Jiakun Yan

jiakuny3@illinois.edu • jiakunyan.github.io

Research Interests

My research interest lies in **parallel computing**, especially in its networking software layer. Currently, I am focusing on improving the multithreaded performance of communication libraries with task-based programming models as the target applications. I am the main developer of *Lightweight Communication Interface (LCI)* and also contributing to *MPICH* , *HPX* , and *Legion* .

Education

University of Illinois at Urbana-Champaign **Illinois, USA**
Aug. 2020 - Dec 2025 (Expected)

- Computer Science PhD student, advised by Marc Snir .
- Research around high-performance networking libraries, especially *Lightweight Communication Interface (LCI)* , and their application to task-based programming models.
- GPA: 4.0/4.0

Shanghai Jiao Tong University **Shanghai, China**
Sep. 2016 - Jun. 2020

- Bachelor's Degree of Engineering, Dept. of Computer Science.
- Zhiyuan Honors Program of Engineering (an elite program for top 5% talented students)
- GPA: 91.88/100 | Ranking: 4^{th}/151.

University of California, Berkeley **California, USA**
Jan. 2019 - May 2019

- Exchange student, Berkeley Global Access Discover Program, GPA: 4.0/4.0.

Experience

GPU Software, Legate Group **NVIDIA**
Software Engineer Intern, worked with Manolis Papadakis and Hessam Mirsadeghi *May. 2024 - Aug. 2024*

- Performance profiling and optimization for *Legion* UCX backend.

Programming Models and Runtime Systems Group **Argonne National Laboratory**
Research Intern, advised by Yanfei Guo *May. 2023 - Aug. 2023*

- Design and Evaluation of MPI Continuation in MPICH.

Programming Systems and Applications Research Group **NVIDIA Research**
Research Intern, advised by Michael Bauer and Michael Garland *May. 2022 - Aug. 2022*

- Realm Collective: design and implement collective communication operations in *Legion* Realm.

PASSION Lab **Lawrence Berkeley Laboratory**
Research Assistant, advised by Aydın Buluç and Katherine Yelick *Aug. 2019 - Jan. 2020*

- Asynchronous RPC Library (ARL): a high-throughput RPC system with node-level aggregation and single-node work-stealing.
- RDMA vs. RPC for Implementing Distributed Data Structures

Publication

- **Jiakun Yan**, Hartmut Kaiser, Marc Snir. *Understanding the Communication Needs of Asynchronous Many-Task Systems – A Case Study of HPX+LCI*, preprint, 2025.
- **Jiakun Yan**, Marc Snir. *Contemplating a Lightweight Communication Interface for Asynchronous Many-Task Systems*, WAMTA 2025 (to appear).
- Gregor Daiß, Patrick Diehl, **Jiakun Yan**, John K. Holmen, Rahulkumar Gayatri, Christoph Junghans, Alexander Straub, Jeff R. Hammond, Dominic Marcello, Miwako Tsuji, Dirk Pflüger, Hartmut Kaiser. *Asynchronous-Many-Task Systems: Challenges and Opportunities – Scaling an AMR Astrophysics Code on Exascale machines using Kokkos and HPX*, preprint 2024.

- **Jiakun Yan**, Hartmut Kaiser, and Marc Snir. *Design and Analysis of the Network Software Stack of an Asynchronous Many-task System – The LCI parcelport of HPX*, SC-W '23.

- Benjamin Brock, Yuxin Chen, **Jiakun Yan**, John Owens, Aydın Buluç, and Katherine Yelick. *RDMA vs. RPC for Implementing Distributed Data Structures*, IA^3 2019.

Project

Lightweight Communication Interface **UIUC**

Advised by Marc Snir *Aug. 2020 - Present*

- *Lightweight Communication Interface (LCI)* is a communication library and research tool focusing on multi-threaded and irregular use cases such as graph analysis, sparse linear algebra, and task-based runtimes. Major features include (a) orthogonal interface design with a wide range of options; (b) explicit user control of communication resources and behavior; and (c) centralized-contention-free critical paths.
- I am the main developer of LCI. LCI is integrated into *PaRSEC* and *HPX* and achieves superior performance compared to their original communication backends.

HPX over LCI **UIUC**

Advised by Marc Snir and Hartmut Kaiser *Aug. 2021 - Oct. 2023*

- *HPX* is a runtime system known for its support for the asynchronous task programming model. In this project, we added an LCI parcelport for HPX, enabling more direct support of HPX communication and scheduling logic.
- Compared to the MPI parcelport, the LCI parcelport achieves up to 100x improvement in microbenchmark and around 2x in a real-world application, Octo-Tiger (on NERSC Perlmutter with 1720 nodes).
- The LCI parcelport has been shipped with HPX releases since HPX 1.8.0 and used in a few HPX applications.

Collective Communication Operations in Realm **NVIDIA Research**

Advised by Michael Bauer and Michael Garland *May 2021 - Aug. 2021*

- Realm is an event-based low-level runtime system providing a high-performance asynchronous task execution model for the higher-level data-centric parallel programming system *Legion* . It offers the ability to perform memory copies across different data buffers, regardless of their physical location. Originally, Realm only supports point-to-point data copy operations. In this project, we extended the copy operation to handle collective broadcast communication.
- We designed and implemented a hierarchical path planning algorithm that includes inter-node radix tree broadcast and intra-node path aggregation. We used a set of synthetic benchmarks to evaluate the broadcast operations and found it achieved significant improvement compared to the original point-to-point copies. (The actual speedup number depends on the benchmark setup.)

TaskFlow: Task-based Runtime on Distributed-memory System **UIUC**

Advised by Josep Torrellas and Marc Snir , CS533 course project *Jan. 2021 - May 2021*

- TaskFlow is a simple but efficient task-based runtime for distributed-memory systems. It adopts the PTG-based task programming model that enables reduced time/memory overhead and fine-grained synchronization. It executes tasks according to an explicit task dependency graph and uses active messages to proactively signal remote tasks.
- We implemented TaskFlow based on Argobots and MPI. We performed a collection of micro-benchmarks and mini-applications to evaluate the performance of its various configurations and compare it with two established PTG-based task systems, TaskTorrent and PaRSEC. The benchmark results showed that TaskFlow generally achieved the best performance under various circumstances.

Honors and Awards

- **Best Poster Award**, WAMTA24 Feb. 2024
  A Lightweight Communication Interface for Asynchronous Many-Task Systems

Skills

- **Programming Language:** C (proficient), C++, Python, Java, Rust, Go
- **Library & Framework:** libibverbs, libfabric, UCX, MPI, GASNet-EX, UPC++, OpenSHMEM, Argobots