Near-Memory and CRNCH Rogues Gallery Introduction

Jeffrey Young • Jason Riedy • Patrick Lavin • Srinivas Eswar • Janice McMahon • Aaron Jezghani • Will Powell • Darryl Bailey

Georgia Institute of Technology

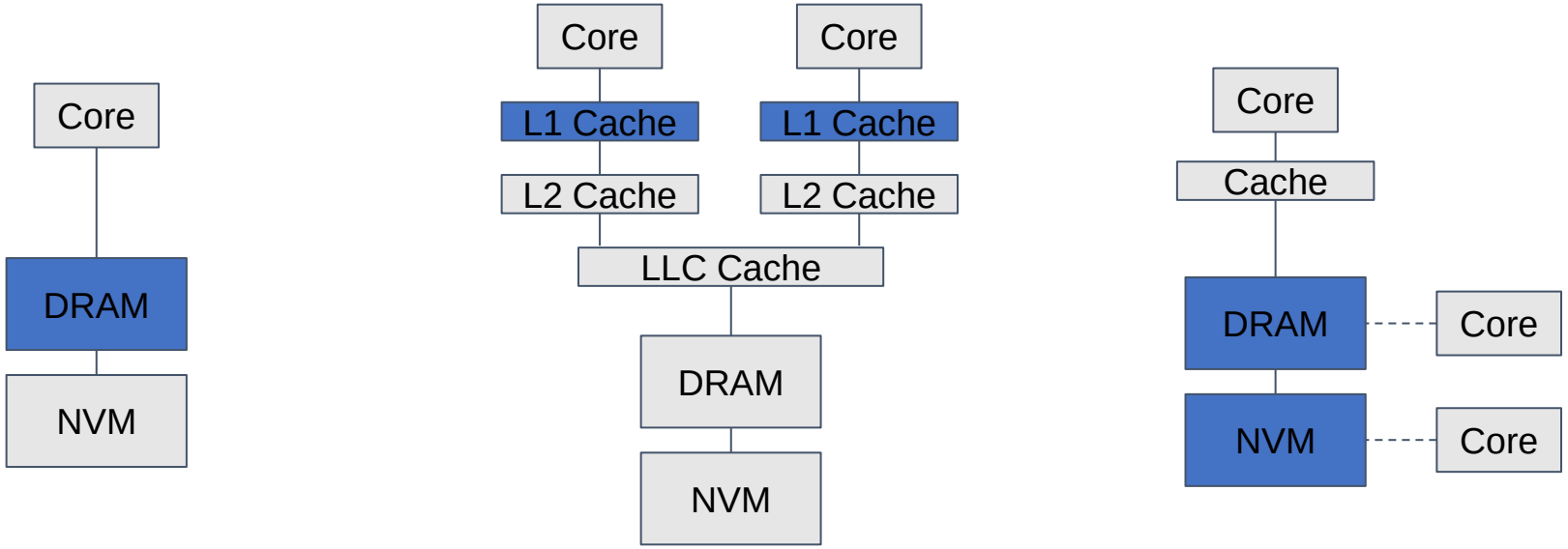
Lucata Corporation

Argonne National Lab

Georgia Tech

What is Near-Memory Computing?

1. Scaling the *memory wall* is a key challenge.
   - Memory technology is unable to keep pace with processors tech (latency and energy).
   - Limited pins mean that today's memory systems are unable to meet processor bandwidth demands.
2. Possible routes.
   - Traditional approaches involve a memory hierarchy to move the **working set** closer to processor.
   - **Near-memory computing** couples compute units closer to the data to minimize data movement.



Georgia Tech

Recent Examples of Near-Memory Accelerators

Research projects and industry have focused on near-memory accelerators with strong overlap with Processing in Memory (PIM). Examples include:

RISC-V PULP-based designs

- Azarkhish, *et al.* TPDS 2017
- Neurostream uses RISC floating point coprocessors to accelerate ConvNet workloads

HMC Samsung HBM-PIM (AquaBolt-XL)

- *Madhu, et al. ICSAS 2021*
- Places a Transport Triggered Architecture core near memory for AI acceleration

Western Digital In-Memory Processing for NVM (circa 2018)

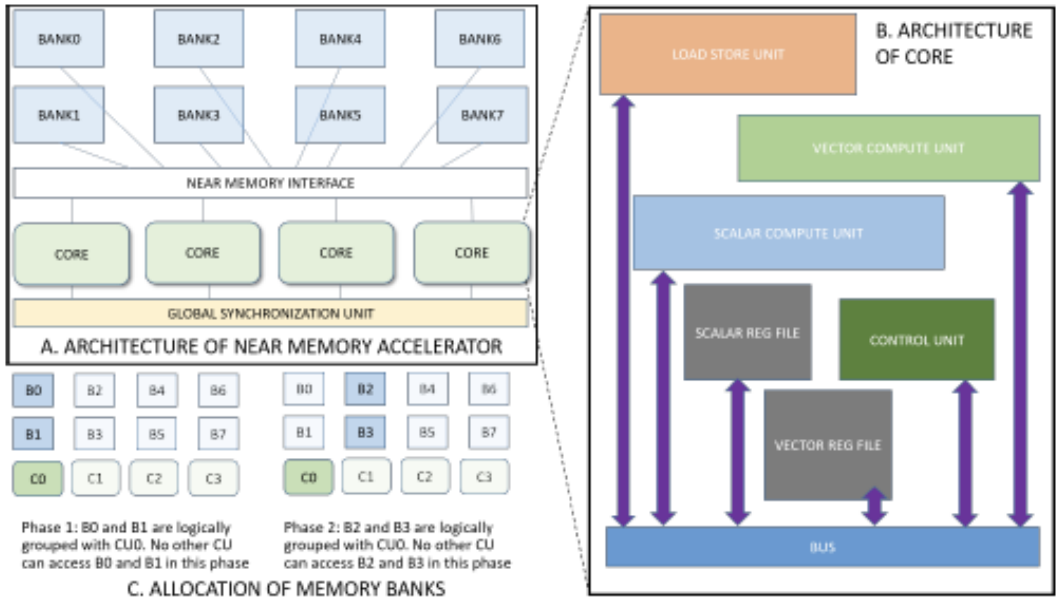


Image from Madhu, et al. "*Transport Triggered Near Memory Accelerator for Deep Learning", ISCAS 2021*

Georgia Tech

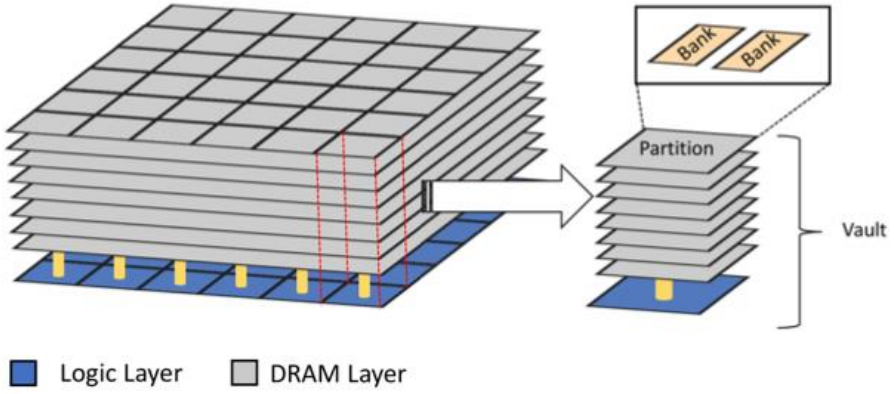
Near-Memory Computing for Graph Analysis

Near-memory computing is promising for graph applications:

- More bandwidth may be available closer to memory
- Latency is lower closer to memory

Several NMC designs have focused on graph applications:

- **GraphPIM** is a Hybrid Memory Cube-based design
  - Individual atomic instructions are offloaded to the vaults on the HMC
  - The overhead for individual instruction offload is high
- **TESSERACT** is another HMC-based design
  - An OOO processor is placed on each HMC vault. These processors can operate on local data and can use message passing to communicate with other vaults
  - OOO cores require a lot of computing resources on the chip



Micron's Hybrid Memory Cube

Image Source: Singh et al. (2019)

Near-Memory Computing Programming Models

A wide range of techniques have been applied to programming NMC devices.

Programming models for NMC can be classified along four axes (Fujiki et al. 2021):

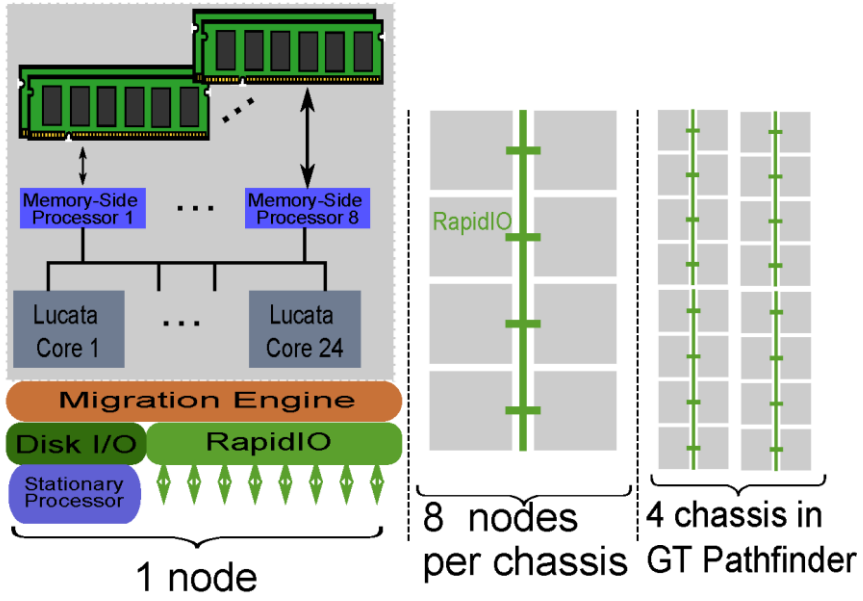
- Offload granularity
  - instruction, function, or application
- Programming complexity
  - ISA extension, custom API, exiting parallel language, automated offload detection
- Address translation and Cache Coherency
  - Translate on host (simplified NMC architecture)
  - Translate on NMC (operations can span pages)
  - Integrate with coherence network
  - Ignore coherence, and potentially fix conflicts after execution

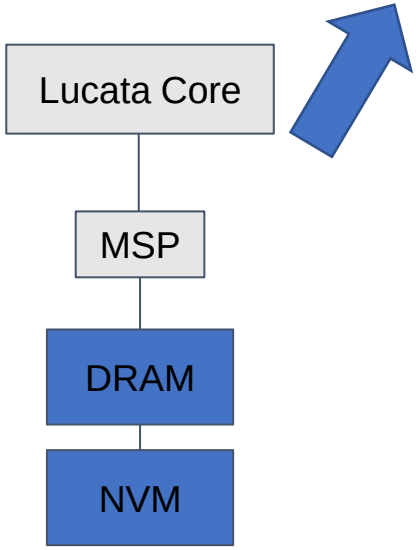
Models in use (Singh et al. 2019):

- Accelerator models
  - CUDA, OpenCL
  - Ex: TOM (Hsieh et al., SIGARCH '16) automatically identifies CUDA BBs that can be offloaded
- Data-parallel models
  - MPI, MapReduce, OpenMP
  - Ex: TESSERACT uses a message passing model
- Custom APIs
  - Ex: PLANAR (Barredo, SC '21) introduces an API that programmers use to offload entire functions to the NMC

The Lucata Pathfinder for Near-Memory Computing

- The Lucata Pathfinder system is based on the concept of “migrating threads”
  - Instead of a deep cache hierarchy, lightweight threads move to the correct data location
  - The memory side processor (MSP) assists with acceleration of specific in-memory operations like remote writes, addition, and atomics
    - The MSP replaces a traditional memory controller with added computational capabilities.





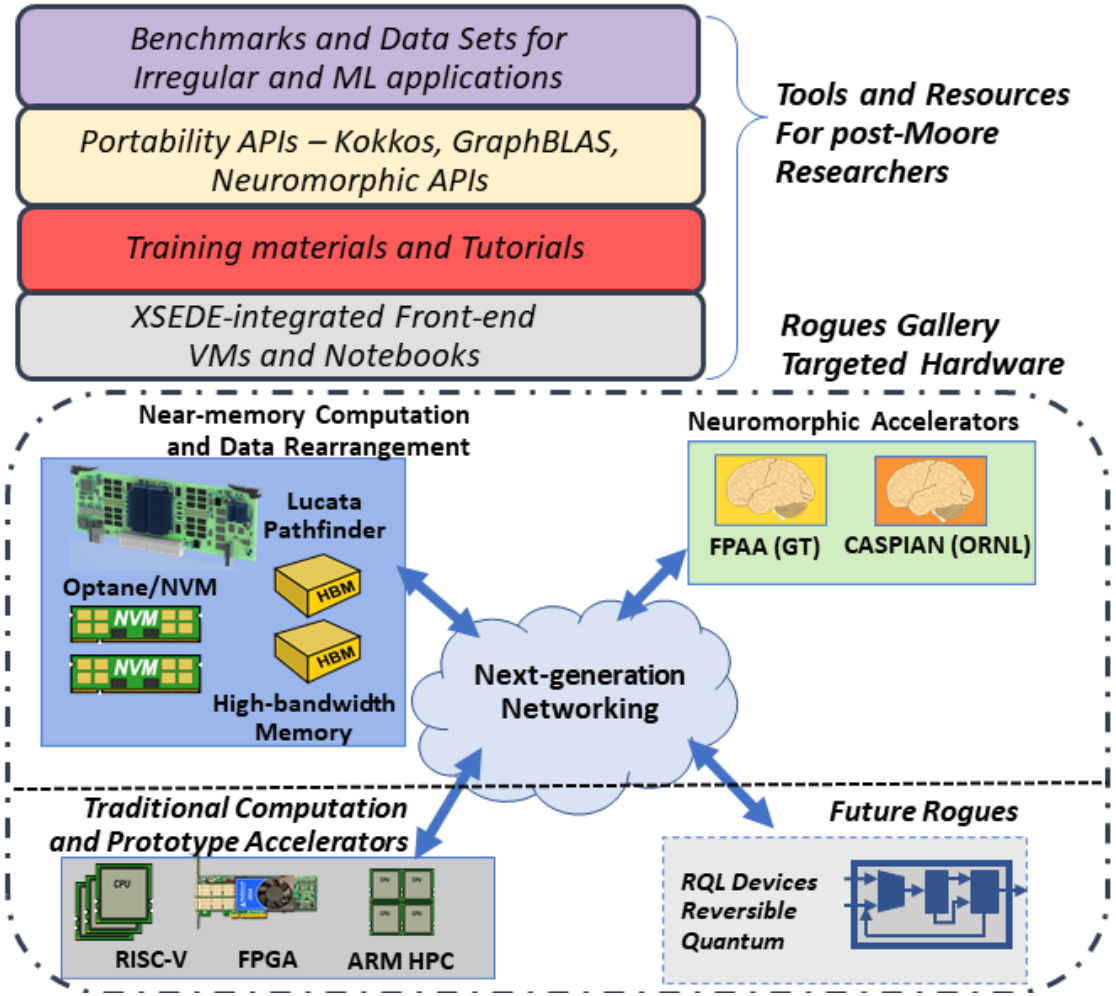
Georgia Tech

CRNCH Rogues Gallery Testbed

The Rogues Gallery is an NSF funded testbed that is focused on increasing access to novel architectures for CISE researchers.

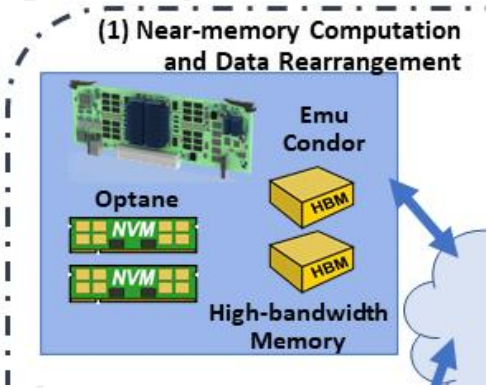
Focus not just on HW deployment but also training and APIs for usage

- Today's tutorial will cover web-based training and GraphBLAS for the Pathfinder



Georgia Tech

Near-memory Computation with the Rogues Gallery



Key idea: Sparse data and data movement costs will continue to dominate application concerns for the near future leading to opportunities for near-memory computing

The Pathfinder-S system was deployed in July 2021

- ~8X the processing elements with faster clocks, improved networking, and software stack
- Bolstered by related NSF projects and efforts like an (alpha) Kokkos backend for Lucata Cilk and GraphBLAS support

The Rogues Gallery also hosts HBM-enabled FPGAs, Hybrid Memory Cube and Optane platforms (both legacy support)

Related Work:

Lucata: Brian Page, Peter Kogge, "Deluge: Achieving Superior Efficiency, Throughput, and Scalability with Actor Based Streaming on Migrating Threads", HPEC 2021

*Brian Page, Peter Kogge, "*Scalability of Streaming on Migrating Threads", HPEC 2020

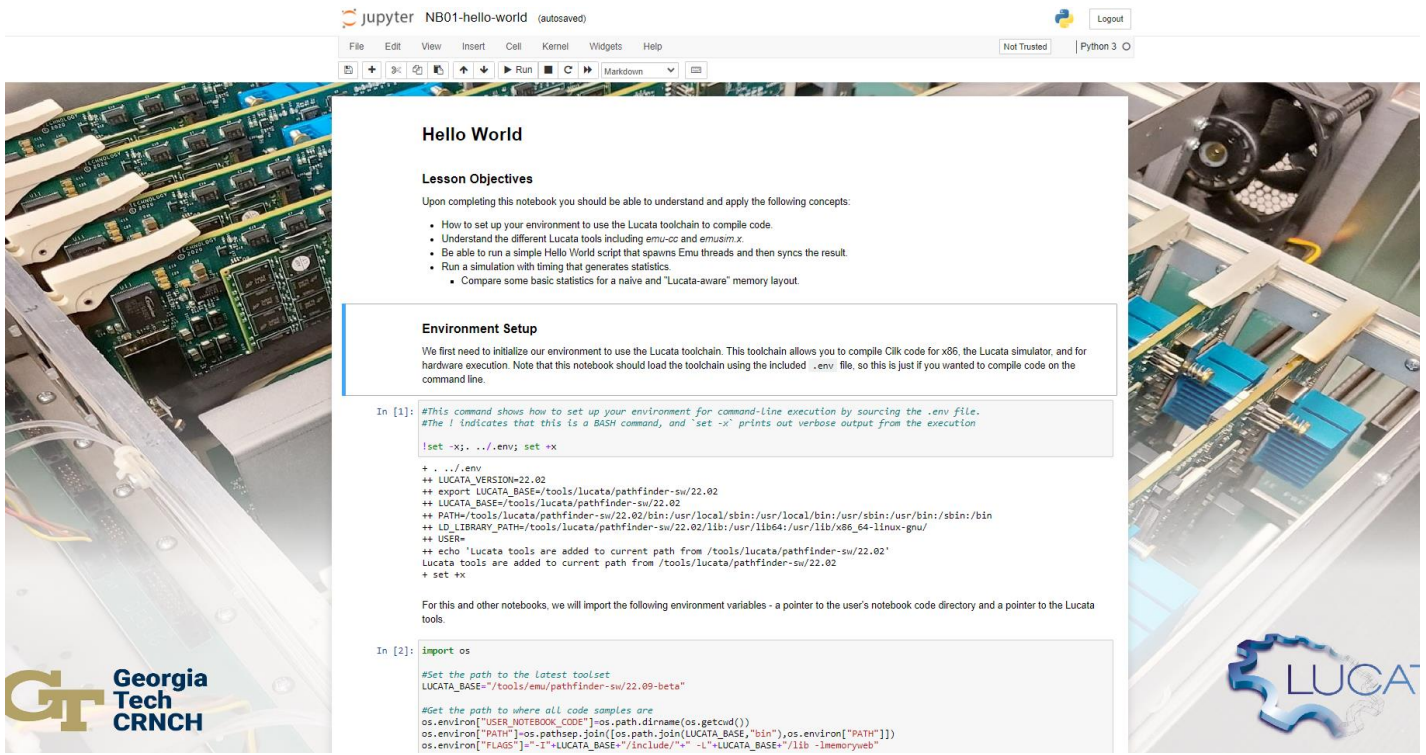
Optane: Tony Mason, Thaleia Dimitra Doudali, Margo Seltzer, Ada Gavrilovska, "Unexpected Performance of Intel® Optane™ DC Persistent Memory", CAL 2020

Georgia Tech

Tutorial Logistics

We will be using the Rogues Gallery Open OnDemand instance to run notebook code in an interactive fashion.

If you would like to participate in a hands-on fashion please enter your first/last name in the spreadsheet to “reserve” a username”. The passwords will be shared via Zoom and will work for the duration of the tutorial. **https://bit.ly/3LB5HRA**



Notice: All tutorial users are using Georgia Tech guest account resources. Please note that you must review and agree to comply with all rules mentioned in the Acceptable Use policy located at https://policylibrary.gatech.edu/information-technology/acceptable-use-policy

Georgia Tech

Tutorial Resources

All tutorial materials (slides, code, etc.) will be located at https://github.com/gt-crnch-rg/lucata-pathfinder-tutorial/

You can ask questions in the chat or join our CRNCH RG Slack if you plan to work on the system in the future (see the Getting Started link for mailing list/Slack information).

Learn more about CRNCH RG and request an account at:
https://crnch-rg.cc.gatech.edu/

Additional documentation at:
https://gt-crnch-rg.readthedocs.io/en/main/general/rg-getting-started.html

Georgia Tech