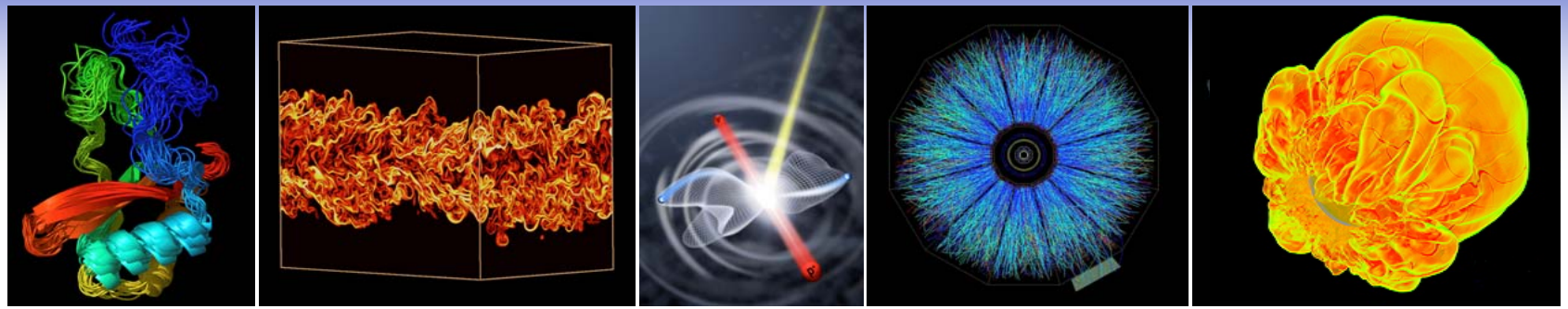


Application Requirements for Interconnects

IAA Interconnects Workshop

San Jose, California
July 21-22, 2008

Office of Science
U.S. DEPARTMENT OF ENERGY

NERSC

BERKELEY LAB

Questions

(How do we determine appropriate interconnect requirements?)

- **Topology:** ***will the apps inform us what kind of topology to use?***
  - **Crossbars: Not scalable**
  - **Fat-Trees: Cost scales superlinearly with number of processors**
  - **Lower Degree Interconnects:** ***(n-Dim Mesh, Torus, Hypercube, Cayley)***
    - **Costs scale linearly with number of processors**
    - **Problems with application mapping/scheduling fault tolerance**

- **Bandwidth/Latency/Overhead**
  - **Which is most important?** ***(trick question: they are intimately connected)***
  - **Requirements for a "balanced" machine?** ***(eg. performance is not dominated by communication costs)***

- **Collectives**
  - **How important/what type?**
  - **Do they deserve a dedicated interconnect?**
  - **Should we put floating point hardware into the NIC?**

IPM *(the "hammer")*

Integrated Performance Monitoring

- **portable, lightweight, scalable profiling**
- **fast hash method**
- **profiles MPI topology**
- **profiles code regions**
- **open source**

```
MPI_Pcontrol(1,"W");
 …code…
MPI_Pcontrol(-1,"W");
```

```
##########################################
# IPMv0.7 :: csnode041 256 tasks  ES/ESOS
# madbench.x (completed) 10/27/04/14:45:56
#
#       <mpi><user><wall> (sec)
#      171.67      352.16      393.80
# …
#############################################
# W
#       <mpi><user><wall> (sec)
#       36.40      198.00      198.36
#
# call              [time]      %mpi   %wall
# MPI_Reduce      2.395e+01   65.8     6.1
# MPI_Recv        9.625e+00   26.4     2.4
# MPI_Send        2.708e+00    7.4     0.7
# MPI_Testall     7.310e-02    0.2     0.0
# MPI_Isend       2.597e-02    0.1     0.0
#############################################
…
```

Developed by David Skinner, NERSC

Application Overview *(the "nails")*

NAME	Discipline	Problem/Method	Structure
MADCAP	Cosmology	CMB Analysis	Dense Matrix
FVCAM	Climate Modeling	AGCM	3D Grid
CACTUS	Astrophysics	General Relativity	3D Grid
LBMHD	Plasma Physics	MHD	2D/3D Lattice
GTC	Magnetic Fusion	Vlasov-Poisson	Particle in Cell
PARATEC	Material Science	DFT	Fourier/Grid
SuperLU	Multi-Discipline	LU Factorization	Sparse Matrix
PMEMD	Life Sciences	Molecular Dynamics	Particle

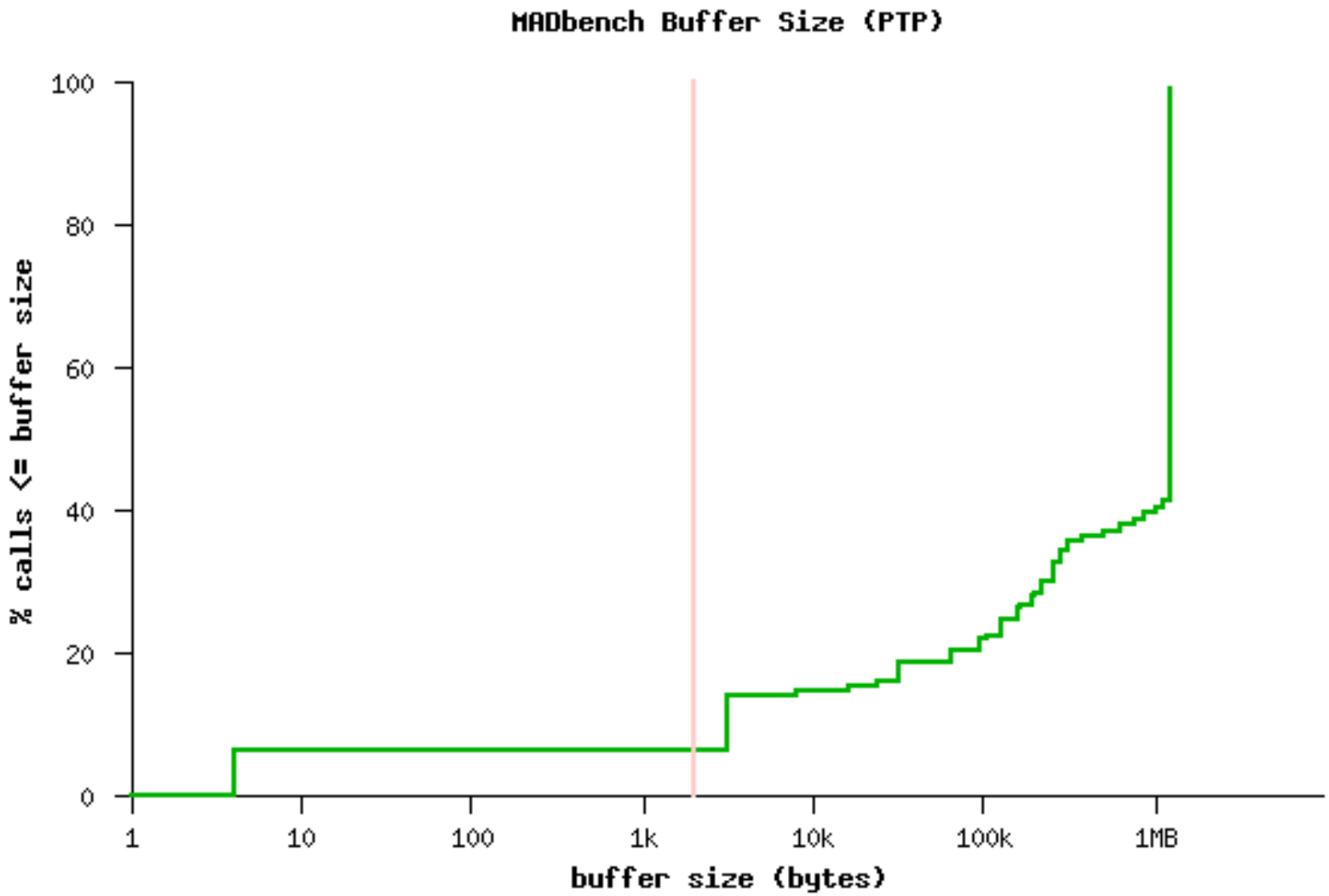
Latency Bound vs. Bandwidth Bound?

- How large does a message have to be in order to saturate a dedicated circuit on the interconnect?
  - $N^{1/2}$ from the early days of vector computing
  - Bandwidth Delay Product in TCP

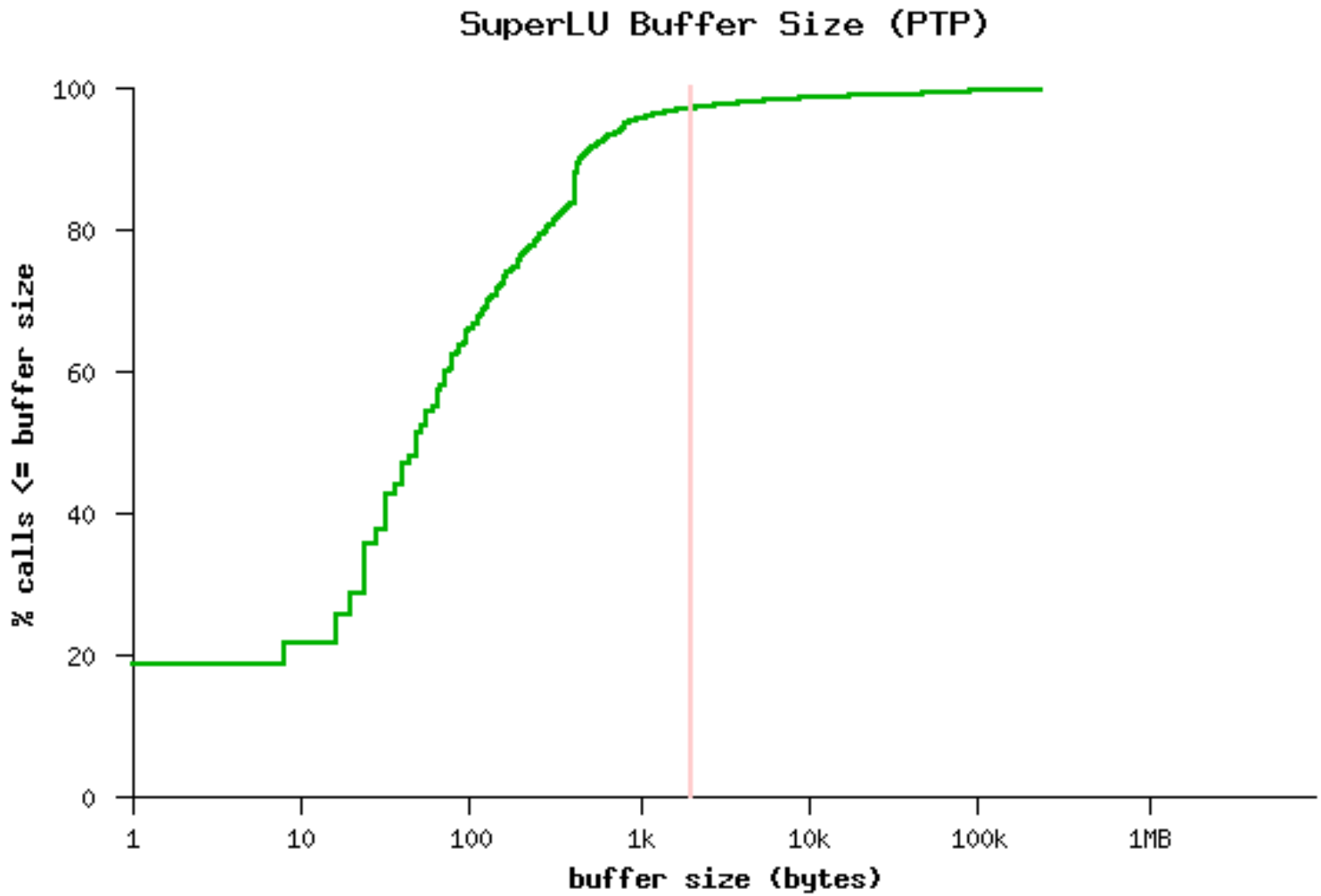
System	Technology	MPI Latency	Peak Bandwidth	Bandwidth Delay Product
SGI Altix	Numalink-4	1.1us	1.9GB/s	2KB
Cray X1	Cray Custom	7.3us	6.3GB/s	46KB
NEC ES	NEC Custom	5.6us	1.5GB/s	8.4KB
Myrinet Cluster	Myrinet 2000	5.7us	500MB/s	2.8KB
Infiniband x86	IB4x	1.7us	2GB/s	3.4KB

- Bandwidth Bound if msg size > Bandwidth*Delay
- Latency Bound if msg size < Bandwidth*Delay
  - Except if pipelined *(unlikely with MPI due to overhead)*
  - Cannot pipeline MPI collectives *(but can in Titanium)*

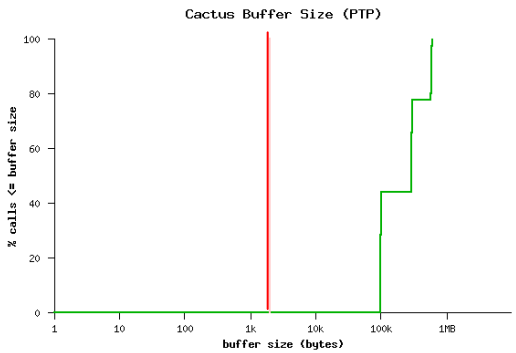
Diagram of Message Size Distribution Function

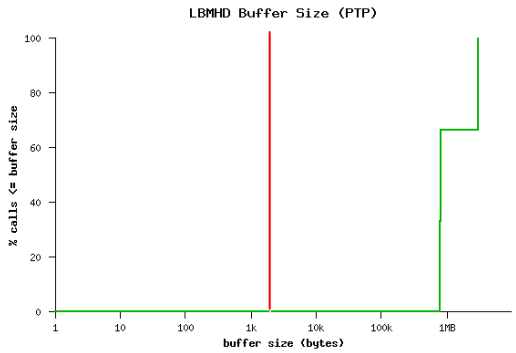


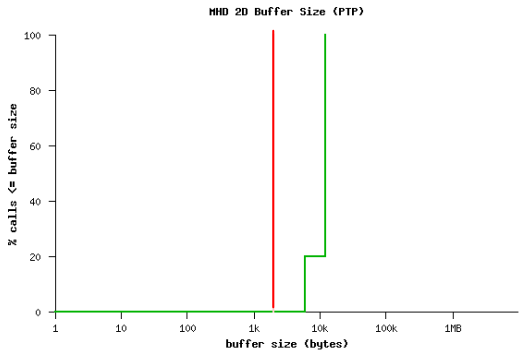
Message Size Distributions

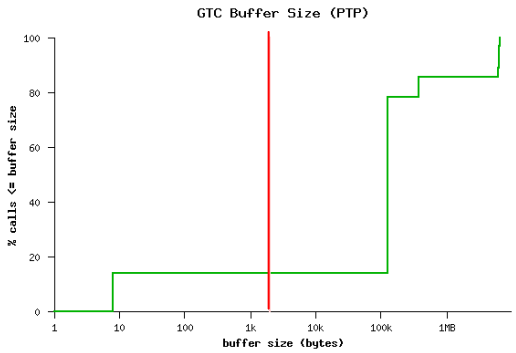


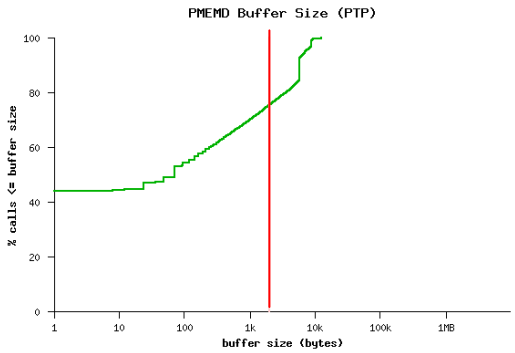
P2P Buffer Sizes

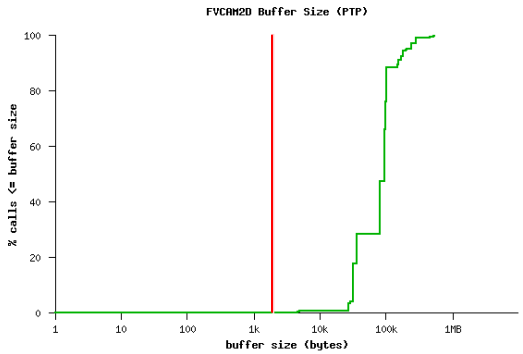


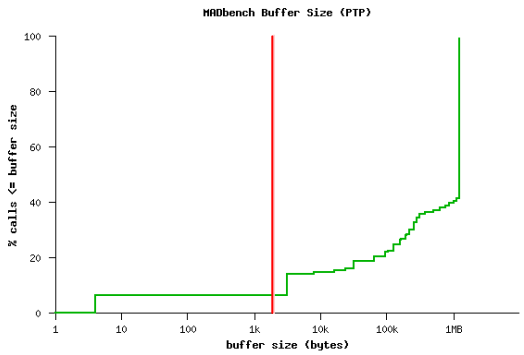


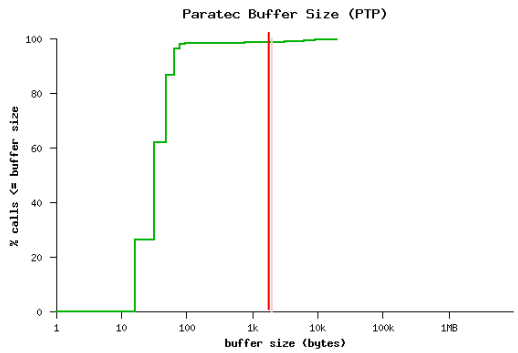


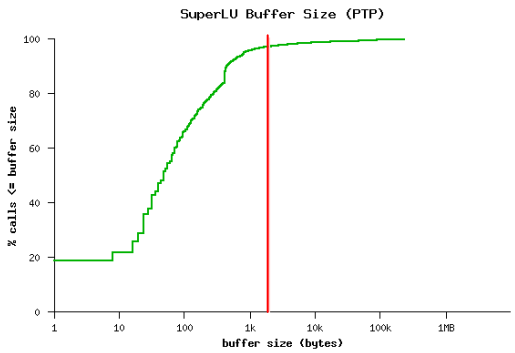




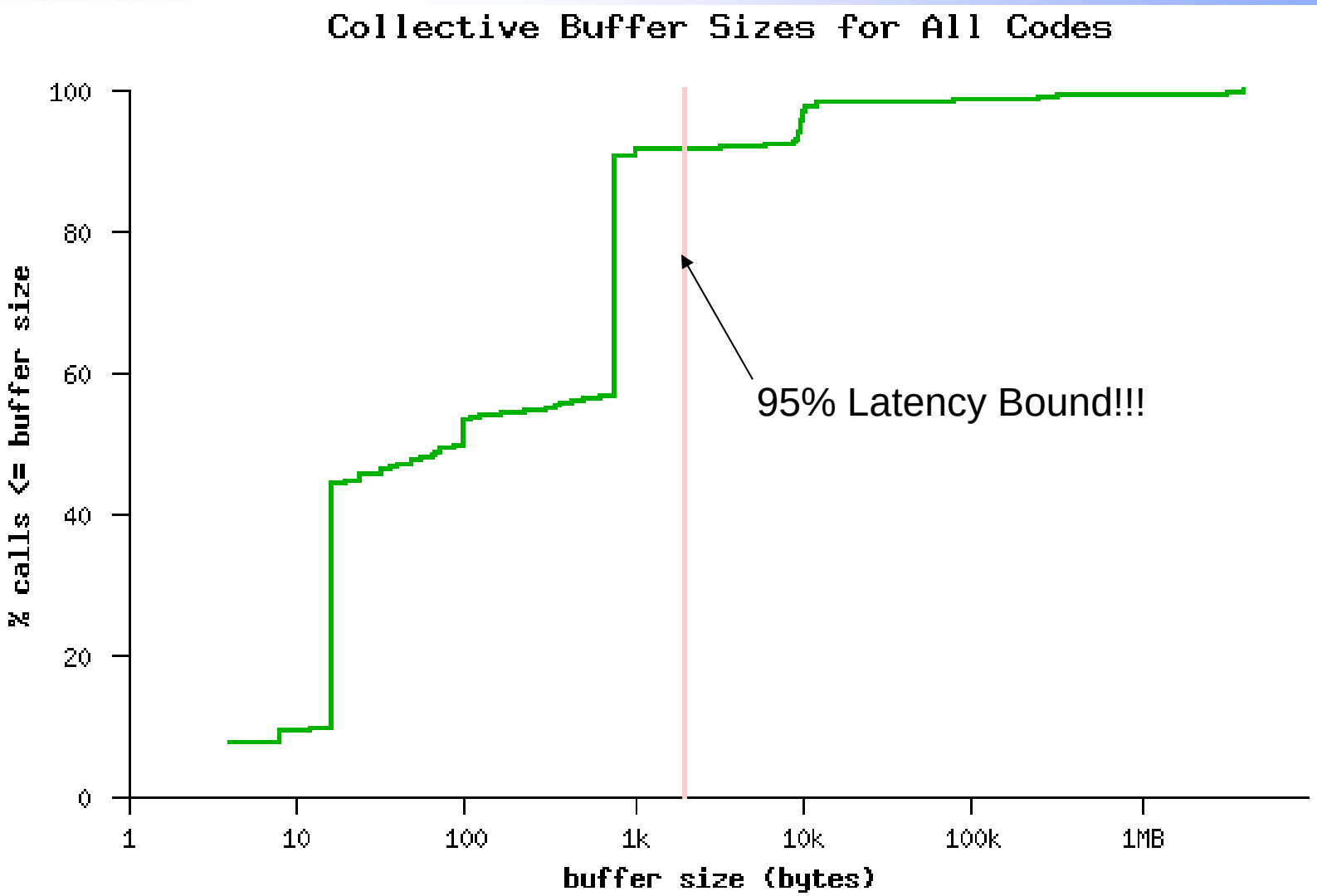






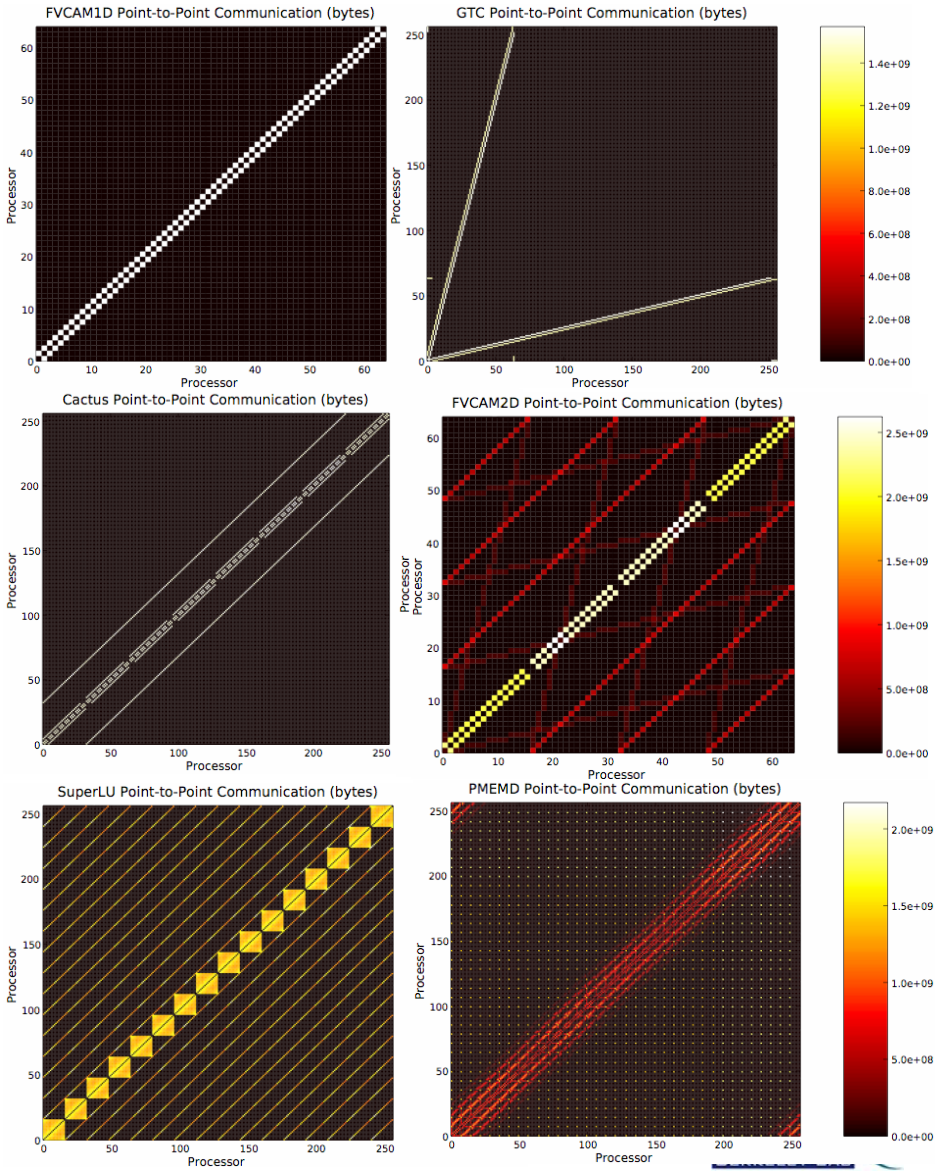


Collective Buffer Sizes



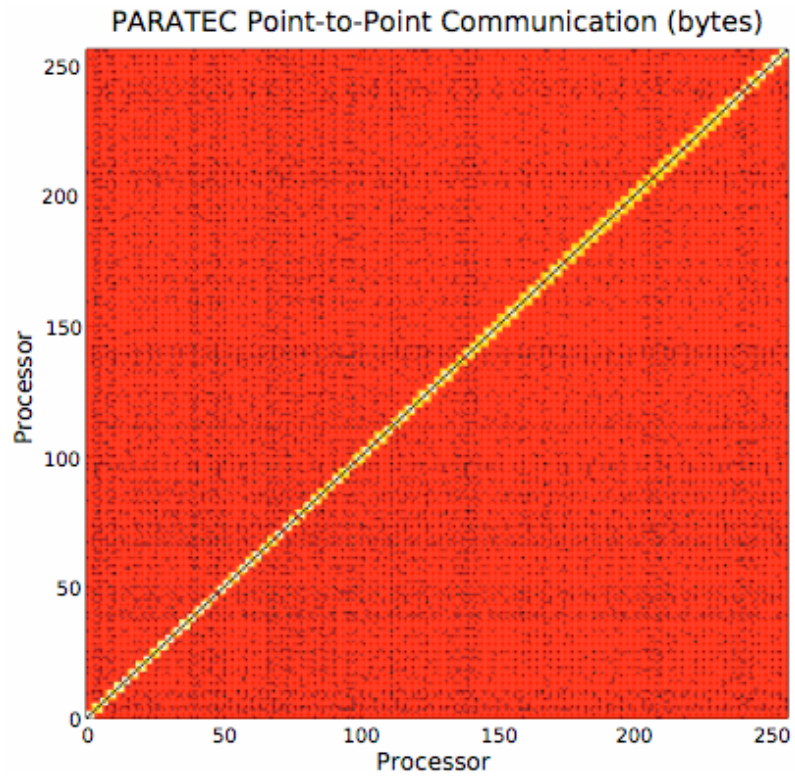
Interconnect Design Considerations for Message Passing Applications

- **Application studies provide insight to requirements for Interconnects (both on-chip and off-chip)**
  - On-chip interconnect is 2D planar (crossbar won't scale!)
  - Sparse connectivity for most apps.; crossbar is overkill
  - No single best topology
  - Most point-to-point message exhibit sparse topology + often bandwidth bound
  - Collectives tiny and primarily latency bound
- **Ultimately, need to be aware of the on-chip interconnect topology in addition to the off-chip topology**
  - Adaptive topology interconnects (HFAST)
  - Intelligent task migration?



Bisection Bandwidth

- **3D FFT easy-to-identify as needing high bisection**
  - Each processor must send messages to all PE's! (all-to-all) for 1D decomposition
  - However, most implementations are currently limited by overhead of sending small messages!
  - 2D domain decomposition (required for high concurrency) actually requires sqrt(N) communicating partners! *(some-to-some)*
  - *The issue is OVERHEAD (more of a limit than latency)*
- **Same Deal for AMR**
  - AMR communication is sparse, but limited by message overhead



The Future of HPC System Concurrency

Must ride exponential wave of increasing concurrency for forseeable future!

Fortunately, most of the concurrency growth is within a single socket

Strong-Scaling Drives Change in Algorithm Requirements

- **Parallel computing has thrived on weak-scaling for past 15 years**
- **Flat CPU performance increases emphasis on strong-scaling**
- **Focus on Strong Scaling will dramatically change interconnect requirements in the future!**
  - **Concurrency:** *Will double every 18 months*
  - **Implicit Methods***: Improve time-to-solution*
  - **Multiscale/AMR methods**: Only apply computation where it is required (both temporal and spatial refinement).
  - **Efficient Lightweight Messaging:** *All of these trends will push point-to-point messaging towards smaller message sizes.*

Where to Find 12 Orders in 10 years?

Jardin& Keyes

Hardware: 3

- ~~1.5 orders: increased processor speed and efficiency~~
- 1.5 orders: increased concurrency

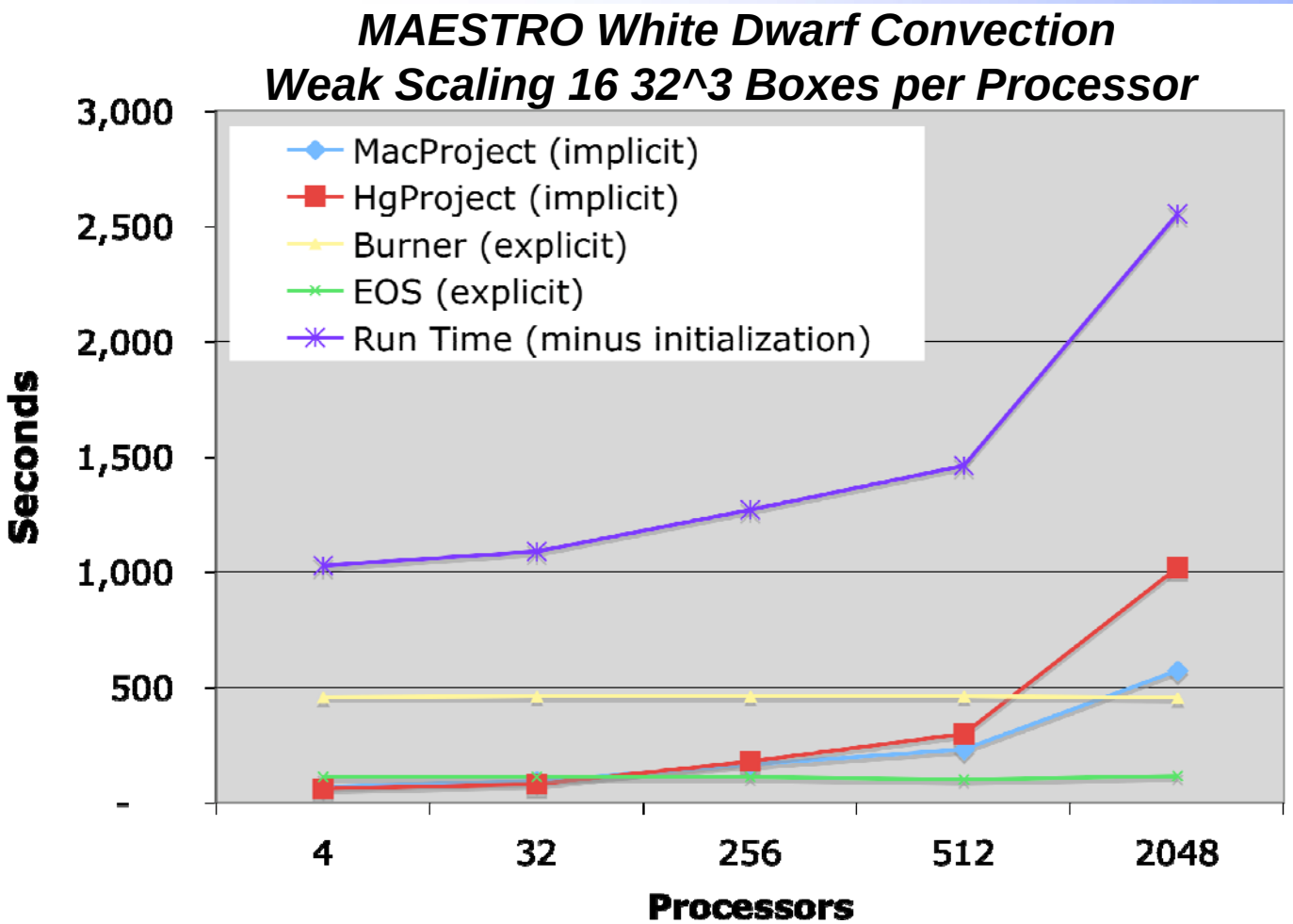
Software: 9

- 1 order: higher-order discretizations
  - Same accuracy can be achieved with many fewer elements
- 1 order: flux-surface following gridding
  - Less resolution required along than across field lines
- 4 orders: adaptive gridding
  - Zones requiring refinement are <1% of ITER volume and resolution requirements away from them are ~10^2 less severe
- 3 orders: implicit solvers
  - Mode growth time 9 orders longer than Alfven-limited CFL

Shape of things to come: MAESTRO

- **Authors: S. Woosley, SciDAC2 APDEC Center (SEESAR)**
- **Science:**
  - **Model convection leading up to Type 1a supernova explosion;**
  - **Method also applicable to 3-D turbulent combustion studies.**
- **Algorithm: Structured rectangular grid plus patch-based AMR (although NERSC-6 code does not adapt);**
  - **hydro model has implicit & explicit components;**
- **Coding: ~ 100,000 lines Fortran 90/77.**
- **Parallelism: 3-D processor non-overlapping decomposition, MPI.**
  - **Knapsack algorithm for load distribution; move boxes close in physical space to same/close processor.**
  - **Expresses AMR communication characteristics (BoxLib)**
  - **Also models requirements for PDE solvers using implicit timestepping schemes (Newton-Krylov methods)**

MAESTRO Scaling



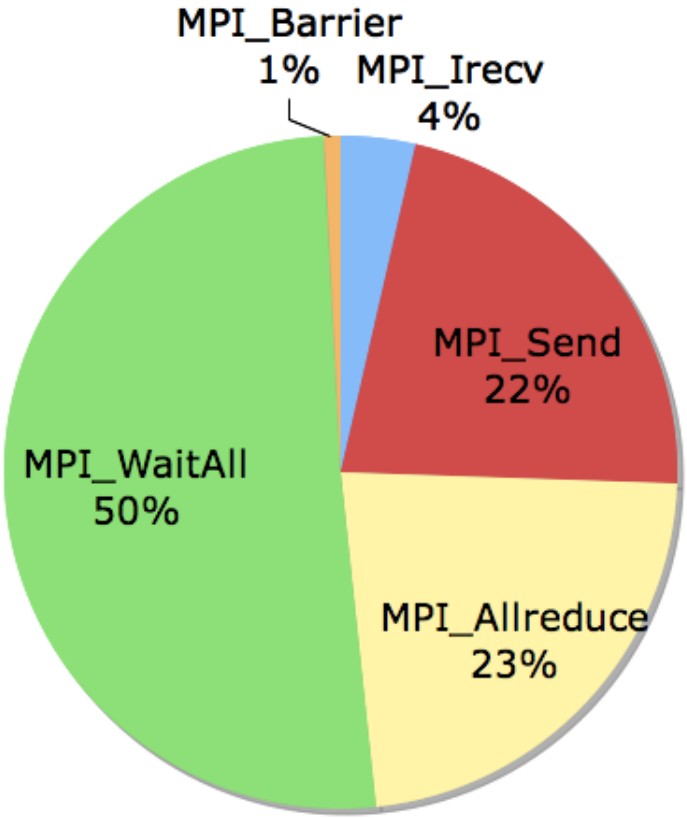
Explicit parts of the code scale very well but implicit parts of code pose more challenges to systems due to global communications

Maestro Communication Patterns

MAESTRO White Dwarf Convection
512 Processors 512x512X1024 Grid from Cray_Pat on Franklin

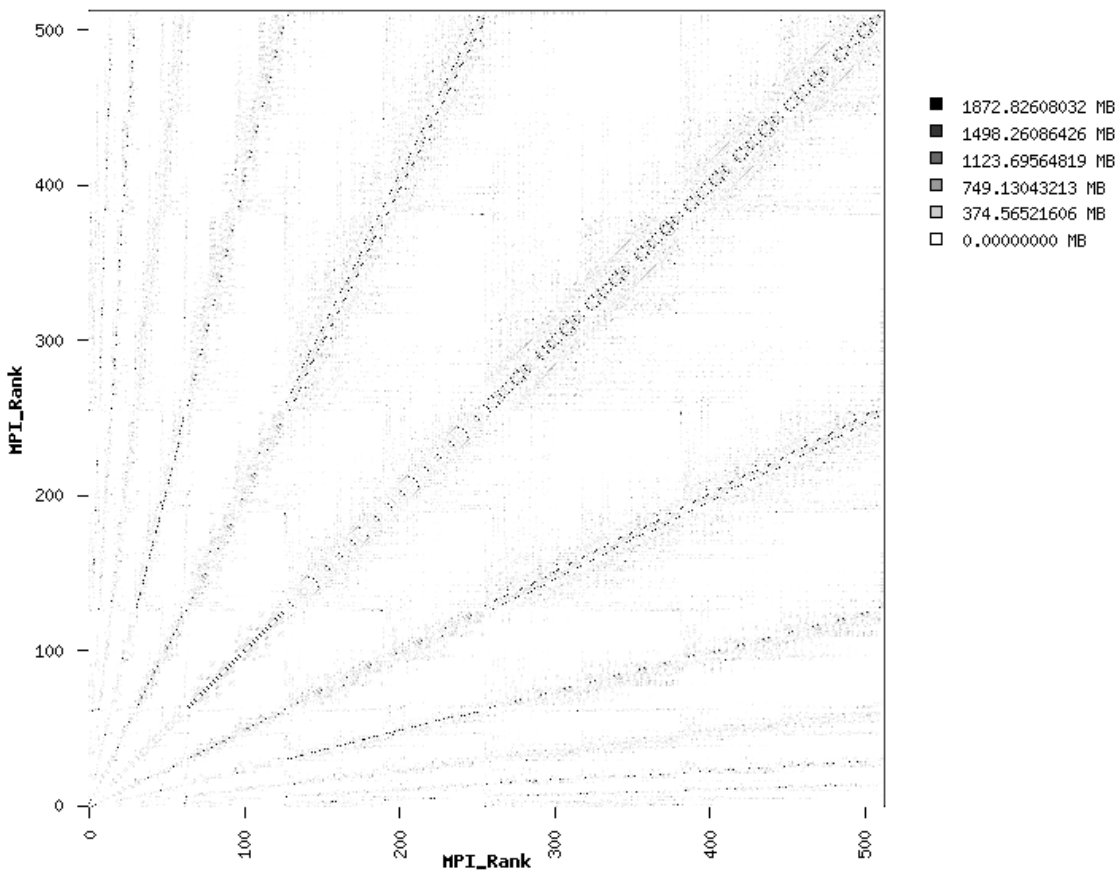
MPI Calls by Count

MPI Calls by Time



Maestro Communication Topology

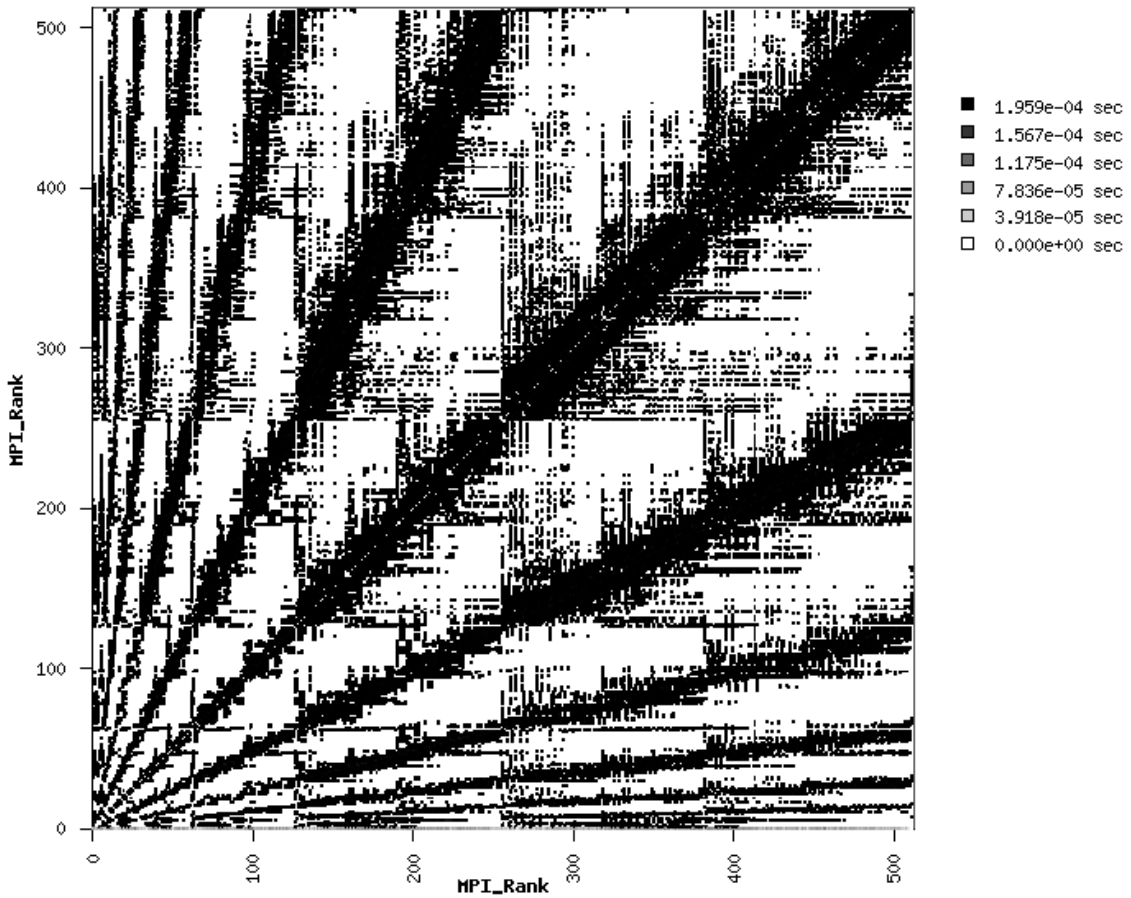
512 procs, 16 32^32 boxes per processor - grid size 512x512x1024 - by amount of data sent



- **Communication pattern based on Boxlib grid**
- **Boxlib works for both adaptive and uniform meshes**
- **Boxes distributed to be load balanced across processors**
- **Next, box location optimized for locality**
- **Result is a clumping effect**

Maestro Communication Topology

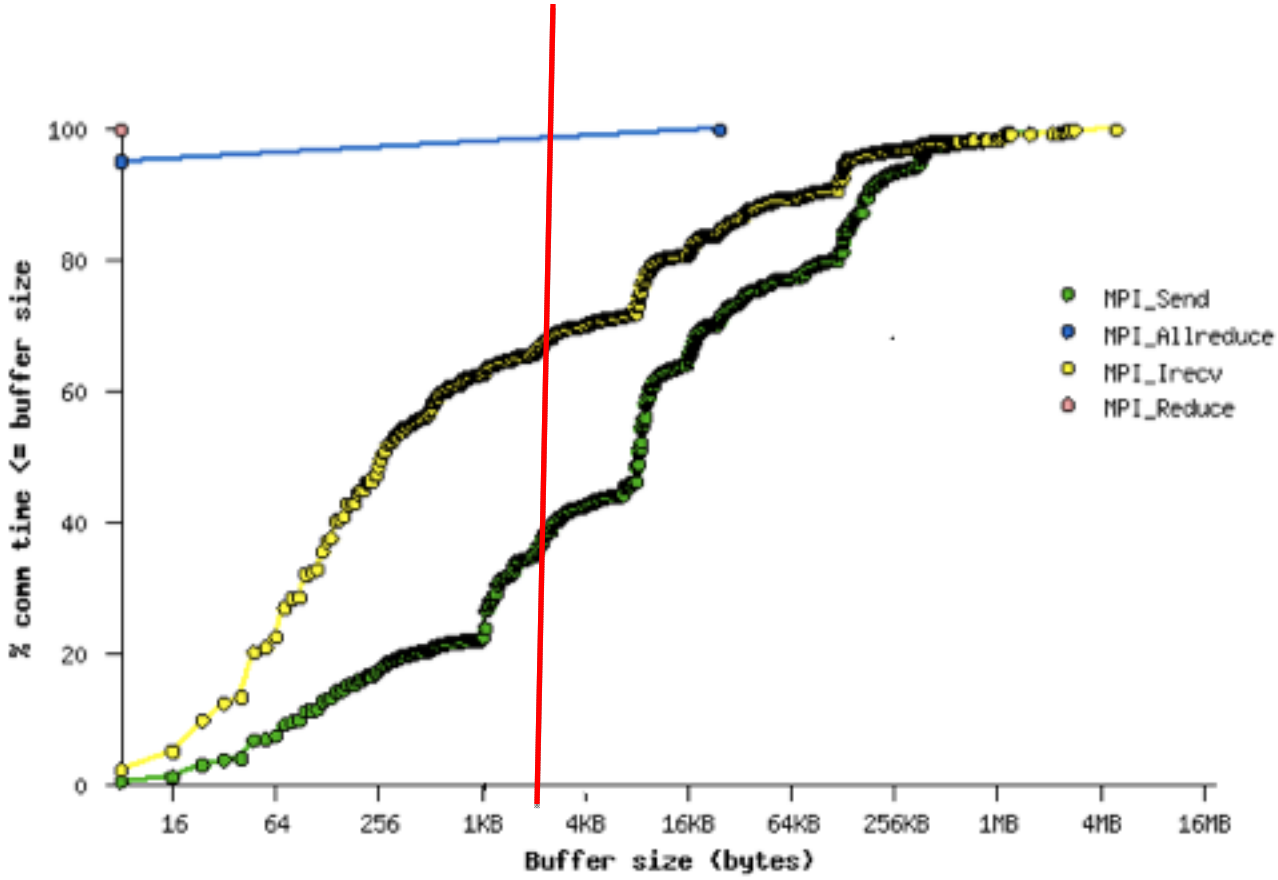
512 procs, 16 32^32 boxes per processor - grid size 512x512x1024 - by amount of time



- **Examining communication topology by time shows cost of short messages close to that of long messages**

Maestro Message Sizes

512 procs, 16 32^32 boxes per processor - grid size 512x512x1024



Message Buffer Size Distribution by Time

Strong-Scaling Drives Change in Interconnect Requirements

- **Concurrency:** *Must reduce memory overhead of identifying peers (eliminate O(N) and O(N^2) growth in messaging*
- **Implicit Methods***: Need much more efficient collectives (all-reduce) for Newton Vlasov methods*
- **Multiscale/AMR methods**: Complex message topology (not bisection limited, but does not map to simple topologies
- **Efficient Lightweight Messaging:** *All of these trends will push point-to-point messaging towards smaller message sizes.*

Additional Requirements

- **For Developers of Performance Tools: Interconnect performance counters**
  - Difficult to measure actual time in async messaging when just timing MPI calls (worse if you use one-sided messaging)
  - Need to understand causality (disambiguating counters)
  - Directly measure LOG-P parameters (instead of inferring them indirectly)
- **For Developers of Advanced Programming Models & Languages**
  - Need compact addressing of peers (avoid overhead of *naming* peers for messaging: hardware should translate peer addresses)
  - DMA must understand effective addresses (must be TLB coherent with processor)
  - Need for lower-cost interaction with device interface (lower overhead)
    - chatty device protocols have high overhead because device writes must be uncached!
    - Overhead is more of a problem than latency per se (can use slack to hide latency)
  - Ultimately, it is a huge advantage to have device interfaces and DMA on same chip as CPUs (SoC)
  - Per-CPU limited injection rate (Bane of Hybrid Programming Model)