Interconnection Architectures for Petabyte-Scale High Performance Storage Systems

Ethan L. Miller
elm@cs.ucsc.edu
Storage Systems Research Center
Univ. of California, Santa Cruz

Andy D. Hospodor
andy.hospodor@ieee.org
Senior Member, IEEE

2004 IEEE / Goddard Conference on
Mass Storage Systems & Technologies

Interconnection architectures for storage

- Petabyte-scale storage systems have thousands of disks that must be connected
  - To storage system clients
  - To each other
- Parallel processors have had this problem for years—what's different about storage?
  - Storage systems are less latency-sensitive
  - Storage systems *must* tolerate (multiple) failures
  - Storage systems may be more cost-sensitive
- Goal: use off-the-shelf networking components to build a scalable interconnection network for storage

Overview

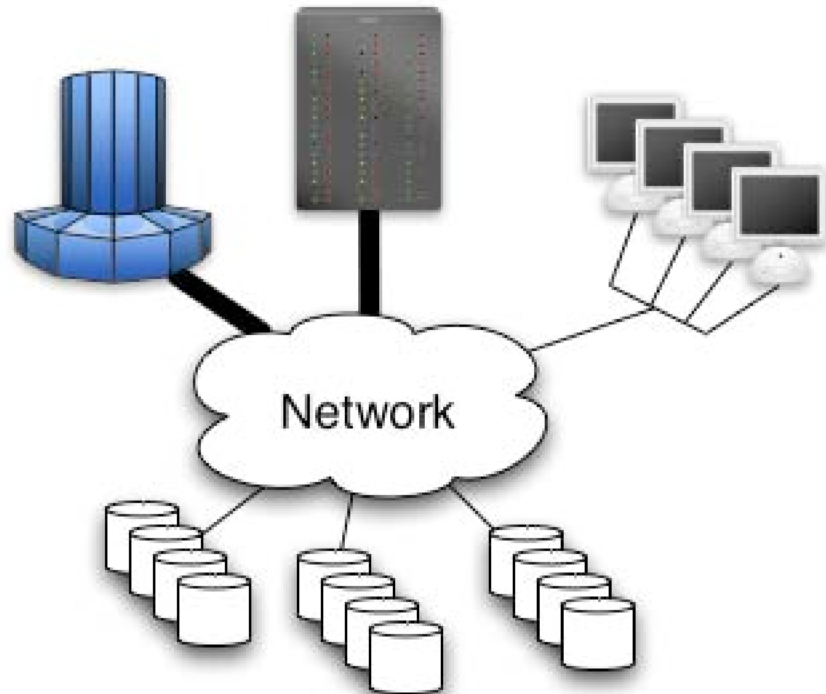
- Why study interconnection architectures for storage?
- Interconnection network basics: cost & performance
- Interconnection network designs
- Design evaluations
  - Cost
  - Performance
  - Complexity
- Directions for future research
  - Resilience to link failures
  - Performance under storage failures

Interconnection network basics

- Commodity networking hardware (currently) is:
  - Gigabit Ethernet connections
  - Single-chip switches for 6–8 connections
  - One port costs around $25 today
  - Latency isn't great, but OK for storage
- Faster hardware might be 10Gb Ethernet
  - Ports are very expensive ($2000+)
  - Switching bandwidth is an issue: full 8-port crossbar for 10GbE must support 80Gb/s
- Cost-performance favors commodity networking for most connections
  - This assumes we can actually build a sufficiently fast network…

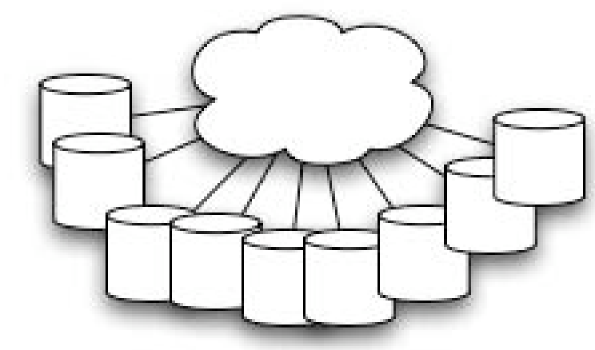
What does the interconnection network do?

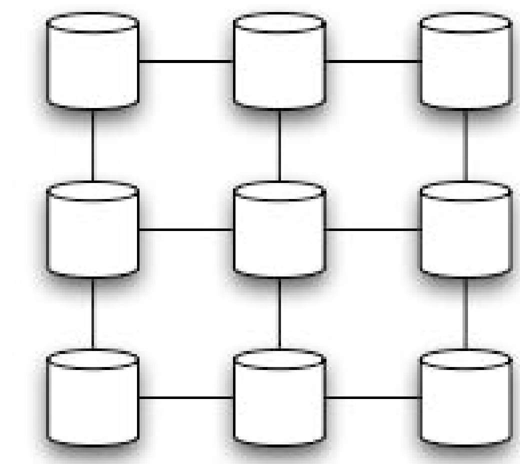
- Connects storage to clients
  - Supercomputers
  - Computing clusters
  - Workstations
  - Per-link bandwidth depends on type of system
    - More lower-speed links OK for clusters
    - Need possibility for high-speed links too
- Network connects disks to one another
  - Replication & load-balancing
  - Redundancy in case of failed disks or links
  - Network need not be monolithic



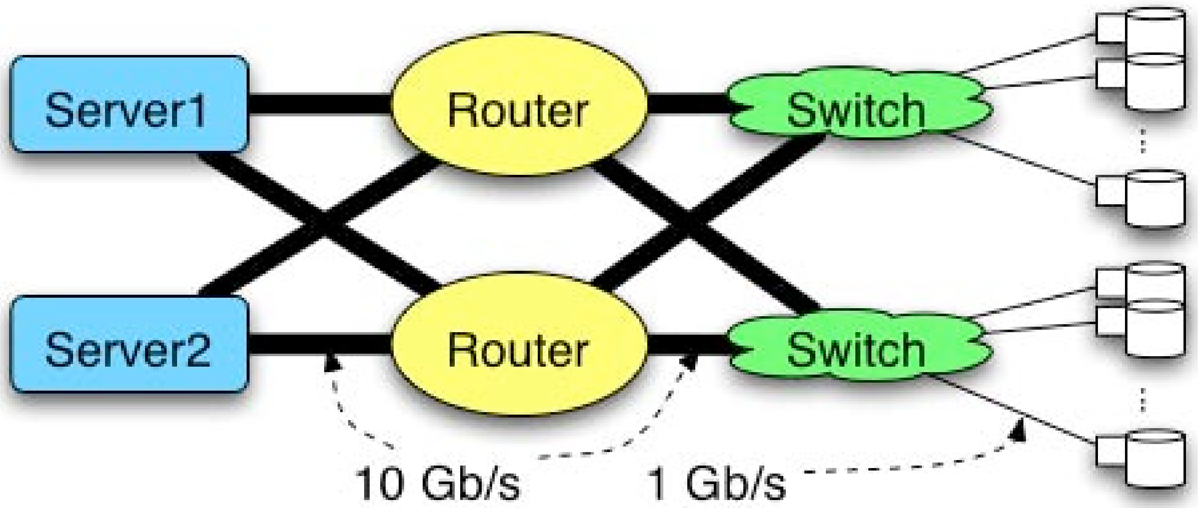
Basic network design tradeoffs

- Two basic options for interconnection networks for storage
- Switching network with disks, clients on the perimeter
  - Perhaps harder to build
  - Scalable?
- Switching network with disks embedded in the network
  - Build it like a cluster computer—scalable, easier to build?
  - Sufficient bandwidth and redundancy?



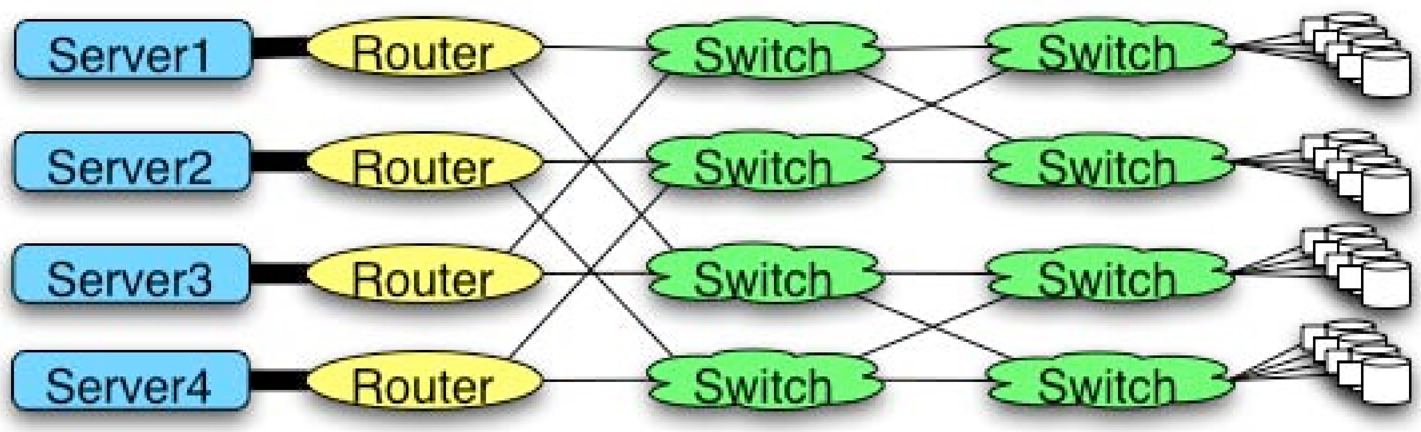


Independent storage clusters & fat trees



- Build independent units from many disks
- Attach each unit to a single client node
- Use redundancy so single component failure doesn't result in unavailable data
- Disadvantages
  - Difficult to aggregate lots of disks this way
  - High-bandwidth links are expensive

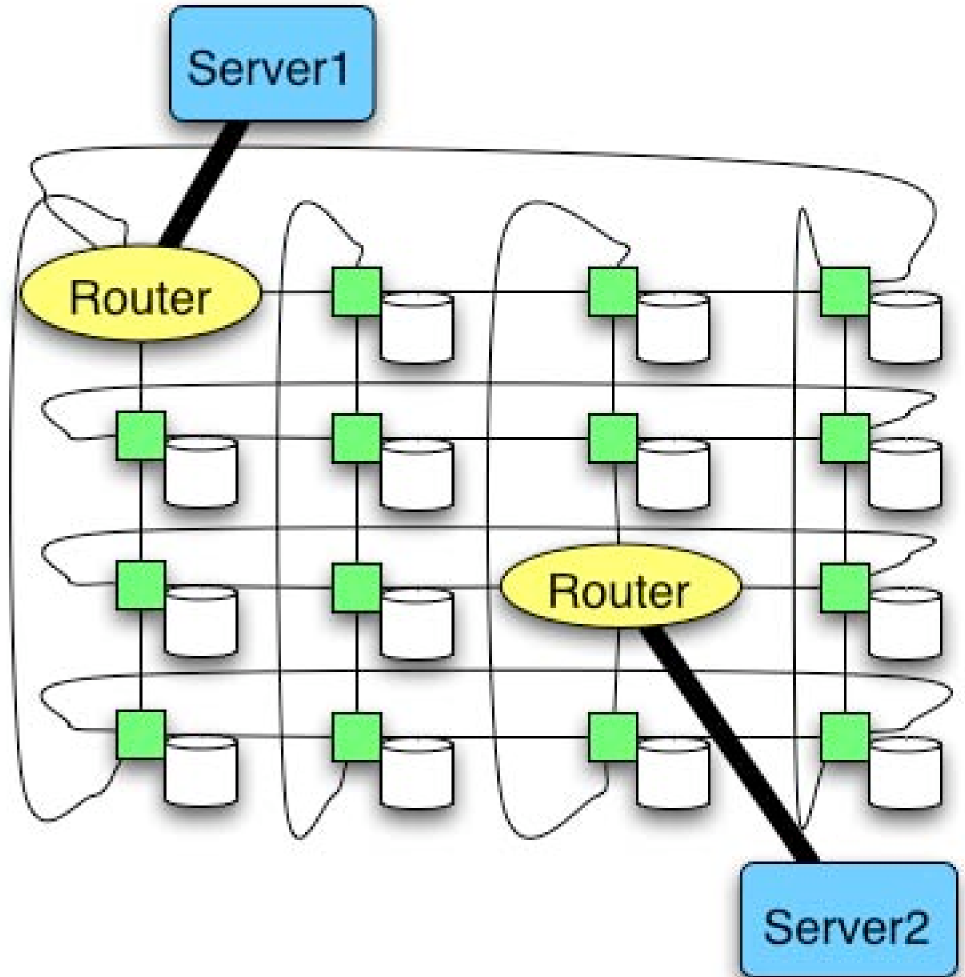
Butterfly network



- More scalable network
  - Several layers of switching (depends on number of disks)
  - Uses more, but lower-speed, links
- Single component failure can make a disk unreachable
  - Unique path from disk to server
- Disks still at the “edge” of the network
  - May add disks and switches separately

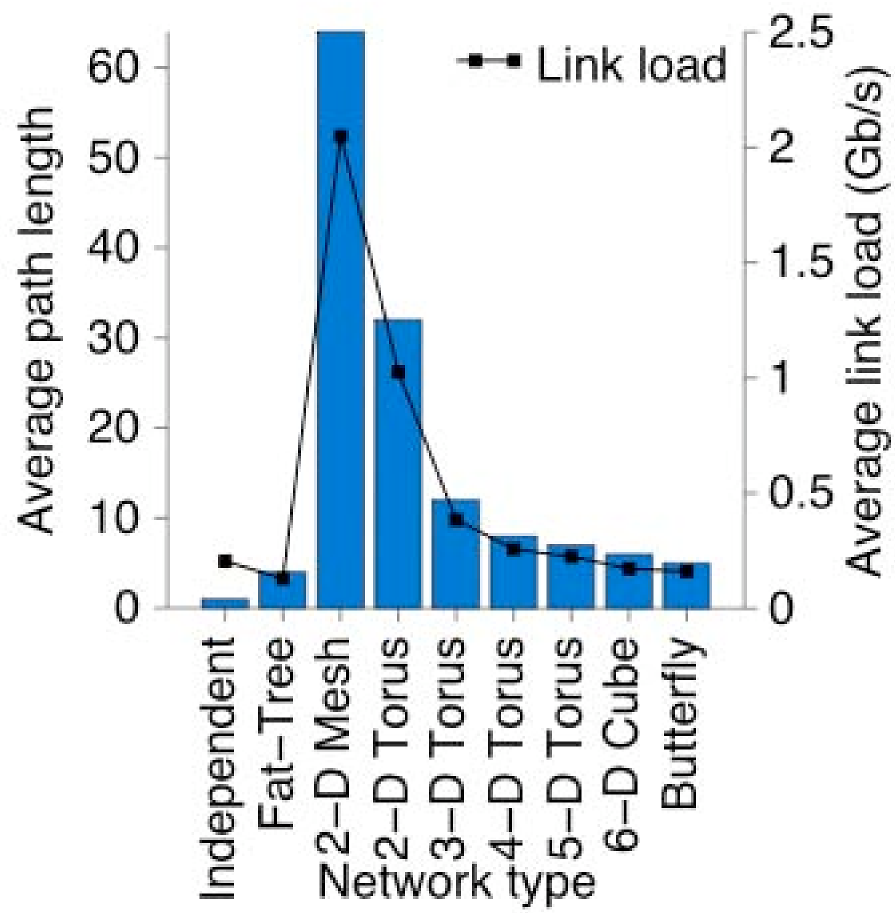
Mesh & torus networks

- Disks embedded in the network
  - Switch at each disk
  - Build storage system from “bricks”
- Dimensionality of network depends on number of ports on each switch
  - Fewer -> cheaper
  - More -> faster
- Some of the “dimensions can be contained within a single brick



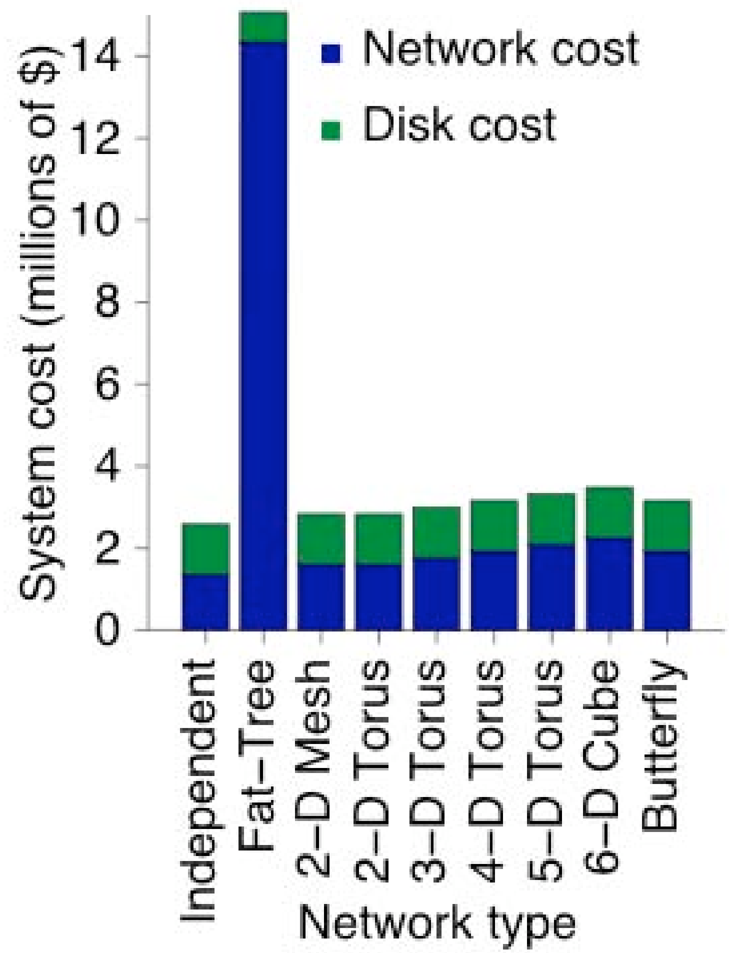
Overall network performance

- Link load is very high for low-dimensional torii
- Link load is low for high-dimensional torii, butterfly
- High link load is caused by long path lengths
  - Butterfly has constant-length (relatively short) paths
  - Torii have more variation in path lengths
- How much does each network cost?



Network cost

- Independent is cheap, but not all storage is connected to all clients
- Fat tree is fast but very expensive
- Torii become more expensive as dimensionality increases
- Butterfly is about the same cost as a mid-dimensional torus

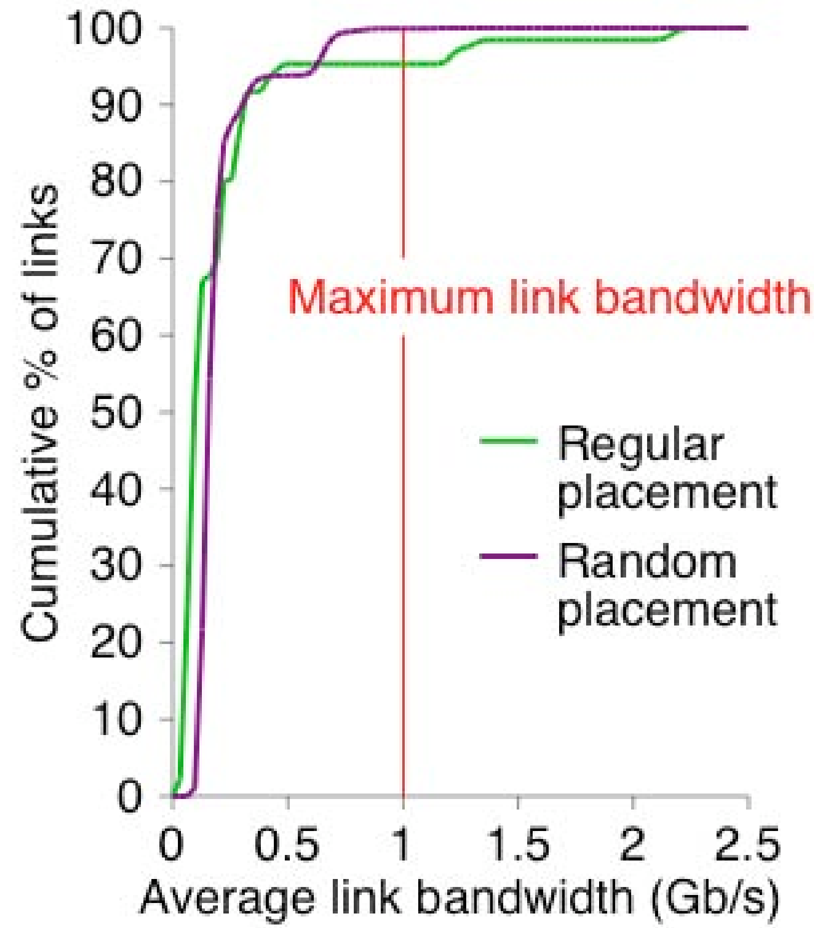


Issues with networks for storage

- Data distribution in storage networks is more even than in many cluster environments
  - Difficult, if not impossible, to optimize data placement within the storage system
  - Data is spread to most disks
  - Relatively few connections to external clients
- Distribution of load on links is important
- Placement of links to the outside world is important

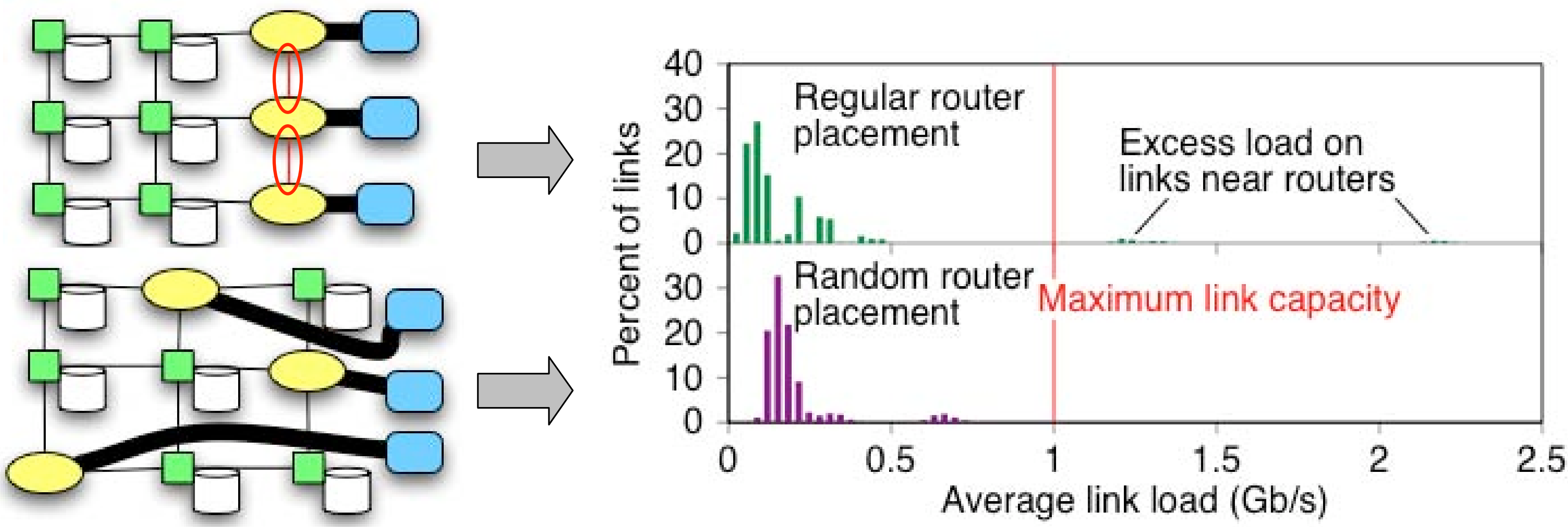
Load distribution on links in a 6-D torus

- Is a 6-D torus a good choice?
- Place routers along the edge
  - Average path length is OK, but…
  - Congestion near the routers
  - Links at the edge overutilized
- Place routers at random locations
  - Better load distribution
  - Few links overutilized
- Placement of connections to storage system clients is important!



Load distribution issues: details

- Maximize the distance between routers to the outside world
  - Ensure the distance has a low variance
- Not following this guideline will dramatically slow storage system performance



Failure resilience

- Many components can fail
  - Disk (often dealt with in the file system)
  - Network switch
  - Link
- Storage system must continue to supply data
- Network should have alternate routes between disks and clients
  - Meshes & torii have redundancy built in
  - Butterfly networks don't have this redundancy
    - Add more links to provide resilience?

Conclusions & future work

- Interconnection network design is crucial for high-performance petabyte storage systems
- Medium-dimension torii are probably the best choice
  - Not too high cost
  - More resilience to network failures
- Placement of connections to clients within the network is critical
  - Poor placement can lead to degraded performance

- Future work
  - Explore the effects of network and disk failures on interconnection network performance (load)
  - Drive the simulation models with real workloads