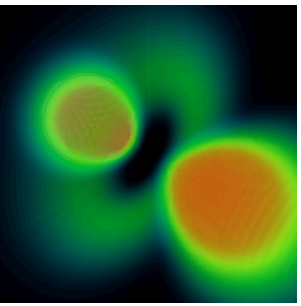
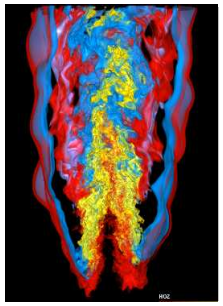
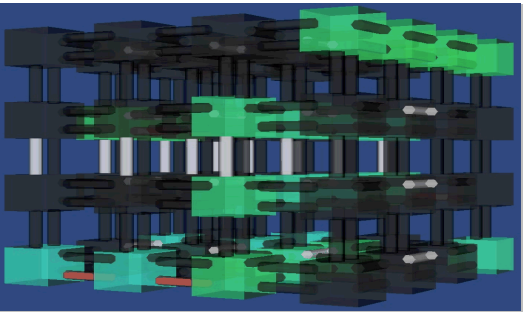
SAND2013-3701C







The Role of Optical Links in HPC System Interconnects

Gilbert Hendry

Sandia National Laboratories

ghendry@sandia.gov

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

NNSA

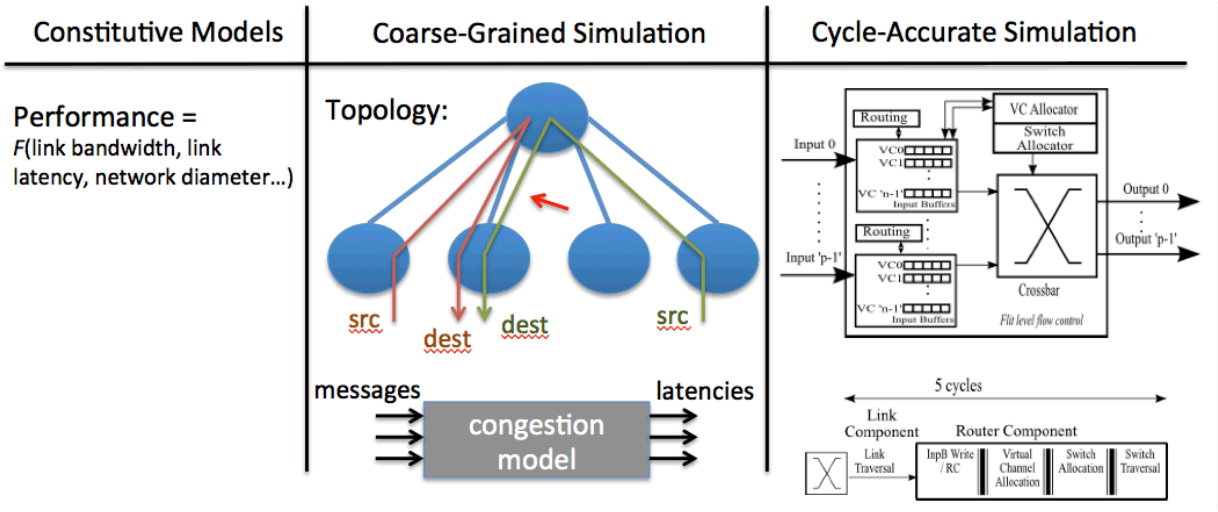
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

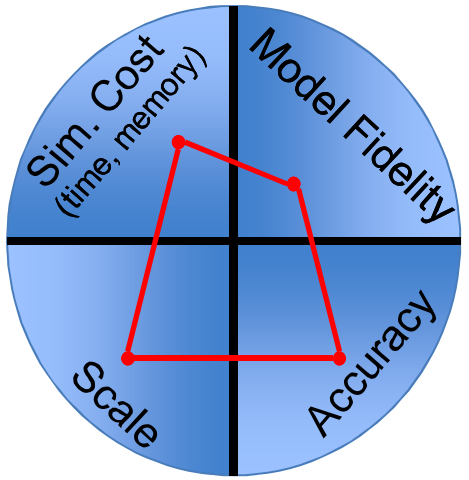
Goal of this talk

Sandia National Laboratories

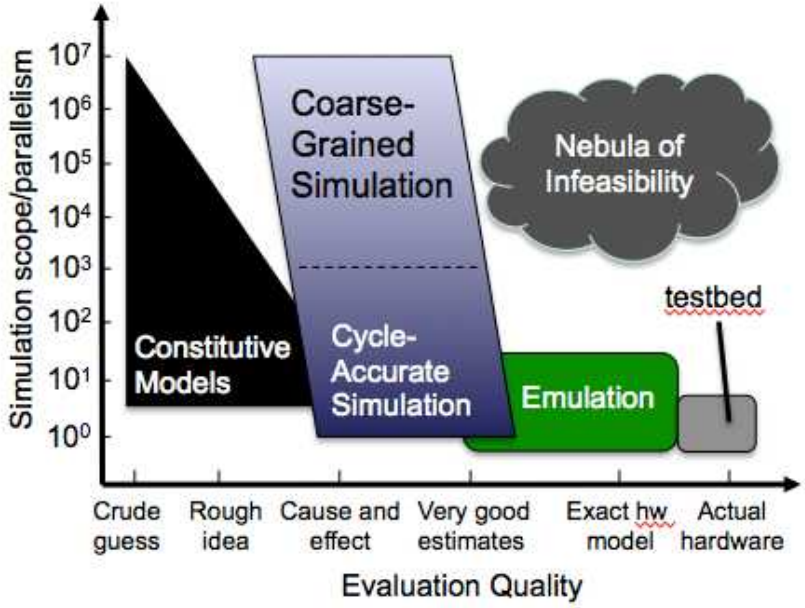
- Currently and in near future: some links in the network are optical (active cabling) to span distances
  - Turns out it's a good bandwidth/$ point
- Provide system-level motivation for optical link cost/performance by showing real applications needs on typical network topologies
- Questions:
  - Would I want faster links (more bandwidth) if I could have them?
  - Could I spend less money on cheaper (slower) optical links?

When is Macroscale/Coarse-Grained Simulation Appropriate?





- Constitutive Models – can be powerful in reasoning about system and tradeoffs, but hard to investigate new concepts and complex interactions
- Coarse-Grained Simulation – moderate accuracy, predicts trends, can scale, but requires approximations
- Cycle-Accurate Simulation – highly accurate for detailed studies, but limited ability to scale
- Emulation – essentially exact and fast, but expensive to scale
- Testbed/prototype – provides real numbers, but time/cost a major factor



The SST/Macro Simulator

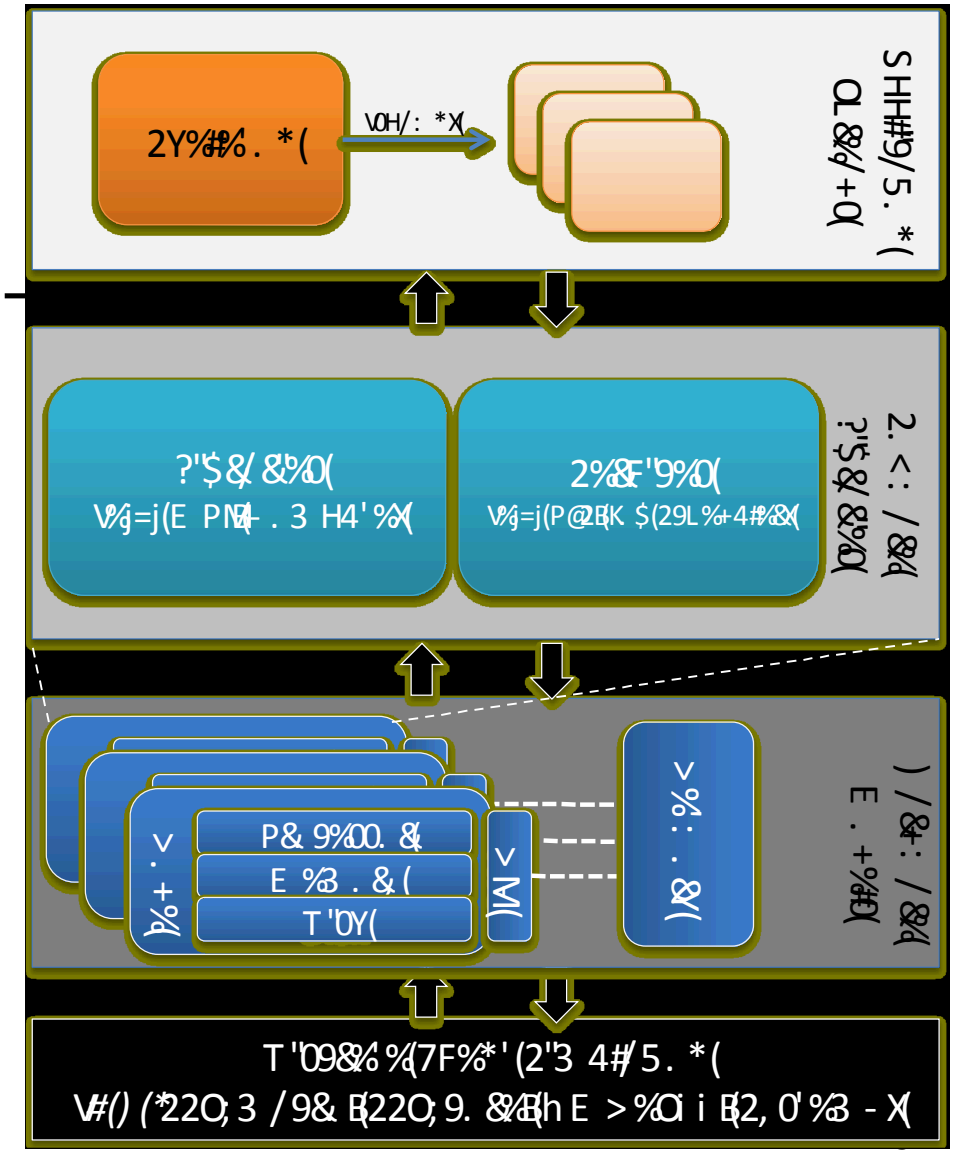
Sandia National Laboratories

- General webiste: sst.sandia.gov
  - Note: SST/Macro is separate from SST
- Open source code, C++ with C/Fortran interfaces
  - Easily modified or extended with new models, topologies, metrics, etc.
- Runs as single process (single address space)
  - Application processes modeled as user-space threads
  - Global data requires special attention (false sharing)
- Offline (trace-based) mode for MPI applications
- Online (skeleton) mode
  - MPI, sockets, OpenSHMEM, HPX
  - Coming: UPC, ARMCI, GASnet, Global Arrays
- Downloads, documentation, issue tracker at http://sst.sandia.gov

SST/macro: A Coarse-Grained Simulator

An application code with minor modifications

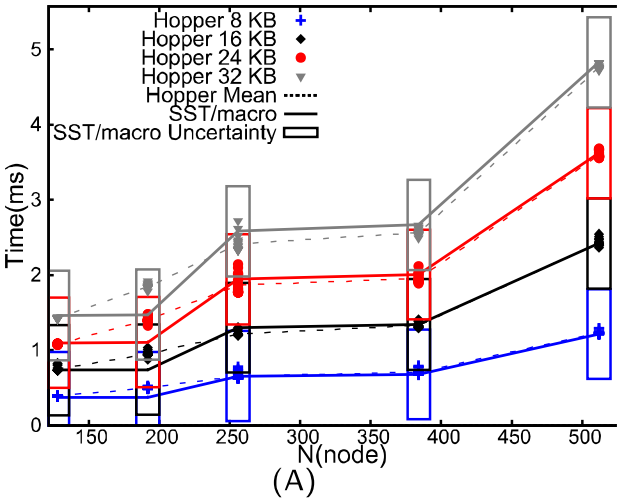
SST/macro-specific implementation of interfaces (such as MPI), which simulate execution and communication



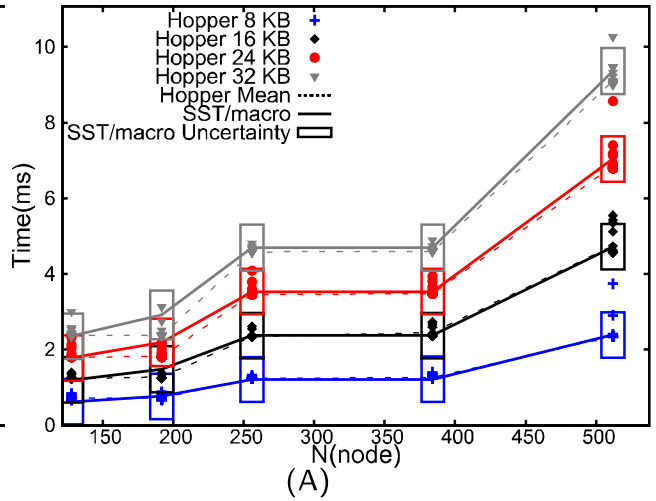
SST/macro Validation

- Validation study has been completed against a Cray XE6 using packet train model
  - sstmacro/configurations/hoppertrain.ini
- Able to capture congestion behavior that results from both hardware and MPI implementation effects
- Demonstrated simulation validation workflow that includes formal UQ methods
- Working on packaging up the UQ tools

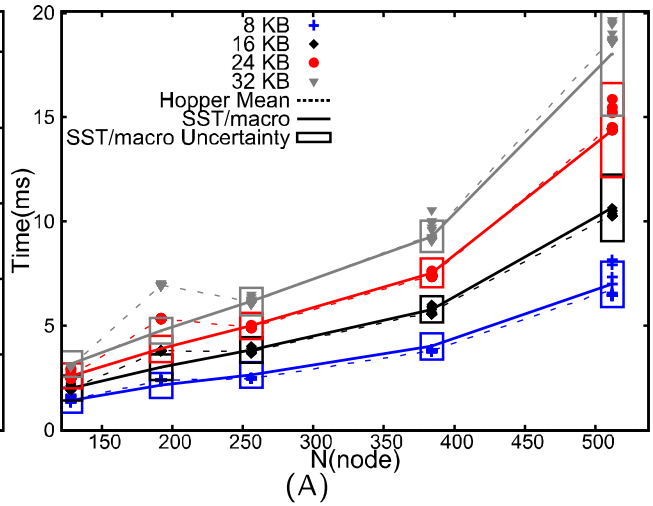
MPI_Scatter



MPI_Gather



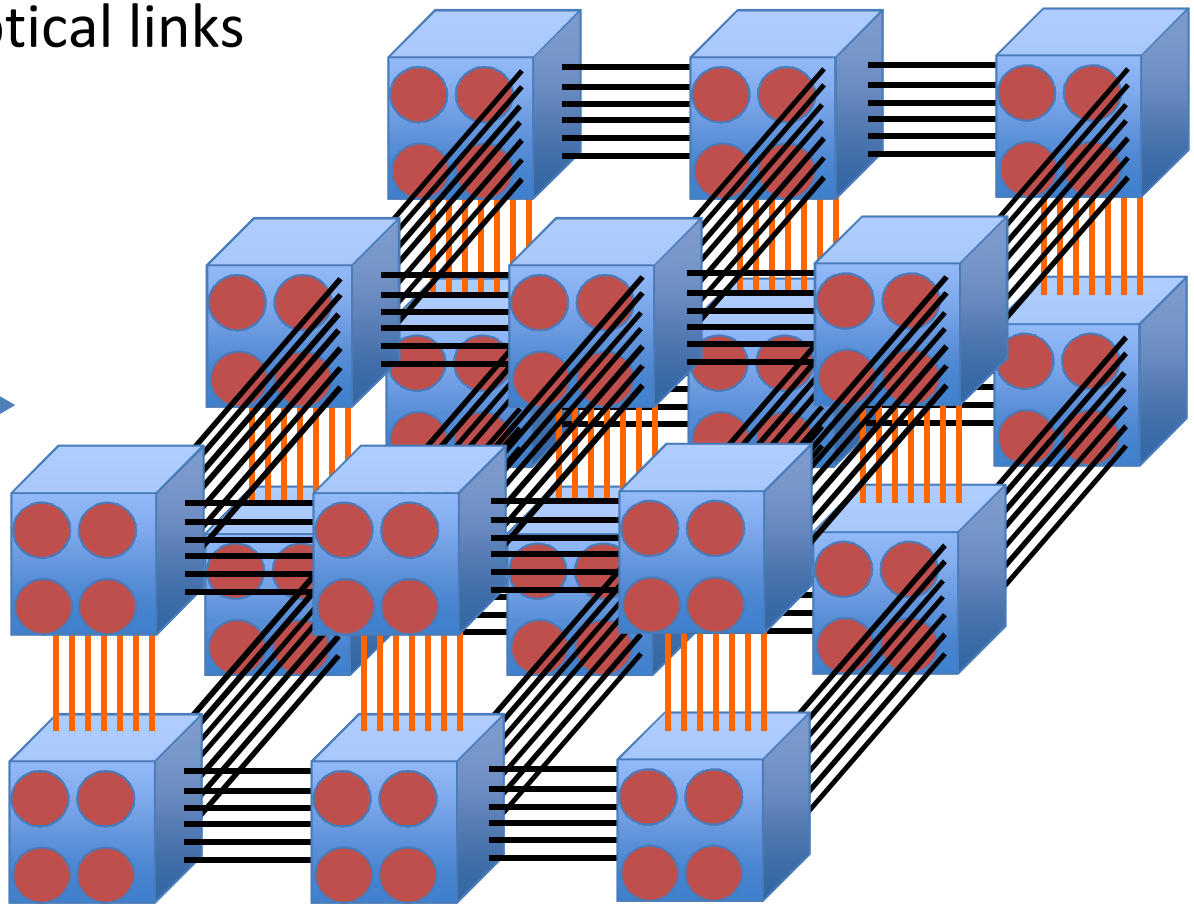
MPI_Allgather

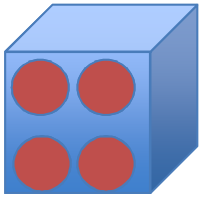


Networks under consideration

- 7-D Torus
- 3 dimensions use optical links
- DOR routing

3D Mesh, you get the idea

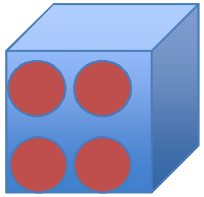




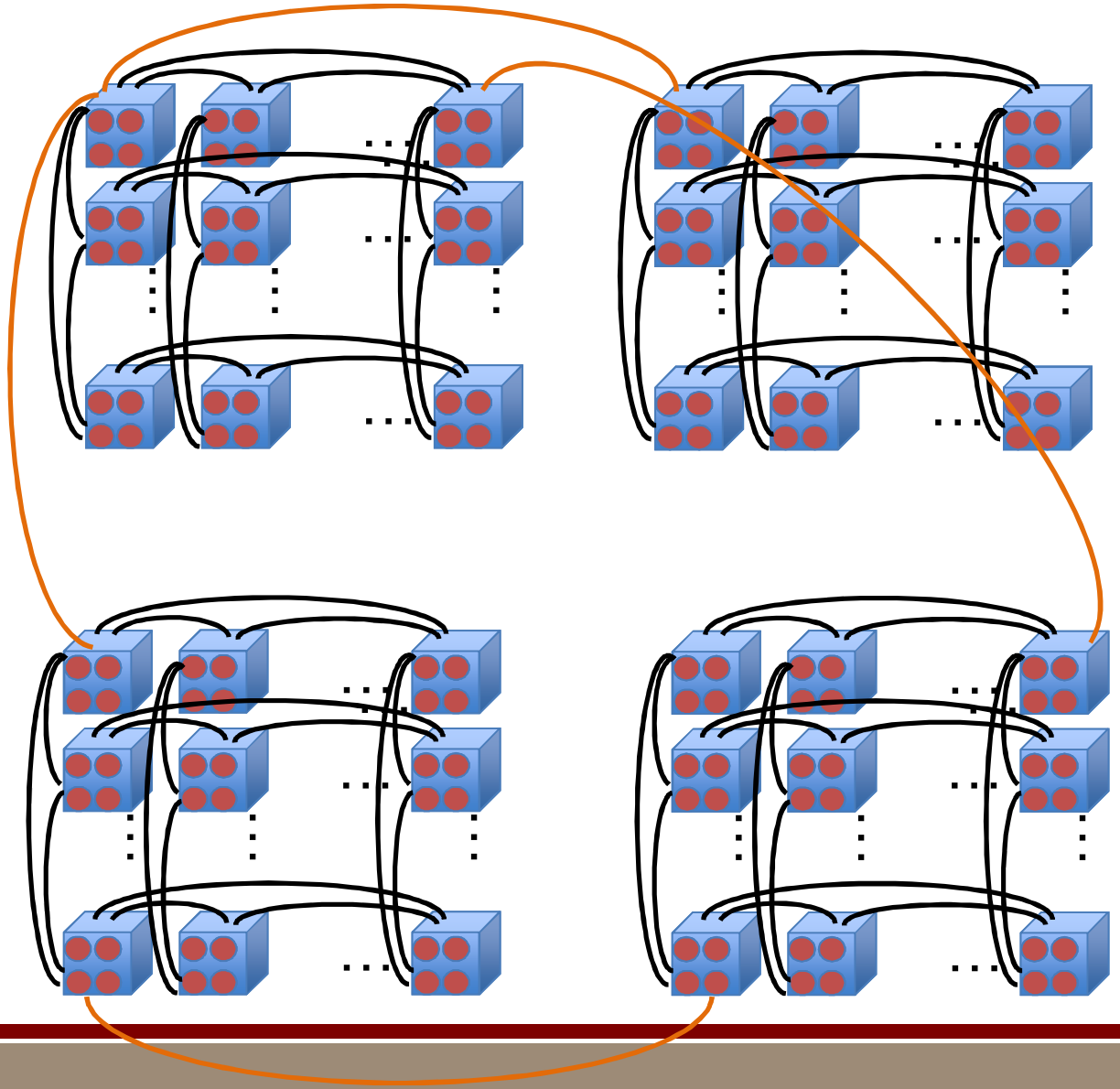
= router with 4 nodes attached to it

Networks: Dragonfly

- 8x8 groups
- Local and global valiant routing
- Links to other groups are optical

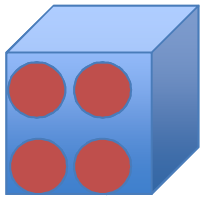


= router with 4 nodes attached to it

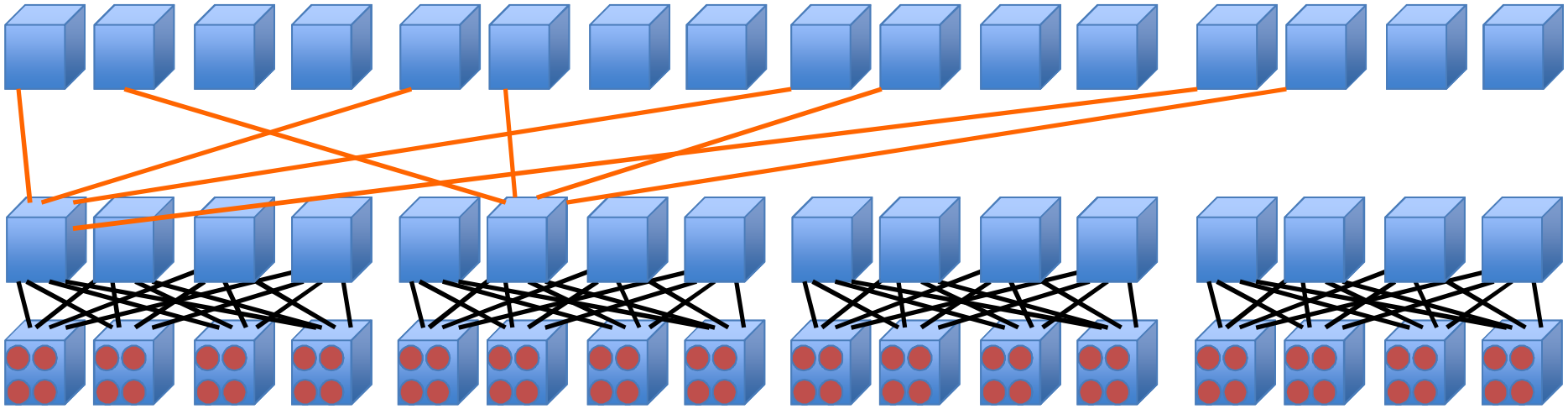


Networks: Fat-Tree

- 4-ary 7-tree
- Links to core switches are optical



= router with 4 nodes attached to it

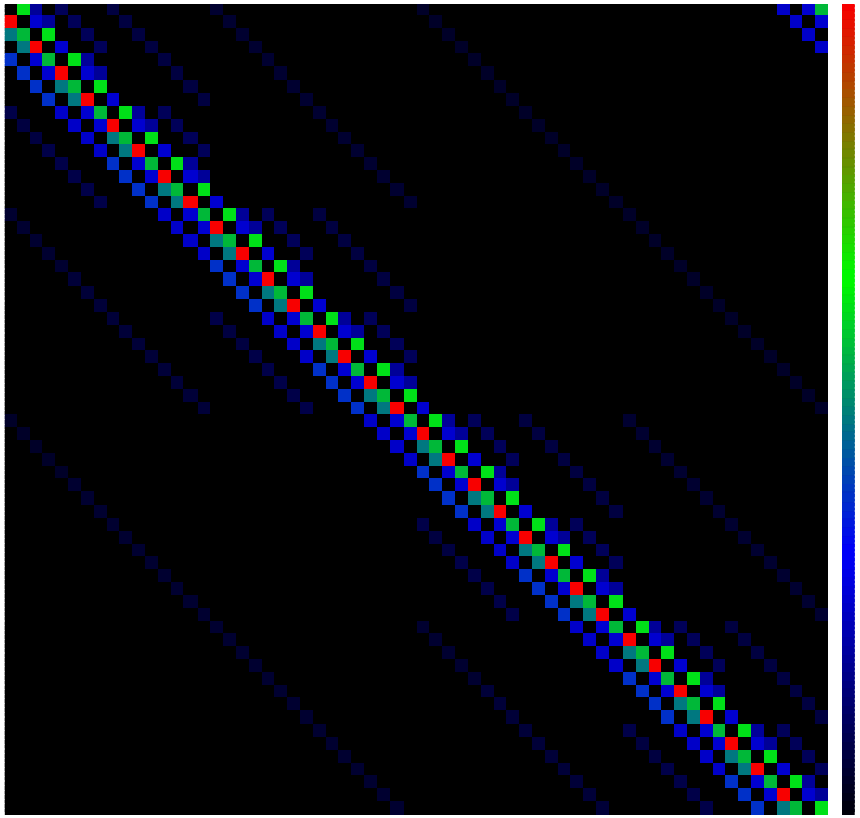


Applications: overall

- Running real code, so can have dynamic interactions at runtime
- Skeletonized (removed most computation)
- Come with SST/macro

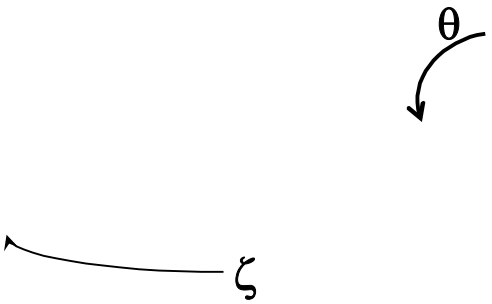
Applications: GTC (NE/physics)

- Particle-in-cell full application



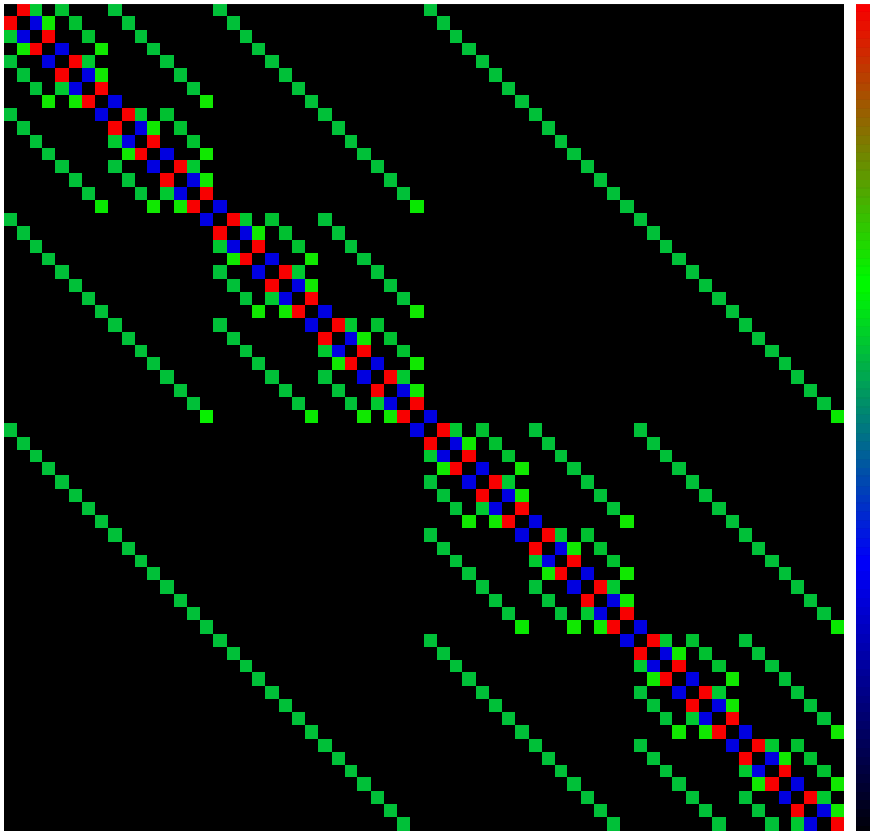
(neighbor-messages are pretty big, they wash out the small but numerous collectives)

- GTC uses PIC method to simulate plasma microturbulence for fusion devices
- Written in MPI + F90
- Scalable to thousands of processors (weak)



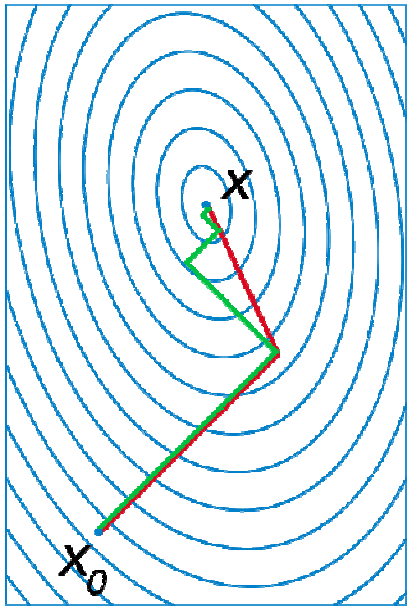
Applications: HPCCG (solver)

- Conjugant gradient solver from www.mantevo.org



(small problem so you can see the small collectives)

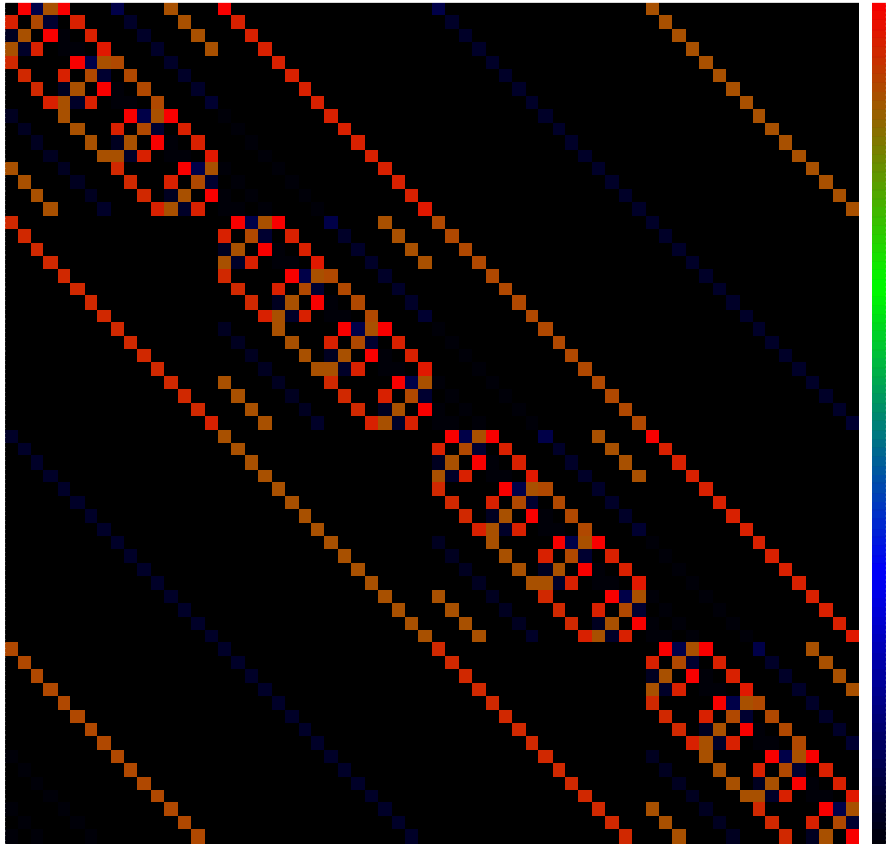
- Iterative/convergent method useful for solving systems of equations that are sparse
- Weak scales by default



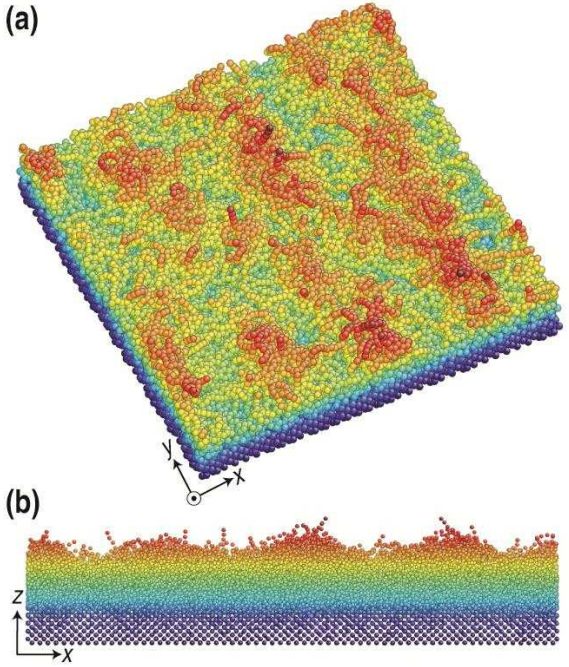
(Wikipedia)

Applications: miniMD (bio/materials)

- Molecular dynamics proxy from www.mantevo.org



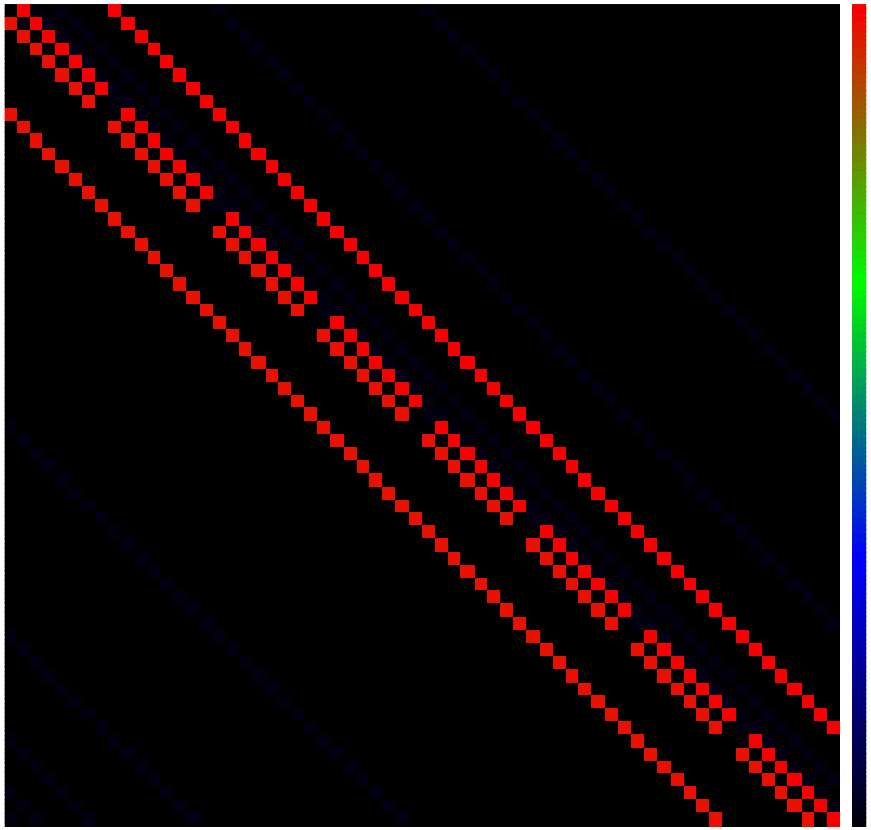
- Meant as a proxy to LAMMPS[1]
- Strong scales
- which can do this (amorphous carbon film growth [2]):



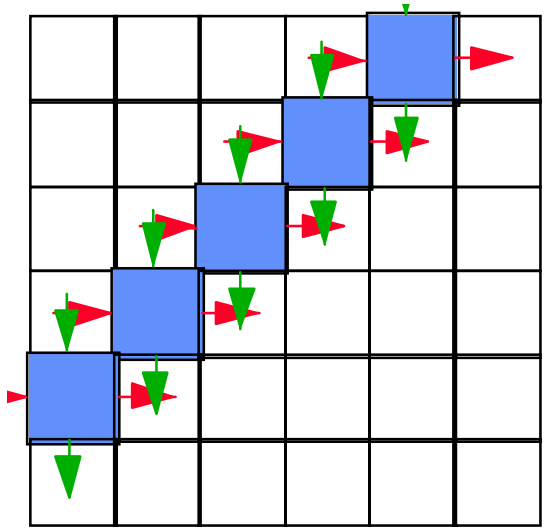
[1] "LAMMPS molecular dynamics simulator," 2009. [Online]. Available: http://lammps.sandia.gov/index.html
[2] Molecular dynamics simulation study of the growth of a rough amorphous carbon film by the grazing incidence of energetic carbon atoms, M. Joe, M.-W. Moon, J. Oh, K.-H. Lee, K.-R. Lee, Carbon, 50, 404 (2012).

Applications: sweep3D

- Sweep3D [1] is a particle transport benchmark



- 2-D transport sweep along a diagonal wavefront:



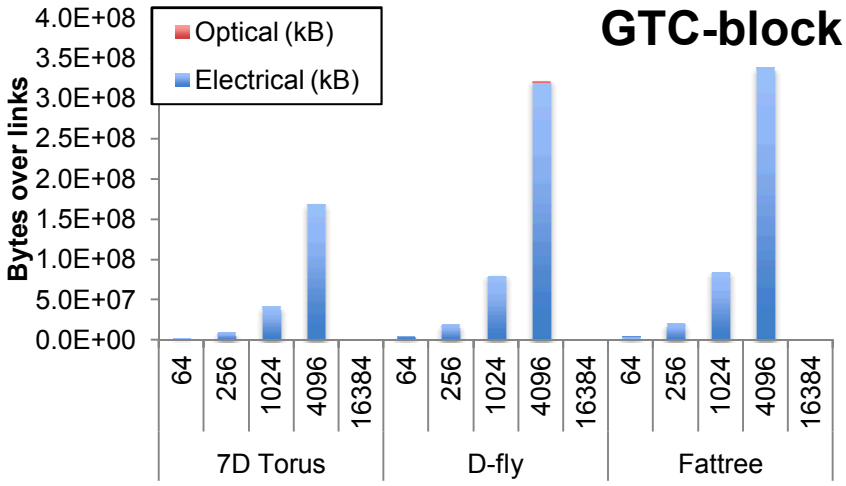
[1] A. Hoisie, H. Lubeck, and H. Wasserman, “Performance and scalability analysis of teraflop-scale parellel architectures using multidimentional wavefront applications,” Int. Journal of High Performance Computing Applications, vol. 14, no. 4, p. 330346, 2000.

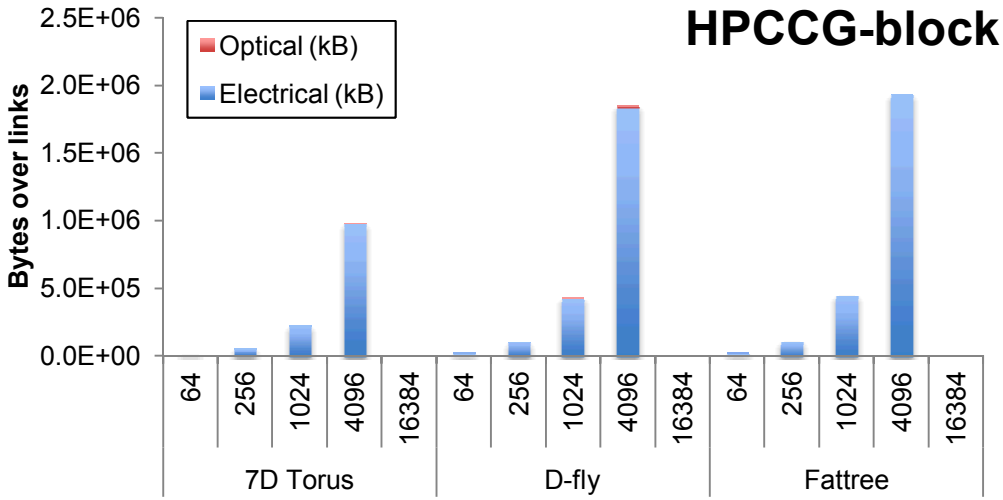
Experiment: Link Usage

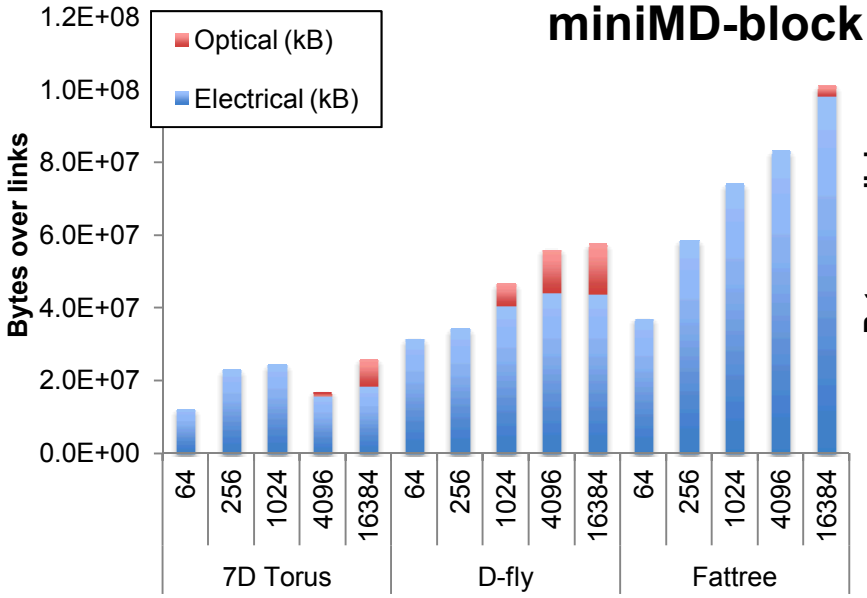
Sandia National Laboratories

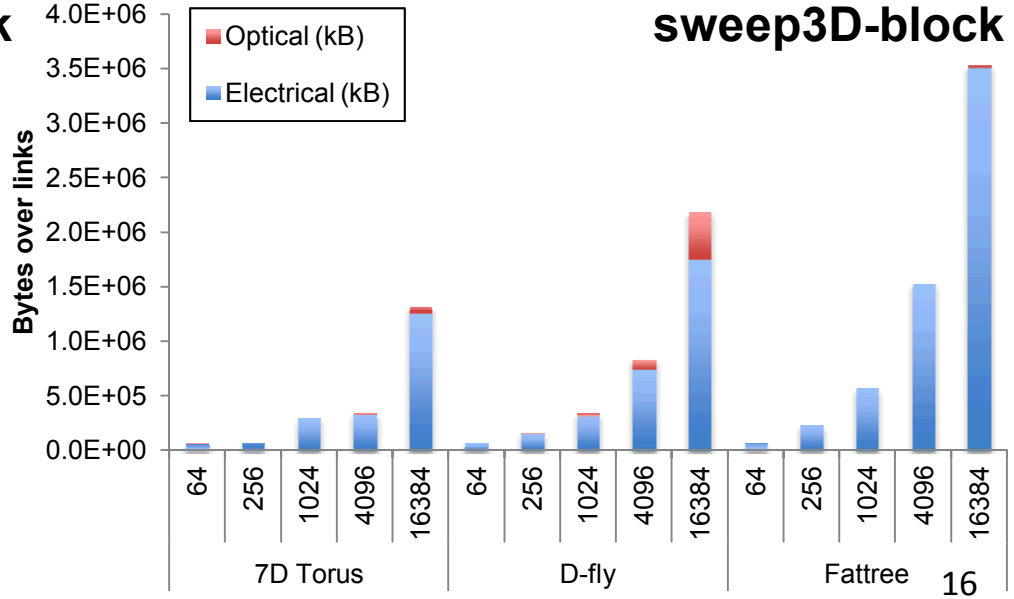
- How much data traverses the optical links for each application?
  - compared to the rest of the links
  - for both canonical and random mapping
- Mapping/allocation plays an important role:
  - Block/canonical: like a blue Gene
  - random: more like a Cray

Results: Link usage: block allocation

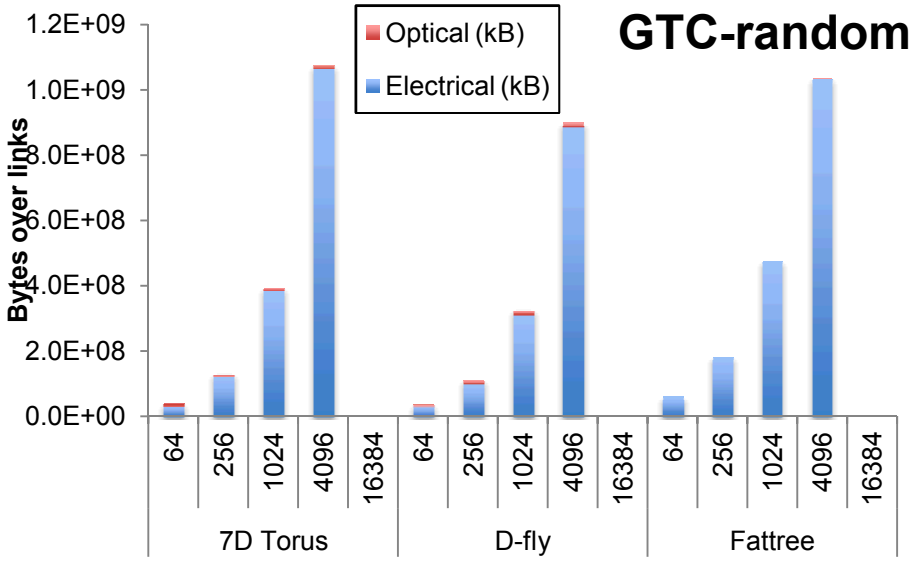


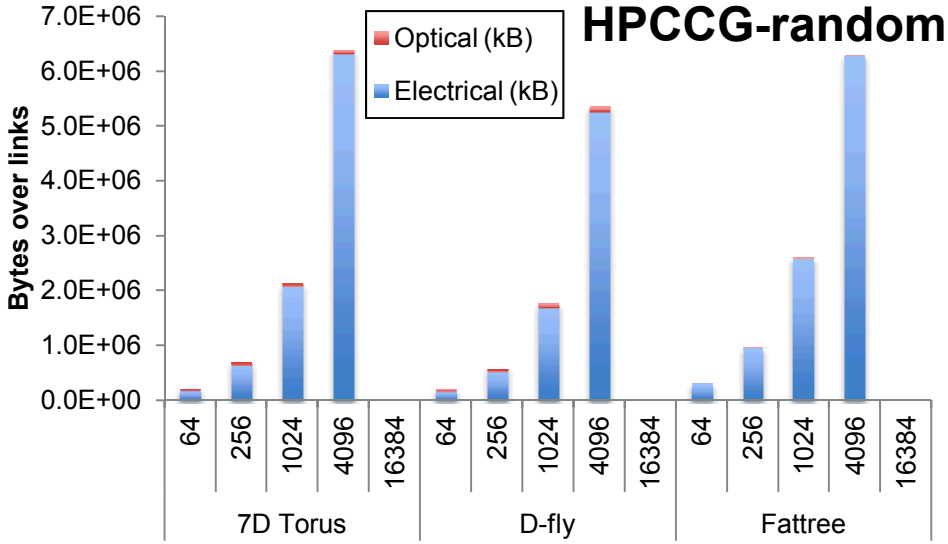


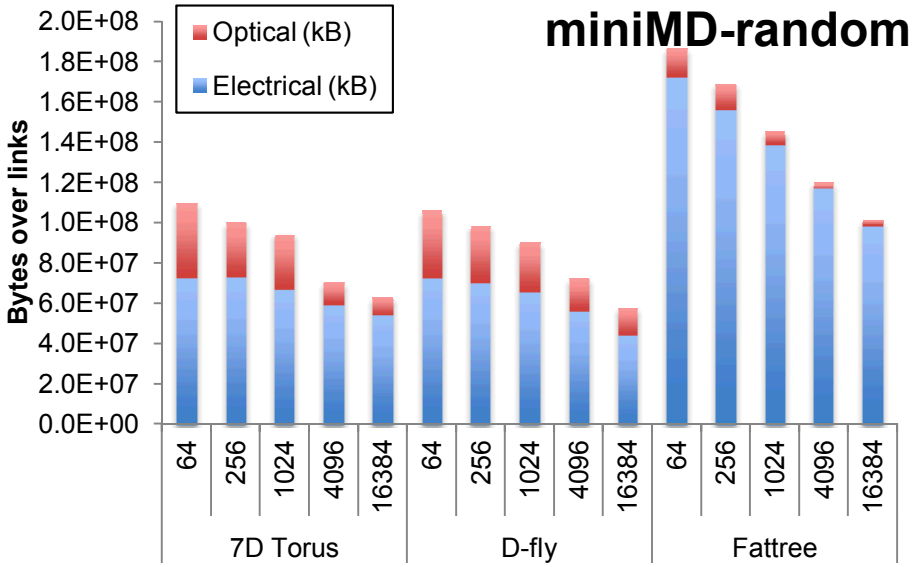


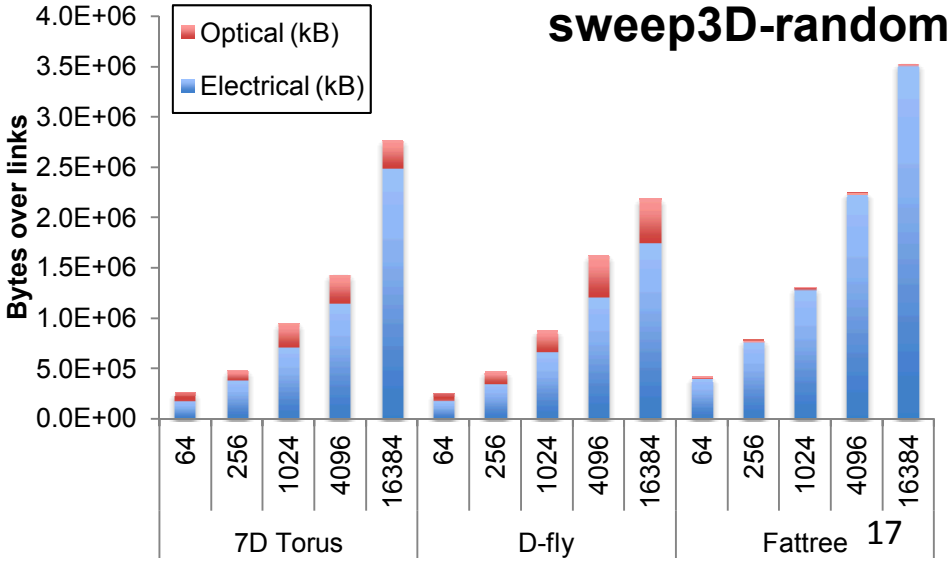


Results: Link usage: random allocation







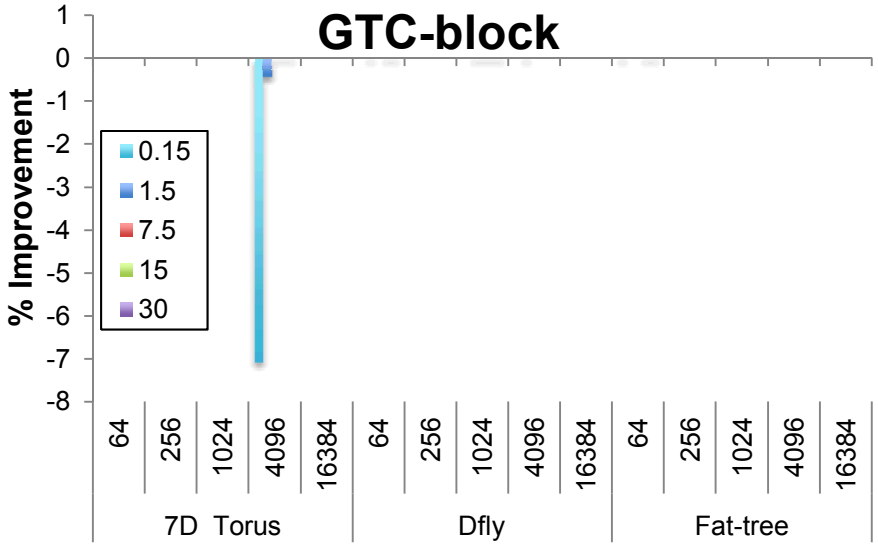


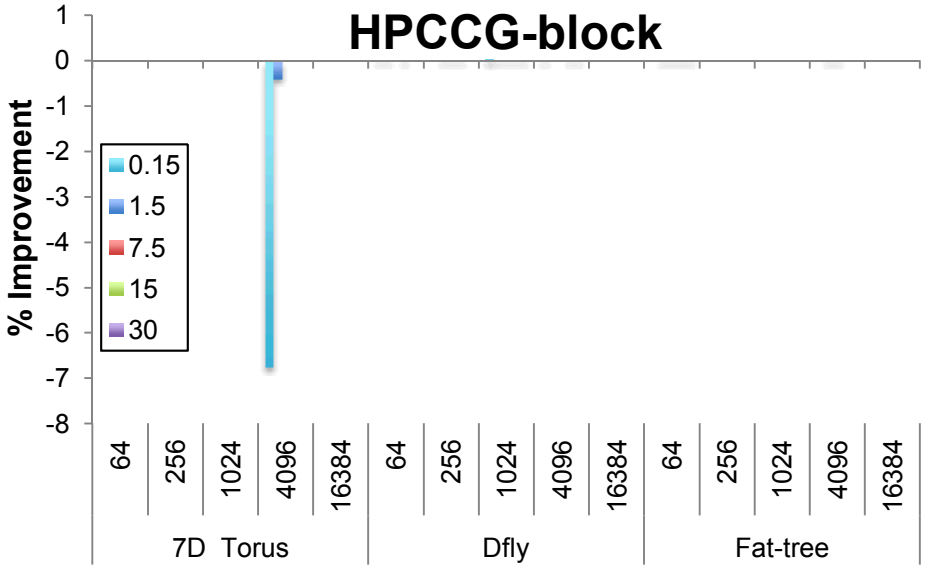
Experiment: Performance

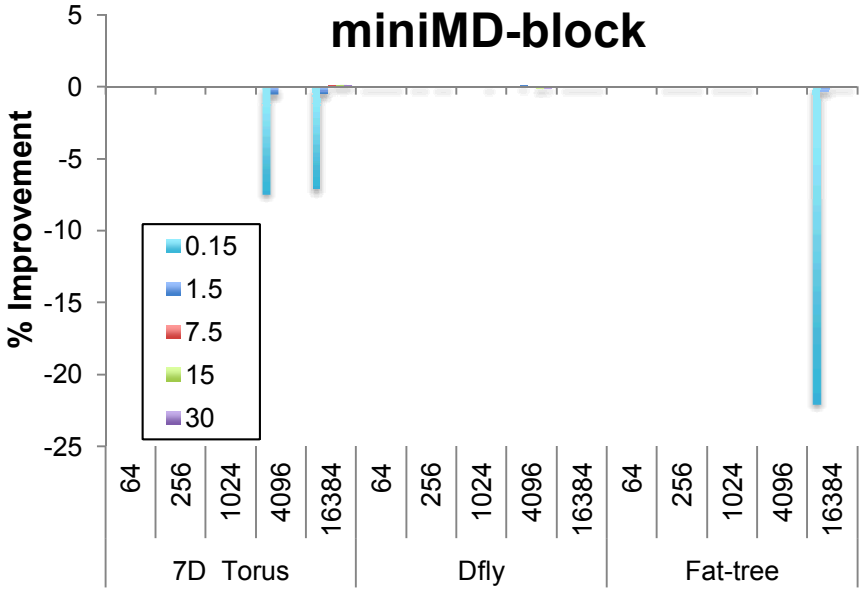
Sandia National Laboratories

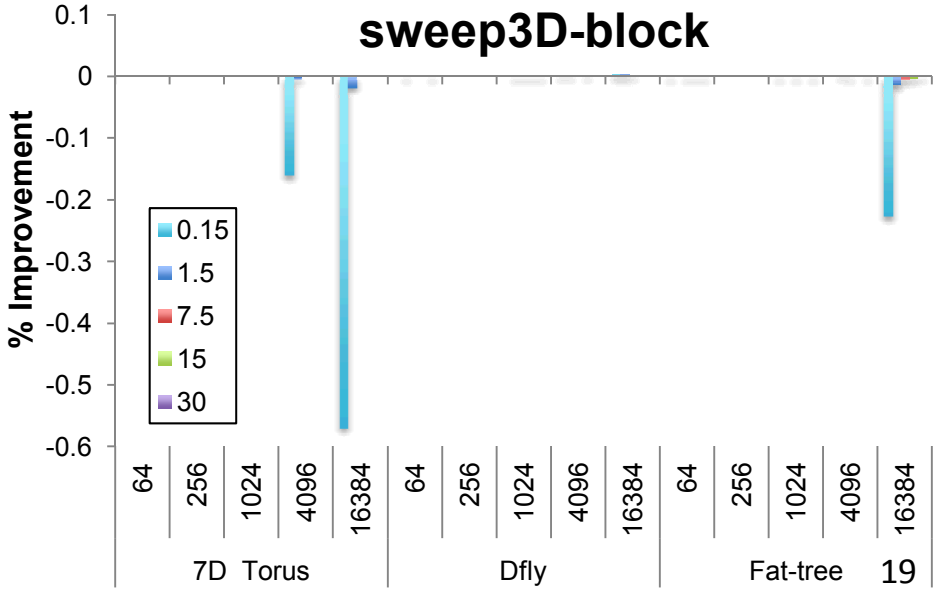
- How much does optical link bandwidth affect overall performance?
  - If we could live with slower links, can we make them cheaper or lower power
  - How much do we gain from faster links
- Switch and NIC performance cranked up
  - so they aren't a constraint, presumably you'd engineer them that way

Results: Performance: block allocation

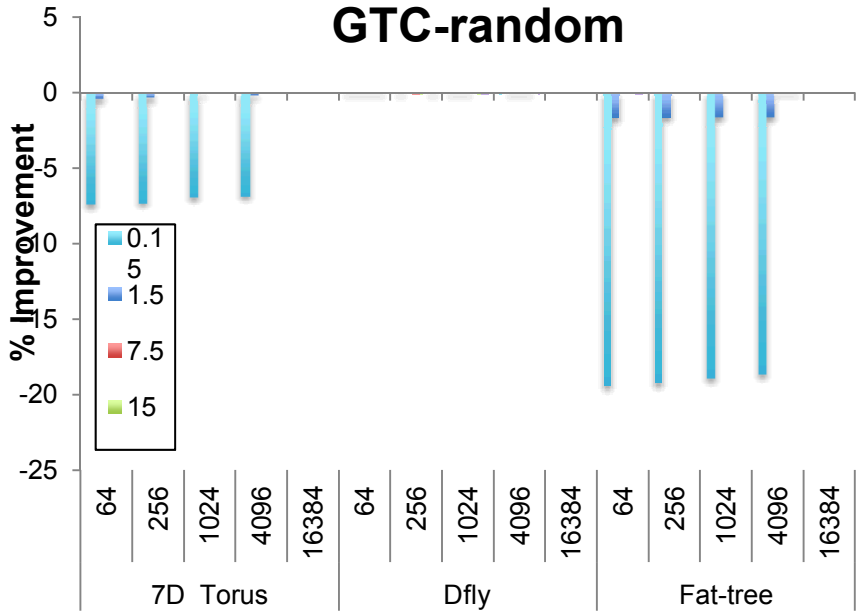


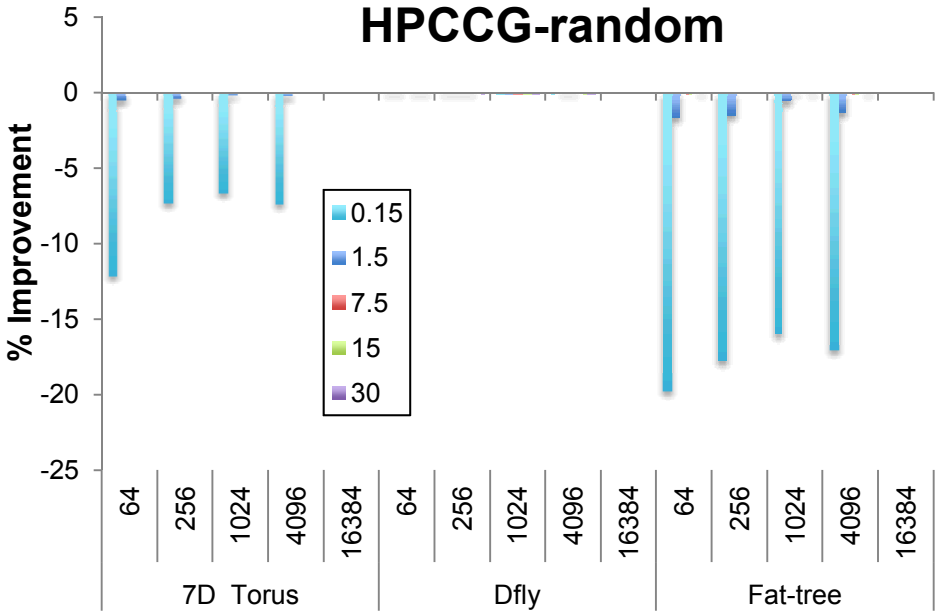


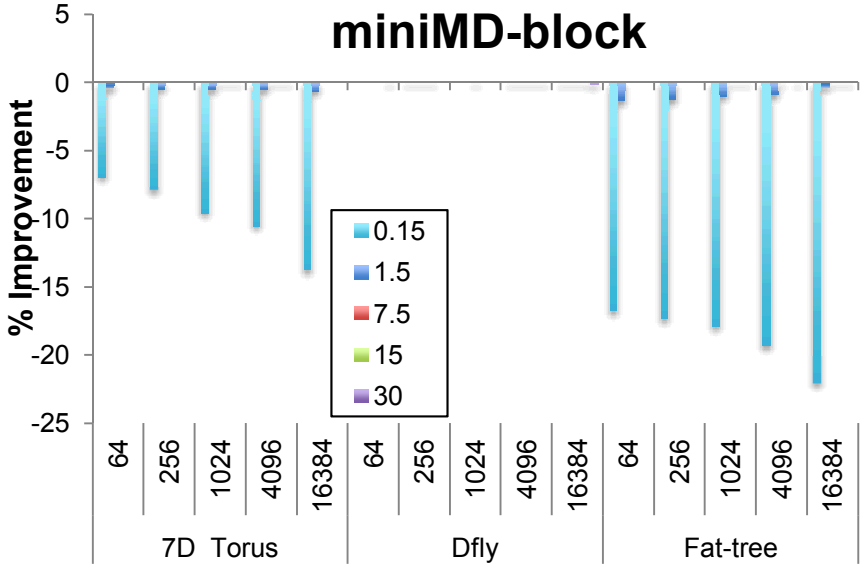


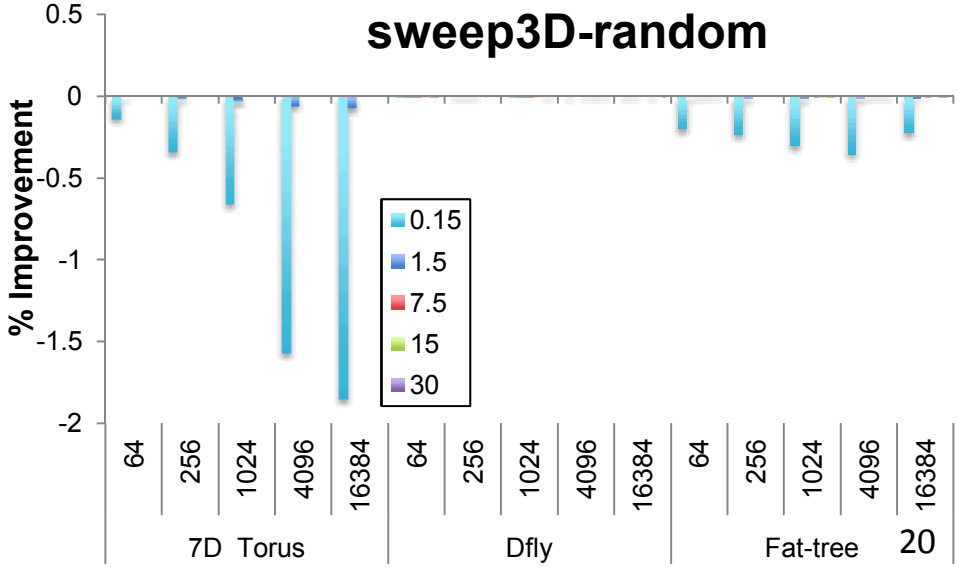


Results: Performance: random allocation









Conclusions

- If using a torus: just don't make them any slower
- If using a Dragonfly: doesn't really matter
- If using a Fat-tree: not much makes it into the root switches

- Overall: just make optical links cheaper and/or lower power

- Try it yourself:

http://sst.sandia.gov/using_sstmacro.html