

Case ID:M22-291P^

Published: 10/18/2023

Inventors

Shail Dave

Aviral Shrivastava

Tony Nowatzki

Contact

Physical Sciences Team

Explainable Design Space Exploration for Efficient Hardware/Software Codesigns

Computing system designers need to characterize the execution of applications on a computing system (e.g., on a processor) and they also need to improve the design (i.e., find better hardware/software configurations for the computing system such as one that produces more throughput at lower energy consumption). Existing mechanisms for computing systems' design, simulation, or optimization are non-explainable, as in they do not inform users about why a certain configuration could lead to a certain cost. For instance, a designer cannot directly infer why a hardware/software configuration of a processor takes a particular amount of time (or energy or chip area) to process the application. This slows down the productivity of the designer.

Likewise, existing methods for optimizing computing system design explore numerous configurations without ever reasoning about why a certain configuration could lead to a certain execution cost. As a result, obtained configurations after optimizations can be less efficient (sometimes even several-fold) as most of the explored configurations during optimizations are random trials without solid reasoning. They also cannot explain why their obtained solutions are optimal ones when establishing the optimality is infeasible due to a vast search space of solutions (e.g., quadrillion solutions). Thus, there is a need to make designs and optimizations of computing systems explainable inherently, and thereby achieve explainable and efficient hardware/software designs quickly.

Researchers at Arizona State University (ASU) and the University of California, Los Angeles (UCLA) have developed a computing system design optimization method for achieving explainability and bottleneck-mitigating optimization. This technique uses bottleneck analysis for achieving explainability and provides a methodology for making such cost analysis and bottleneck-mitigating optimization for computing system design. It can explain why a hardware or software configuration of a computing system leads to a particular cost for an application's execution.

The technique also provides a generic methodology and framework for explainable optimization of an arbitrary cost function. Unlike most existing optimization tools, this optimization method can perform multi-functional optimization. Bottlenecks can be accurately pinpointed when the total cost is produced by multiple factors (e.g., when different parts of the application program to be executed have different computational/memory characteristics).

Potential Applications:

- Computing system design optimization tool for processors, DNN accelerators, ML accelerators, etc.

Benefits and Advantages:

- Leads to explainable design analysis, simulation, and optimization of computing systems, for example:
 - Explanations as to why a hardware or software configuration of a computing system leads to a particular cost for an application's execution
 - Explainable optimization of an arbitrary cost function
 - Performs multi-functional optimization (i.e., when a single solution is required for reducing multiple costs)
- Achieves several-fold efficient hardware and software configurations of a computing system
- Converges to efficient solutions more quickly (tens of iterations vs. thousands)
- Can accurately pinpoint bottlenecks when total cost is produced by multiple factors