xBGAS: Extended Base Global Address Space for High Performance Computing

John Leidel[1], Xi Wang[2,6], Brody Williams[2], Nathan Stoddard[2], Yong Chen[2], David Donofrio[1], Alan Ehret[3], Miguel Mark[3], Michel Kinsy[3], Farzad Fatollahi-Fard[4], Kurt Keville[5]

[1]Tactical Computing Labs, [2]Texas Tech University, [3]Texas A&M University, [4]Lawrence Berkeley National Lab, [5]Massachusetts Institute of Technology, [6]RISC-V International Open Source Laboratory (RIOS)



Overview

- Introduction
- Remote Atomic Extension
- Request Aggregation
- xBGAS Filesystem
- Ongoing Work

What is xBGAS?

- Extended Base Global Address Space (xBGAS)
- Goals:
  - Provide extended addressing capabilities without ruining the base ABI
    - EG, RV64 apps will still execute without an issue
  - Extended addressing must be flexible enough to support multiple target application spaces/system architectures
    - Traditional data centers, clouds, HPC, etc..
  - Extended addressing must not specifically rely upon any one virtual memory mechanism
    - EG, provide for object-based memory resolution
- What is xBGAS NOT?
  - …a direct replacement for RV128

Why xBGAS?

- **Performance**: high-performance remote memory accesses
  - ISA-level RMA support - No redundant software overheads induced by heavy weight communication libraries like MPI, OpenSHMEM, etc.
- **Scalability**: targeted at datacenter-scale HPC systems
- **Generalizability**: compatible with standard OS and ABI
- **Applicability**: applicable to diverse application domains
  - HPC-PGAS, MMAP-I/O, File systems, Security, HPA-flat, etc.

ISA Extension

xBGAS Instructions are split into three blocks:

- Address management
  - Store extended addresses
  - E.g. eaddie, etc.
- Base integer load/store
  - Remote load/store with immediate
  - E.g. eld, esd, etc.
- Raw integer load/store
  - Remote load/store with registers
  - E.g. erld, ersd, etc.

	Mnemonic	Base	Funct3	Dest	Opcode
I-Type	eaddie rd, ext1, imm	rs1	111	extd	1111011
	eld rd, imm(rs1)	rs1+ext1	011	rd	1110111

.....

	Mnemonic	Src	Base	Funct3	Opcode
S-Type	esd rs2, imm(rs1)	rs2	rs1+ext1	011	1111011

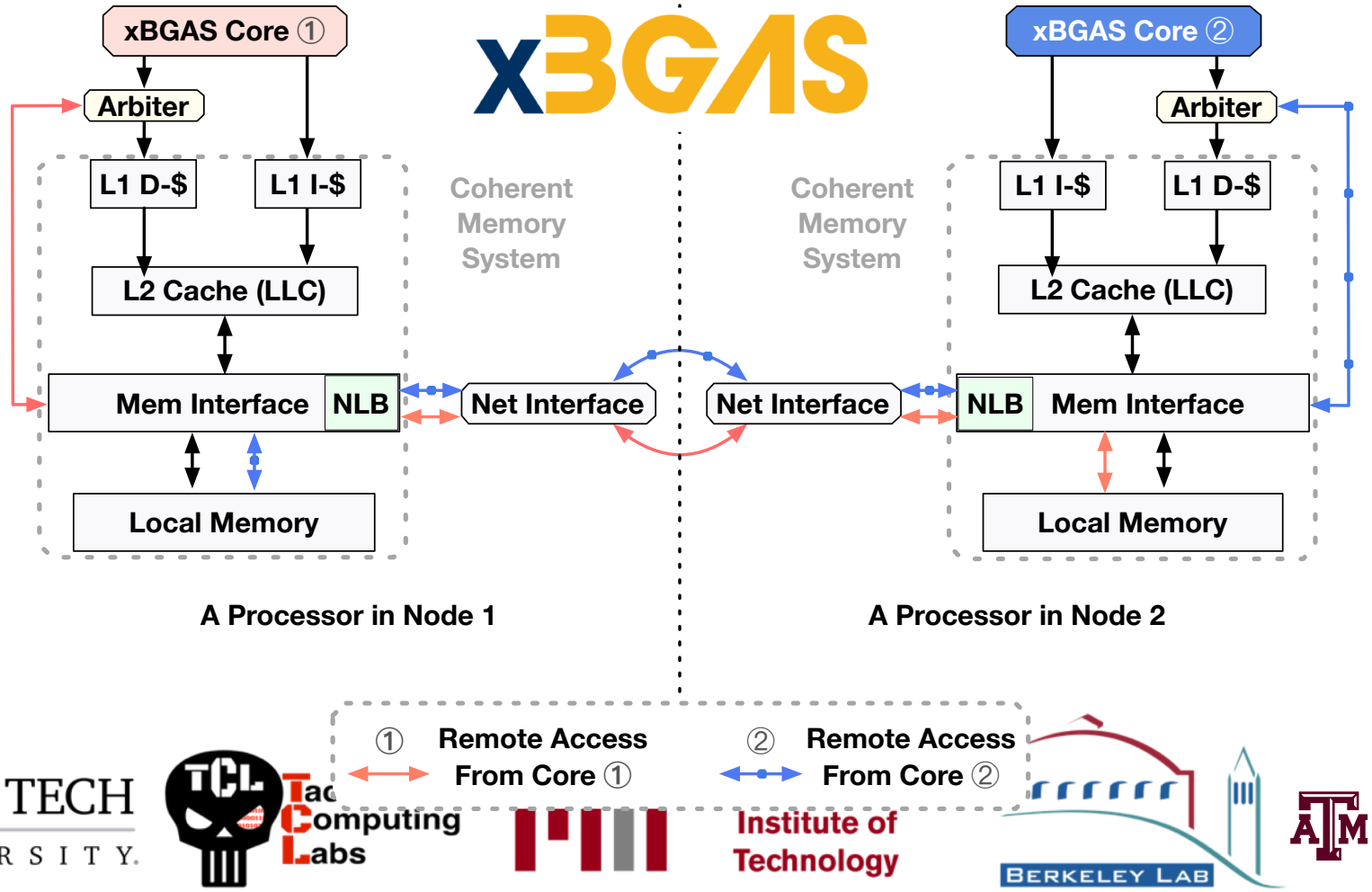
.....

	Mnemonic	Funct7	RS2	RS1	Funct3	RD	Opcode
R-Type	erld rd, rs1, ext2	1010101	ext2	rs1	011	rd	0110011
	ersd rs1, rs2, ext3	0100010	rs2	rs1	011	ext3	0110011

.....

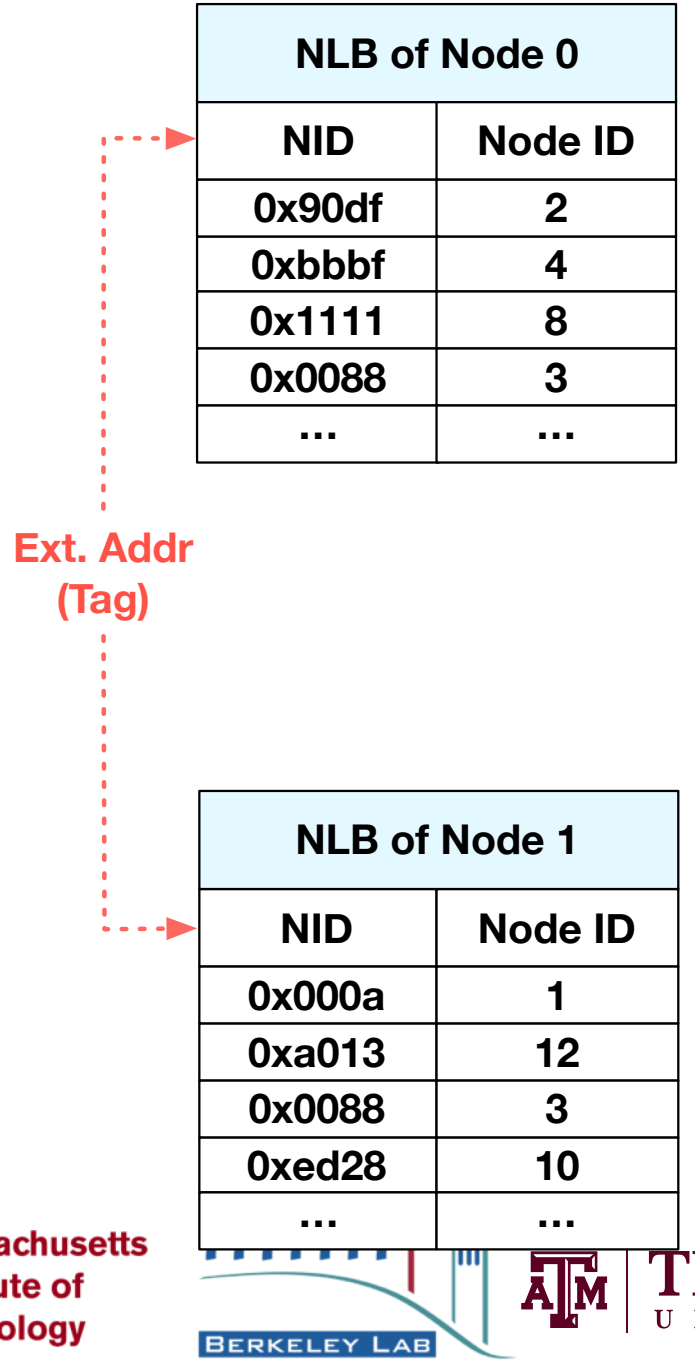
xBGAS Architecture

- Microarchitecture extension for remote data accesses



NLB

- NLB: Namespace Lookaside Buffer.
- NLB maps the extended address space (bit[127:64]) to the remote nodes.
  - Namespace ID (NID) is unique
  - Each NID corresponds to a remote node ID



Remote Atomic Extension

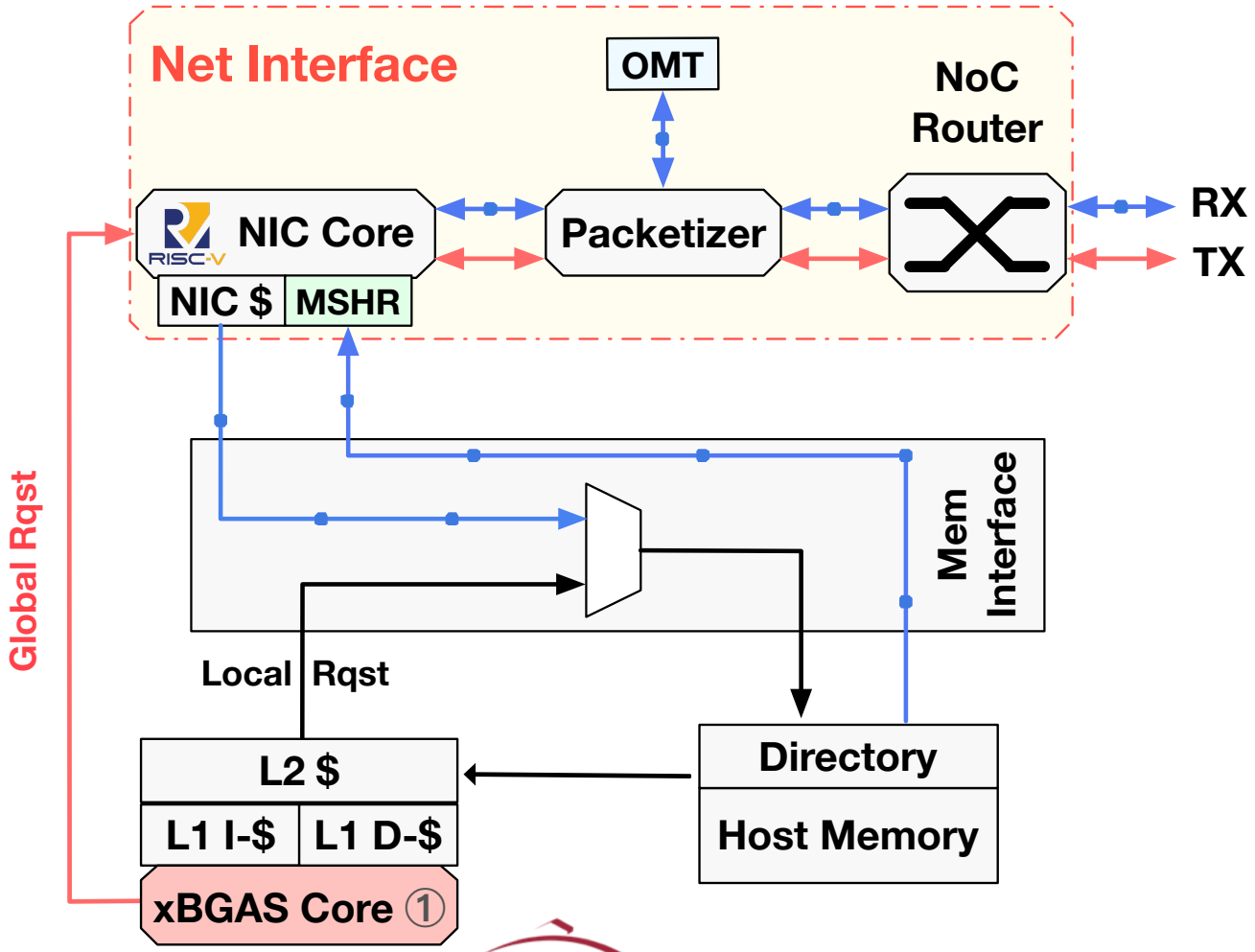
- Beyond basic remote load/store operations, global atomic support is also desired
  - Graph analysis, synchronizations, etc.

- Rather than relying on heavy-weight software, we also introduce inter-node atomic operations
  - Fetch-and-add, compare-and-swap, etc.

- One-sided operations with global atomicity

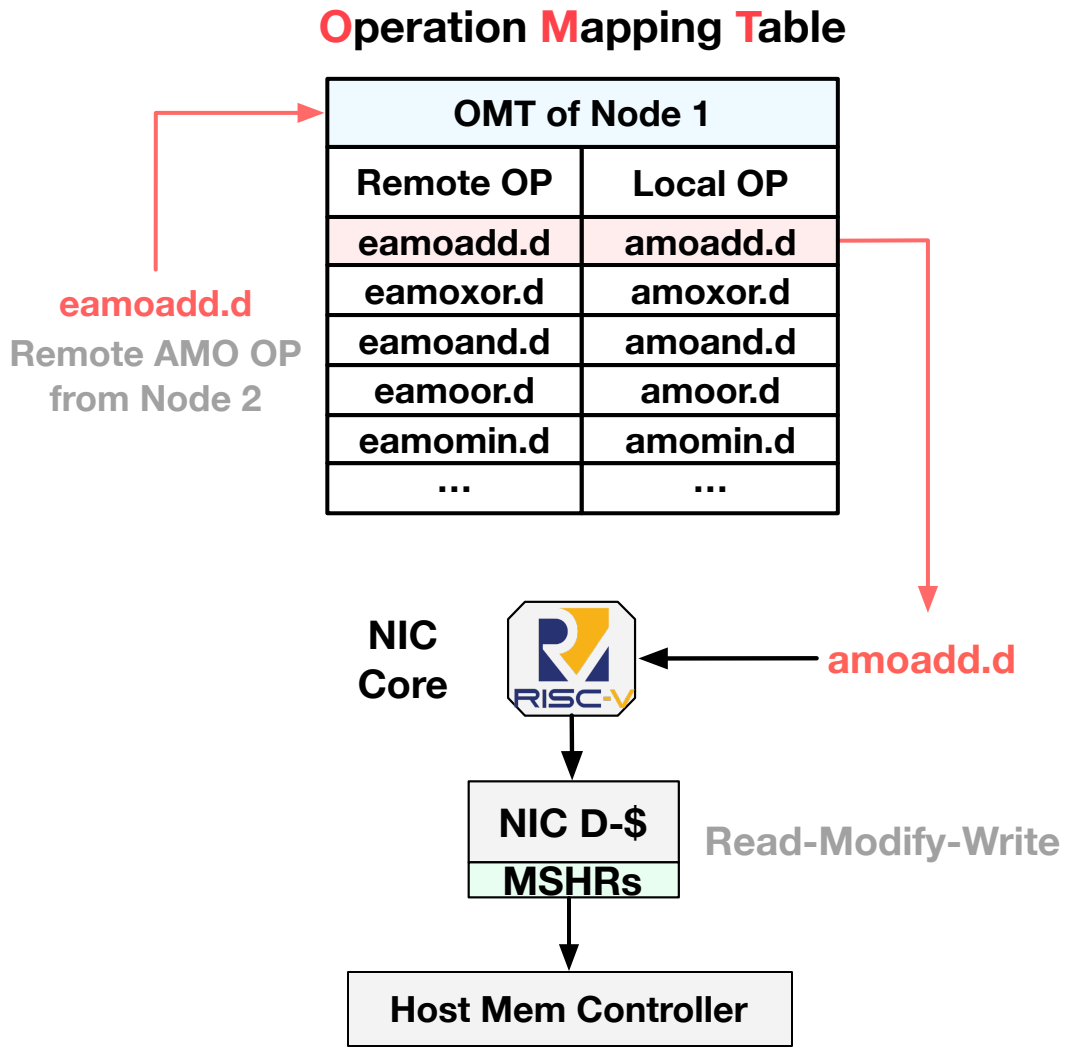
Remote Atomic Extension

- **Acceleration**
  - Offloading remote AMO requests to NIC cores
- **Operation Mapping Table**
  - OMT converts remote AMOs to local counterparts
- Directory-based coherency



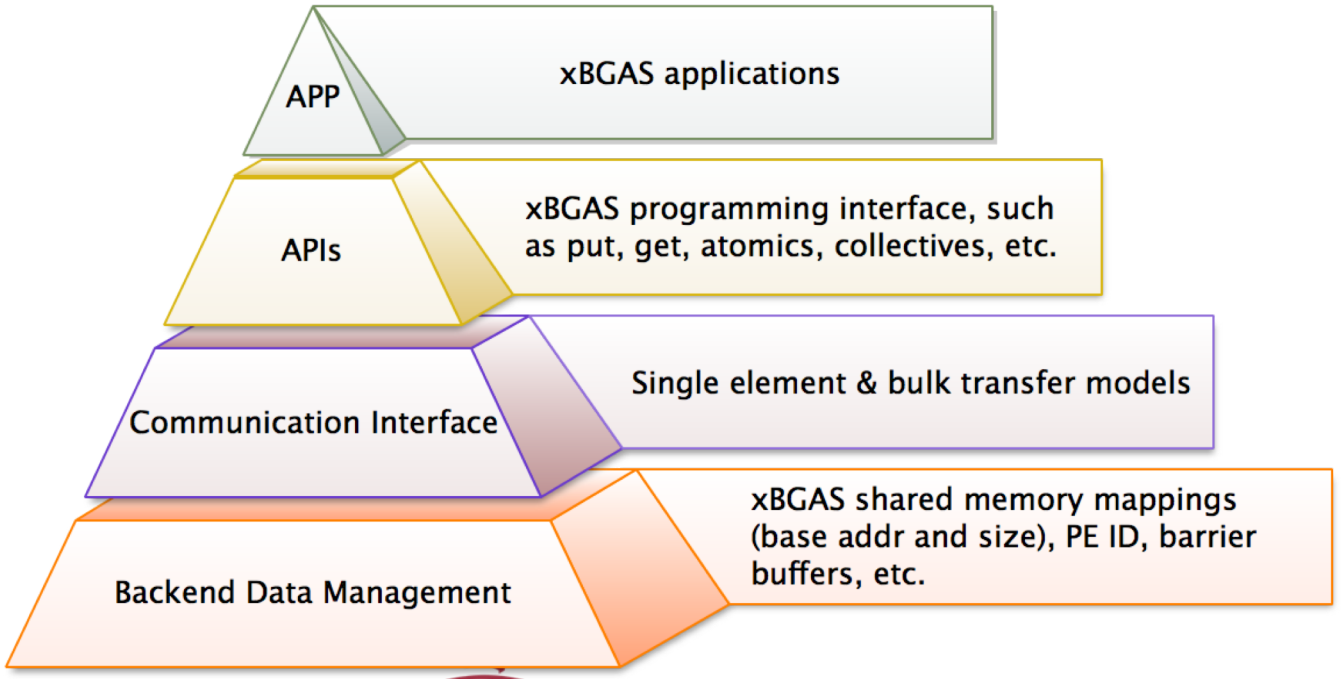
OMT Design

- **OMT**
  - A lookup table
  - Maps between remote and local AMO operations
- **RISC-V Core**
  - Each extended AMO operation corresponds to a native RISC-V atomic instruction



Aggregation

- We provide a bulk transfer interface in the xBGAS runtime layer to provide the support of aggregated data movement
  - Automatic optimizations based on the payload size
  - Register-width (1B~8B): single element transfer
  - Otherwise, bulk transfer will be invoked



Aggregation

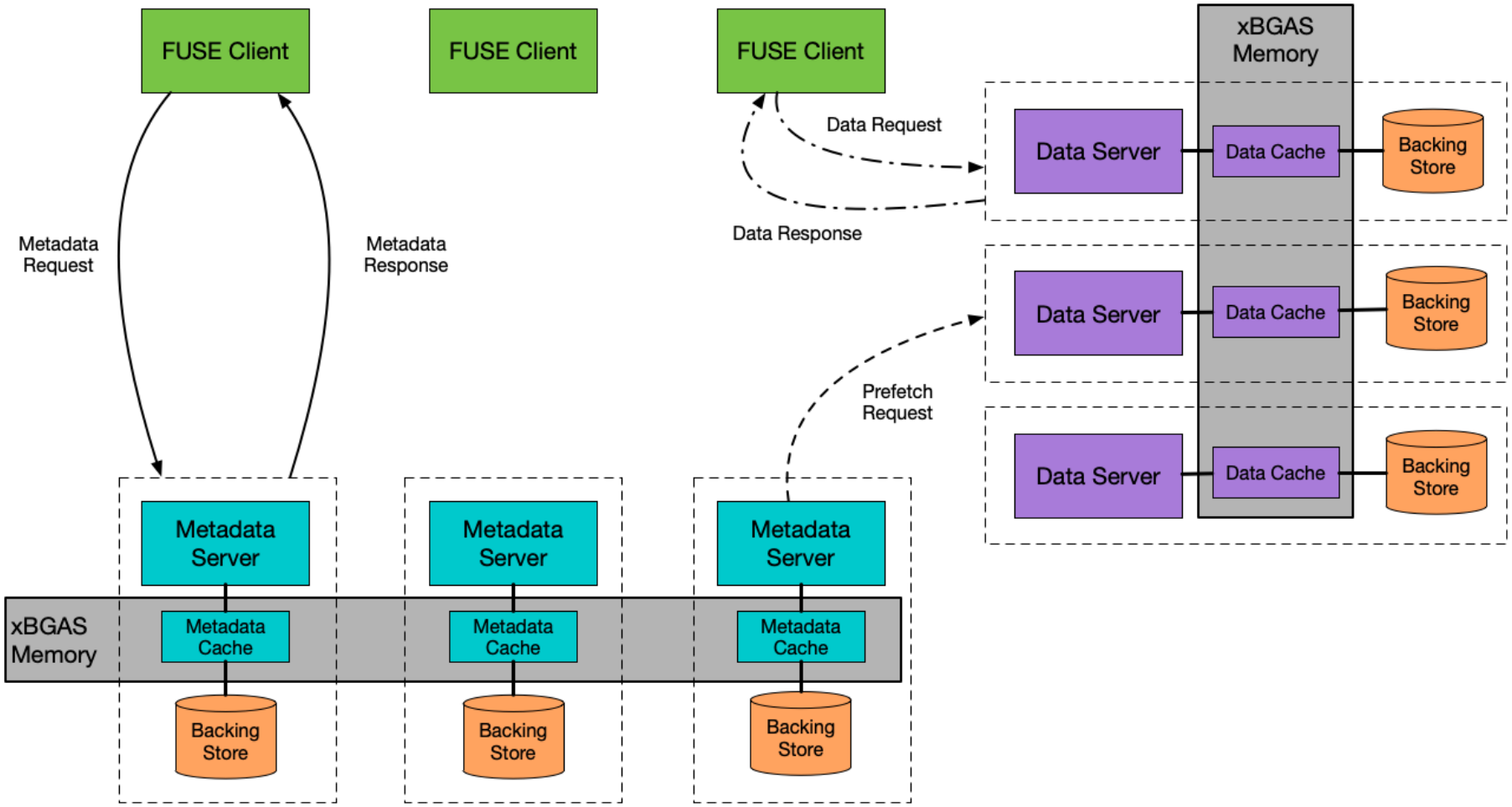
- We provide a bulk transfer interface in the xBGAS runtime layer to provide the support of aggregated data movement

- We implement the bulk transfers based on a DMA engine and control status registers (CSRs)
  - *esrc*: lower 64 bits of the base source address
  - *esrce:* extended source address
  - *edst*: base destination address
  - *edste:* extended destination address, respectively.
  - *ecsr*: control information: transfer status (idle/busy), length, and stride

xBGAS-FS - Motivation

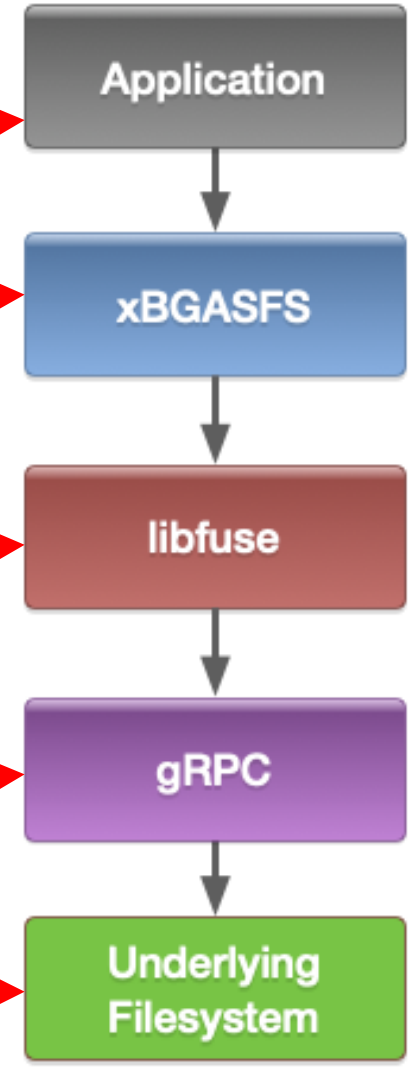
- Modern HPC systems require the use of parallel and/or shared file systems for scratch, user data, etc
- These high-performance parallel file systems split functional operations into three areas:
  - File system presentation (POSIX I/O Interfaces)
  - File system I/O operations (read, write, sync)
  - File system metadata operations (attributes, ls, create)
- File system scalability is often gated by metadata performance
  - Especially for small file I/O
- xBGAS-FS seeks to solve scalability issues with file system metadata by utilizing xBGAS extensions to share metadata operations/memory across metadata servers

xBGAS-FS



xBGAS-FS - Software Stack

- Performs metadata and/or file I/O
- xBGAS accelerated PFS
- Run on client devices
  - Intercepts read, write, open, etc. calls
  - Redirects to xBGASFS calls
- Used by xBGAS-FS calls to facilitate RPC between clients and storage servers
- MST/OST filesystem



XBGAS-FS Status

- XBGAS-FS “Passthrough” Prototype
  - Successful integration of Libfuse + gRPC
  - Updated to support newest Libfuse release (3.9.2)
  - 37/42 gRPC-enabled Libfuse high-level functions implemented with Linux system calls
  - Based on synchronous Libfuse & gRPC models

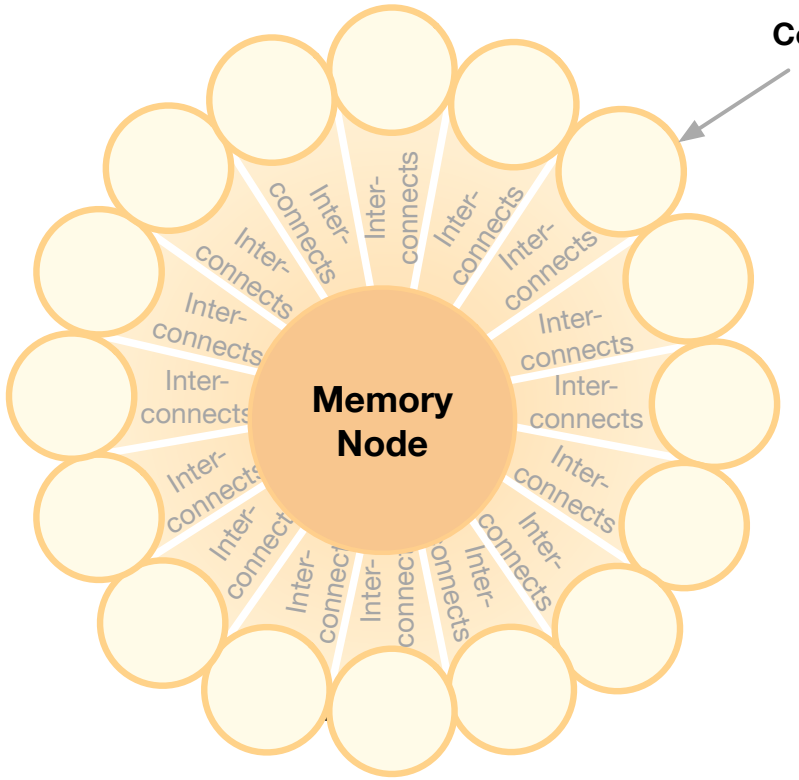
- XBGAS-FS/xBGAS Toolchain Integration
  - riscv-unknow-elf-* compilers & Spike simulator incompatible with xBGASFS
    - Minimal system call support, do not support Pthreads
    - Currently exploring other options including SiFive Freedom U SDK

Ongoing work

- Ever-growing datasets of data-intensive workloads that cannot be effectively sharded lead to the necessity of a memory node that provides
  - Disaggregated fabric-attached memory (FAM) pool
  - Can be allocated on the fly
  - Compatible with current distributed shared memory programming paradigm

Flora

- We thus introduce Flora, a memory-centric system with memory nodes:
  - Disaggregated memory detached from compute nodes
  - Heterogeneous memory system support (DDRx/NVM)
  - Fine-grained control over disaggregated memory (allocation/deallocation/operations/volatile/persistent)
  - Maintain the support of SPMD model with symmetric shared memory
    - Extension from xBGAS model to bridge FAM



xBGAS Specification & Codebases

- xBGAS Spec: https://github.com/tactcomplabs/xbgas-archspec
- xBGAS Toolchain: https://github.com/tactcomplabs/xbgas-tools
- xBGAS ISA Tests: https://github.com/tactcomplabs/xbgas-asm-test
- xBGAS Runtime: https://github.com/tactcomplabs/xbgas-runtime
- xBGAS Benchmarks: https://github.com/tactcomplabs/xbgas-bench

We welcome comments/collaborators!

Publications

- Xi Wang, John D. Leidel, Brody Williams, Alan Ehret, Miguel Mark, Michel Kinsy, and Yong Chen, *xBGAS: A Global Address Space Extension on RISC-V for High Performance Computing*, In the Proc. of IEEE Conference on International Parallel & Distributed Processing Symposium (IPDPS) 2021.

- Xi Wang, Brody Williams, John Leidel, Alan Ehret, Michel Kinsy and Yong Chen. Remote Atomic Extension (RAE) for Scalable High Performance Computing. In the Proc. of the 57th Design Automation Conference (DAC), 2020

- Brody Williams, Xi Wang, John Leidel and Yong Chen, Collective Communication for the RISC-V xBGAS ISA Extension, In the Proc. of the Parallel Programming Models and Systems Software for High-End Computing (P2S2) workshop, 2019.

- John D. Leidel, Xi Wang, Yong Chen, David Donofrio, Farzad Fatollahi-Fard and Kurt Keville. xBGAS: Toward a RISC-V ISA Extension for Global, Scalable, Shared Memory, In the Proc. of the Memory Centric High Performance Computing (MCHPC) workshop, 2018

TEXAS TECH UNIVERSITY

Tactical Computing Labs

Massachusetts Institute of Technology

BERKELEY LAB

TEXAS A&M UNIVERSITY