Peng Cheng

Email: pcpeng26@gmail.com | Homepage: https://pengmacro.github.io/

RESEARCH INTERESTS:

- **Systems (Operating, Storage, Distributed Systems, Software Engineering)**
- **Interactions between Systems and Machine Learning**

EDUCATION

The University of Chicago Chicago, IL
Pre-Doctoral MS in Computer Science Sep. 2019 - Dec. 2020

- Overall GPA: 3.7/4.0; Systems GPA: 3.8/4.0
- Received a **merit-based research scholarship** equivalent to 50% of total tuition

The University of Wisconsin-Madison Madison, WI
B.S. Electrical Engineering & Computer Sciences Sep. 2015 - May 2019

- Overall GPA: 3.6/4.0; Major GPA: 3.7/4.0
- Dean's Honor List: 2016 - 2017, Fall 2017 - 2018

PUBLICATION

Storage Benchmarking with Deep Learning Workloads
Peng Cheng, Haryadi S. Gunawi. (In preparation)
An early draft of this workshop paper is available on my homepage.

RESEARCH EXPERIENCES

Study of CORTX | Group Leader Aug. 2020 - Present
UChicago UCARE Group | Advisor: Prof. Haryadi S. Gunawi (UChicago) and John Bent (Seagate)

- Built and tested various modules of CORTX, a distributed object storage system.
- Collaborated with Seagate Engineers to debug compilation errors.
- Hacked and modified dataflow of CORTX Motr module.
- Evaluated CORTX transaction modules using Distributed system Model CheckKing (DMCK).

Storage Benchmarking with Deep Learning workloads | Independent Study April - Aug. 2020
UChicago UCARE Group | Advisor: Prof. Haryadi S. Gunawi (UChicago)

- Benchmarked data loading performance in two object storage systems (MinIO, Ceph) and three key-value storage systems (MongoDB, Redis, Cassandra) using MNIST and CIFAR-10 Datasets.
- Evaluated the impact of different access patterns, data locations, data formats, and storage disaggregation granularity on data loading performance.

External Memory Numpy Implementation | Group Leader May 2018 - May 2019
ADvanced Systems Laboratory (ADSL) | Advisor: Prof. Remzi H. Arpaci-Dusseau (UW)

- Configured experiment environment with various computing memory and disk requirements.
- Applied Linux toolkits (cgroups, blktrace) to control the memory size and trace I/O performance.
- Analyzed Dask, a popular library supporting out-of-core algorithms through dynamically tracing workflow for various numpy operations, as well as decomposing their running time.
- Identified potential bottleneck (unnecessary repacking data in the optimization process) of Dask.

Hybrid Multiplier Implementation | Independent Study Jan. - May 2018

Wisconsin Computational Intelligence Lab (WiCIL) | Advisor: Prof. Li Jing (UW)

- Implemented classical multipliers (in Verilog), such as Booth, Wallace tree multiplier.
- Measured and analyzed those multipliers with respect to their latency and area.
- Implemented a hybrid multiplier with low latency through integrating Booth multiplier and Wallace tree multiplier.

COURSE PROJECTS

Performance Evaluation of Distributed Deep Learning: A Networking Perspective Oct. - Dec. 2020

- Benchmarked and evaluated the performance of various deep learning models with Tensorflow based on network latency, network bandwidth, and packet loss.
- Observed computation-intensive models like CNN are more sensitive to the change of network condition.
- Discovered fault tolerance is inefficient in the mainstream deep learning frameworks, and users need to store the intermediate training status using checkpoint mechanisms.

Pipelined Processor Implementation Mar. - June 2018

- Implemented (in Verilog) a 5-stage pipelined processor containing a set of 16 instructions specified for a 6-bit data-path with load/store architecture.
- Built module with data forwarding to increase IPC and static branch prediction.
- Designed and implemented a cache with LRU eviction algorithm and write-through policy.

Flight Controls of a Quadcopter Sep. - Dec. 2017

- Designed and implemented (in System Verilog) the flight controls of a quadcopter receiving commands wirelessly via Bluetooth to control the speed of 4 motors.
- Implemented communication protocols (UART, SPI, and I2C) and inertial sensor interface.
- Built PID control scheme and Analog to Digital Converters (ADC) module.

TECHNICAL SKILLS

Testbed: Emulab Cluster, Chameleon Cloud

OS: **Hacking** Linux kernel

ML/AI: Tensorflow, Keras, PyTorch

Systems Hacking: Spark, Hadoop, HDFS, Dask, CORTX

Using: Redis, Cassandra, Ceph, MinIO, MongoDB

Programming Languages: C, Python, Java, MATLAB, Verilog/System Verilog, SQL