Characterizing and Improving Power and Performance in HPC Networks

Doctoral Showcase by Taylor Groves

Advised by Dorian Arnold

Department of Computer Science

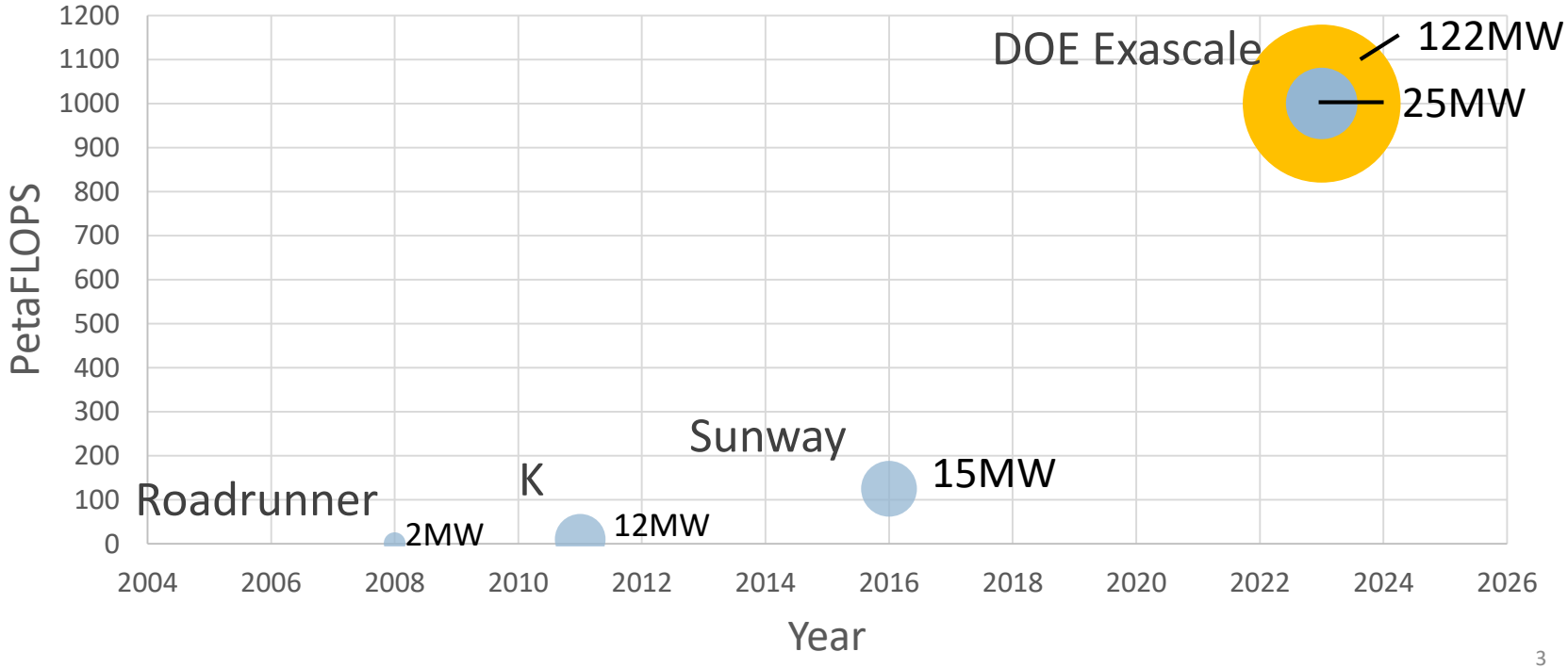
Quick Bio

- B.S. C.S. 2009 – Texas State University
- M.S. C.S. 2012 – University of New Mexico
- Ph.D. CS 2016 – University of New Mexico (Looking for job)
- Intern 2014-2016 @ Sandia National Labs

Obligatory Picture of Family

The Exascale Challenge

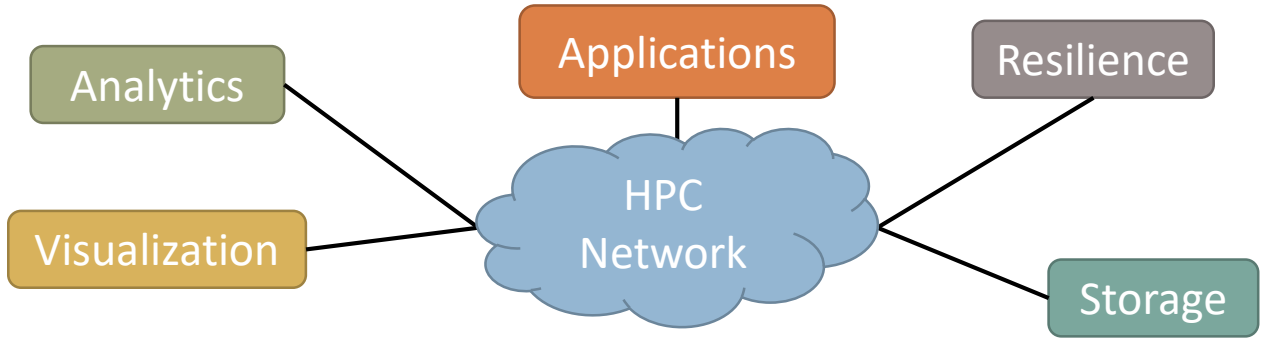
- The Department of Energy wants to hit Exascale (10^{18} flops) by 2025
  - Using 20-30 MW of power
- Compared to Sunway TaihuLight, **Increase performance by 8X**
  - With **only 1.3X – 2X increase in power**



Focus on the Network + Communication

Why focus on the Networks?

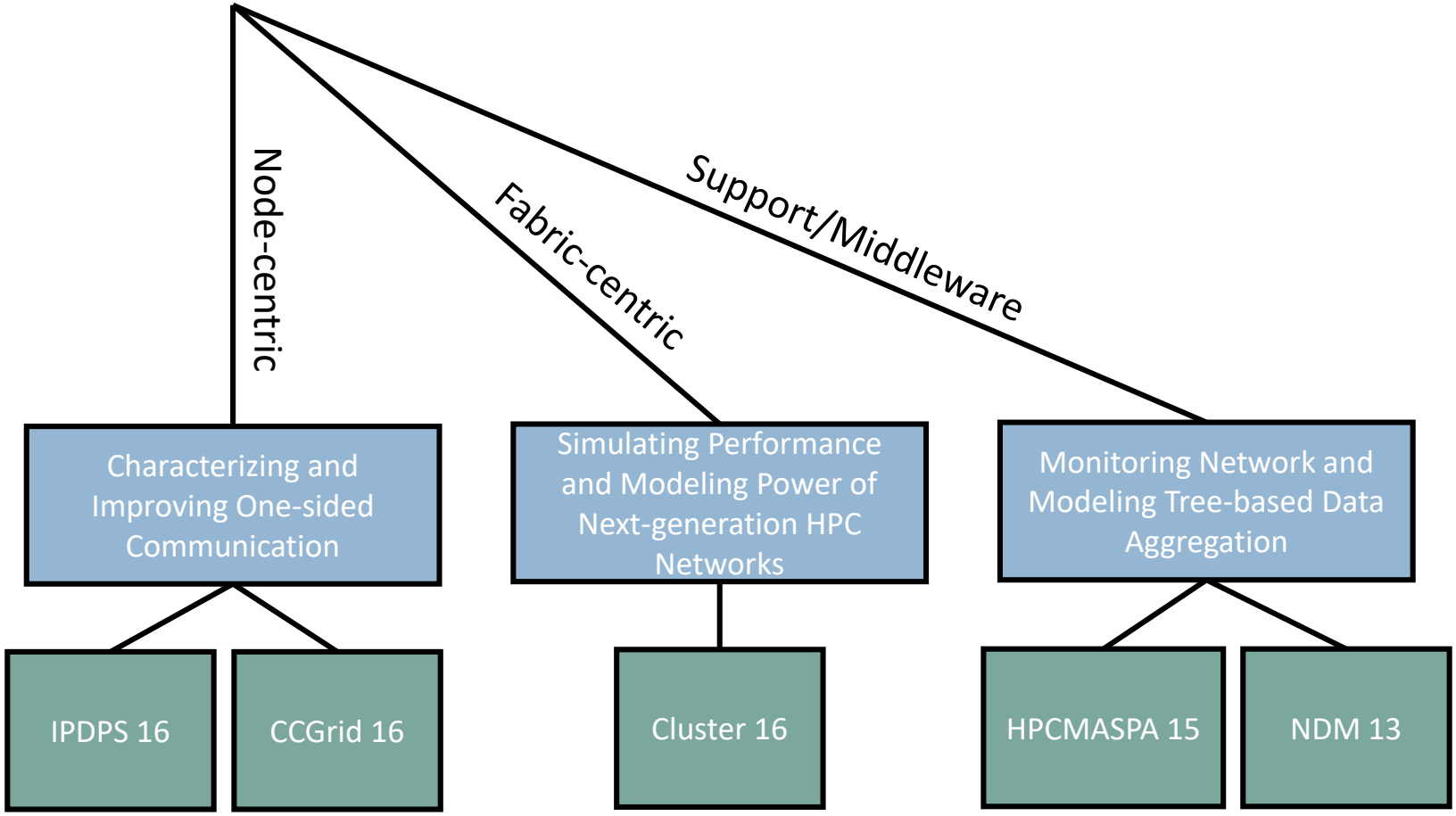
1. Networks are the central component that ties the system together
   - Far reaching performance impact



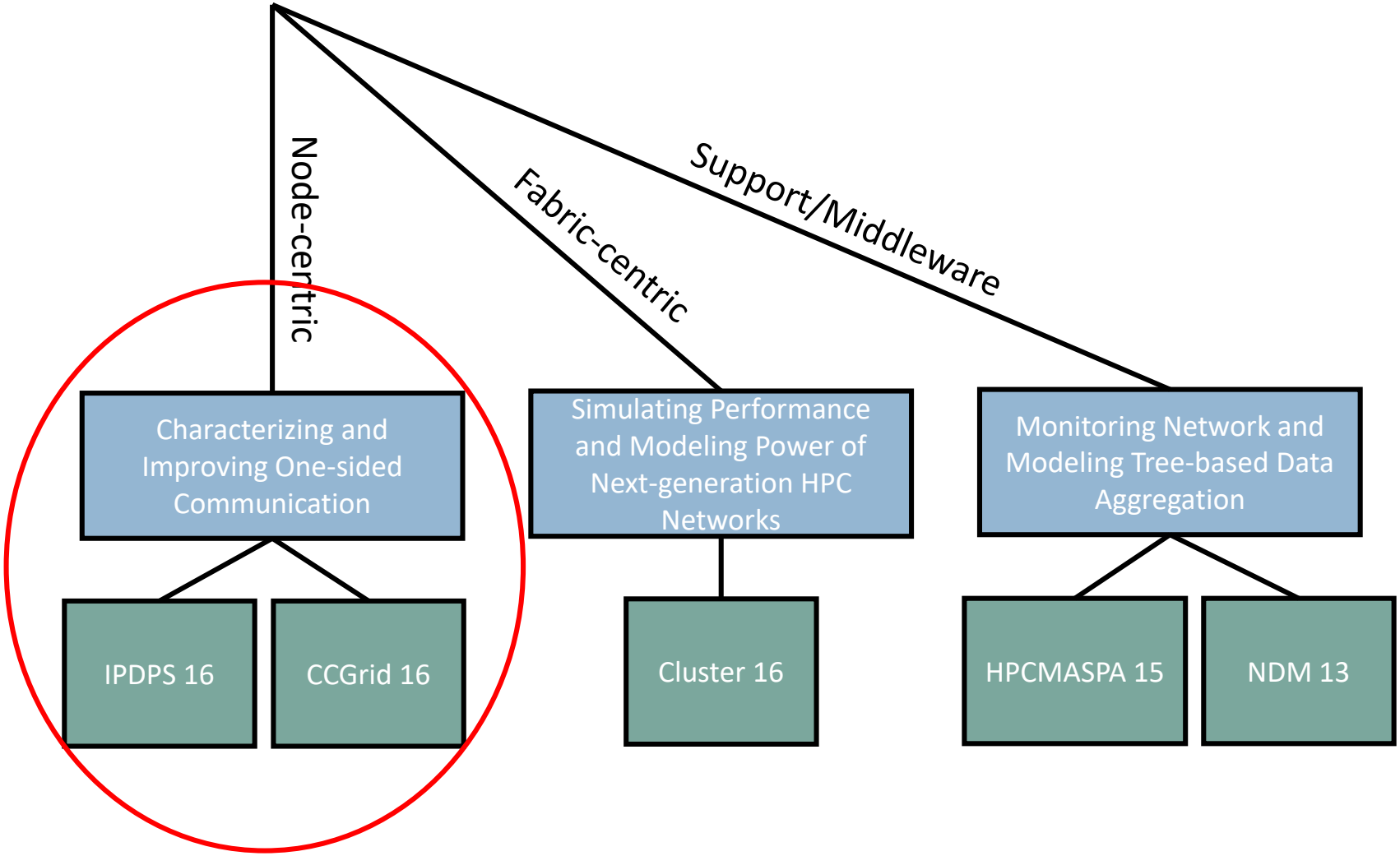
2. ~10% of Exascale Power Budget
   - And they provide opportunities for dynamic savings

3. We need to extract both power and performance at **every** opportunity to meet the Exascale challenge

Characterizing and Improving Power and Performance in HPC Networks



Characterizing and Improving Power and Performance in HPC Networks



Characterizing and Improving One-sided Communication (CCGrid 16, IPDPS 16)

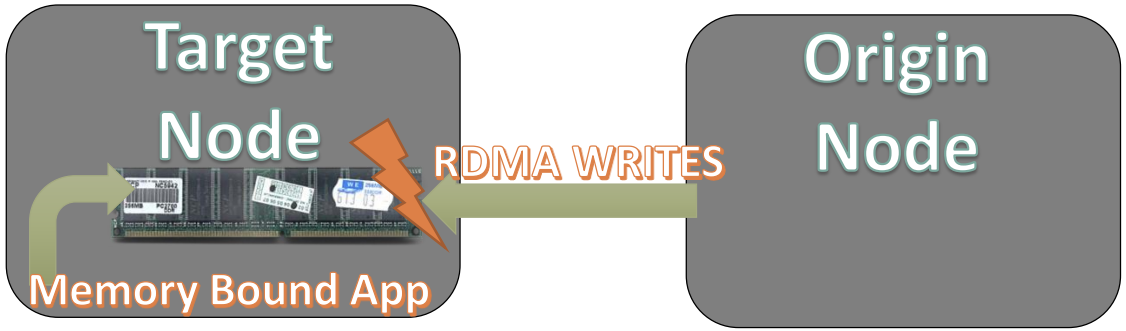
- One-sided communication is part of the Exascale solution
  - Decouples synchronization from data transfer, i.e. asynchronous data movement

1. We evaluate multi-threaded Remote Memory Access in MPI
   - Release first publicly available multi-threaded RMA benchmarks for MPI

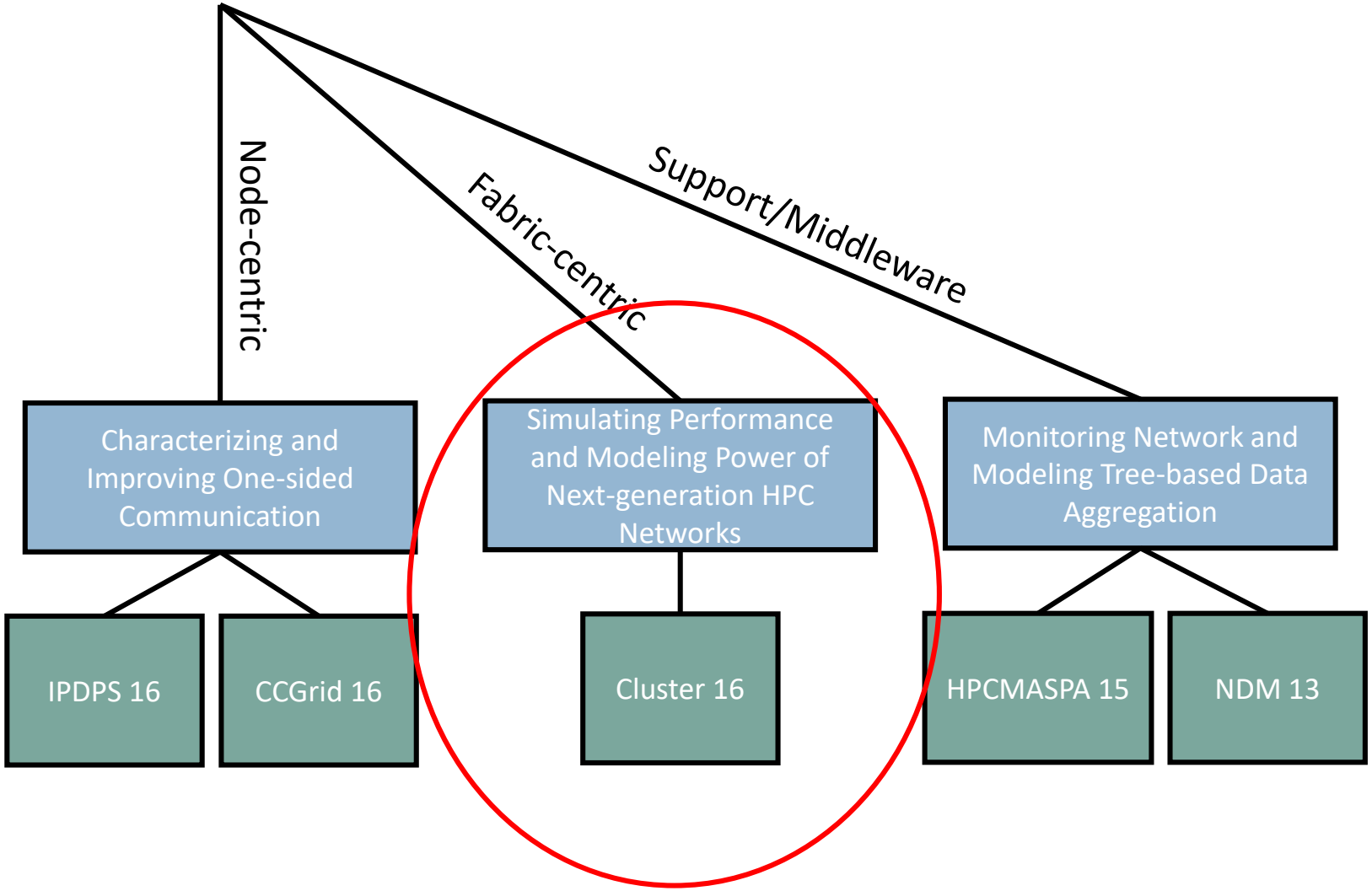
2. We ask how Remote **Direct** Memory Access (IB Verbs) might create contention in the memory subsystem
   - Introduce the concept of Network-induced Memory Contention (NiMC)
     - NiMC can result in 3X increase to runtime at the scales evaluated
   - Detect and predict NiMC impact using machine learning
   - Evaluate three candidate solutions

Application level

Transport level

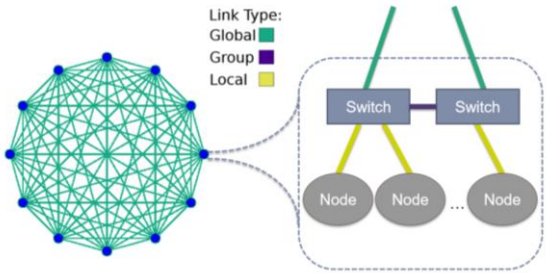


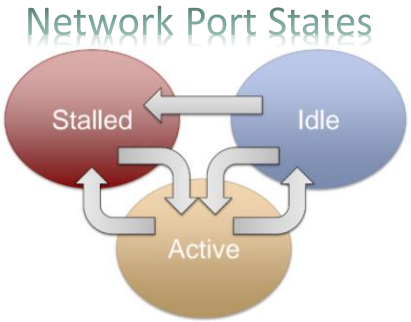
Characterizing and Improving Power and Performance in HPC Networks

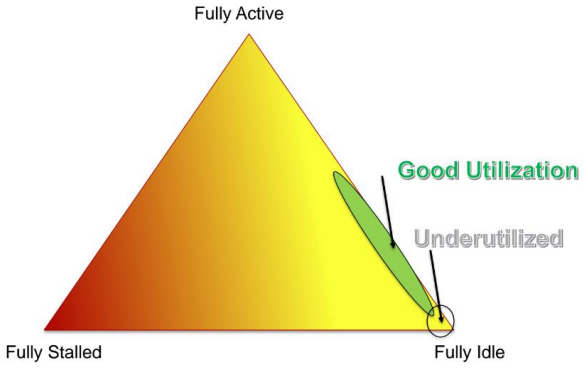


Simulating Performance and Modeling Power of Next-generation HPC Networks (Cluster 16)

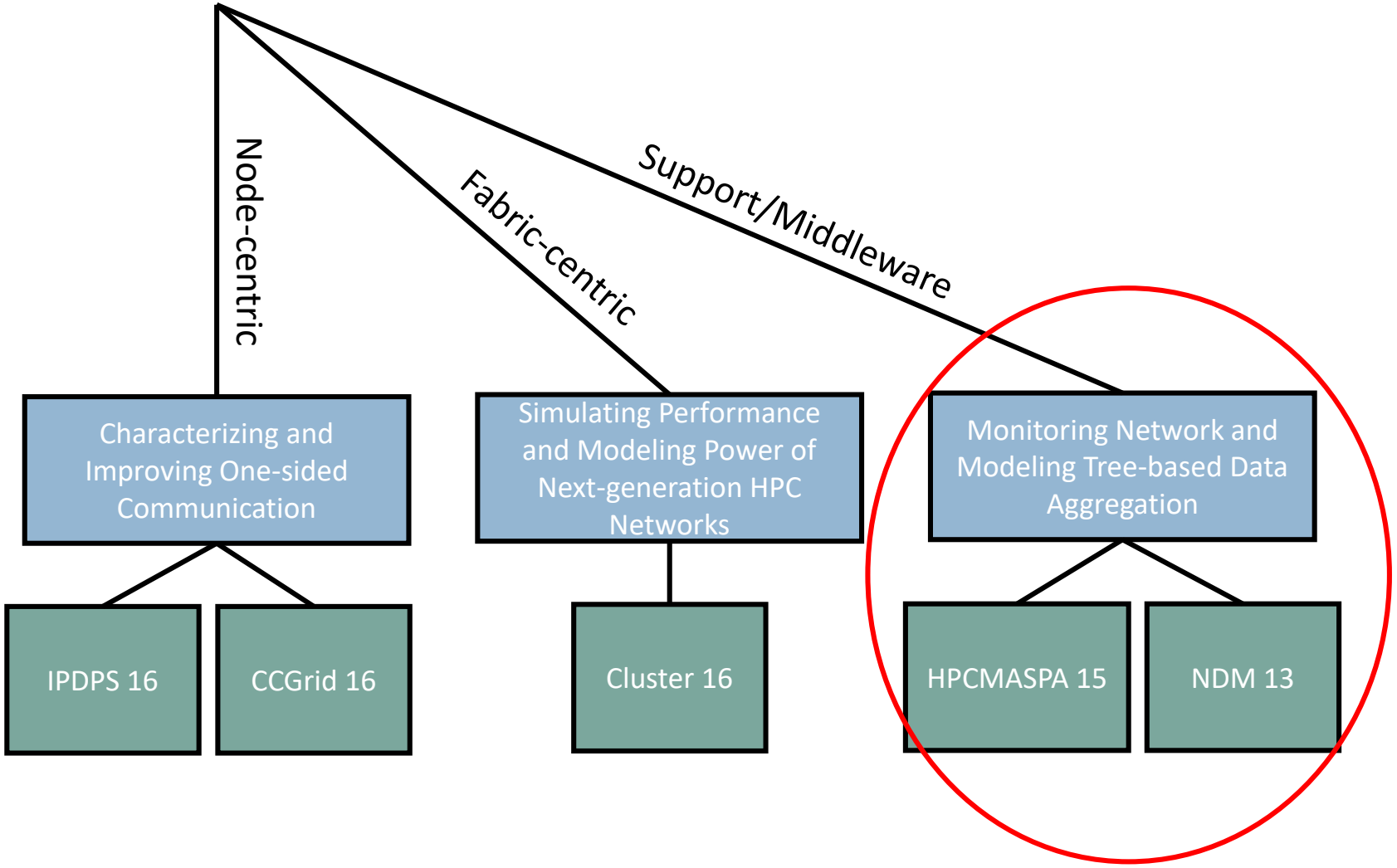
- Exascale networks are expected to consume around 10% of total system power.
- We need to take a detailed look at the tradeoffs between power and performance in the design of the network fabric.
  - Simulated a variety of 100,000 node dragonfly networks and workloads
  - Modeled power proportional networks and potential for static/dynamic reductions to network power
  - Introduced new technique for analyzing utilization of thousands of ports







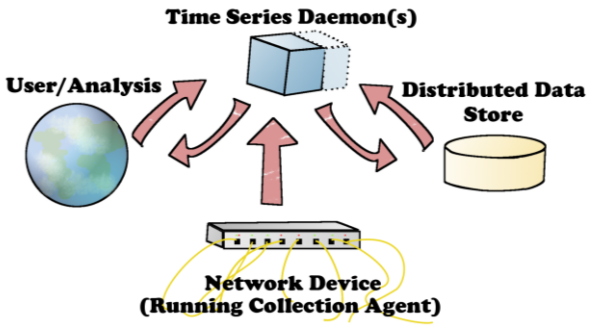
Characterizing and Improving Power and Performance in HPC Networks

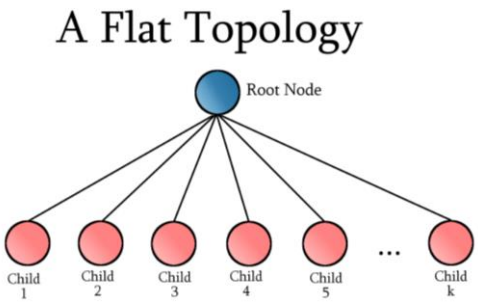


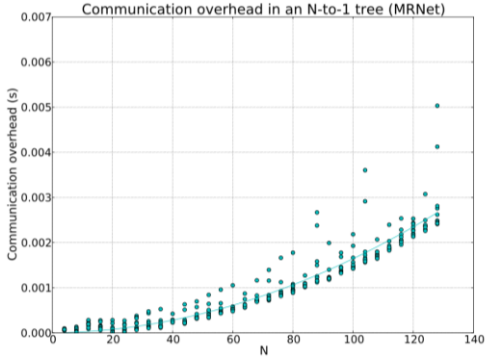
Monitoring Network and Modeling Tree-based Data Aggregation (NDM 13, HPCMASPA 15)

- Increased demand for dynamic solutions that can adapt to their environment
  - E.g. routing, power management, contention mitigation/throttling
- This knowledge of the environment relies on the monitoring system

1. Increase the scalability and responsiveness of network monitoring
   - Develop push-based in-network monitoring agents
2. Develop new models for hierarchical data aggregation
   - Utilized to create scalable topologies for overlay networks

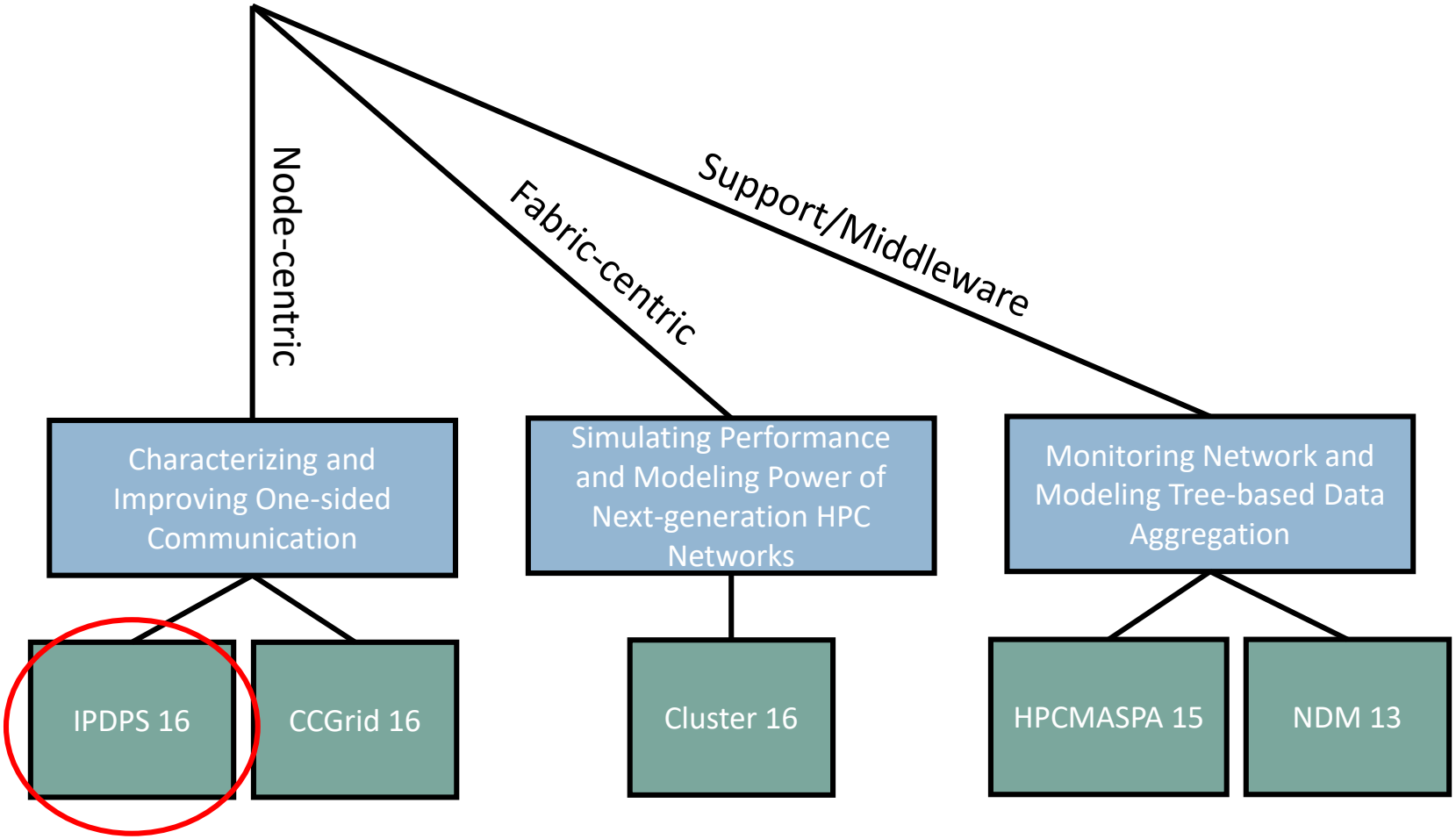




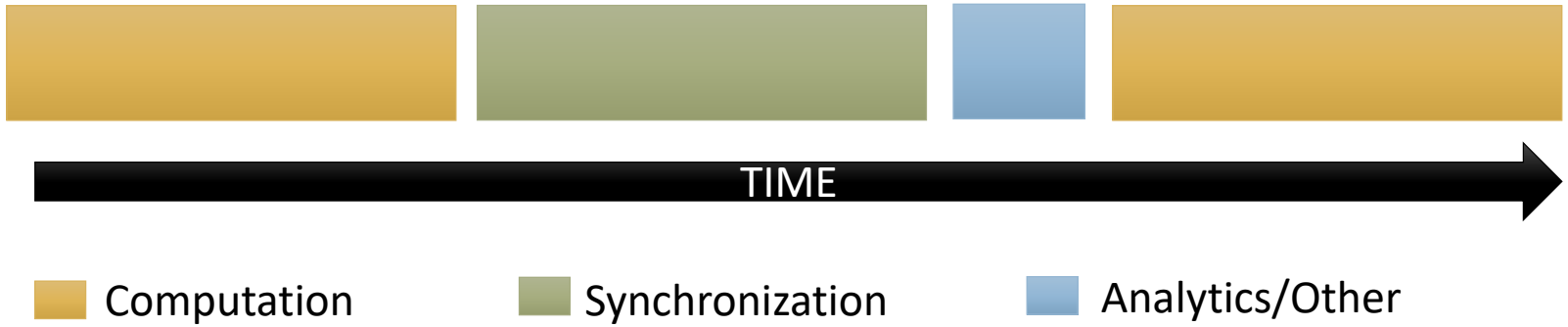


A Quick Tour of NiMC

With the remaining time, lets go a little deeper into one of the sub-topics (NiMC).

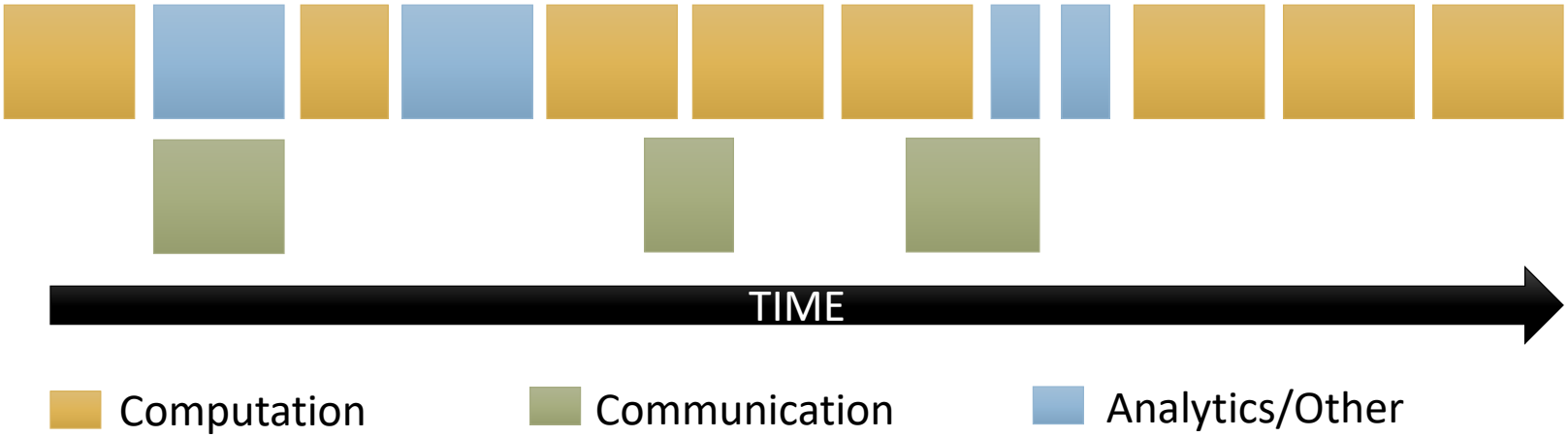


Traditional HPC



- Bursty workloads
- Synchronous communication models
- Contention for shared resources, e.g. memory, networks

Future HPC:

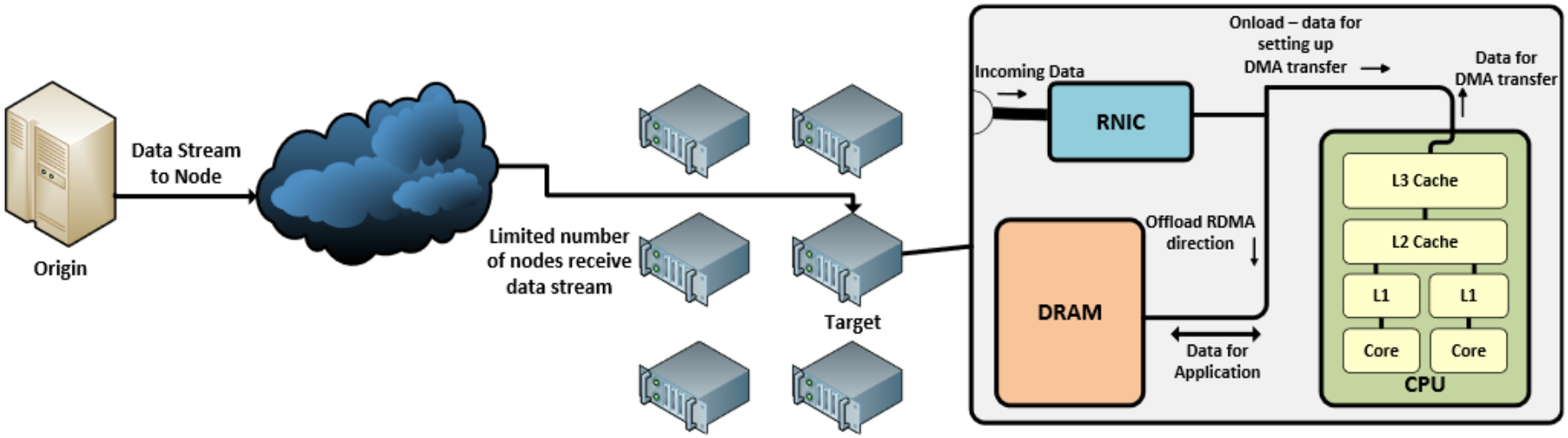


- Asynchronous many-task models
- Partitioned Global Address Space
- Analytics to improve effective resource management

Most of these techniques want to leverage Remote Direct Memory Access (RDMA)

Background (RDMA)

- Remote Direct Memory Access (RDMA)
  - Bypass the CPU and access memory directly
- Facilitates overlap between communication and computation



- However, there's a downside.

Increased Contention for Memory

"Fir0002/Flagstaffotos"

What is NiMC?

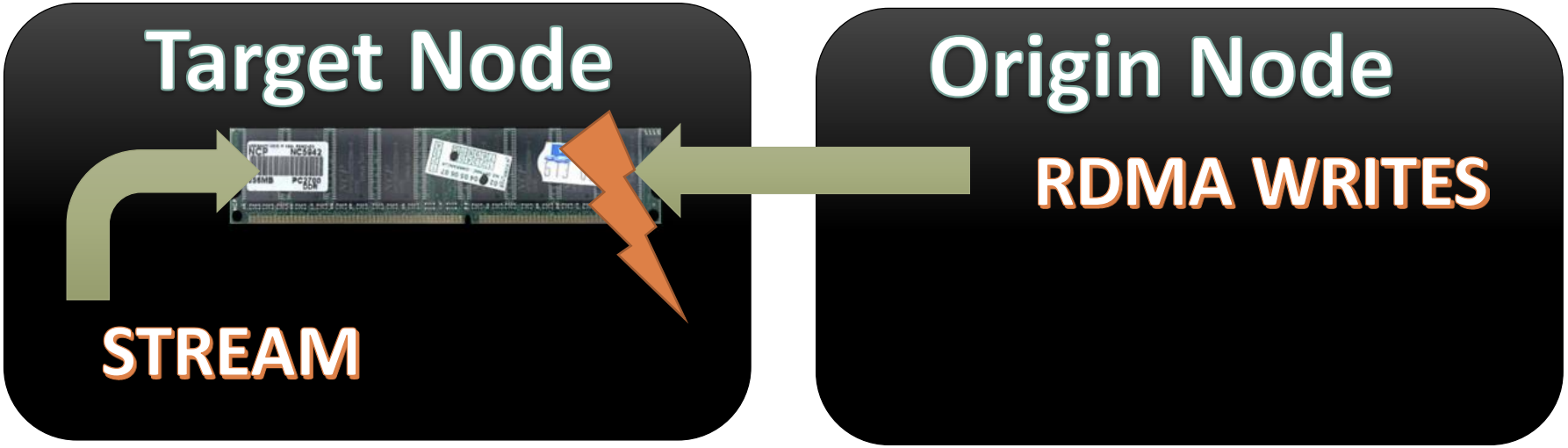
Network-induced Memory Contention:

Contention for local memory resources due to asynchronous communication originating from a remote node

Preliminary Evaluation

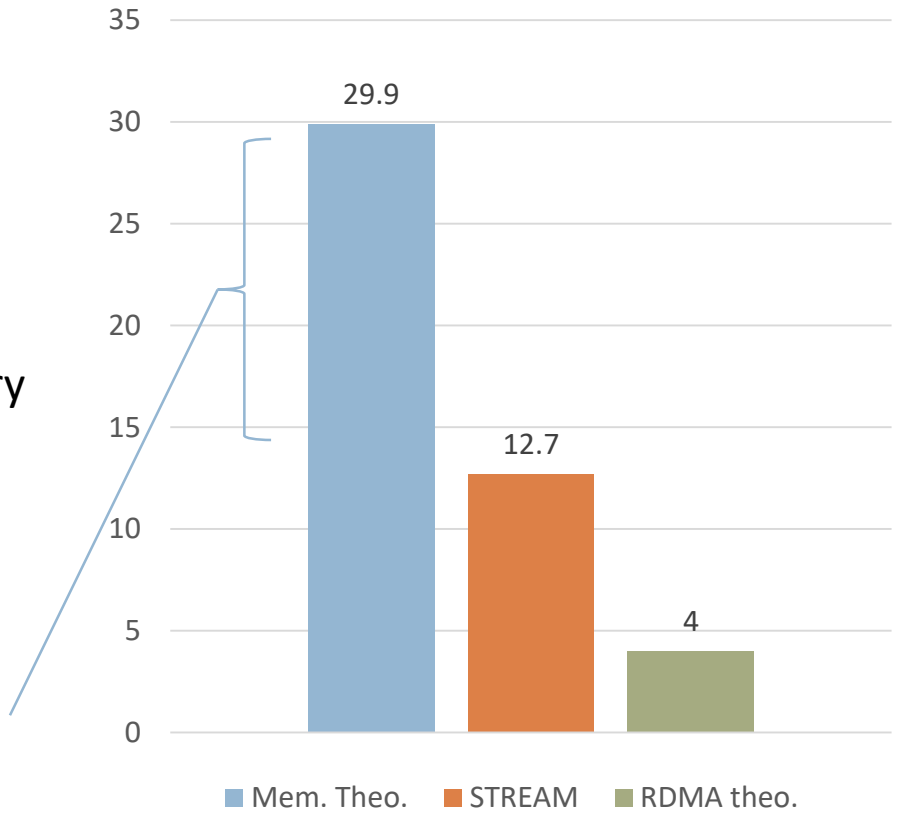
Test the worst case scenario

1. Run memory intensive workload
2. From a separate node, RDMA writes/puts to push as much data as possible into the machine to further increase pressure.



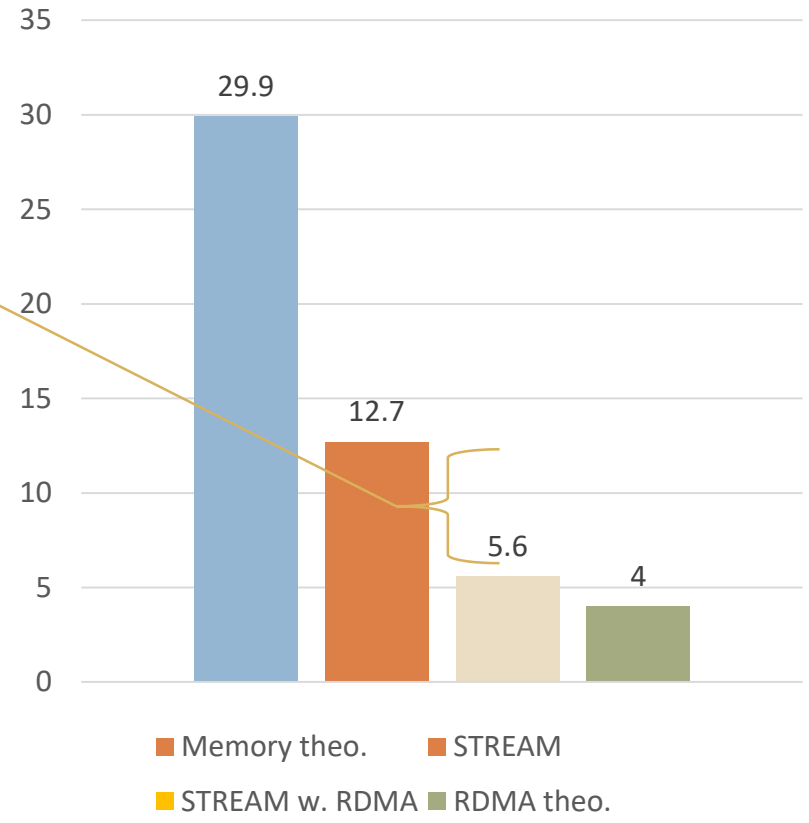
Preliminary Tests

- Experiments on small cluster of AMD Piledriver (4 cores)
  - Theo. Memory Bandwidth 29.9 GBps
  - Theo. Network bandwidth 4GBps
- STREAM benchmark (without RDMA)
  - Observed 12.7 GBps sustainable memory bandwidth
- 17 GBps headroom for RDMA



STREAM w. RDMA write

- However, performance worsens
- **56% penalty to STREAM**
- The penalty is greater than the total amount of RDMA
- Why is performance so bad?



Further Evaluation

- **Need more results to draw meaningful conclusions**
- 7 different CPU architectures
  - Ranging from Westmere (4-core) to Xeon Phi (57-core)
- 3 variations of Infiniband Networks
  - Including onload and offload NICs
- 6 different memory frequencies

6 out of 8 systems impacted by NiMC

- 4-56% reduction in sustainable bandwidth
- Most noticeable for systems with onload NIC's
- 3 offload systems see a reduction proportionate to the volume of RDMA writes

Machine	Triad no RDMA (GB/s)	Triad w. RDMA (GB/s)	Diff. (GB/s)	Diff. %
Westmere @ 800MHz, 1066MHz (offload)	12.9, 16.8	9.7, 12.8	-3.2, -4.0	-25%, -24%
Lisbon @ 800MHz, 1066MHz, 1333MHz (offload)	14, 17.9, 19.7	10.8, 14.3, 16.5	-3.2, -3.6, -3.2	-23%, -20%, -16%
Piledriver @ 1600MHz (onload)	12.4	7.4	-5	-40%
Piledriver @ 1866MHz (onload)	12.7	5.6	-7.1	-56%
SandyBridge-X2 (offload)	77.8	77.6	-0.2	0%
SandyBridge-X2 (onload)	73.4	36.1	-37.3	-51%
Xeon-Phi (on-chip, offload)	126.4	121.7	-4.7	-4%
Haswell-X2 (offload)	116.6	116.9	+0.3	0%

Further Evaluation

- Similar setup to earlier STREAM experiment

- 6 additional workloads of varying memory intensity
  - CNS
  - HPCCG
  - LAMMPS
  - Lulesh
  - SNAP
  - XSBench

Small Scale Results (Sandy-Onload)

What about CNS?

Why is LAMMPS more impacted than STREAM?

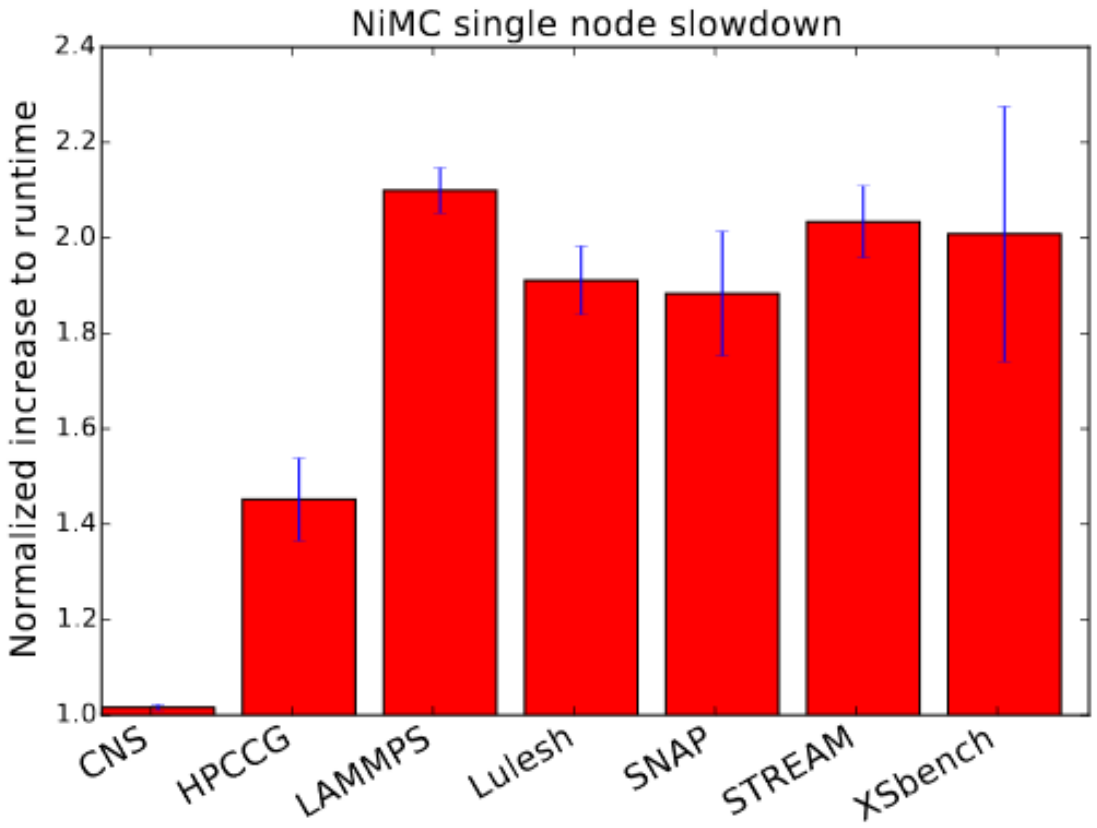


Fig. 3: Normalized impact of NiMC on single node runs.

Applying Machine Learning for In-depth Analysis

- Learn more about how NIMC occurs?

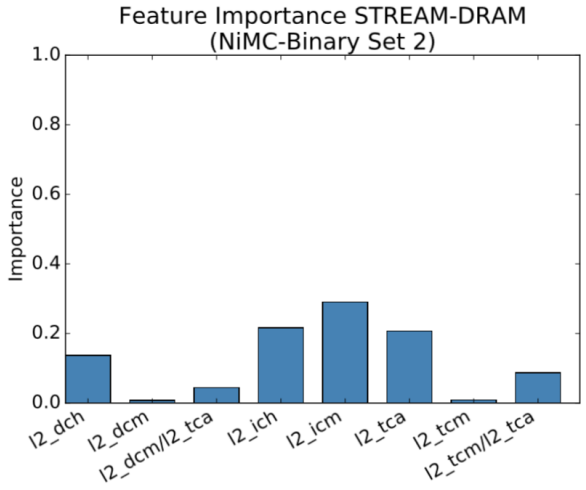
- Use basic counters to **detect** presence and **predict** NiMC impact?
  - L1, L2, L3, TLB, miss, hit, etc

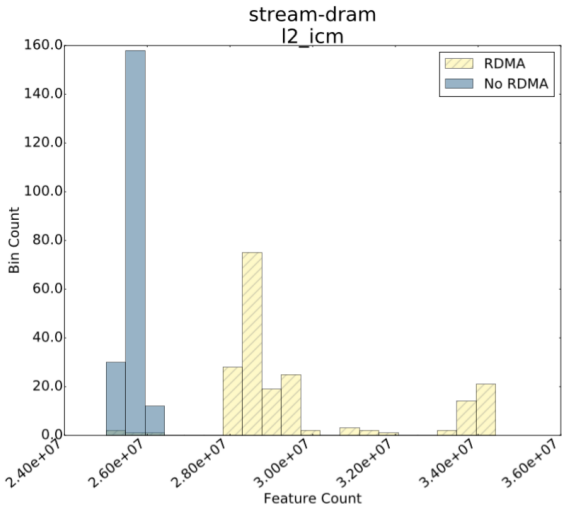
- Applied random forests to variety of workloads with 17 features
  - hundreds of runs with and without additional RDMA traffic

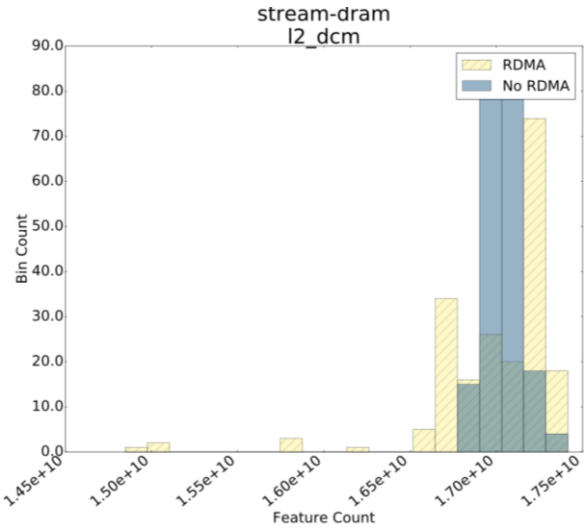
Applying Machine Learning for In-depth Analysis

- Learn more about how NIMC occurs?
  - Not just limited to contention on the memory bus & controllers
  - NiMC also polluting cache

- Use basic counters to **detect** presence and **predict** NiMC impact?
  - Yes, with high accuracy (> 96% correct classification)







Impact at Scale

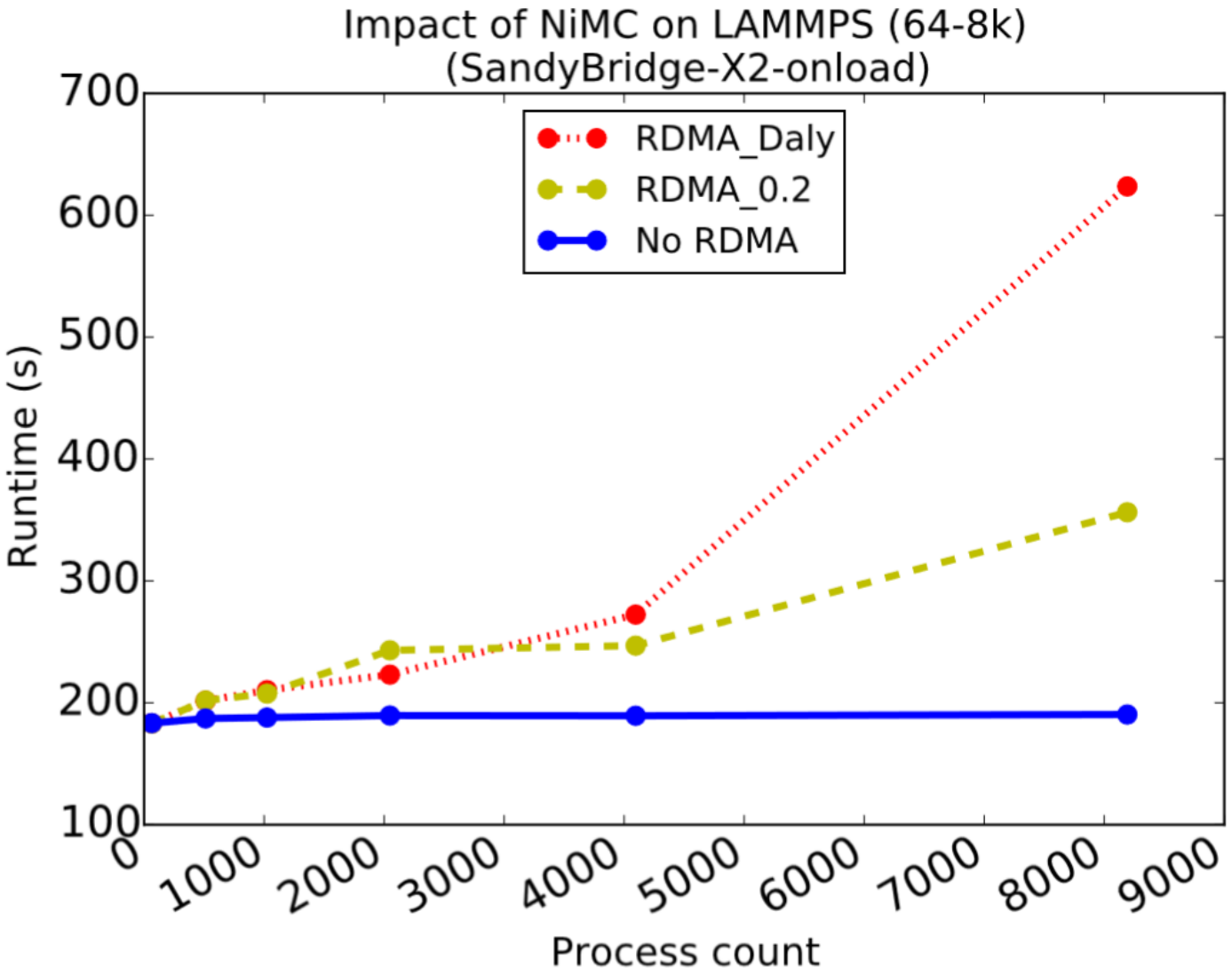
- LAMMPS, scaling up to 8,192 processes
- Ran on the SandyBridge-X2 Onload system.
- Interested in minimum runtime
  - Don’t want to capture performance degradation due to nearby jobs

Impact at Scale

Impact of NiMC given a reasonable amount of traffic?

- Hypothetical example: uncoordinated in-memory checkpoints
  - Reduced duration of RDMA writes (1 second)
  - Only writing to a subset of nodes at any point in time:
    1. 0.2% of nodes
    2. Daly's Optimal Coordinated Checkpoint Interval as an estimate (0.2-0.5%)

Impact at Scale



Solutions for Congestion

- Network Bandwidth Throttling

- Offload Network Cards
  - (for current-gen CPUs)

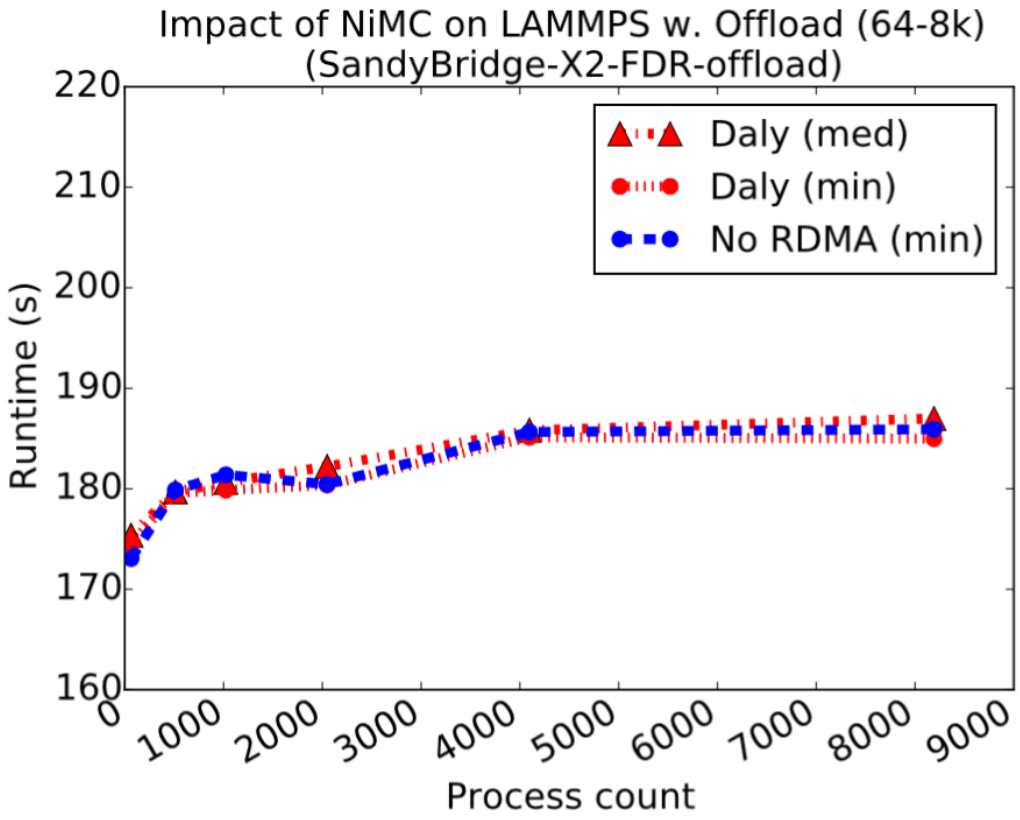
- Core Reservation

Mariordo (Mario Roberto Duran Ortiz)

Offload NIC

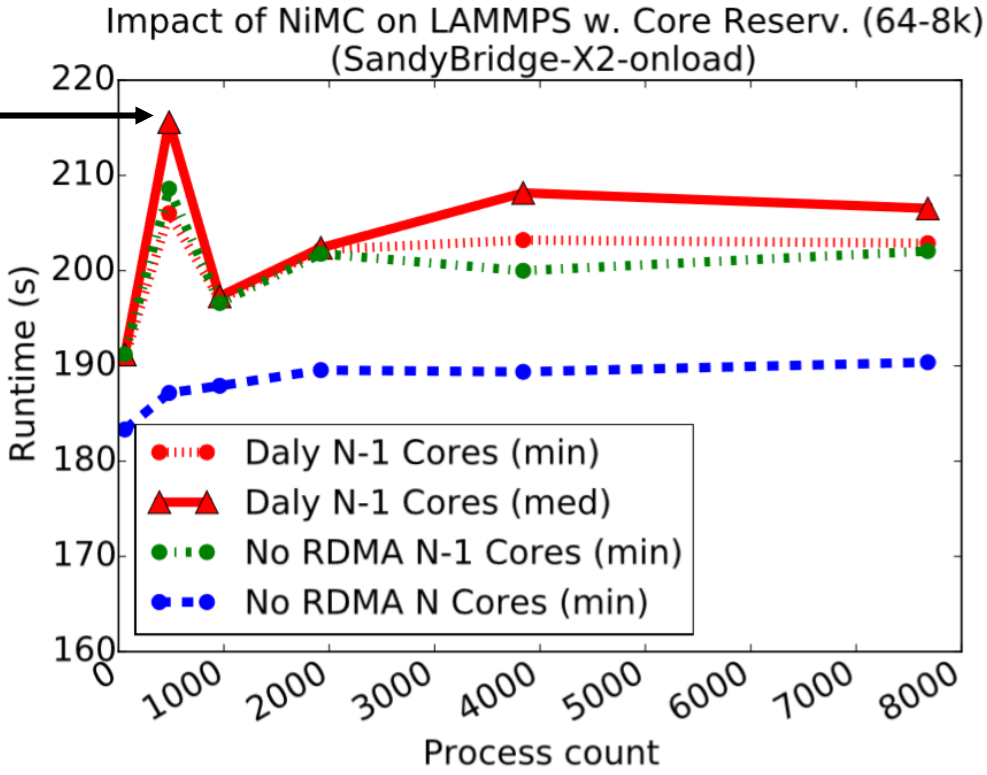
- Not a solution for earlier gen. CPUs (Westmere & Lisbon)

- Requires headroom between effective and theoretical memory bandwidth



Core Reservation

- Near constant overhead (approx. 6% increase to runtime)
- Bump caused by poor mapping with 15 procs per node
- If cores are “free” this is a pretty good solution

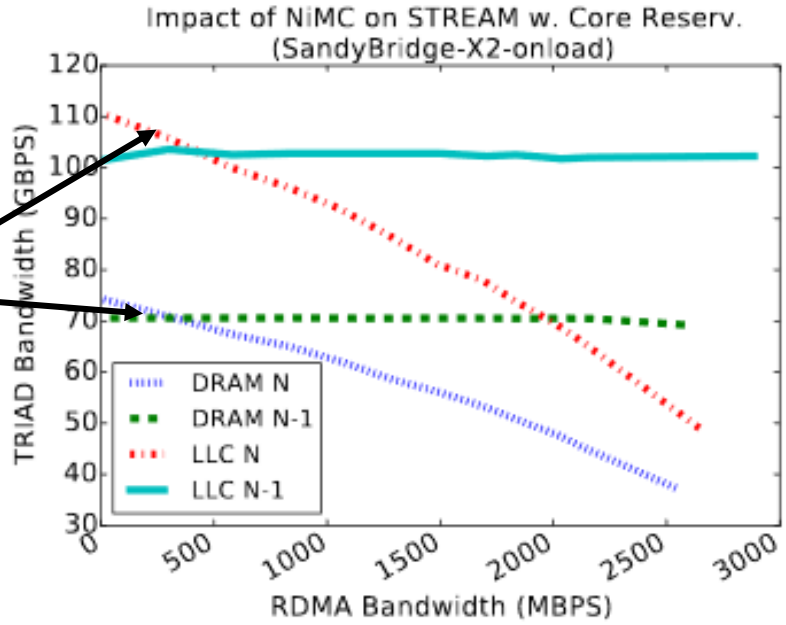


Bandwidth Throttling

- Evaluated for both LLC and DRAM

- Flat lines show core reservation

- Interesting opportunities for dynamically choosing the best solution

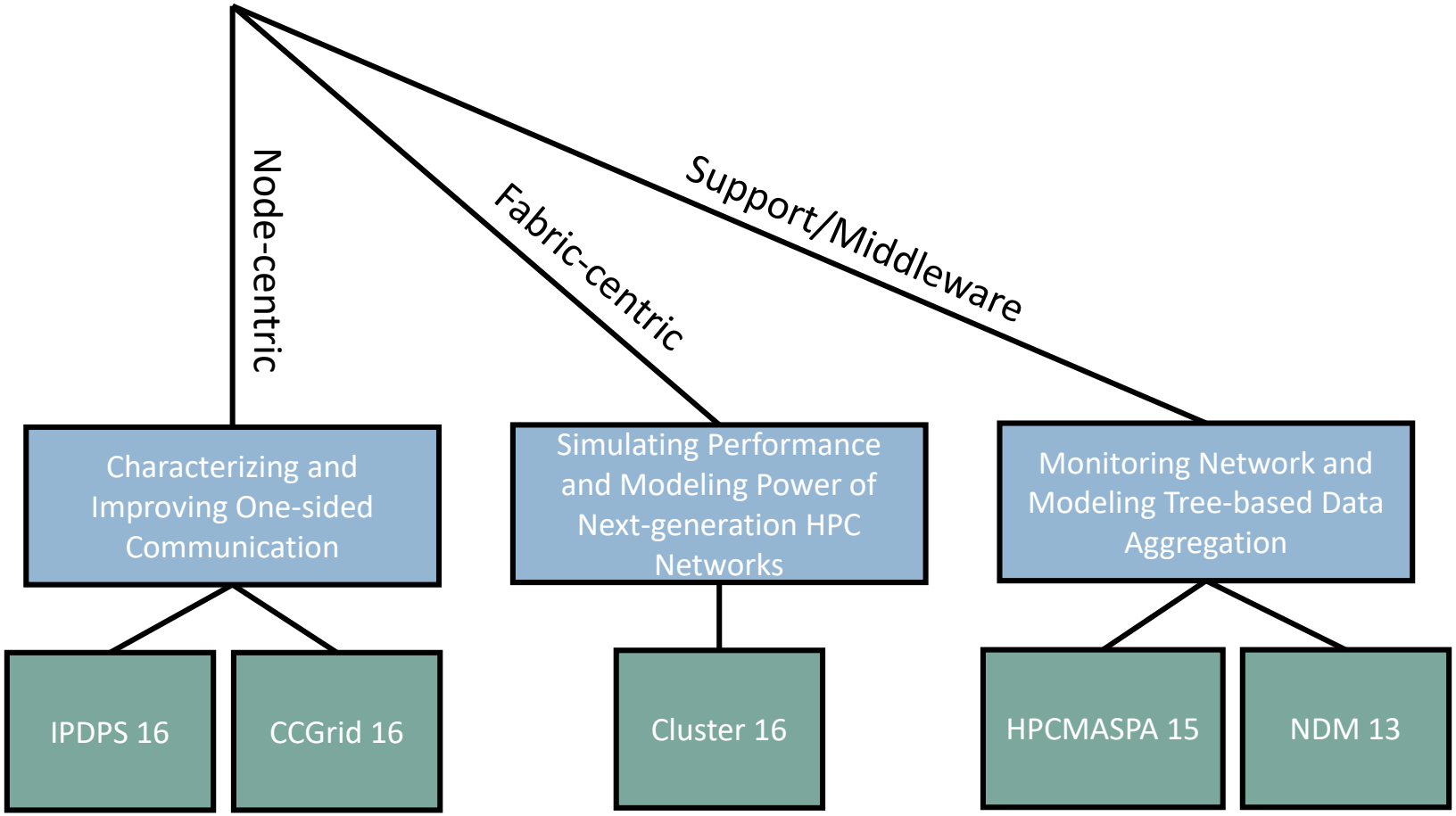


NiMC Takeaways:

- RDMA isn't free:
  - NiMC degraded performance on 6 out of 8 evaluated systems
- NiMC impact depends on architecture + workload:
  - Ranges from no impact to,
  - 3X slowdown in LAMMPS running on an onload system with 8k processes
- We can deal with NiMC, if we are conscious of its impact:
  - Offload NICs (for current CPUs)
  - Network throttling
  - Core reservation

This is just a part of this dissertation

As a whole, this work provides both a breadth and depth of work that improves power and performance in HPC systems with a focus on the network



Conclusions

- Future systems require massive improvements in both performance and power, of which networks will play a crucial role

- We took a comprehensive approach, targeting one-sided communication in the network end-points, power and performance tradeoffs in the fabric and scalability of network monitoring

- This resulted in numerous contributions, including:
  - Development of publicly available software
  - 5 peer reviewed publications

Summary of Contributions

Book Chapters

Ryan E. Grant, **Taylor Groves**, Simon Hammond, K. Scott Hemmert, Michael Levenhagen, Ron Brightwell, Network Communications, Book Chapter, Handbook of Exascale Computing, Eds. Ishfaw Ahmad, Sanjay Ranka,. ISBN:978- 1466569003 copyright Chapman and Hall (under review)

Posters

Taylor Groves (November 2016) "Characterizing and Improving Power and Performance of HPC Networks", Doctoral Showcase at the International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2016), Salt Lake City, UT, (60% acceptance).

Taylor Groves, Ryan Grant, Dorian Arnold (April 2016) "Network-induced Memory Contention." Poster session presented at the Salishan Conference on High Speed Computing, Gleneden Beach, OR, (by invitation).

Tech. Reports

Taylor Groves, Ryan Grant (2015) Power Aware, Dynamic Provisioning of HPC Networks Sandia National Laboratories SAND2015-8717.

Taylor Groves, Kurt Ferreira (2014) Balancing Power and Time of MPI Operations Center for Computing Research Sandia National Labs.

Contributions (cont.)

Peer Reviewed Publications

Taylor Groves, Ryan E. Grant, Aaron Gonzales, Dorian Arnold (2017 exp.), "Machine Learning to Predict, Characterize and Prevent Network-induced Memory Contention (NiMC)", *Journal In submission*.

Taylor Groves, Ryan Grant, Scott Hemmert, Simon Hammond, Michael Levenhagen, Dorian Arnold (2016) "(SAI) Stalled, Active and Idle: Characterizing Power and Performance of Large-Scale Dragon y Networks", In 2016 IEEE International Conference on Cluster Computing (CLUSTER), *(24% acceptance).*

Taylor Groves, Ryan Grant, Dorian Arnold (2016) "NiMC: Characterizing and Eliminating Network-Induced Memory Contention" In: 30th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2016), *(23% acceptance).*

Matthew Dosanjh, **Taylor Groves**, Ryan E Grant, Patrick Bridges, Ron Brightwell (2016) "RMA-MT : A Benchmark Suite for Assessing MPI Multi-threaded RMA Performance" In: 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2016), *(20% acceptance).*

Taylor Groves, Samuel K Gutierrez, Dorian Arnold (2015) "A LogP Extension for Modeling Tree Aggregation Networks" In: Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications (HPCMASPA 2015) in association with Cluster IEEE.

Taylor Groves, Dorian Arnold, Yihua He (2013) "In-network, Push-based Network Resource Monitoring : Scalable, Responsive Network Management" In: Proceedings of the Third International Workshop on Network-Aware Data Management (NDM 2013) in association with the International Conference for High Performance Computing, Networking Storage and Analysis, *(64% acceptance).*

Thanks!

tgroves@unm.edu

NiMC Architectures

TABLE I: Evaluated Architectures

machine	nodes	kernel	CPU	cores	channels	DRAM	DRAM GB/s	Network
Westmere@(800 MHz, 1066 MHz)	1	3.2.0 (Ubuntu12)	Intel E5620	4	2	16GB	12.8, 17.1	QDR IB off
Lisbon@(800 MHz, 1066 MHz, 1333 MHz)	1	3.13.6 (UN12)	AMD 4170 HE	6	2	16GB	12.8, 17.1, 21.3	QDR IB off
Piledriver-1600	70	2.6.32 (RHEL6)	AMD A10-5800K	4	2	16GB	25.6	QDR IB on
Piledriver-1866	2	2.6.32 (RHEL6)	AMD A10-5800K	4	2	64GB	29.9	QDR IB on
Sandy Bridge-X2-FDR-offload	6400	2.6.32 (Cent6.3)	2× Intel E5-2680	8	4	64GB	85.3	FDR IB off
Sandy Bridge-X2-onload	1196	2.6.32 (RHEL6.2)	2× Intel E5-2670	8	4	64GB	102.4	QDR IB on
Xeon-Phi (on-chip bandwidth)	49	2.6.38.8+mpss3.1.2	Xeon Phi 3120P	57	12	6GB	240	QDR IB off
Haswell-X2	33	3.14.23 (RHEL6.5)	Intel E5-2698	16	4	128GB	136	FDR IB off

NiMC Large-scale Study -- Number of Concurrent Writers

TABLE V: Number of concurrent RDMA writes

Application node (rank) count	Writes/s (Daly) QDR-onload	Writes/s (Daly) FDR-offload	Writes/s (0.2%)
64	0	0	0
512	1	1	1
1024	2	2	2
2048	5	6	4
4096	15	17	8
8192	42	47	16