YONGQIN WANG

+1 (425)-445-5266|yongqin@usc.edu| Pasadena CA. 91103 | Google Scholar

Education

University of Southern California (Los Angeles, CA)
Ph.D. Student in Computer Engineering GPA: 3.78
Research Direction: Computer Architecture and Systems

University of Washington (Seattle, WA) June 2019
Bachelor of Science in Electrical Engineering GPA: 3.77
Specialized in Computer Architecture and Embedding Systems

Skills

- Machine Learning Frameworks: Pytorch, Tensorflow
- Hardware Languages & Tools: SystemVerilog, Verilator
- Programming Languages: C, C++, Python

Research Areas

- Trusted Execution Environments
- Oblivious RAM
- Multi-Party Computing

Internship Experience

Research Intern **Pasadena, CA**
Meta AI *May 2021 - December 2021*
Studies major Transformer-based model inference runtime overheads and potential optimizations when Multi-party computing (MPC) is implemented. Results are published in ISPASS 2022.

Papers

MPC-Pipe: an Efficient Pipeline Scheme for Secure Multi-party Machine Learning Inference. **[under submission]**
Authors: ***Yongqin Wang***, Rachit Rajat, Murali Ananvaram
Available: https://arxiv.org/pdf/2209.13643.pdf
MPC-Pipe is a novel efficient MPC framework that utilizes communication and computation overlaps to reduce ML model inference runtime latency. There are three major pipeline schemes introduced in the paper: 1) inter-linear pipeline, 2) inner-layer pipeline, and 3) inter-batch pipeline. MPC-Pipe achieves 33% throughput improvement and 13% latency improvement for the state-of-art DNNs.

LAORAM: A Look Ahead ORAM Architecture for Training Large Embedding Tables. **[ISCA 2023]**
Authors: ***Yongqin Wang****, Rachit Rajat*, Murali Ananvaram
Available: https://arxiv.org/pdf/2107.08094.pdf
LAORAM proposes a private training method for CPU-based large embedding tables using ORAM. LAORAM proposes an aggressive superblock formation mechanism that significantly reduces the number of access to the CPU-based large embedding tables and uses a fat-tree organization to mitigate contentions over the stash in the GPU client. Those schemes combined can reduce the ORAM access latency by up to 5.4x.

PageORAM: An Efficient DRAM Page Aware ORAM Strategy. **[MICRO 2022]**
Authors: Rachit Rajat, ***Yongqin Wang***, Murali Ananvaram
Available: https://ieeexplore.ieee.org/abstract/document/9923803
In this work, we introduce a new method to read paths in PathORAM to reduce stash management costs by fetching additional data nodes in the same subtree in the ORAM tree organizations to have more opportunities for data block evictions, reducing the contention to the stash space.

Characterization of MPC-based Private Inferences for Transformer-based Models
[ISPASS 2022 & Neurips 2021 PriML Workshop]
Authors: ***Yongqin Wang*****,** G Edward Suh, Wenjie Xiong, Benjamin Lefaudeux, Brian Knott, Murali Annavaram, Hsien-Hsin S Lee

Available: https://ieeexplore.ieee.org/document/9804616
In this work, we provide an in-depth character study of the performance overhead for running the now popular Transformer models with secure multi-party computing (MPC). Three unique challenges are identified: 1) significant **Softmax** runtime, 2) significant embedding table lookups, and 3) fixed point numerical stability issue.

Byzantine-Robust and Privacy-Preserving Framework for FedML. [ICLR Workshop 2021]
Authors: Hanieh Hashemi, ***Yongqin Wang***, Murali Annavaram
Available: https://aisecure-workshop.github.io/aml-iclr2021/papers/15.pdf
This is a federated machine learning framework that provides security against a subset of malicious clients that may send inaccurate gradient data to undermine model accuracy and convergence, and information-theoretic data privacy clients cannot access other clients' gradients. Untrusted servers only can access encoded gradients. We provide a rigorous analysis to guarantee bounds of information leakage are infinitesimally small.

DarKnight: A Data Privacy Scheme for Training and Inference of Deep Neural Networks. **[MICRO 2021]**
Authors: Hanieh Hashemi, ***Yongqin Wang***, Murali Annavaram
Available: https://dl.acm.org/doi/abs/10.1145/3466752.3480112
This work provides a cloud machine learning training & inference computation scheme that achieves input image privacy and performance improvements. We use coded computing and **trusted execution environments** to achieve input image privacy and GPU to achieve performance gains.

Origami inference: Private inference using hardware enclaves. **[IEEE CLOUD 2021]**
Authors: Krishna Giri Narra, Zhifeng Lin, ***Yongqin Wang***, Keshav Balasubramanian, Murali Annavaram
Available: https://ieeexplore.ieee.org/document/9582200
This work provides a cloud machine learning inference computation mode, which distributes layers in machine learning model inference to different computation units to achieve performance and input image privacy. Layers whose inputs are highly correlated with original images are computed on trusted hardware to achieve privacy and layers whose inputs have low correlation with original images are computed on GPU to improve performance. Our model can achieve considerate performance improvements on our baseline.

Project Experiences

Purple Jade Processor (An out of order ArmV6 Processor) **Seattle, WA**
University of Washington/ Capstone Project *April 2019 - June 2019*
- *https://github.com/iihihiuh/Purple_Jade (Please take a look! It contains more detailed documentation)*
- Key Features:
  - Physical Register Renaming, FSM Replica for mis-prediction roll back mechanism
  - Tomasulo Algorithm, Issue Table
  - Store Order Buffer, Reorder Buffer
- Implemented An ArmV6 ISA Simulator and detailed Test Suites for verification (**C++**)
- Trace Replay using Verilator and VCS for Functional and Physical Verification
- Around 74 FO4, Synthesizable Verilog, Post Synthesize verified

Black Parrot Project (RISC-V multicore with Linux capabilities) **Seattle, WA**
Bespoke Silicon Group / Advised by Prof. Michael Taylor *July, 2018 - March 2019*
- Worked on defining interfaces between processor front-ends and back-ends
- Worked on documenting Linux booting systems on RISC-V architectures
- Worked on back-end inter-module-interfaces and back-end design
- Worked on firmware writing

Random RISC-V Instruction Test Infrastructure **Seattle, WA**
University of Washington
- An instruction generator that produces random instructions from RISC-V to make RTL design more robust
- Provided program that runs generated random tests to produce a trace file
- Trace files from the customized simulator are used to indicate when processors first made a mistake
- Utilizes Systemc verification library
- Key Features:
  - Infinite loop exclusion
  - Compatible with RISC-V testing infrastructure (riscv-gcc and spike simulator)