ILLINOIS
CSL | Coordinated Science Lab
COLLEGE OF ENGINEERING

Saurabh Jha[1], Archit Patke[1], Jim Brandt[2], Ann Gentile[2], Benjamin Lim[1], Mike Showerman[1,3], Greg Bauer[1,3], Larry Kaplan[4], Zbigniew T. Kalbarczyk[1] and William T. Kramer[1,3], Ravishankar K. Iyer[1,3]

[1]University of Illinois at Urbana-Champaign [2]Sandia National Laboratories

[3]National Center for Supercomputing Applications [4]Cray Inc.

Measuring Congestion in High-Performance Datacenter Interconnects

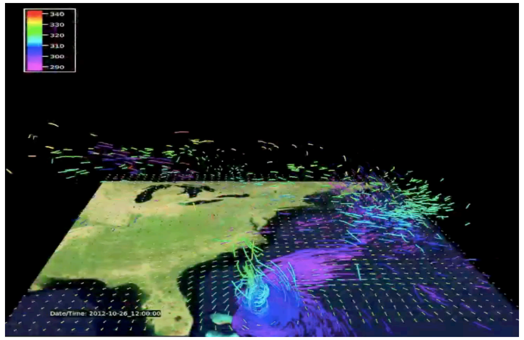
NCSA

Sandia National Laboratories

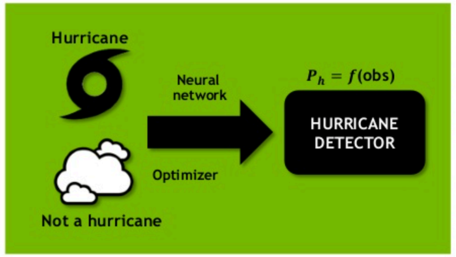
CRAY

High-Performance Computing (HPC)

HPC solves critical science, finance, AI, and other problems



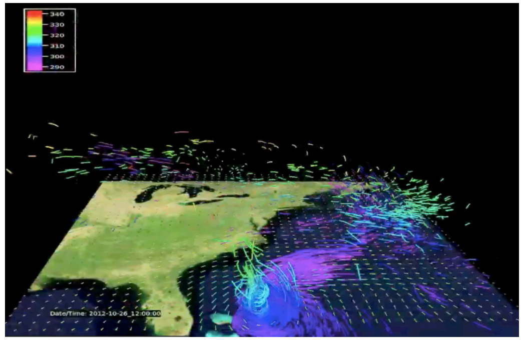
WRF: Largest Weather forecast simulation

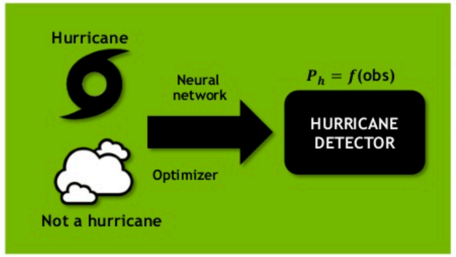


Hurricane detector using AI
(Courtesy: Nvidia)

High-Performance Computing (HPC)

HPC solves critical science, finance, AI, and other problems

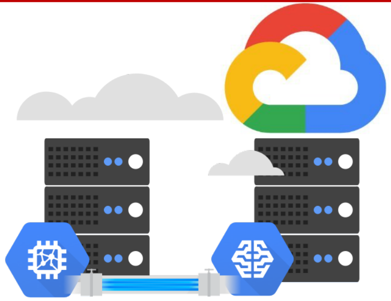


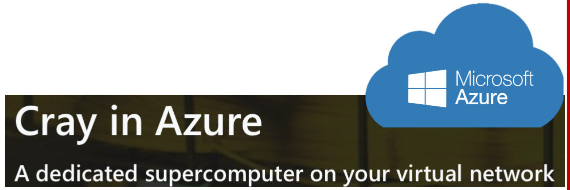


WRF: Largest Weather forecast simulation

Hurricane detector using AI
(Courtesy: Nvidia)

HPC on Cloud





HPC in Academic and National Labs

TOP500 SUPERCOMPUTER SITES

BLUE WATERS SUSTAINED PETASCALE COMPUTING

NCSA (UIUC)

summit

Oakridge National Lab

High-Performance Computing (HPC)

High-speed Networks (HSN)

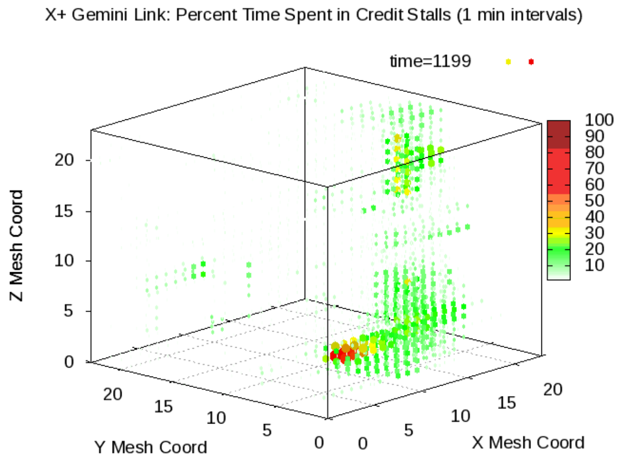
- Low per-hop latency [1][2]
- Low tail-latency variation
- High bisection bandwidth

[1] https://www.nextplatform.com/2018/03/27/in-modern-datacenters-the-latency-tail-wags-the-network-dog/
[2] https://blog.mellanox.com/2017/05/microsoft-enhanced-azure-cloud-efficiency/

Networking and Performance Variation

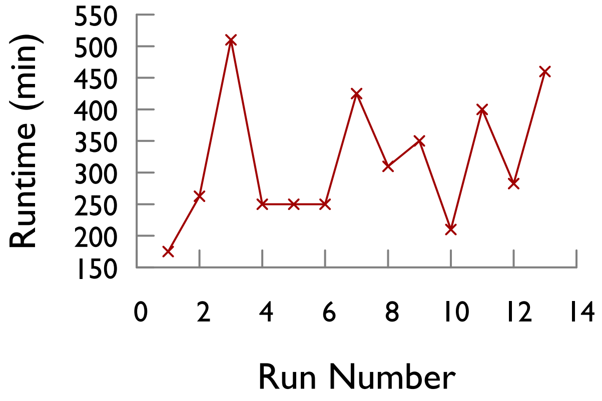
Despite the low-latency, high-speed networks (HSN) are susceptible to high congestion

Such congestion can cause up to 2-4X application performance variation in production settings



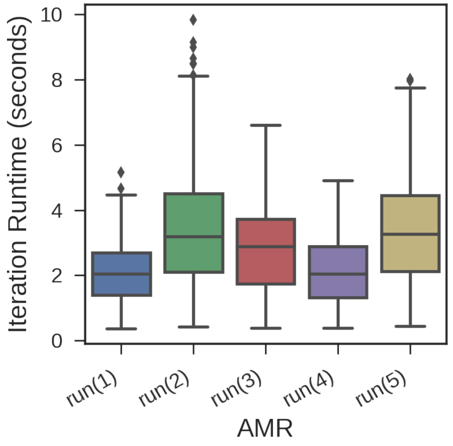
1000-node production molecular dynamics code.

Up to $1.89\times$ slowdown compared to median runtime of 282 minutes



256-node benchmark app (AMR)

Up to $4\times$ slowdown compared to the median loop iteration time of 2.5 sec



Networking and Performance Variation

Despite low-latency, high-speed networks (HSN) susceptible to high congestion

Such congestion can lead to up to 2-3X application performance variation in production settings

Questions:

- How often system/applications are experiencing congestion ? [Characterization]
- What are the culprits behind congestion? [Diagnostics]
- How to avoid and mitigate effects of congestion ? [Network and System Design]

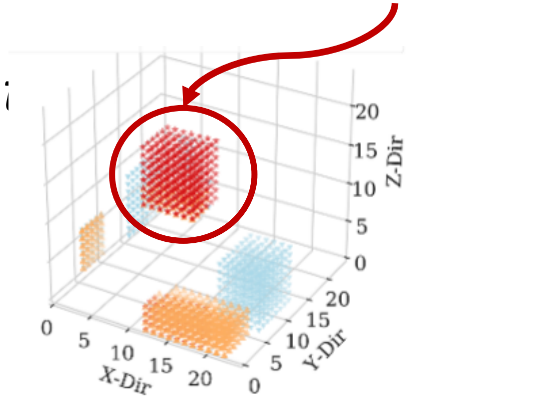
Highlights

- *Created data mining and ML-driven methodology and associated framework for*
  - Characterizing network design and congestion problems using empirical data
  - Identifying factors leading to the congestion on a live system
  - Checking if the application slowdown was indeed due to congestion

- *Empirical evaluation of a real-world large-scale supercomputer: Blue Waters at NCSA*
  - *Largest 3D Torus network in the world*
  - 5 months of operational data
  - 815,006 unique application runs
  - 70 PB of data injected into the network

- Largest dataset on congestion (first on HPC networks)
  - Dataset (51 downloads and counting!) and code released

Key Findings



- *HSN congestion is the biggest contributor to app performance variat*
  - *Continuous presence of high congestion regions*
  - *Long lived congestion (*may persist for >23 hours)

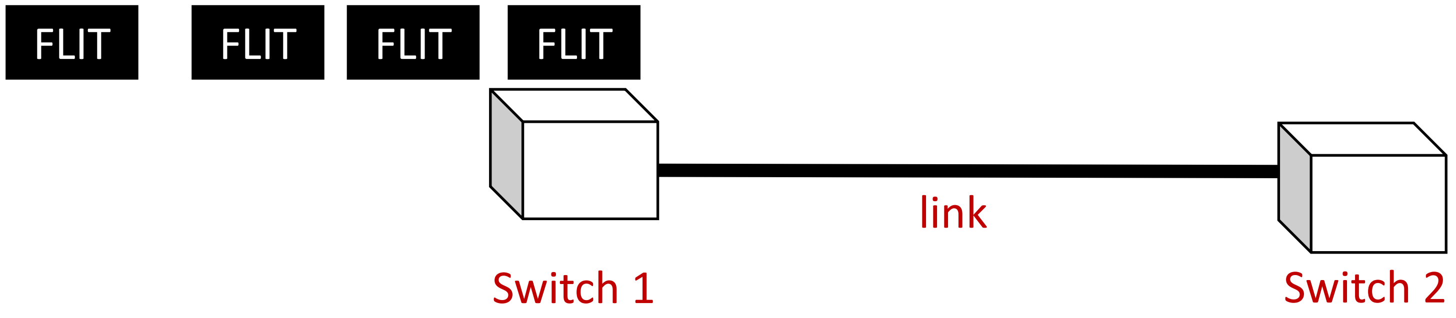
- *Default congestion mitigation mechanism have limited efficacy*
  - *Only 8 % (261 of 3390 cases) of high congestion cases found using our framework were detected and acted by default congestion mitigation algorithm*
  - *In ~30% of the cases the default congestion mitigation algorithm was unable to alleviate congestion*

- Congestion patterns and their tracking enables identification of culprits behind congestion
  - *critical to system and application performance improvements*
  - *E.g., intra-app congestion can be fixed by changing allocation and mapping strategies*

Congestion in credit-based flow control Network

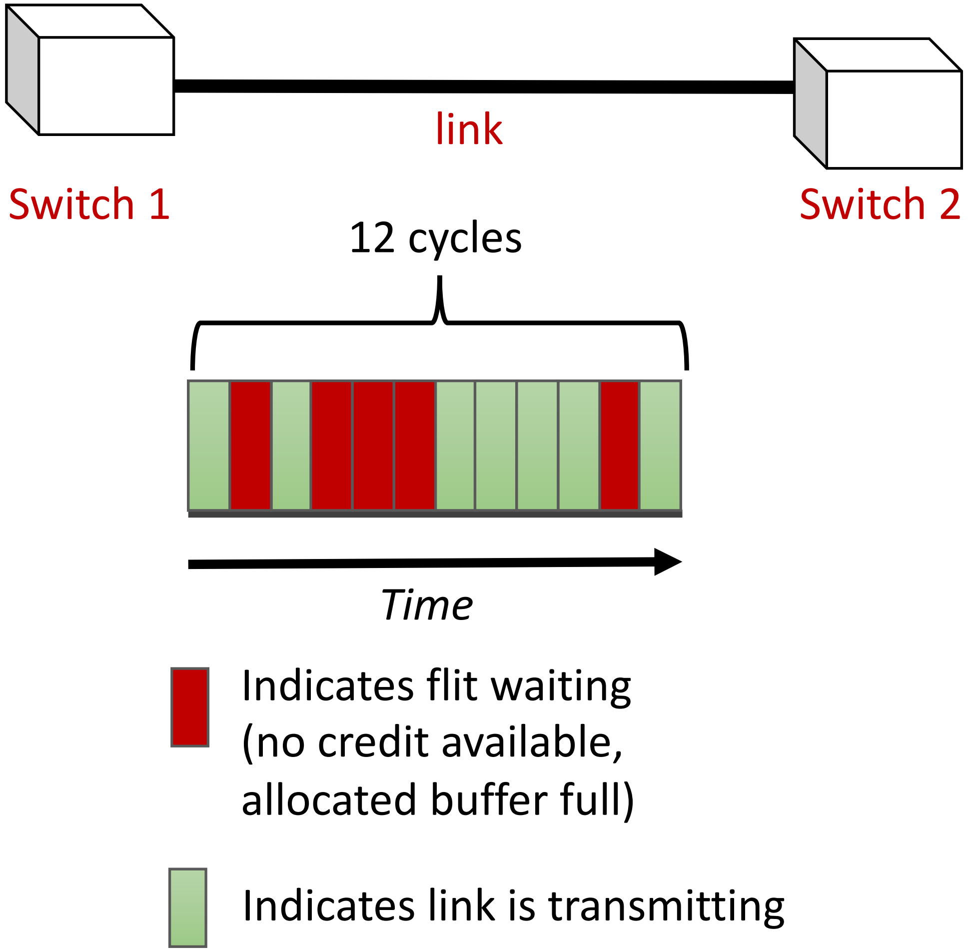
- Focus on evaluation of credit-based flow control transmission protocol
- Flit is the smallest unit of datum that can be transferred
- Flits are not dropped during congestion
- Backpressure (credits) provides congestion control

Available Credits: 0



If credit = 0, flit cannot be sent / **If credit > 0, flit can be sent**

Measuring Congestion



Congestion measured using **Percent time stalled** (P_{Ts})

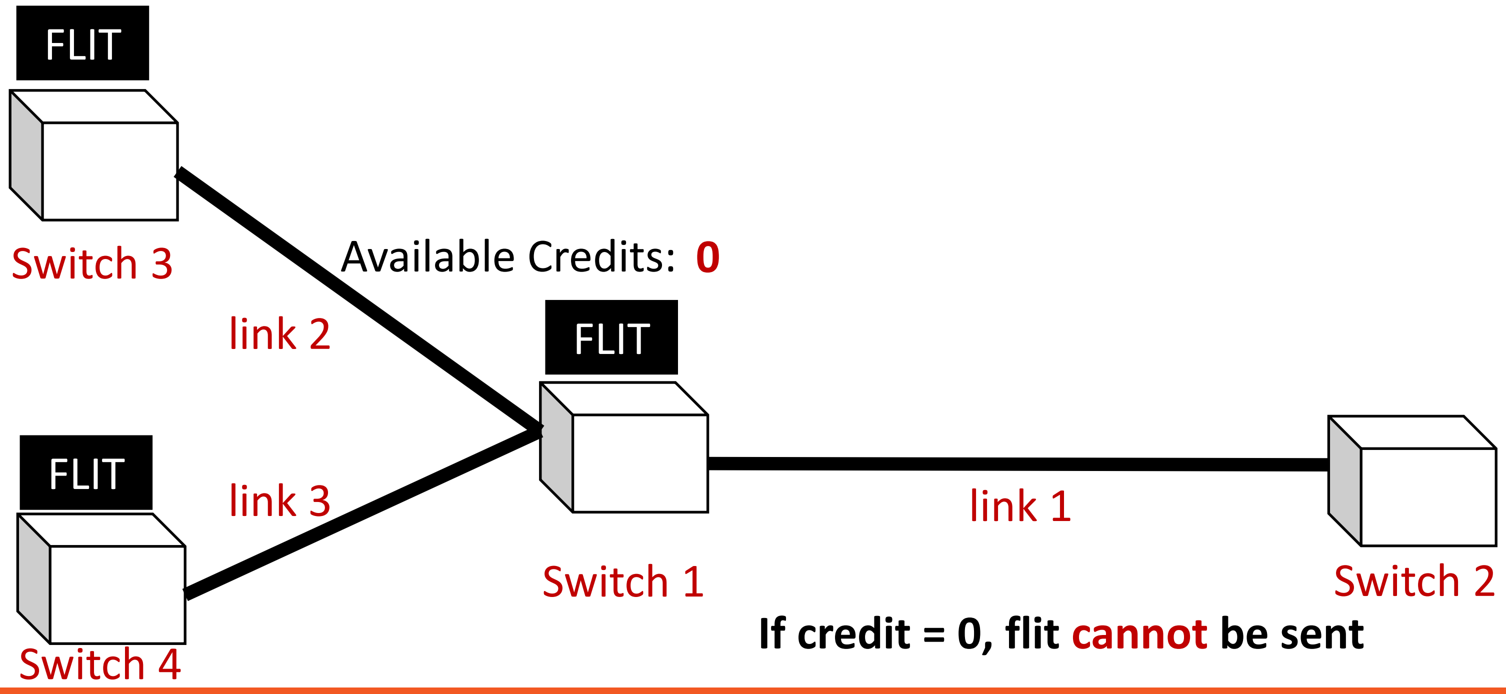
$$\boldsymbol{P^{i}_{Ts}} = \mathbf{100} \times \frac{\boldsymbol{T^{i}_{s}}}{\boldsymbol{T^{i}}} = 100 \times \frac{5}{12} = 41.67\ \%$$

$\boldsymbol{T^{i}}$: # network cycles in i^{th} measurement interval (fixed value)

$\boldsymbol{T^{i}_{s}}$: # total cycles the link was stalled in T^{i} (i.e., flit was ready to be sent but no credits available.)

Congestion in credit-based flow control Network

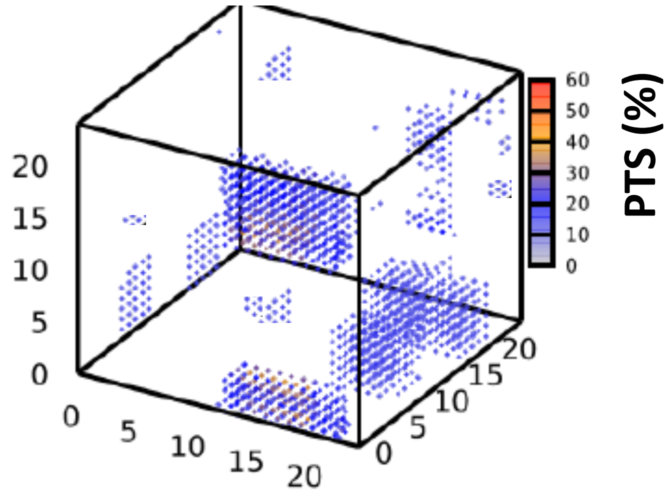
Insight: Congestion spreads locally (i.e., fans out from an origin point to other senders).



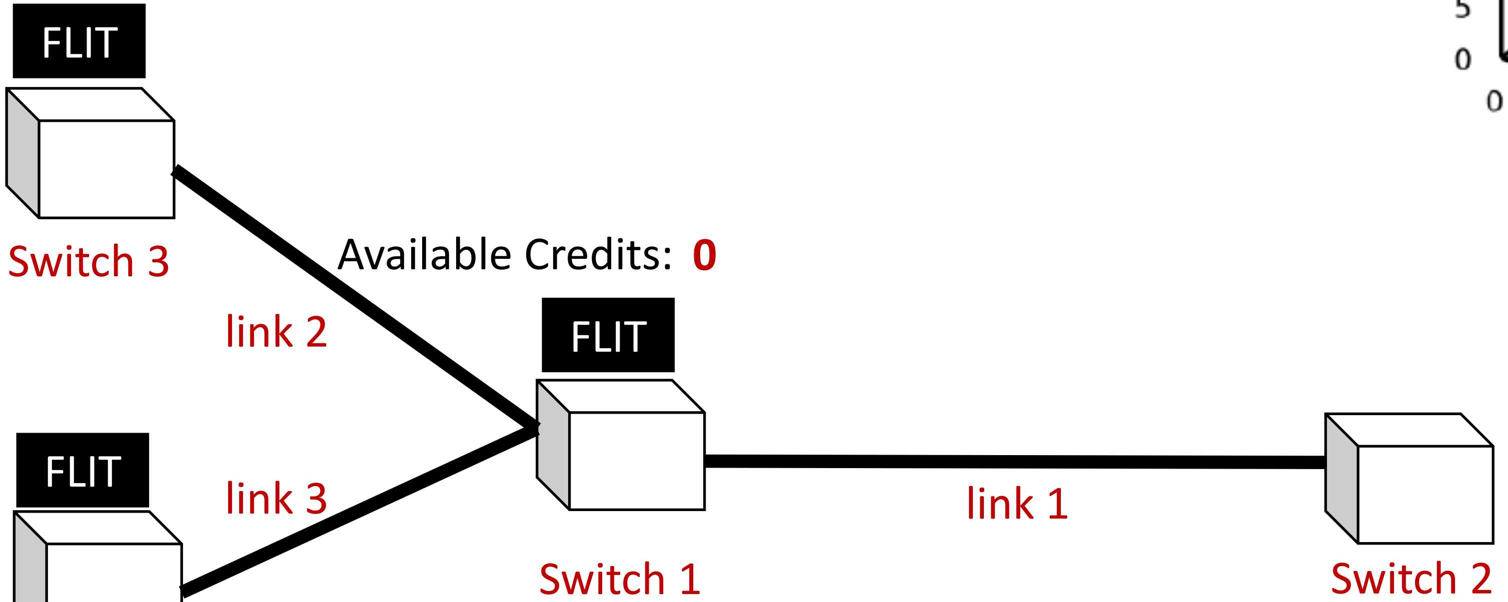
If credit = 0, flit cannot be sent

Congestion in credit-based flow control Network

Insight: Congestion spreads locally (i.e., fans out from an origin point to other senders).



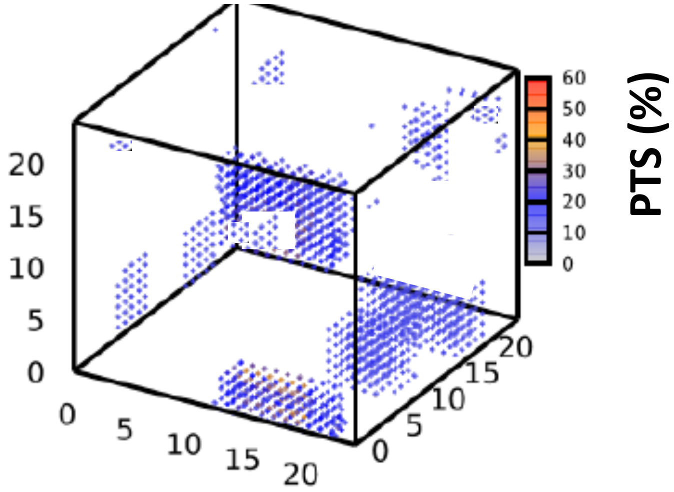
Congestion Visualization



If credit = 0, flit cannot be sent

New unit for measuring congestion

Measure congestion in terms of regions, their size and severity

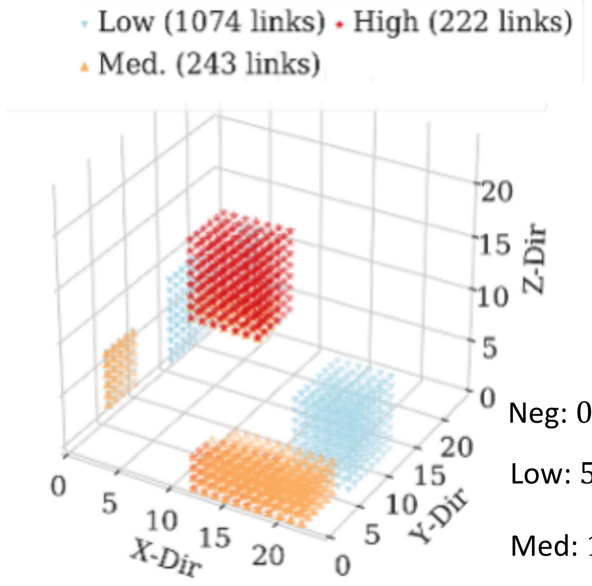


Raw Congestion Visualization

Unsupervised clustering

distance is small: $d_\delta(x,y) \leq \delta$

stall difference is small: $d_\lambda(x_s - y_s) \leq \theta p$



Neg: $0\% < P_{TS} \leq 5\%$

Low: $5\% < P_{TS} \leq 15\%$

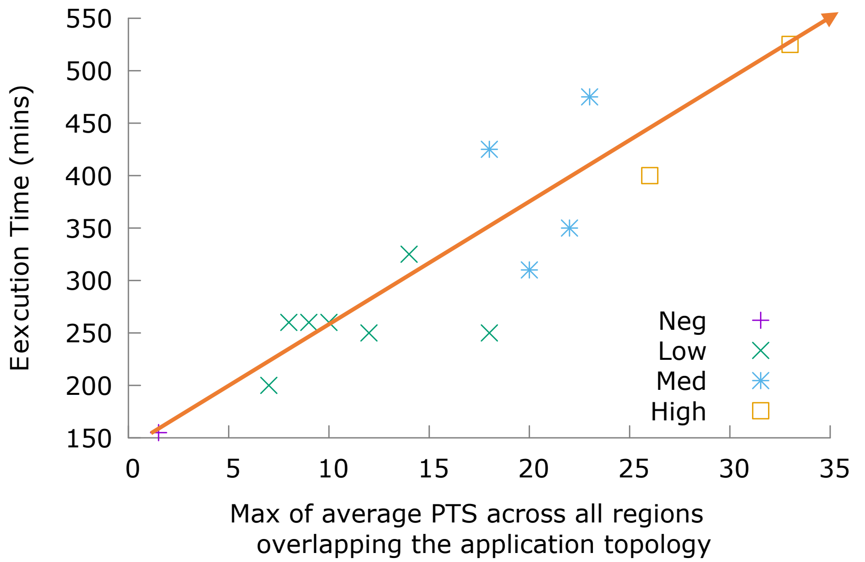
Med: $15\% < P_{TS} \leq 25\%$

High: $25\% < P_{TS}$

Congestion Regions Proxy for Performance Evaluation

Congestion Regions (CRs) captures relation between congestion severity and application slowdown and therefore can be used for live forensics and debugging!
(details in paper)

1000-node production molecular dynamics (NAMD) code.

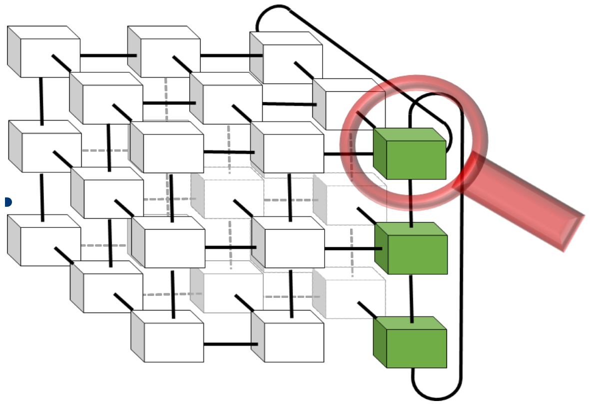


Low: $5\% < P_{Ts} \leq 15\%$

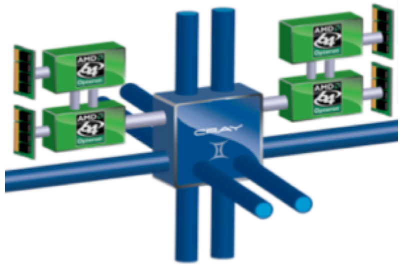
Med: $15\% < P_{Ts} \leq 25\%$

High: $25\% < P_{Ts}$

System, Monitors, and Datasets



Blue Waters Networks



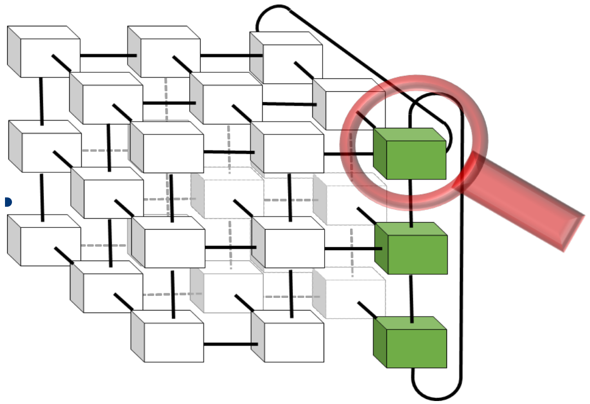
Cray Gemini Switch

Courtesy: Cray Inc. (HP)

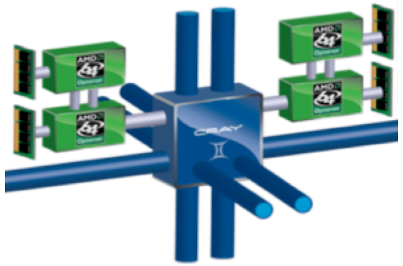
- **Topology:** 3D Torus (24x24x24)
- **Compute nodes :** 28K nodes
- **Avg. Bisection Bandwidth:** 17550 GB/sec
- **Per hop latency:** 105 ns [1]

[1] https://wiki.alcf.anl.gov/parts/images/2/2c/Gemini-whitepaper.pdf

System, Monitors, and Datasets



Blue Waters Networks



Cray Gemini Switch

Courtesy: Cray Inc. (HP)

- **Topology:** 3D Torus (24x24x24)
- **Compute nodes :** 28K nodes
- **Avg. Bisection Bandwidth:** 17550 GB/sec
- **Per hop latency:** 105 ns [1]

	Monitoring logs	Characterization (5 months)	Live Analytics (60 seconds)
Cray Network Monitors	**Network Failures**	**100 GB**	**~**
Lightweight Distributed Metric service (LDMS) [2]	**Performance counters**	**15 TB**	**40 MB**
Scheduler	**Workload**	**8 GB**	**55 MB**

[1] https://wiki.alcf.anl.gov/parts/images/2/2c/Gemini-whitepaper.pdf

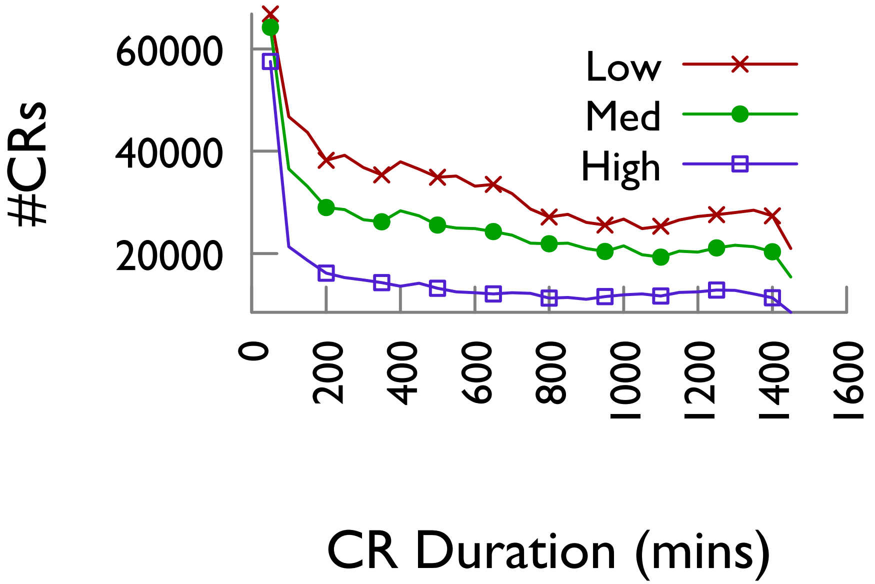
[2] A. Agelastos et al. Lightweight Distributed Metric Service: A Scalable Infrastructure for Continuous Monitoring of Large-scale Computing Systems and Applications. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 154–165, 2014.

1. Congestion is the biggest contributor to app performance variation

Long-lived congestion

- *Congestion Region* can persist up to ~24 hours
  (median: 9.7 hours)

- *Congestion Region* count decreases with increasing duration

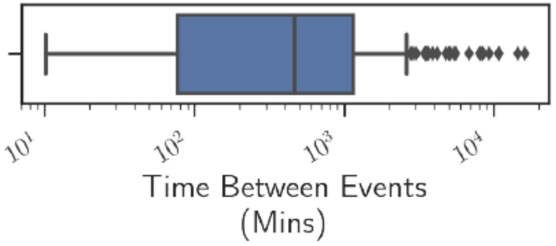


Low: $5\% < P_{Ts} \leq 15\%$
Med: $15\% < P_{Ts} \leq 25\%$
High: $25\% < P_{Ts}$

2. Limited efficacy of default congestion detection and mitigation mechanisms

- *#congestion mitigating triggered : 261*
- *Median time between events: 7 hours*
- *Failed to alleviate congestion in 29.8% cases*

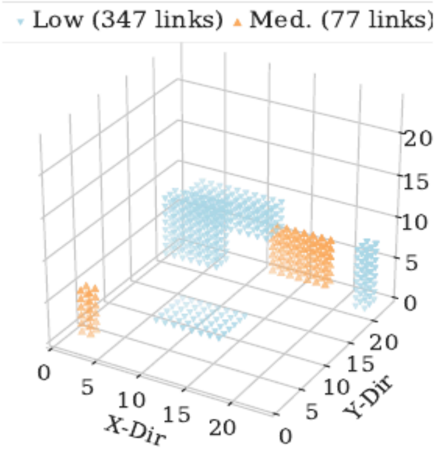
Median: 7 hours



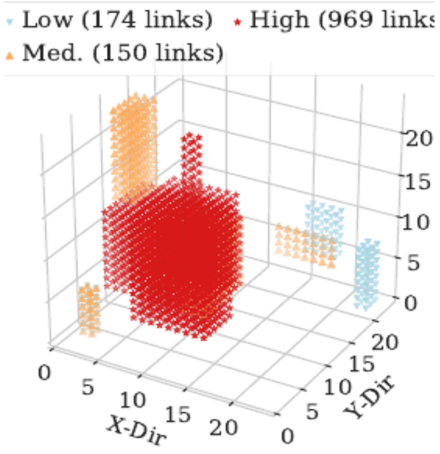
Default system congestion detection and mitigation

Default mitigation throttles all NICs such that

aggregate traffic injection bandwidth across all nodes < single node bandwidth ejection



Before congestion mitigation

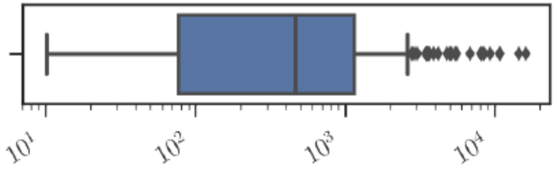


After congestion mitigation

2. Limited efficacy of default congestion detection and mitigation algorithms

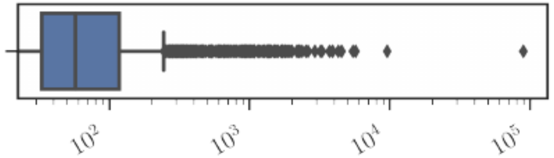
- *Default congestion mitigating triggered : 261*
- *Median time between events: 7 hours*
- *Failed to alleviate congestion in 29.8% of the cases*

Median: 7 hours, #events: 261



Default system congestion detection and mitigation

Median: 58 minutes, #events: 3390

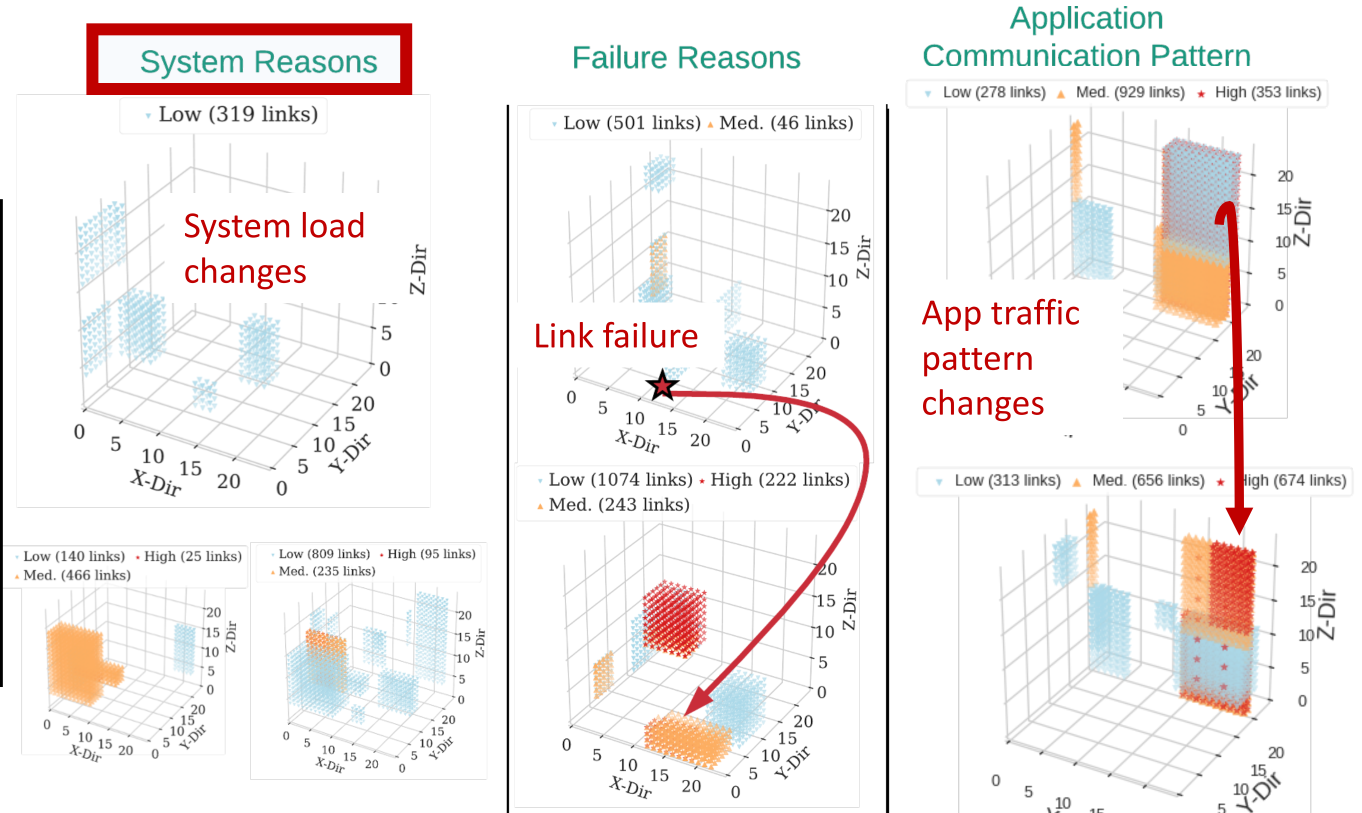


Monet detection

Only 8 % (261 of 3390 cases) of high congestion cases found by Monet were detected and acted by default congestion mitigation algorithm

3. Congestion patterns and their tracking enables identification of culprits behind congestion

- Network design and congestion-aware scheduling
- E.g., topology-aware scheduling [1] improved system throughput by 56% by tuning resource allocation strategies



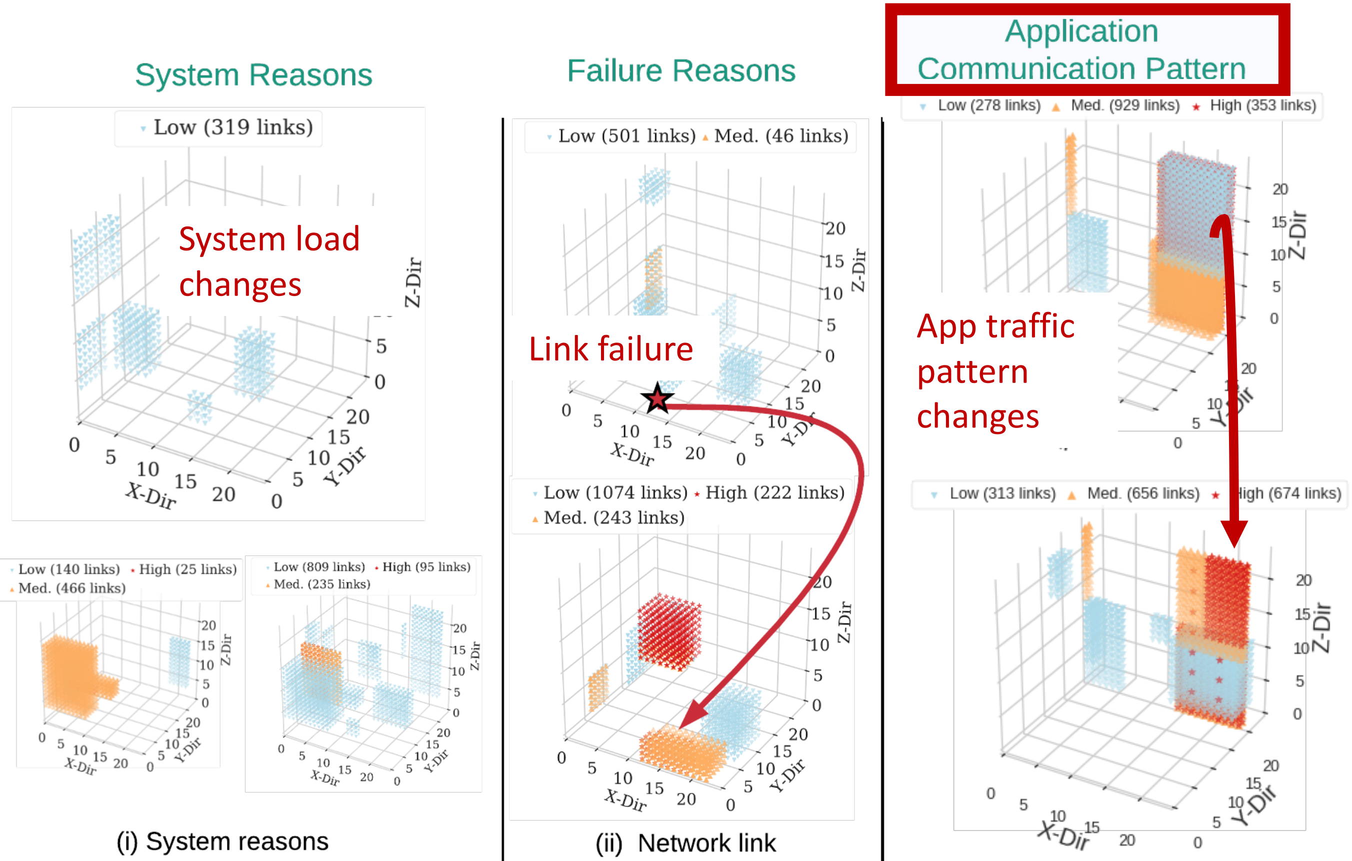
(i) System reasons such as system load

(ii) Network link Failure

[1] J Enos et al. Topology-aware job scheduling strategies for torus networks. In *Proc. Cray User Group*, 2014.

3. Congestion patterns and their tracking enables identification of culprits behind congestion

- Node mapping within the allocation reduces intra-app congestion
- E.g., TopoMapping [2] for finding optimal process rank mapping for the allocated resource



(i) System reasons such as system load

(ii) Network link Failure

[2] Galvez et al. Automatic topology mapping of diverse large-scale parallel applications. In *Proceedings of the International Conference on Supercomputing*, ICS ’17, pages 17:1–17:10, New York, NY, USA, 2017. ACM.

Conclusion

- Developed and validated the proposed methodology on production datasets

- Code and dataset online (51 downloads and counting!)
  - https://databank.illinois.edu/datasets/IDB-2921318
  - https://github.com/CSLDepend/monet

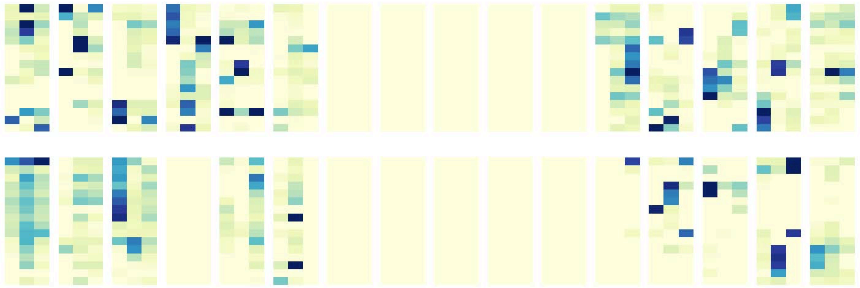
Future Work

Congestion avoidance and mitigation is an ongoing problem !

Developing workload-aware high-speed networks
- **Inferring and meeting application demands**
- **Optimizing congestion control and routing strategies**

Meet us at the poster session!

Wednesday 6:30 PM - 8:00 PM
Cypress Room



Congestion Visualization on a production Cray Aries (DragonFly Network)

Questions?