BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Applications and their challenges on Blue Waters

Greg Bauer

GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION

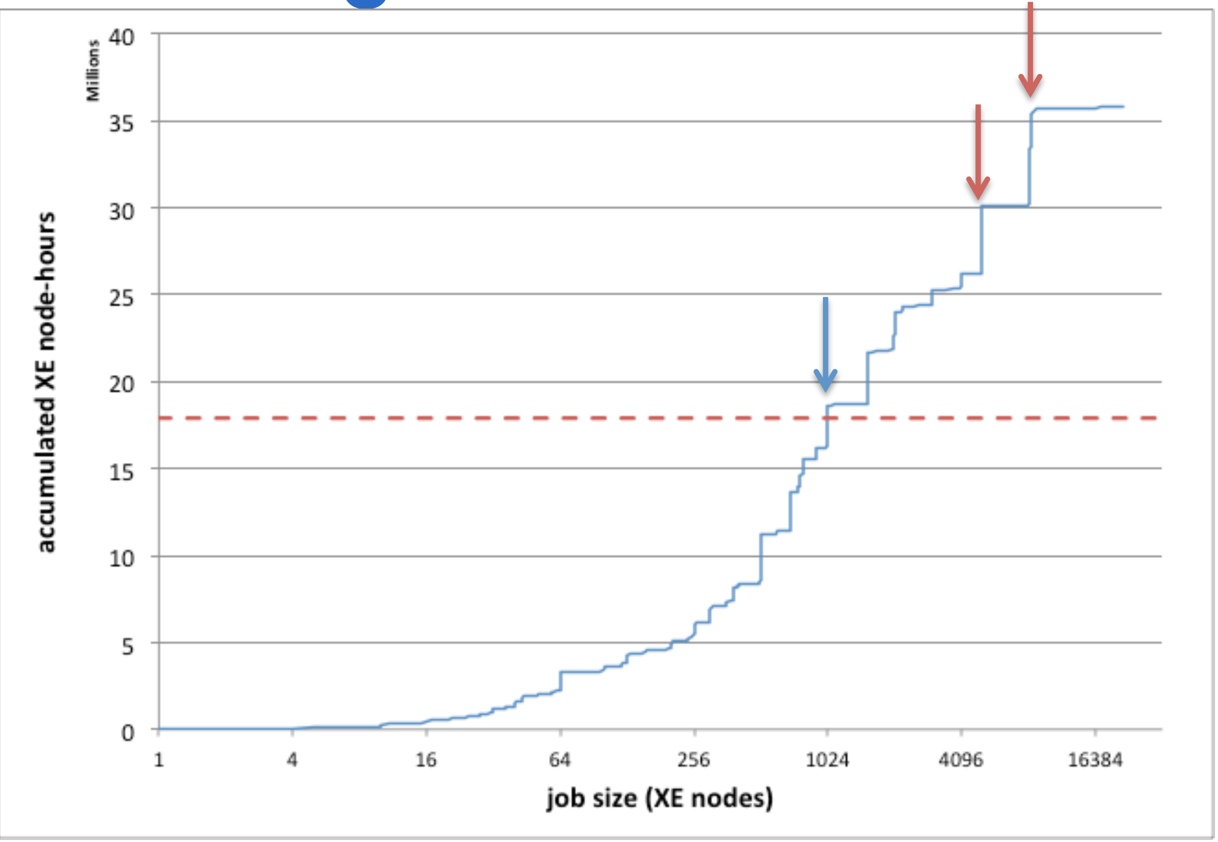
Overview

- What is running on Blue Waters?
- What are the issues and what to do about them?
  - Scalability
  - Runtime consistency
  - Other job interference
  - IO
  - Congestion Protection
  - Interrupts

Changes to the system

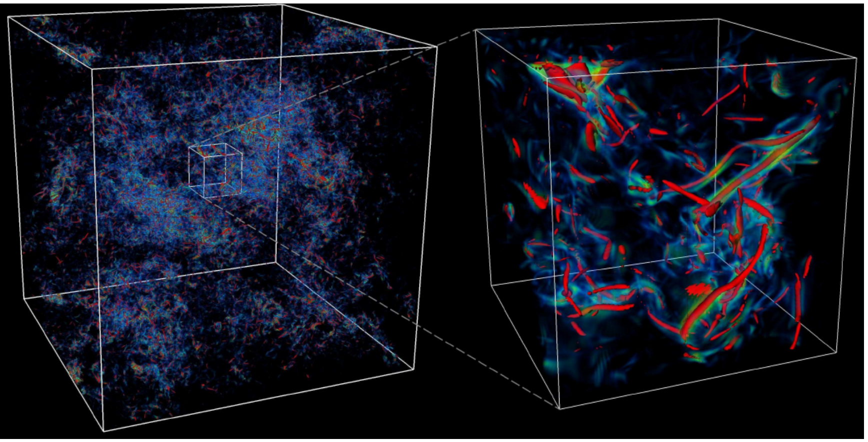
- More XK nodes
  - From 3,072 to 4,224.
- Flattened XK region in torus
  - From 8x8x24 to 15x6x24.
- LNET nodes redistributed across XE and XK
  - Good – Improved aggregate bandwidth within the XK region of the torus (more X links, fewer Y links). LNETs in XK region provide possible (future) co-location of compute and IO.
  - Not so good – LNETs in region (IO was going through XK region anyway). X dimension now greater than ½ total X dimension. Requires topology aware scheduling.
- Testing with XK acceptance applications showed either little change or improved performance for 'before/after' comparison.

XE Usage in the last 3 months



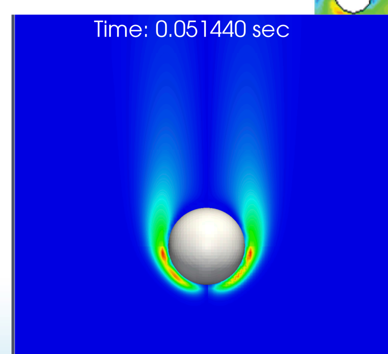
- 50% of usage is 1,024 nodes or larger.
- Two teams using 5,000 and 8,192 nodes.
- During Friendly User period, several teams sustained runs at full system.
- Nothing prevents users from submitting very large jobs and priority goes to larger jobs.
- Average expansion factor for large jobs < 10.

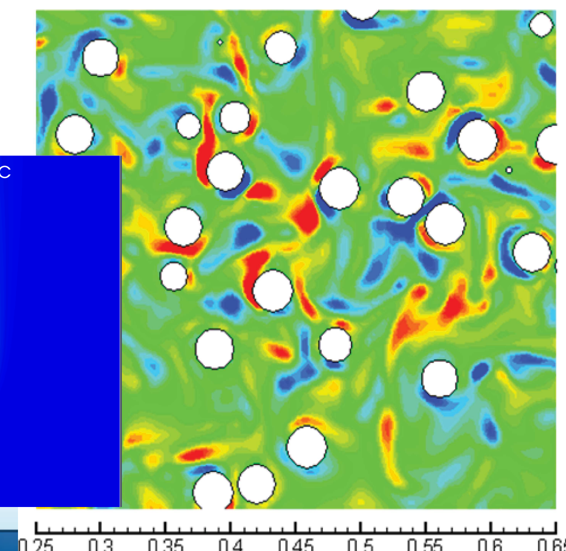
Turbulence



PSDNS – 3D FFTs, off-node transposes using CAF replacement for the concurrent Alltoalls. Routinely running at 8,192 nodes (262,144 tasks) for $8{,}192^3$ problem in 48 hr. chunks.

- DISTUF – DNS using PETSc CG for direct Poisson solve. Looking at using MG. Scaling and code validation underway. Up to 512 nodes.



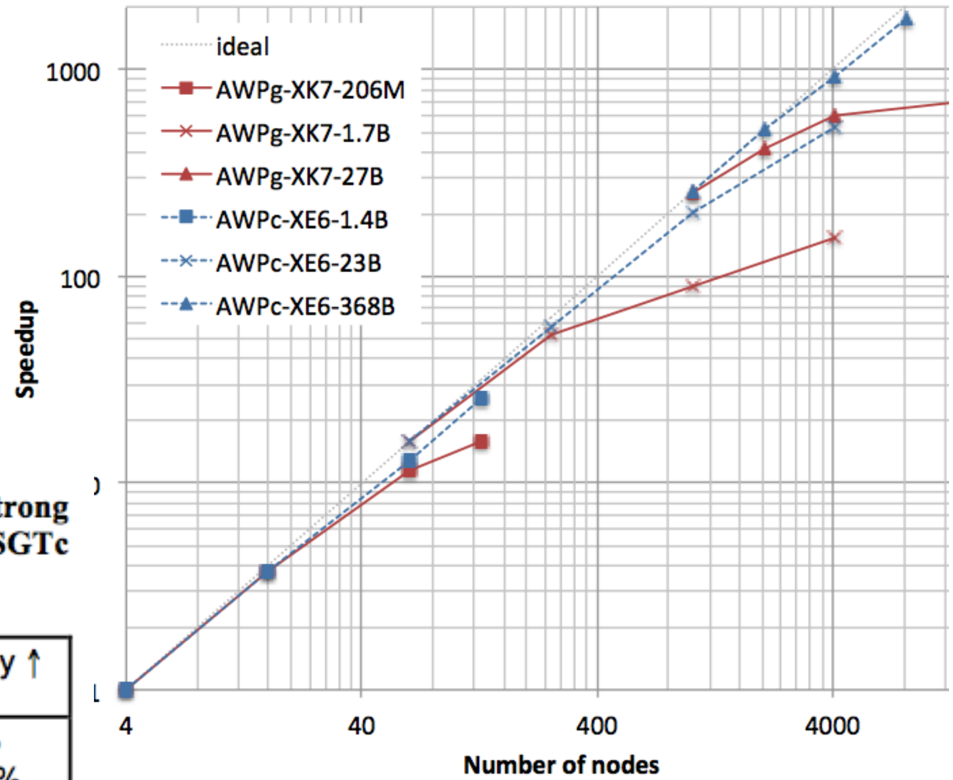


Cybershake

- Scalability issues with jobs on busy system. Cray (Fiedler) Topaware improved node selection and rank ordering.
- Looking at ways of using host CPU on XK nodes for part of workflow while GPU is doing computation.

Table 1: Topology tuning with Topaware tool improved strong scaling efficiency for fixed 45B mesh point AWP-SGTc benchmark calculation with 64, 512, and 4096 nodes

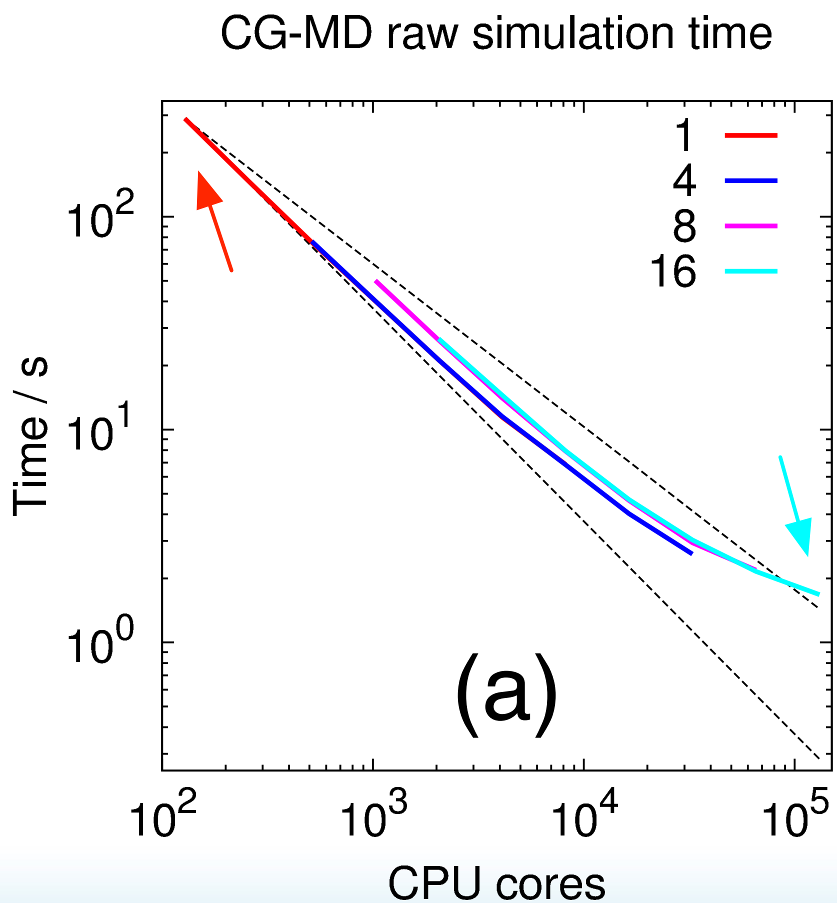
#nodes	Default	Topaware	Speedup	Efficiency ↑
64	4.006	3.991	0.37%	100% → 100%
512	0.572	0.554	3.15%	87.5% → 90%
4096	0.119	0.077	35.29%	52.6% → 81%



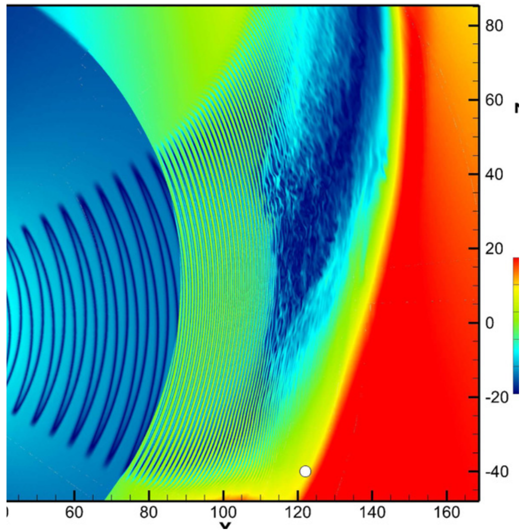
Paper in Extreme Scaling Workshop 2013

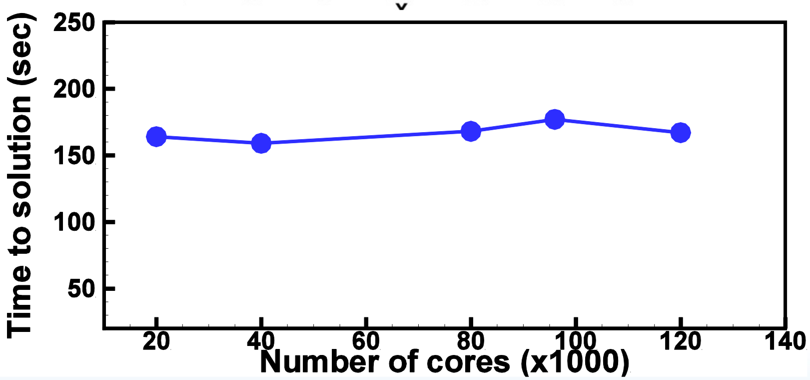
Coarse Grained MD

- Novel MD algorithm.
- Improved memory usage.
- Hilbert space filling curve (SFC) for load balancing.
- Dynamic communications mapping to handle irregular SFC boundaries.
- Scaling to 16,250 nodes (260,000 FP cores). Earlier data shown at right.
- Mostly MPI but replaced some functionality with DMAPP when faster.
- BW Symposium - https://bluewaters.ncsa.illinois.edu/web/portal/symposium-may-2013
- ACS paper - http://pubs.acs.org/doi/full/10.1021/ct400727q



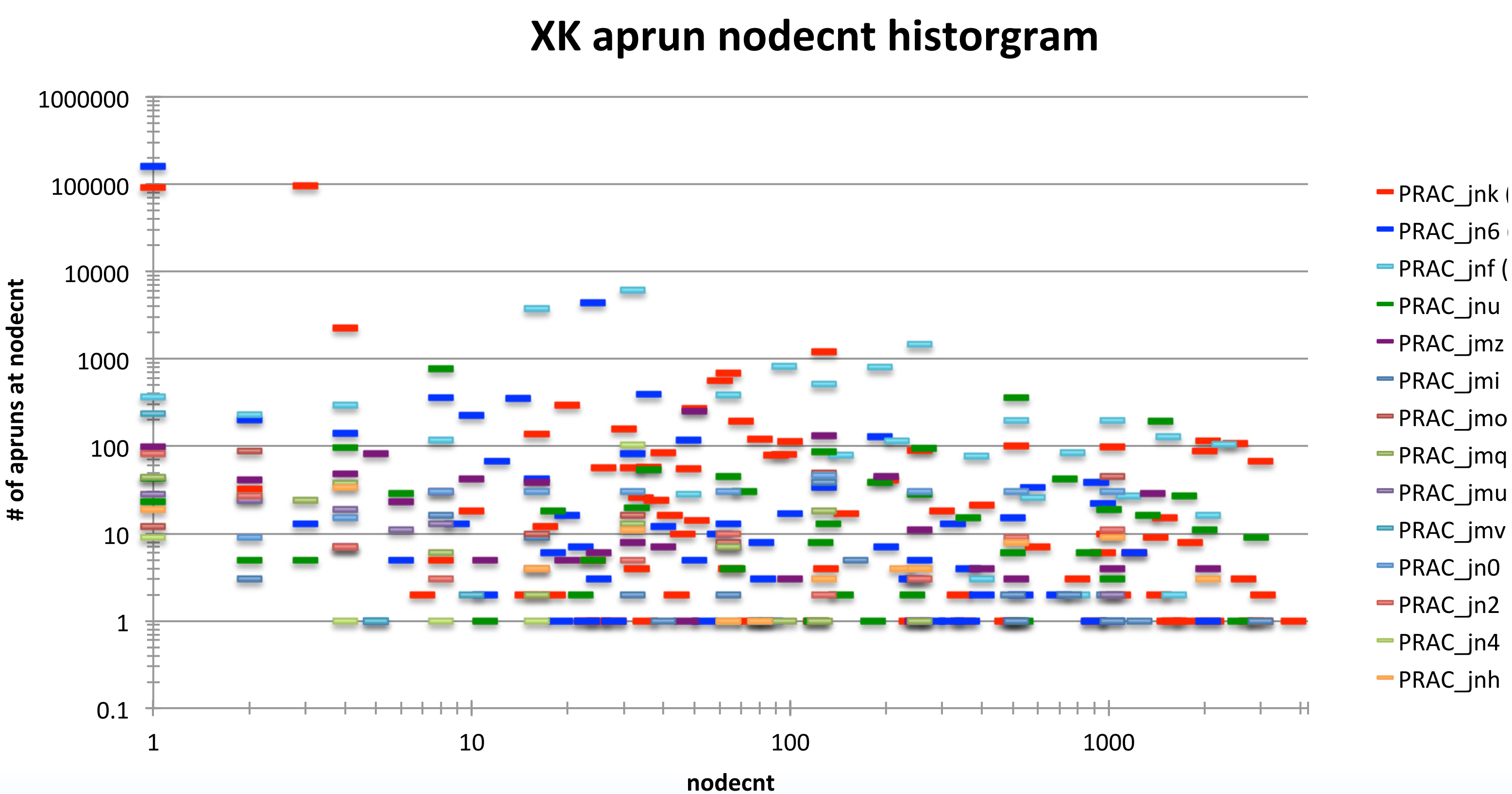
Multi-Scale Fluid-Kinetic Simulation





- MHD-kinetic code to modeling the solar wind.
- Chombo framework for AMR and dynamic load balancing.
- P3DFFT
- Good strong (starting at 1,250 nodes) and weak scaling to 7,500 nodes.
- http://adsabs.harvard.edu/abs/2013ASPC..474..165P

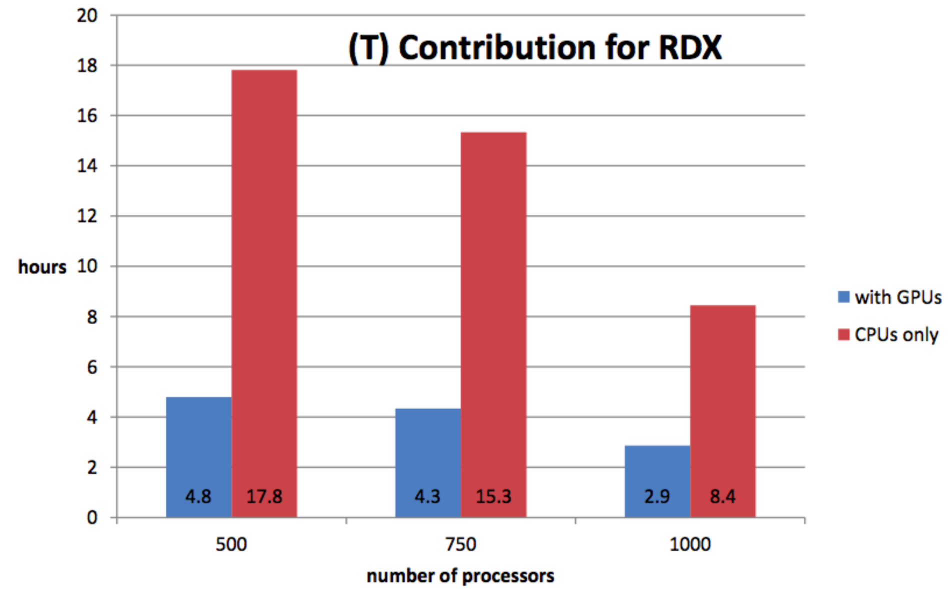
XK jobs as of end of September



- By looking at aprun instances and not job node count we can see when workloads are many single nodes bundled in a larger job.
- A large number of >3,000 node apuns.

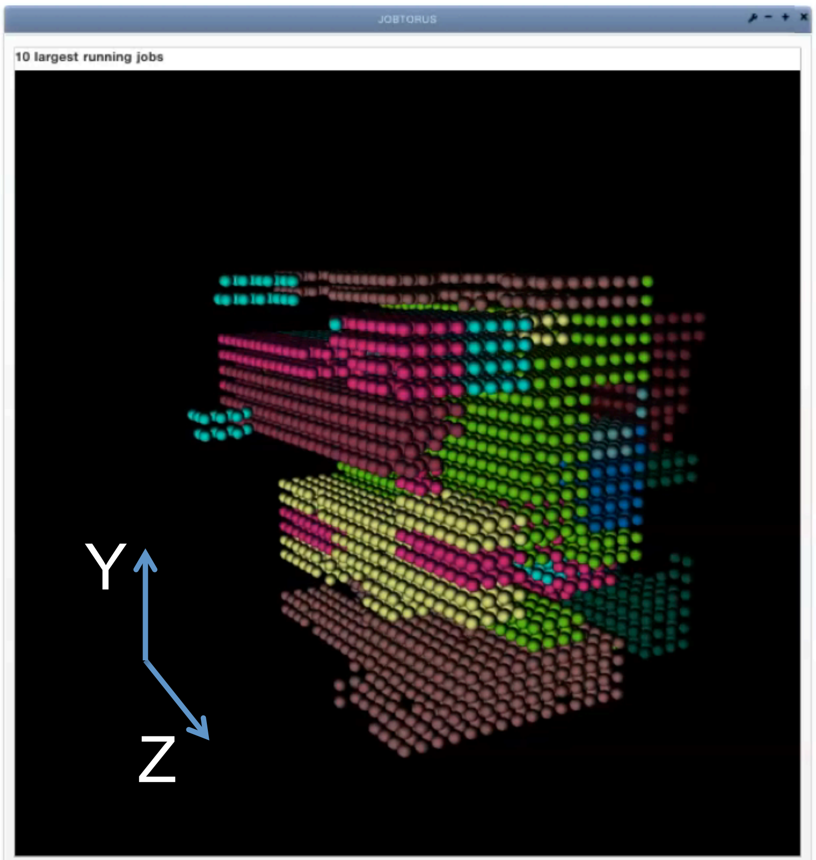
XK use scenarios

- Adoption of GPU
  - SIAL (ACESIII) – user annotation (SIAL directives) to assist CUDA code generator to get best speed-up. (T) – triples from CCSD(T).
  - https://bluewaters.ncsa.illinois.edu/web/portal/symposium-may-2013



- NEMO – PETSc + MAGMA to utilize GPU. Working on issues with sparse matrices and developing load balancing strategy across GPU and host CPU.
- https://bluewaters.ncsa.illinois.edu/web/portal/symposium-may-2013

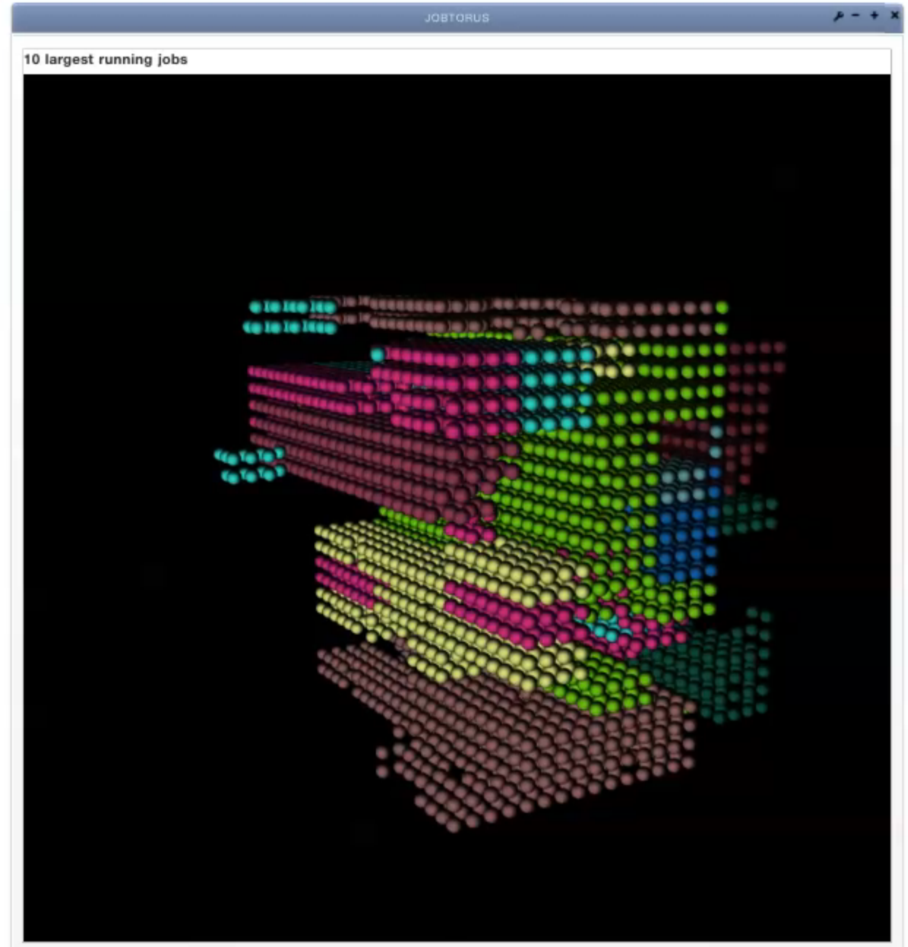
TorusView of 10 largest running jobs



- Relatively compact allocations.
- Some scattered clustering.
- Lots of concave shapes.
- Not showing all the small jobs filling in the rest of the torus.

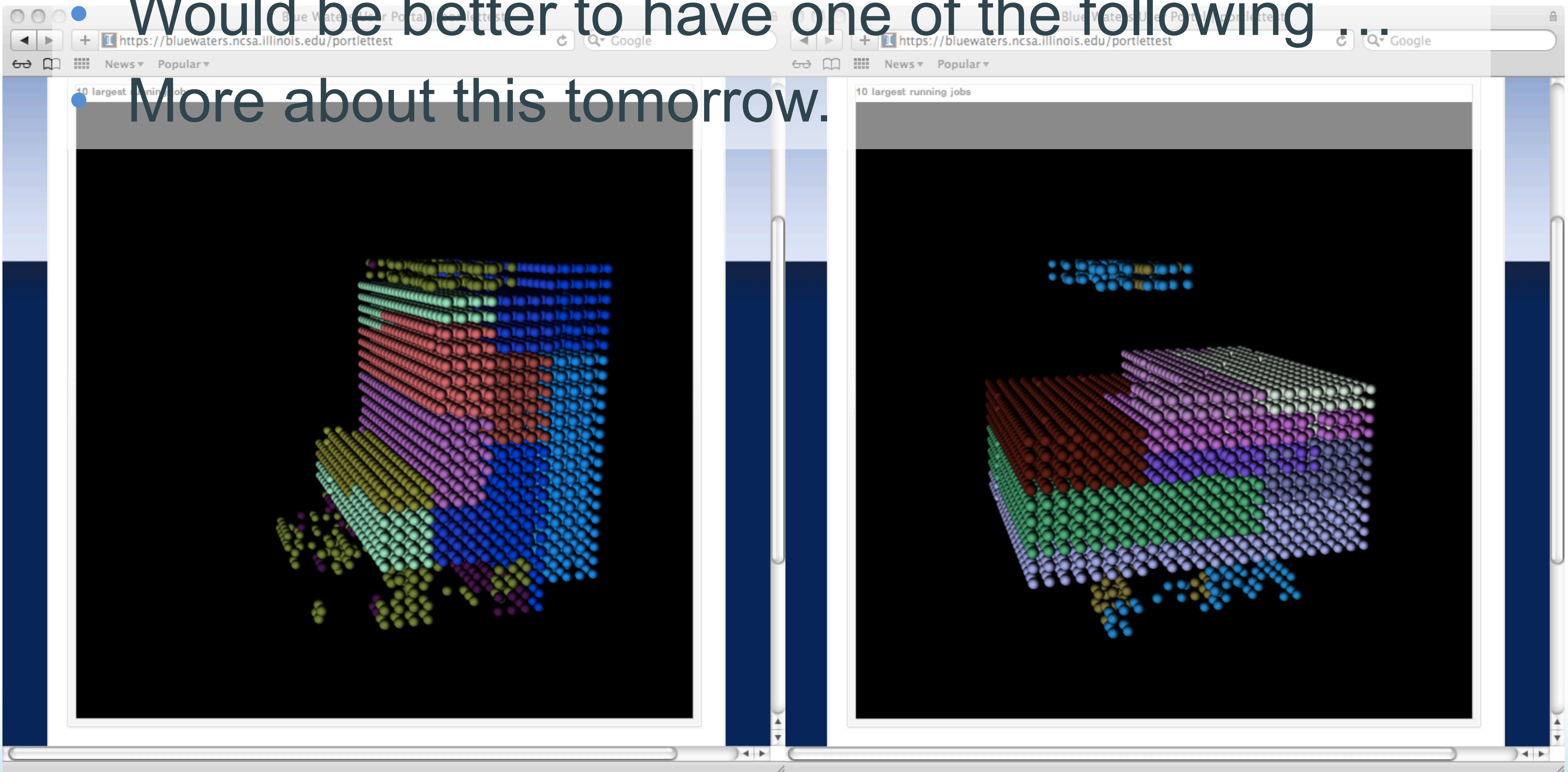
TorusView of 10 largest running jobs

- Allocations shift planes as the end of the Z direction is hit.
- Voids where larger job allocations wrap around smaller ones.



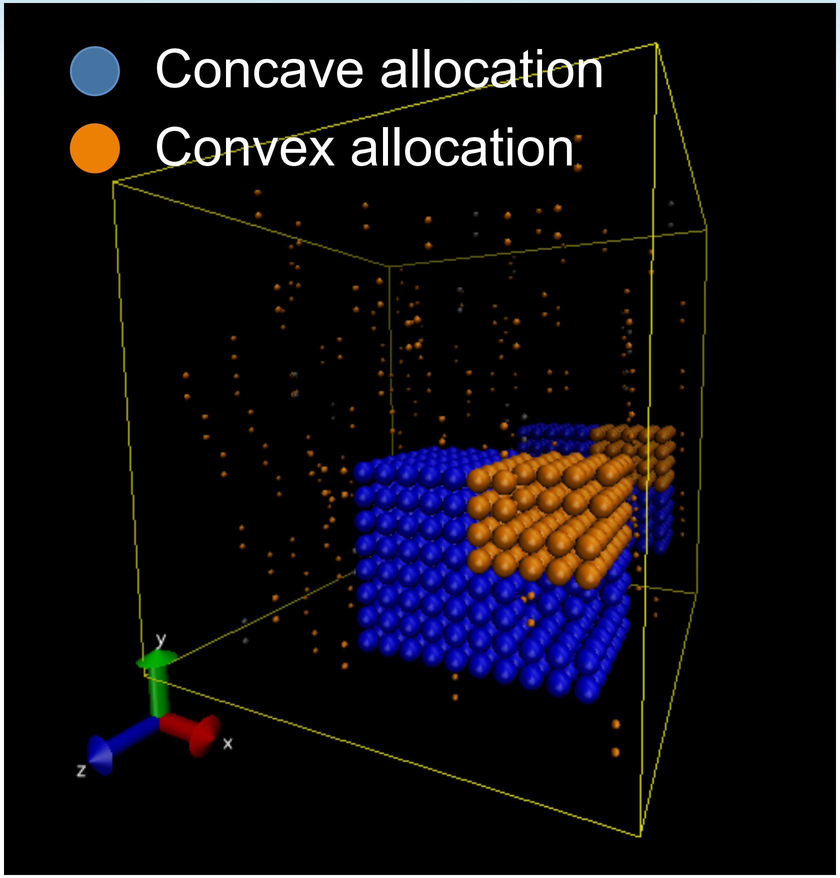
Better nid allocation

- Would be better to have one of the following …
- More about this tomorrow.



Impact of nid allocation

- Job – Job interaction
  - Analysis of key application communication intensity and sensitivity
  - 20% slowdown typical, 100% or more possible.

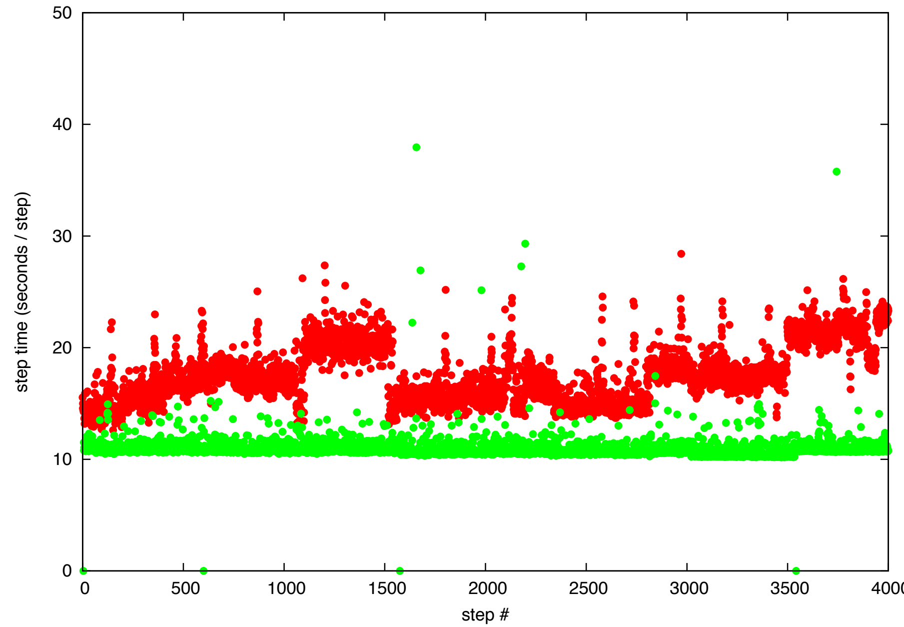


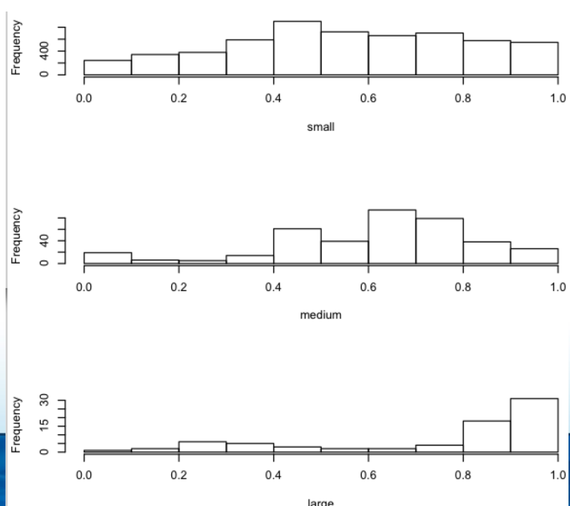
Communication	MILC	NAMD	NWCHEM	PSDNS	WRF
Intensive	2	2	3	2	1
Sensitive	2	3	1	2	1

1 – low 3 – high
as viewed by convex app.

Impact of poor nid allocation - Consistency

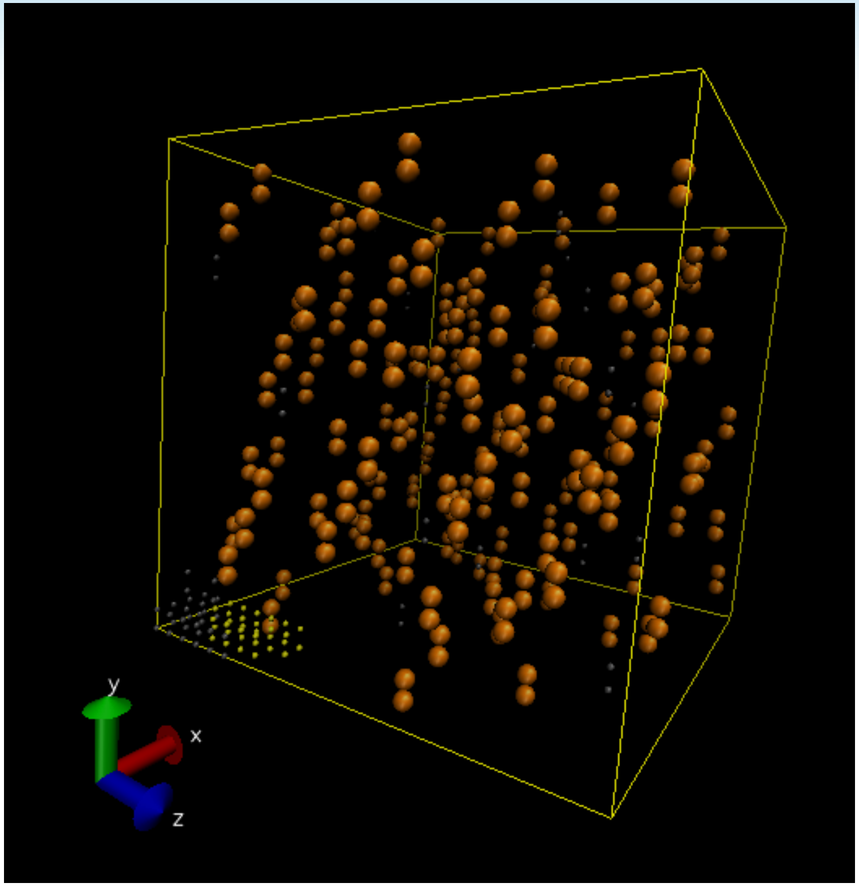
- Two jobs (8,192 nodes) with nearly same nid allocation (s10_8972n). Red job affected by other workload communicating through the region.





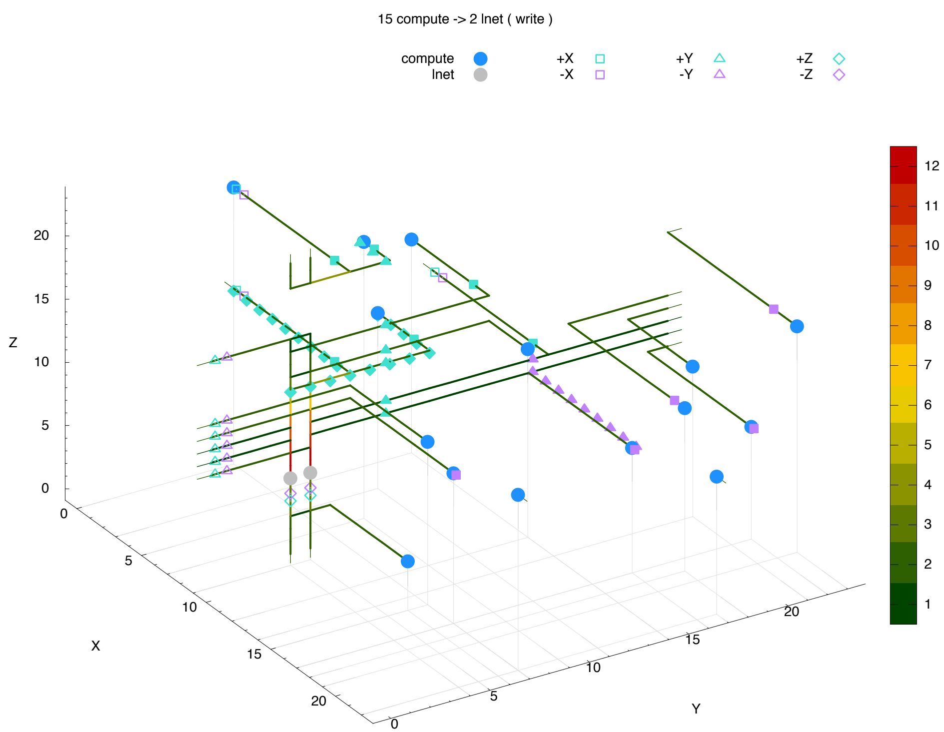
- Run time variation - poor wallclock accuracy (padding wallclock).

IO



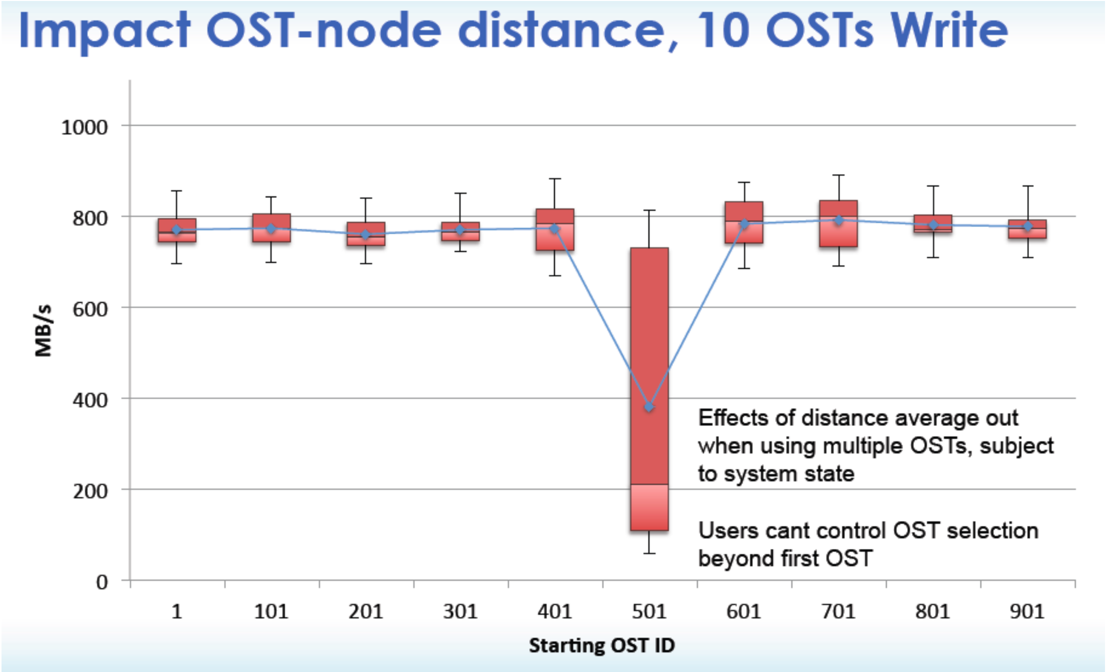
- LNETs scattered across the torus (orange colored geminis).
- Specific OSTs served by specific LNETs (not a full fat tree for the IB between OSTs and LNETs).
- IO is “topology sensitive”.

Routing of IO write



- 15 compute geminis (•) (30 nodes) writing to files served by a LNET pair (•).
- Color scale is the number of convergent routes on the link.

“Topology” aware IO library



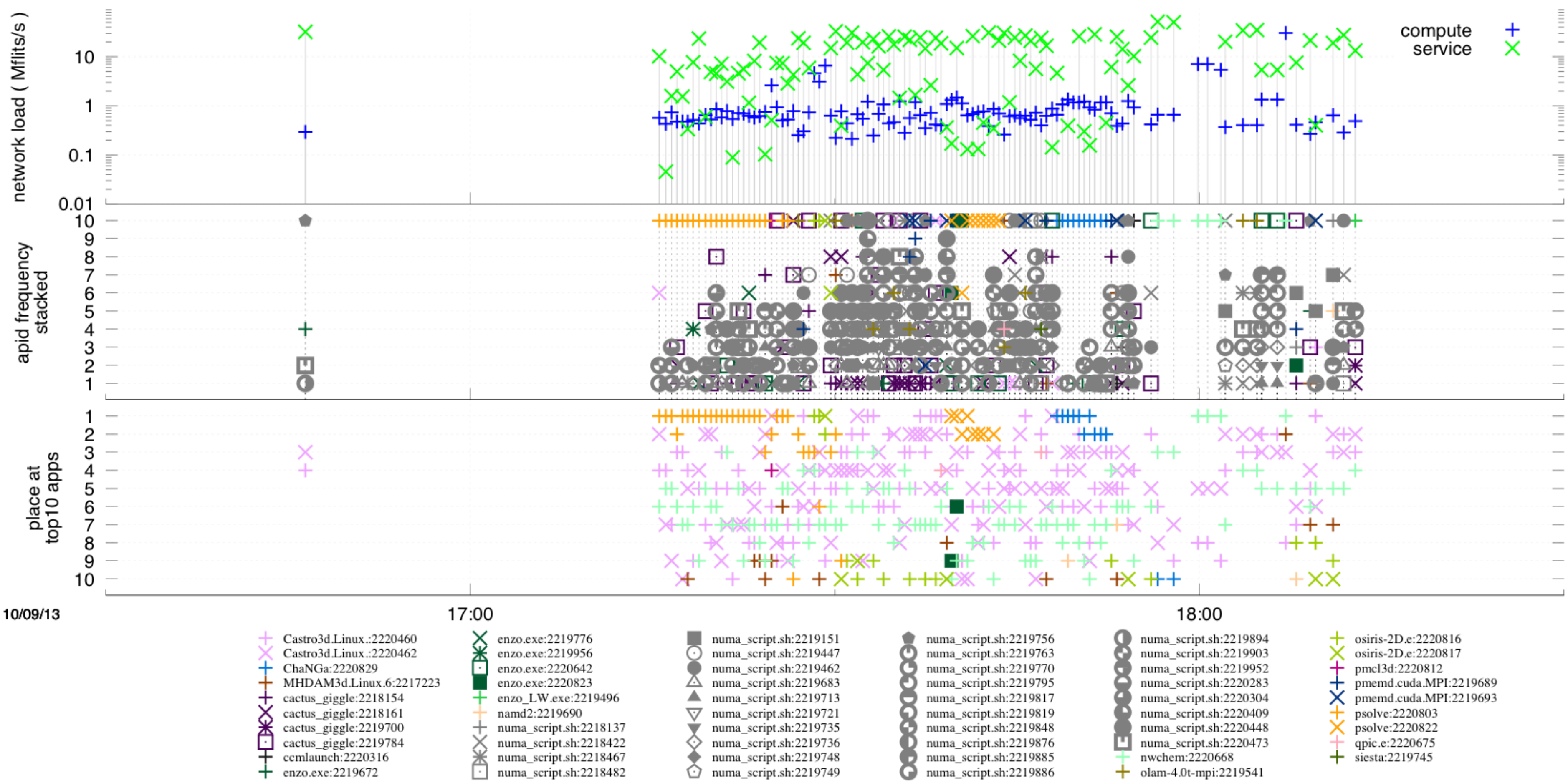
- Analysis of the Blue Waters File System Architecture for Application IO Performance - CUG 2013, May 6, 2013 Authors: Kalyana Chadalavada, Rob Sisneros

Congestion Protection

- To avoid data loss, traffic injection is throttled for a period of time, when reaching a point where forward progress is stalling. Throttling is applied and removed until congestion is cleared.
- System monitors percentage of time that traffic trying to enter the network from the nodes and percentage of time network tiles are stalled.
- Fortunately not a common occurrence. It does happen, typically in bursts.
- Can happen with node-node (MPI, PGAS) or node-LNET (IO) traffic.
- Many-to-one and long-path patterns.
- Libraries and user can control node injection as a precaution.
- In CP reports, flit rates represent data arriving at the node from the interconnection network.

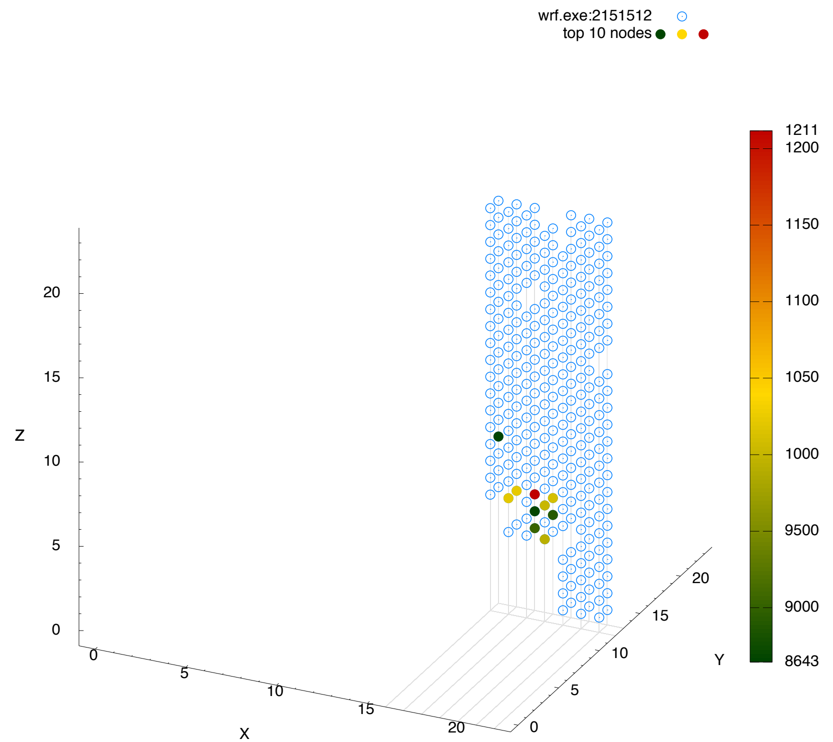
```
 Max
APID    Name                        Nodes   Flits/s   UID     Start       End
---------------------------------------------------------------------------------
2220460 Castro3d.Linux.              2048     31698   46466   16:00:45    19:41:40
2220462 Castro3d.Linux.              2048     81115   46466   16:01:05    19:37:03
2218386 namd2                        2000       --    43448   01:58:31    18:02:09
2220803 psolve                       2000     45732   47252   17:12:34    17:30:30
2218759 su3_rhmd_hisq_q              1536       --    12940   07:29:16
2219859 nwchem                       1000       --    32745   13:58:50    18:02:07
2220668 nwchem                       1000   4128749   32745   17:00:22    18:15:32
2219678 ks_spectrum_his               768       --    12940   11:30:04
2219512 namd2                         700       --    42864   10:35:55
…
…

=====================================================
Top Bandwidth Applications
=====================================================
0: apid 2218386 userid  43448 numnids  2000 apname           namd2 Kflits/sec: Total    3075
1: apid 2219859 userid  32745 numnids  1000 apname          nwchem Kflits/sec: Total    2743
2: apid 2220462 userid  46466 numnids  2048 apname  Castro3d.Linux. Kflits/sec: Total    2715
3: apid 2220460 userid  46466 numnids  2048 apname  Castro3d.Linux. Kflits/sec: Total    2691
4: apid 2219517 userid  42864 numnids   700 apname           namd2 Kflits/sec: Total    2271
5: apid 2219519 userid  42864 numnids   700 apname           namd2 Kflits/sec: Total    2073
6: apid 2218759 userid  12940 numnids  1536 apname  su3_rhmd_hisq_q Kflits/sec: Total    2071
7: apid 2219514 userid  42864 numnids   700 apname           namd2 Kflits/sec: Total    1762
8: apid 2220646 userid  12940 numnids   512 apname  ks_spectrum_his Kflits/sec: Total    1596
9: apid 2217219 userid  47296 numnids   500 apname          python Kflits/sec: Total    1389
…
=====================================================
Congestion Candidate COMPUTE Nodes
=====================================================
1: c17-0c1s0n1 (64051 flits/sec) (nid 18401; apid 2220473 userid 14394 numnids 32 apname numa_script.sh)
2: c9-0c0s1n0 (61950 flits/sec) (nid 23036; apid 2219894 userid 14394 numnids 32 apname numa_script.sh)
3: c10-1c0s3n2 (24438 flits/sec) (nid 5798; apid 2219756 userid 14394 numnids 32 apname numa_script.sh)
4: c3-10c0s5n1 (24238 flits/sec) (nid 25867; apid 2219672 userid 35077 numnids 64 apname enzo.exe)
5: c12-1c0s2n2 (22544 flits/sec) (nid 8026; apid 2219756 userid 14394 numnids 32 apname numa_script.sh)
6: c5-10c0s6n3 (20193 flits/sec) (nid 24813; apid 2219672 userid 35077 numnids 64 apname enzo.exe)
7: c12-1c0s2n0 (20161 flits/sec) (nid 8004; apid 2219756 userid 14394 numnids 32 apname numa_script.sh)
8: c14-1c0s3n0 (19784 flits/sec) (nid 8120; apid 2219756 userid 14394 numnids 32 apname numa_script.sh)
9: c10-1c0s2n1 (19273 flits/sec) (nid 5819; apid 2219756 userid 14394 numnids 32 apname numa_script.sh)
10: c10-1c0s3n0 (17453 flits/sec) (nid 5816; apid 2219756 userid 14394 numnids 32 apname numa_script.sh)

=====================================================
Top 100 Congestion Candidate Nodes (614 compute nodes: 134938785 flits/s, 590 service nodes: 1257373796 flits/
s)
=====================================================
1: c20-10c0s3n0 4128749 flits/sec nid 12038; apid 2220668 userid 32745 numnids 1000 apname nwchem
2: c20-10c0s3n3 3396088 flits/sec nid 12057; apid 2220668 userid 32745 numnids 1000 apname nwchem
3: c21-11c1s1n2 3351520 flits/sec nid 15484; apid 2220668 userid 32745 numnids 1000 apname nwchem
4: c17-10c0s3n2 3233871 flits/sec nid 17894; apid 2220668 userid 32745 numnids 1000 apname nwchem
5: c21-11c1s1n3 2912123 flits/sec nid 15485; apid 2220668 userid 32745 numnids 1000 apname nwchem
6: c20-10c1s1n3 2739003 flits/sec nid 12067; apid 2220668 userid 32745 numnids 1000 apname nwchem
7: c20-10c1s1n2 2727704 flits/sec nid 12066; apid 2220668 userid 32745 numnids 1000 apname nwchem
8: c21-11c1s2n0 2629574 flits/sec nid 15524; apid 2220668 userid 32745 numnids 1000 apname nwchem
9: c15-11c1s4n0 2619990 flits/sec nid 19030; apid 2220668 userid 32745 numnids 1000 apname nwchem
10: c21-11c1s2n3 2604278 flits/sec nid 15483; apid 2220668 userid 32745 numnids 1000 apname nwchem
```

Congestion Protection Burst



Congestion Protection Analysis

- Look at application to node relation.
- wrf listed as top application and the top 10 nodes are wrf nodes.
- nwchem running at same time (listed #4).
- The OVIS state of the network data should help here.



Interrupts

- We provide to the user a checkpoint interval calculator based on the work of J. Daly, using recent node and system interrupt data.
- September data
  - 22,640 XE nodes MTTI ~ 14 hrs.
  - 4,224 XK nodes MTTI ~ 32 hrs.
  - System interrupts MTTI ~ 100 hrs.
- Checkpoint intervals on the order of 4 – 6 hrs. at full system (depending on time to write checkpoint).