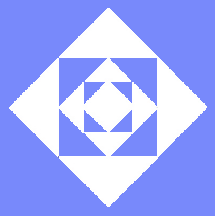


Resource allocation and utilization in the Blue Gene/L supercomputer

Tamar Domany, Y. Aridor, O. Goldshmidt, Y. Kliteynik, E.Shmueli, U. Silbershtein

Agenda

- Blue Gene/L Background
- Blue Gene/L Topology
- Resource Allocation
- Simulation Results

Blue Gene/L - Overview

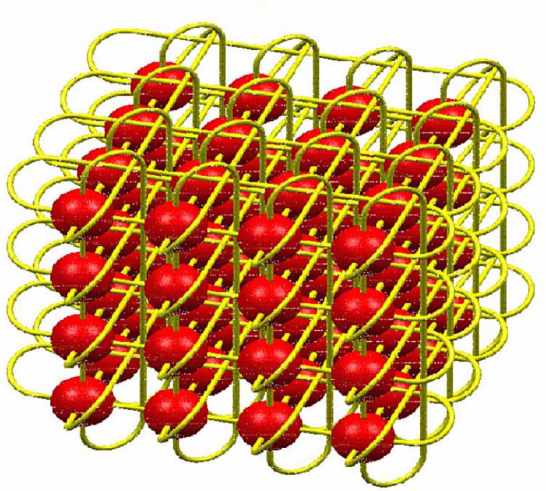
- First member of IBM Blue Gene family of supercomputers
- Machine configurations range from 1000 to 64,000 nodes
- The world fastest supercomputer
  - Rated first in the last top500 list (November 2004)
  - Machine size of 16K nodes
- Selected customers:
  - Lawrence Livermore National Laboratory
  - Japan's National Institute of Advanced Industrial Science and Technology
  - Lofar radio telescope run by Astron in the Netherlands
  - Argonne National Laboratory

Blue Gene/L Philosophy

- Designed for highly parallel applications
- Traditional Linux and MPI programming models
- Extendable and manageable
  - simple to build and operate
- Vastly improved price/performance
  - choosing simple low power building block
  - highest possible single threaded performance is not relevant, aggregate is!
- Floor space and power efficiency
- BlueGene/L = Cellular architecture + aggressive packaging + scalable software

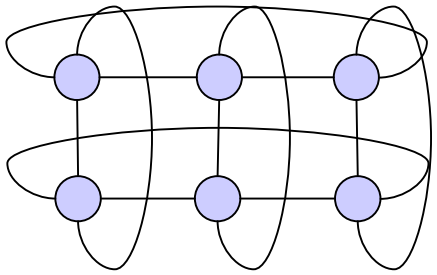
BlueGene/L cellular architecture

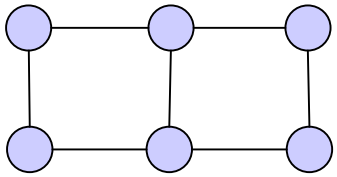
- The design of BlueGene/L is substantially different from the traditional supercomputers (NEC Earth Simulator, ASCI machines) that uses large clusters of SMP nodes
- Very large number (64K) of simple identical nodes
  - Low cost, low power, PPC microprocessors (700Mhz)
- Geometry: 64x32x32, based on 3D torus
  - Low latency, high bandwidth propriety interconnect
  - I/O physically separated from computations
  - At most one process per CPU at a time
- Scalable and extendable architecture
  - Computational power of the machine can be expanded by adding more “building blocks”



Jobs in Blue Gene/L

- Blue Gene/L runs parallel jobs
  - Set of task running together, communicating via message-passing
  - Each job has a set of attributes
    - Size – # of threads (and thus nodes)
    - 3D Shape
      - size 8 can be “slim” (e.g. 8x1x1) or “fat” (2x2x2)
    - Communication pattern – torus or mesh

torus 

mesh 

What is a Job Partition ?

- A partition is
  - A set of nodes
  - A set of communication links
    - Which connect the nodes as a torus or a mesh
- Partitions are isolated
  - A single partition accommodates a single job
  - No sharing of nodes or links between partitions

Job Management for Blue Gene/L

- Users submit jobs to the Blue Gene/L scheduler
  - The scheduler maintains a queue of submitted jobs
- The scheduler's task:
  - Choose the next job to run from the queue
  - Allocate resources for the job
  - Launch the job
  - Monitor the job until termination
  - Signals, debugging…

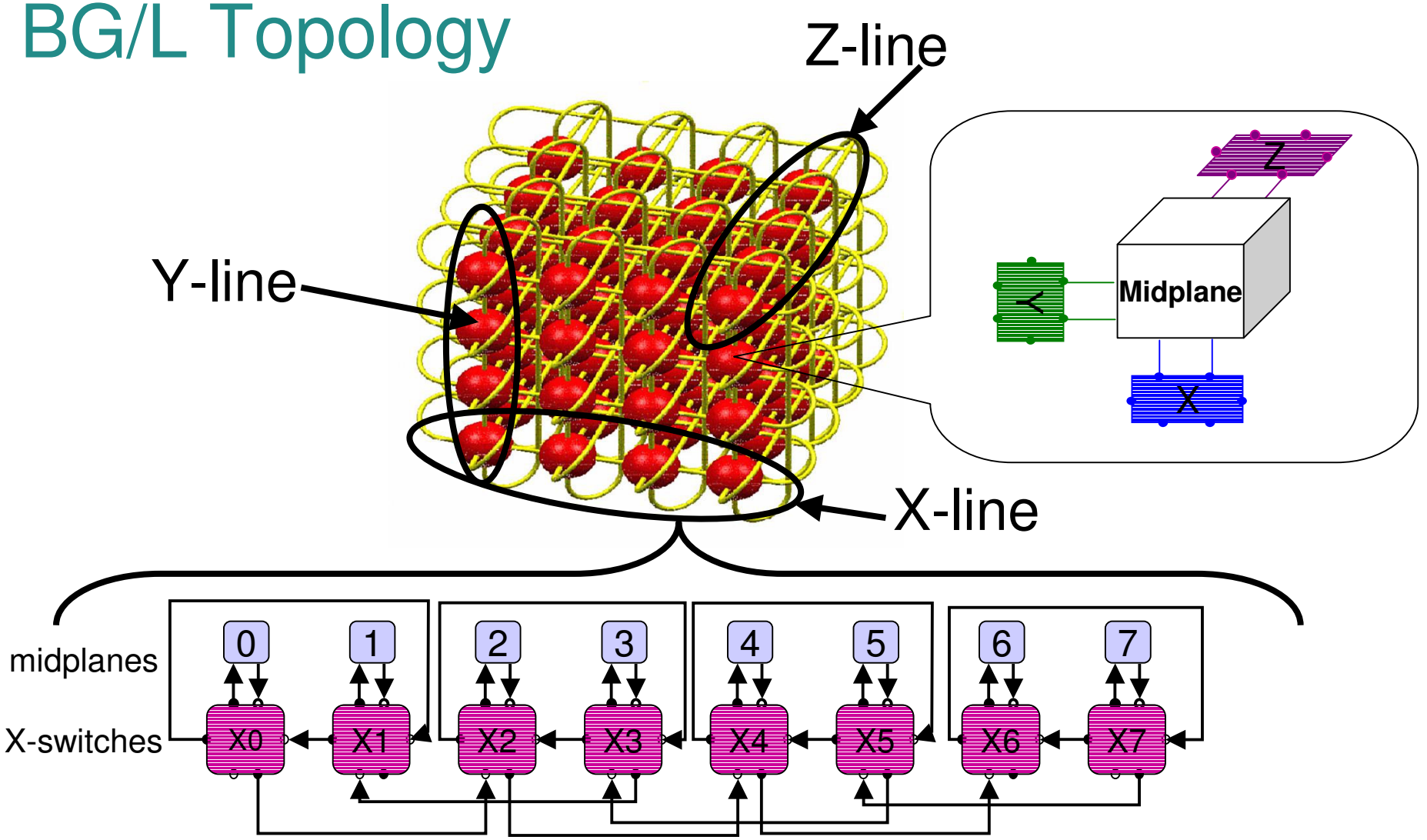
Job Management Challenges

- How do we scale beyond a few thousands nodes ?
  - Group nodes into midplanes
- How do we maximize machine utilization ?
  - Extend toroidal topology to multi-toroidal topology

Scalability via Midplanes

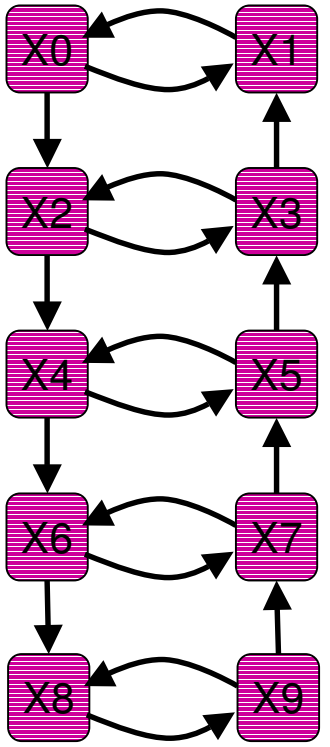
- Nodes are grouped into 512-node units called *midplanes*
  - A midplane is an 8x8x8 3D mesh
  - Each internal node is connected directly to at most six internal neighbors
  - Midplanes are connected to each other through switches
- Scalability achieved by sacrificing granularity of management
  - Midplane is the minimal allocation unit
    - Not all nodes may be utilized for a given job
  - In practice, we deal with a 128-node machine instead of 64K nodes
    - For all aspects of job management

BG/L Topology



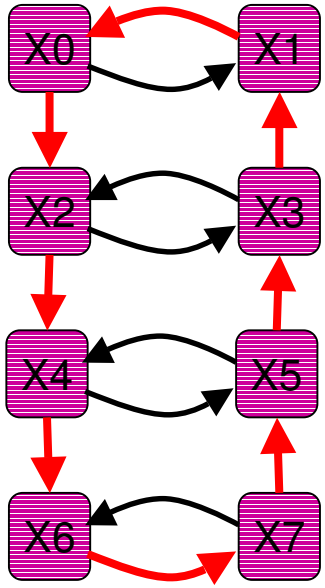
Line connectivity - properties

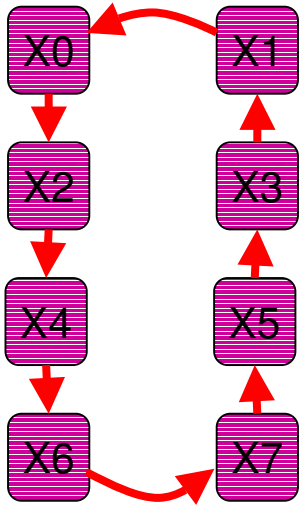
- Lines have “multi-toroidal topology”
  - Can be easily extended



Line connectivity - properties

- Lines have “multi-toroidal topology”
  - Can be easily extended
  - Can be connected as a torus

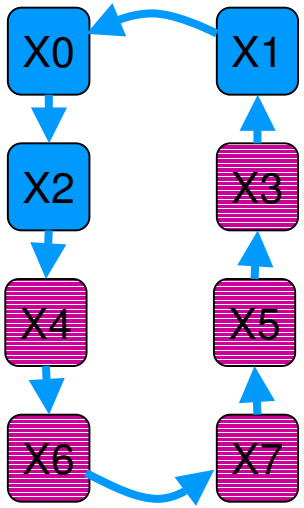




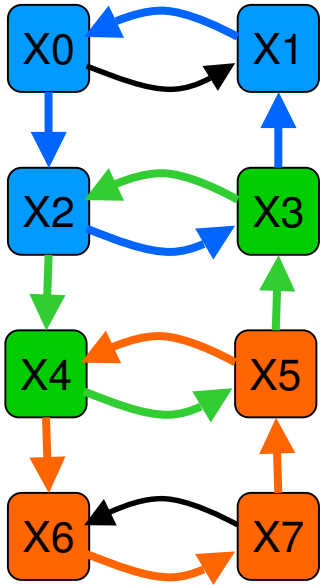
3D torus

Line connectivity - properties

- Lines have “multi-toroidal topology”
  - Can be easily extended
  - Can be connected as one torus
  - Multiple toroidal partitions can co-exist

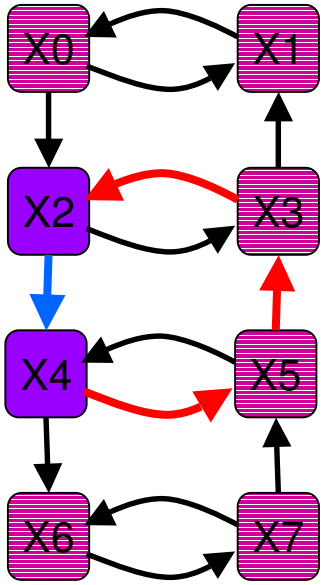


3D torus



Line connectivity - properties

- Lines have “multi-toroidal topology”
  - Can be easily extended
  - Can be connected as a torus
  - Multiple toroidal partitions can co-exist
  - More than one way to wire a set of midplanes



Resource Allocation

- Challenges
  - High machine utilization
  - Short response time (of jobs)
  - On-line problem
- Requirements
  - Satisfy job requests for size, shape, and connectivity (torus or mesh)
  - Deal with faulty resources (nodes and wires)
- Two kinds of dedicated resources to manage
  - Node allocation
  - Link allocation

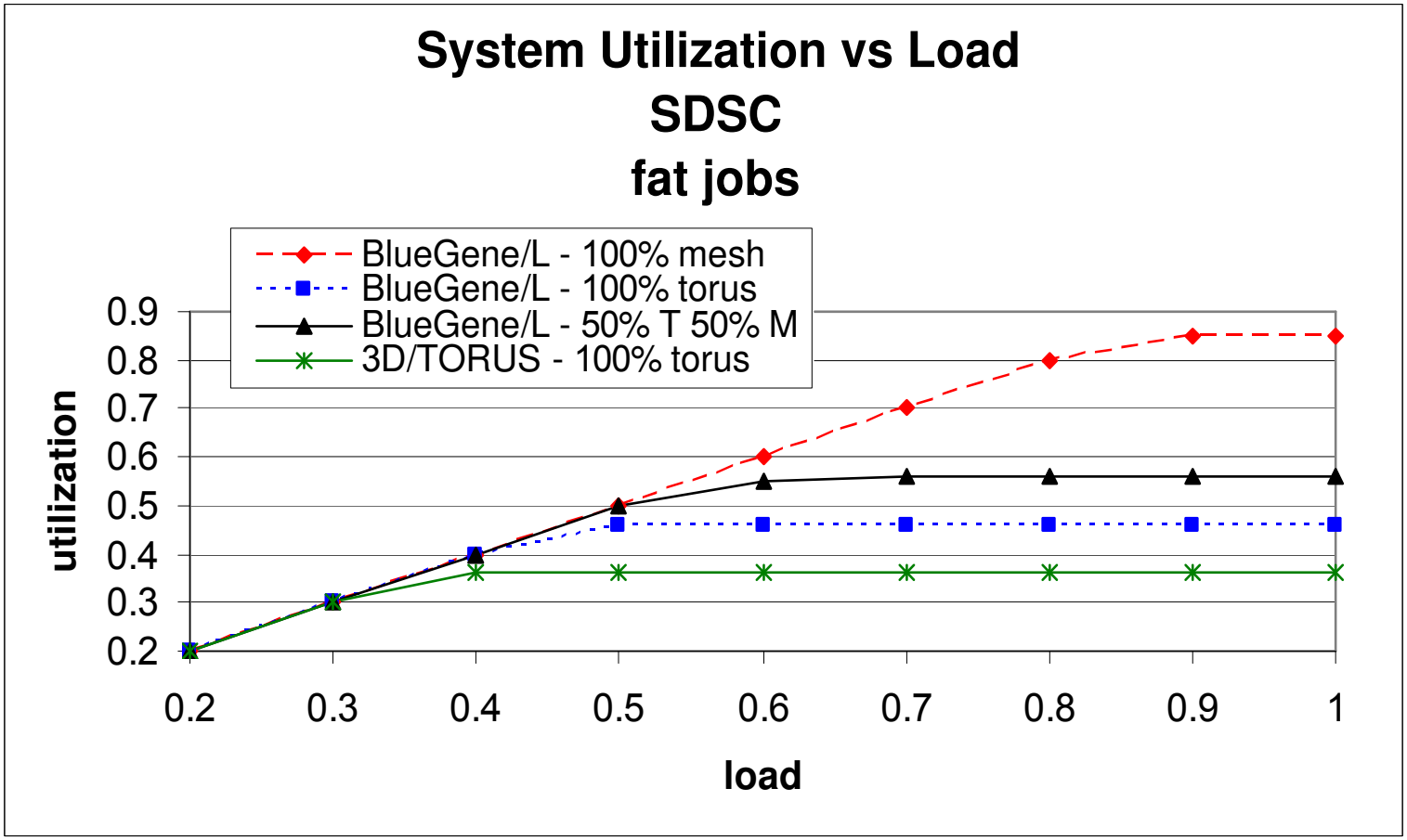
Allocation Algorithm

- Finding a partition: scan the 3D machine
  - Find all free partitions that match the shape/size of a job
  - For each candidate partition, find if and how it can be wired
  - From all wireable partitions, choose the “best” partition
    - use flexible criteria e.g. minimal number of links
- Wiring a partition
  - Static wire lookup tables per dimension
    - Availability of wires (previous allocation or faults) is checked
  - Find suitable links in (almost) constant time
  - Small memory footprint despite the huge number of links

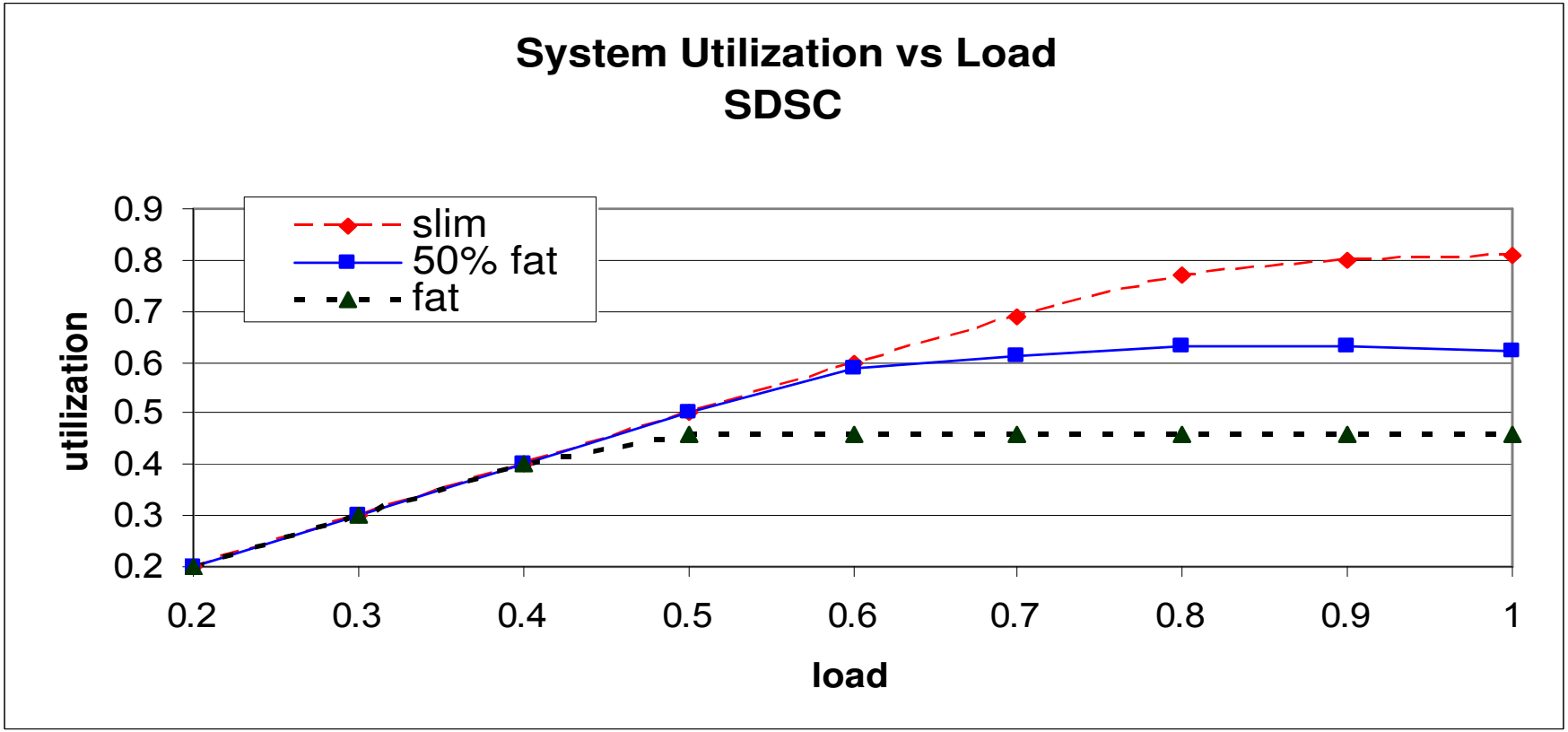
Simulated Environment

- Faithful simulation of Blue Gene/L
  - 128 midplanes
- Scheduler invoked when a job arrives or terminates
- Scheduling policy
  - Aggressive backfilling
    - If the job at the head of the queue cannot be accommodated we try to allocate another job out of order
- Workloads (benchmarks)
  - Arrival times, runtimes, size, shape, torus/mesh
  - Based on real parallel systems' logs
    - This presentation: San Diego Supercomputer Center (SDSC)

The benefits of multi-toroidal topology



The influence of job shapes on utilization



Summary

- Blue Gene/L brings with it a new level of supercomputer scalability – and many new challenges
- Scalability of system management is achieved by sacrificing granularity
  - Represent the machine as a smaller system consisting of collections of nodes
- Blue Gene/L's novel network topology has considerable advantages compared to traditional interconnects (such as 3D tori)
- The challenges are successfully met with a combined hardware and software solution

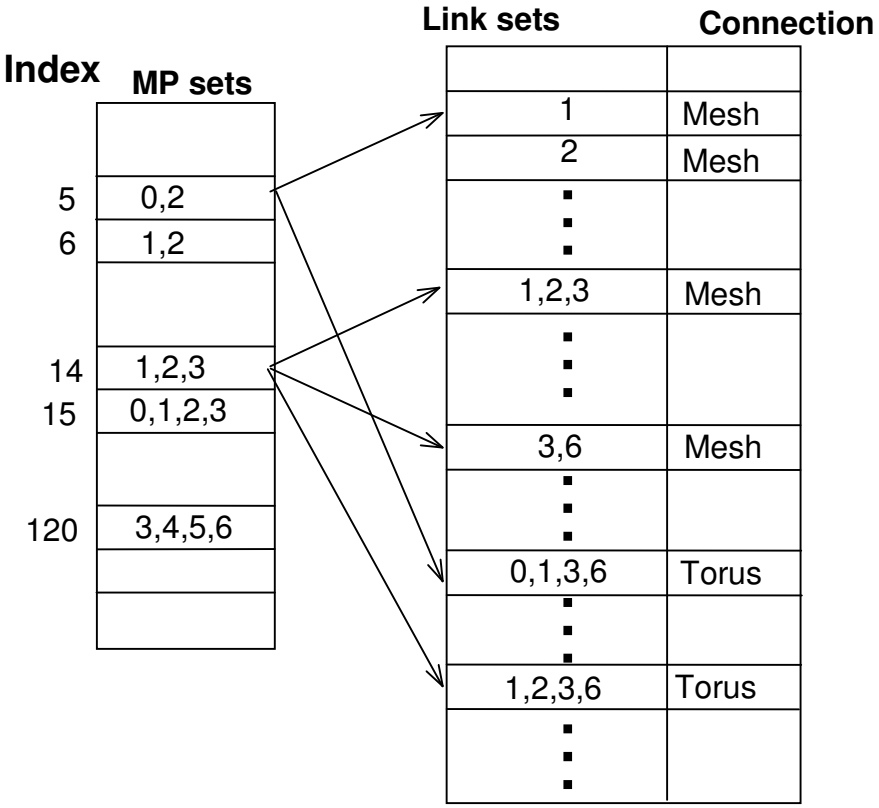
End

Link Allocation

- The problem:
  - Given a partition, fined links in all the lines that participate in the partition for all three dimensions to wire a partition attempting to best utilize future allocations.
- Solution main idea:
  - Build a lookup table with the partitions wiring possibilities
  - The dimension are independent → Table per dimension
  - All lines in a dimension are equal → Table contain information on one line
  - There are not so many whys to wire a partition → consume relatively small amount of memory

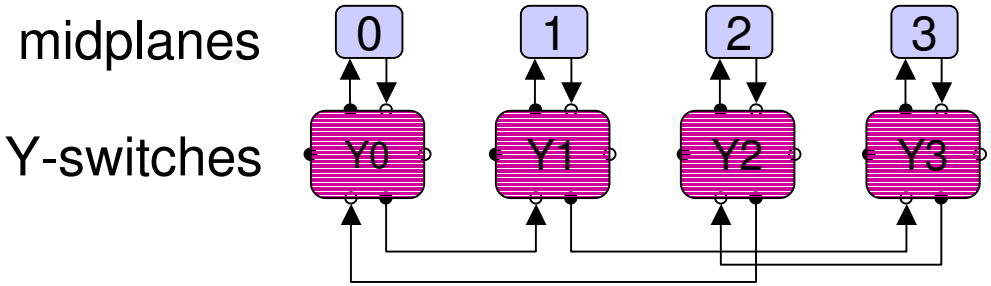
The Lookup table

- A table per topology dimension
  - The index is a possible set of midplanes
  - Each entry contains all sets of links that can wire it as a torus or as a mesh
- Built once at startup time
- Given a partition, use tables to find link set in each dimension
- Eliminate non-available sets, output “best” among available

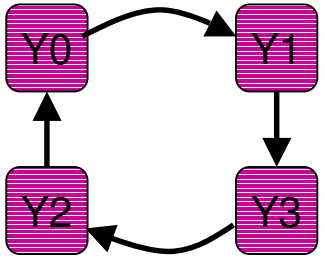


Y & Z lines Connectivity

- No “multi-toroidal topology”



- Or can be drown that way (without the midplanes):



  - Can accommodate only one torus partition at a time