

# YANG YANG

## PERSONAL INFORMATION

---

homepage [Elio-yang.github.io](https://Elio-yang.github.io)  
official email [yangyang1519@mails.jlu.edu.cn](mailto:yangyang1519@mails.jlu.edu.cn)  
personal email [jluelioyang2001@gmail.com](mailto:jluelioyang2001@gmail.com)  
github [Github.com/Elio-yang](https://Github.com/Elio-yang)  
address Jilin University, 2699 Qianjin Street, Changchun, Jilin

## EDUCATION

---

*Bachelor of  
Computer Science  
and Technology*

*Jilin University, Changchun, China* *Feb. 2019 – Present*  
GPA: 3.69/4.0  
Rank: 9%  
Major: Computer Science and Technology  
Interests: Computer Architecture · Compiler · High Performance Hardware · Machine Learning System

## PUBLICATION

---

**Yang Yang**, Xueying Wang, Guangli Li\*. *Facilitating Profile Guided Compiler Optimization with Machine Learning*. In Student Research Competition of the 21<sup>st</sup> IEEE/ACM International Symposium on Code Generation and Optimization. [\[Under Review\]](#) [\[Poster\]](#)

- Achieving an average of  $1.03\times$  and  $1.95\times$  speedups on representative real-world applications and *Polybench* benchmark suite over the baseline (i.e., the programs without PGO), respectively.
- The performance of our machine learning-aided PGO is very close to the classic PGO ( $1.05\times$  and  $1.97\times$  speedups over the baseline) while reducing 58.3% and 94.8% optimization costs.

## RESEARCH EXPERIENCE

---

*ETECA Lab*

[Emerging Technology Enabled Computer Architecture Lab](#) *Feb. 2022 – Present*  
*Jilin University, Changchun, Jilin, P.R.China*  
Research Assistant, Advisor: [Prof. Jingweijia TAN](#)  
Research on: GPU Architecture & Reliability & Energy Efficiency & Accelerator  
What We Do:

- Extended the **microarchitecture** of General-Purpose Graphics Processing Unit (**GPGPU**).
- Explored the **process variation** of **MCM**-GPUs based on **FinFET** and state-of-the-art **chiplet** technology.
- Exploited the potential of **FPGA** for building open-sourced GPU like **Vortex**.

**Project: LLAM: A Low-Level Power Modeling and Prediction Framework for Nvidia Ampere GPU**

- Implemented a **Low-Level Analysis** and **Modeling** framework for **NVIDIA** Ampere GPU.
- Applied **deep learning** techniques for accurate power modeling.
- Examined the power-level effect of the instruction **control flag** when generating the SASS.

*SKL Computer  
Architecture*

[State Key Laboratory of Computer Architecture](#) *Jul. 2022 – Present*  
*Institute of Computing Technology, Chinese Academy of Science, Beijing, P.R.China*  
Research Assistant, Advisor: [Prof. Guangli Li](#)  
Research on: Compiler & Programming Systems & Deep Learning  
What We Do:

- Improved the **optimization** ability of compilers based on application's **run-time** characteristics.
- Using **machine learning** methods to guide the **LLVM** compiler for better machine code **generation**.

**Project: Facilitating Profile Guided Compiler Optimization with Machine Learning**

- Formed a classification task based on over 2,000,000 branches distribution.
- Proposed a **branch predictor** using **XGBoost** based on **static** features.

- Explore the speedup sensibility of different programs towards different feature design.

## SKILLS

Languages	C/C++ · Assembly (x86, RISC-V) · Python · Go
Frameworks	CUDA · Pytorch · LLVM
Hardware	Verilog · Vivado · FPGA
Software	LINUX/Windows · L <sup>A</sup> T <sub>E</sub> X · Markdown · GNU compiler (gcc, etc.) · gpgpu-sim · Varius-TC

## AWARDS

Undergraduate Academic Year Scholarship	The First Class Fellowship	Sept. 2020
	The Second Class Fellowship	Sept. 2021
	The Third Class Fellowship	Sept. 2022

## PROJECTS

MapReduce Engine	<p><i>MapReduce Engine</i> is a <b>Go</b> language implementation of the paper<sup>1</sup>. <span style="float: right;">Apr. 2022</span></p> <ul style="list-style-type: none"> <li>• <b>Fault tolerance</b> (failures like crash and communication-lose of workers) master and a worker cluster.</li> <li>• Characterized cluster size and working functions (mapf &amp; reducef).</li> <li>• Communicate with the master through <b>Remote Procedure Call</b>.</li> </ul> <p>This Engine is a basic component for building a large-scale distributed system. [Codes <a href="#">here</a>.]</p>
EOS	<p><i>EOS</i> is a 32bit *nix operating system developed in <b>C</b> language. <span style="float: right;">Sept. 2021</span></p> <ul style="list-style-type: none"> <li>• Basic <b>bootloader</b>, 2-level <b>paging</b>, 4GB <b>memory management</b> and <b>kernel multithreads</b>.</li> <li>• Provide a set of traditional shell programs and <b>multi-process</b> mechaism.</li> <li>• Follow the <b>x86 ABI</b>, so it's easy to port those x86 applications.</li> </ul> <p>This project is still <i>active</i> and it will provide a library and compiler support in the future. [Codes <a href="#">here</a>.]</p>
WYZ-BAR	<p><i>WYZ-BAR</i> is a bar management system developed in <b>C</b> language. <span style="float: right;">Mar. 2020</span></p> <ul style="list-style-type: none"> <li>• WYZ-BAR is a <i>collaborative project</i> (WYZ stands for 3 members and Y is for me) and I am the leader.</li> <li>• <b>Multi-process</b> organization for effective system building.</li> <li>• Follow the <b>x86 ABI</b>, so it's easy to port those x86 applications.</li> <li>• Re-implemented a simple <b>sqlite style database</b>.</li> <li>• Used lots of <b>parsing</b> techniques for input checking.</li> </ul> <p>WYZ-BAR is my <i>first</i> course project in the university. [Codes <a href="#">here</a>.]</p>
Others	<p>You can find more projects including course labs (like MIT 6.828), Android application (SmogDetector), CUDA operators (FFT) <i>etc.</i>, in <a href="#">GitHub</a>.</p>

## OTHER INFORMATION

Languages	CHINESE · Native proficiency.
	ENGLISH · Professional proficiency.
Interests	Literature (Latin-American, magic realism) · Physics · NBA (Golden State Warriors) · Classical (Chopin)
Characteristic	Strong patience · Highly self-motivated · Creative · Communication and collaboration skilled.

<sup>1</sup> J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008, doi:[10.1145/1327452.1327492](https://doi.org/10.1145/1327452.1327492).