Optimized Virtual Channel Assignment in the Cray XT

Dennis Abts, Deborah Weisser, Jim Nowicki, and Robert Alverson

{dabts, dweisser, nowicki, bob}@cray.com

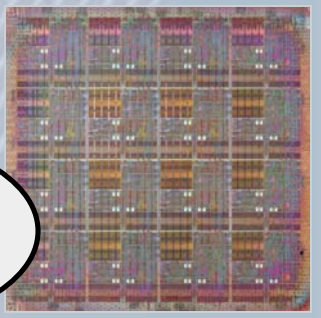
CRAY
THE SUPERCOMPUTER COMPANY

Outline

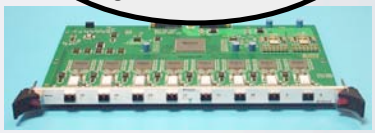
- Overview of the Cray XT network
  - Topology, Routing and Flow control
- Microarchitecture of the Cray SeaStar router
  - Switch allocation and virtual channels
- Deadlock avoidance
- Optimizing virtual channel buffer assignment
- Results
- Conclusions

Interconnection Networks

Supercomputers

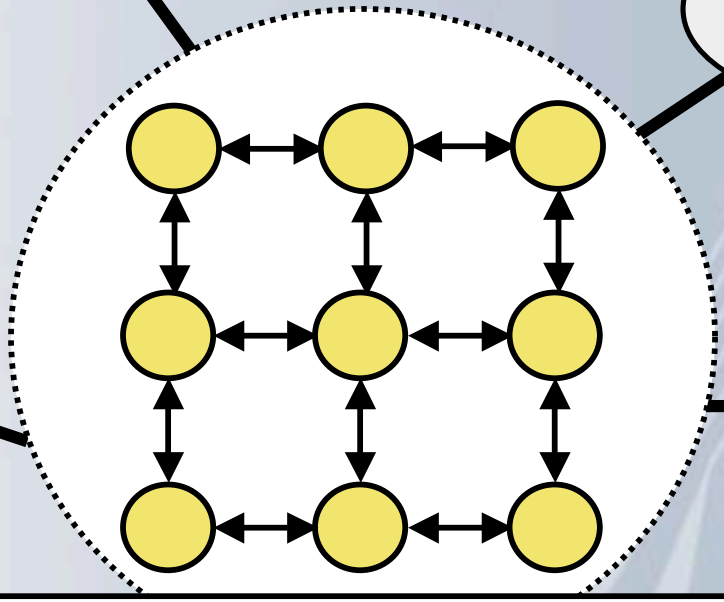


On-chip



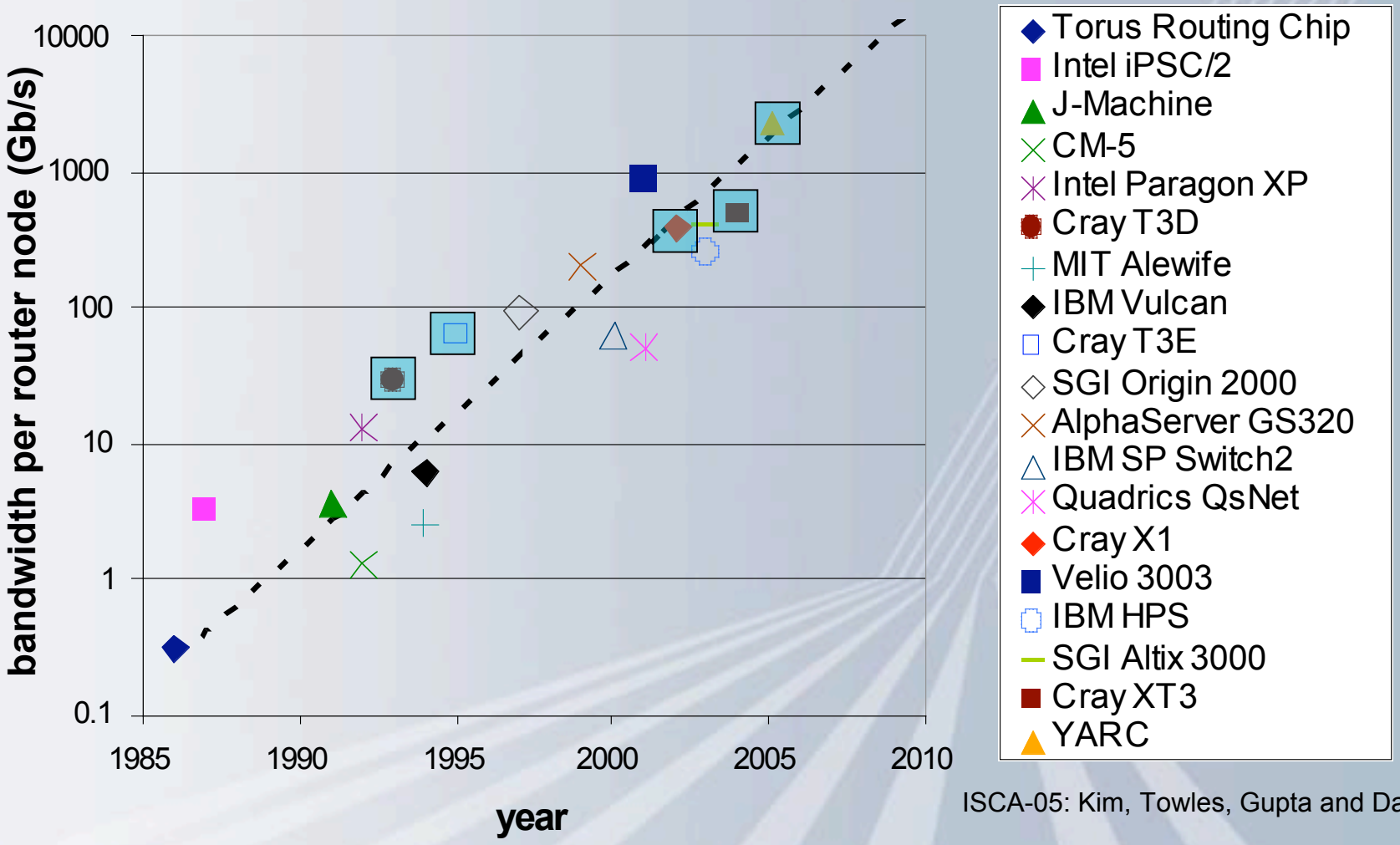
I/O systems

Router fabrics



Interconnection networks

Technology Trends for Router Bandwidth

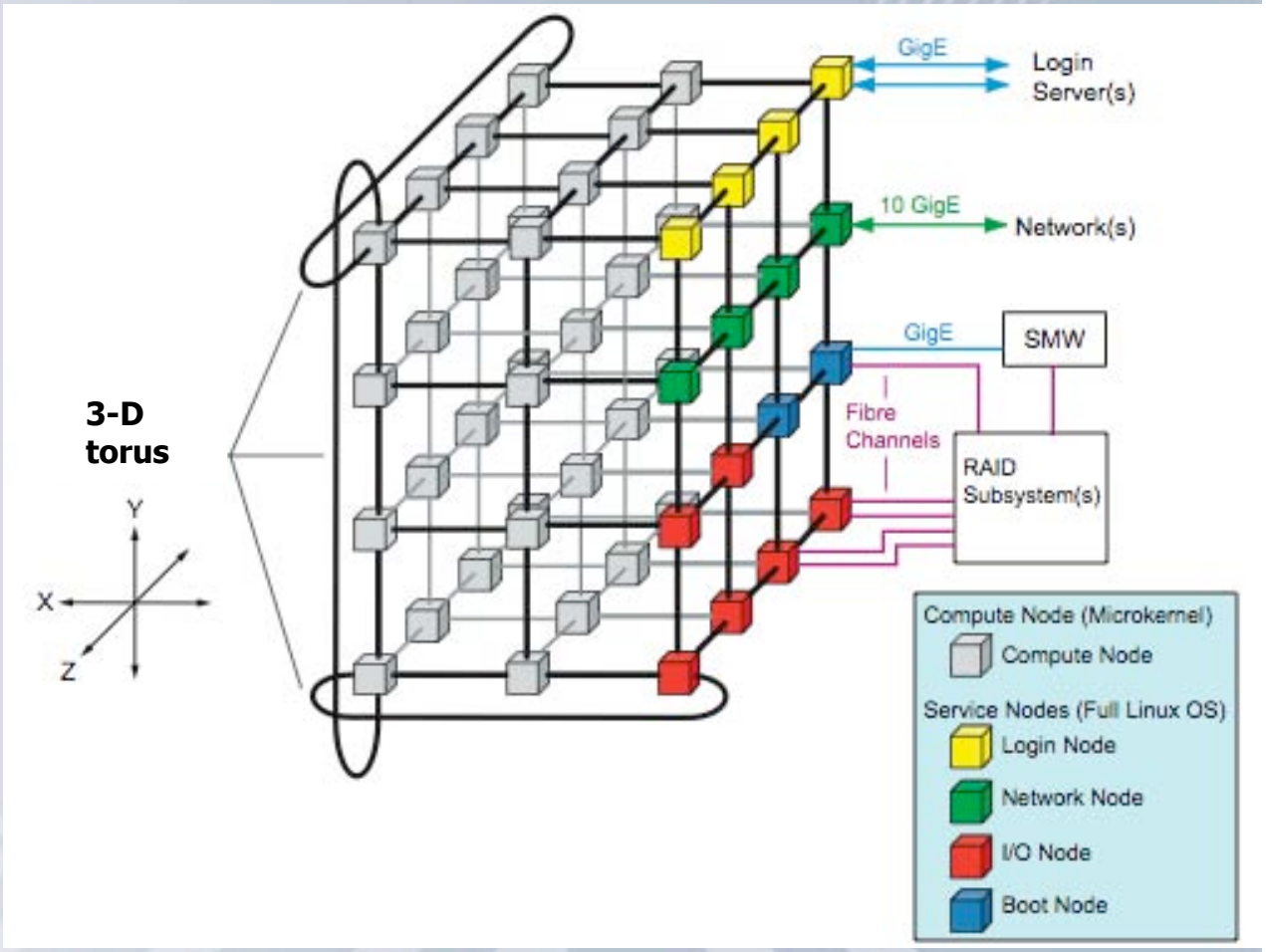


ISCA-05: Kim, Towles, Gupta and Dally

Cray XT Network Overview

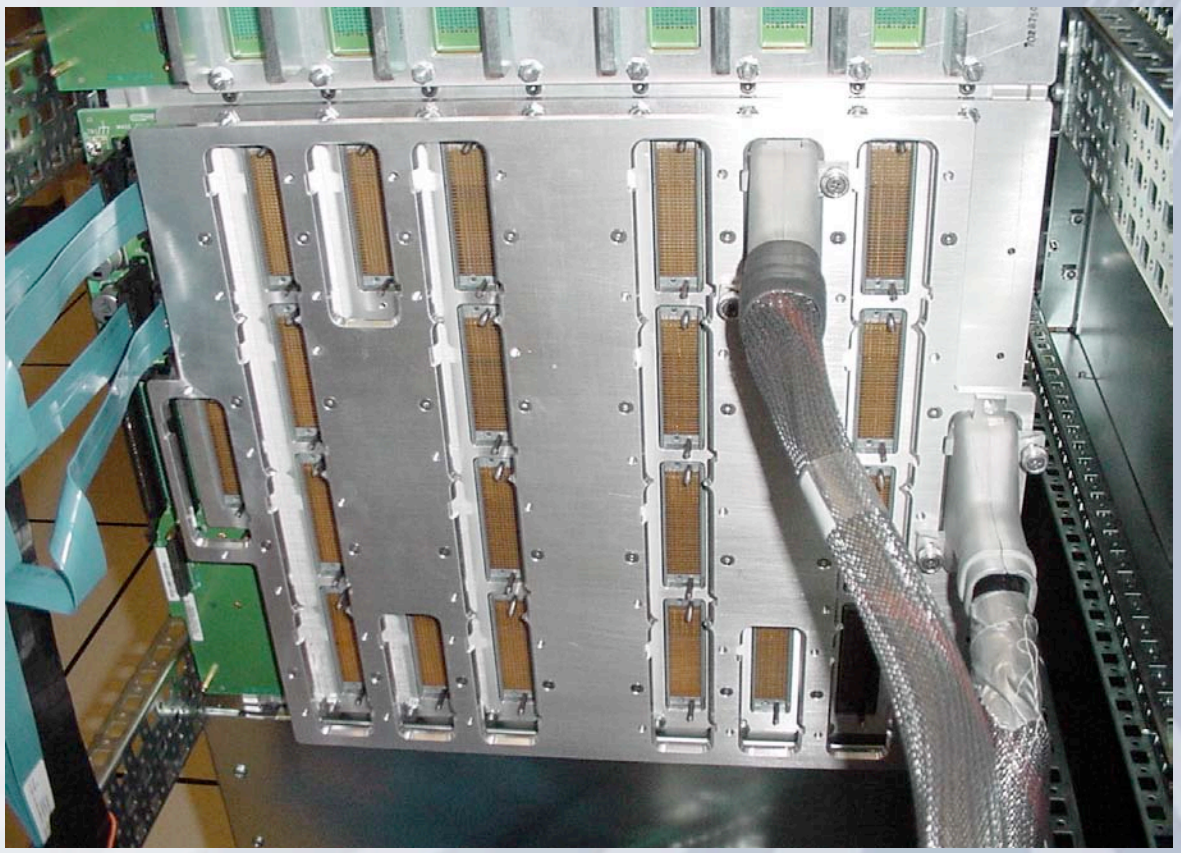
Architecturally scales up to 32K nodes

7-ported router (Seastar) 1 for the processor injection/ejection and 6 directions



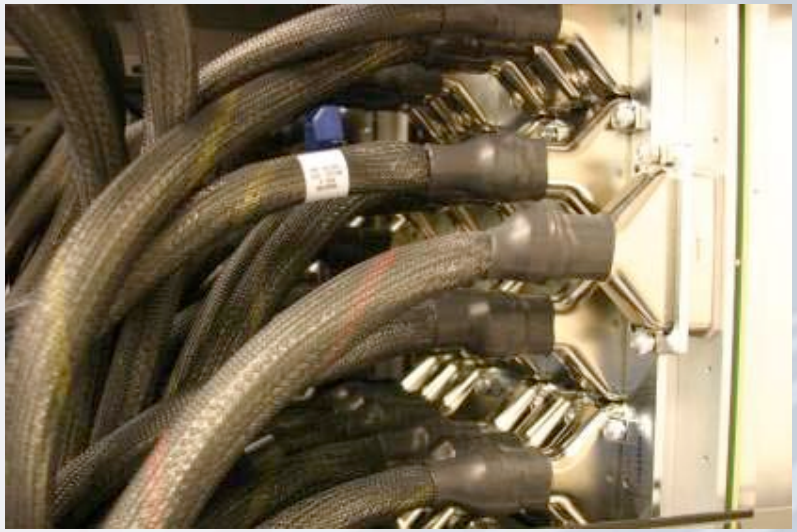
Cray XT3 interconnect

- Network links are *aggregated* 4 links per cable
  - Reduce cable bulk
  - Reduce cost



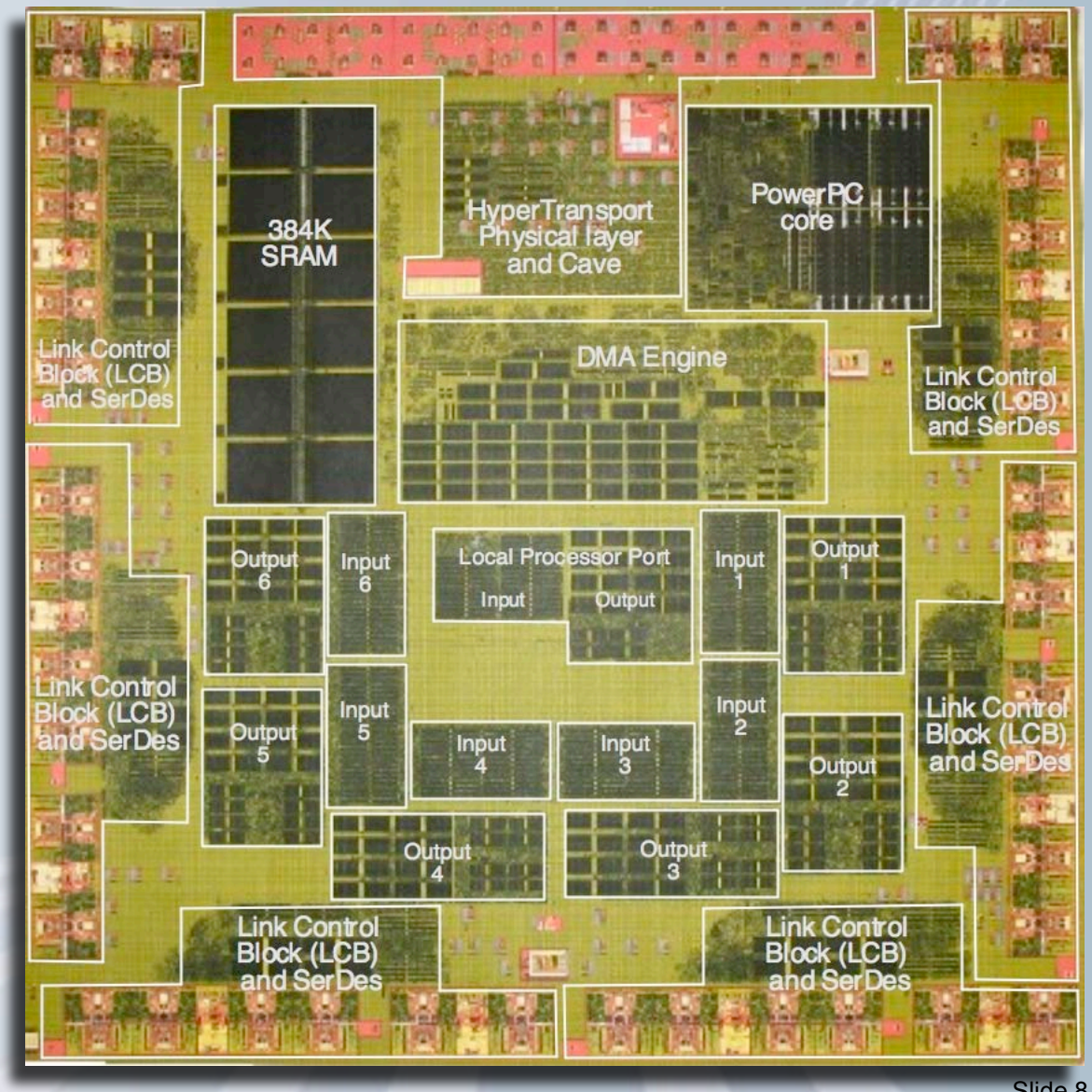
SeaStar Cables

- Bottom to top cooling allows for a dense cable mat
- Each SeaStar cable carries 38 GB/sec

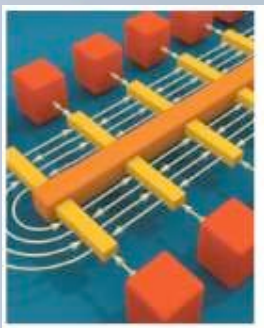


Cray SeaStar (System-on-Chip) SOC

- 130nm ASIC
- embedded 3-D torus router
- 500 MHz
- 3.2Gb/s signal rate
- 12-bit wide ports
- 460 Gb/s off-chip bandwidth



Seastar Interconnect: Balance



Guest Editors' Introduction

HIGH-PERFORMANCE INTERCONNECTS

SEASTAR INTERCONNECT: BALANCED BANDWIDTH FOR SCALABLE PERFORMANCE

THE SEASTAR, A NEW ASIC FROM CRAY, IS A FULL SYSTEM-ON-CHIP DESIGN THAT INTEGRATES HIGH-SPEED SERIAL LINKS, A 3D ROUTER, AND TRADITIONAL NETWORK INTERFACE FUNCTIONALITY, INCLUDING AN EMBEDDED PROCESSOR IN A SINGLE CHIP.

May-June issue of IEEE Micro

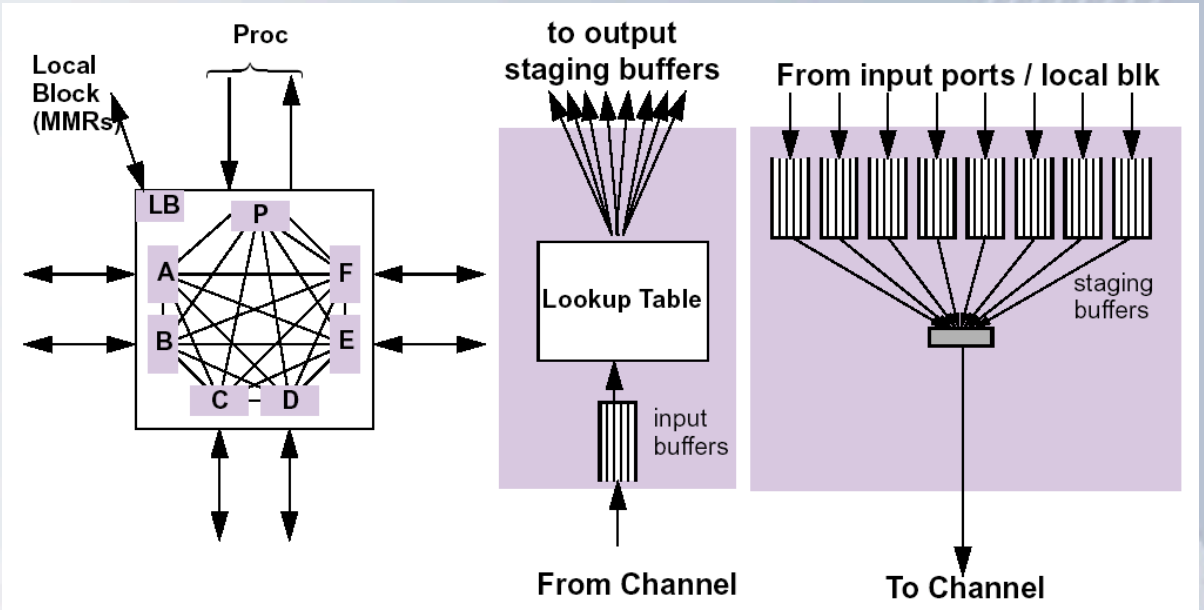
- Issue is dedicated to high-performance interconnects
- Features a paper from Brightwell, Underwood & Pedretti on Cray's SeaStar Interconnect
- Good description on how SeaStar is designed specifically for MPI
- Table from paper

Table 1. Network bandwidth balance ratios.

Machine	Peak node speed (Gflops)	Peak node bandwidth (Gbytes/s)	Ratio (Gbytes/Gflops/s)
ASC Purple	48.0	8.0	0.17
ASC Red Storm	4.0	4.8	1.2
Blue Gene/L	5.6	0.35	0.0625
Columbia	24.0	6.4	0.27
Earth Simulator	64.0	12.3	0.192
Mach 5	16.0	0.5	0.03
MareNostrum	17.6	0.5	0.028
Thunder	22.4	1.0	0.04
Thunderbird	14.4	2.0	0.13

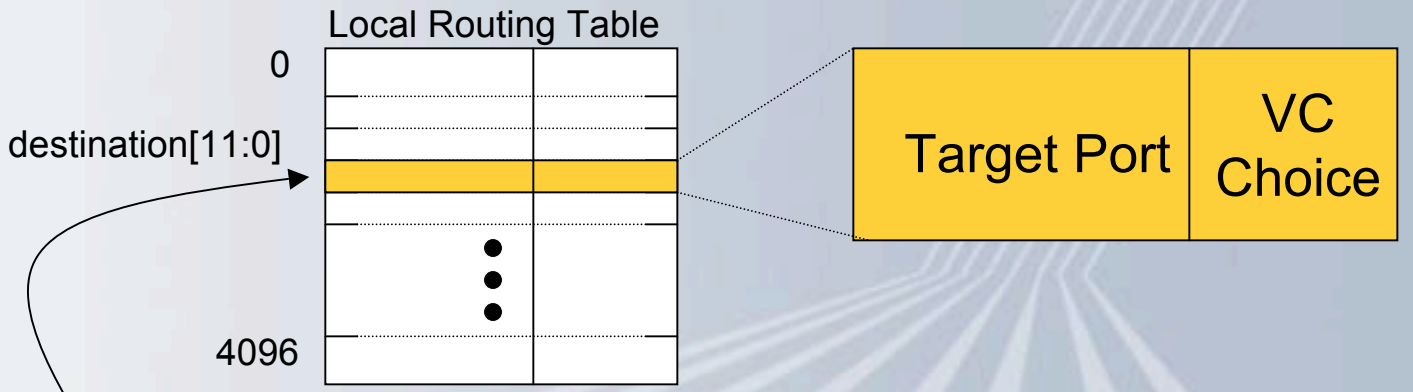
Seastar router block diagram

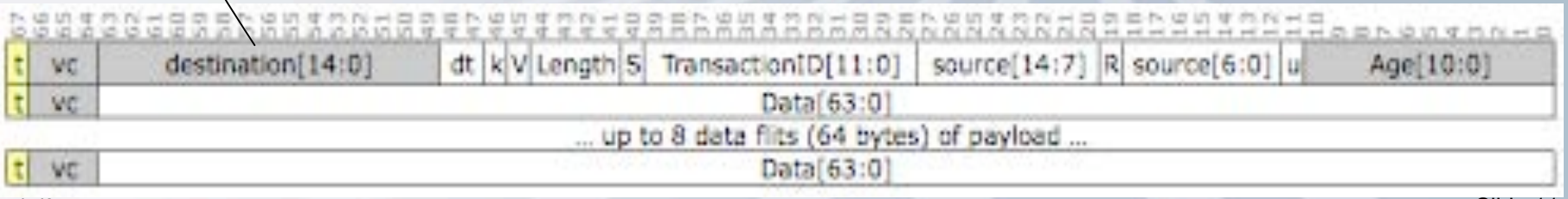
- 6-ported 3-D torus router with 12-bit network channels
- Fully buffered input and output ports
- Only 6 clocks of zero-load (fall-through) latency
- Routing is performed by a lookup table at each input port
  - Can route a new packet every clock cycle



Packet Structure

- Packets consist of one or more *flits*
- Header flit contains routing fields
  - Virtual channel, Destination, and Age
- Global and local routing tables
  - 8 **global** table entries (selected by destination[14:12])
  - 4K **local** routing table entries

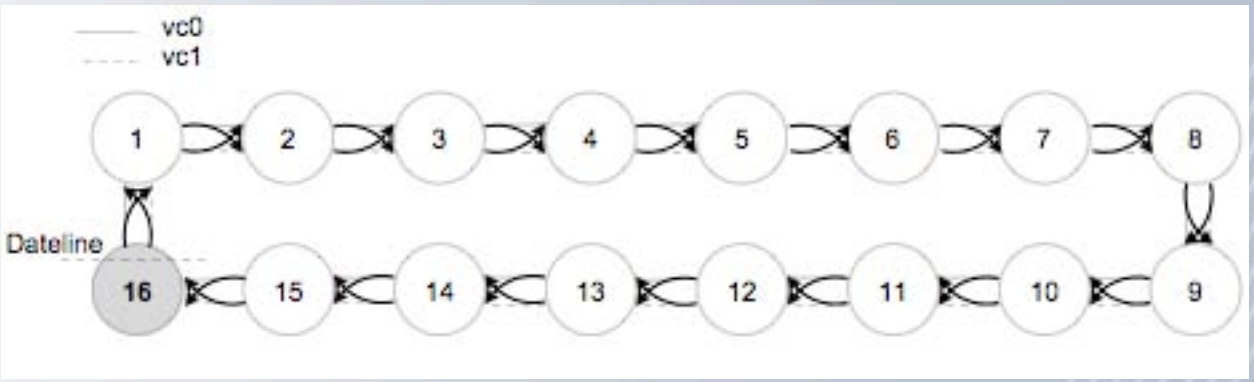


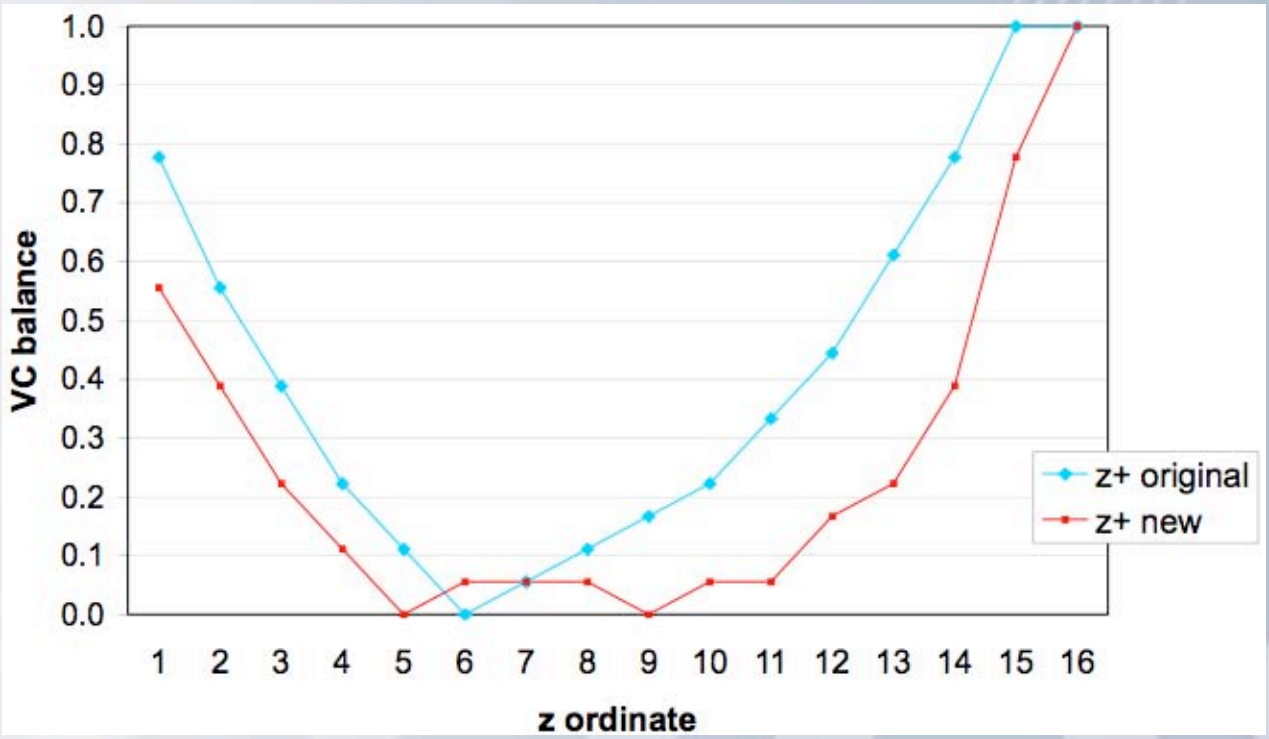


Virtual Channels

- Virtual channels (VCs) allow multiple buffers to be multiplexed onto the same *physical* channel.
- Provide better throughput by avoiding head-of-line blocking
- Primarily used for avoiding deadlock on torus links
  - Cray XT uses VC0 and VC1
- One node in each dimension is labeled as the “dateline”
  - Ensures that traffic “crossing” the dateline is on the appropriate VC
  - A packet that is going to cross the dateline to reach its destination must start on VC0 and switch to VC1 when it crosses the dateline.
- This introduces imbalance near the dateline

VC Datelines





Balancing the load on virtual channels…

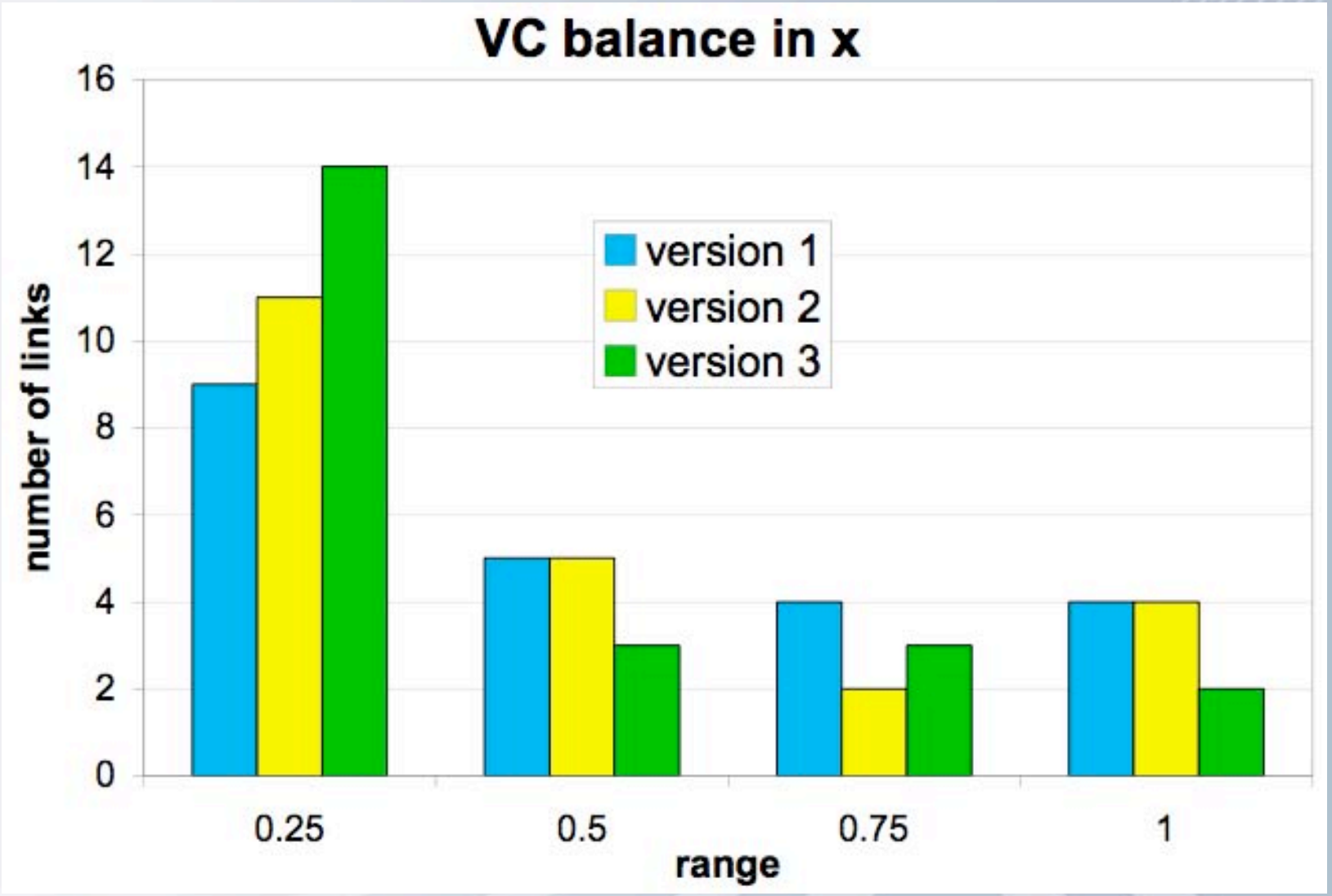
Improving VC buffer utilization

- Non-uniform virtual channel usage can result in a significant variation in network performance depending on the processors position in the network [Adve and Vernon]
- By *balancing* the relative traffic carried be each virtual channel, we can have a significant effect on the overall network performance.
  - Reducing the effects of head-of-line (HoL) blocking that leads to congestion in the network

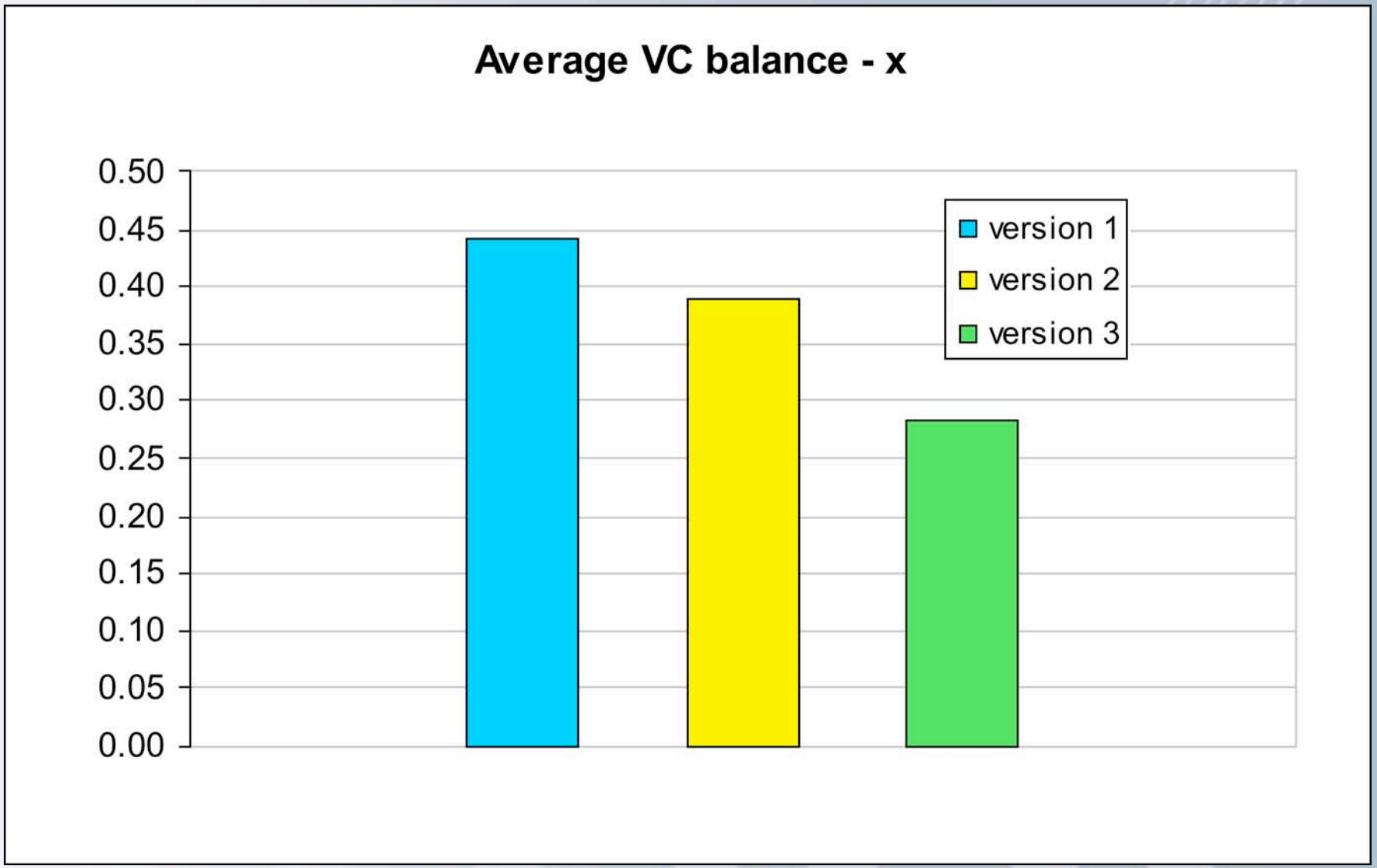
VC Assignment Policy

- Version 1 - dateline neighbors
  - All traffic *into* the dateline enter on VC0 and exit on VC1
  - This causes acute VC imbalance around the dateline node and its neighbors
- Version 2 - global routing table and dateline crossing
  - The local routing tables can precisely route up to 4K nodes
  - Systems >4K nodes must first route to the correct “global” region using the Global lookup table (GLUT)
  - Determining if the packet will cross a dateline in the X dimension when moving toward the correct global region makes it a candidate for balancing
- Version 3
  - Dateline crossing within the region is always selected on the edge of the region
    - Guarantees that packets entering the region (within one direction) will never cross the dateline
      - otherwise it would be destined to a different global region

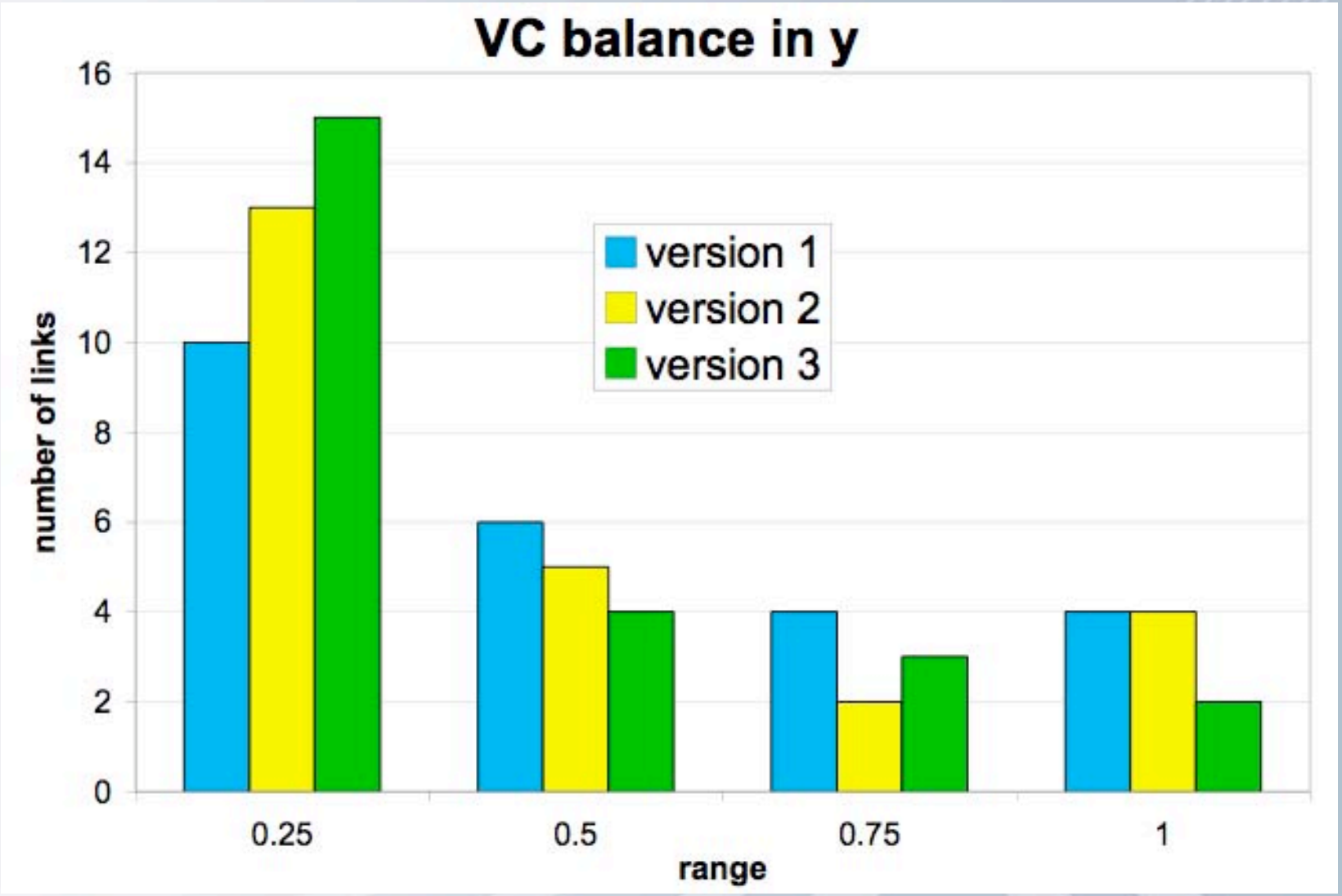
Results - X dimension



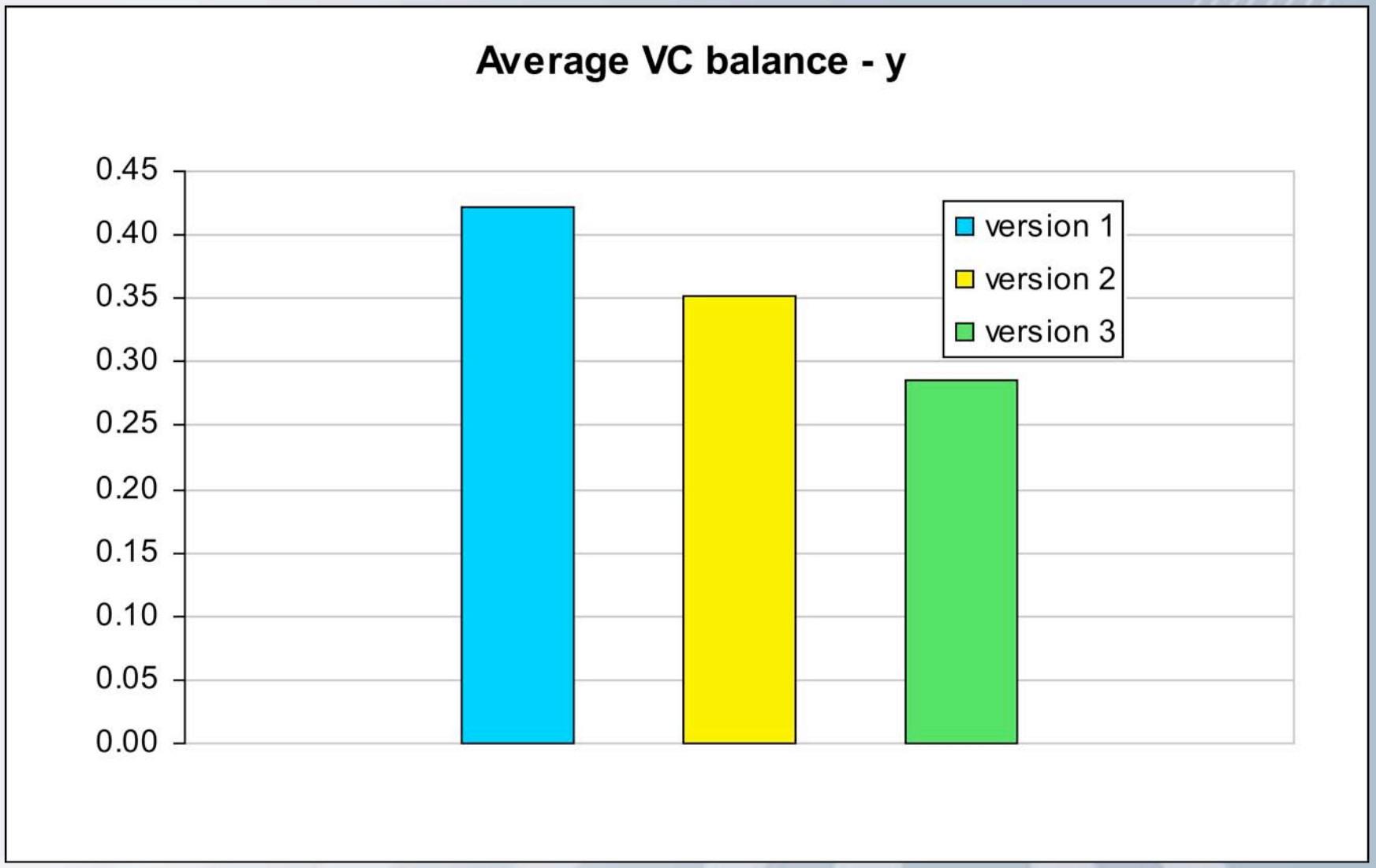
Results Summary - X dimension



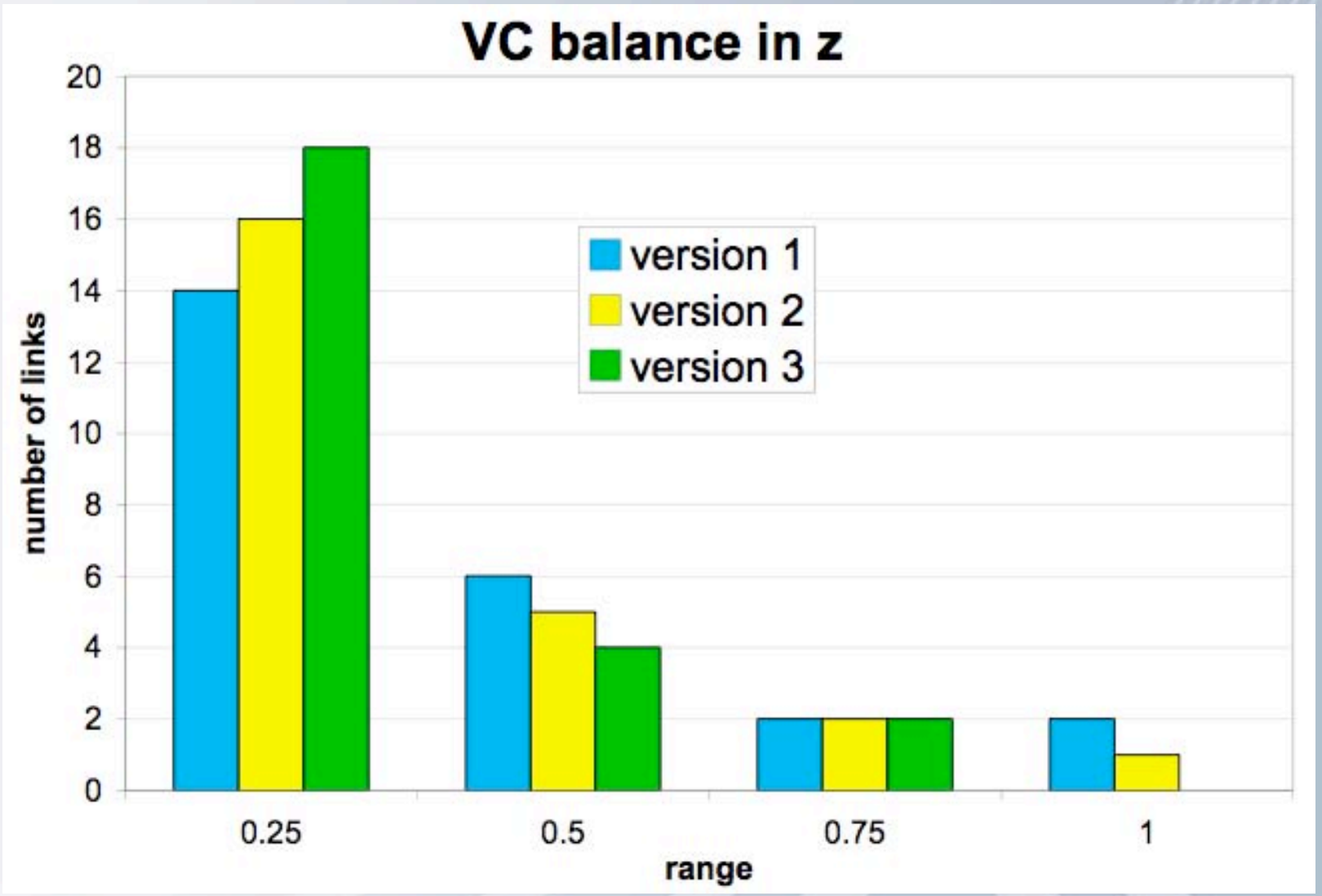
Results - Y dimension



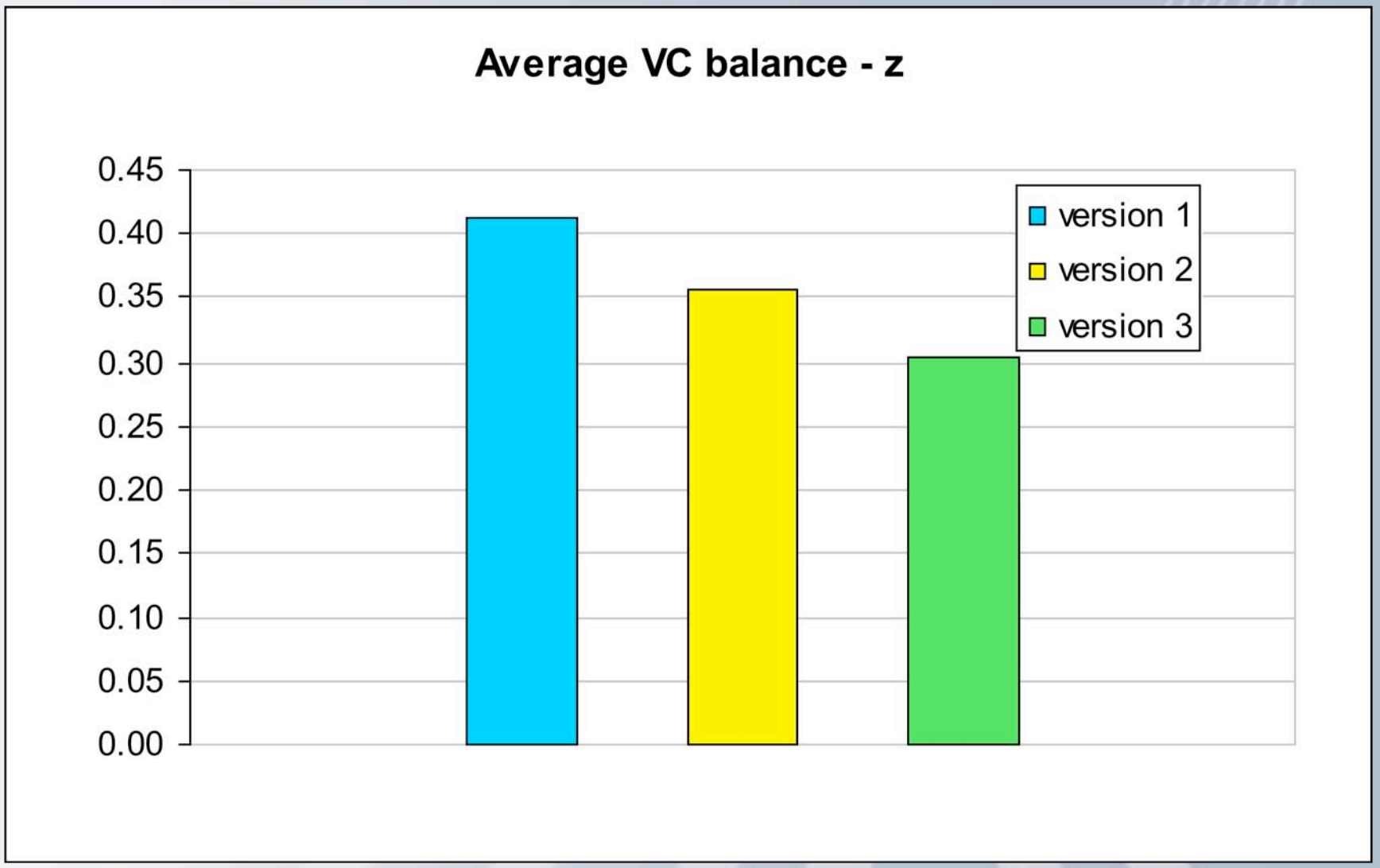
Results Summary - Y dimension



Results - Z dimension



Results Summary - Z dimension



HPCC Results

Measurements made on a production XT3 system 11x12x16 torus “BigBen” at PSC

Benchmark	% improvement
MPI-FFT (2048 PEs)	18.1%
PTRANS (2016 PEs)	17.4%

Future Improvements

- The Cray T3D proposed a simulated annealing approach to balance virtual channels across datelines
- Reconfiguring the NID assignment to allow different placement of the dateline
- Study the effects of job placement of smaller jobs to avoid imbalance around the dateline nodes

```
temp = HOT
repeat
   for tries = 1 to TRIES_PER_TEMP
      pick a random unconstrained route to flip (change VC)
      E = change in cost function from flipping route
      if (E<0)
          accept the modification
      else
          accept the modification with probability e^(-E/temp)
   flips = number of accepted flips at this temperature
   if flips > 0.6*TRIES_PER_TEMP
       temp = temp / 2
   else
       temp = temp * COOL
until flips = 0
```

Conclusions

- We use virtual channel “datelines” to avoid overlapping dependencies around the torus links
- Buffer space in a high-performance router is a precious commodity and must be balance to avoid unnecessary head of line (HoL) blocking
- We show results of our optimized VC balance algorithm that improved performance
  - Show the optimization and its impact on buffer utilization
- Improved VC balance by about 50%
  - X dimension: 44% down to 27%
  - Y dimension: 42% down to 28%
  - Z dimension: 42% down to 30%
- The improved buffer utilization produced 18.1% increase for MPIFFT and 17.4% improvement in PTRANS results

Acknowledgements

- We would like to give special thanks to the team at Pittsburgh Supercomputing Center (PSC) who collaborated and gave feedback on early versions of the software
- BigBen is an 11x12x16 XT3 system (now with Dual core!)

Thank You…

Questions?

CRAY
THE SUPERCOMPUTER COMPANY