Aengus McGuinness github.com/AengusMcGuinness (805) 570-1436
Santa Fe, NM amcguinness@college.harvard.edu

Education

- **Harvard College** A.B. Computer Science
  3.8 GPA / ACT 35, John Harvard Scholar (2023-2024) *Expected May 2028*
  - Coursework: Operating Systems (CS1610), Distributed Systems (CS2620), Computer Architecture (CS1411), Modern Data Storage Systems (CS2640), Machine Learning, Algorithms, Probability, Linear Algebra

Work Experience

- **Axilon** New York, NY
  Software Engineer (incoming) *June 2026 - present*
  - Joining an early-stage industrial AI startup (BoxGroup, RTX Ventures) to work on path-critical infrastructure deploying LLMs into industrial operations and engineering workflows.
- **Harvard Generative AI Research Program** Cambridge, MA
  Research Fellow *June 2025 - Aug 2025*
  - Designed distributed deep learning pipelines executing across GPU clusters using SLURM orchestration and dynamic resource allocation.
  - Integrated Bayesian optimization into existing OpenFold workflows, achieving 2× validation improvement while maintaining scalable execution across compute nodes.
- **Los Alamos National Laboratory, B-GEN** Santa Fe, NM
  Student Researcher *June 2022 - Aug 2023, Summer 2024*
  - Optimized distributed HPC pipelines processing terabyte-scale biological datasets, reducing workflow runtime by 80% through SLURM parallelization and memory tuning.
  - Engineered GPU-accelerated data processing modules for large-scale biological computation, eliminating throughput bottlenecks in production research workflows.

Selected Systems & Performance Projects

- **RCache-RDMA: One-Sided RDMA Distributed Cache (C++)** Harvard CS2640
  Networking and Distributed Systems Project *2026*
  - Built a key-value cache with three communication paths in C++ using `libibverbs`: TCP/RPC baseline, two-sided RDMA messaging, and one-sided RDMA reads over a registered hash-table memory region.
  - Achieved 974k ops/s with stable 12 μs p99 latency on one-sided RDMA — a 13× throughput improvement and 20× tail-latency reduction over the TCP baseline on CloudLab Mellanox hardware.
  - Isolated the cost of cache-policy metadata maintenance by adding RDMA `FETCH_AND_ADD` atomics to the read path, quantifying a consistent 3–4 μs p99 latency penalty for recency tracking.
- **Adaptive Stream Buffer Prefetcher (C++, Intel Pin)** Harvard CS1411
  Computer Architecture Project *2026*
  - Implemented Jouppi's fixed-depth stream buffer and Palacharla-Kessler's adaptive prefetch policy as Intel Pin tools, modeling a two-level cache hierarchy with explicit latency costs.
  - Designed a histogram-driven depth-selection mechanism that learns stream-length distributions online; achieved 5.02× speedup on libquantum and 83% prefetch accuracy on dealII (vs. 78% for static next-line).
  - Reduced wasted memory bandwidth on irregular workloads by an order of magnitude versus next-line prefetching, with total hardware cost of 250–400 bytes.
- **High-Performance Async RPC Client (C++, gRPC)** Harvard CS2620
  Distributed Systems Project *2026*
  - Designed an asynchronous RPC client using completion queues and a multi-threaded polling architecture.
  - Increased throughput from ∼8k to 55k+ RPC/s via batching, non-blocking I/O, and flow-control tuning.
- **Chickadee OS Kernel (C++)** Harvard CS1610
  Operating Systems Project *2026*
  - Implemented blocking system calls (`sys_waitpid`, `sys_msleep`) by redesigning scheduler interactions and eliminating busy-wait loops.
  - Built a timer-driven sleep mechanism using a hashed timing structure, maintaining correctness under multi-core execution and preventing lost wakeups and race conditions.

Skills

- **Languages:** C++, Python, Bash
- **Systems:** Linux, Concurrency, Multithreading, Distributed Systems, RDMA (`libibverbs`), Intel Pin
- **Infrastructure:** SLURM, HPC, CloudLab, Git, gRPC
- **ML/Compute:** PyTorch, NumPy, Scikit-learn, CUDA-aware HPC