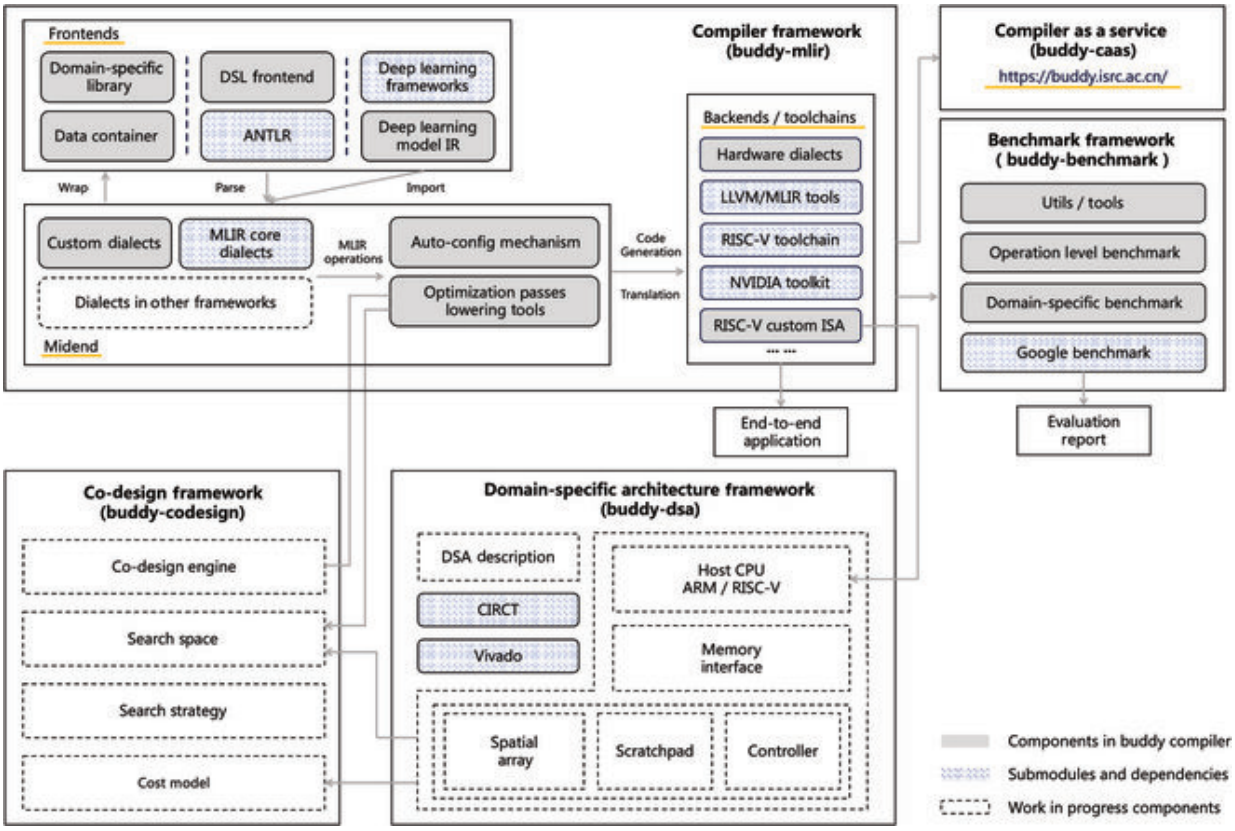
How computers and artificial intelligence evolve together

June 30 2023



The Buddy Compiler framework, a work in progress, will include a benchmark framework, a domain-specific architecture framework, a co-design module, and a compiler-as-a-service platform in addition to a compiler framework. Credit: Hongbin Zhang et al.

Co-design, that is, designing software and hardware simultaneously, is

one way of attempting to meet the computing-power needs of today's artificial intelligence applications. Compilers, which translate instructions from one representation to another, are a key piece of the puzzle. A group of researchers at the Chinese Academy of Sciences summarized existing compiler technologies in deep learning co-design and proposed their own framework, the Buddy Compiler.

The group's review paper was published in the journal *Intelligent Computing*.

Although others have summarized optimizations, hardware architectures, co-design approaches, and compilation techniques, no one has discussed deep learning systems from the perspective of compilation technologies for co-design. The researchers studied deep learning from this angle because they believe that "compilation technologies can bring more opportunities to co-design and thus can better achieve the performance and power requirements of deep learning systems."

The review covers five topics:

- The history of deep learning and co-design
- Deep learning and co-design now
- Compilation technologies for deep learning co-design
- Current problems and future trends
- The Buddy Compiler

The history of deep learning and co-design

Since the 1950s, neural networks have gone through many rises and falls leading up to today's explosive growth of deep learning applications and researches. Co-design began in the 1990s and has since then been adopted in various fields, progressing from manual work to computer-aided design and ultimately becoming a complex process involving

modeling, simulation, optimization, synthesis, and testing.

Since 2020, a network model called a transformer has seen great success: ChatGPT is a chatbot built using a "generative pre-trained transformer." Current AI applications like ChatGPT are reaching a new performance bottleneck that will require hardware-software co-design again.

Deep learning and co-design now

The breakthrough of deep learning comes from the use of numerous layers and a huge number of parameters, which significantly increase the computational demands for training and inference. As a result, relying solely on software-level optimization, it becomes challenging to achieve reasonable execution times. To address this, both industry and academia have turned to domain-specific hardware solutions, aiming to achieve the required performance through a collaborative effort between hardware and software, known as hardware-software co-design.

Recently, a comprehensive system has emerged, comprising deep learning frameworks, high-performance libraries, domain-specific compilers, programming models, hardware toolflows, and co-design techniques. These components collectively contribute to enhancing the efficiency and effectiveness of deep learning systems.

Compilation technologies for deep learning co-design

There are two popular ecosystems that are used to build compilers for deep learning: the tensor virtual machine, known as TVM, and the multi-level intermediate representation, known as MLIR. These ecosystems employ distinct strategies, with TVM serving as an end-to-end deep learning compiler and MLIR acting as a compiler infrastructure. Meanwhile, in the realm of hardware architectures customized for deep

learning workloads, there are two primary types: streaming architecture and computational engine architecture.

Hardware design toolflows associated with these architectures are also embracing new compilation techniques to drive advancements and innovations. The combination of deep learning compilers and hardware compilation techniques brings new opportunities for deep learning co-design.

Current problems and future trends

With performance requirements increasing too fast for processor development to keep up, effective co-design is critical. The problem with co-design is that there is no single way to go about it, no unified co-design framework or abstraction. If several layers of abstraction are required, efficiency decreases. It is labor-intensive to customize compilers for specific domains. Unifying ecosystems are forming, but underlying causes of fragmentation remain. The solution to these problems would be a modular extensible unifying framework.

The Buddy Compiler

The contributors to the Buddy Compiler project are "committed to building a scalable and flexible hardware and software co-design ecosystem." The ecosystem's modules will include a compiler framework, a compiler-as-a-service platform, a benchmark framework, a domain-specific architecture framework, and a co-design module. The latter two modules are still in progress.

The authors predict continued development of compilation ecosystems that will help unify the work being done in the rapidly developing and somewhat fragmented field of deep learning.

More information: Hongbin Zhang et al, Compiler Technologies in Deep Learning Co-Design: A Survey, *Intelligent Computing* (2023). DOI: 10.34133/icomputing.0040

Provided by Intelligent Computing