James Gleeson

Toronto, Ontario
Canada

647-298-9193
jagleeso@gmail.com
https://jagleeso.github.io

Education

- **University of Toronto** Toronto, ON
  Ph.D, Computer Science *Jan. 2015 - Present*
  - Co-advisors: Eyal de Lara, Gennady Pekhimenko
  - **Optimizing data collection in deep reinforcement learning workloads.**
    Based on my profiling insights from surveying RL workloads, I designed two optimizations for speeding up the time-consuming data collection phase of RL training: (1) *GPU vectorization:* moves simulation to the GPU to run more parallel simulator instances than possible on the CPU, and (2) *simulator kernel fusion:* fuses multiple simulation steps to run in a single GPU kernel launch to reduce global memory bandwidth requirements. I demonstrated that GPU vectorization leads to a 1024× speedup, and simulator kernel fusion leads to a 11× speedup; furthermore, these speedups are combinable. Unlike costly cluster scale-up solutions that can cost millions of dollars (e.g., AlphaGo, OpenAI's DotA 2), these optimizations are affordable since they are achievable on a single machine.
  - **Profiling CPU/GPU usage in deep reinforcement learning workloads.**
    I built an open-source profiling tool called RL-Scope for performing cross-stack analysis of RL training workloads. Unlike more commonly studied GPU-bound supervised learning workloads, RL training workloads must collect training data at runtime through simulation leading to a large CPU-bound component. RL-Scope provides a fine-grained breakdown of time CPU/GPU time into GPU-side kernel execution, and CPU-side execution of high-level language code (Python) and low-level framework code (CUDA API calls, TensorFlow framework). Using RL-Scope, we discovered at least 38% of training time in today's state-of-the-art RL workloads time is CPU-bound in simulation, with robotics workloads spending up to 74% in simulation; hence, future ML systems research must optimize the CPU-bound portion of the ML software stack to speedup RL. We surveyed scale-up workloads that parallelize multiple inference requests in an attempt to exploit GPU hardware parallelism, but discovered that coarse-grained GPU utilization metrics in off-the-shelf profiling tools led developers to make poor use of the available GPU hardware parallelism.
  - **GPU reallocation through live migration of GPU compute workloads.**
    The first publication of my PhD focussed on enabling efficient live migration of GPU compute workloads running inside a virtual machine instance. Device passthrough using IOMMUs in today's cloud data center deployments allows assignment of GPUs to VM instances with zero overhead, but this simultaneously prevents hypervisor device interposition needed for enabling VM migration (e.g., during machine maintenance). Our SYSTOR 2017 paper on Crane showed how to enable live migration with low-overhead ($< 5\%$) by transparently implementing state-tracking and interposition at the user-level GPU programming API (OpenCL), and enabling continued use of a GPU during live migration by forwarding GPU API calls to a proxy domain running on the source machine. In a followup HotCloud 2017 paper, we illustrated how GPU API level virtualization can be used to enable live migration between heterogeneous GPU models, enabling dynamic reallocation of GPUs in the cloud. GPU reallocation can avoid inefficient VM termination in spot instance offerings by moving running compute workloads to sufficient capacity GPUs that fall within the user's bidding price.

- **University of Toronto** Toronto, ON
  M.Sc., Computer Science *Sep. 2013 - Jan. 2015*
  - Advisor: Eyal de Lara
  - My masters research focused on operating system level solutions to mobile security problems that are efficient without sacrificing user experience, and transparent to avoid complecting existing application requirements.
  - **Memory Encryption on Mobile Devices.**
    I measured the performance and energy tradeoffs of protecting against cold boot attacks on Android through the encryption of sensitive processes. To support background applications running while encrypted, I swap encrypted DRAM pages into and out of a tightly managed decrypted secure memory. I extended encryption of user-level application pages to kernel stacks to prevent leaking AES state, tracking page accesses inside the kernel using in-kernel page faults.
- **University of Waterloo** Waterloo, ON
  H.B.Sc. Computer Science: Specialization in Bioinformatics *2007-2012*
  - Graduated with with Distinction, **89%** cumulative average

Publications

- Refereed Publications:
  - **Optimizing Data Collection in Deep Reinforcement Learning**
    James Gleeson, Daniel Snider, Moshe Gabel, Eyal de Lara, Gennady Pekhimenko
    MLBench 2022.
  - **MoIL: Enabling Efficient Incremental Training on Edge Devices**
    Jiacheng Yang, James Gleeson, Mostafa Elhoushi, Gennady Pekhimenko
    HAET 2021.
  - **RL-Scope: Cross-Stack Profiling for Deep Reinforcement Learning Workloads**
    James Gleeson, Srivatsan Krishnan, Moshe Gabel, Vijay Janapa Reddi, Eyal de Lara, Gennady Pekhimenko
    MLSys 2021.
  - **Heterogeneous GPU Reallocation**
    James Gleeson, Eyal de Lara
    HotCloud 2017.
  - **Crane: Fast and Migratable GPU Passthrough for OpenCL Applications**
    James Gleeson, Daniel Kats, Charlie Mei, Eyal de Lara
    SYSTOR 2017.
  - **Protecting Data on Smartphones and Tablets from Memory Attacks**
    Patrick J. Colp, Jiawen Zhang, James Gleeson, Sahil Suneja, Eyal de Lara, Himanshu Raj, Stefan Saroiu, Alec Wolman.
    ASPLOS 2015.
- Patents:
  - **Method and apparatus for protecting kernel control-flow integrity using static binary instrumentation**
    James Gleeson, Ahmed Azab, Wenbo Shen, and Rohan Bhutkar
    U.S. Patent 10,289,842, issued May 14, 2019.

Work Experience

- **Microsoft Research** Redmond, Washington
  Research Intern *Sep. 2015 - Nov. 2015*
  - Worked on FaRM distributed shared memory computing platform (extending work in NSDI 2014 paper). Existing implementation of fault tolerance in FaRM stores triply replicated copies of every object in the system, requiring a total of three RDMA writes to the primary the two backup servers. Investigated implementing erasure coding to reduce storage needed for storing replicas. Considered both block-based and object-based erasure coding schemes, settling on a block-based scheme to minimize coordination amongst replicas on the critical path of transactions.

  Samsung Research America Mountain View, California
  Research Intern *May. 2015 - Aug. 2015*
  - Worked on Samsung's ARM hypervisor based KNOX secure smartphone product. Designed and implemented a low-overhead and secure OS-level defense against kernel control-flow integrity attacks that attempts to achieve privilege escalation by overwriting return addresses to chain together Turing complete computations (return-oriented programming attacks).
  - Implemented fast XOR-based encryption of return addresses using static binary instrumentation of compiled kernel binary. Made use of abundant supply of ARM64 registers to securely stash XOR one-time pad encryption key. To protect against XOR key leakage to user-space, implemented re-generation of encryption key and re-encryption of return addresses at each user-to-kernel entry (system calls) and context switch. This research led a software patent.

  Innovative Medicine, Mount Sinai Hopsital Toronto, ON
  Software Developer *Oct. 2012 - Aug. 2013*
  - Elicited requirements by sitting down with molecular biologists to formalize what heuristics they apply to determine genotypes from raw SNP data.
  - Constructed an in-house pipeline for processing raw SNP data into phenotypes (e.g. reactivity to drugs) that inform doctors to create customized therapeutic treatments.
  - Pipeline was modeled as a dependency graph of stages backed by database tables allowing a declarative style of programming, real-time visualization of progress in a web front end, and reports informing doctors how therapeutic recommendations were generated from raw SNP data.
  - Delegated tasks to undergraduate coop students, getting them up to speed and contributing to the project.

  DemonWare Vancouver, BC
  Software Developer *Sept. 2011 - Dec. 2011*
  - Worked cooperatively and concurrently in a five person team, contributing a Bamboo continuous integration test suite that involved compiling and packaging software into RPMs for rapid deployment, and executing unit tests.
  - Quickly learned and utilized an in-house service-oriented Python framework to develop net services for games.

  The Hospital for Sick Children Toronto, ON
  Research Trainee in Bioinformatics *Jan. 2011 - April 2011*
  May 2010 - Aug. 2010
  Sept. 2009 - Dec. 2009
  - Surveyed and summarized scientific papers to determine top performing disease-gene prediction algorithms that make use of protein-protein interaction (PPI) networks, clarifying with authors when necessary.
  - Implemented top algorithms using a combination of Perl and MATLAB, and evaluated their predictive performance using leave-one-out cross-validation on OMIM and HPRD datasets.
  - Collaborated with members of the ProHits project at the University of Toronto to create Perl scripts for loading mass spectrometry (MS) data into a MySQL database, and for querying data in a format suitable for Significance Analysis of the Interactome (SAINT) software tools.

Course Projects

- **CSC2228: Mobile and Pervasive Computing**
  GPU Encrypt: AES Encryption on Mobile Devices *Sept. 2013 - Dec. 2013*
  - Used the general purpose GPU programming language OpenCL to implement AES on a Nexus 4 Android phone, performing GPU-specific optimizations to maximize throughput.
  - Benchmarked OpenCL against a CPU-based OpenSSL implementation, achieving a 1.79% speedup using the GPU (smaller than desktop benchmarks due to the abundance of cores on desktop platforms)
- **CSC2604: Human-Centered and Interdisciplinary Computing**
  Calm: Talking to Background Applications *Sept. 2013 - Dec. 2013*
  - Explored voice-activated interfaces for aiding interaction with background desktop applications without leaving the foreground application.
  - Extended Instant Messenger with voice commands to RESPOND to the last message sender, Music Player to play a TRACK selected using keyboard based autocomplete, and Window Manager with tile-based windowing commands (e.g. PUT TOP LEFT)
  - Addressed sensitivity and false positives by initiating a conversation with an application of interest while ignoring others, using keyboard for free-form dictation, and recording macros for common operations

Teaching Experience

- **University of Toronto** Toronto, ON
  Teaching Assistant *Sep. 2013 - Present*
  - CSC2228: Advanced Topics in Mobile, Pervasive Computing, and Edge Computing - *Fall 2019*
    CSC369: Operating Systems - *Winter 2016*
    CSC209: Software Tools and Systems Programming - *Fall 2014*
    CSC108: Introduction to Computer Programming - *Fall 2013, Winter 2013*

Awards and Interests

Bell Graduate Scholarship	2021 - 2022
	2020 - 2021
Vector Institute Research Grant	2020 - 2021
	2019 - 2020
NSERC Postgraduate Scholarships-Doctoral (PGSD) program	2015 - 2018
President's Entrance Scholarship (for 90-94.9% average)	2007
The Governer General's Academic Medal	2007

Technical Skills: C/C++, CUDA, PyTorch, TensorFlow, Docker, ARM Assembly, Python, Ruby, Perl, Java, JavaScript, MATLAB, MySQL, CMake, LaTeX, Linux

Interests: Fishing, attending concerts, gaming, functional programming, open-source software