Clang/LLVM + offload

Yuri Takigawa

June 24, 2025

The university of Tokyo, EEIC, Taura Lab

Contents

Why LLVM/Clang + offload (ja)

This is what I wrote in the research plan (in Japanese):

> 近年、身近になった *LLM* を支える *AI* ハードウェアアクセラレータは、計算速度・メモリ帯域・メモリ容量・スケジューラなどの面で目覚ましい進歩を遂げている。一方で、これらのハードウェアアクセラレータの性能を最大限に引き出すには、低レベルのプログラミングが求められることが多く、ユーザビリティの向上との両立は依然として大きな課題である。ハードウェアの優れた抽象化と、ユーザプログラムを効率的に最適化・変換する言語処理系（コンパイラ）の重要性は今後さらに高まっていくと考えている。特に、*NVIDIA GPU* に特化した *CUDA* が事実上の標準となりつつある現状において、アーキテクチャの汎用性を高める観点からも、記憶階層の使用最適化や *Tensor Core* をはじめとする特殊ハードウェアの活用最適化の研究が重要であると考えている。

Why LLVM/Clang + offload (en)

In recent years, AI hardware accelerators that support LLM, which has become more accessible, have made remarkable progress in terms of computing speed, memory bandwidth, memory capacity, schedulers, and other aspects. On the other hand, in order to maximise the performance of these hardware accelerators, low-level programming is often required, and balancing this with improved usability remains a major challenge. We believe that the importance of excellent hardware abstraction and language processing systems (compilers) that efficiently optimise and convert user programs will continue to grow in the future. The importance of excellent hardware abstraction and language processing systems (compilers) that efficiently optimise and convert user programs is expected to grow even further in the future. Especially in the current situation where CUDA, specialised for NVIDIA GPUs, is becoming the de facto standard, research on optimising the use of memory hierarchies and special hardware such as Tensor Cores is important from the perspective of enhancing architectural versatility.

Setup and Compile option

Contents

How to write offload code

aa

A Conceptual View of LLVM/Clang + offload

References

1. cerebras SDK Documentation (1.4.0)
2. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#
3. https://resources.nvidia.com/en-us-data-center-overview-mc/en-us-data-center-overview/grace-hopper-superchip-datasheet-partner
4. https://docs.nvidia.com/cuda/cuda-runtime-api/index.html
5. https://www.nas.nasa.gov/hecc/support/kb/basics-on-nvidia-gpu-hardware-architecture_704.html
6. https://developer.nvidia.com/blog/unified-memory-cuda-beginners/
7. https://leimao.github.io/blog/CUDA-Coalesced-Memory-Access/
8. https://www.nvidia.com/content/pdf/fermi_white_papers/p.glaskowsky_nvidia's_fermi-the_first_complete_gpu_architecture.pdf
9. https://developer.nvidia.com/ja-jp/blog/nvidia-hopper-architecture-in-depth/
10. https://qiita.com/tarako1889/items/963e8972daa8c490efd4