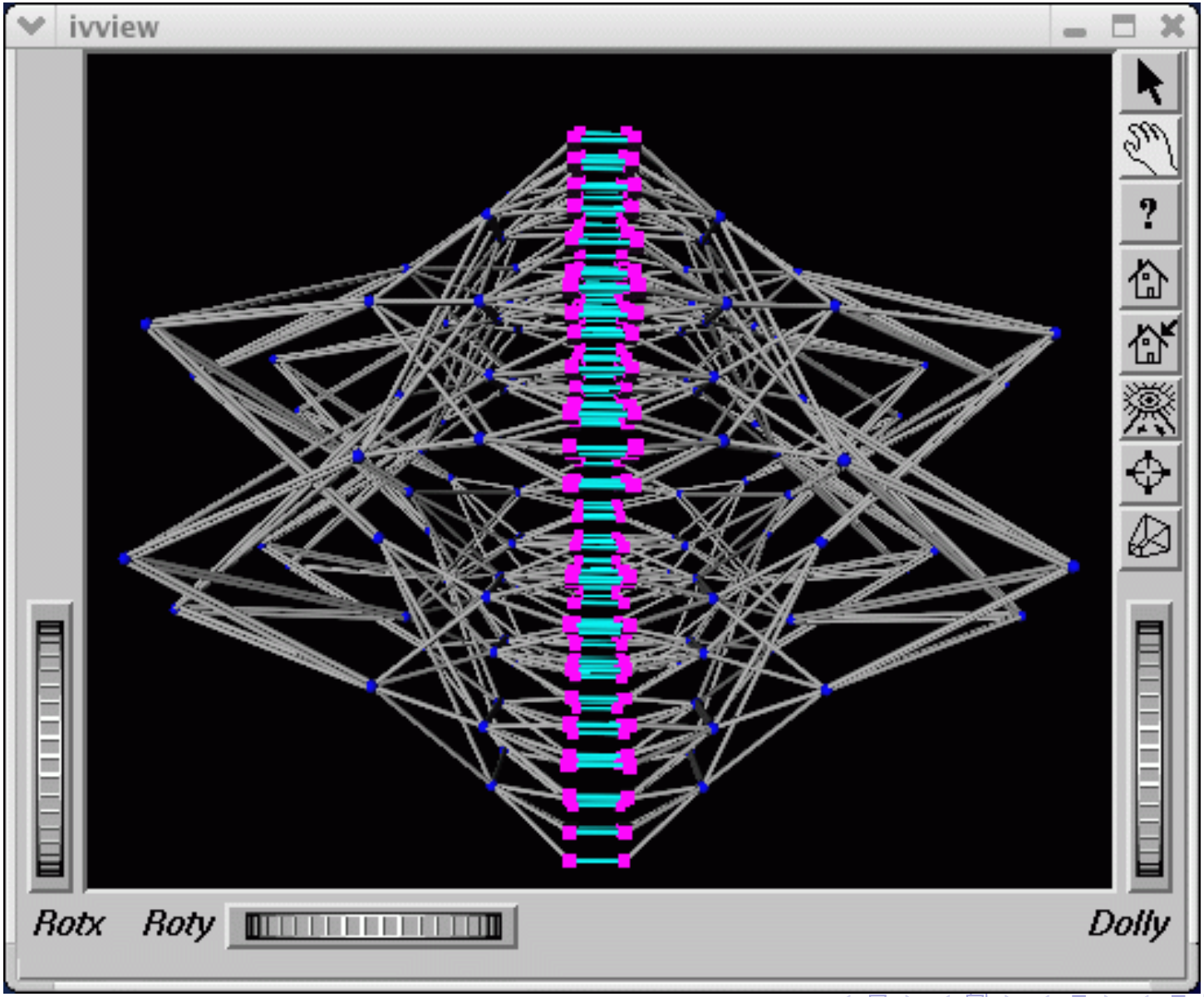
Dual Fat Tree



Dual Fat Tree



Dual Fat Tree



MPT

What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

MPT

What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

MPT

What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

MPT

What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

MPT

What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

MPT

What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

MPT

What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

MPT

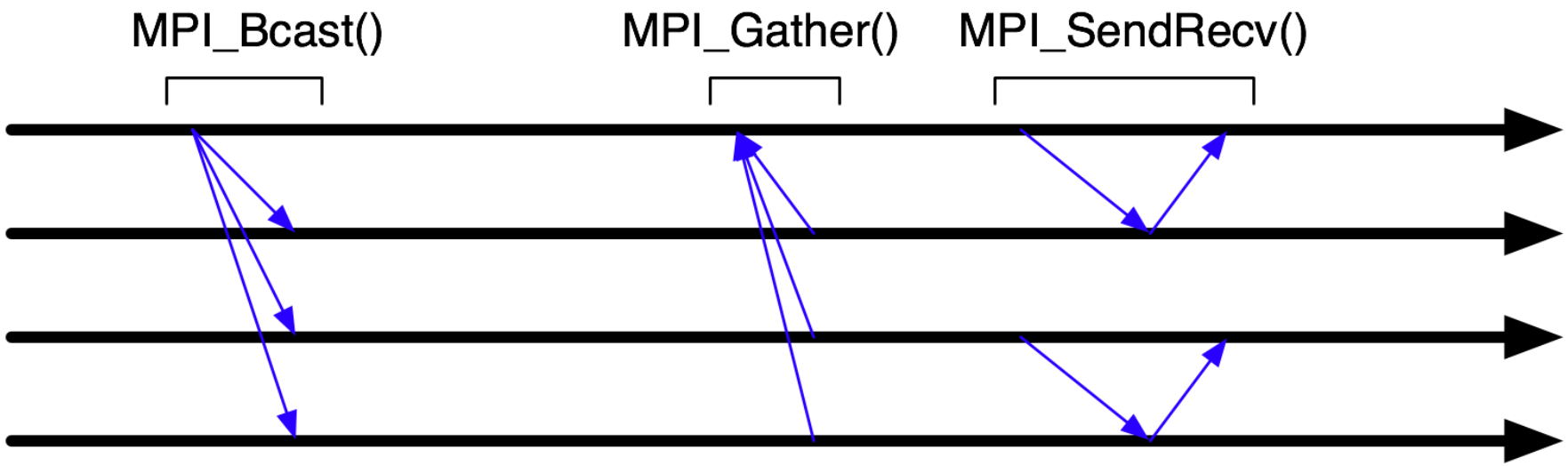
What is MPI?

- Message Passing Interface
- DMP Distributed Memory Paralell
- Hard to Program, Uses Function calls
- Hardware is Cheap
- Scales to 1000's of CPUS (Bluegene/L)

MPT

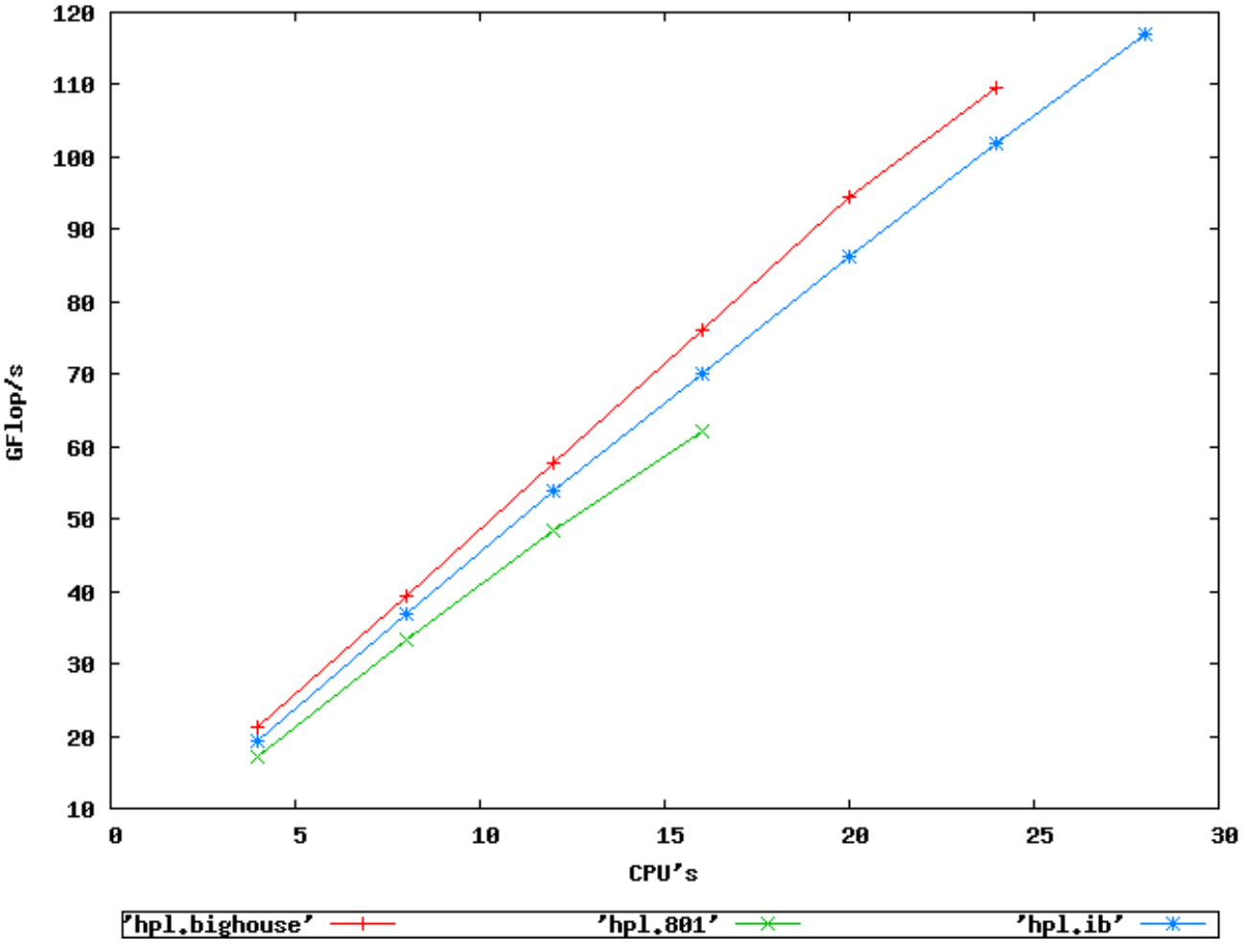
- MPT SGI MPI-1/2 Implementation
- Makes Strong Use of NUMAlink
- Lots of Copy on Write

What is MPI?



The Challenger

MPI Perfromance/HPL



OpenMP

OpenMP

- Shared Memory Parallel
- Easy to Program, Uses Pragmas
- Hardware is Expensive and Proprietary
- Can Solve Any Problem (DMP or SMP)
- Scaling Issues, Hybrid Programming
- More important with Dual/Quad/Many Core CPU's

OpenMP

OpenMP

- Shared Memory Parallel
- Easy to Program, Uses Pragmas
- Hardware is Expensive and Proprietary
- Can Solve Any Problem (DMP or SMP)
- Scaling Issues, Hybrid Programming
- More important with Dual/Quad/Many Core CPU's

OpenMP

OpenMP

- Shared Memory Parallel
- Easy to Program, Uses Pragmas
- Hardware is Expensive and Proprietary
- Can Solve Any Problem (DMP or SMP)
- Scaling Issues, Hybrid Programming
- More important with Dual/Quad/Many Core CPU's

OpenMP

OpenMP

- Shared Memory Parallel
- Easy to Program, Uses Pragmas
- Hardware is Expensive and Proprietary
- Can Solve Any Problem (DMP or SMP)
- Scaling Issues, Hybrid Programming
- More important with Dual/Quad/Many Core CPU's

OpenMP

OpenMP

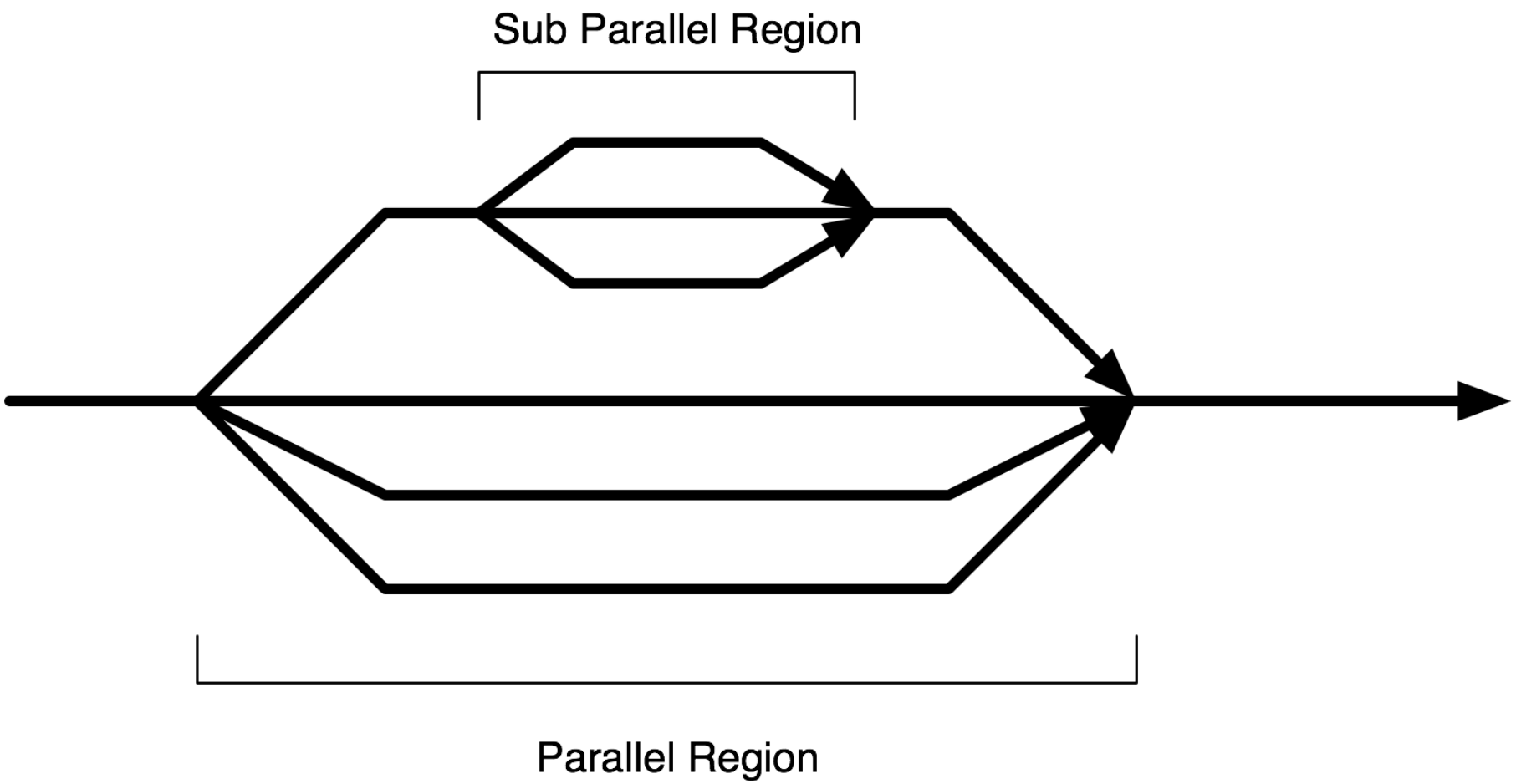
- Shared Memory Parallel
- Easy to Program, Uses Pragmas
- Hardware is Expensive and Proprietary
- Can Solve Any Problem (DMP or SMP)
- Scaling Issues, Hybrid Programming
- More important with Dual/Quad/Many Core CPU's

OpenMP

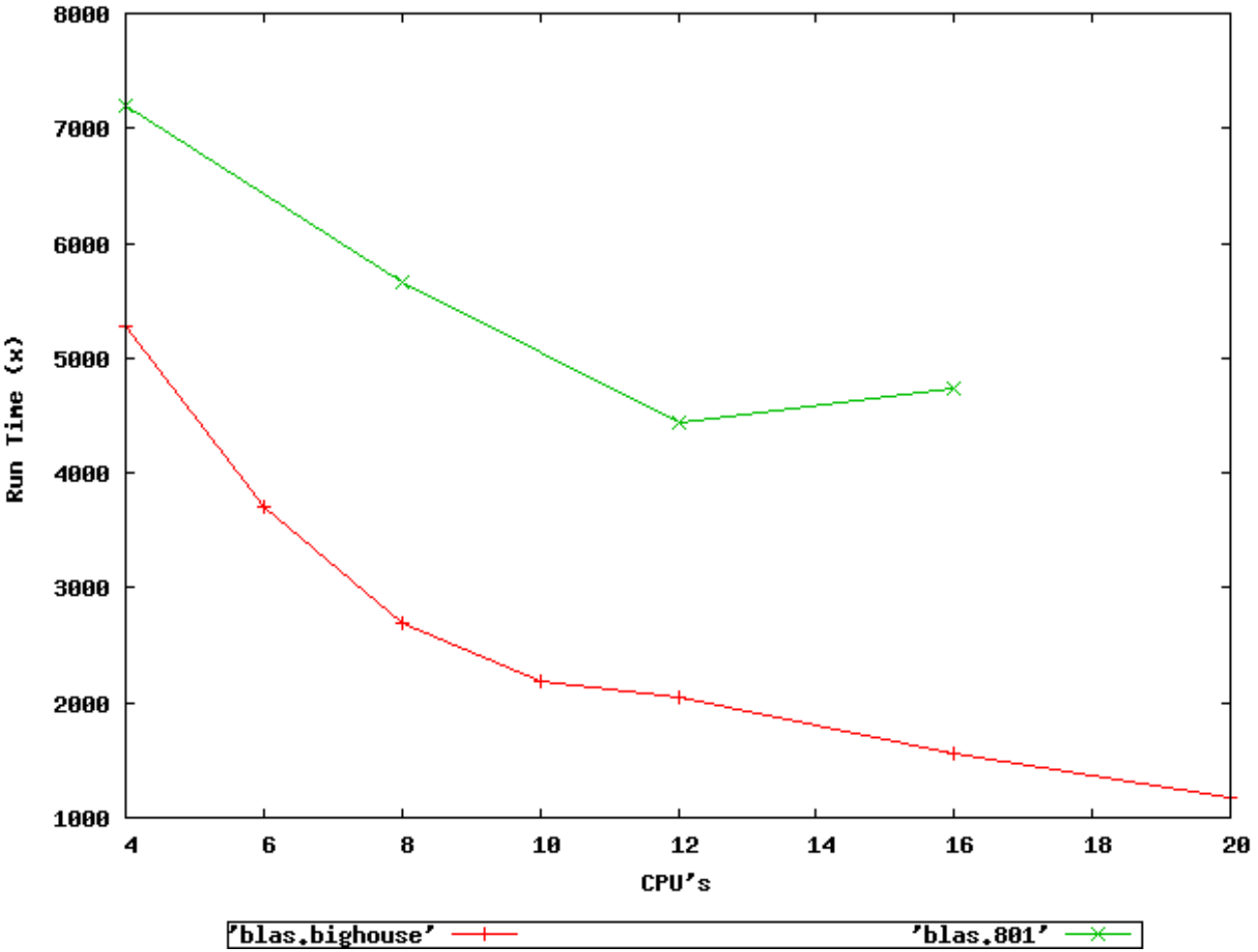
OpenMP

- Shared Memory Parallel
- Easy to Program, Uses Pragmas
- Hardware is Expensive and Proprietary
- Can Solve Any Problem (DMP or SMP)
- Scaling Issues, Hybrid Programming
- More important with Dual/Quad/Many Core CPU's

OpenMP Fork and Join



OpenMP Performance/dgemm 36,621 MByte



Example Cases

Example Cases

- NUMA Memory Placement `dlook(1)` `dplace(1)` `cpuset(1)`
- Example Cpuset OH NO SWAP
- Memory placement ccNUMA Knows where to put memory (numa_hit numa_miss)
- Example `stream.c` measures memory bandwidth

Example Cases

Example Cases

- NUMA Memory Placement `dlook(1)` `dplace(1)` `cpuset(1)`
- Example Cpuset OH NO SWAP
- Memory placement ccNUMA Knows where to put memory (numa_hit numa_miss)
- Example `stream.c` measures memory bandwidth

Example Cases

Example Cases

- NUMA Memory Placement `dlook(1)` `dplace(1)` `cpuset(1)`
- Example Cpuset OH NO SWAP
- Memory placement ccNUMA Knows where to put memory (numa_hit numa_miss)
- Example `stream.c` measures memory bandwidth

Example Cases

Example Cases

- NUMA Memory Placement `dlook(1)` `dplace(1)` `cpuset(1)`
- Example Cpuset OH NO SWAP
- Memory placement ccNUMA Knows where to put memory (numa_hit numa_miss)
- Example `stream.c` measures memory bandwidth

Questions

Questions?

Questions?

`http://cac.engin.umich.edu/resources/bighouse.html`

`cac-support@umich.edu`