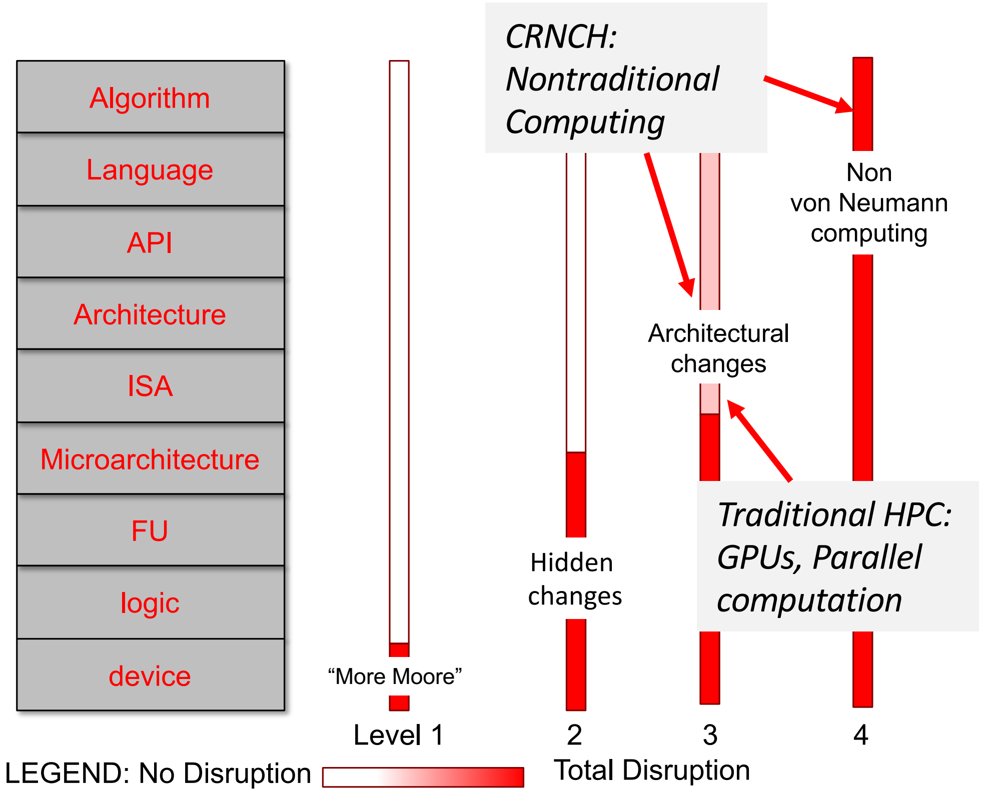
Experimental Insights from the Rogues Gallery

Jeffrey Young, Jason Riedy, Prasanth Chatarasi, Sriseshan Srikanth, Thomas M. Conte, Vivek Sarkar

Georgia Tech Computer Science

Presented by: Jeffrey Young
CRNCH Rogues Gallery Co-Director
November 7th, 2019

Center for Research into Novel Computing Hierarchies (CRNCH)



- CRNCH is focused on developing tools and techniques for using "post-Moore" computing technologies, such as quantum, neuromorphic, approximate, and thermodynamic computing.

- Disruptive technologies require new hardware, programming models, benchmarks, and training!

Rogues Gallery – Enabled Research

Georgia Tech Computer Science

Rogues Gallery – Enabled Research Pillars

Performance Portability

- Kokkos
- RISC-V
- Tensor, Streaming Graph APIs

Code Transformations

- Polyhedral compilation
- Habanero runtime

Data Management and Migration

- Sensitive data analysis
- In-situ analysis
- Integration of novel memories

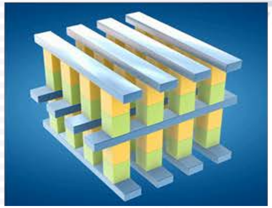
Instrumentation and Introspection

- Power, thermal analysis
- Network introspection

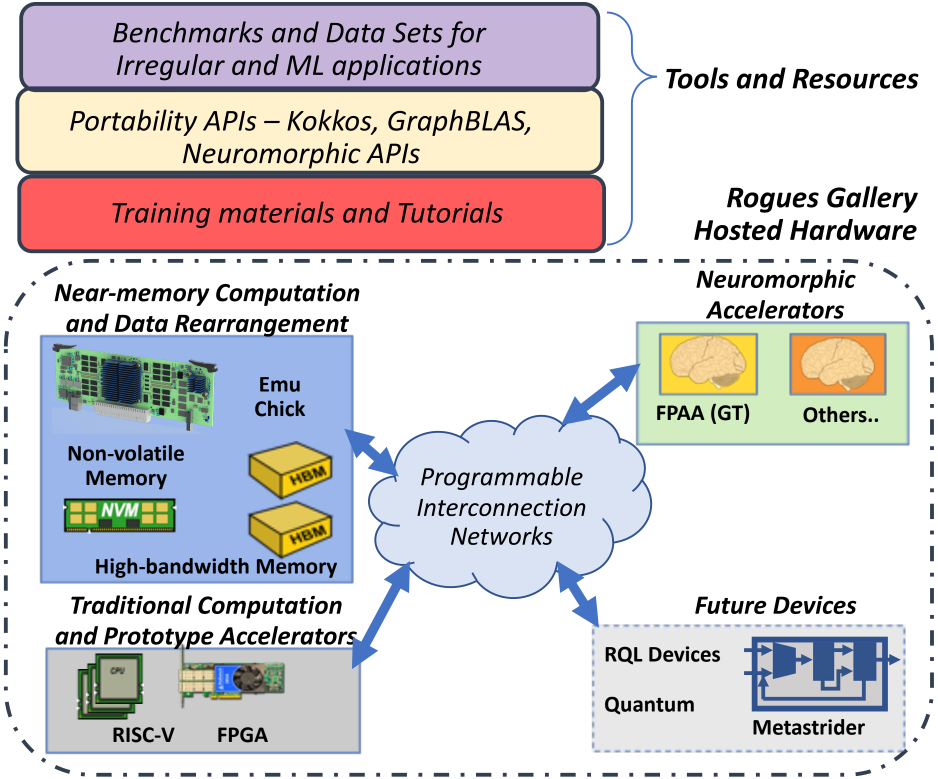
Rogues Gallery Deployment - Addressed Research Challenges

- Access management of embedded and novel devices
- Integration of extremely heterogeneous components
- Metrics and measurements for usage of novel hardware

"Rogues" – Unique hardware possibilities for post-Moore computing

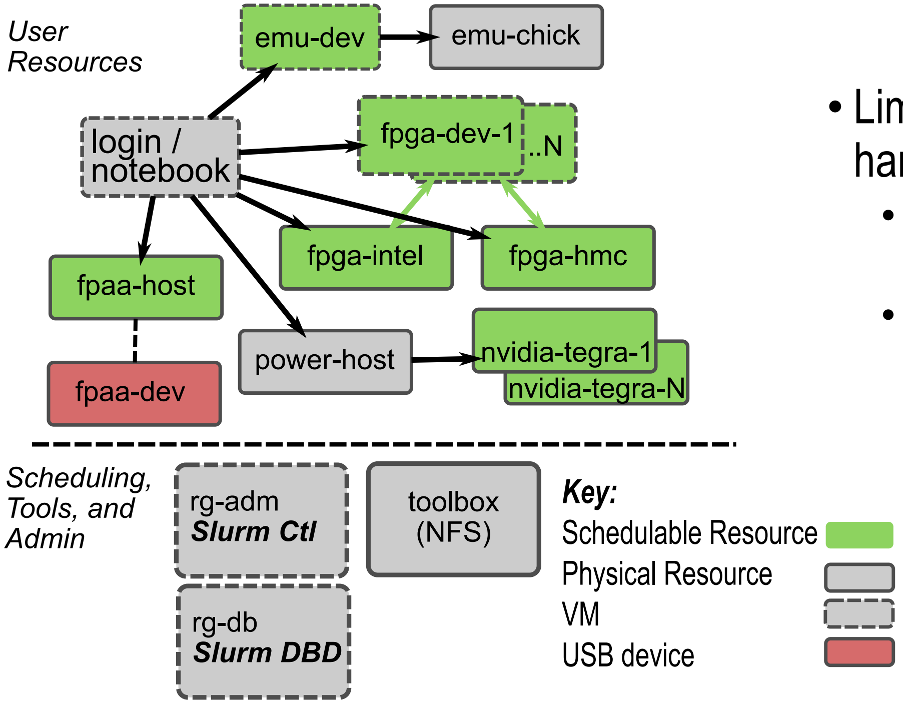


CRNCH Testbed – Rogues Gallery


Benchmarks and Data Sets for Irregular and ML applications
Portability APIs – Kokkos, GraphBLAS, Neuromorphic APIs
Training materials and Tutorials
Tools and Resources
Rogues Gallery Hosted Hardware
Near-memory Computation and Data Rearrangement
Emu Chick
Non-volatile Memory
NVM
HBM
HBM
High-bandwidth Memory
Neuromorphic Accelerators
FPAA (GT)
Others..
Programmable Interconnection Networks
Traditional Computation and Prototype Accelerators
CPU
RISC-V
FPGA
Future Devices
RQL Devices
Quantum
Metastrider

- The Rogues Gallery is a community testbed for new and unique architectures that will help pave the way to performance in the post-Moore's Law era.
  - Free for students, industry, and government partners to use.
  - Hardware is supplemented by tools and APIs developed by researchers at Georgia Tech.
    - Habanero runtime, Kokkos API for the Emu, Spatter and STINGER benchmarks
  - Training and education are a key pillar of the testbed with tutorials on new hardware and an associated VIP class for GT students.

CRNCH Testbed – Rogues Gallery



User Resources

emu-dev

emu-chick

login / notebook

fpga-dev-1

..N

fpga-intel

fpga-hmc

fpaa-host

fpaa-dev

power-host

nvidia-tegra-1

nvidia-tegra-N

Scheduling, Tools, and Admin

rg-adm
Slurm Ctl

toolbox
(NFS)

rg-db
Slurm DBD

Key:

Schedulable Resource	🟩
Physical Resource	▭
VM	⬚
USB device	🟥

- Limit the use of physical resources to novel hardware
  - Use VMs wherever possible for tools, debugging, support
  - Schedule everything! *(In progress)*

Will Powell, Jason Riedy, Jeffrey S. Young, and Thomas M. Conte. 2019. Wrangling Rogues: A Case Study on Managing Experimental Post-Moore Architectures. PEARC 2019.

Key Insights from the Rogues Gallery

- Post-Moore computing research is based on cross-stack environments with varying levels of software support
  - Emu Chick – has several libraries but limited debugging support; user-driven data layouts
  - FPGAs – many high-level synthesis techniques but limited support for traditional libraries

- Small “wins” often provide an entry point for larger questions
  - The Emu seems to be well-suited for SpMV and graph analytics. How do we scale this up for larger applications with poor data layouts?
  - The FPAA can efficiently implement hhNeurons – how do we build and program a larger array?

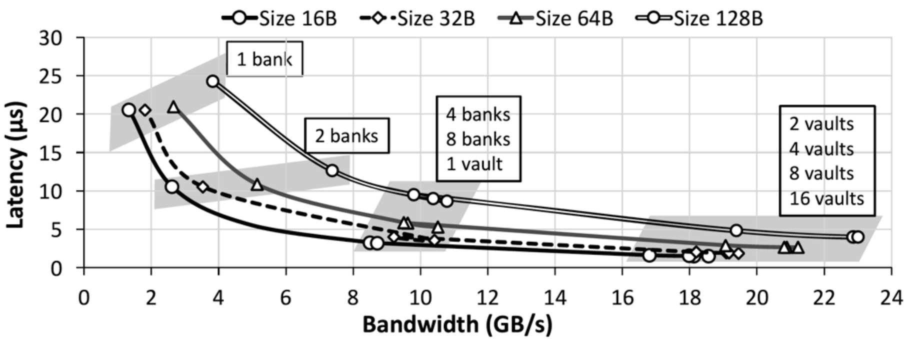
- We should spend more of our time on developing:
  - Tools, runtimes, and benchmarks to evaluate novel architectures
  - Education and training

Case Studies from the RG

- Evaluating novel memory architectures with reconfigurable computing
- Characterization and benchmarking of the Emu Chick
- Neuromorphic Computing – Field Programmable Analog Array
- Where should be spending our time and effort?
  - Tools, runtimes, and benchmarks
  - Education and training

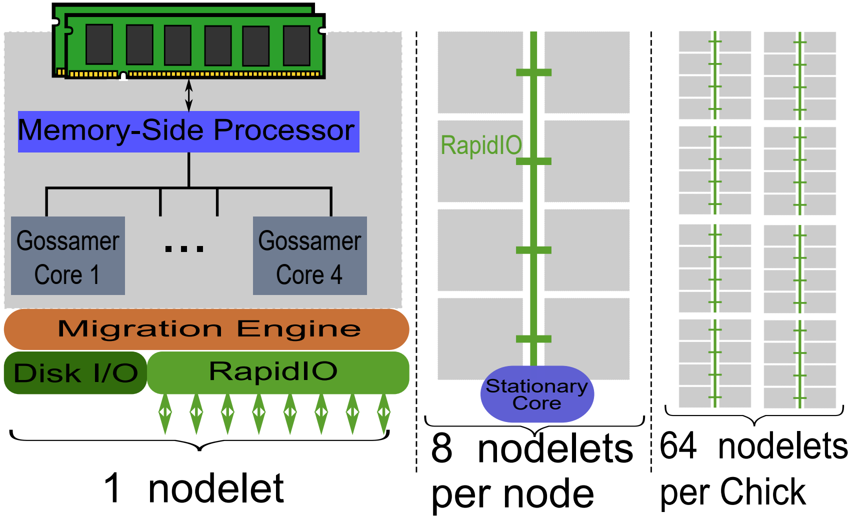
Reconfigurable Architectures

- Hybrid memory cube provided a case study in benchmarking a low-level novel architecture
- Detailed characterizations looked at thermal, power, and performance trade-offs for using HMC
    - Packet-based memory accesses provide explicit trade-off between latency and bandwidth



Hadidi, Ramyad, Bahar Asgari, Jeffrey Young, Burhan Ahmad Mudassar, Kartikay Garg, Tushar Krishna, and Hyesoon Kim. "Performance implications of NoCs on 3D-stacked memories: Insights from the hybrid memory cube." ISPASS 2018

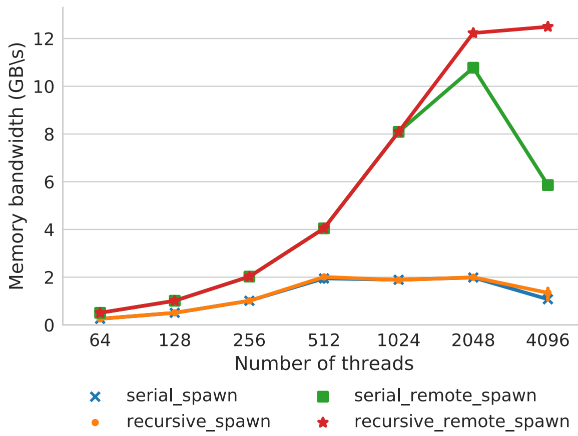
Emu Chick – Overview



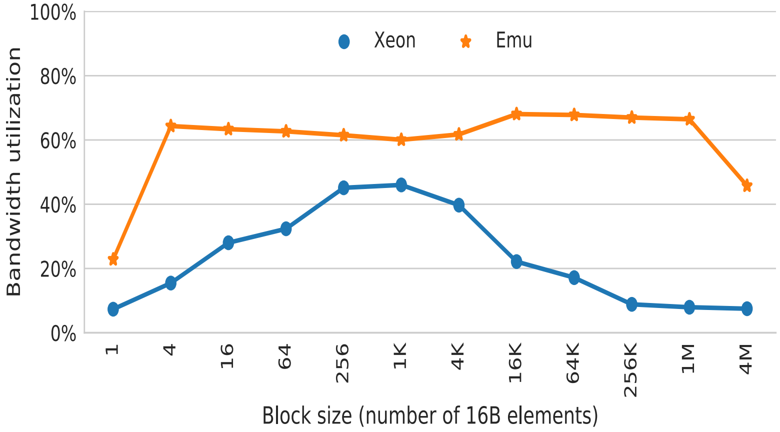
The Emu system migrates lightweight threads to the data rather than using cache-based accesses. The “Chick” combines 8 FPGA boards in one chassis.

Emu Chick – Microbenchmarks

STREAM Multi-node Benchmark



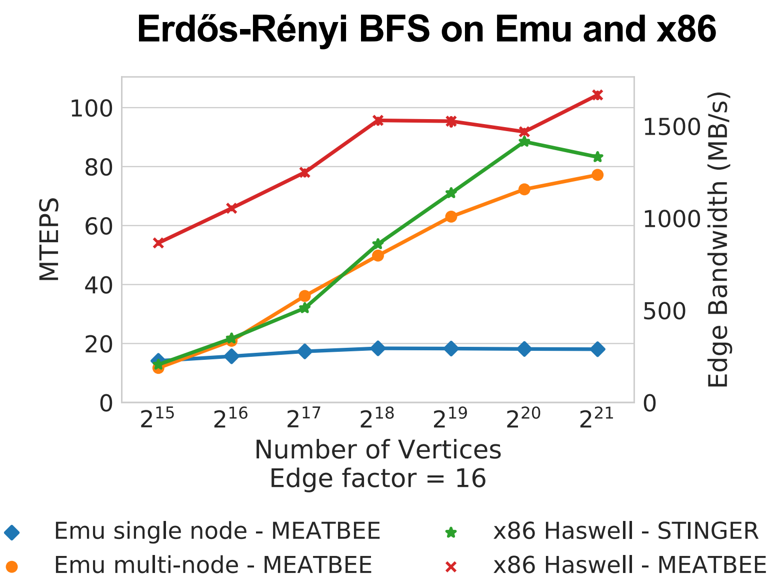
Pointer Chasing Multi-node Benchmark

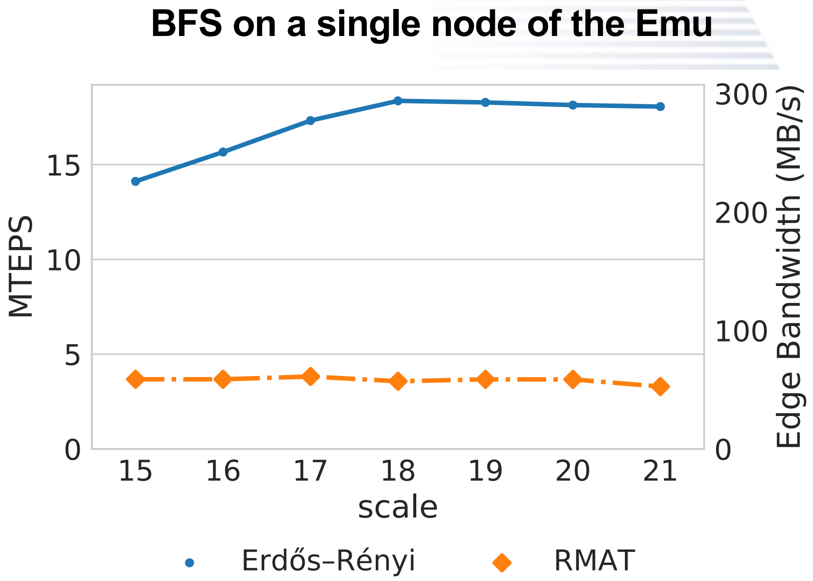


- Initial Emu microbenchmarking results show good STREAM bandwidth and GUPS-like performance that is size-invariant

Jeffrey S. Young, Eric Hein, Srinivas Eswar, Patrick Lavin, Jiajia Li, Jason Riedy, Richard Vuduc, Tom Conte, "A microbenchmark characterization of the Emu chick, Parallel Computing", Volume 87, 2019.

Emu Chick – Graph Analytics





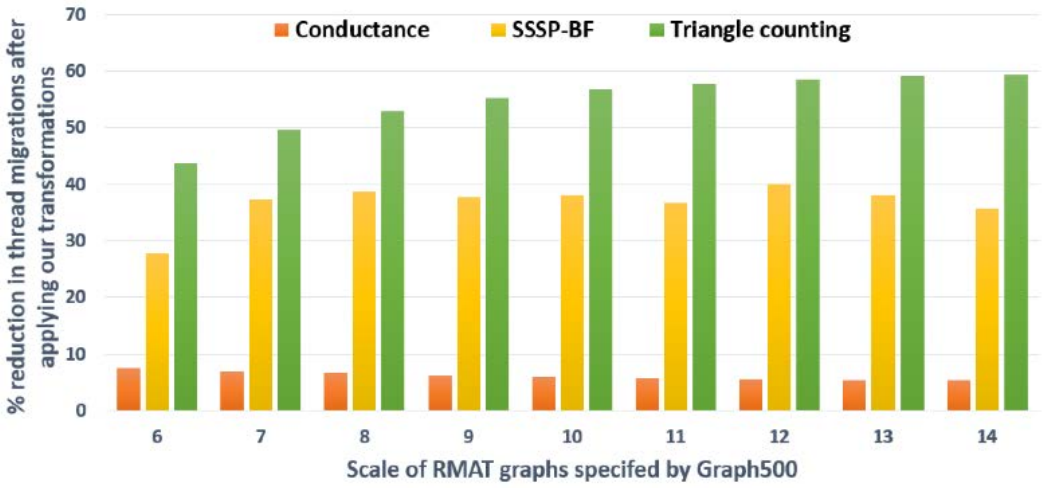
- The Emu Chick performs well at scale for balanced graphs, but unbalanced graphs suffer from large performance penalties due to hotspots that incur thread migrations.

Hein, Eric, Srinivas Eswar, Abdurrahman Yaşar, Jiajia Li, Jeffrey S. Young, Thomas M. Conte, Ümit V. Çatalyürek, Rich Vuduc, Jason Riedy, and Bora Uçar. "Programming Strategies for Irregular Algorithms on the Emu Chick." TOPC 2019

Emu – Compiler-oriented Optimizations

- Performance of basic graph algorithms like SSSP, TC, and Bellman-Ford also can suffer from imbalance and spurious thread migrations
- Compiler techniques like loop fusion, edge flipping, and the addition of remote atomic operations reduce migrations and improve performance overall.

% Reduction in Migrations with all Optimizations

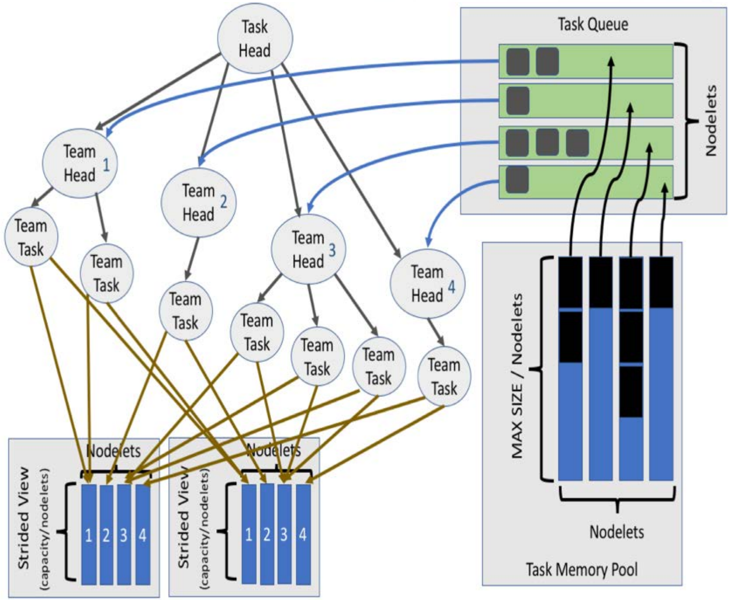


Chatarasi, Prasanth, and Sarkar, Vivek. "A Preliminary Study of Compiler Transformations for Graph Applications on the Emu System." In Proceedings of the Workshop on Memory Centric High Performance Computing, pp. 37-44. ACM, 2018.

Emu Chick – What have we learned?

- We need a better way to map data layouts to the Emu system!
  - It's too tough for application/algorithm designers to discern on their own how to do data placement and partitioning.
  - Replication of data values only works for small data sets.

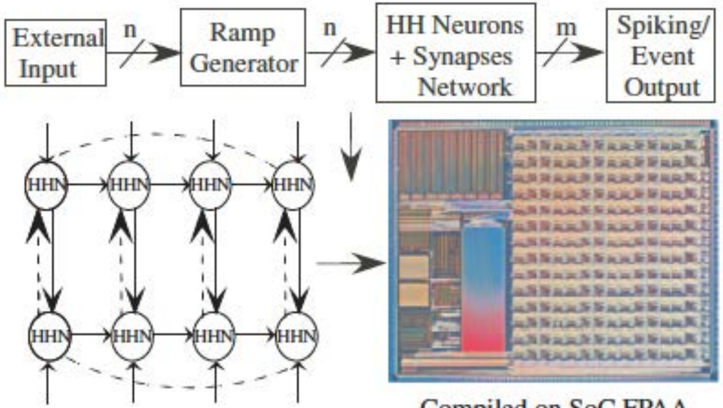
- **One approach:** Use the Kokkos performance portability API to provide options for multiple data layouts
  - "PaperWasp" implementation of BFS for the Emu has been ported to Kokkos using the Kokkos task-based parallelism



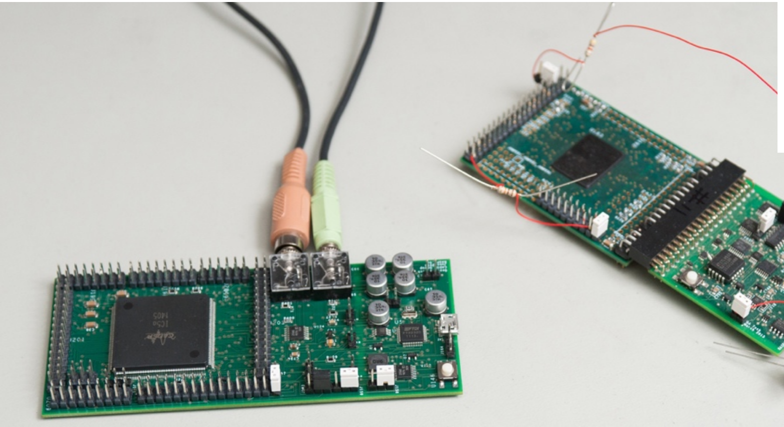
In progress – work by J. Young, J. Miles (Sandia), E. Hein (Emu)

Neuromorphic Computing

- The Field Programmable Analog Array (FPAA) provides an opportunity to implement hhNeuron structures efficiently (low-power) in a mixed digital-analog device.
- Positive results lead us to ask “How we can map higher-level algorithms to hhNeuron structures? How many neurons can fit on a 350nm FPAA chip?”



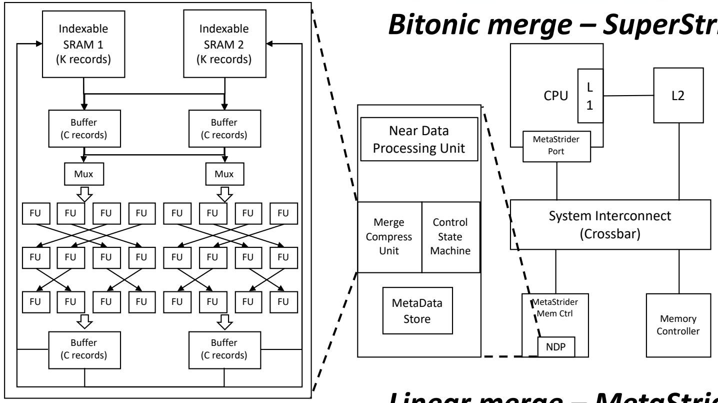
Compiled on SoC FPAA



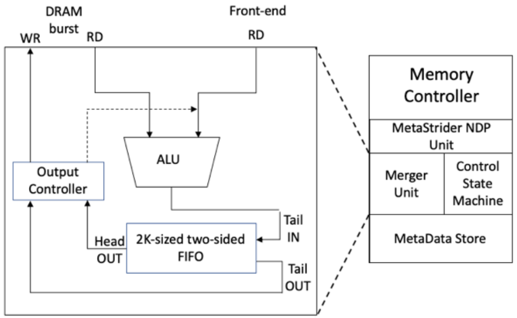
A. Natarajan and J. Hasler, "Implementation of Synapses with Hodgkin Huxley Neurons on the FPAA," 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 2019, pp. 1-5.

Sparse Accelerators - Strider

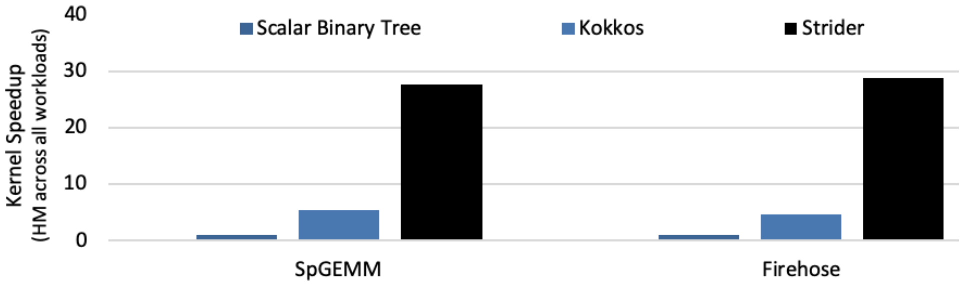
Bitonic merge – SuperStrider



Linear merge – MetaStrider



Strider Simulated Speedup versus HW/SW Techniques



- Two different architectures built into or close to DRAM
  - Bitonic merge provides sorting of data as it is accessed/used
  - Linear merge operates efficiently on pre-sorted input vectors
  - Provides dramatic speedups for algorithms like SpGEMM's accumulation phase and streaming data access

- *E. P. DeBenedictis, J. Cook, S. Srikanth, and T. M. Conte, "Superstrider associative array architecture," in 2017 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 2017, pp. 1–7.*
- *S. Srikanth, A. Jain, J. Lennon, T. Conte, E. DeBenedictis, and J. Cook, "Metastrider: Architectures for scalable memory centric reduction of sparse data streams," ACM Transactions on Architecture and Code Optimization (TACO), 2019.*

Case Studies - Takeaways

- Even small benchmarking studies can lead to avenues for further research and investigation
  - Examples: Emu characterization, FPAA designs

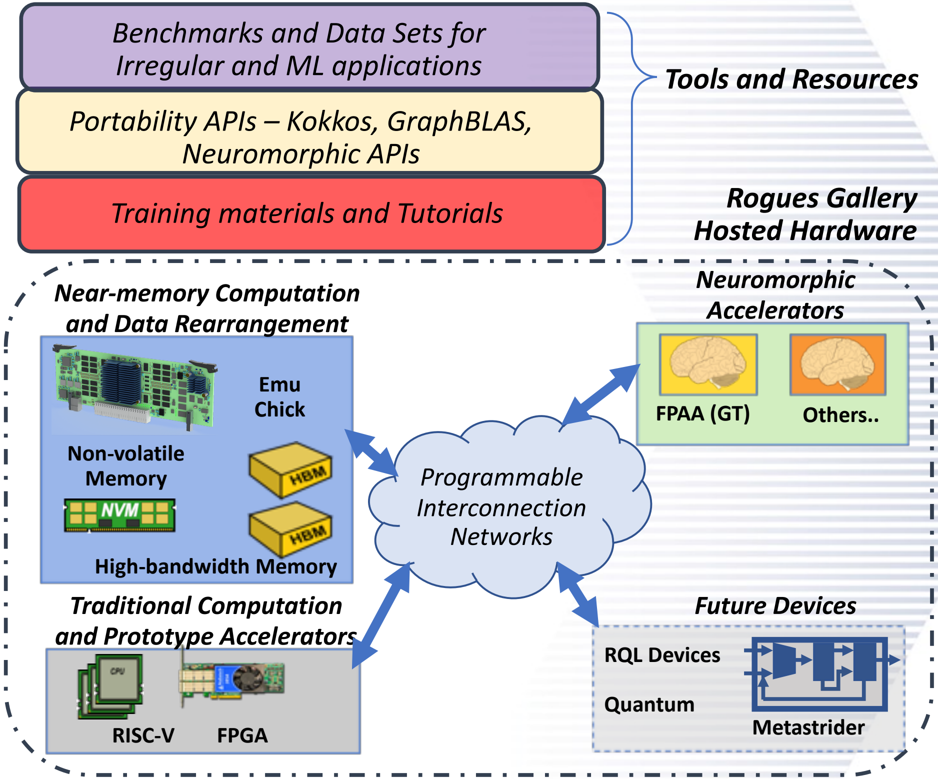
- Results vary and depend on the level of software and tool support
  - Some architectures can only run microbenchmarks or a single function

So… where should be spending our time and effort?

- Tools, runtimes, and benchmarks
  - Kokkos portability API, Habanero runtime, Spatter and STINGER benchmarks
- Education and training
  - Tutorials and undergraduate research

Rogues Gallery - Benchmarks

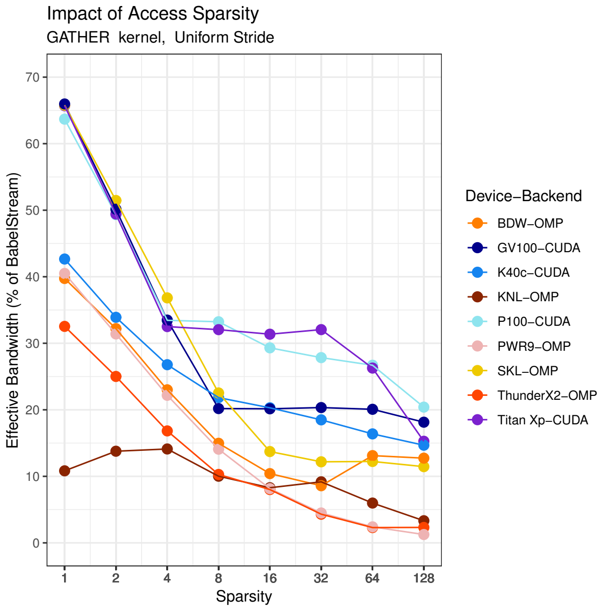
- Pointer chasing (GUPs-like)
- Local, global STREAM
- Streaming graph analytics - STINGER, PaperWasp (Emu), Hornet (NVIDIA)
- Spatter – tunable gather/scatter for multiple platforms
- ParTI – tensor decomposition operations


Benchmarks and Data Sets for Irregular and ML applications
Portability APIs – Kokkos, GraphBLAS, Neuromorphic APIs
Training materials and Tutorials
Tools and Resources
Rogues Gallery Hosted Hardware
Near-memory Computation and Data Rearrangement
Emu Chick
Non-volatile Memory
NVM
HBM
HBM
High-bandwidth Memory
Neuromorphic Accelerators
FPAA (GT)
Others..
Programmable Interconnection Networks
Traditional Computation and Prototype Accelerators
CPU
RISC-V
FPGA
Future Devices
RQL Devices
Quantum
Metastrider

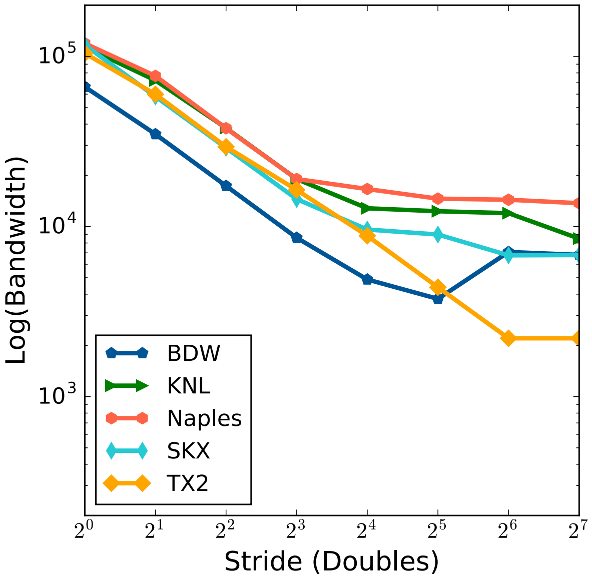
Benchmarking – Spatter (Spatter.io)

Georgia Tech Computer Science

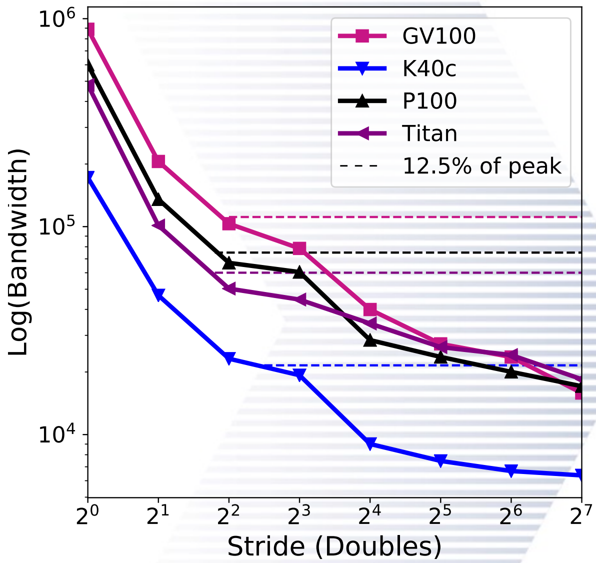
Gather - CPU/GPU



Gather – CPU OpenMP

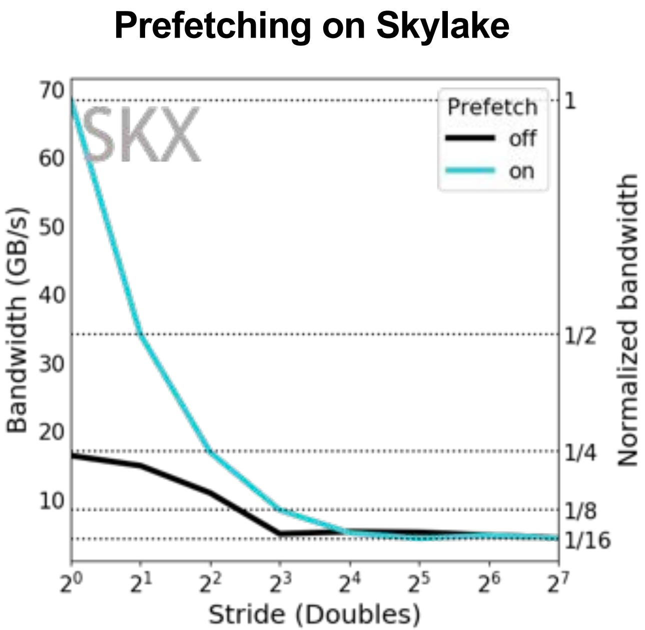


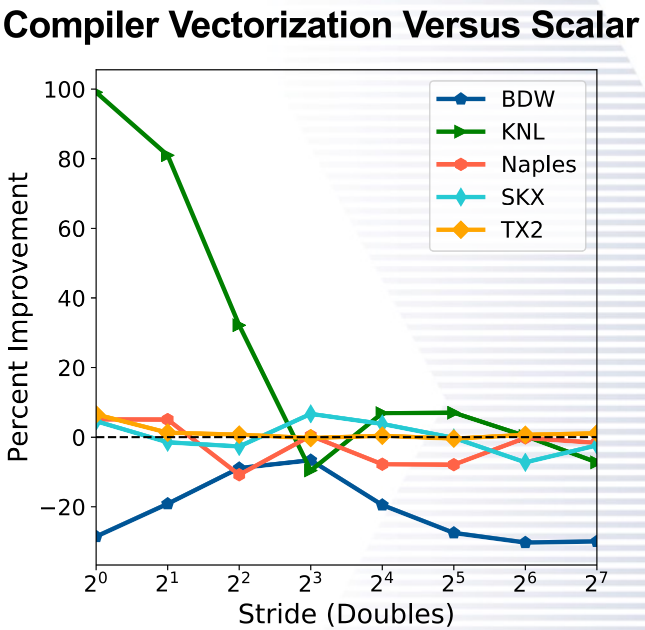
Scatter – GPU CUDA



- Spatter allows us to evaluate strided or pattern-based on CPU, GPU, or other novel architectures
- OpenMP, CUDA Backends; SyCL, HIP, and Kokkos backends coming soon!

Benchmarking – Spatter (Spatter.io)





- Even a relatively simple benchmark can be used for multiple purposes: architecture investigations, compiler evaluation, and application characterization

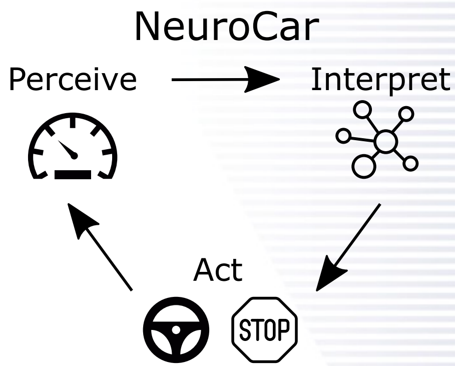
Lavin, Patrick, Jason Riedy, Rich Vuduc, and Jeffrey Young. "Spatter: A Benchmark Suite for Evaluating Sparse Access Patterns." arXiv preprint arXiv:1811.03743 (2018).

Education – Tutorials

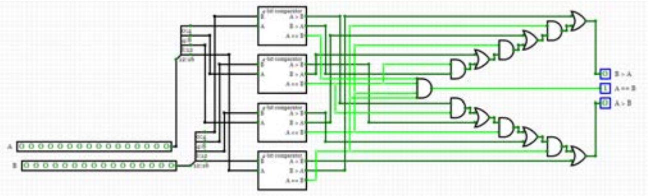
Training is key to engaging students and potential users

- Neuromorphic tutorial in April 2018 focused on the Field Programmable Analog Array (FPAA); hands on experience with using the FPAA
- Emu tutorials at ASPLOS and PEARC 2019 used Jupyter notebooks and remote access to the Emu to provide a “crash course”

Vertically Integrated Projects (VIP) Team



- Undergraduate research opportunity for credit; teams are self-directed with guidance from faculty.
- Current projects:
  - NeuroCar – implement sensing and control using SNNs with Nengo FPGA platform; replicate results of the autonomous GT Rally Car with lower power
  - Qubit allocation optimization – evaluate techniques using IBM's Q experience and ORNL's XACC and attempt to build a linear systems algorithm approach to test possible solutions
  - No-history branch prediction – sort the register file on the fly to assist with branch prediction and limit security vulnerabilities



Rogues Gallery – Takeaways

We think a collaborative testbed provides opportunities to leverage:

- Cross-cutting work in architecture implementation, compiler and runtime design, benchmarking, and algorithm design
  - RG has supported benchmarking, API design, and compiler techniques to improve the Emu environment.
- Economy of scale in terms of utilizing scarce resources: hardware funding and researcher and student time
  - Rogues Gallery supports **50 GT users** across **2 colleges and 3 departments**; many novel architectures have led to collaborative publications
- Interactions with government labs and corporations investigating next-generation hardware
  - We host **18 external users** from **13 different institutions**

Rogues Gallery – Engagement Opportunities

Georgia Tech Computer Science

Request an account on the Rogues Gallery

- http://crnch.gatech.edu/request-rogues-access

Attend and/or speak at the **CRNCH Summit (January 31st, 2020)**

- We are always looking for interesting speakers that would like to connect with GT computing students and researchers. http://crnch.gatech.edu/content/crnch-summit

Corporate sponsorships/partnerships

- CRNCH Rogues Gallery is set up to help test computing hardware for interested external industry partners as part of sponsorship and partnership agreements.

Vertically Integrated Projects (VIP) team

- Suggest project ideas for our undergraduates to work on!
- Learn more at https://www.vip.gatech.edu/teams/future-computing-rogues-gallery

Thank you!

Georgia Tech Computer Science

Advanced Architecture Testbeds Birds of a Feather at SC19:

Thursday, November 21st 12:15pm - 1:15pm, Room 710

Rogues Gallery:

http://crnch.gatech.edu/request-rogues-access

CRNCH Summit:

http://crnch.gatech.edu/content/crnch-summit

Rogues Gallery VIP class:

https://www.vip.gatech.edu/teams/future-computing-rogues-gallery

Acknowledgments

This work is supported by:

- Hardware donations by Micron and Intel
- The NSF XScala project (ACI-1339745), the NSF SuperSTARLU project (OAC-1710371), and IARPA.
- Sandia National Laboratory supports the Rogues Gallery VIP undergraduate research class, and parts of this research have been funded through the Laboratory Directed Research and Development (LDRD) program at Sandia. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and 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-NA-0003525.