2023-3-15

Agenda

Hugginface and Intel partnering to democratize ML hardware acceleration	Julien Simon, Hugginface Matrix Yao, Intel
Joint Matrix: A Unified SYCL Extension for Matrix Hardware Programming	Dounia Khaldi, Intel

Attendees

Zhu, Wei2	Intel	Ye, Jason Y	Intel
Zhu, Hong	Intel	Sunita (AWS)	AWS
Yang, Qun	Intel	Qin, Zhennan	Intel
Huang, Mingxiao	Intel	Lv, Tao A	Intel
Wang, Quintin	Intel	Wang, Chuanqi	Intel
Yu, Guangye	Intel	Davanlou, Ramtin	Accenture
Wang, Dewei	Intel	Cheng H. Lee	Anaconda
Julien Simen	Hugginface	Jin, Youzhi	Intel
Cohn, Robert S	Intel	Chen, Ciyong	Intel
Karasev, John	Intel	Narayanamoorthy, Srinivasan	Intel
Zheng, Jiexin	Intel	Zheng, Hongming	Intel
Gu, Yonghao	Intel	Guoliang @ Vastai	Vastai
Gouicem, Mourad	Intel	Cave, Vincent	Intel
Li, Jian Hui	Intel	Sheng Zha (Guest)	AWS
Chen, Chao	Intel	Palangappa, Poovaiah M	Intel
Aananthakrishnan, Sriram	Intel	Yao, Matrix	Intel
He, Jianhang	Intel	Parikh, Ratnam	Intel
Khaldi, Dounia	Intel	Kazakov, Sergey	Intel
Shahneous Bari, Md Abdullah	Intel	Wu, Shufan	Intel
Ling, Liyang	Intel	Balas, DorotheeX Marie Clotilde	Intel
Richards, Alison L	Intel	Zhong, Zhicong	Intel
Rod Burns	Codeplay	Zhang, Rong A	Intel
Mehdi Goli	Codeplay	Emani, Ashok	Intel
Kawakami, Kentaro/川上 健太郎	Fujitsu	Jayaram Bobba	Habana
Penporn Koanantakool (Guest)	Google	Deshpande, Gauri1	Intel
Lee, Sang Ik	Intel	Prajapati, Dimpalben R	Intel

Zhao, Patric	Intel	Mitchell, Frost	Intel
Frank Brill	Cadence	Shi, Yuankun	Intel
Mao, Harry	Intel		

Julien Simon and Matrix Yao presented "Hugginface and Intel partnering to democratize ML hardware acceleration"

What would be the next step for Hugginface?

We are constantly catching up with new models, and new neural network architecture coming up every day. We make sure they are supported by tools, like quantization and pruning, and make them really easy to use.

We see a lot of solutions for using CPU. Sometimes small models are more cost-effective running on CPU. Cost is a concern. The workflow is usually first training the model on GPU, then pruning it, and then using it on CPU. AWS supports Sapphire Rapids , as well as Google Cloud.

Are there any use cases for FP8 and what is accuracy?

Hardware like Habana Gaudi supports FP8, so we are enabling it. The accuracy goal is to be close to FP32. Lots of use cases are in FP32 and BF16, and FP8 give more options for Data Scientist to find the right mix of precision in the model.

Dounia Khaldi presented "Joint Matrix: A Unified SYCL Extension for Matrix Hardware Programming"

Where can I find the LLVM extension and SPIR-V extension for the joint matrix?

You can find SPIR-V extension in the public Intel LLVM repo (https://github.com/intel/llvm/blob/sycl/sycl/doc/design/spirv-extensions/SPV_INTEL_joint_matrix.asciidoc)

The LLVM IR extension for joint-matrix is not public. There was a matrix extension in the LLVM IR, and we find it very limited as it uses vector, and layout and shape are not extensible, with no concept of scope. We are working with broader academic and industry partners to further improve matrix extension in LLVM IR.

Can we apply Joint matrix to other languages?

Yes. Joint matrix is designed to be general so it works on a different type of matrix hardware unit. For example, it would still work when the new hardware evolves to add asynchronous behavior like in Nvidia GPUs. Joint matrix works on intel CPU, intel GPU, and Nvidia GPU, and Codeplay is porting it to AMD GPU

The Joint matrix concept should apply to other languages. For example, in C++ augmented with the executors proposal (std::execution) which allows your code to run on the GPU, a variant of joint matrix can be used there. It might be a challenge for a language that already has its own matrix API.