BIPS

C O M P U T A T I O N A L   R E S E A R C H   D I V I S I O N

Communication Analysis of Ultrascale Applications using IPM

John Shalf, Shoaib Kamil, Leonid Oliker, David Skinner

NERSC/CRD

jshalf@lbl.gov

DOE, NSA Review, July 19-20, 2005

Overview

- CPU clock scaling bonanza has ended
  - Heat density
  - New physics below 90nm *(departure from bulk material properties)*
- Yet, by end of decade mission critical applications expected to have 100X computational demands of current levels *(PITAC Report, Feb 1999)*
- The path forward for high end computing is increasingly reliant on massive parallelism
  - Petascale platforms will likely have hundreds of thousands of processors
  - System costs and performance may soon be dominated by interconnect
- What kind of an interconnect is required for a >100k processor system?
  - What topological requirements? *(fully connected, mesh)*
  - Bandwidth/Latency characteristics?
  - Specialized support for collective communications?

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?**

Approach

- **Identify candidate set of "Ultrascale Applications" that span scientific disciplines**
  - **Applications demanding enough to require Ultrascale computing resources**
  - **Applications that are capable of scaling up to hundreds of thousands of processors**
  - ***Not every app is "Ultrascale!"***

- **Find communication profiling methodology that is**
  - **Scalable: Need *to be able to run for a long time with many processors. Traces are too large***
  - **Non-invasive: *Some of these codes are large and can be difficult to instrument even using automated tools***
  - **Low-impact on performance: *Full scale apps… not proxies!***

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");
```

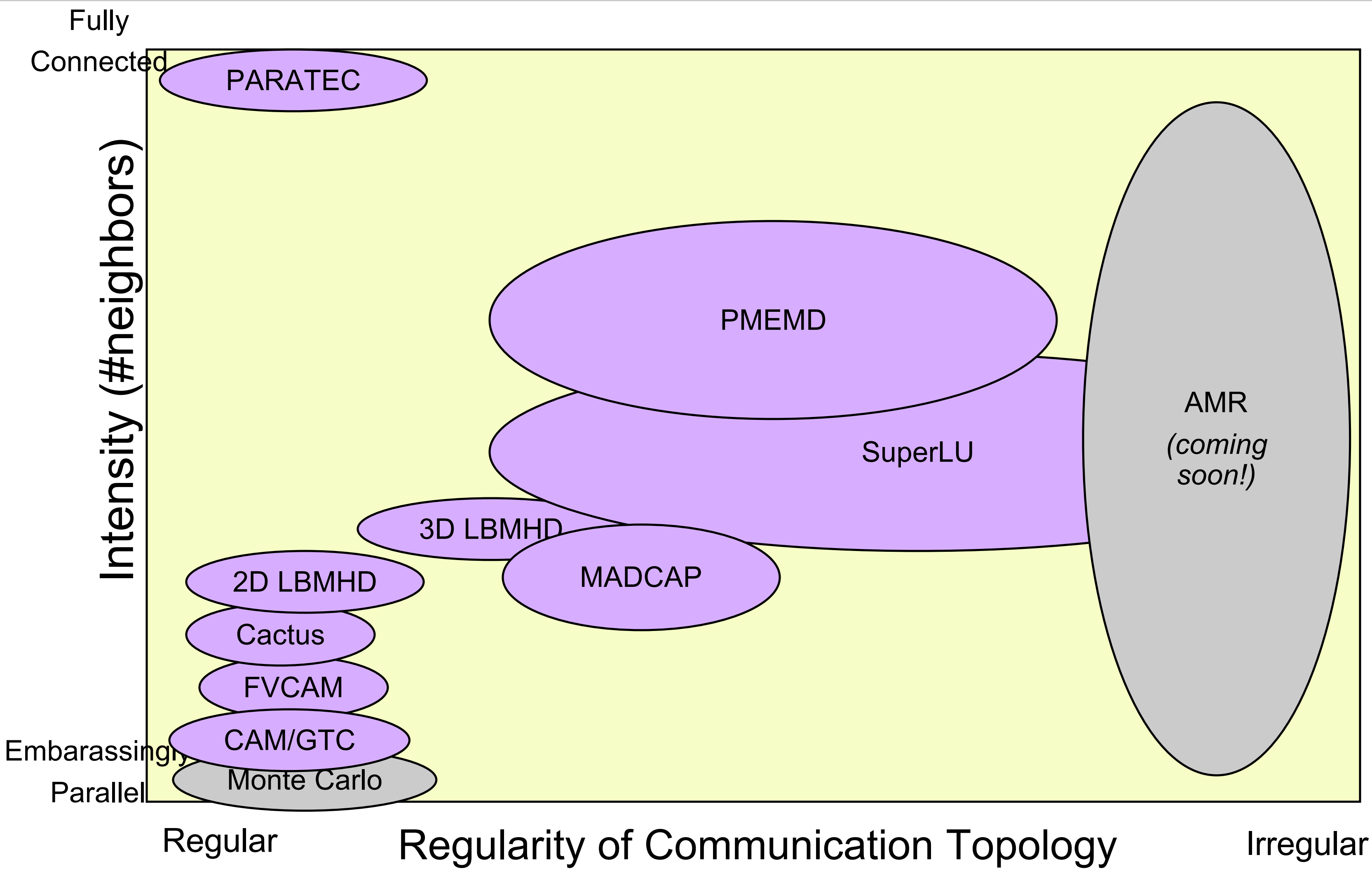
```
###########################################
# IPMv0.7 :: csnode041 256 tasks  ES/ESOS
# madbench.x (completed) 10/27/04/14:45:56
#
#         <mpi>       <user>       <wall> (sec)
#        171.67       352.16       393.80
#...
###############################################
# W
#         <mpi>       <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, LBNL

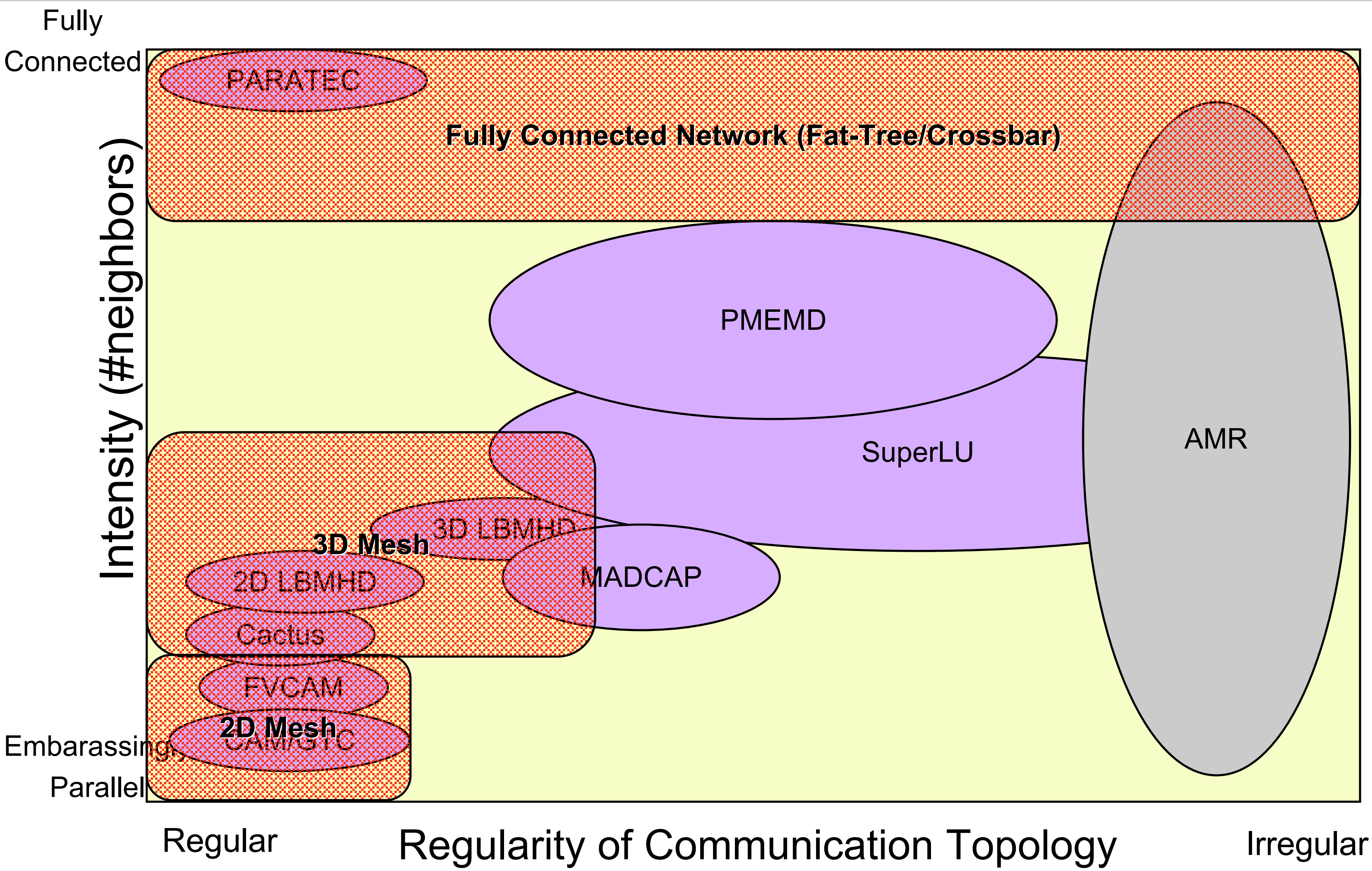
Application Overview *(the "nails")*

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

Presumed Communication Requirements



Communication Requirements



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**

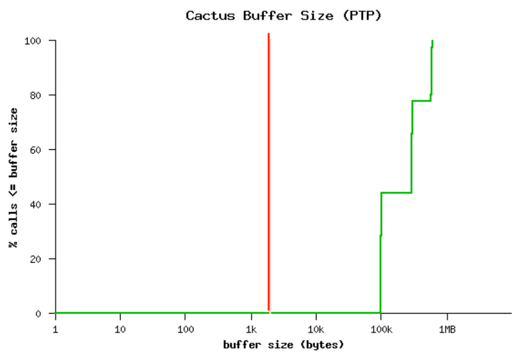
System	Technology	MPI Latency	Peak Bandwidth	Bandwidth Delay Product
SGI Altix	Numalink-4	1.1us	1.9GB/s	2KB
Cray X1	Cray Custom	7.3us	6.3GB/s	46KB
NEC ES	NEC Custom	5.6us	1.5GB/s	8.4KB
Myrinet Cluster	Myrinet 2000	5.7us	500MB/s	2.8KB
Cray XD1	RapidArray/IB4x	1.7us	2GB/s	3.4KB

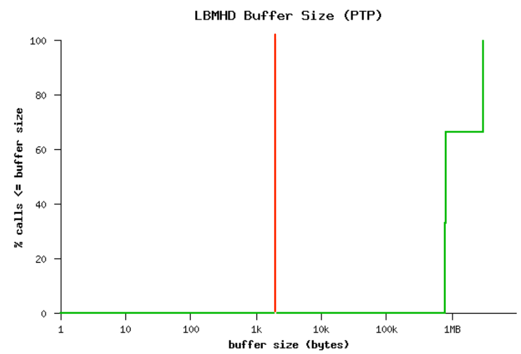
- **Bandwidth Bound if msg size > Bandwidth*Delay**
- **Latency Bound if msg size < Bandwidth*Delay**
  - **Except if pipelined *(unlikely with MPI due to overhead)***
  - **Cannot pipeline MPI collectives *(but can in Titanium)***

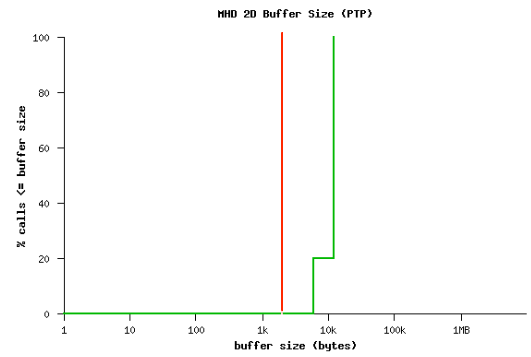
Call Counts

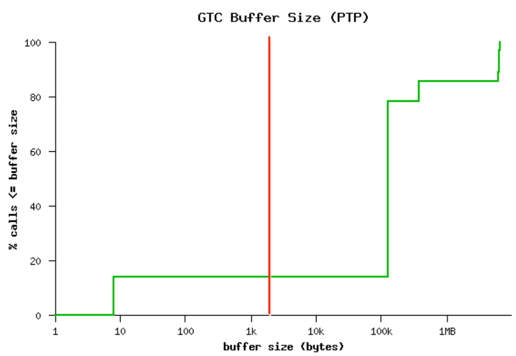
Cactus
Wait
Irecv
Isend
LBMHD
WaitAll
Isend
Irecv
FVCAM2D
Wait
WaitAll
Isend
Irecv
Paratec
Irecv
Wait
Isend
SuperLU
Send
Recv
Wait
Irecv
Isend
MADbench
Reduce
Send
Recv
GTC
SendRecv
Gather
Allreduce
PMEMD
WaitAny
Irecv
Isend
MPI_Bcast
MPI_Gatherv
MPI_Gather
MPI_Allreduce
MPI_Alltoall
MPI_Reduce
MPI_Waitall
MPI_Recv
MPI_Barrier
MPI_Alltoallv
MPI_Irecv
MPI_Isend
MPI_Sendrecv
MPI_Wait
MPI_Scatter
MPI_Waitany
MPI_Allgather
MPI_Send

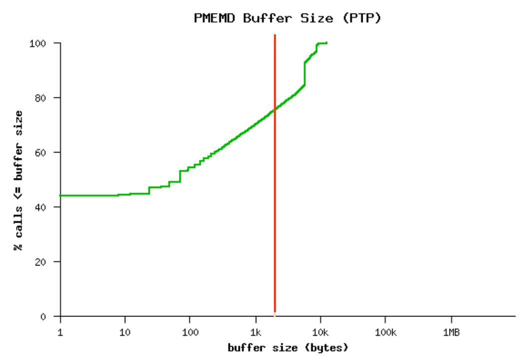
P2P Buffer Sizes

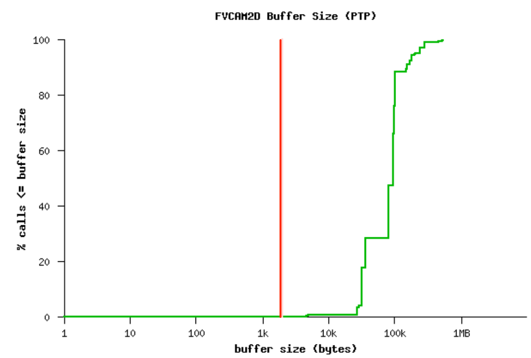


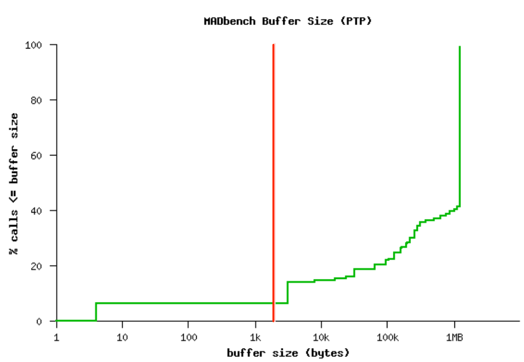


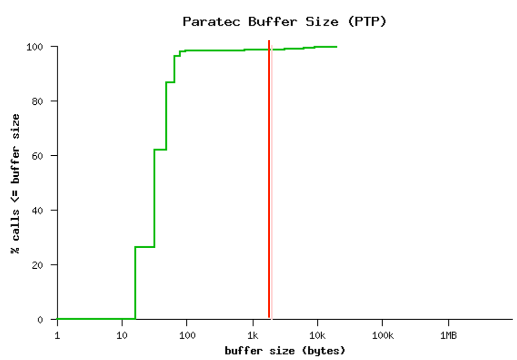


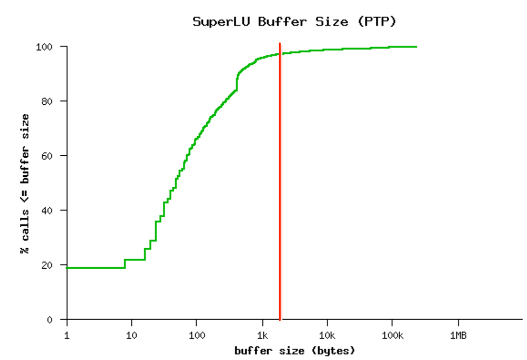




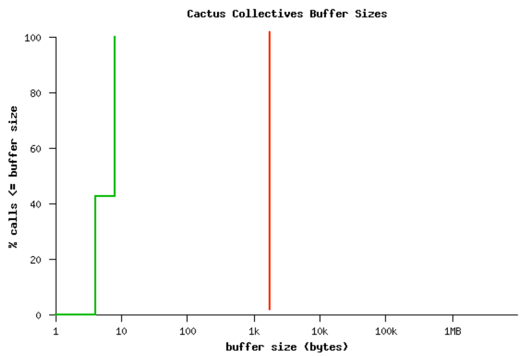


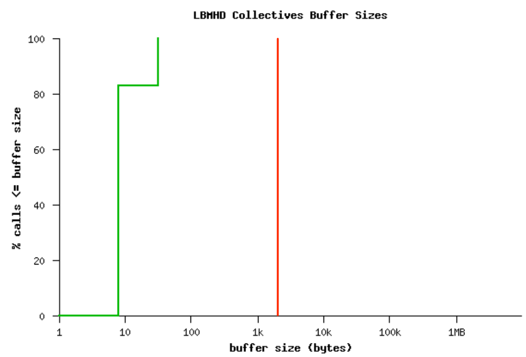


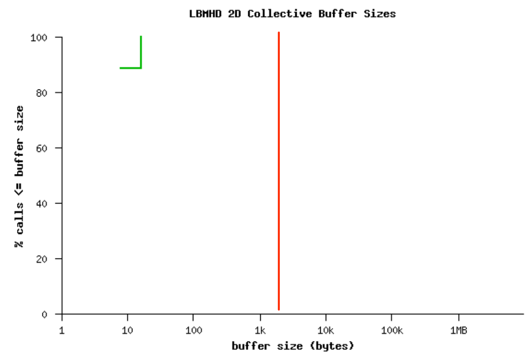


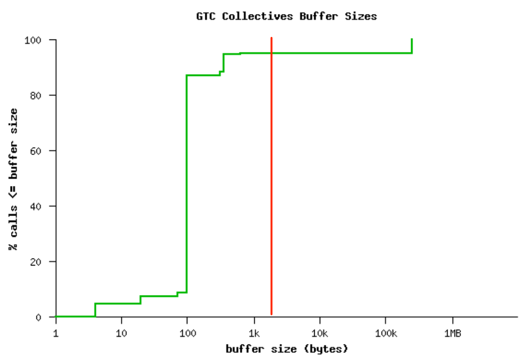


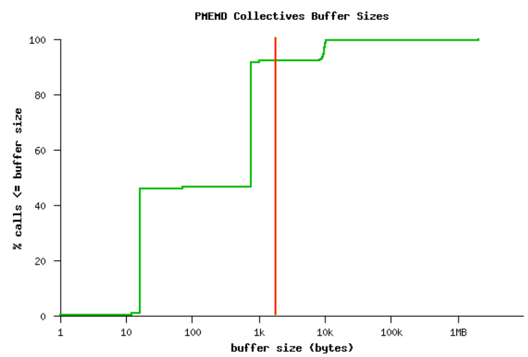
Collective Buffer Sizes

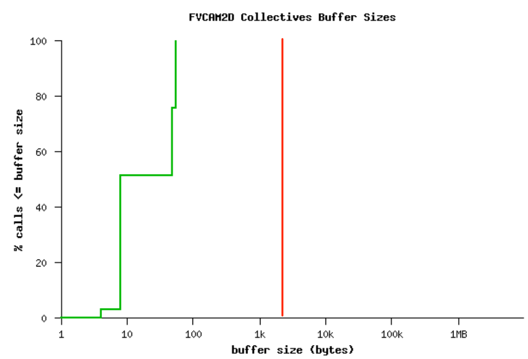


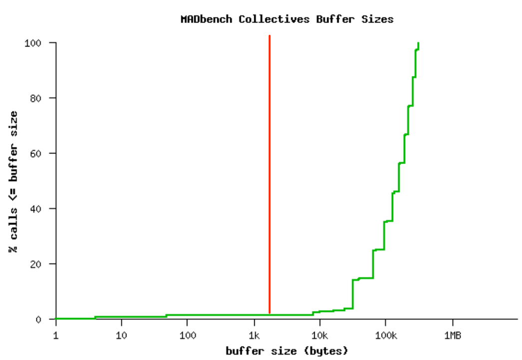


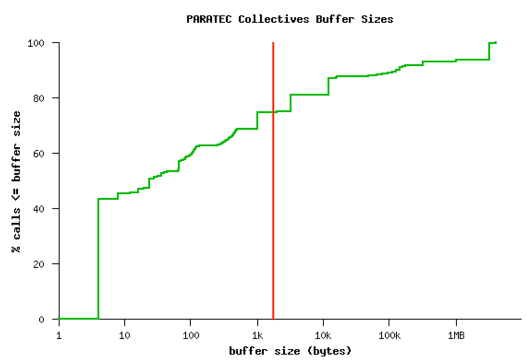


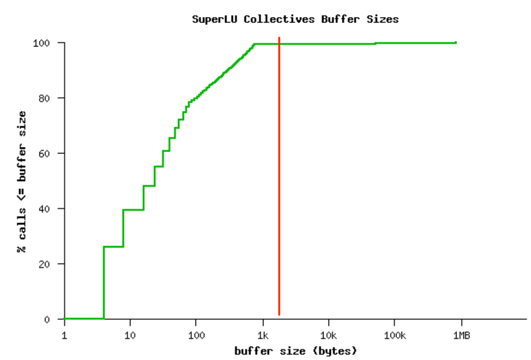




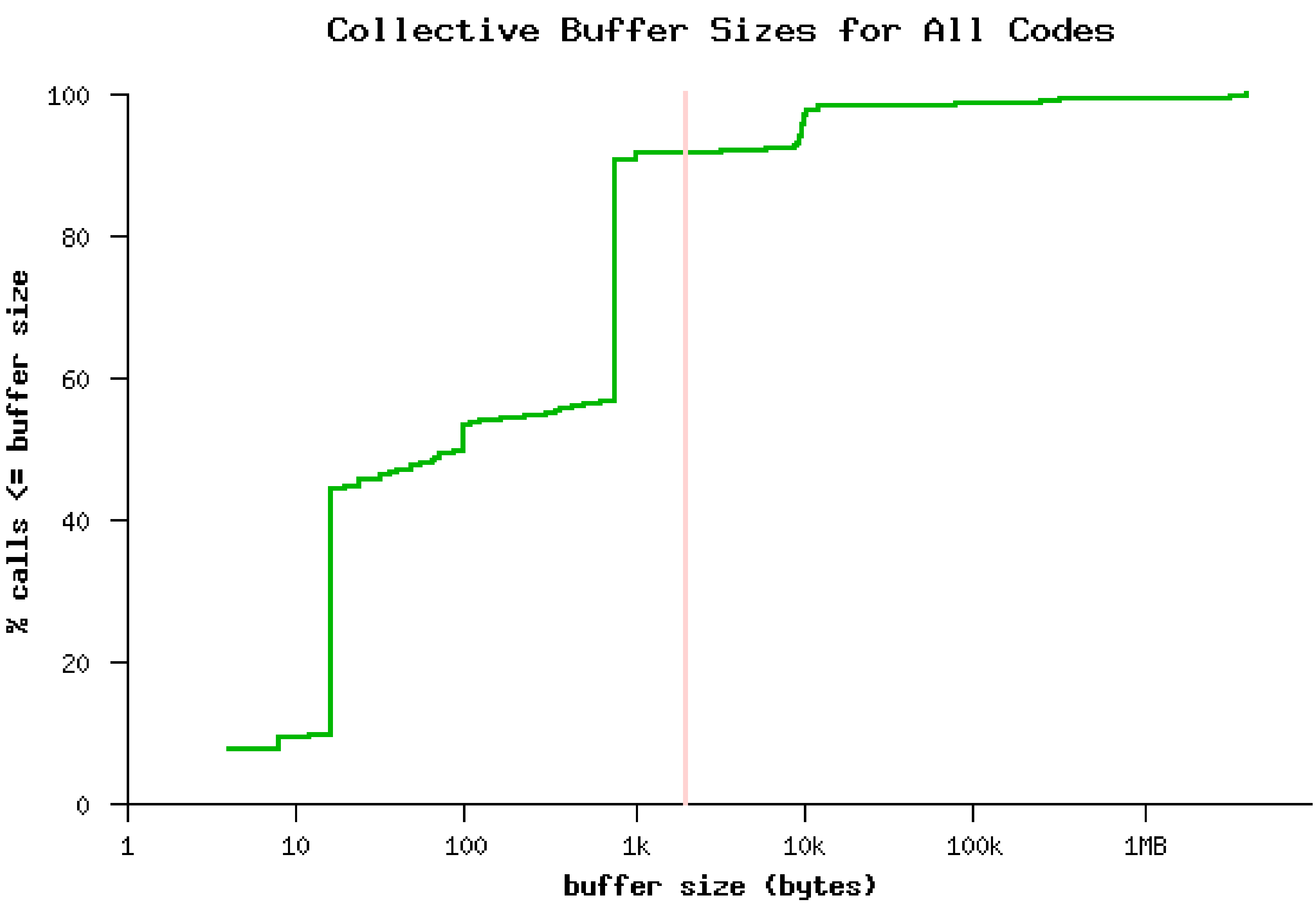




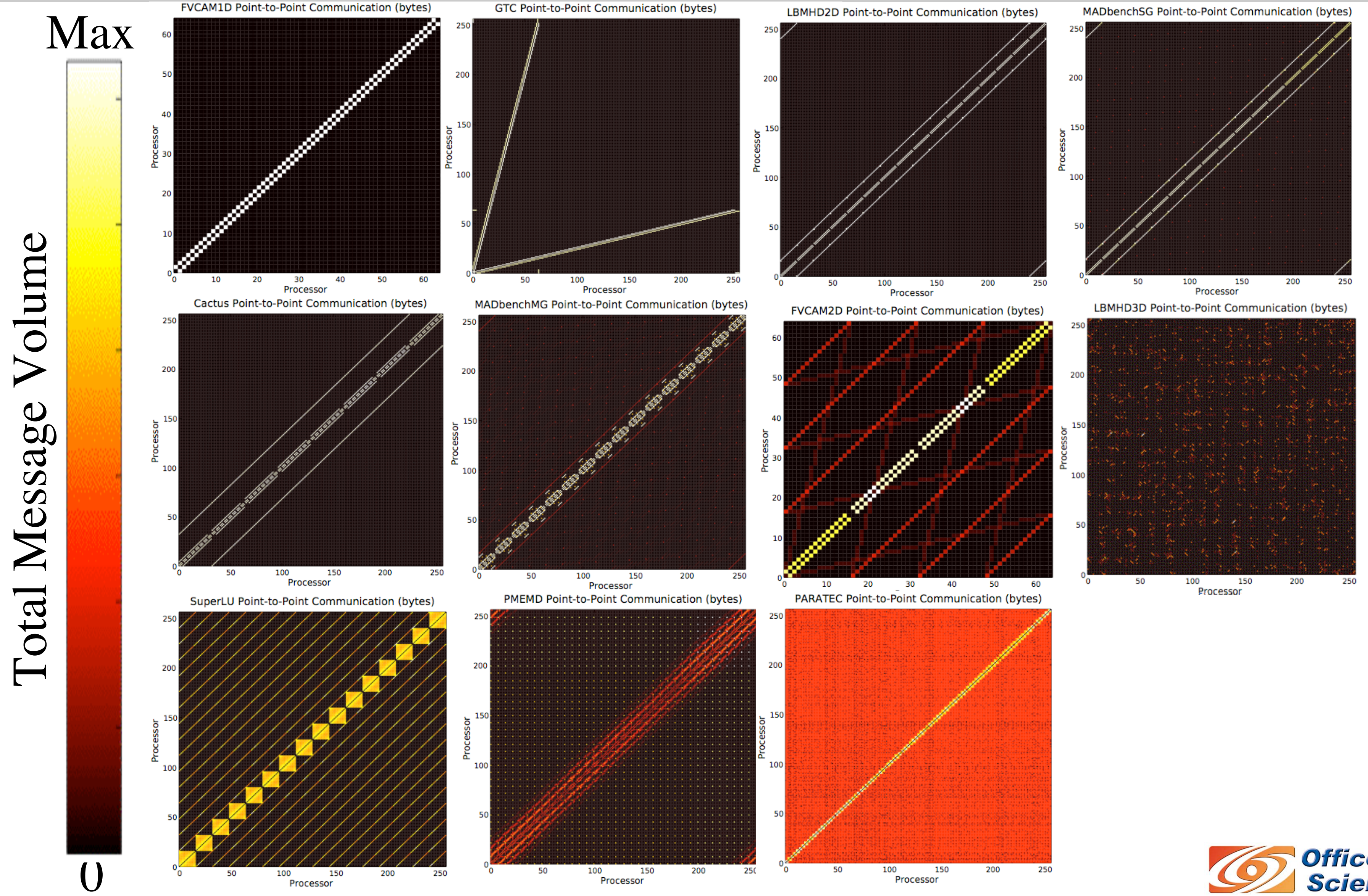




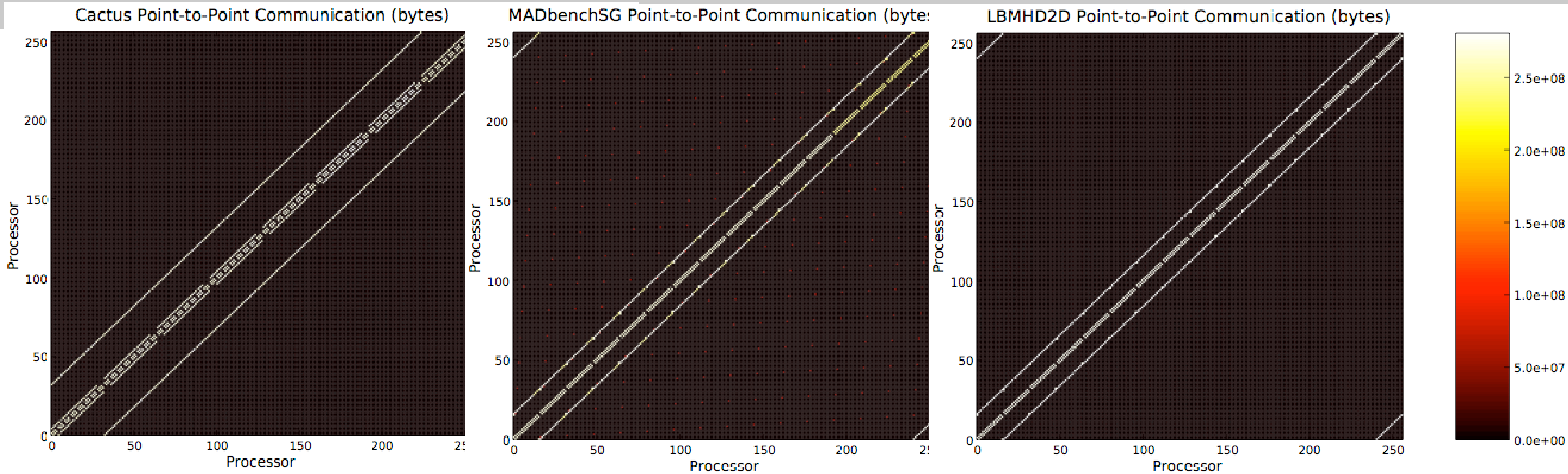
Collective Buffer Sizes

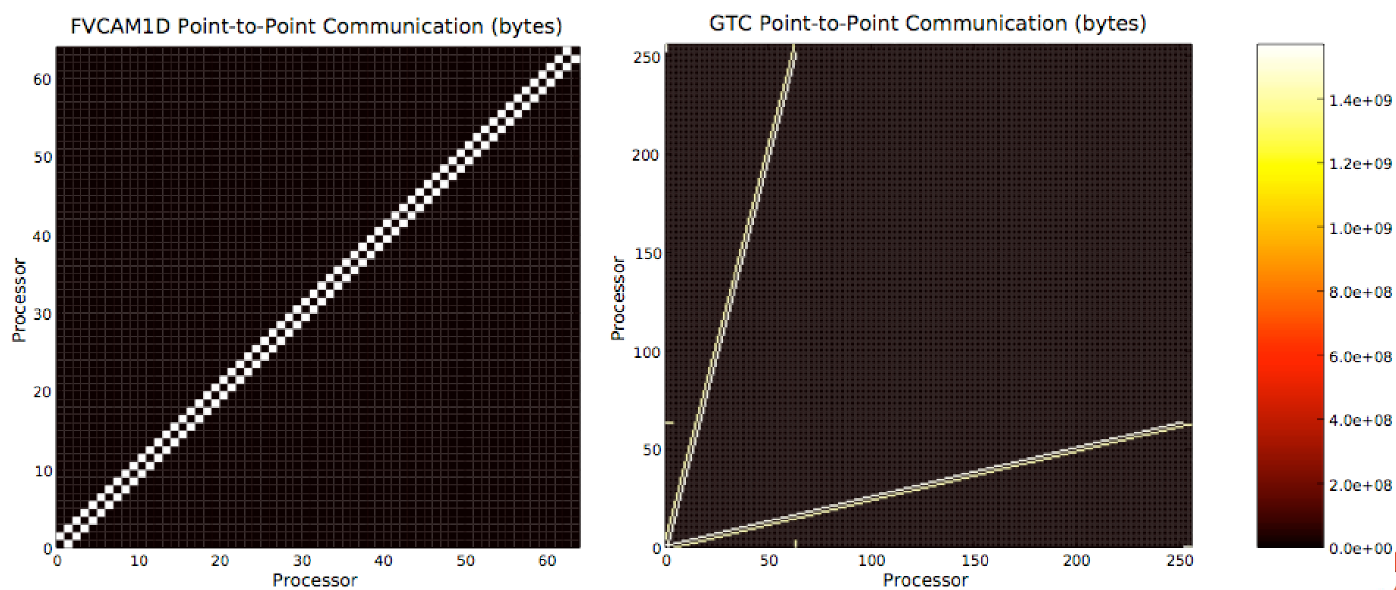


P2P Topology Overview



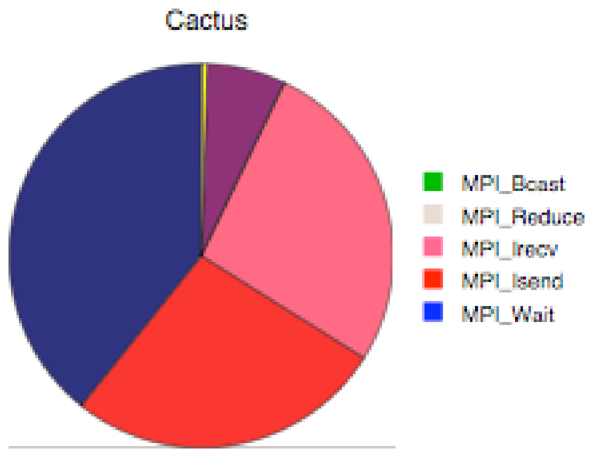
Fully Regular Communication Patterns

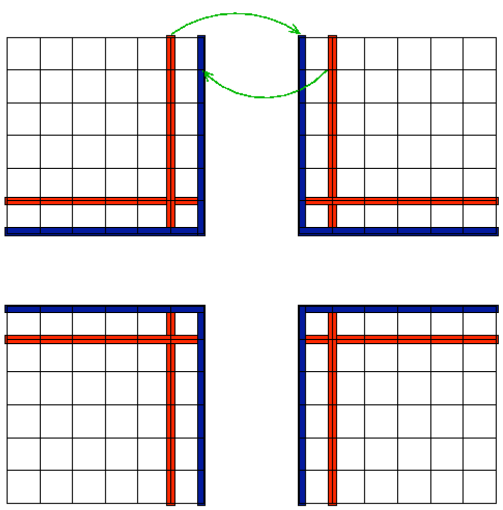


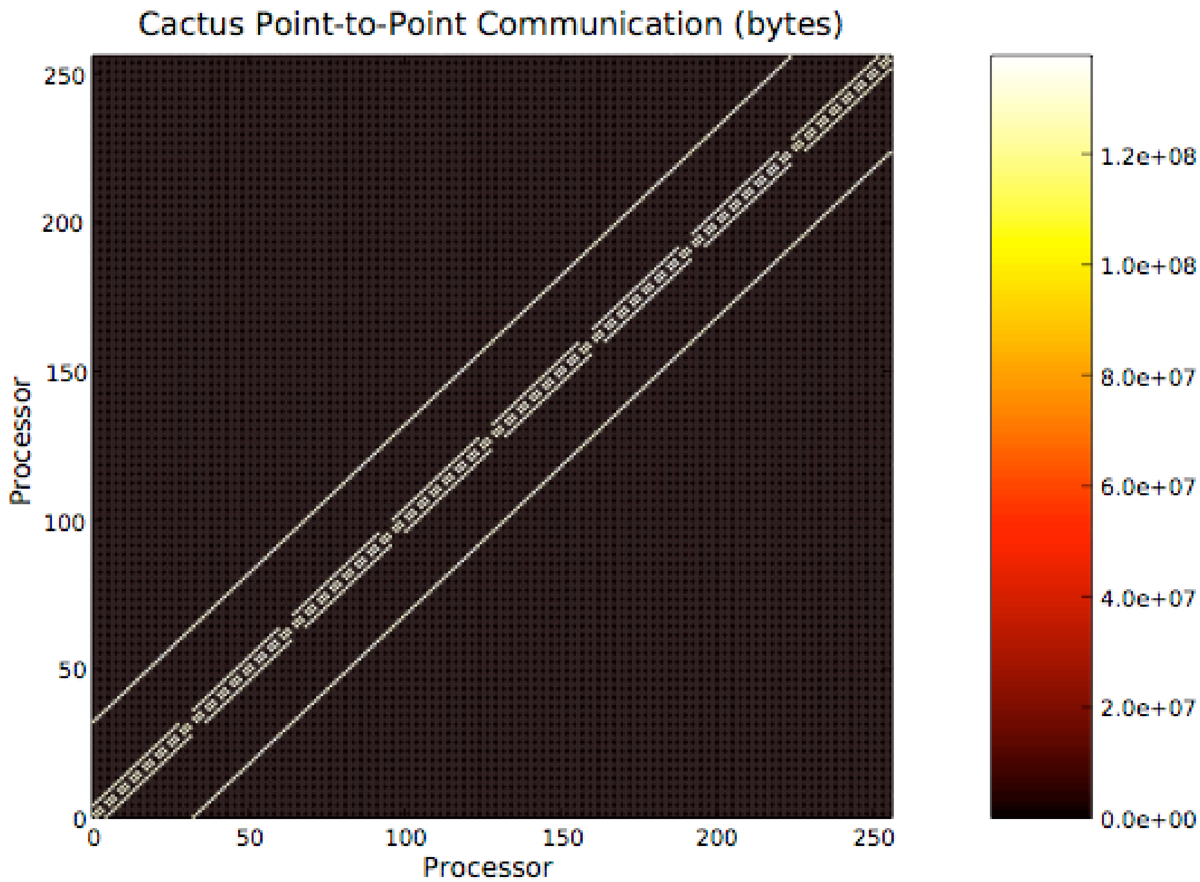


Cactus Communication

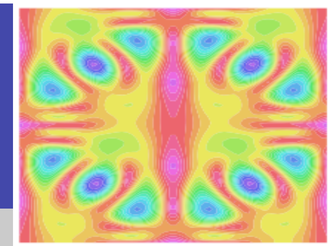
PDE Solvers on Block Structured Grids

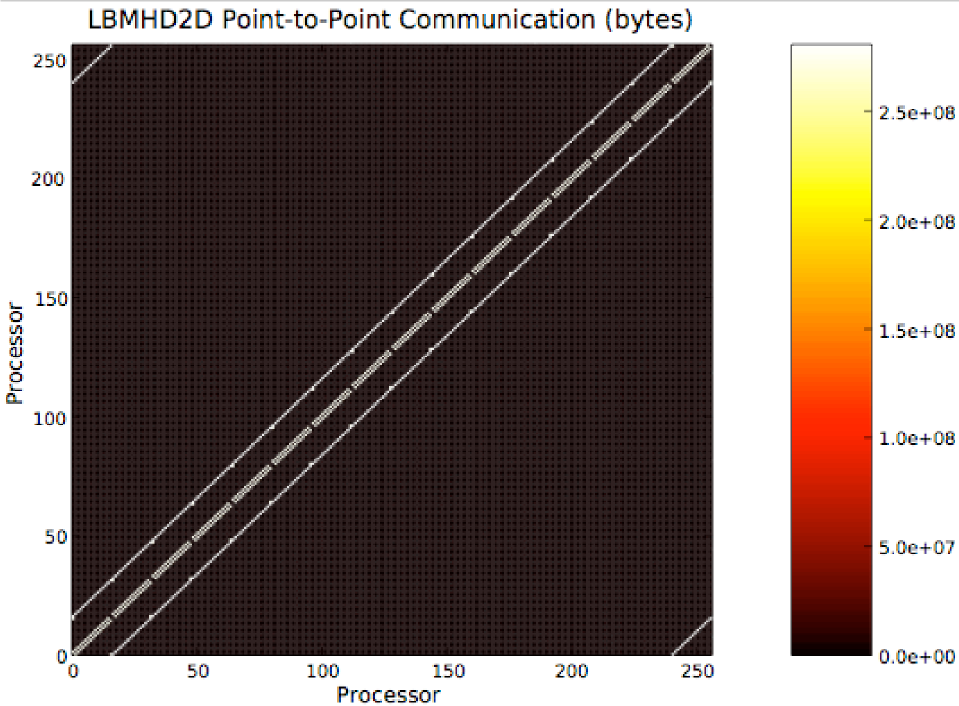


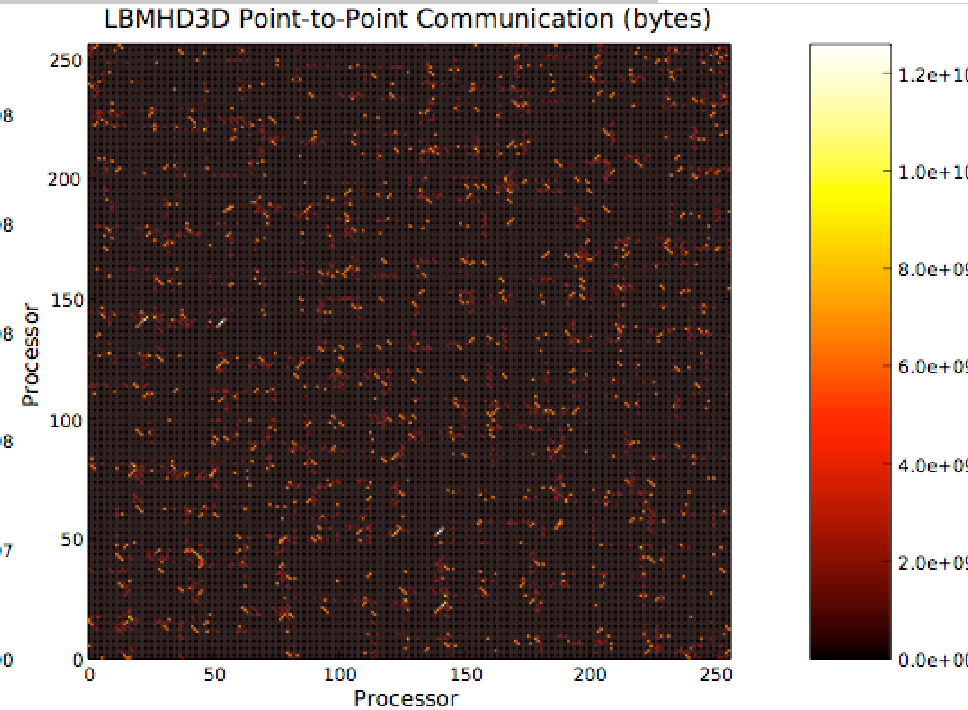


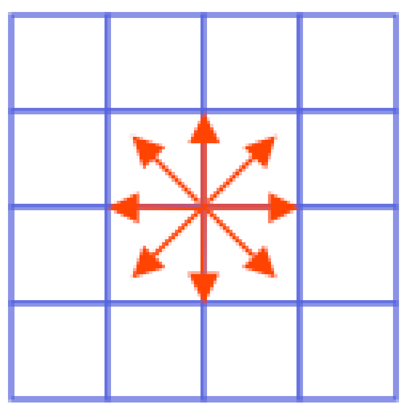


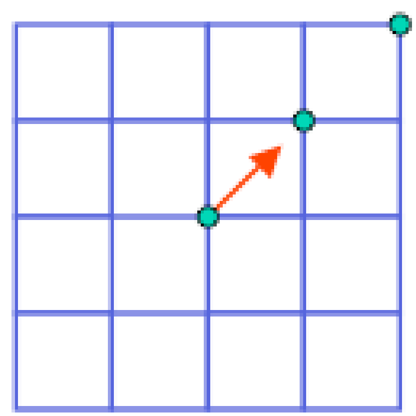
LBMHD Communication



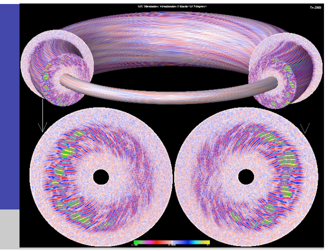


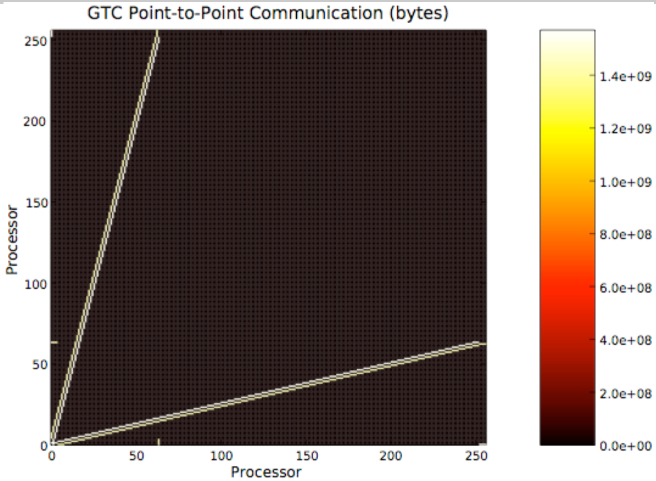


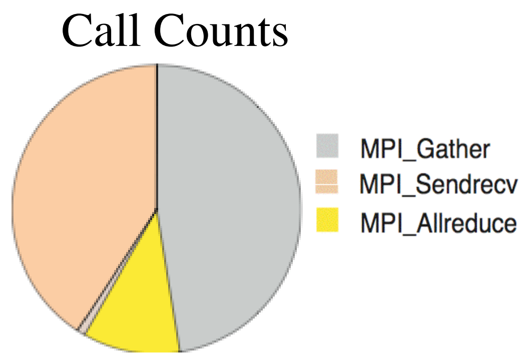


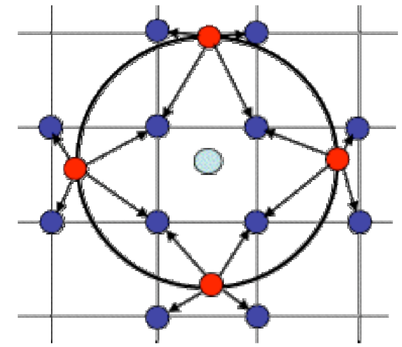


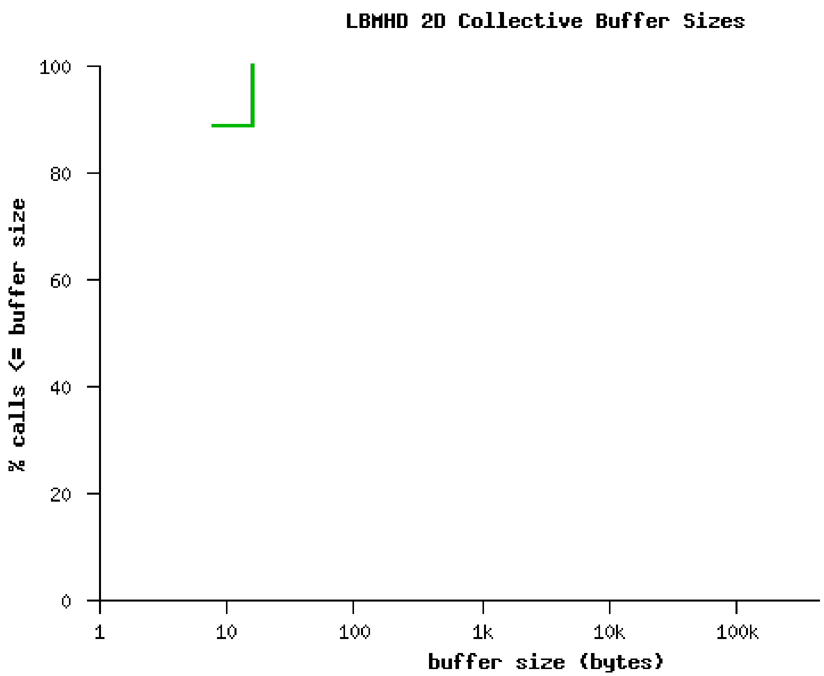
GTC Communication

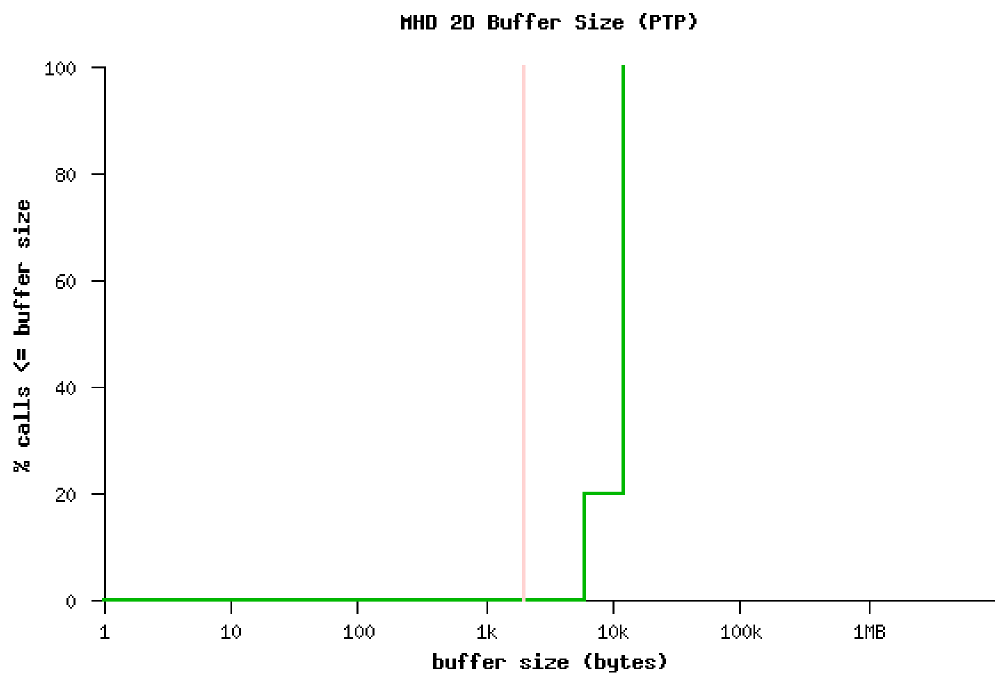






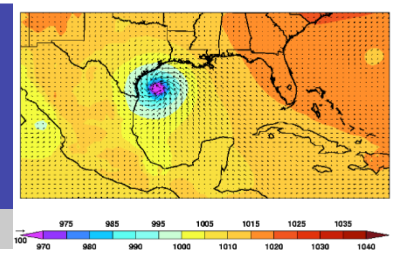


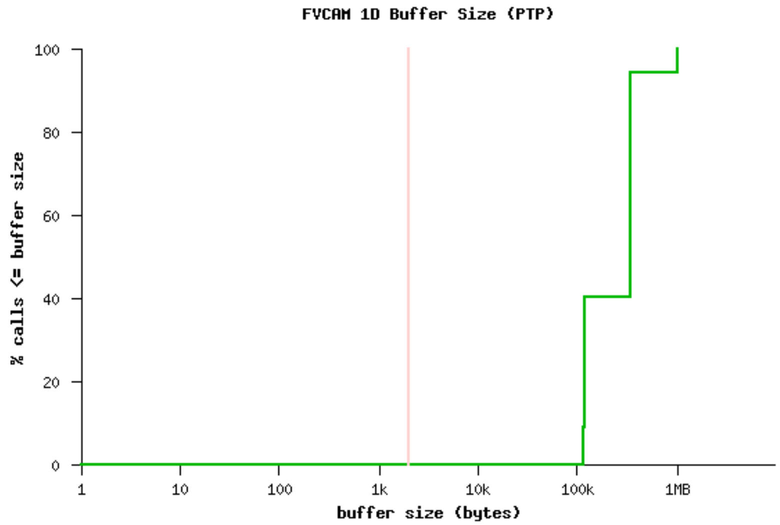


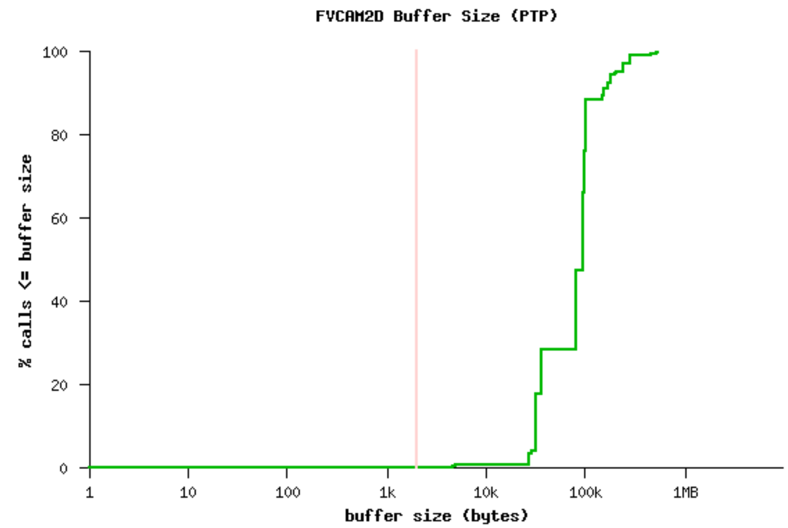


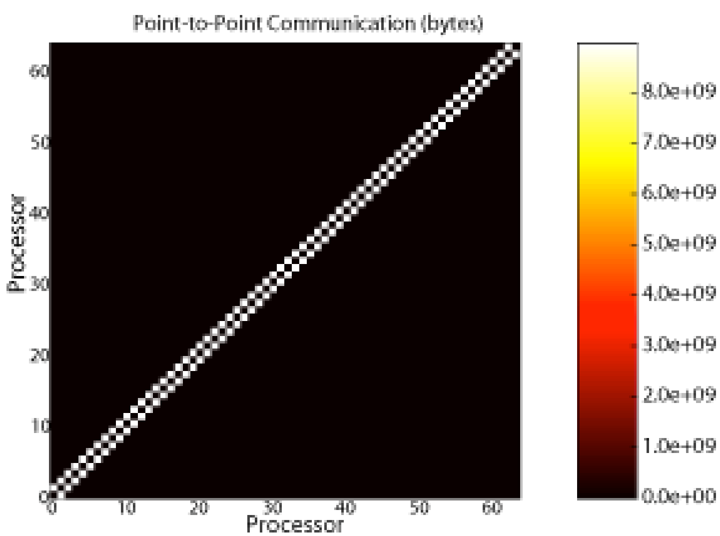
BIPS

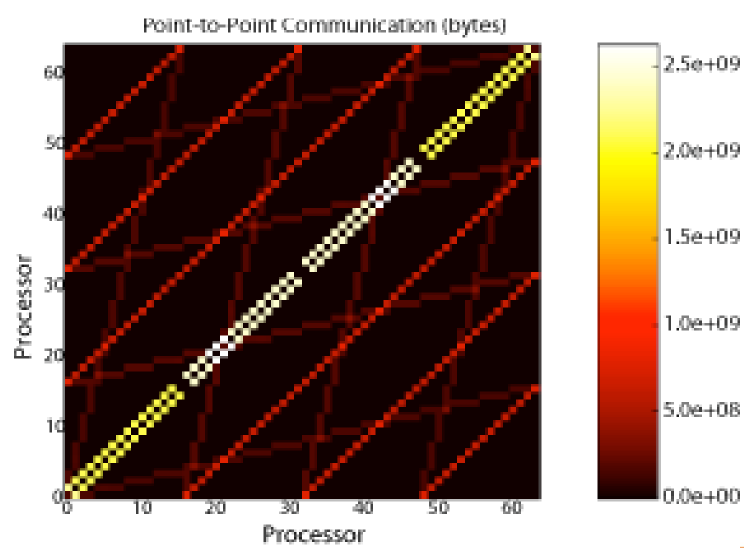
FVCAM Communication





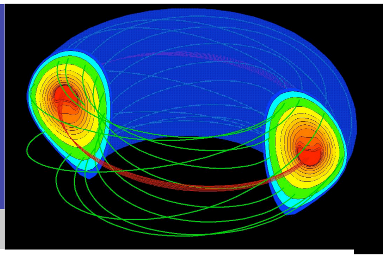


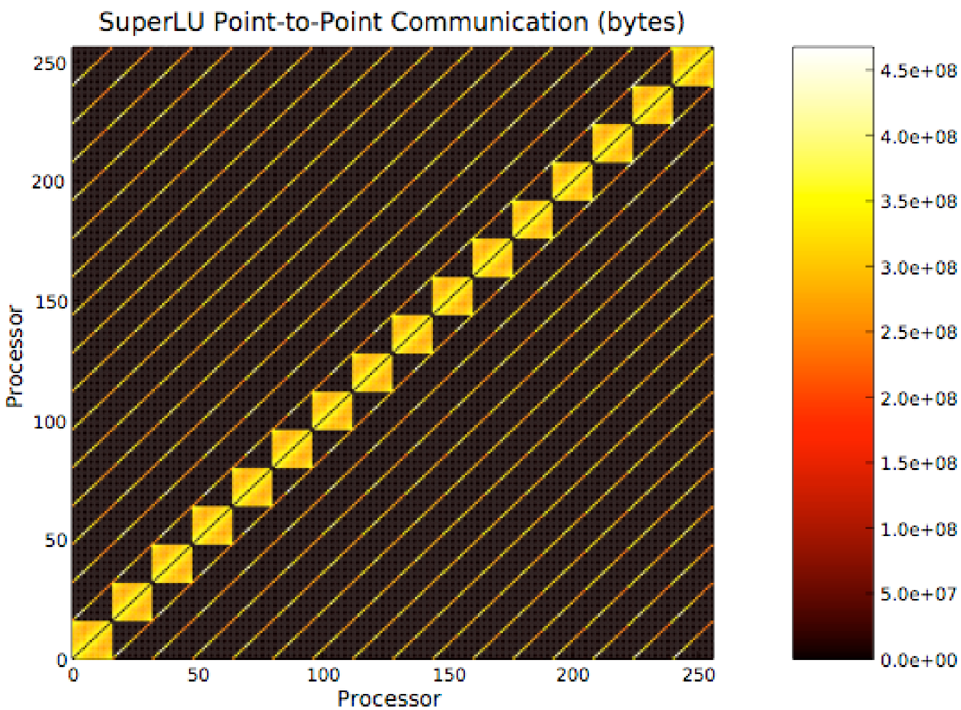


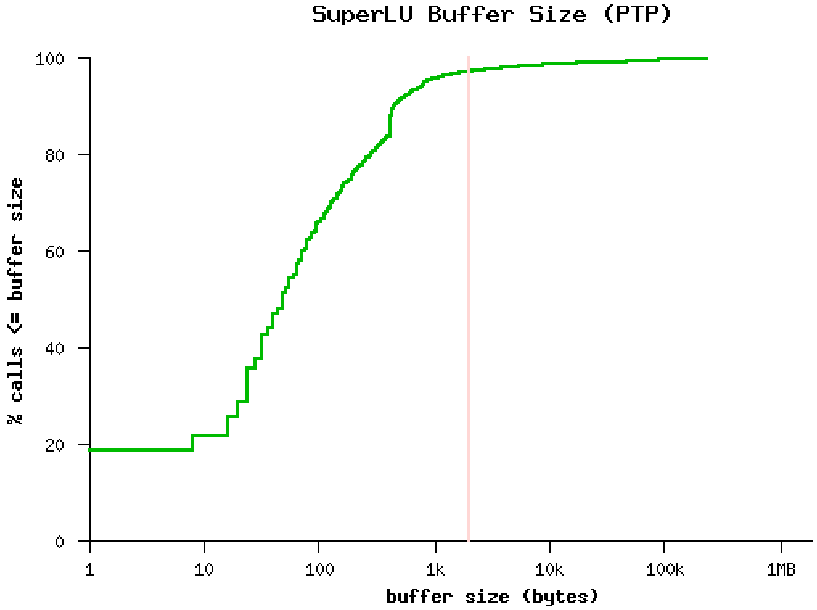


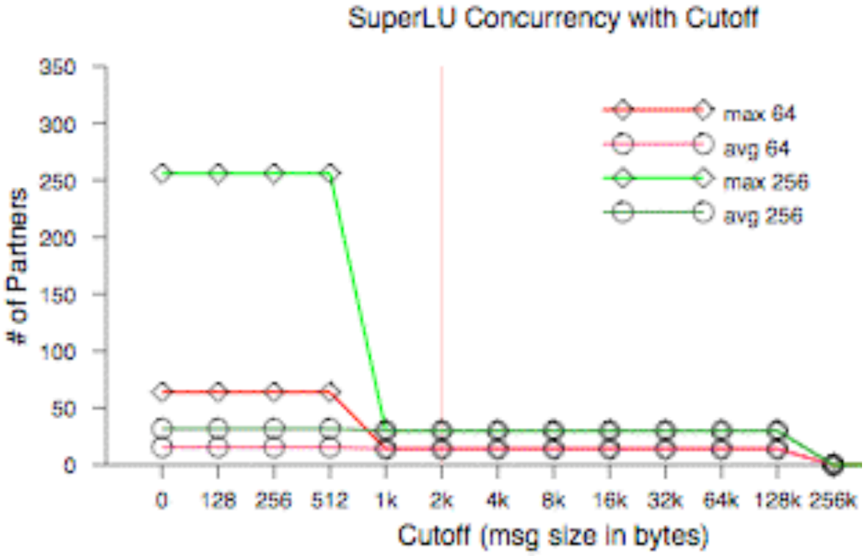
Office of Science
U.S. DEPARTMENT OF ENERGY

SuperLU Communication

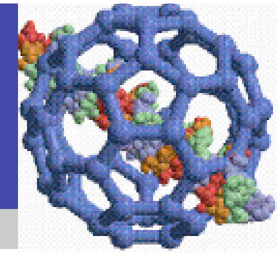


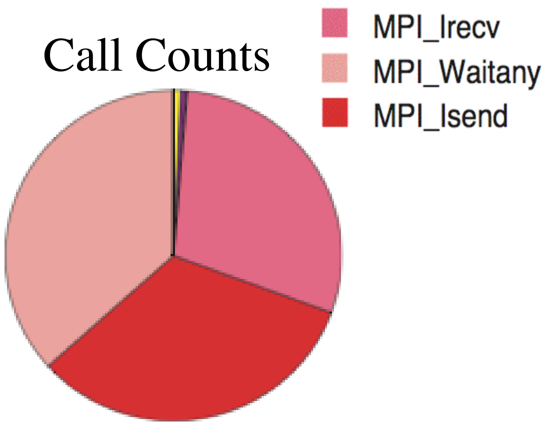


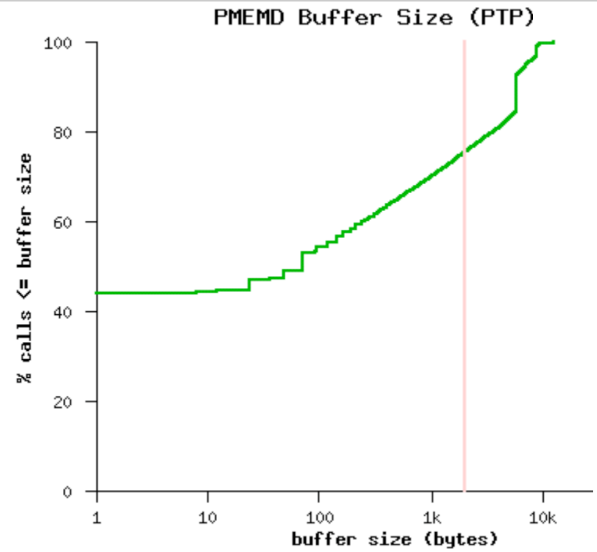


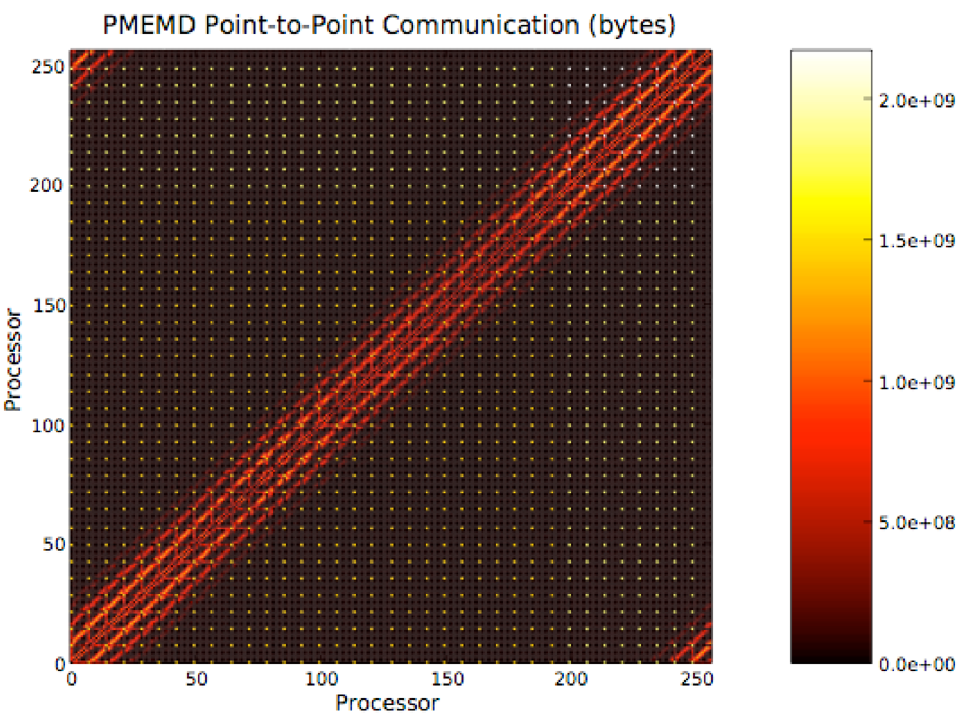


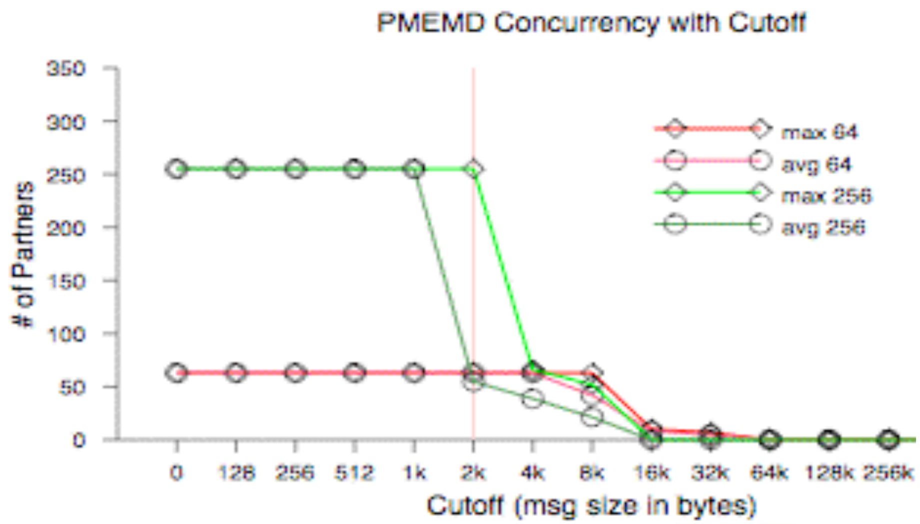
PMEMD Communication



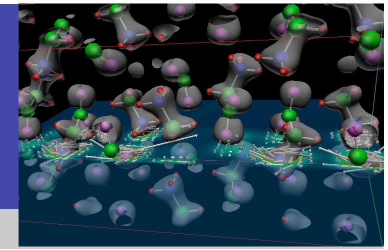


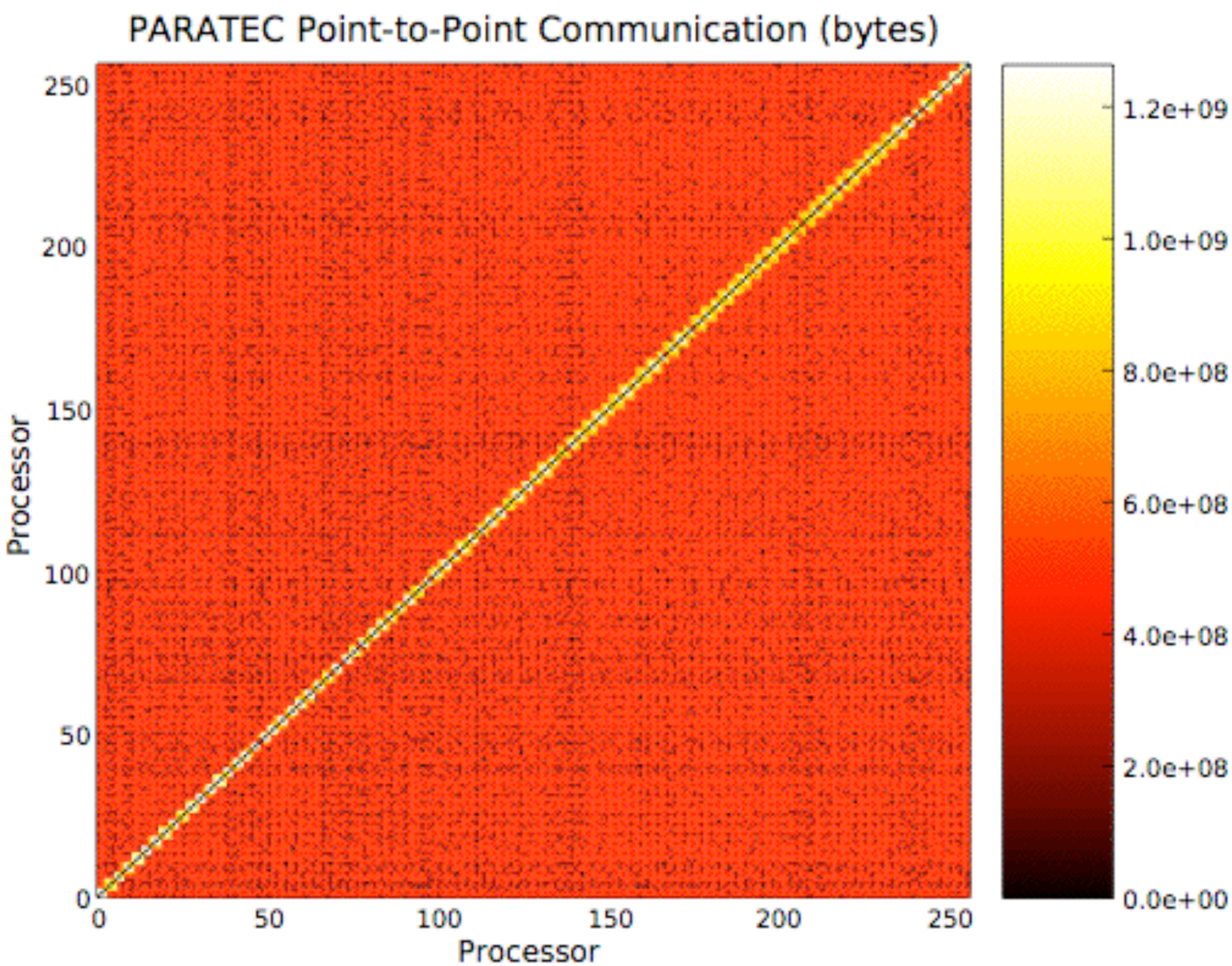


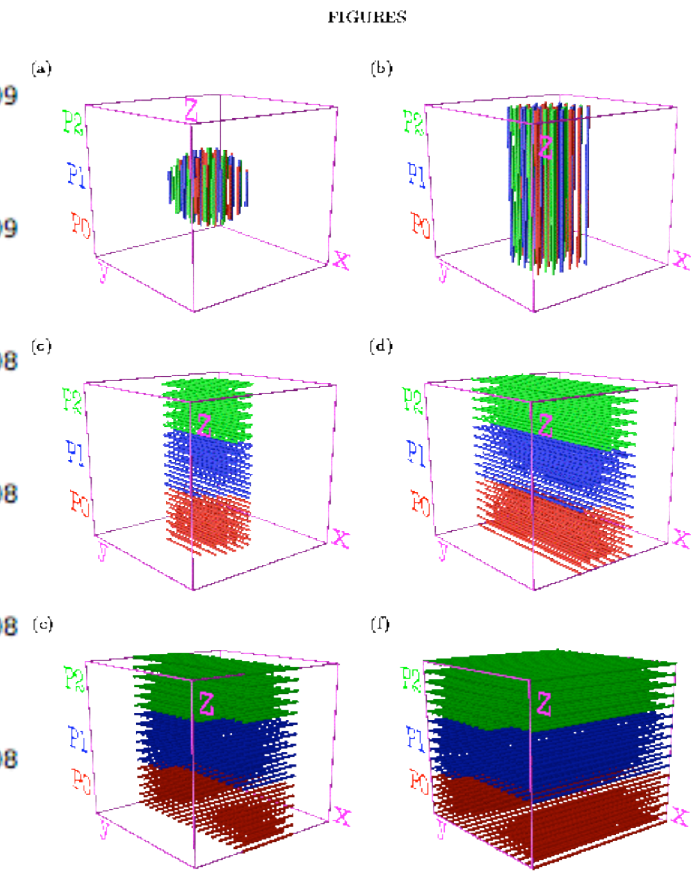




PARATEC Communication







Summary of Communication Patterns

Code 256procs	%P2P : %Collective	Avg. Coll Bufsize	Avg. P2P Bufsize	TDC@2k max,avg.	%FCN Utilization
GTC	40% : 60%	100	128k	10 , 4	2%
Cactus	99% : 1%	8	300k	6 , 5	2%
LBMHD	99% : 1%	8	3D=848k 2D=12k	12 , 11.8	5% 2%
SuperLU	93% : 7%	24	48	30 , 30	25%
PMEMD	98% : 2%	768	6k or 72	255 , 55	22%
PARATEC	99% : 1%	4	64	255 , 255	100%
MADCAP-MG	78% : 22%	163k	1.2M	44 , 40	23%
FVCAM	99% : 1%	8	96k	20 , 15	16%

Revisiting Original Questions

- Topology
  - Most codes require far less than full connectivity
    - PARATEC is the only code requiring full connectivity
    - Many require low degree (<12 neighbors)
  - Low TDC codes not necessarily isomorphic to a mesh!
    - Non-isotropic communication pattern
    - Non-uniform requirements

- Bandwidth/Delay/Overhead requirements
  - Scalable codes primarily bandwidth-bound messages
  - Average message sizes several Kbytes

- Collectives
  - Most payloads less than 1k *(8-100 bytes!)*
    - Well below the bandwidth delay product
    - Primarily latency-bound *(requires different kind of interconnect)*
  - Math operations limited primarily to reductions involving sum, max, and min operations.
  - Deserves a dedicated network *(significantly different reqs.)*

Whats Next?

- What does the data tell us to do?
  - P2P: Focus on messages that are bandwidth-bound (eg. larger than bandwidth-delay product)
    - Switch Latency=50ns
    - Propagation Delay = 5ns/meter propagation delay
    - End-to-End Latency = 1000-1500 ns for the very best interconnects!
  - Shunt collectives to their own tree network *(BG/L)*
  - Route latency-bound messages along non-dedicated links *(multiple hops)* or alternate network *(just like collectives)*
  - Try to assign a direct/dedicated link to each of the distinct destinations that a process communicates with

Conundrum

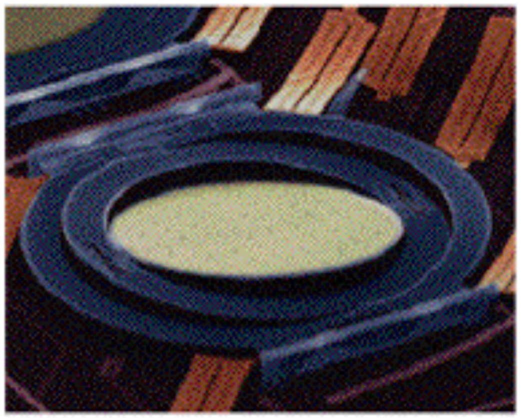
- Can’t afford to continue with Fat-trees or other Fully-Connected Networks (FCNs)
- Can’t map many Ultrascale applications to lower degree networks like meshes, hypercubes or torii
- How can we wire up a custom interconnect topology for each application?

Switch Technology

- **Packet Switch:**
  - Read each packet header and decide where it should go *fast!*
  - Requires expensive ASICs for line-rate switching decisions
  - Optical Transceivers

Force10 E1200

1260 x 1GigE
56 x 10GigE



Two-axis tilting micromirror
(Hecht, 2001, p. 125)

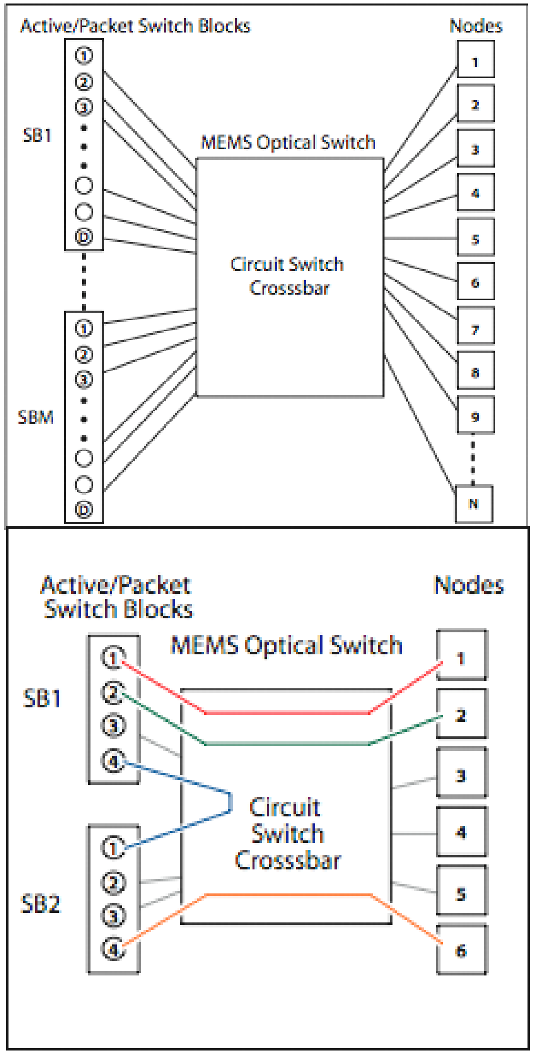
400x400λ
1-40GigE
Movaz iWSS

- **Circuit Switch:**
  - Establishes direct circuit from point-to-point *(telephone switchboard)*
  - Commodity MEMS optical circuit switch
    - Common in telecomm industry
    - Scalable to large crossbars
  - Slow switching *(~100microseconds)*
  - Blind to message boundaries

Office of Science
U.S. DEPARTMENT OF ENERGY

A Hybrid Approach to Interconnects HFAST

- **Hybrid Flexibly Assignable Switch Topology (HFAST)**
  - Use optical circuit switches to create custom interconnect topology for each application as it runs *(adaptive topology)*
  - Why? Because circuit switches are
    - Cheaper: Much simpler, passive components
    - Scalable: Already available in 1024-port crossbars
    - Allow non-uniform assignment of switching resources
  - GMPLS manages changes to packet routing tables in tandem with circuit switch reconfigurations



HFAST

- **HFAST Solves Some Sticky Issues with Other Low-Degree Networks**
  - **Fault Tolerance: 100k processors… 800k links between them using a 3D mesh** ***(probability of failures?)***
  - **Job Scheduling: Finding right sized slot**
  - **Job Packing: n-Dimensional Tetris…**
  - **Handles apps with low comm degree but not isomorphic to a mesh or nonuniform requirements**

- **How/When to Assign Topology?**
  - **Job Submit Time:** ***Put topology hints in batch script (BG/L, RS)***
  - **Runtime:** ***Provision mesh topology and monitor with IPM. Then use data to reconfigure circuit switch during barrier.***
  - **Runtime:** ***Pay attention to MPI Topology directives (if used)***
  - **Compile Time:** ***Code analysis and/or instrumentation using UPC, CAF or Titanium.***

HFAST Outstanding Issues
Mapping Complexity

- Simple linear-time algorithm works well with low TDC but not for TDC > packet switch block size.

- Use clique-mapping to improve switch port utilization efficiency
  - The general solution is NP-complete
  - Bounded clique size creates an upper-bound that is < NP-complete, but still potentially very large
  - Examining good "heuristics" and solutions to restricted cases for mapping that completes within our lifetime

- Hot-spot monitoring
  - Gradually adjust topology to remove hot-spots
  - Similar to port-mapper problem for source-routed interconnects like Myrinet

Conclusions/Future Work?

- Not currently funded
  - Outgrowth of Lenny's vector evaluation work
  - Future work == getting funding to do future work!
- Expansion of IPM studies
  - More DOE codes *(eg. AMR: Cactus/SAMARAI, Chombo, Enzo)*
  - Temporal changes in communication patterns *(AMR examples)*
  - More architectures *(Comparative study like Vector Evaluation project)*
  - Put results in context of real DOE workload analysis
- HFAST
  - Performance prediction using discrete event simulation
  - Cost Analysis *(price out the parts for mock-up and compare to equivalent fat-tree or torus)*
  - Time domain switching studies *(eg. how do we deal with PARATEC?)*
- Probes
  - Use results to create proxy applications/probes
  - Apply to HPCC benchmarks *(generates more realistic communication patterns than the "randomly ordered rings" without complexity of the full application code)*