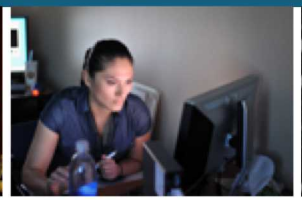
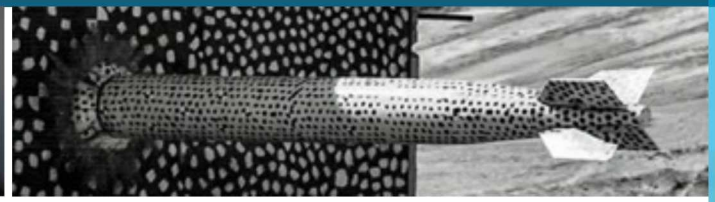
Sandia National Laboratories

SAND2019-9507PE

Proxy Applications: Curation and Assessment





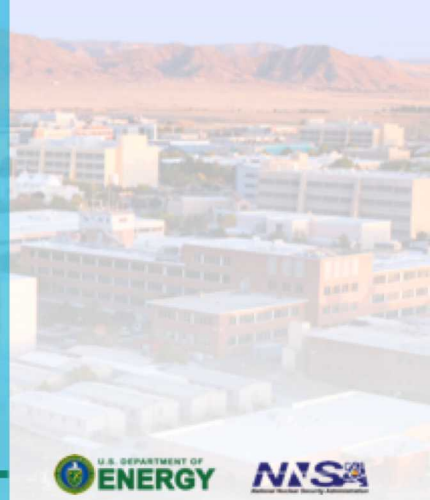
PRESENTED BY

Jeanine Cook

Collaborators: Omar Aaziz (SNL) , Jonathan Cook (NMSU), Jeff Kuehn (LANL), Courtenay Vaughan (SNL)

CIS External Review, August 26-29, 2019

SAND 2019XXX-XX



U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Exascale Computing Project (ECP): Proxy Application Project

- Began in 2017; ~2M/year
  - Is an Application Development, Co-Design project
  - Interact with Application Development teams, Co-Design centers, and Hardware & Integration projects
- Significance of results
  - First determination of proxy/parent representativeness for almost entire current ECP proxy app suite
    - Data generated provides consumers with where/how proxies faithfully model parent applications and where/how they do not
  - First extensive set of low-level hardware performance characterization data on large set of ECP proxies and applications
- Significant impacts
  - Vendor interaction
  - Infrastructure development
  - Procurement
  - Early testbed systems
    - Performance and software stack

What are Proxies and They So Important?

- Proxies are relatively small programs that are intended to capture fundamental aspects of a real scientific application
- Important because
  - Export-controlled and classified applications can not be openly distributed
    - Want broader community to explore optimizations
    - Want application behavior/hardware bottlenecks to be known to vendors
  - Ease of use, complexity and time reduction for
    - Programming model exploration
    - Algorithm development/optimization
    - Use by vendors in future system development
    - Hardware/system simulation

Good proxies provide a faster vehicle for developing systems that are well-suited to our applications and applications that are optimally implemented for those systems.

Important question: what is ”good”?

Proxy Intent and Fidelity

- Programming model exploration and algorithm optimization
  - Do not necessarily precisely model underlying behavior of real application on hardware or real application algorithm(s)
  - Kokkos and Raja are heavily leveraged in programming model exploration
- Characterize important performance issue of parent application
  - Precisely models underlying behavior of real application on hardware, but does not necessarily precisely model real application algorithm(s)
    - Underlying behavior may be one/all/or some of compute, memory, I/O, network, and communication behavior

How precisely the proxy models the parent application algorithm or performance ➔ fidelity

Related Projects

- Mantevo 1.0 Release (December 2012) – Mike Heroux
  - First release of tri-lab set of mini-apps
- CSSE PMAT (Performance Modeling and Analysis Team) project (started around 2012; ~1M per year)
  - Developed miniGhost, miniAMR
  - First efforts to validate proxies against parents
  - Became the current ATDM/APT project
- IC/Co-design (2013)
  - NNSA Tri-lab collaboration
  - Offshoot meetings
    - JOWOG 34 ACS
    - DOE PPP
  - Release of HPCG 1.0
- ECP Proxy Applications
  - Is an Application Development, Co-Design project
  - Interact with Application Development teams, Co-Design centers, and Hardware & Integration projects

ECP Proxy App Project first that not just formally curates but assesses a large suite for representativeness

Exascale Proxy App Project

Lawrence Livermore National Laboratory

Dave Richards (PI)

Argonne NATIONAL LABORATORY

Hal Finkel
Brian Homerding

BERKELEY LAB
Bringing Science Solutions to the World

Peter McCorquodale

Los Alamos NATIONAL LABORATORY EST. 1943

Christoph Junghans
Robert Pavel
Vinay Ramakrishnaiah

OAK RIDGE National Laboratory

Tiffany Mintz
Shirley Moore
Greg Watson

Sandia National Laboratories

Omar Aaziz
Jeanine Cook
Courtenay Vaughan

Related Work

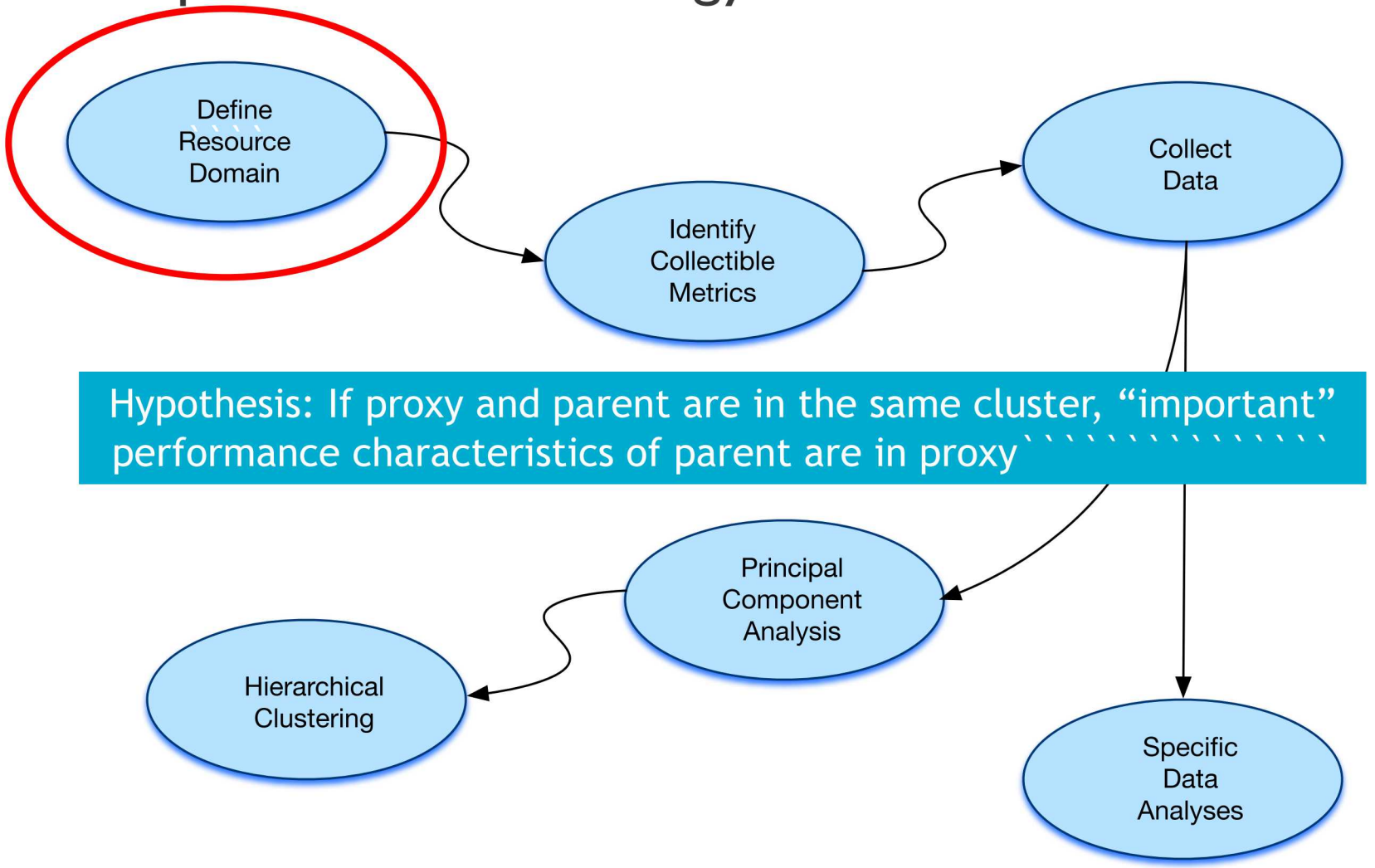
- Performance characterization studies of proxy/parent pairs (Tramm et al using OpenMC and XSBench; Sreepathi et. al using DOE co-design center proxies)
  - Focus on characterization but not formal methods of proxy/parent comparison to assess representativeness
  - Do not examine communication
- Veritas: framework to compare hardware resource coverage of proxy and parent apps (Islam et. al)
  - Similar to our work, but different methodology
  - Constrained to Veritas framework
  - Does not examine communication
- Performance comparison of parent and proxy from an application perspective (Barrett et. al)
  - Proxy and parent application key functions are compared by time (e.g., for MD, total time, time for force calculation, neighbor list construction)
  - Does not examine hardware-related performance

Key project goals: Curation and Assessment

- **Curation**: Curate an ECP proxy app **suite** to comprise **proxies** developed by ECP projects that **represent the most important features** (especially performance) of exascale **applications**
  - **Assessment:** A need to understand if important performance features in parent exist in proxy
    - Characterization and performance comparison
- Improve the quality of proxies created by ECP and maximize the benefit received from their use
  - A need to feedback results of characterization and performance comparison to developers for improvement

Representative suite of proxies should be minimal and accurate with respect to representing important performance features of parent applications

Are Important Performance Characteristics of Parent in Proxy? Comparison Methodology



Simple approach and does not depend on a specific framework or tools

Resource Domains

- Basic node
  - CPU and memory
- Communication
  - Pattern/MPI communication
  - Network
- Accelerator
  - E.G. GPU or other
- Storage I/O
  - Filesystem

Good results, but still working on improvements

Future work

Platform, Proxy/Parent Pairs and Tools

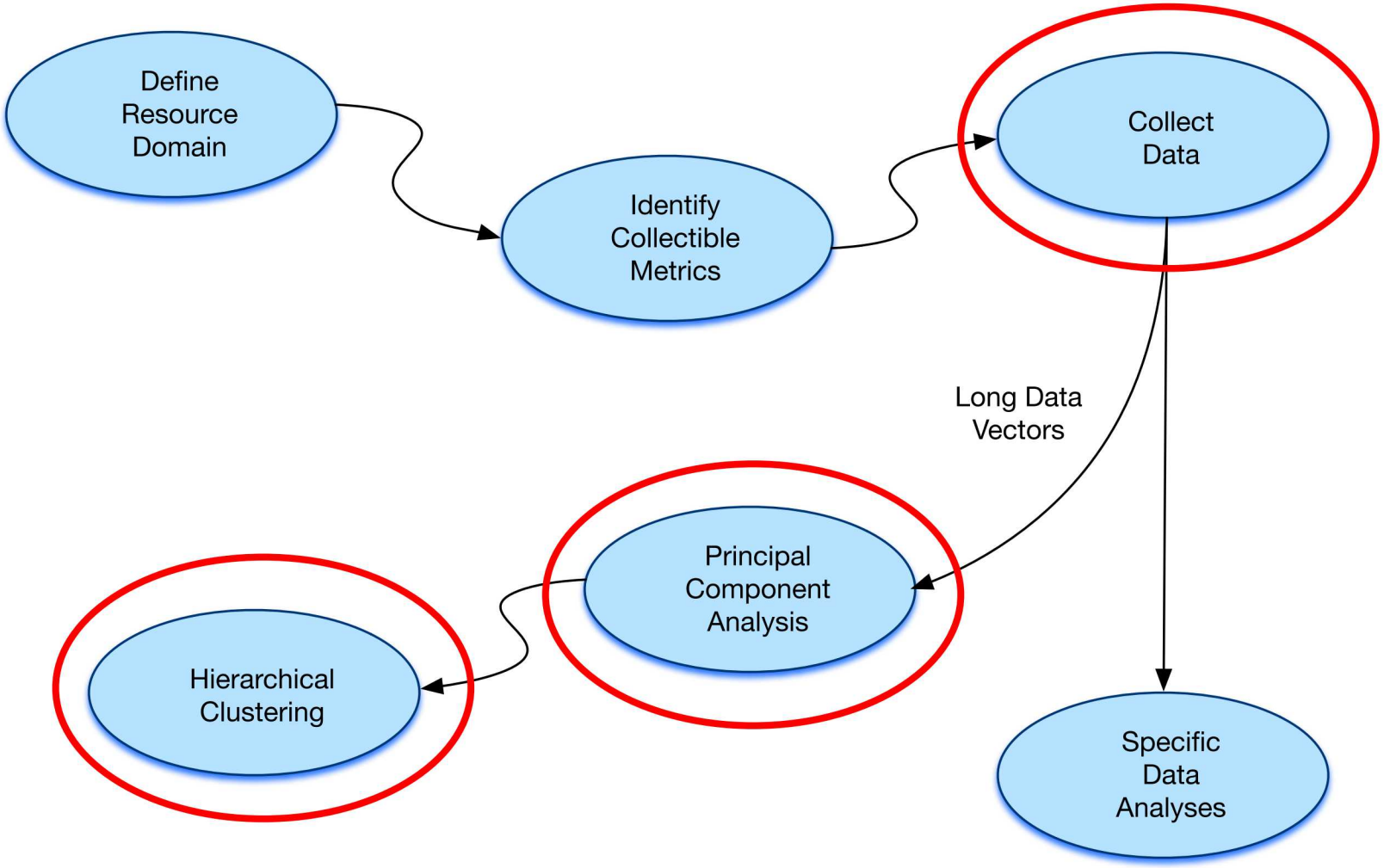
- Multiple Intel systems and Intel 18 compiler
- Tools
  - LDMS hardware performance counter sampler
  - LDMS mpiP sampler
  - R Statistical Computing Tool
- Run Configuration
  - Input/problems suggested by developers
  - Equivalent input/problem in parent and proxy
  - One process per core
  - 128 MPI ranks distributed across 8 nodes, using 16 cores per node

Proxy/Parent	Scientific Domain	Computational Motifs
SWFFT/HACC	Cosmology	Particles, sparse LA, dense LA, spectral, structured grid
SW4lite/SW4	Seismic modeling	Sparse LA, spectral, unstructured grid, structured grid, dynamic programming
ExaMiniMD/LAMMPS	Molecular dynamics	Particles, sparse LA, spectral, structured grid
Nekbone/Nek5000	Thermal transport	Dense LA, spectral, structured grid

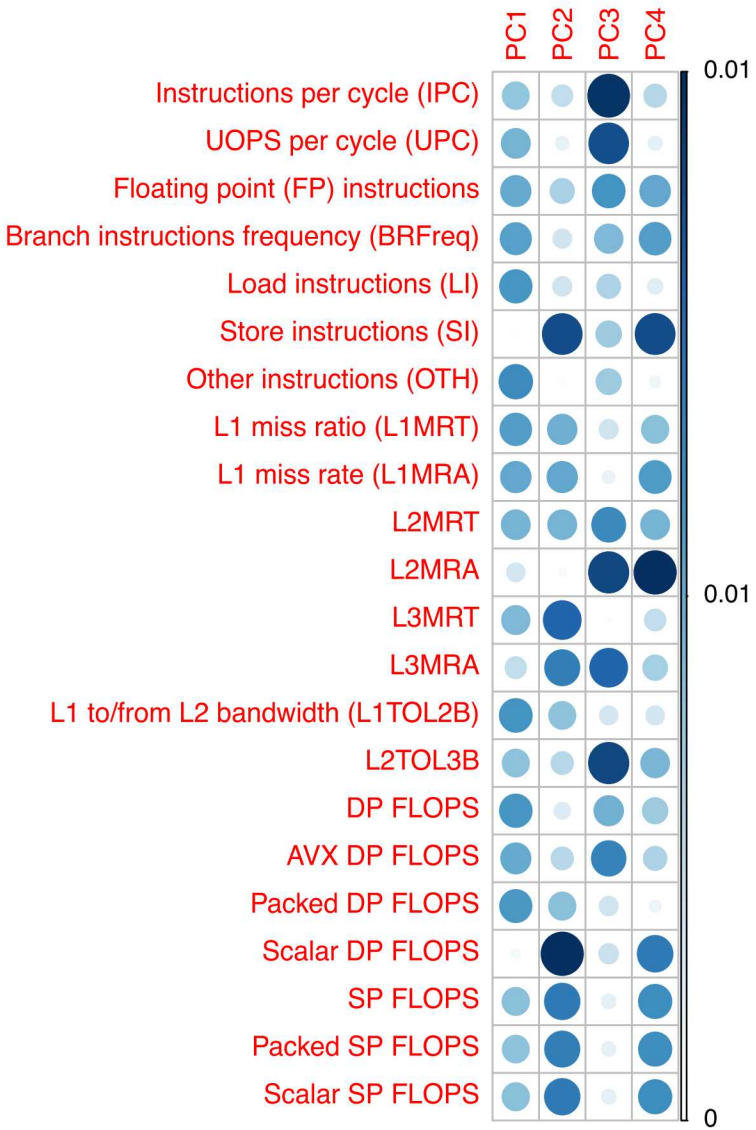
Graph500 used as a ”control”

All proxies meant to model computation, memory and communication behavior of parent application

Comparison Methodology



Which Characteristics are Important in Clustering?

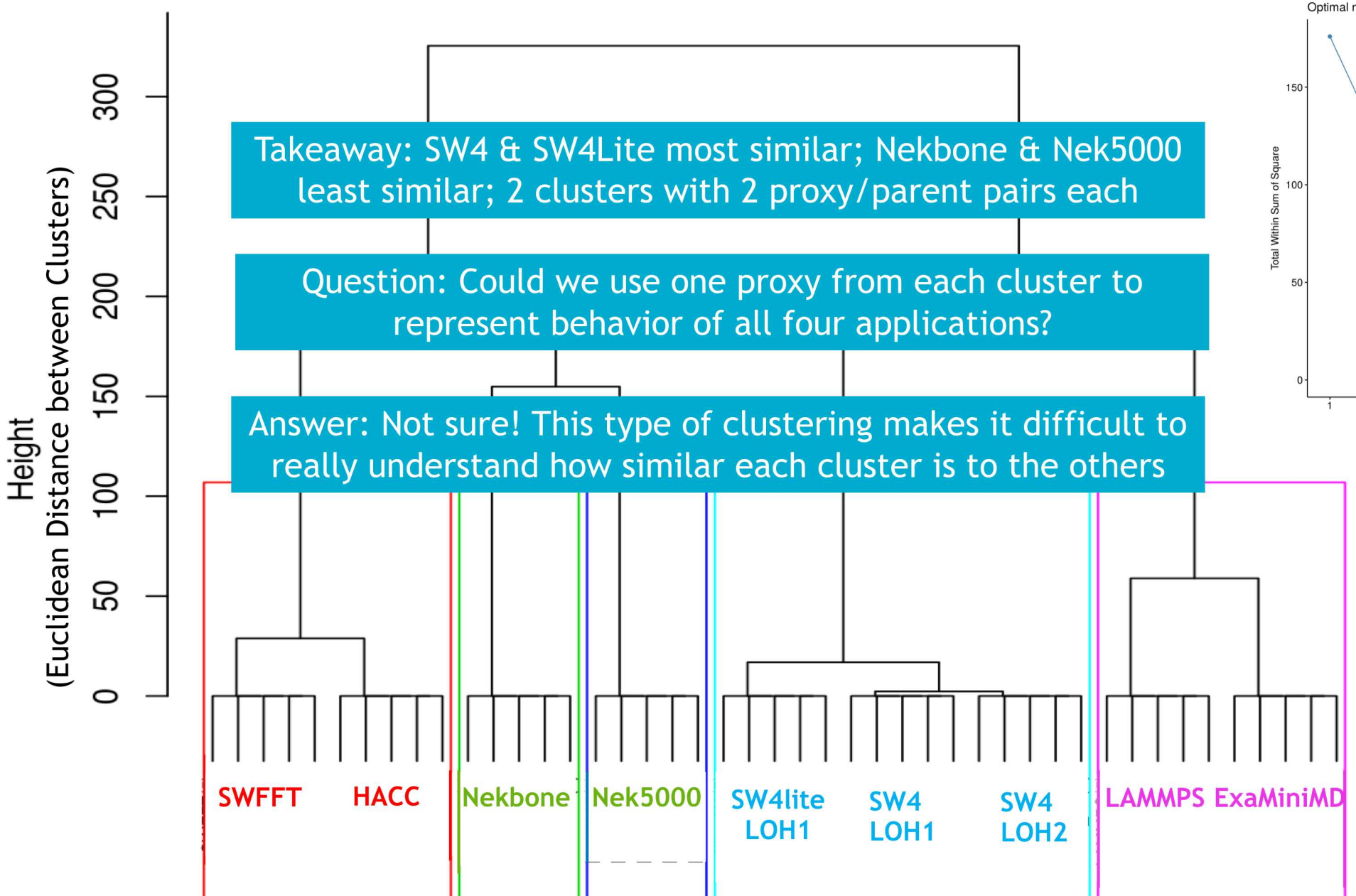


System 1

Takeaway: Instruction mix (DP FLOPS on System 1), L1 miss ratios and rates, and L3 miss rates seem to be important metrics for differentiating clusters/applications

- Each metric vector input to PCA has 22 *8 data points
  - Out of 128 ranks we use only data from Rank 0 and 7 other ranks randomly chosen
- PC1 – PC4 input to clustering algorithm

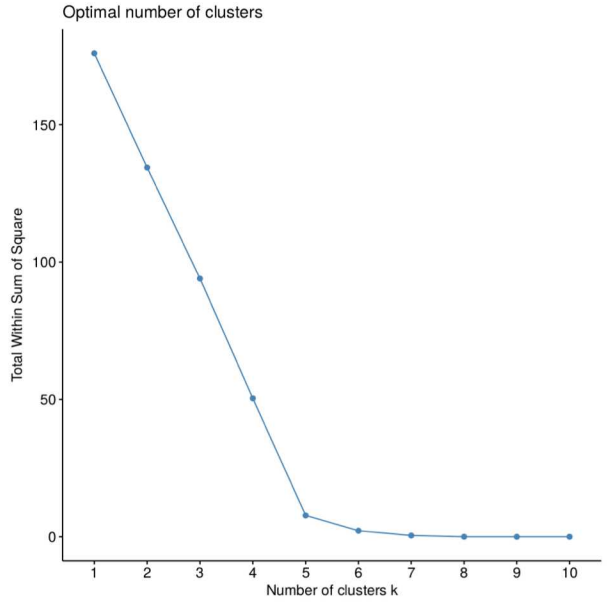
System 1 Node Clustering



Takeaway: SW4 & SW4Lite most similar; Nekbone & Nek5000 least similar; 2 clusters with 2 proxy/parent pairs each

Question: Could we use one proxy from each cluster to represent behavior of all four applications?

Answer: Not sure! This type of clustering makes it difficult to really understand how similar each cluster is to the others



Height indicates similarity; lower the height, the more similar; each application executed 5 times

More on Results

- Executing methodology on additional different Intel systems
  - Proxies always clustered with parents in same relative order of similarity
  - Highest level clusters do NOT remain consistent across architectures
    - Dependent on system and application characteristics
- Applying methodology to communication data gave us little to no information

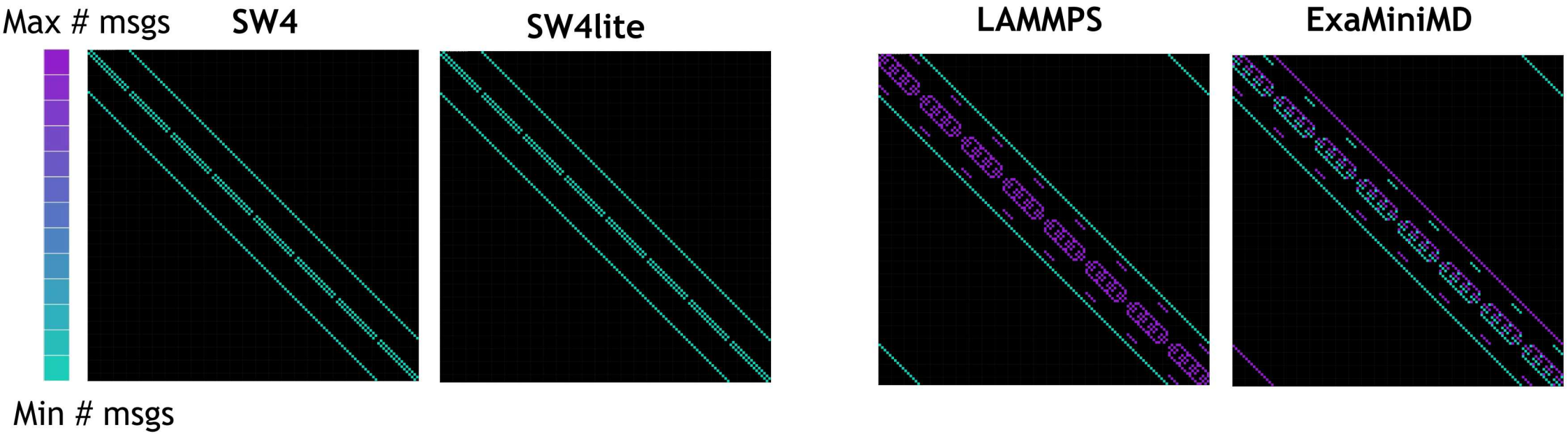
Pairwise Communication Data Comparison

Proxy		
Src	Dst	#Msg
0	1	152120
0	10	153422
0	100	1302
68	64	13020
68	65	13020
68	67	153422
68	69	1302

Parent		
Src	Dst	#Msg
0	1	35046
68	64	86
68	67	62
68	69	5887
68	70	24
68	71	29090
68	75	34984
68	91	5916

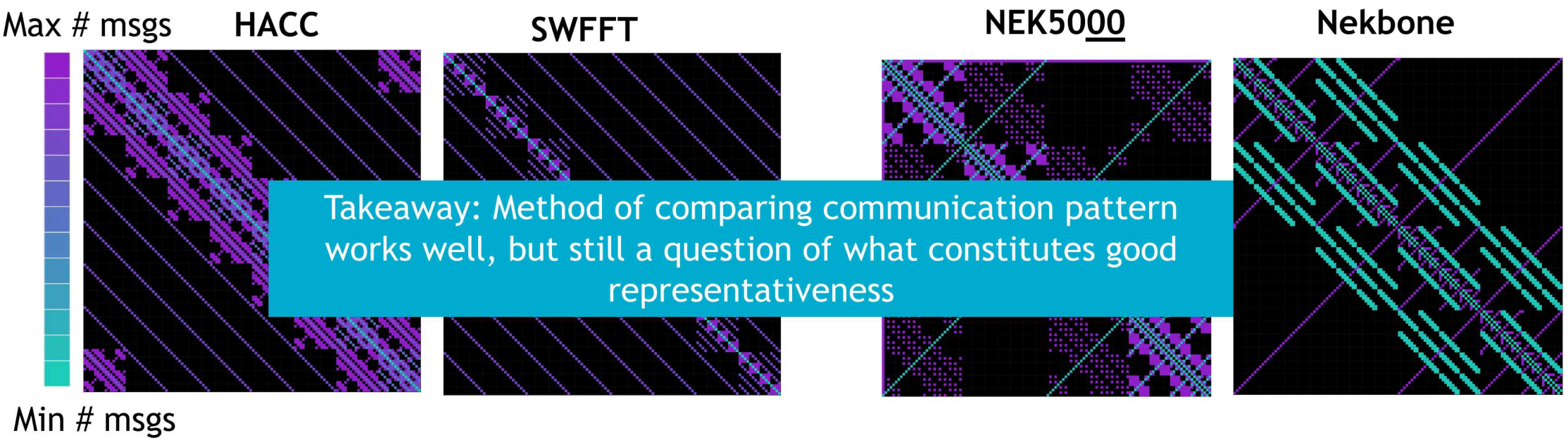
- Compare pairwise communication data in proxy and parent:
  - Full sets
  - Parent communication matching proxy
  - Proxy communication matching parent
- Use Spearman and Pearson correlation on resulting data sets

Results



- 100% of # messages and communicating pairs in parent are in proxy for
- 100% of # messages and communicating pairs in proxy are in parent
- Spearman and Pearson correlation ZERO for LAMMPS and ExaMiniMD
  - LAMMPS has bimodal distribution of message counts not seen in ExaMiniMD
- Spearman and Pearson correlation ONE for SW4 and SW4lite

Results



- HACC and SWFFT: All of proxy communication is in the parent, but not all parent communication is in the proxy
  - Correlation indicates this and is from 0.84 – 0.97
- Nek5000 and Nekbone: About 50% of proxy communication is in parent and 50% of parent communication in proxy
  - Poor correlation

Impact: Sandia Mission or External

- Enabled significant development of **application** monitoring in LDMS
  - Cross-platform node, memory, communication monitoring
    - Ivy bridge, Haswell, Broadwell, Skylake, Power9
  - Intel's Top-Down Microarchitecture (TMA) method
    - Developing similar bottleneck analysis method for Power9
- Vendor collaboration
  - Massively increased interaction with vendors with proxy app team member on every vendor working group team
  - Vendors pleased that they're "finally getting this sort of data from the labs"
  - Delivering proxies that better represent key application behavior to vendors
- Procurement
  - Collaboration with performance procurement teams to inform and run experiments
  - Proxies dominantly used for procurement benchmarking
- Proxy development
  - Feed back findings to application development teams
    - Working with teams to potentially refactor proxies where needed
- Testbeds
  - Proxies most commonly used as benchmarks on testbeds

Future Impacts: ATDM and Early System Use

- Leveraging infrastructure development in APT
  - Initial experiments on Sparc (analyzing data)
  - Plans for EMPIRE
  - Examine proxies for these apps as they're available
- Early system porting/monitoring and analysis
  - Implementing Arm capability into LDMS samplers for Astra experimentation
  - GPU MEASUREMENT IS CRITICAL CAPABILITY
    - Developing GPU sampler for Vortex (Sierra) experimentation
  - Early access to testbeds in procurement (ORNL and LLNL)
  - Exercising software stack (runtime, compilers, performance tools) to provide early feedback to vendors and facility teams

Improvements and Enhancements

- Refine definition of "good" proxy
  - miniQMC/QMCPACK study initial attempt
    - Based on determining if hardware bottlenecks are the same
    - Collaborate with vendors
- Determine improved measures of similarity
  - More intuitive, single number to describe similarity
    - Hierarchical clustering height too ambiguous
  - Measures from ML community such as cosine similarity and others
- Understand ECP proxy/parent pairs as a workload
  - Edison experiments
    - Measured node, memory, communication, network; injected network congestion
- Understand ECP workflows

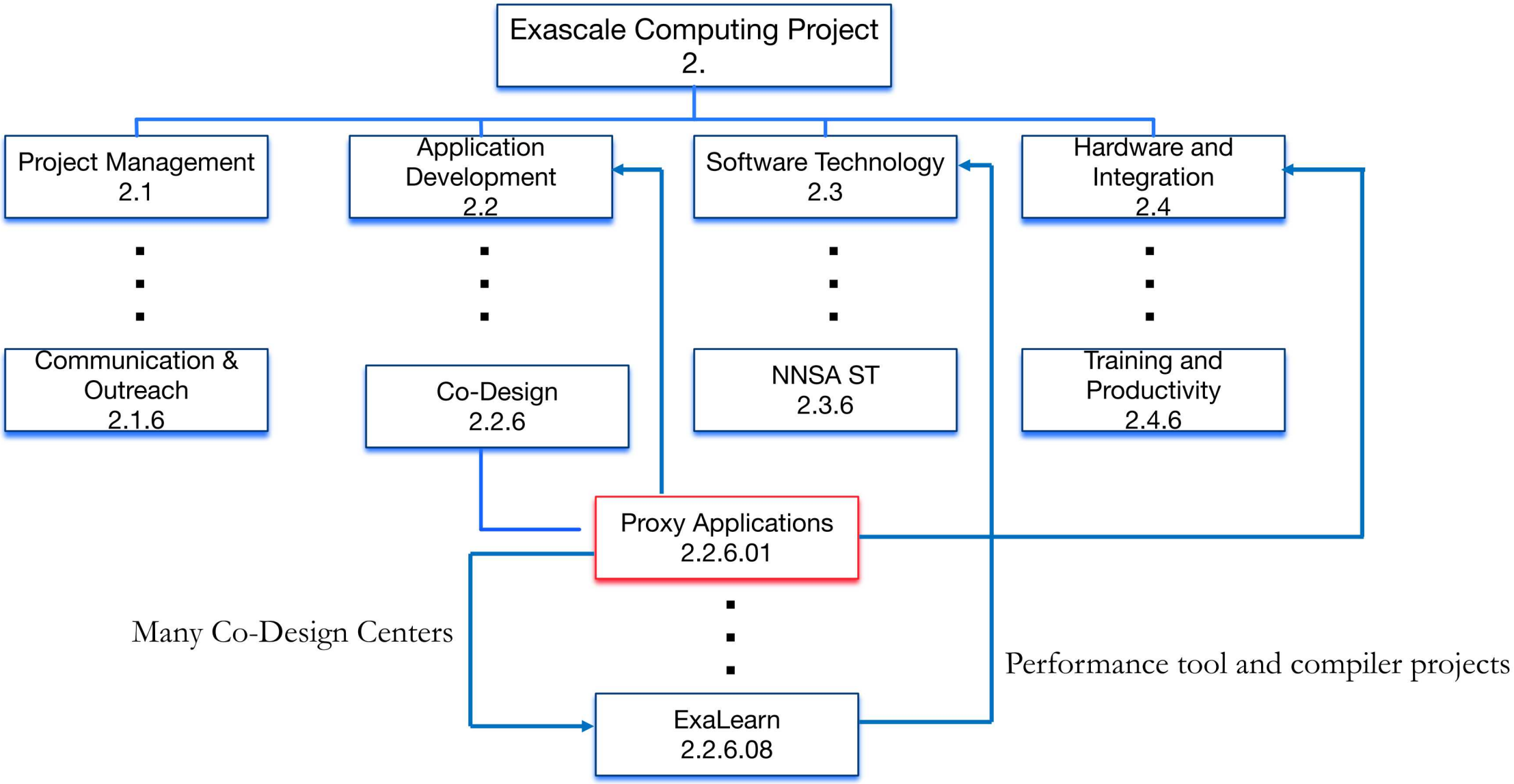
More Information

- Papers
  - O. Aaziz, J. Cook, and C. Vaughan, "Proxy or Imposter? A Method and Case Study to Determine the Answer", to appear in Workshop on Monitoring and Analysis for HPC Systems Plus Applications (HPCMASPA) at IEEE Cluster 2019.
  - O. Aaziz, J. Cook, J. Cook, and C. Vaughan,"Exploring and quantifying how communication behaviors in proxies relate to real applications," in *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, Nov 2018, pp. 12–22.
  - O. Aaziz, J. Cook, J. Cook, T. Juedeman, D. Richards, and C. Vaughan, "A methodology for characterizing the correspondence between real and proxy applications," in *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2018, pp. 190–200.
- Reports
  - D. Richards, O. Aaziz, J. Cook, H. Finkel, B. Homerding, P. McCorquodale, Y. Mintz, S. Moore, V. Ramakrishnaiah, C. Vaughan, and G. Watson, "Quantitative Performance Assessment of Proxy Apps and Parents", Report for ECP Proxy App Project Milestone AD-CD-PA-504-5
  - D. Richards, O. Aaziz, J. Cook, H. Finkel, B. Homerding, T. Juedeman, P. McCorquodale, Y. Mintz, and S. Moore, "Quantitative Performance Assessment of Proxy Apps and Parents", Report for ECP Proxy App Project Milestone AD-CD-PA-1040.

•jeacook@sandia.gov

THE END

Exascale Proxy App Project

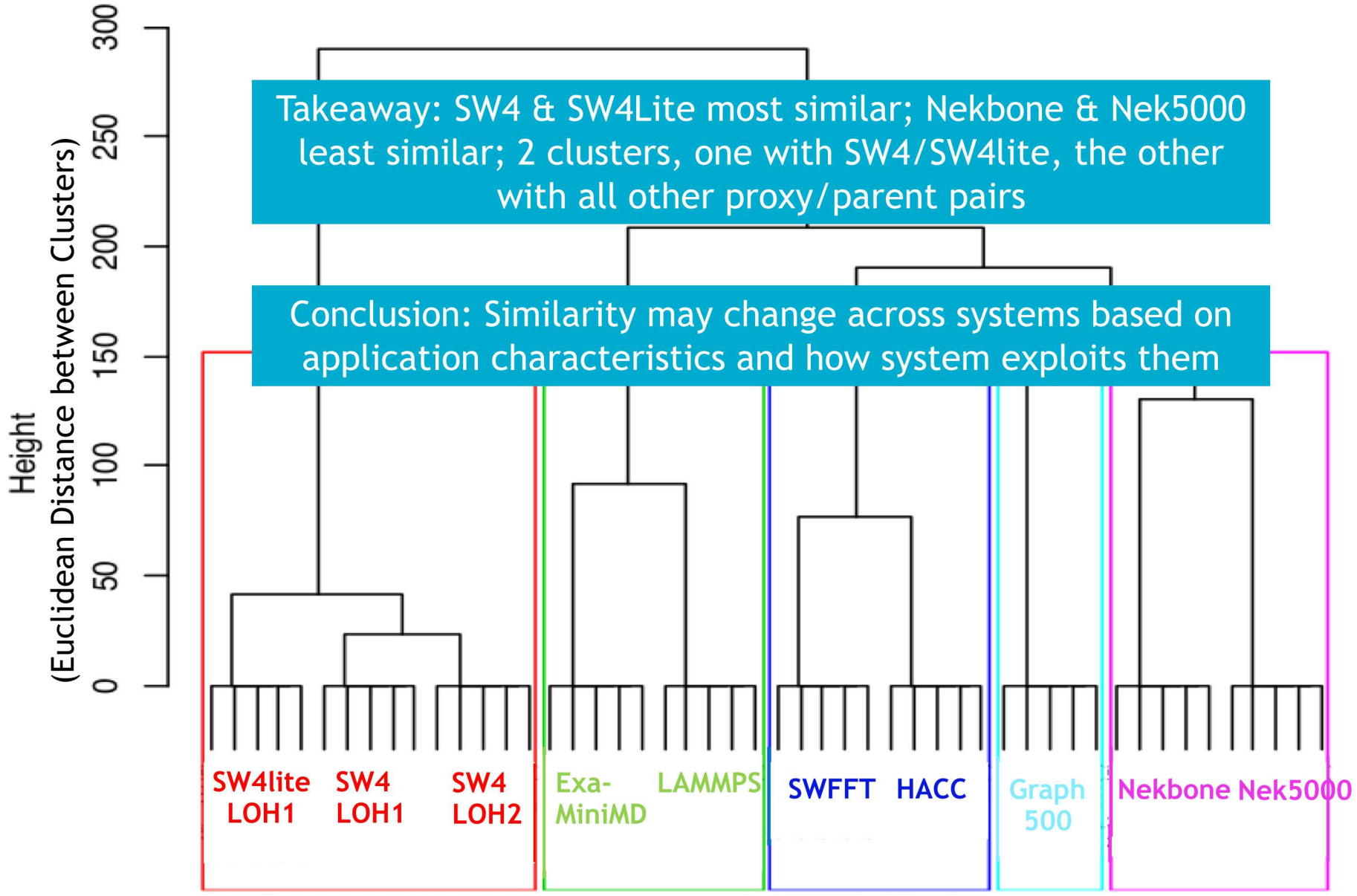


Selected Metrics

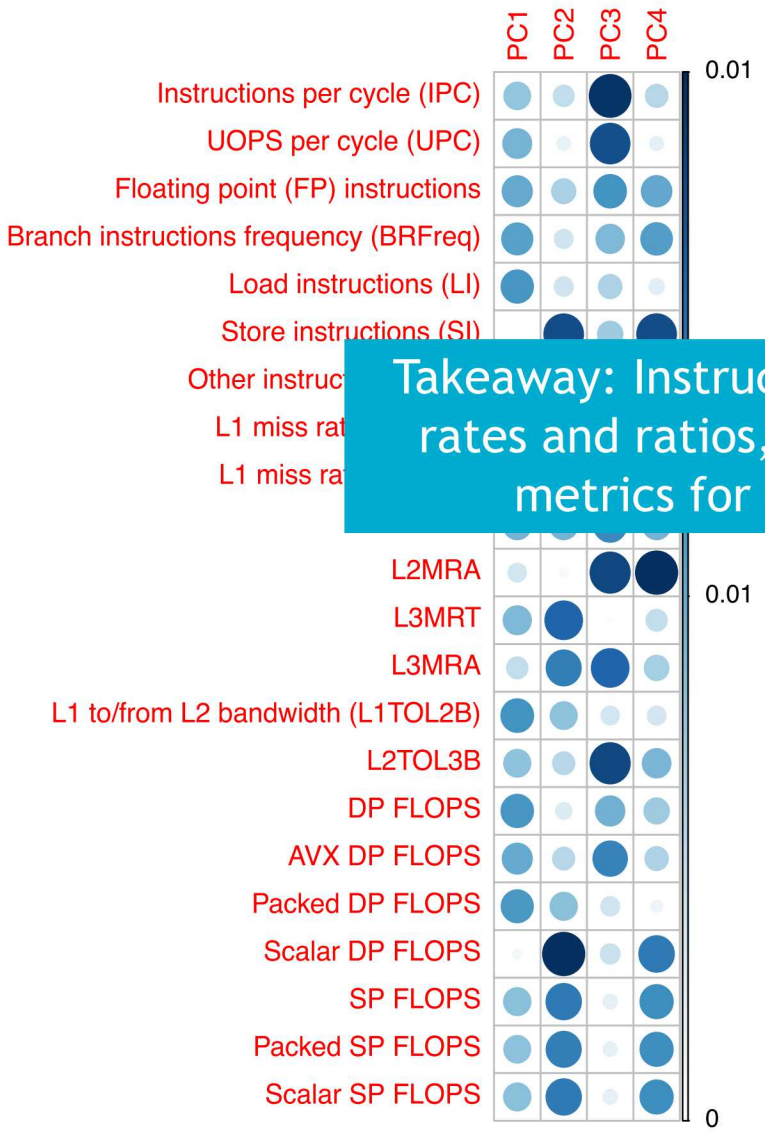
- CPU
  - IPC (instructions per cycle), UPC (micro-ops per cycle), instruction mix (5 categories), FLOPs (1 – N categories dependent on architecture)
- Memory:
  - L1/L2/L3 miss rates (per insn) and ratios (per access)
  - L1-L2-L3 bandwidths
- Communication
  - Histogram of frequency of message size (10 buckets)
  - Three summary metrics:
    - Total number of messages
    - Total size of data transferred (KB) divided by total execution time (KB/sec)
    - Total size of data transferred (KB) divided by total time spent in MPI (MPI KB/sec)

Minimal set defined through experimentation to minimize collection

System 2 Node Clustering

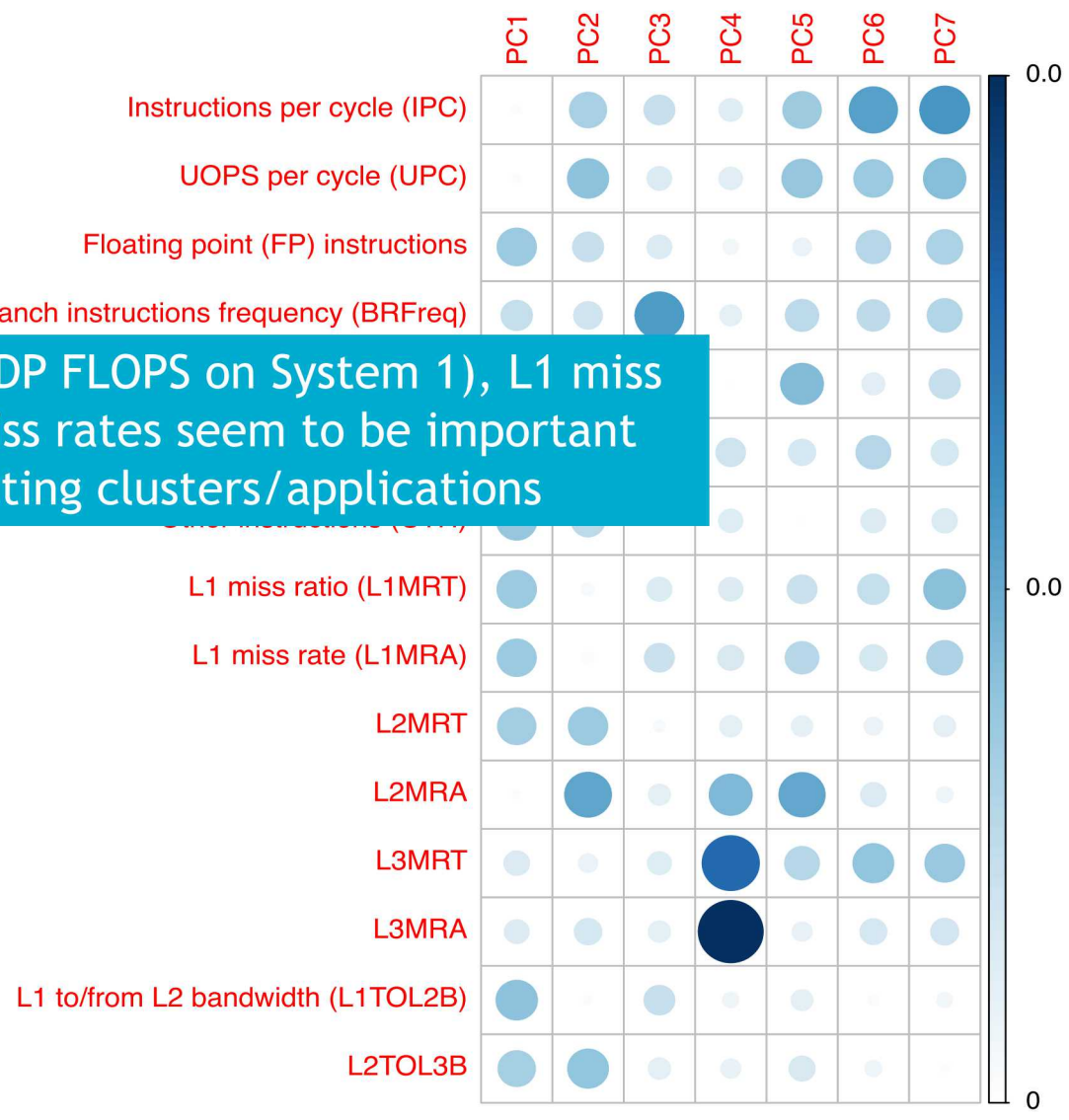


Which Characteristics are Important in Clustering?



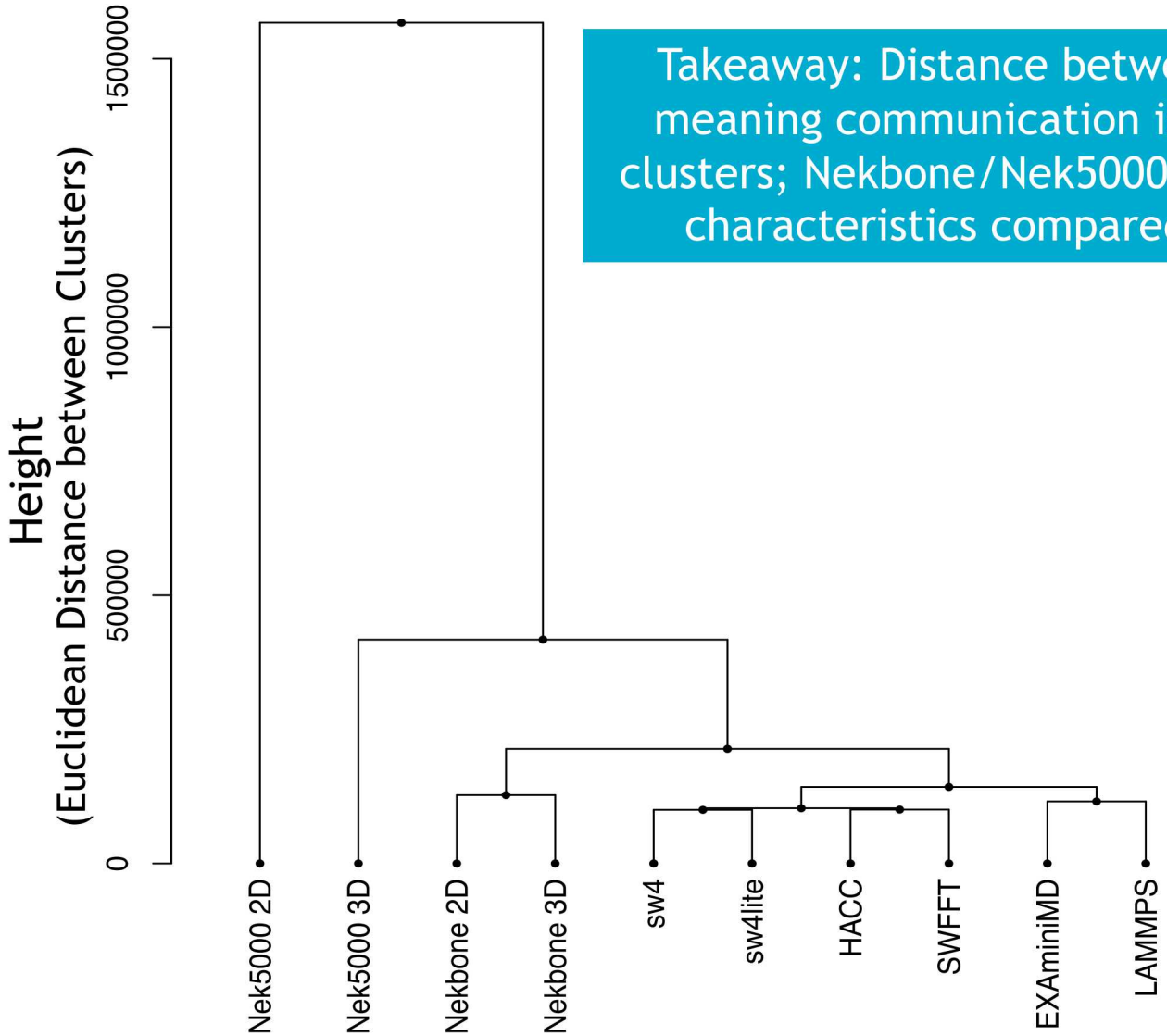
System 1

Takeaway: Instruction mix (DP FLOPS on System 1), L1 miss rates and ratios, and L3 miss rates seem to be important metrics for differentiating clusters/applications



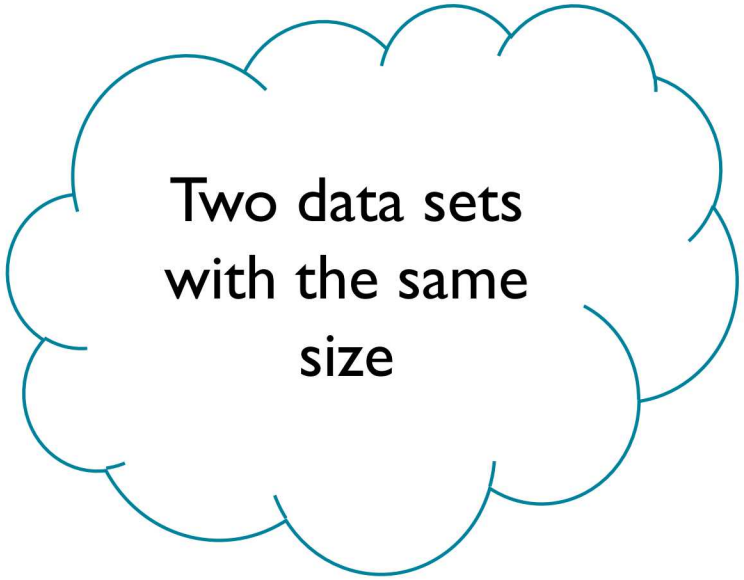
System 2

Clustering Communication Data



Takeaway: Distance between clusters is relatively large, meaning communication is likely very different between clusters; Nekbone/Nek5000 have distinctive communication characteristics compared to other proxy/parent pairs

Parent filtered by proxy



Proxy		
Src	Dst	#Msg
0	1	152120
0	10	153422
0	100	1302
68	60	13020
68	65	13020
68	67	153422
68	69	1302

Parent		
Src	Dst	#Msg
0	1	35046
0	10	0
0	100	0
68	60	0
68	65	0
68	67	62
68	69	5887

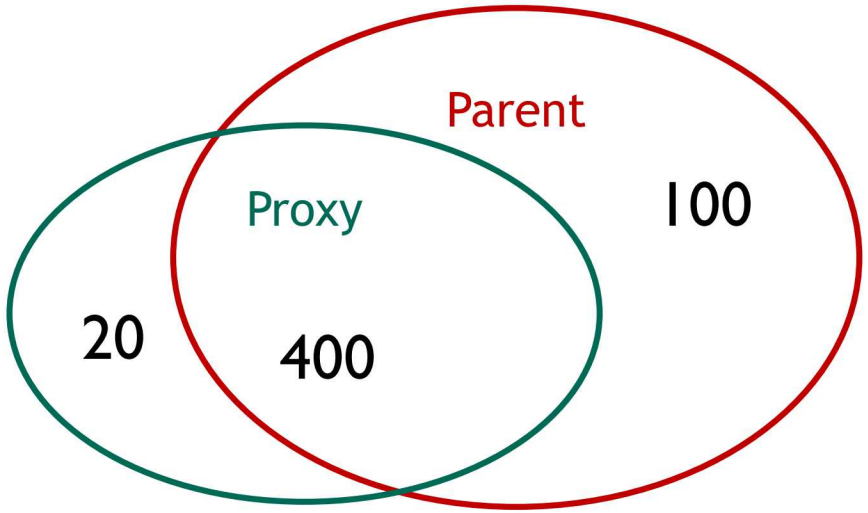
Key project goals: Curation and Assessment

- ECP proxy app **suite** to comprise **proxies** developed by ECP projects that **represent the most important features** (especially performance) of exascale **applications**
  - A need to understand if important performance features in parent exist in proxy
    - Characterization and performance comparison
- Improve the quality of proxies created by ECP and maximize the benefit received from their use
  - A need to feedback results of characterization and performance comparison to developers for improvement

Representative suite of proxies should be minimal and accurate with respect to representing important performance features of parent applications

Similarity Metrics: % Parent Covered by the Proxy

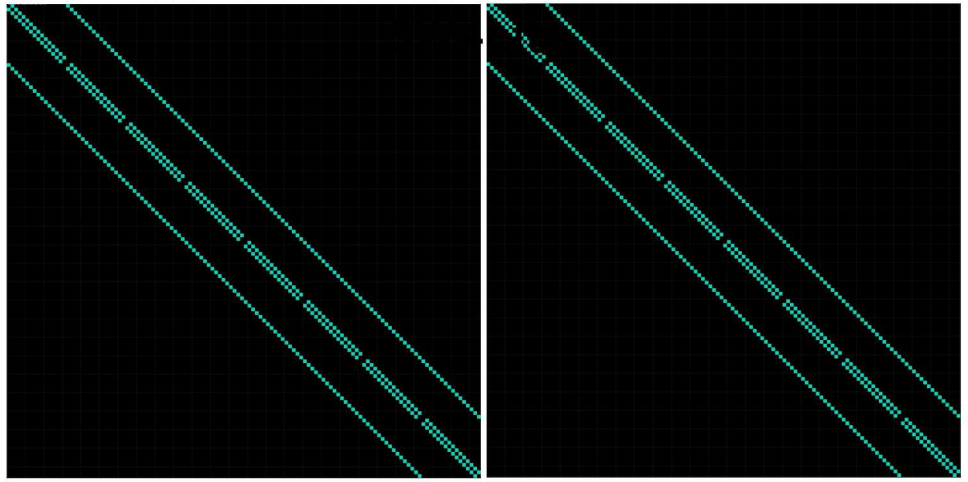
1. Percentage of parent communication that is covered by the proxy
   - by number of pairs



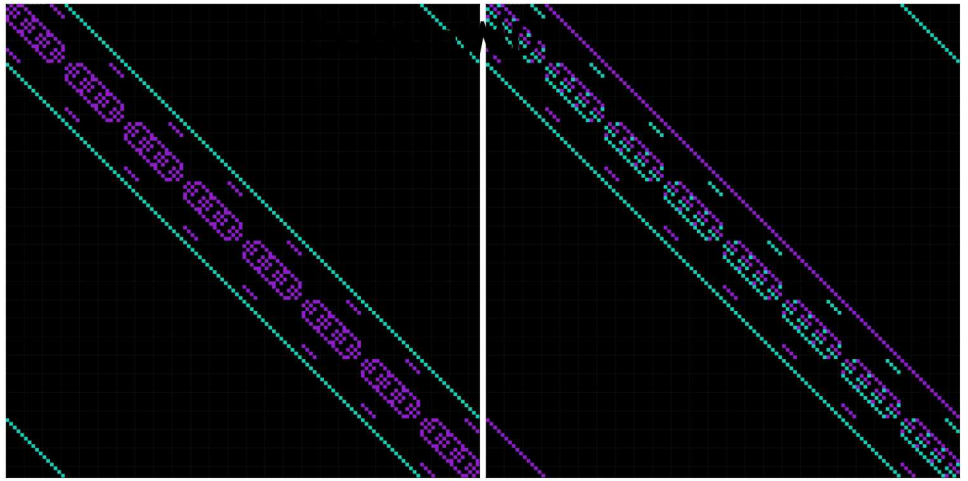
= 400/500*100
= 80%

Results

SW4 /



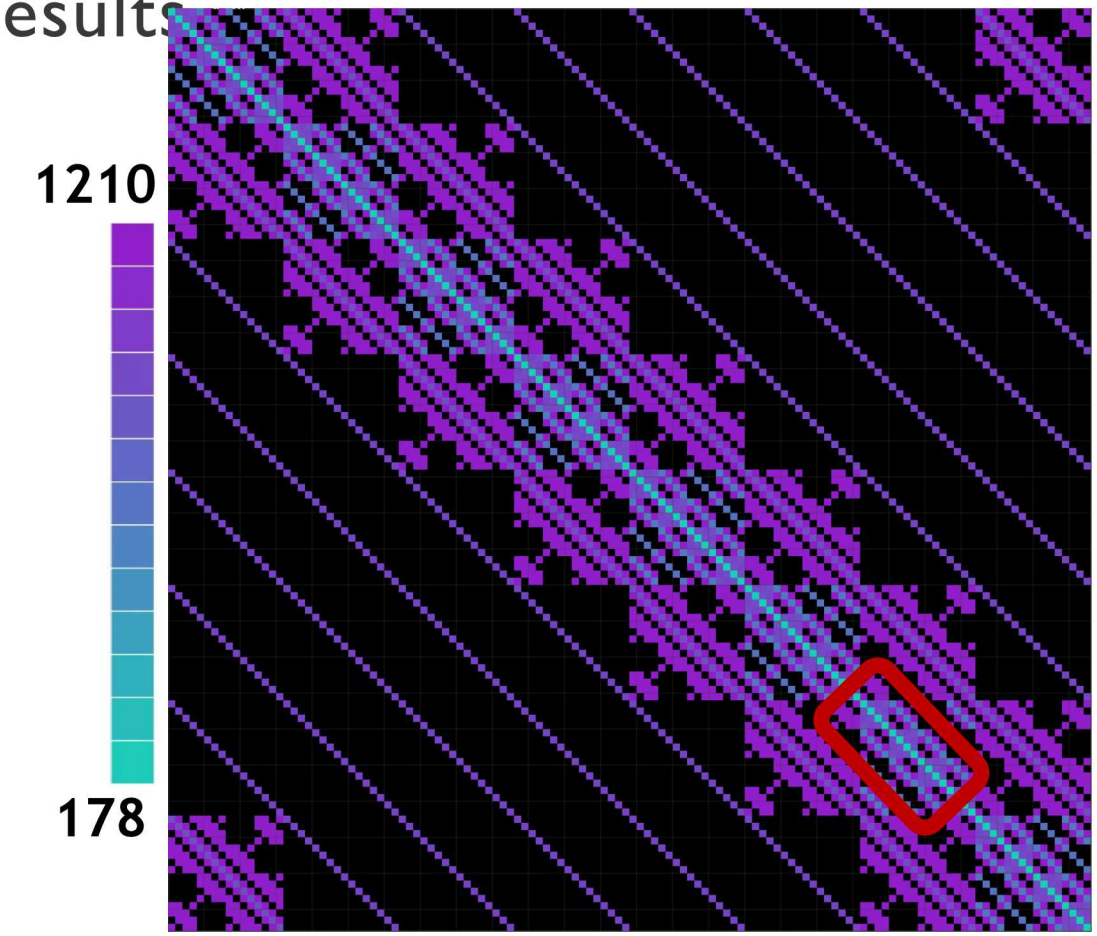
LAMMPS /



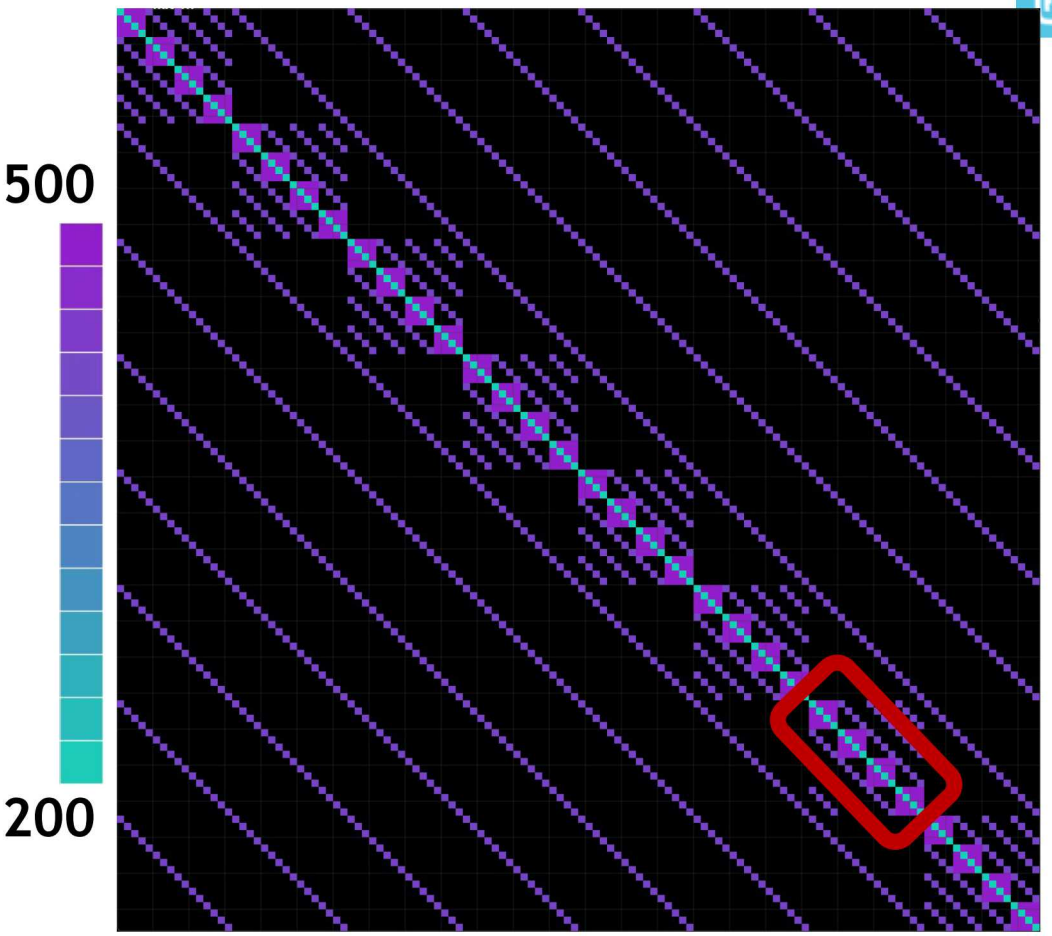
Parent/Proxy	Parent in Proxy		Proxy in Parent		Full Set		Parent in Proxy		Proxy in Parent	
	#msg	#pair	#msg	#pair	PCorr	SCorr	PCorr	SCorr	PCorr	SCorr
LAMMPS/ ExaMMD	100	100	100	100	0	0	0	0	0	0
Nek5K 2D/ Nekbone 2D	99.9	57.4	37.5	62.8	0	0.06	-0.47	-0.05	0.55	0.93
Nek5K 3D/ Nekbone 3D	99.9	51.4	58.0	68.4	-0.1	-0.05	-0.65	-0.23	0.04	0.49
SW4/ SW4lite	100	100	100	100	1	1	1	1	1	1
HACC/ SWFFT	51.7	29.4	71.4	71.4	0.58	0.31	0.61	0.28	0.87	0.81

Results

HACC



SWFFT



Parent/Proxy	Parent in Proxy		Proxy in Parent		Full Set		Parent in Proxy		Proxy in Parent	
	#msg	#pair	#msg	#pair	PCorr	SCorr	PCorr	SCorr	PCorr	SCorr
HACC/ SWFFT	68.7	41.1	100	100	0.97	0.99	0.92	0.84	0.97	0.99

Takeaways

- Nekbone is likely not a good proxy of Nek5000
  - Memory behavior and FLOPs very different
- SW4lite is a very good proxy of SW4
  - Code is very close to identical
  - Haswell and Broadwell clustering supports this
- SWFFT/HACC and ExaMiniMD/LAMMPs
  - Cluster together but the height (Euclidian distance) is large ➔ from method, can't really make solid conclusion
- Heatmap conclusion
  - Instruction mix, DP FLOPs, L1 and L3 cache behavior important distinguishing characteristics
- Communication clustering was less informative than examining communication pattern similarity
  - Our statistical method identified mismatch in problem decomposition of SWFFT and HACC

Hierarchical clustering height is too ambiguous to make strong conclusions without having more data. We need a better method to really understand similarity in node behavior.