Topology-Aware Job Scheduling Strategies for Torus Networks

Cray User Group
May, 2014

J. Enos, G. Bauer, R. Brunner, S. Islam
NCSA Blue Waters Project

R. Fiedler
Cray, Inc.

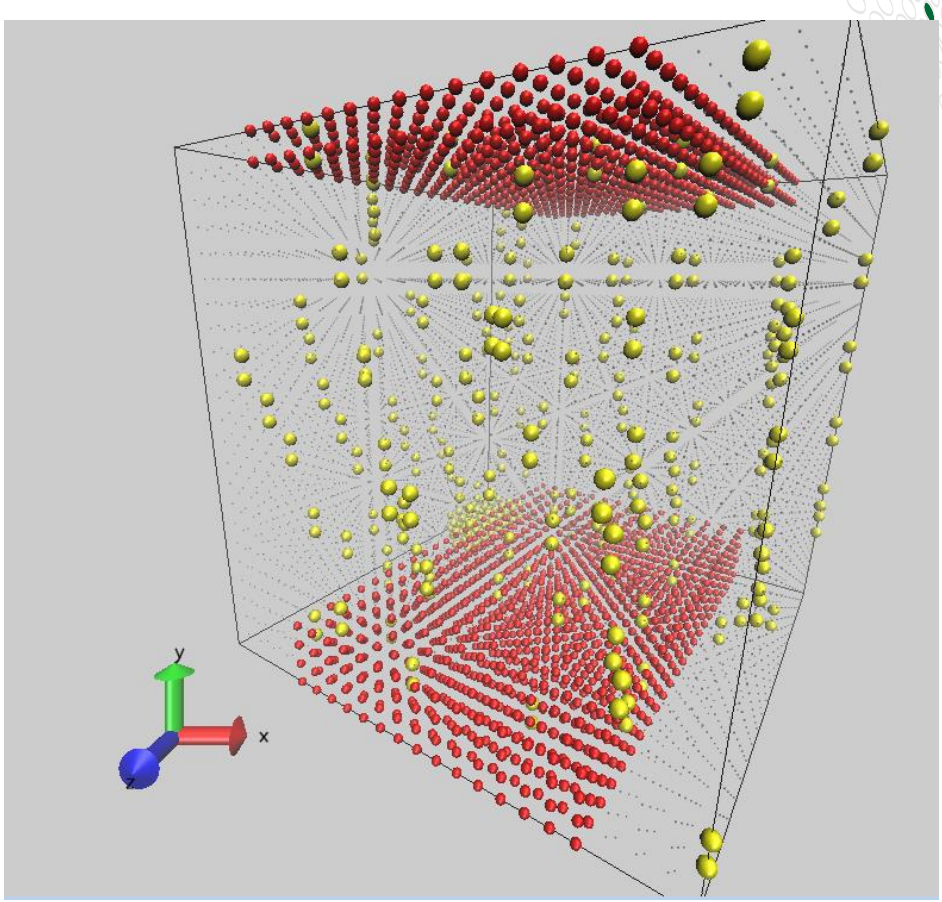
M. Steed, D. Jackson
Adaptive Computing

Outline

- **Application run time variability due to task placement**
- **Mitigation attempts**
  - New ALPS node ordering scheme
  - Predefined node allocation shapes
- **Topology-Aware scheduling in Moab**
  - Goals & design
  - Synthetic workload
  - Preliminary results on utilization, application performance, throughput
- **Leveraging Topaware task layout tool**
  - New scheduler & Topaware features enable near-optimal layouts
  - Performance improvements for halo exchanges
  - Performance improvements for MILC
- **Conclusions and next steps**

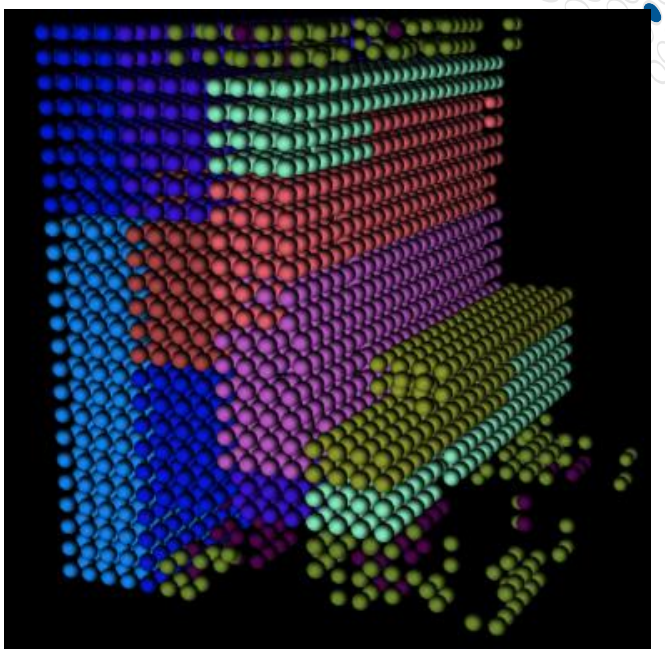
Application Run Time Variability

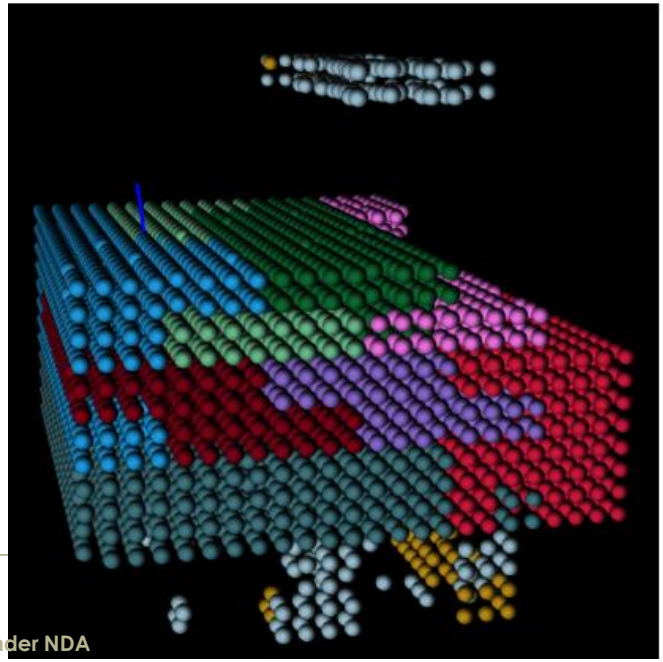
- **Blue Waters torus**
  - 24x24x24 gemini routers, 2 nodes each
  - XEs plus15x6x24 XK block
  - Scattered service nodes
  - Links along x & z dimensions 2X faster than links along y
- **Run times varied widely**
  - More if communication intensive (PSDNS, MILC)
  - Up to 4X longer than ideal
  - ALPS favored lower bandwidth yz slabs
  - Allocations often fragmented, increases link contention



New Node Ordering Scheme

- **Default ALPS node ordering**
  - Favors yz slabs 4 thick in x
  - Lower bisection bandwidth than xz slabs due to link speeds
  - PSDNS runs 1.6X faster in 24x6x24 gemini prism than in 6x24x24 prism
- **Developed new scheme**
  - Major help from Cray's C. Albing
  - Favors xz slabs, 2 or 4 thick in y
  - Reverses direction when jumping to next level in y
  - Does not reduce job-job interference
  - Helps application performance





New Node Ordering Scheme

- Workload test: speedups for 7-8 concurrent applications

App	Nodes	Nov.11 2Y	Nov.11 4Y	Nov.4 2Y	Nov.4 4Y
MILC	1372	1.00	1.02	1.15	0.91
MILC	2744	1.52	1.47	1.43	1.31
NWChem	3000	1.34	1.22	1.32	1.39
PSDNS	1024	1.09	1.15	1.22	1.74
Changa	1024	--	--	1.00	0.95
NAMD	1368	1.62	1.77	0.91	0.91
WRF	1386	--	--	1.01	1.01
CESM	600	1.01	1.00	--	--
DNS_distuf	512	1.13	1.13	1.05	0.98
AVERAGE		1.24	1.25	1.14	1.19

Predefined Node Allocation Shapes

- **Uses Moab "nodesets"**
  - Favorable shapes like 24x6x24, 12x12x12, 12x6x12 geminis
  - Shapes can overlap
  - Same shape can be replicated throughout system
    - E.g., several different 24x6x24 prisms in 24x24x24 torus
- **Target nodesets at time of job submission**
  - #PBS –l nodeset=ONEOF:FEATURE:s1_24x6x24,s2_24x6x24,…
  - Job will run in first available requested feature
  - Good run time consistency for PSDNS in 24x8x24 nodeset
  - Queue wait times can be long
  - Special arrangements needed to reserve a specific nodeset
  - Limited number of predefined shapes to choose from
  - Number of compute nodes in each nodeset differs
  - Job-job interference reduced but not eliminated

Topology-Aware Scheduling in Moab

- **NCSA/Cray/Adaptive collaboration**
- **Goals**
  - Improve application performance on large XE and XK systems through better-localized job placement
  - Improve application run-time consistency by eliminating job-job interference due to application communication
  - Improve system throughput and maintain reasonably high utilization
- **Approach**
  - Allocations are regular right prisms
  - Eliminate interference: allocation either spans a torus dimension, or spans < ½ of geminis in a dimension
  - Favor xz slabs
  - Boost utilization by more freely placing jobs that perform/are not affected by application communication
  - Allow applications to request specific allocation shapes (Topaware)

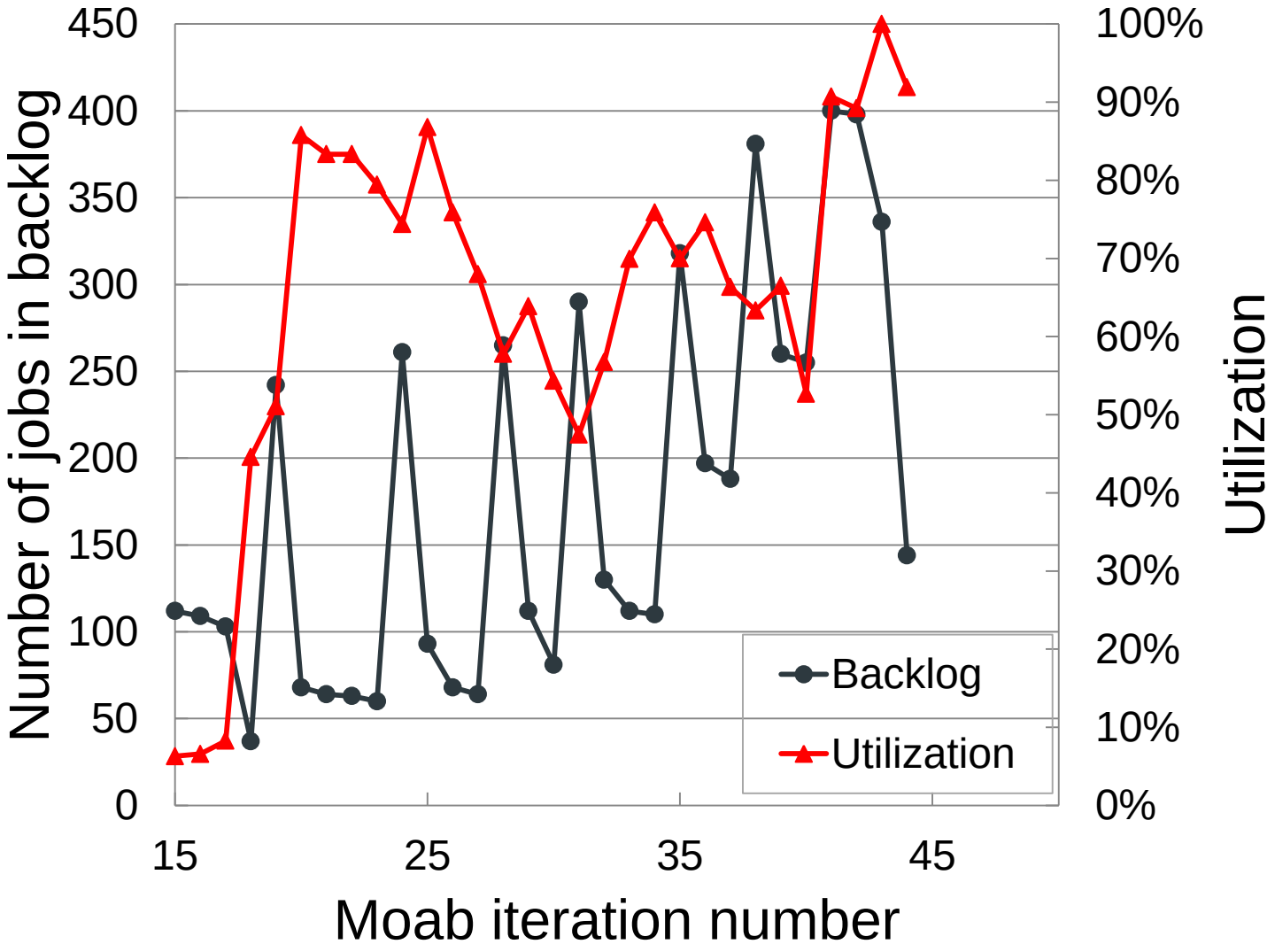
Workload Test

- **Synthetic workload with representative applications**
  - MILC, PSDNS, NAMD, NWCham, Changa, QMCPACK, DNS_distuf, WRF, SpecFEM3D_globe
  - Broad range of communication patterns
  - Numerous representative node counts and run times based on actual Blue Waters production logs
  - Small node counts much more numerous than large node counts, but bulk of service units consumed by larger jobs
  - Tested at scale on Blue Waters
  - Matches backlog, job submission timing, node count distribution, requested run time distribution (scaled to 3 hr test window), node type
  - Matches scheduler policies, priorities, limitations
  - Starting state matches fragmentation of production environment
- **Measure utilization, app performance & consistency**

Preliminary Workload Test Results

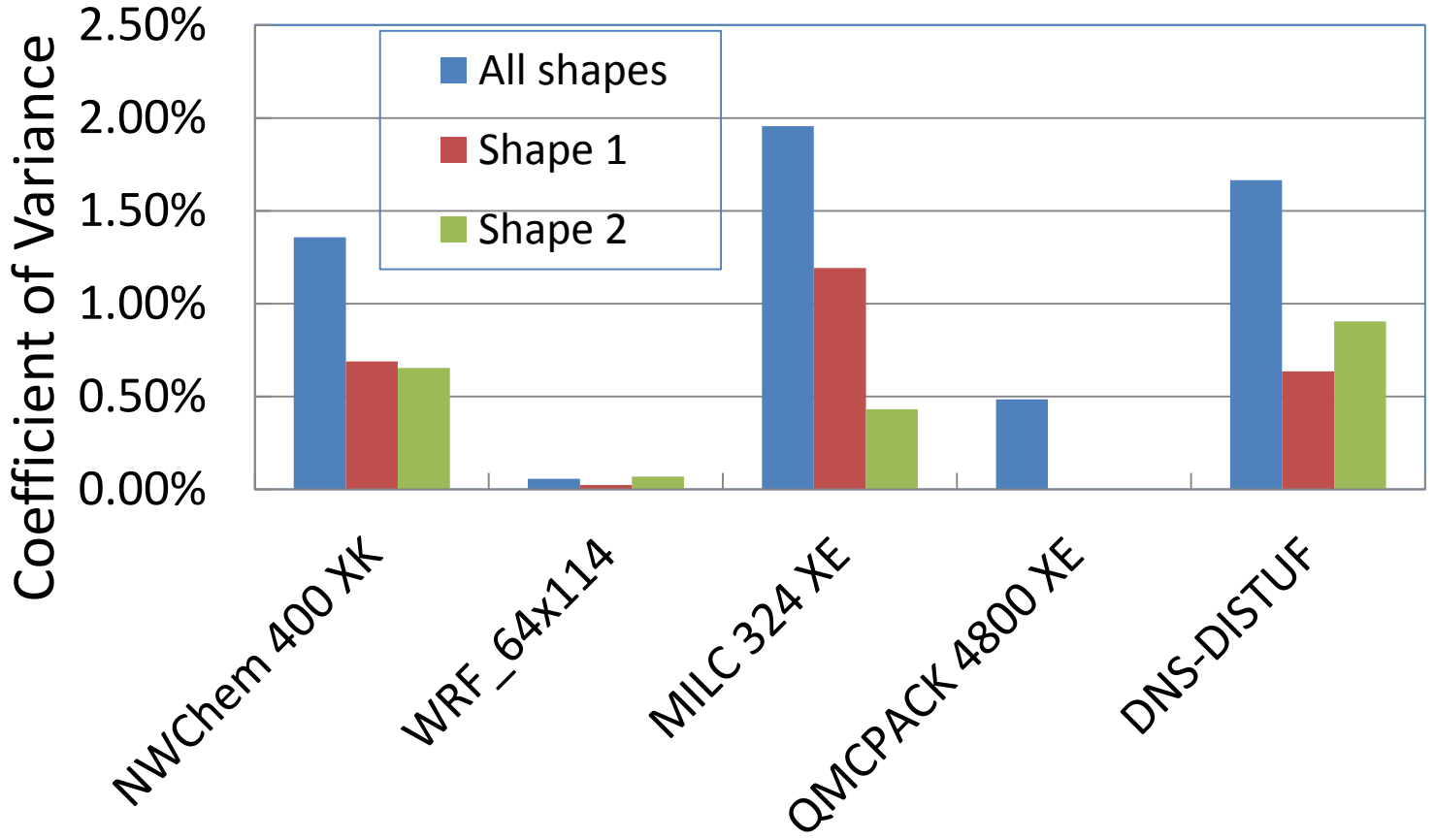
- **Topology aware test conducted Apr 22-23**
- **1544 jobs run in two hour window**
  - Gemini network failure 2 hours into 3 hour scale test
  - Hardware warm-swapped later – aborted non-topology ‘control’ test
- **Good scheduler performance (71.1% average utilization)**
  - Backfill constrained by workload – utilization could have been better
    - 200 jobs per submission cycle (38% of jobs 8 nodes or less)
    - Backlog too small (184 jobs avg in test vs ~300 for production operation)
    - Job durations too consistent (91% of all jobs had same duration)

Preliminary Workload Test Results
Utilization and Backlog



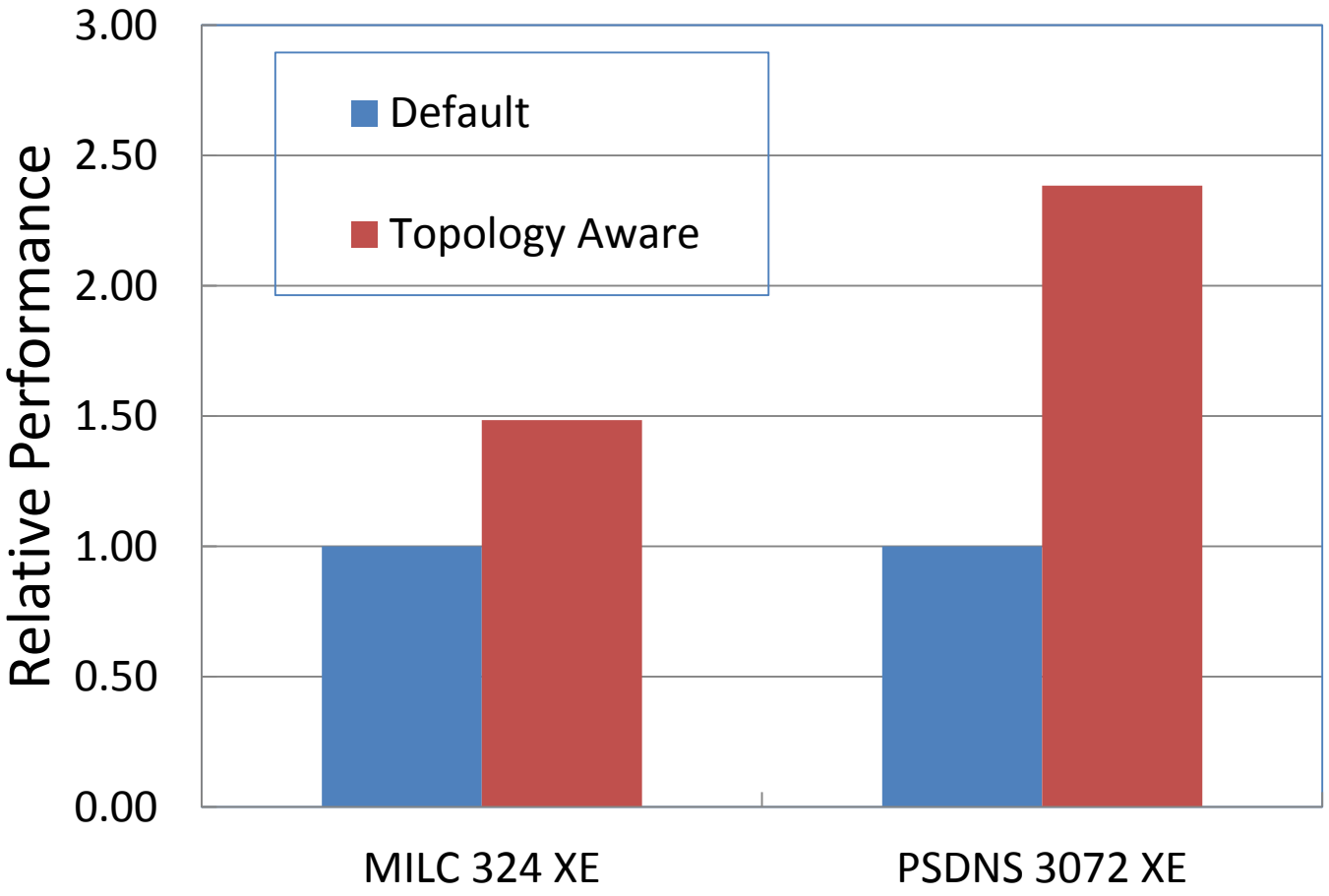
Preliminary Workload Test Results

- **Application Consistency**
  - Worst Application run time CoV is less than 2%
  - Worst 'Per Shape' Application CoV is less than 1.25%



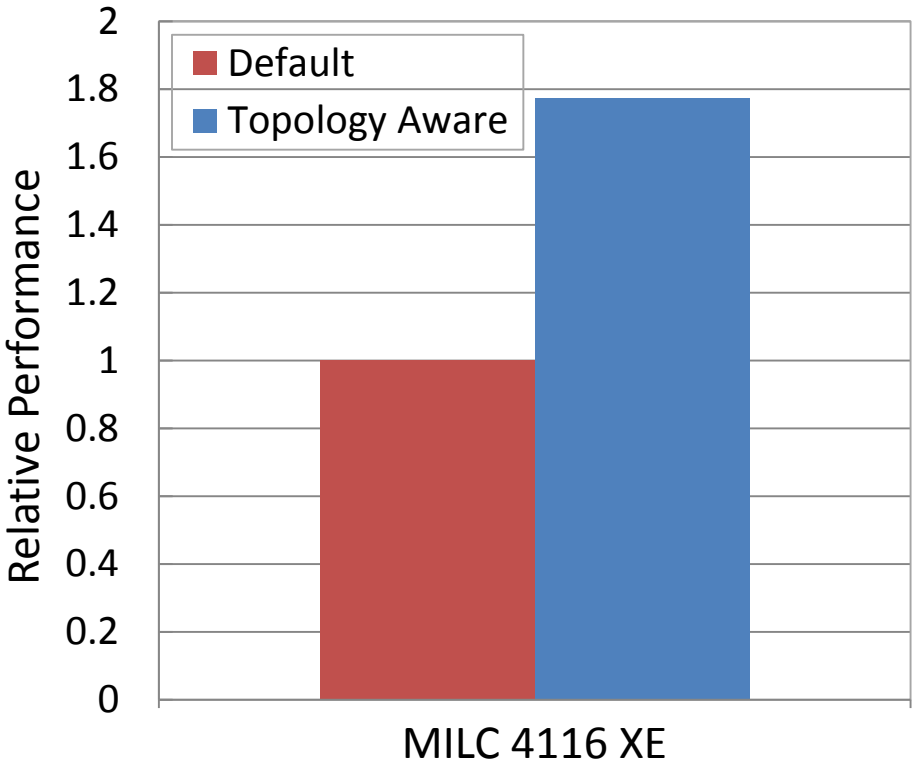
Preliminary Workload Test Results
Application Run Time Comparison

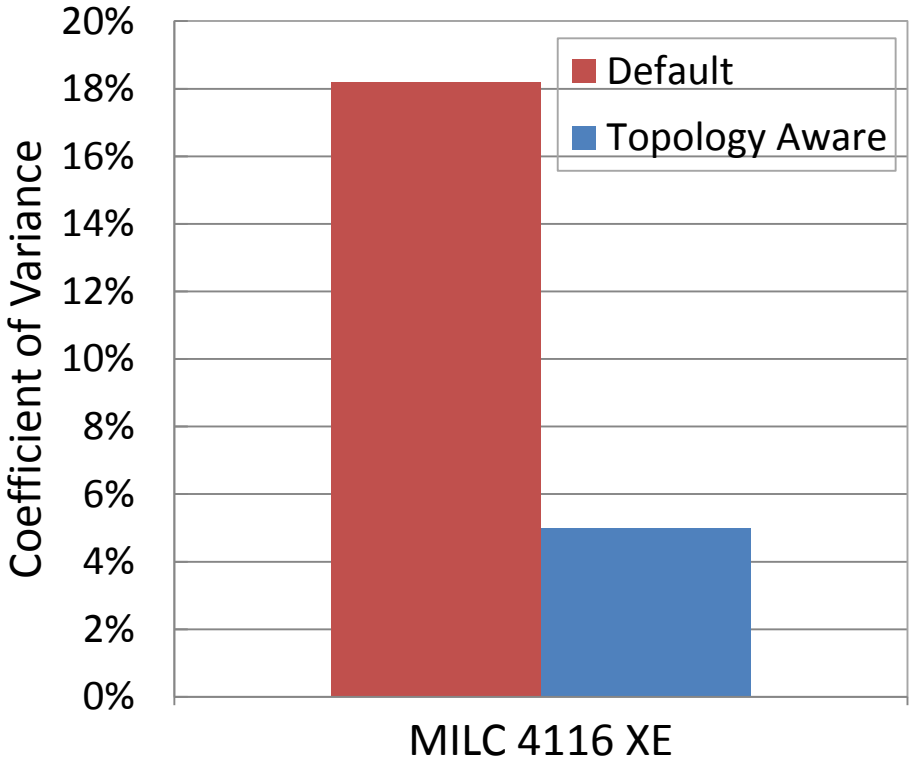
- **Limited default scheduler results for comparison**
  - Sample size too small, but gives example speedups



Default vs.Topology Runs (using grid_order)

- Avg. run time w/default scheduler 79% higher (8 samples)
- CoV < 5% w/topology-aware scheduler (2 samples)



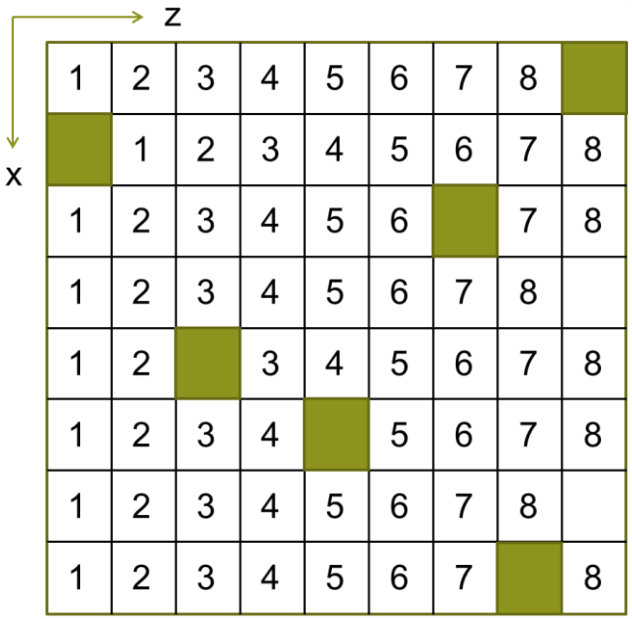


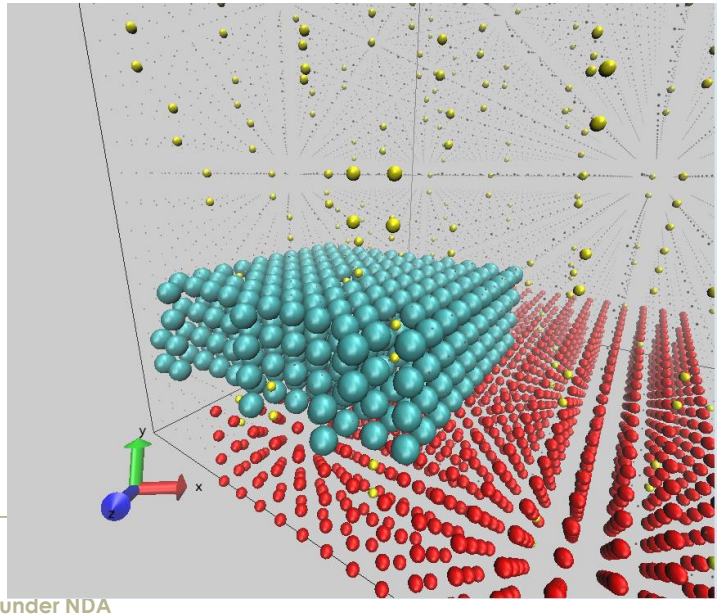
Effective Throughput Estimate

- Effective Throughput =
  % scheduling performance * % application performance
- Biggest improvements in application performance seen for largest jobs
- Over 76% of Blue Waters compute cycles delivered to jobs > 512 nodes
- Scheduling efficiency in benchmark and simulation runs ~75-85% vs. 90% for default scheduler
- Sample size too small without ‘control run’, but >40% improvement in large application performance is common
- Assuming 77% scheduling performance and 140% weighted application performance boost, throughput improvement = (77/90) * 1.40 = 1.19X

Topaware Node Selection and Task Layout Tool

- **Provides near optimal task mapping for 2, 3, & 4D Cartesian grid virtual topologies**
  - In each z-pencil, set of selected geminis along z is extended if needed to skip unavailable nodes
  - Determines multiple valid layouts and evaluates layout quality
  - Allows unbalanced layouts
    - Nodes on prism boundaries may have fewer tasks
    - Enables more good layouts for more virtual topology sizes
  - Scheduler ensures allocation has desired gemini count in each z-pencil





Topaware Unbalanced Layouts

- Halo exchanges for virtual topology: 32 by 32 by 32

Placement	Iter time (ms)	Max hops
Default 8x8x8	11.315	9
Grid_order 8x8x8	7.722	16
Topaware 8x8x8	2.771	2
Topaware 11x6x11 (unbalanced)	1.287	2
Topaware 11x8x8 (unbalanced)	1.147	2
Topaware 8x8x11 (unbalanced)	1.214	2
Topaware 11x7x8 (unbalanced)	1.782	2
Topaware 8x7x11 (unbalanced)	1.737	2
Topaware 11x8x7 (unbalanced)	1.580	2
Topaware 7x8x11 (unbalanced)	1.690	2

Topaware and Real Apps

- **MILC**
  - Virtual topology 21 by 2 by 21 by 24
  - 1764 nodes, 12 tasks each
  - 21x2x21 geminis
  - 2.2x faster with Topaware than with grid_order –c 2,2,2,2 on same nodes

Placement	Run Time (10 iterations)
Grid_order	254.0
Topaware	116.4

Conclusions and Next Steps

- **New ALPS node ordering scheme favors xz slabs & improves application performance by 12-19%**
- **New topology-aware Moab scheduler provides prism-shaped allocations which further improve application performance**
  - Maximize bisection bandwidth, reduce latency
  - Eliminate job-job interference
- **Workload test demonstrates 40% better overall large application performance and 4X better run time consistency**
  - Utilization averages 71%, expect this to increase to ~77%
  - Expected system throughput improvement estimated at 19%
- **Topaware provides impressive speedups for nearest-neighbor communication**
  - MILC 2.2X faster than with grid_order on same nodes
- **New scheduler provides prism allocations required for near-optimal Topaware layouts**
- **Workload retest needed to compare default scheduler and new scheduler + Topaware for MILC runs**

CRAY