Toward Runtime Power Management of Exascale Networks by On/Off Control of Links

Ehsan Totoni, Nikhil Jain, Laxmikant Kale
University of Illinois at Urbana-Champaign
HPPAC May 20, 2013

Power challenge

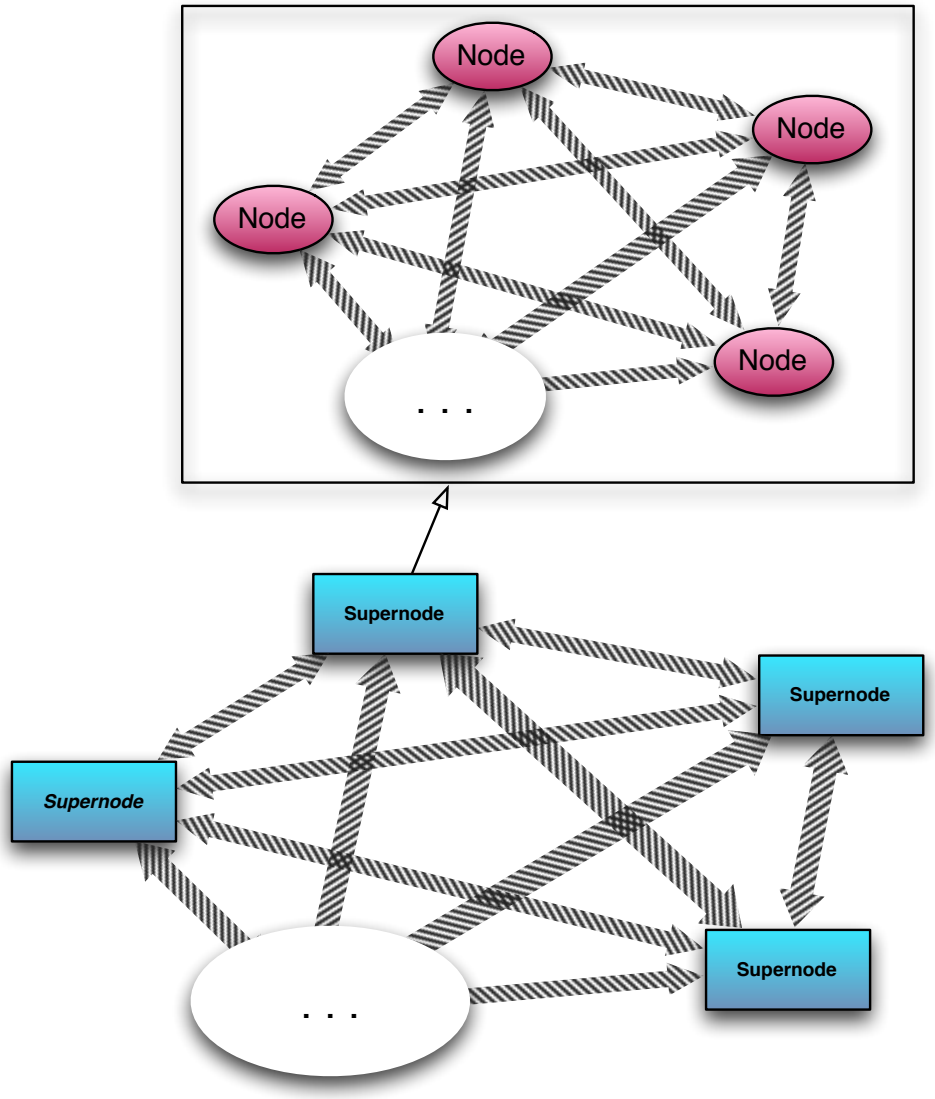
- Power is a major challenge
- Blue Waters consuming up to 13 MW
  - Enough to electrify a small town
  - Power and cooling infrastructure
- Up to 30% of power in network
  - Projected in Exascale report
  - Saving 25% power in current Cray XT system by turning down network
    - Work from Sandia

Network link power

- Network is not "energy proportional"
  - Consumption is not related to utilization
  - Near peak most of the time
  - Unlike processor
- Recent study:
  - Work from Google in ISCA'10
  - 50% of power in network of data center
  - When CPU is underutilized
- Up to 65% of network's power is in links

Exascale networks

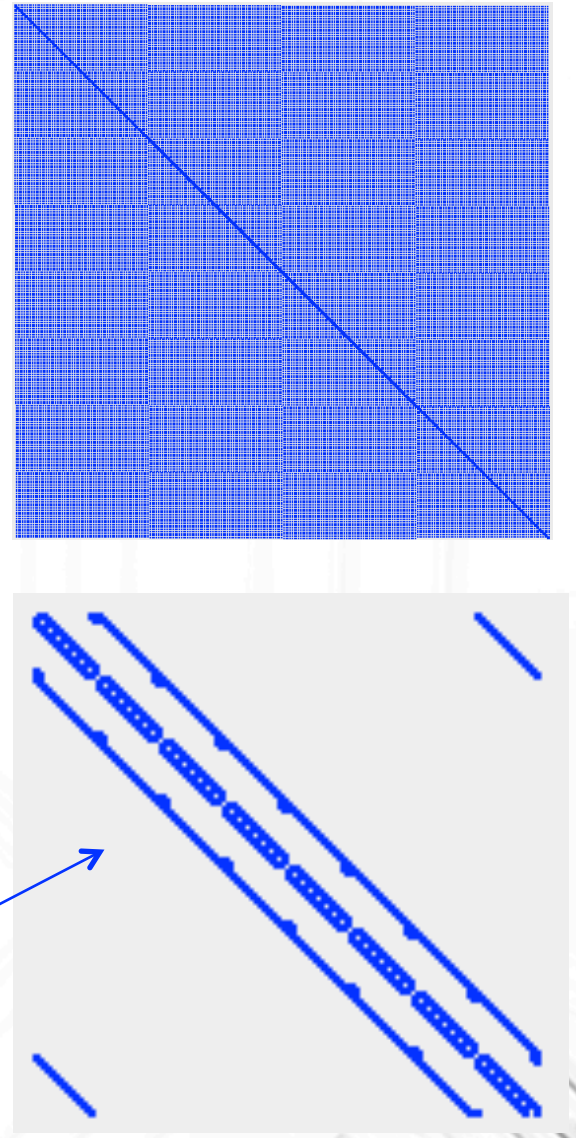
- Dragonfly
  - IBM PERCS in Power 775 machines
  - Cray Aries network in XC30 “Cascade”
  - DOE Exascale Report
  - Multilevel directly connected
    - “All-to-all” links in each level
- High dimensional Tori
  - 5D Torus in IBM Blue Gen/Q
  - 6D Torus in K Computer
- Higher radix -> a lot of links!
  - Essential for performance



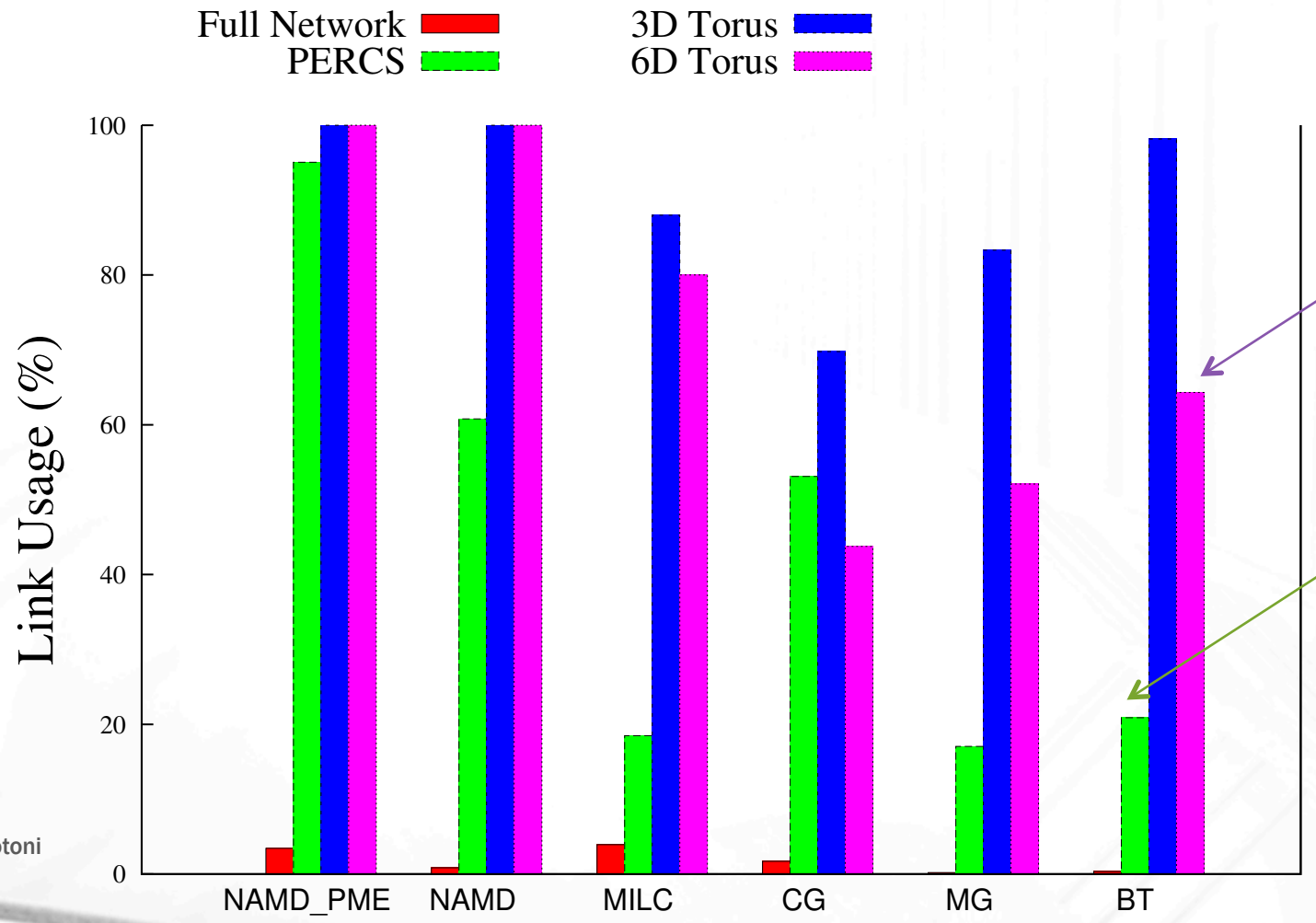
Communication patterns

- Applications' communication patterns are different
  - Only some node pairs communicate
  - Nearest neighbor most common
  - Sometimes global

- Network topology designed for a wide range of applications
  - Including worst cases
    - e.g. All-to-all in FFT



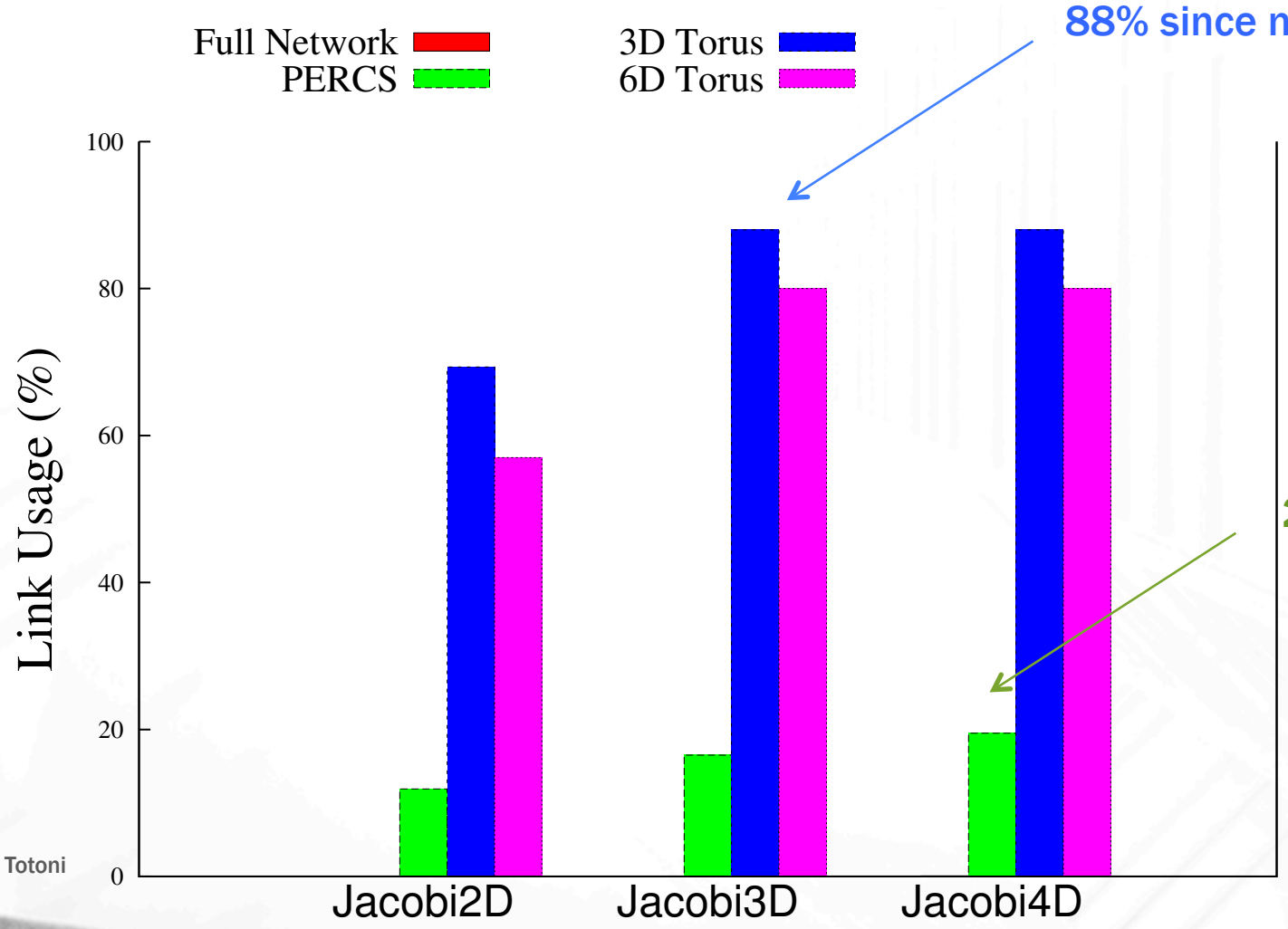
Fraction of links ever used



60% of 6D torus links

21% of Dragonfly links

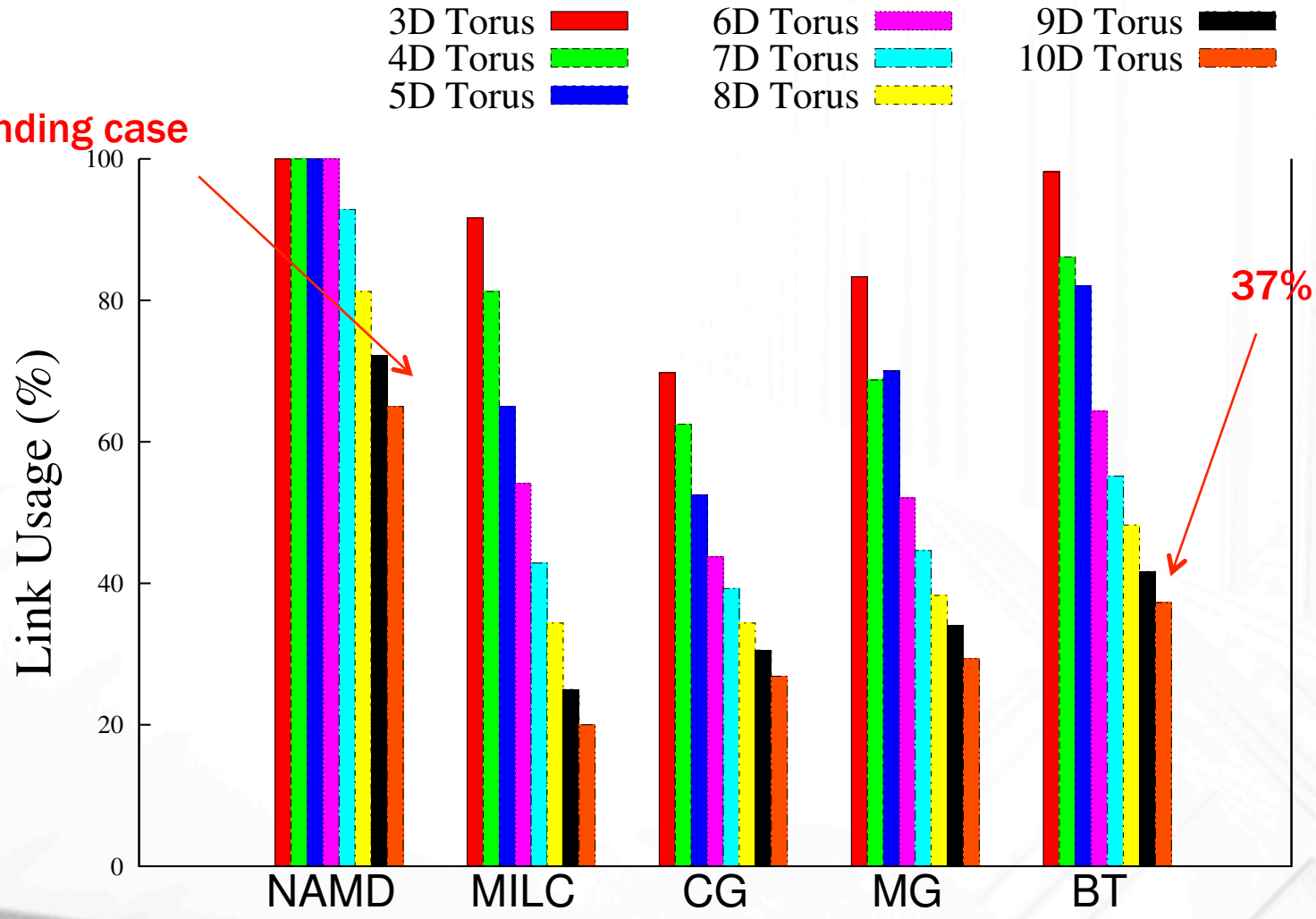
Nearest neighbor usage



88% since not mapped perfectly

20% of Dragonfly links used

Increasing torus dimensions



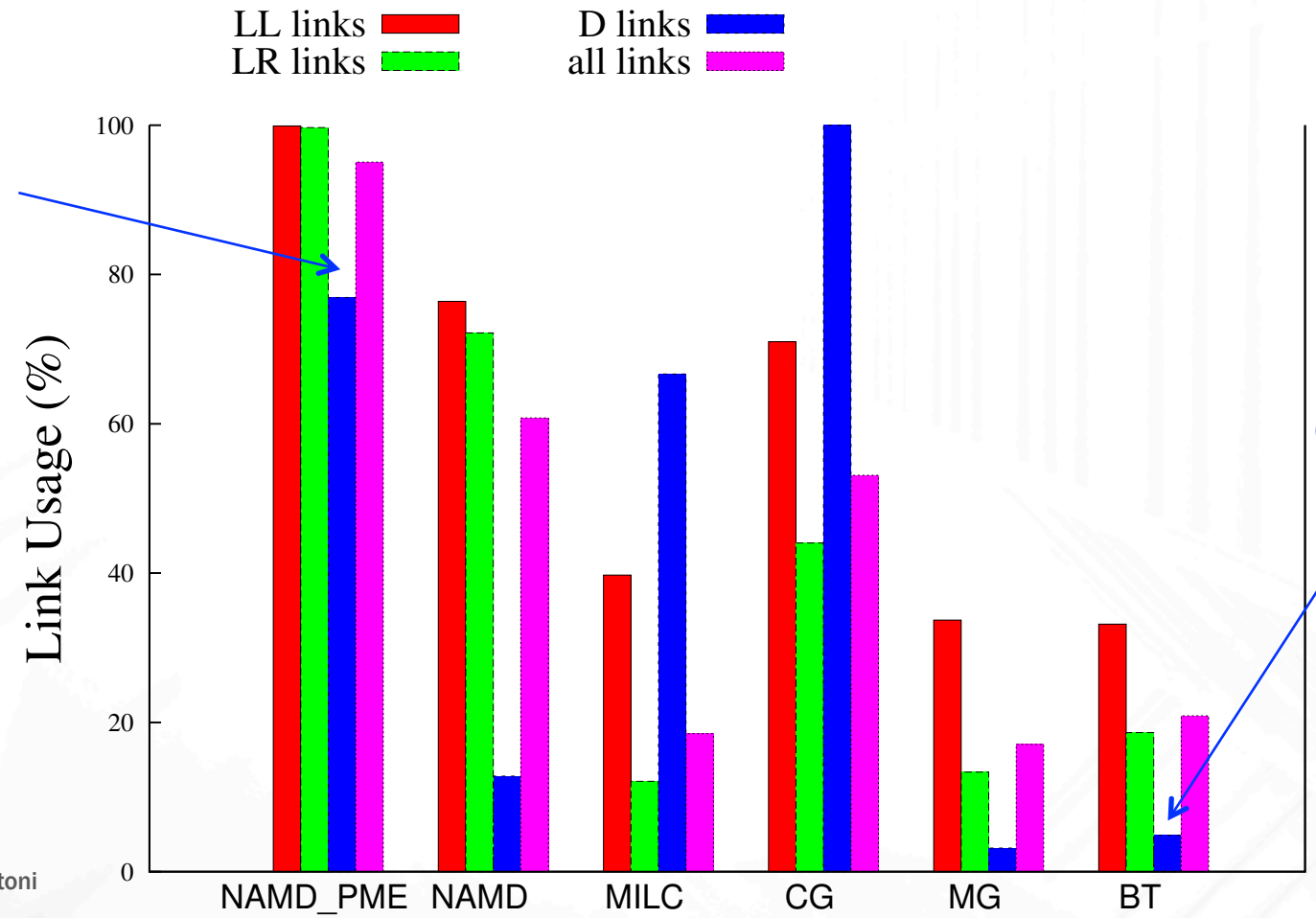
65% for demanding case

37% of 10D torus links used

Cost of different links

- Links have different costs
  - Different underlying technology
  - Electrical vs. Optical
- Conversion cost
- Dragonfly/PERCS
  - LL-links: “local” within drawer
  - LR-links: “remote” across drawers
  - D-links: between “supernodes”
- More expensive links are used less often
  - Locality in applications

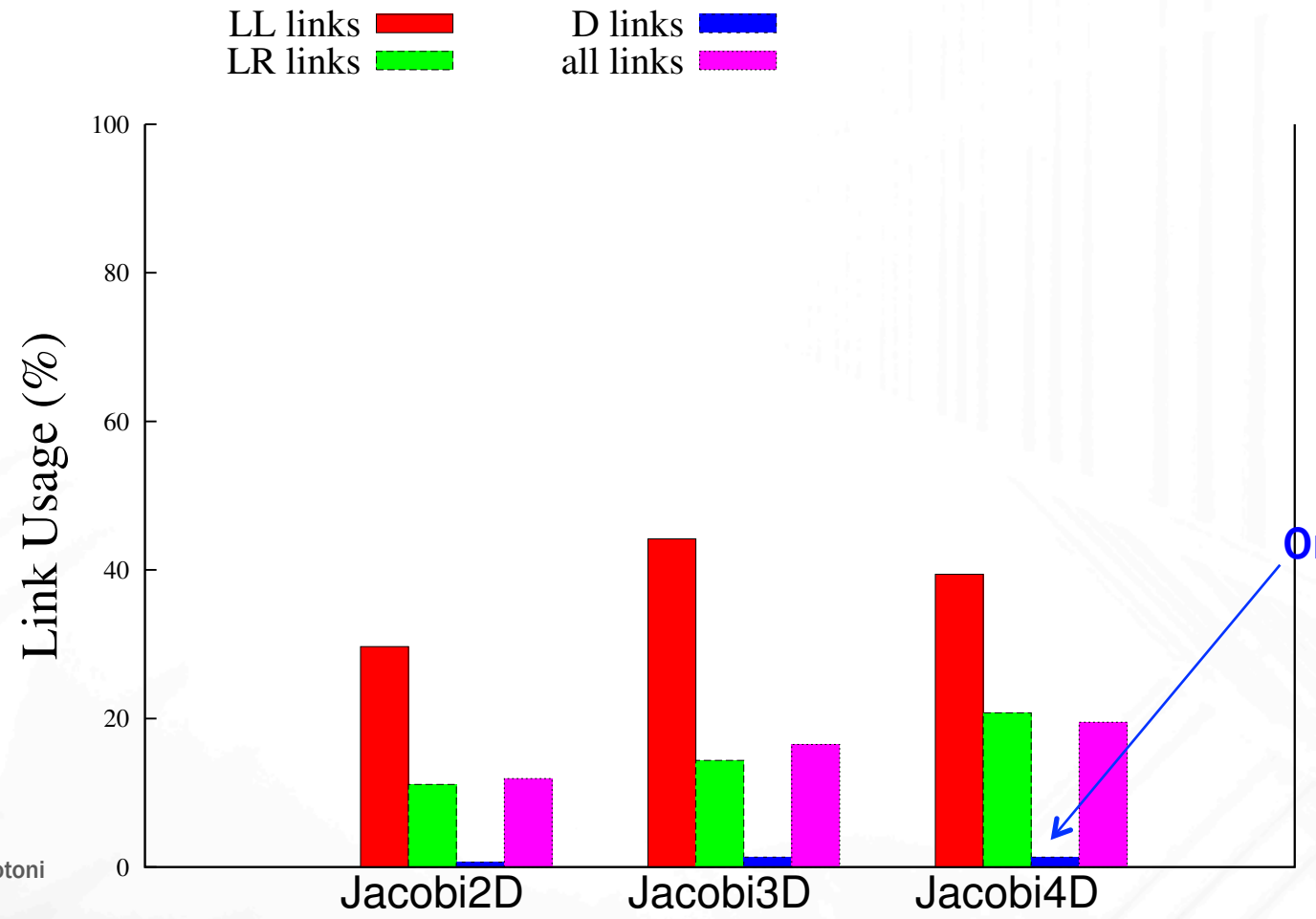
More expensive links



77%

Only 5% of global “D-links” used!

Nearest neighbor



Only 1% of global “D-links” used!

Solution to power waste

- Many of the links are never used
  - For common applications
- Are networks “over-built”? Maybe
  - FFTs are crucial
    - Can’t have weaker networks
  - processors are also overbuilt
- Let’s make them “energy proportional”
  - Consume according to workload
  - Just like processors
- Turn off unused links
  - Commercial network exists (Motorola)

Hardware implementation feasibility

- Can change network configuration in some HPC machines
  - E.g. Cray XT
  - Needs reboot- impractical now
- On/off links exist in some commercial machines
  - Motorola
    - Can turn off some of board-to-board links
    - In 10us
- Feasible to add to HPC networks

Runtime system solution

- Hardware can cause delays
  - According to related work
  - Not enough application knowledge
    - Small window size
- Compiler does not have enough info
  - Input dependent program flow
  - SPMD: “if (rank==0) …”
- Application does not know hardware
  - Significant programming burden to expose
  - Hurts portability

Runtime system solution

- Runtime system is the best
  - E.g. MPI, Charm++
  - mediates all communication
    - knows the application
    - knows the hardware
  - This info is used for other purposes
    - Communication optimization
    - Load balancing
    - Topology mapping
    - Power management

Algorithm

- At each node:
  - Collect the list of destinations of messages
  - For each destination
    - Mark local links
    - Ask intermediate nodes to mark their links
  - Turn off unmarked links

Invocation

- Most applications are iterative
  - And “static”
    - Doing the same thing over and over
  - Profiling few iterations is enough for common applications
    - Communication pattern is constant
  - Measure performance to avoid degradation
    - Turn links back on if performance affected

Dynamic applications

- NAMD:
  - Communication changes in load balancing steps
  - Switch links accordingly
  - Load balancing is done in runtime system
- Self correction:
  - Measure performance
  - Turn links back on if harmful

Feasibility

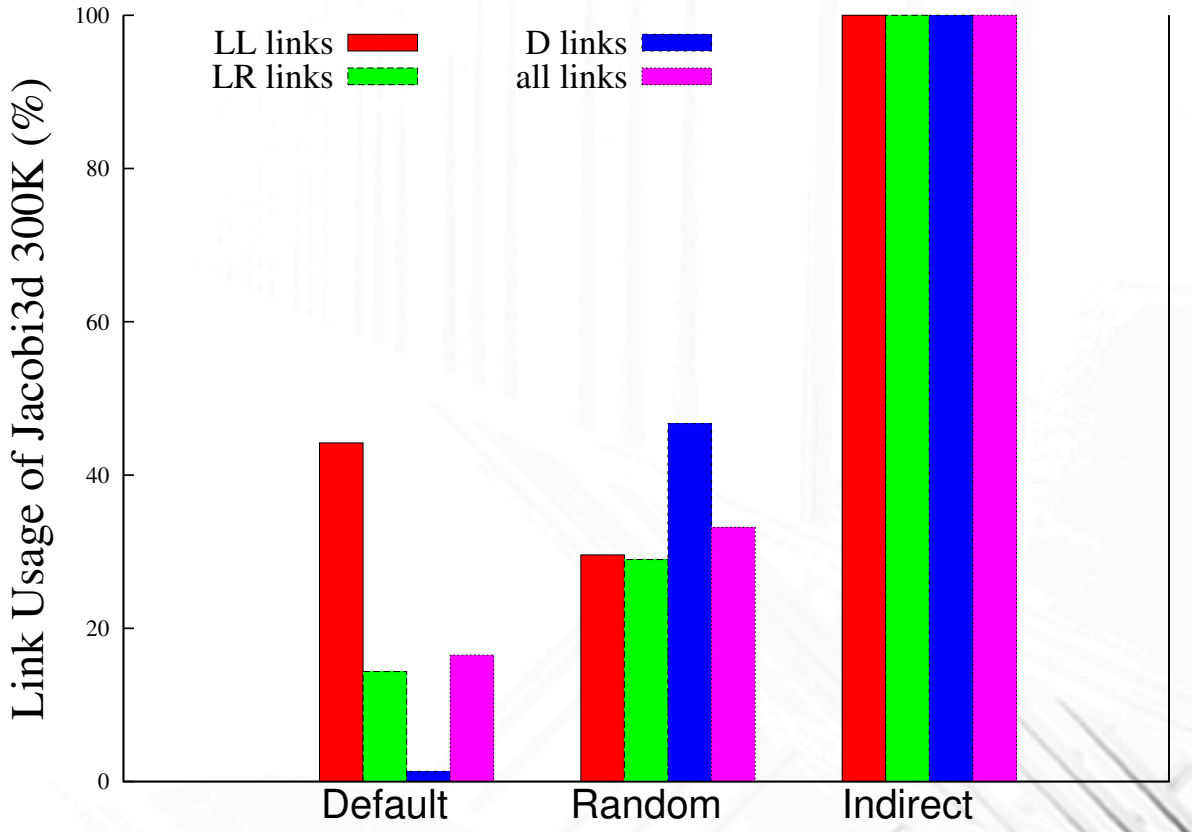
- Not probably available for your cluster
  - Need to convince hardware vendors
  - Runtime hints to hardware, small delay penalty if wrong
- Multiple jobs: interference
  - Isolated allocations are becoming common
    - For performance!
    - Blue Genes allocate cubes already
  - Capability machines are for big jobs
  - I/O inside partition

Design choices

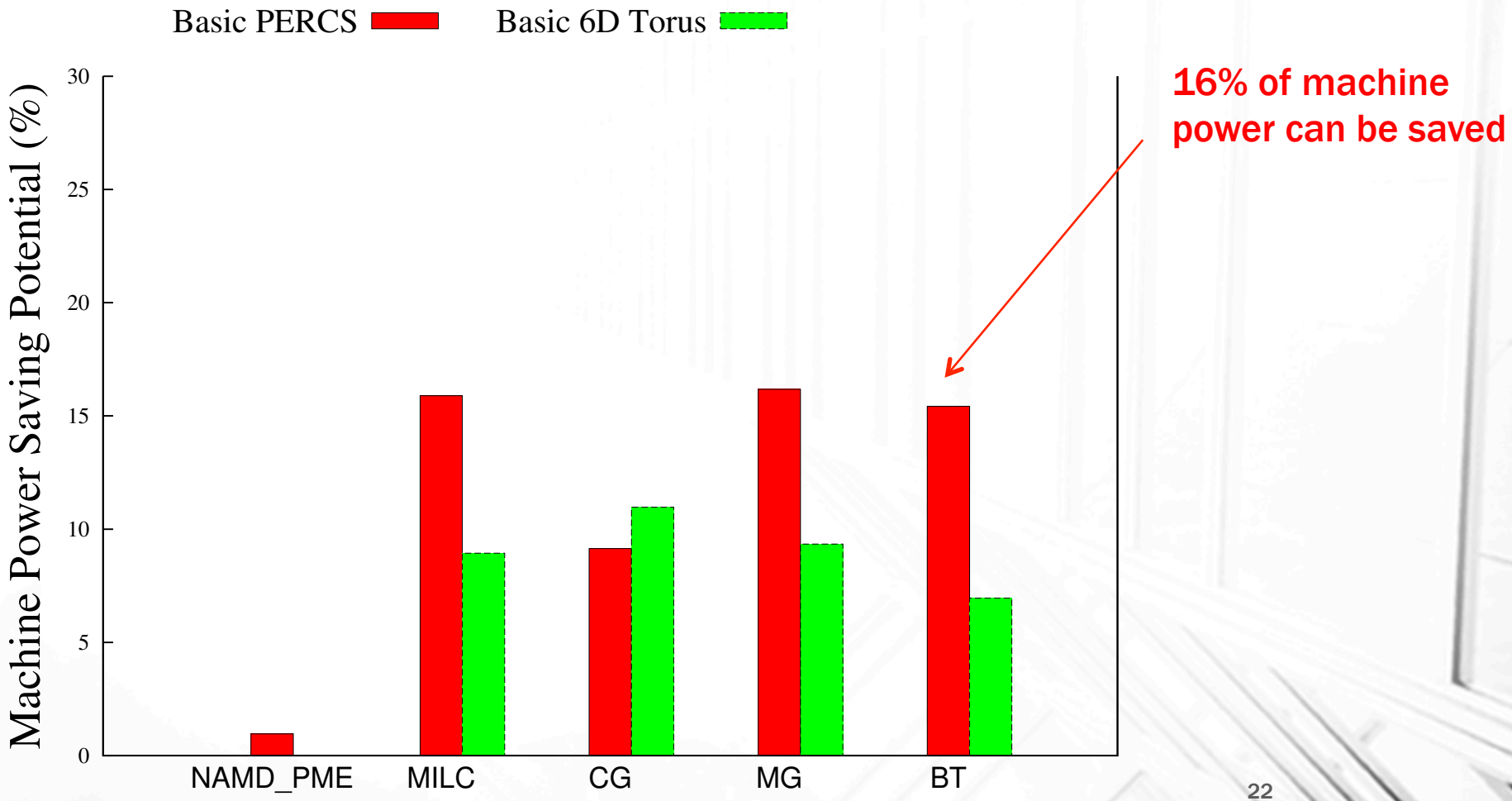
- Direct routing: send directly to destination
  - Congestion in Dragonfly with simple mapping
    - Because of “locality” in communication
  - Doesn’t use many links
- Alternatives: Indirect routing and Random mapping
  - Bhatele et al. at SC11
- Indirect routing: send to random intermediate supernode
  - Uses more links
- Random mapping: eliminate locality

Design choices

- Random mapping vs. indirect routing
  - similar performance
  - different link usages



Results summary



16% of machine power can be saved

Questions?