Los Alamos NATIONAL LABORATORY

Leveraging ModSim to Inform ATS-5 and Beyond

Kevin Sheridan - Presenter
Jered Dominguez-Trujillo
Galen Shipman

August 12, 2026

LA-UR-26-26878

Outline

- Los Alamos's applications
- Collaboration with ARM technology
- GEM5 indirect access memory accelerator design
- Trace-based benchmarking
- ATS-5 aka Mission outcome

Mission Centric Computing at Los Alamos

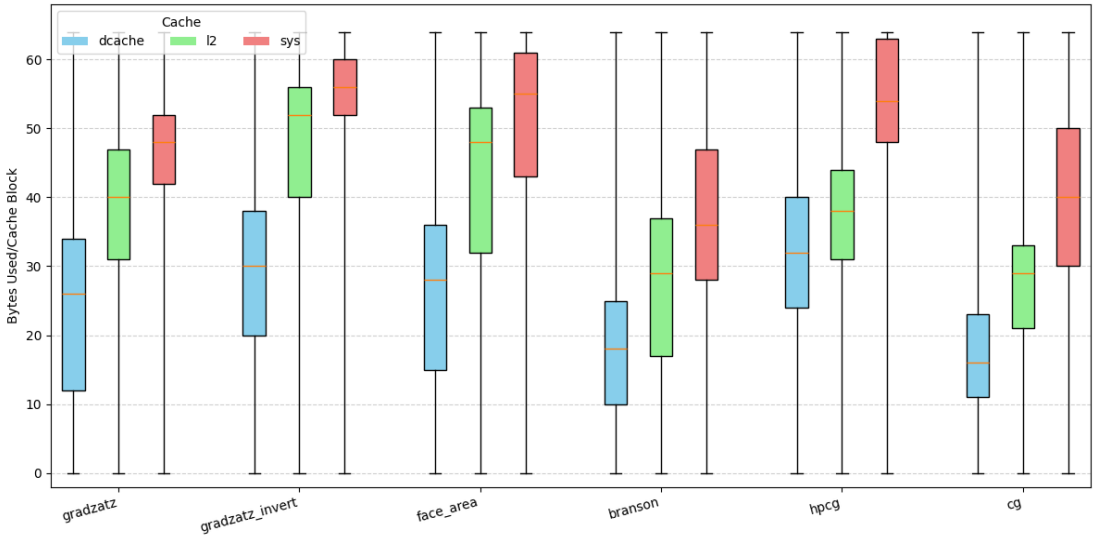
- Computing has played a pivotal role at Los Alamos from the 1940's to today – Advanced computing underpins our nation's nuclear deterrent.
- Large scale 3D hydro high energy multi-physic simulations.
- Memory bandwidth typically the bottleneck.
- Agentic AI and deep learning are enhancing LANL's HPC applications.
- **Future hardware/software success depends on the partnerships with colleagues from NNSA, DOE labs, US government, foreign governments, vendors and academia!**

Hardware-Software Co-Design for Multi-Physics Applications

Full Applications	Proxy Applications	Micro-Benchmarks	Tools, Traces and Replay	Microarchitecture Simulation
xRAGE	UME	STREAM	Intel PIN	DRAMSys
Flag	Pennant	Spatter	DynamoRio	DRAMSim
PARTISN	EAP Patterns	GUPS	NVBit	Gem5
Jayenne	Branson	HPCG	GS Patterns	ChampSim
	Snap	DGEMM	Loop Patterns	NVArchSim
	Parthenon-VIBE			GPGPUSim

Where do we Spend Time in our Codes?

- Significant (60%) time spent in memory and integer operations
  - Sparse data structures (AMR, Hydro, Monte Carlo)
  - Indirect memory access operations
- Results in poor bandwidth and cacheline utilization



Instruction	Percentage
Load	**18%**
Branching	16%
Integer Add	**14%**
Array Indexing	**13%**
Conditional	9%
Store	**7%**
Type cast	5%
Sign extension	4%
Stack frame allocation	3%
FP multiplication	3%
FP comparison	3%
INT multiplication	**3%**

**Shipman et.al., "Early Performance Results on 4th Gen Intel Xeon Scalable processors with DDR and Intel Xeon Processors codenamed Sapphire Rapids with HBM" arXiv:2211.05712*

IMPROVE BENCHMARKS

- Robust representation of real workloads

CUSTOM HARDWARE

- Design and build accelerators

VENDOR ENGAGEMENTS

- Express performance and software needs
- Compiler research

PUBLICATIONS

- Application behavior and hardware design performance analysis

ADAPT APPLICATIONS

- Use existing cutting edge hardware/software
- Port/rewrite code

ACADEMIA

- Partner with faculty to progress co-design research

DOE PROGRAMS

FASTFORWARD
Accelerating the future of high performance computing

DESIGNFORWARD
Co-designing future computing technologies

PATHFORWARD
Exploring innovative paths to exascale and beyond

ECP EXASCALE COMPUTING PROJECT
Unleashing the power of exascale computing

AMT ADVANCED MEMORY TECHNOLOGY
Building next-generation memory technologies

Beyond a Sea of Cores: Bandwidth per Core

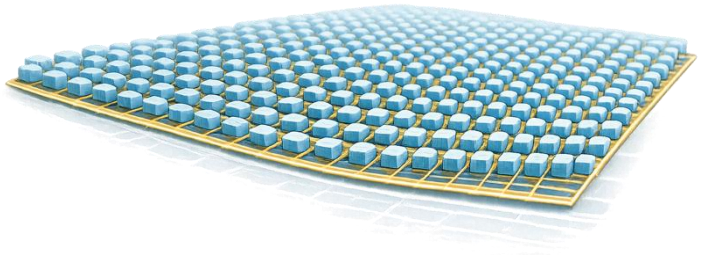
Decade ARM Collaboration

Workload sharing

- Identified bandwidth starved cores in LANL applications

Architecture response

- Larger coherent fabric real estate
- High concurrency / outstanding mem requests
- Improved NOC bi-section bandwidth
- Higher bandwidth per core
- Aggressive prefetching
- Monolithic die



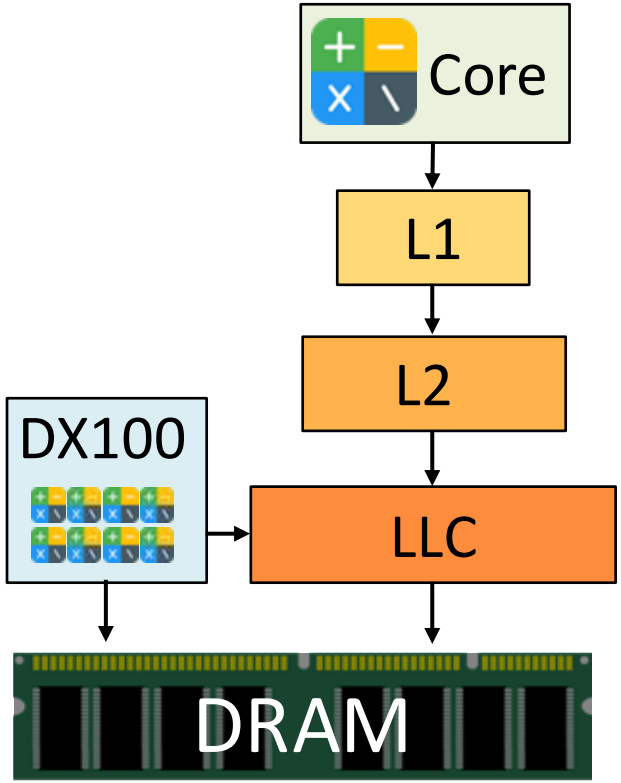
Bandwidth trajectory

	Platform	Cores	Mem BW	Single-core BW	Fabric / NoC
	ThunderX2	32	~160 GB/s	~12GB/s	~550GB/s
Crossroads →	SPR DDR	56	~307 GB/s	~20GB/s	~1.6TB/s per die
Crossroads →	SPR HBM	56	~700 GB/s	~20GB/s	~1.6TB/s per die
	Grace CPU	72	512 GB/s	**~30GB/s**	**~3.2 TB/s**
Mission →	**Vera CPU**	88	1.2 TB/s	**~58GB/s**	**~3.4 TB/s**

Grace → Vera: memory BW ~2.3X core count ~1.2X

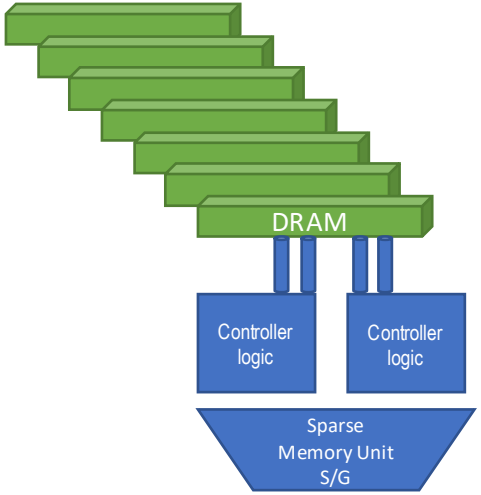
The design target shifts from more cores to more useful bandwidth per core.

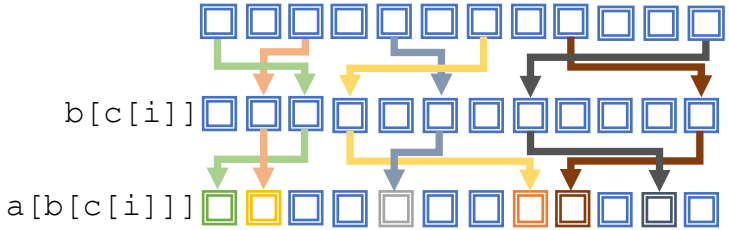
Vera performance numbers published by Phoronix



DX100: Data Access Accelerator for Indirection

A critical area: Optimizing for indirect loads



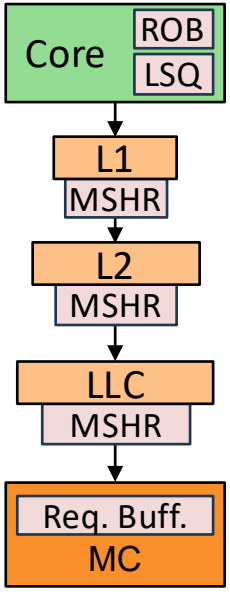


Potential: Integrate scatter/gather accelerations into controller logic allowing coalescing of small grain accesses to maximize bandwidth

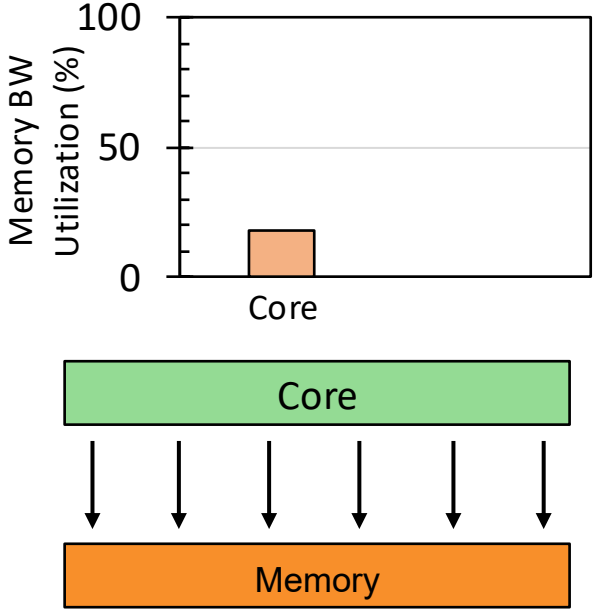
But many challenges:

1) Explicit programming scatter/gather?
2) Changes to MMU / Virtual memory / Caches?
3) Changes to ISA and/or µarch
4) Scratchpads or caches?

Microarchitecture Bottlenecks for Indirect Access




```
A[B[C[i]]]
for (i=0; i<N; i++)
    c = C[i]
    b = B[c]
    a = A[b]
    compute(a)
```

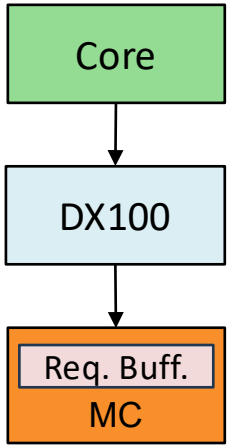


Micro-architecture
Multi-stage Buffering

Dependency Chain

Low Memory Request Rate

Microarchitecture Bottlenecks for Indirect Access

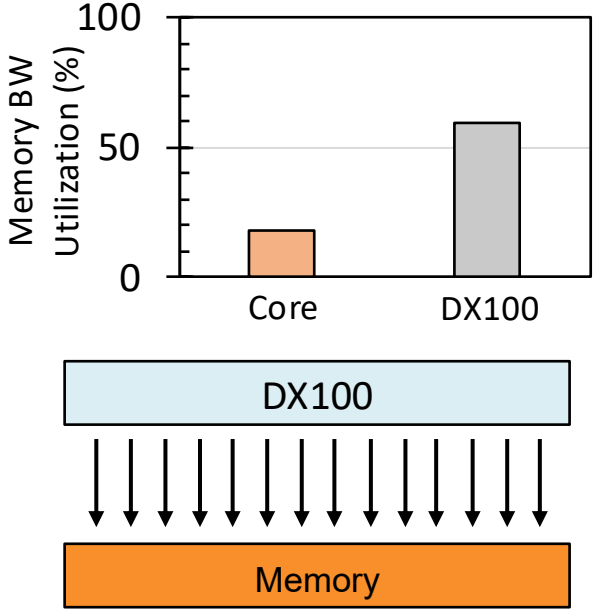



```
A[B[C[i]]]

// b_p[0:N] = B[C[0:N]]
b_p = DX100.LOAD(B, C)

// a_p[0:N] = A[b_p[0:N]]
a_p = DX100.LOAD(A, b_p)

for (i=0; i<N; i++)
   compute(a_p[i])
```



Direct Access to Memory Controllers

Hoist Data Access
Data Access Acceleration

High DRAM Request Rate

DX100 + Skylake-like 4 OoO Cores

Indirect data access – low memory bandwidth utilization

- Chain of instruction dependency + multi-level buffering = limited memory-level parallelism
- Non-contiguous memory access – low row-buffer hit-rate

Application study

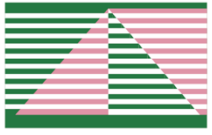
- LD/ST/RMW access type, conditional accesses, single and range loops
- 12 benchmarks spanning scientific computing, database, and graph applications

DX100 – general-purpose data access accelerator for indirection

- Bulk data access acceleration + direct access to MC = significant memory-level parallelism
- Data access reordering, interleaving, and coalescing – high row-buffer hit-rate

2.5×
Performance

3.7×
Bandwidth Utilization



Published in ISCA 2025
June 21-25, 2025 – Tokyo, Japan

A. Khadem, K. Kamalakkannan, Z. Zhu, A. Poptani, Y. Gu, J. Benjamin Dominguez-Trujillo, N. Talati, D. Fujiki, S. Mahlke, G. Shipman, and R. Das, "*DX100: Programmable Data Access Accelerator for Indirection*," In Proceedings of the 52th Annual International Symposium on Computer Architecture (ISCA 2025).

Trace-Based Benchmarking and Sharing

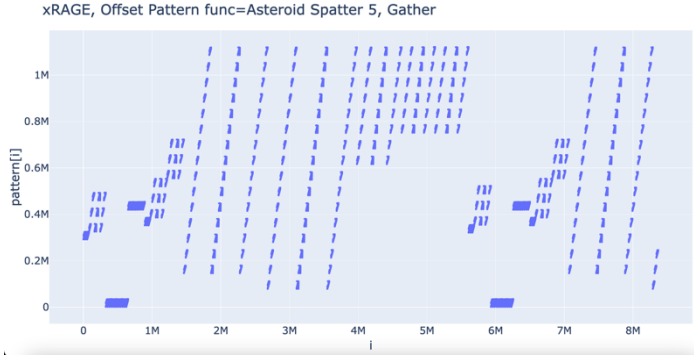
Trace-Based Benchmark and Analysis Motivations

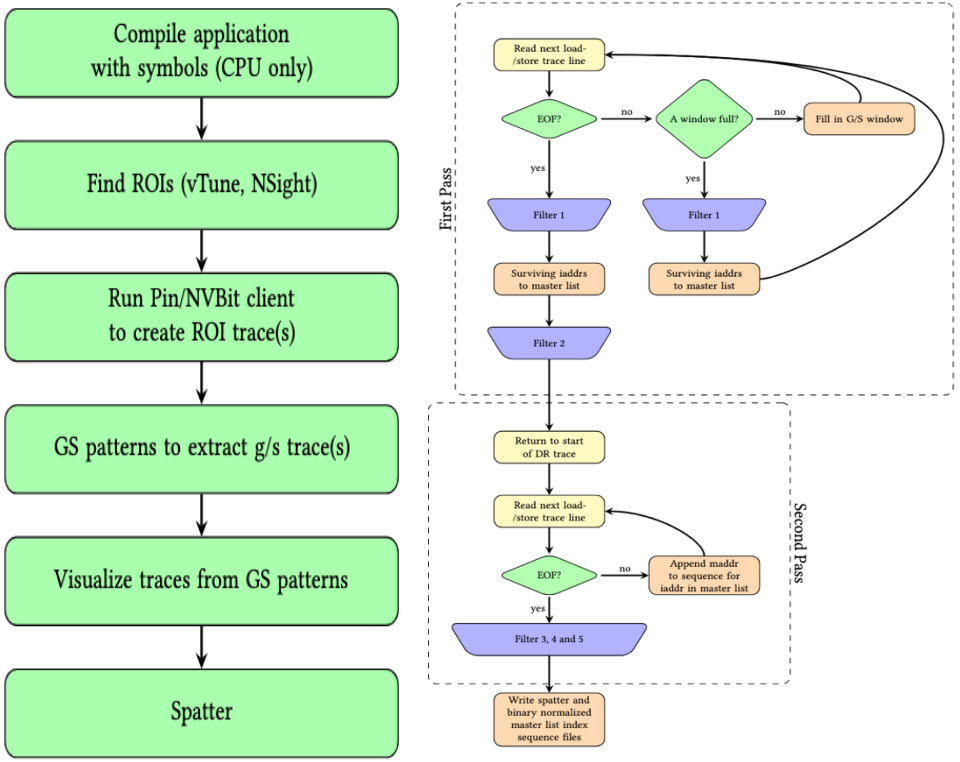
- Cannot share code! Source code for flagship applications are Export Controlled / Classified
- Improve benchmarks by sharing access patterns closer to ground truth.
- Benchmarks stay up-to-date.
- Once the trace is created, no longer need to deal with the application/inputs.
- Provide capability to do deep analysis on non-sampled, complex, dynamic multi-physics codes on realistic problems
  - Identify specific inputs, timesteps, ranks, and regions of interest for analysis

Tool: GS Patterns - Collect indirect accesses from instr addrs

In Collaboration with Georgia Tech: Christopher Scott, Agustin Vaca Valerde, Richard Vuduc, Jeffrey Young
And Sandia National Labs: Patrick Lavin

- DynamoRio/PIN Tool/NVBit to collect memory trace
  - Collected in DynamoRio format
- Perform analysis on trace
  - Auto Identify instr addrs with indirect access
  - Filter top gather/scatter patterns
- Run traces through Spatter to evaluate effective bandwidth





Tool: Loop Patterns

Auto Find and Extract Loop Traces

Find top memory offending loops automatically.

Optimized for large multirank production applications.

Metrics split into FP and INT for L1 misses, cacheline util, memory reuse, ...

Basic simulator to collect each loop's metrics.

INPUT

DynamoRio traces.

WORK

Finds "worst" memory performing loops by looking at windows of repeating instruction addresses.

Loops are ranked with chosen metric.
There are 13 metrics available to choose from.

OUTPUT

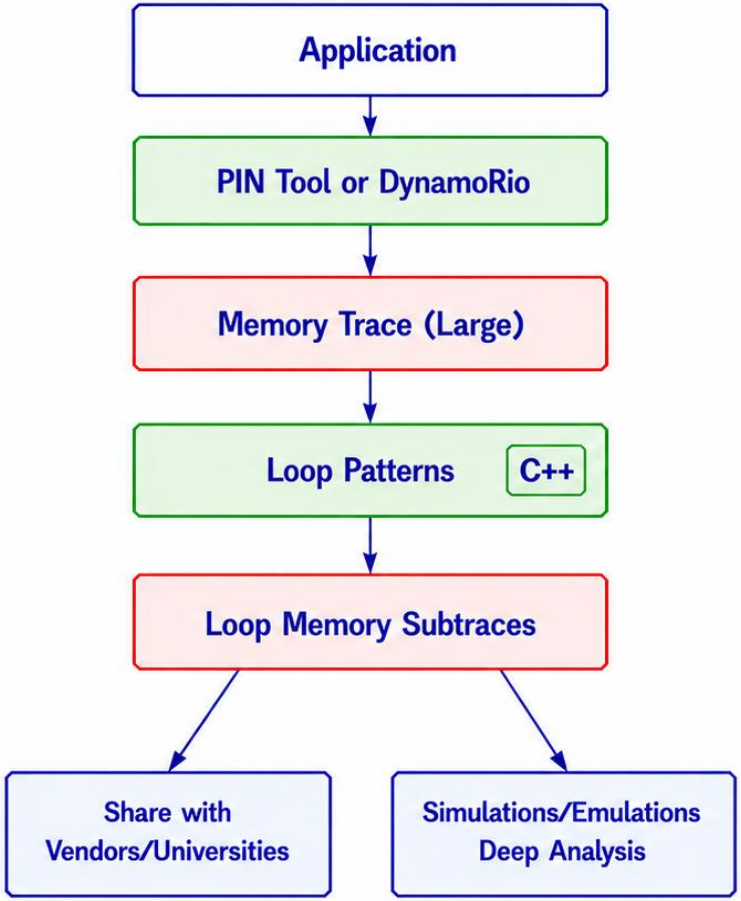
Source code file and line start and end for top offending loops.

Writes obscured DynamoRio formatted loop memory address sequences.

Warmup for each loop included with tag separation.

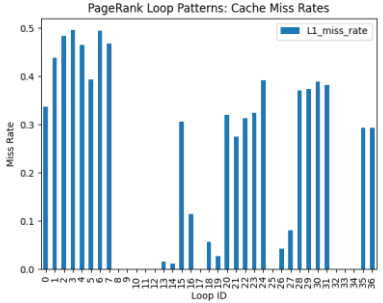
Profile of all metrics (csv).

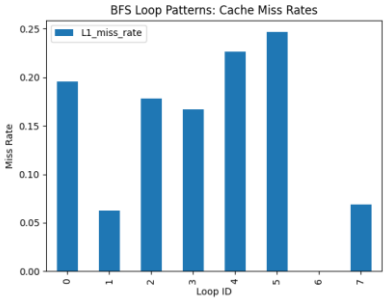
Spatter-like replay benchmark being worked on by **GTech, SNL** and **LANL.**
Releasing loop patterns source with paper soon!

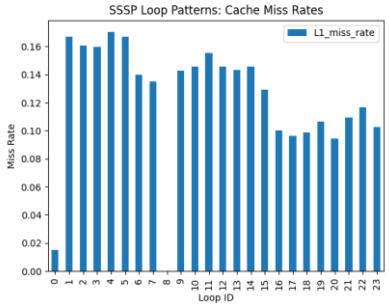


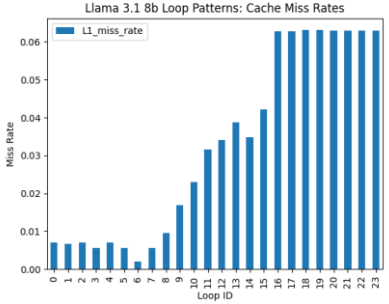
Tool: Loop Patterns – Cache Miss Rates - SPR

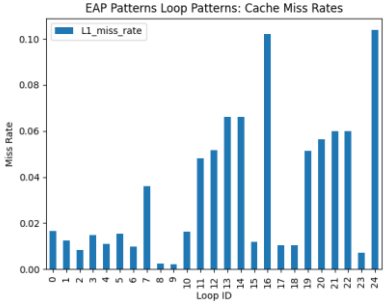
- Ran on 6 Applications
  - GAPBS
    - PR, BFS, SSSP
  - Llama 3.1 8b
  - EAP Patterns
    - Hydro AMR proxy
  - xRAGE
    - LANL Hydro-rad code

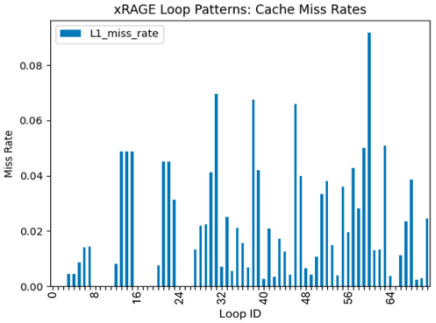






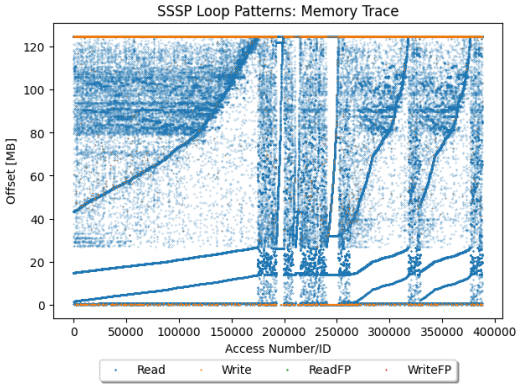


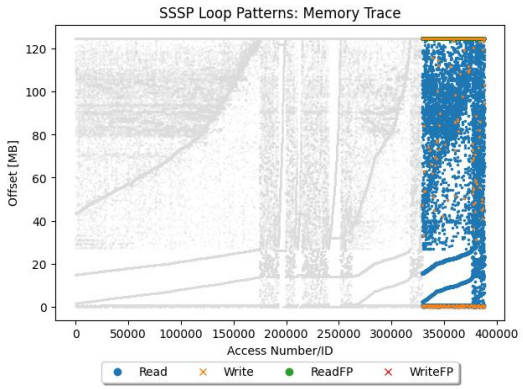


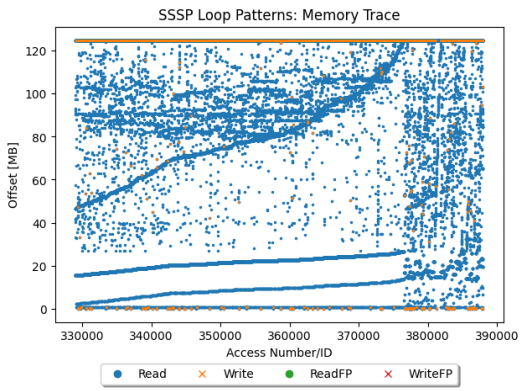


Tool: Loop Patterns – Memory Access Patterns - SPR

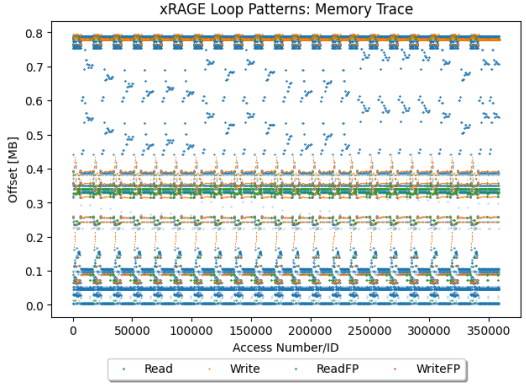
SSSP

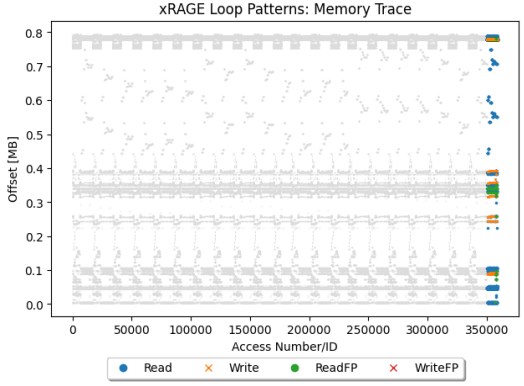


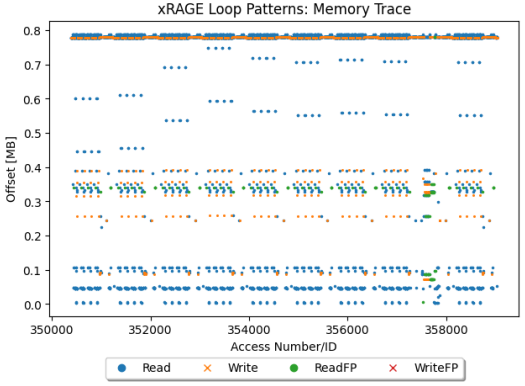




xRAGE







Loop + Warmup

Greyed out Warmup

Only Loop

Co-Design for Sparse Memory Access

Tooling to collect memory traces and perform analysis

DynamoRio/Intel PIN Tools
Collect memory traces

Los Alamos NATIONAL LABORATORY · Georgia Tech · Sandia National Laboratories

GS Patterns
Identify instructions which perform sparse gather/scatters

Loop Patterns
Extract memory aggressive loops

Benchmarks to replay memory traces

Los Alamos NATIONAL LABORATORY · Georgia Tech · Sandia National Laboratories

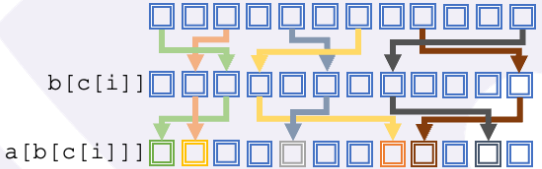
Spatter
Gather/Scatter benchmark

LP Replay (work in progress)
Loop benchmark

Inform

Developers, Vendors and Hardware Design

- Require identification of sparse kernels and patterns from ***full applications*** to inform accelerator design
- Require proxies and benchmarks that can ***replay sparsity patterns*** *and* ***loops*** for evaluation



Los Alamos NATIONAL LABORATORY

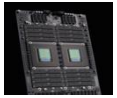
Georgia Tech · Sandia National Laboratories · University of Michigan · UC Davis · gem5 · SK hynix · NVIDIA · intel

ATS-5 - Mission - Delivery 2027

Sandstone
Sparse optimized compute

Large-scale high-complexity mod/sim

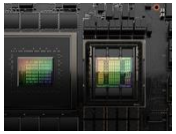
- Vera/Vera CPU superchip
- Scale-out for mod/sim
- Support for up to ~1 PiB memory footprint in a single job
- Large memory / core
- Network injection rate optimized scale-out network



Ranger
Dense optimized compute

Time-to-solution

- NVL-4 Vera/Rubin
- Moderate scale mod/sim
- AI Training & moderate complexity Inference
- Major increase in SSI over Crossroads
- Large HBM4 memory footprint



Starlight
Inference optimized compute

High complexity AI

- NVL-72 Vera/Rubin
- Large shared memory
- High-complexity chain-of-thought inference
- Agentic AI workflows
- Science/Engineering Copilots
- Strong scaling

High Performance Storage

- Connected to all partitions
- All flash configuration
- Network optimized for bandwidth
- AI optimized file system
- HPC optimized file systems

Infrastructure

- Front-end/Workflow nodes
- Software optimized for Mod/Sim on V/V
- Software optimized for AI + Mod/Sim on V/R
- Software optimized for AI inference on NVL-72
- Same arch as compute

ArchEvolve

An Agentic AI System Utilizing Evolutionary Principles for Advanced Co-Design

LANL: Galen Shipman, Kyle Roarty, Sumathi Lakshmiranganatha
UMich: Reetuparna Das, Scott Mahlke, Seah Kim
Intel: Eric Fetzer

- Vision: Use AI to dramatically accelerate application-specific accelerator platform design.
- Approach: Multi-agent framework that jointly co-designs algorithms, software, application accelerators, memory systems, and interconnects (on-package).
  - Analyze scientific workloads
  - Generate candidate accelerator platforms
  - Optimize software mappings and runtimes
  - Evaluate performance, power, area, and cost
  - Iteratively improve designs through evolutionary search
- Impact: Reduce design cycles from years to months while enabling rapid adoption of new algorithms, data structures, and hardware technologies.

Special thanks

Jeff Young (Georgia Tech)
Patrick Lavin (SNL)
and all others who contributed

Questions?

Backup

2014	LANL begins collaboration with Broadcom on ARM technologies
2016	Cavium announces the ARM based ThunderX2
2018 May	ThunderX2 launched by Cavium
2018 July	Cavium acquired by Marvel
2018 Oct	LANL announces ThunderX2 deployment with Cray
2018 Nov	LANL funds development of ThunderX4, focus on bandwidth
2020 Aug	Marvell cancels ThunderX3 (and follow-on)
2020 Aug	ThunderX team departs Marvell to Nvidia and SiPearl
2020 Oct	LANL partners with HPE (Cray) & Nvidia on ARM based Grace processor
2021	Deep codesign of Grace with particular focus on high bandwidth LPDDR
2023	Deep codesign of GraceNext (Vera) focusing on: bandwidth per core, increased number LPDDR memory channels, NOC improvements
2024	**Venado** deployed, first Grace/Grace supercomputer
2025	**Mission, Vision and Veritas** announced, purpose built compute for complex workflows -> Vera/Vera, Vera/Rubin NVL-4 and NVL-72
2026	Coming soon: Vera/Rubin and Vera/Vera at LANL

More than a decade of investments in Chip designs