COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

DISSERTATION DEFENSE

SUNGHYUN PARK

Compiler Auto-tuning For Code Optimization

Friday, September 10, 2021
10:30am - 12:30pm
Virtual (Passcode: 197695)

ABSTRACT: To deliver the best performance to users, modern compilers apply hundreds of optimizations that transform a program into a more efficient form. Since a program execution is a complicated process of the delicate interplay between software and hardware, each compiler optimization should be carefully determined with consideration for its trade-offs.

Today, most of the important optimization decisions are made by hand-crafted heuristics which often largely depend on the developers' expertise. However, as the system complexity continues to increase, such manual approach often overly simplifies interactions between diverse system components and results in the failure to achieve maximum performance. Furthermore, a huge amount of time and cost need to be repeatedly invested for this manual tuning process whenever one of the system components is updated.

To attack these challenges, this thesis proposes a suite of auto-tuning methods that can successfully improve optimization decisions inside state-of-art compilers. By focusing on one of the most representative compiler optimizations, the first part of this thesis suggests a methodology that automatically constructs the best affordable decision model for the dynamic binary translator in a mobile system. By effectively learning the patterns between optimal decisions and workload features, this method significantly outperforms the best heuristics handwritten by industry experts. Next, a group of optimizations is considered. To identify the best use of existing optimizations, the second part proposes an intelligent pure search method, called SRTuner, which customizes effective optimization settings for each workload by exposing important inter-optimization relations. Finally, the last work of this thesis proposes Collage which is an auto-tuning system that attacks the practical problem of identifying the best mixed use of diverse backends to run deep learning workloads. The experimental results demonstrate that this system efficiently customizes a fast execution plan that outperforms the hand-written strategies in the existing deep learning frameworks.

CHAIR: Prof. Scott Mahlke