CS/ECE 570 Project, Winter 2024

Heterogeneous Computing in an AI Context

Amber Kahklen and Ninad Anklesaria

Outline

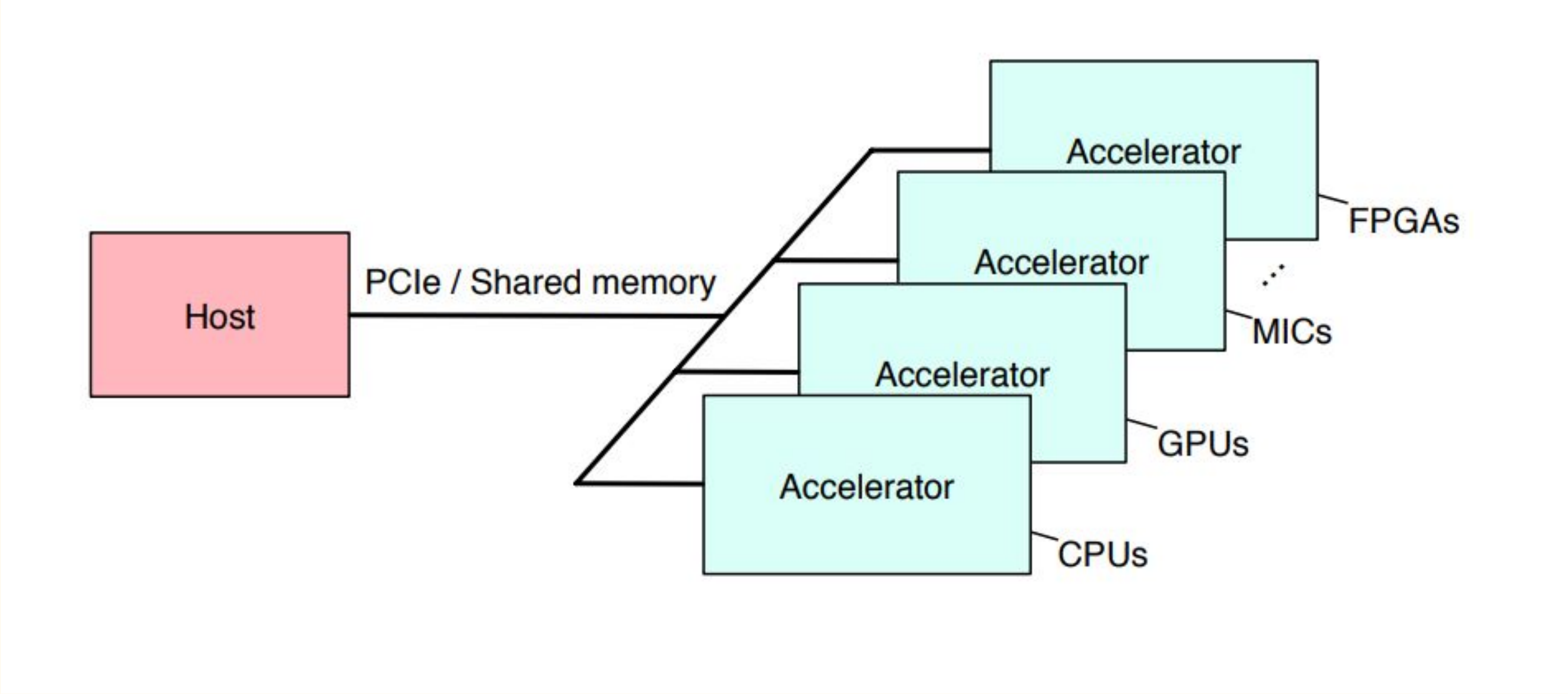
Introduction

- Heterogeneous Computing (HC) and Artificial Intelligence (AI) often go hand in hand
- Uncommon to find AI that doesn't use heterogeneous computing
    - Especially with Machine Learning and Deep Learning
- Survey on different implementations: HC for AI & AI for HC
    - Impact these two topics have on each other
    - challenges, benefits, and limitations
    - Infrastructure and framework: How is it implemented?

Background - Heterogeneous Computing (1)

- Computing architecture
- Incorporates multiple types of processors or cores in one system
- Each specialized for different task
- Leverages strengths of various processing units:
    - CPUs, GPUs, DSPs, and FPGAs
- To optimize:
    - Performance
    - Energy Efficiency
    - Computational Speed
- More effective than homogeneous computing for many applications

Background - Heterogeneous Computing (2)



Background - Heterogeneous Computing (3)

History

- Became more prominent late 2000's early 2010's
- One of the 1st major milestones: NVIDIA's CUDA (Compute Unified Device Architecture)
    - parallel computing platform and programming model
    - Released in 2006
    - Use <u>both</u> CPU and GPU to increase performance
    - Boosted use of heterogeneous computing

Background - Artificial Intelligence (1)

Why is heterogeneous computing relevant to AI?

- Enhance performance, improve energy efficiency, and provide flexibility
- Leverages various types of processors such as:
    - CPUs for general tasks
    - GPUs for parallel processing
    - Specialized accelerators for specific AI functions
- Can handle intensive computational demands
- Helpful for complex and varied applications of AI:
    - Faster processing
    - reduces power consumption
    - supports scalability

Background - Artificial Intelligence (2)

Why is AI relevant to heterogeneous computing?

- Power of AI predictions allow for a decrease in time and monetary costs by using AI to:
    - Design and implement
    - Predict various metrics instead of simulating
- EdgeCortix developed Machine-learning Enhanced Runtime Acceleration (MERA) software and compiler framework:
    - Smart compiler - optimized code is generated for various type of processors [6]
    - Runtime - execution and adaptation of heterogeneous systems is dynamically managed [6]

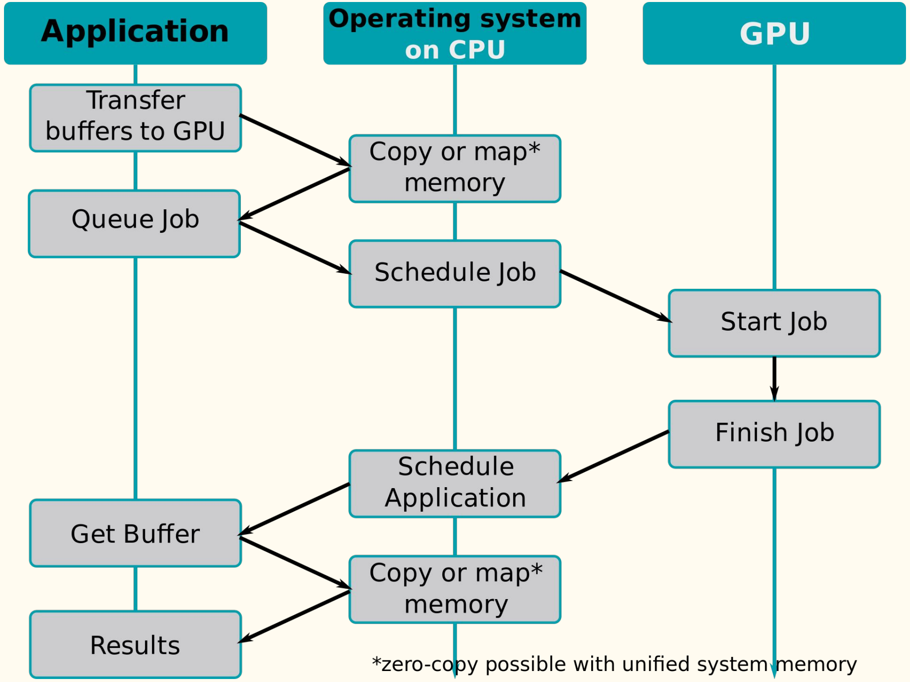
Infrastructure and Frameworks

- Key Architectures/Frameworks
  - **HSA (Heterogeneous System Architecture):** Enables efficient hardware acceleration, integrating CPUs, GPUs for seamless computation.
  - **CUDA (Compute Unified Device Architecture):** NVIDIA's platform for GPGPU, pivotal in AI and deep learning.
  - **OpenCL (Open Computing Language):** Open standard for cross-platform programming across CPUs, GPUs, and more, ensuring flexibility and broad applicability.
  - **TPU Architecture (Tensor Processing Unit):** Google's custom-designed processors optimized for TensorFlow operations, significantly accelerating deep learning computations.
  - **ROCm (Radeon Open Compute):** AMD's open source GPU computing framework.
  - **Intel oneAPI:** A unified programming model by Intel to streamline development across CPUs, GPUs, FPGAs, and AI accelerators.

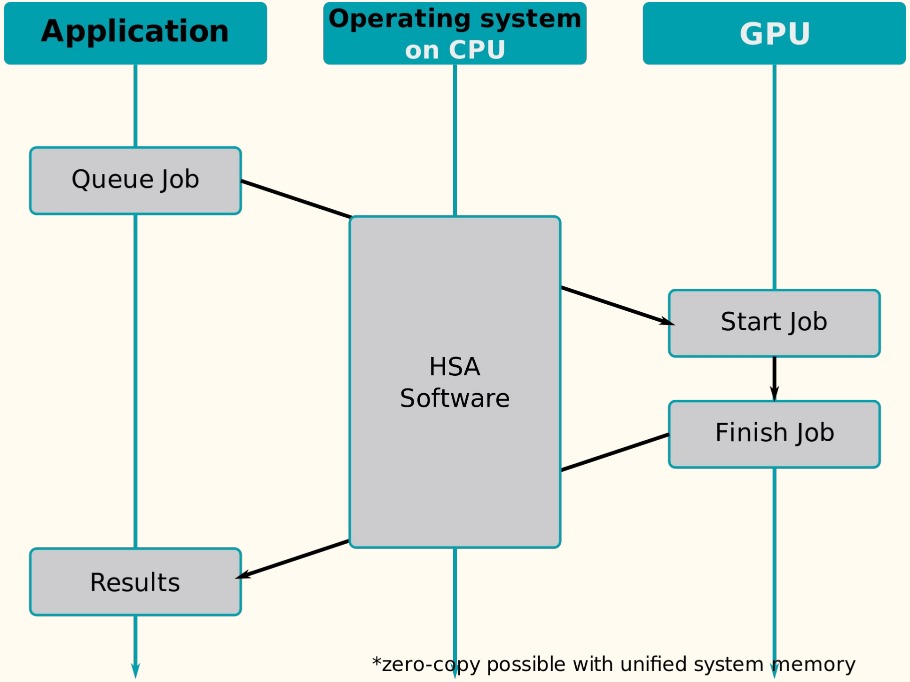
HSA (Heterogeneous System Architecture) (1)

- HSA integrates CPUs and GPUs for shared memory and tasks.
- Led by the HSA Foundation, which includes AMD and ARM.
- Reduces latency between compute devices and simplifies programming.
- Enhances execution performance of programming languages and models like CUDA and OpenCL.
- Enables direct GPU floating point calculations without separate scheduling.

HSA (Heterogeneous System Architecture) (2)



non-HSA system



HSA system

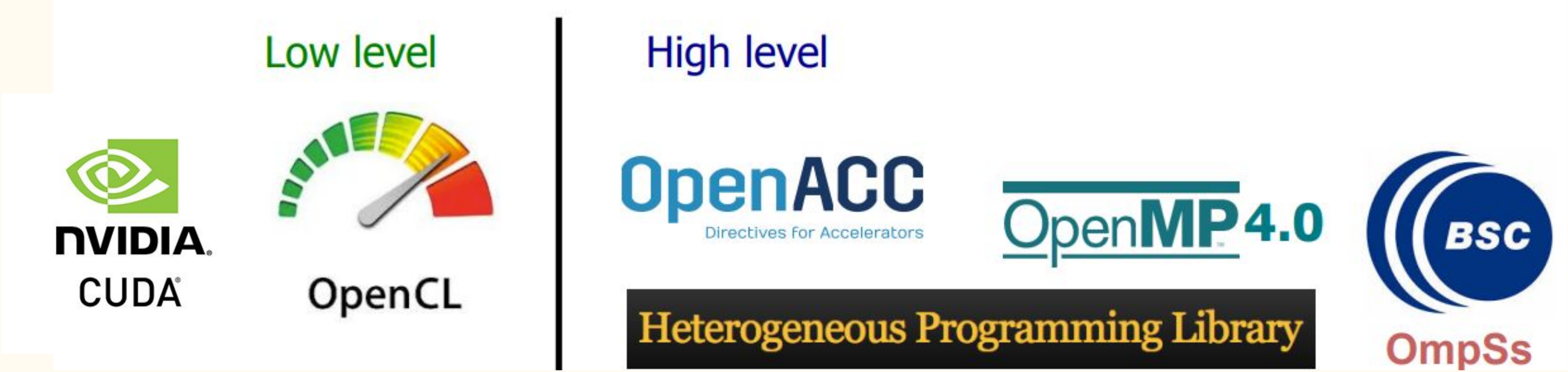
Programming models for Heterogeneous computing (1)

- CUDA
    - NVIDIA proprietary
- OpenCL
    - Open standard, functionally portable across multi-cores
- OpenACC
    - High-level, pragma-based
- Different libraries, programming models, and DSLs for different domains

Level of abstraction increases

Programming models for Heterogeneous computing (2)

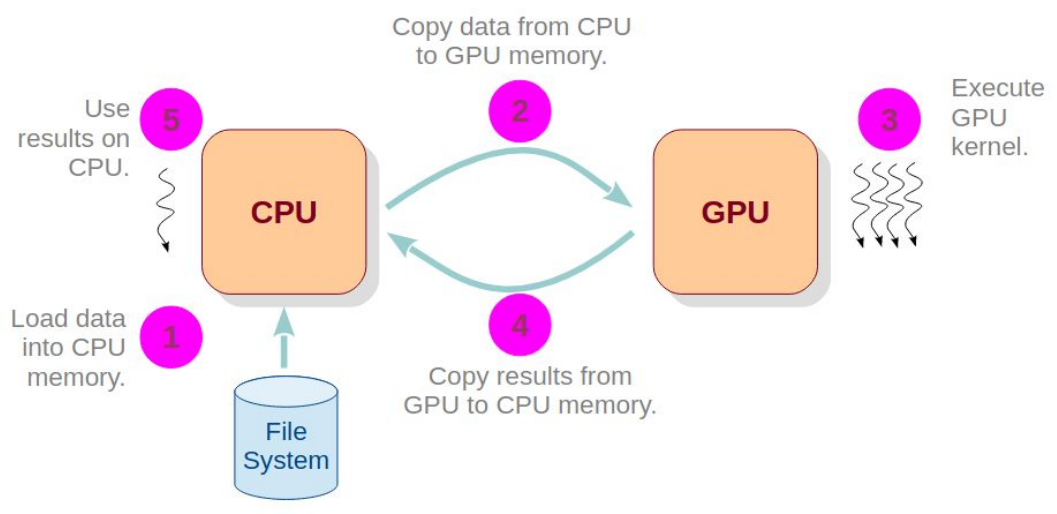
- Mix of programming models
  - One(/several?) for CPUs – OpenMP
  - One(/several?) for GPUs – CUDA
- Single programming model (unified)



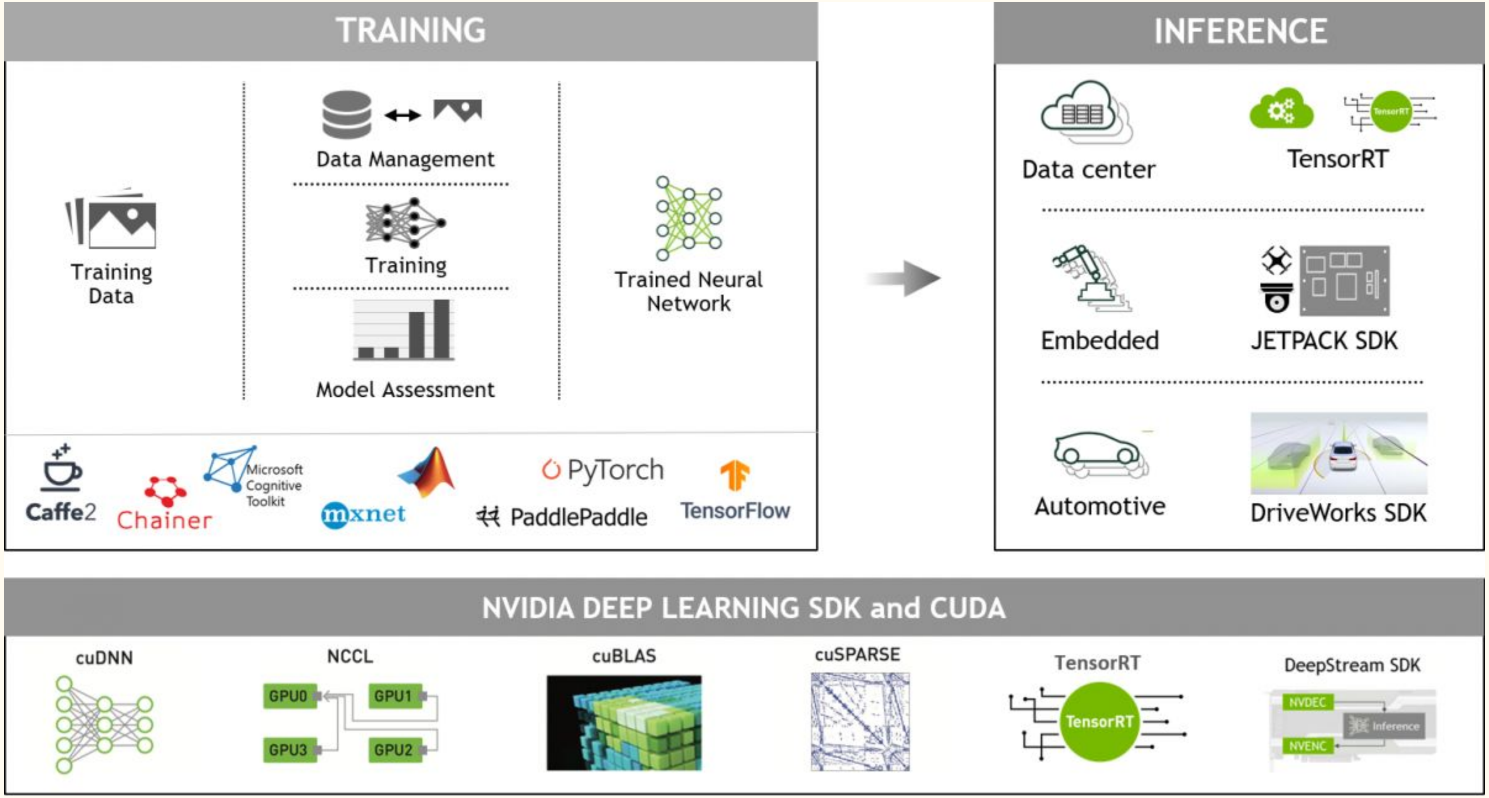
CUDA(Compute Unified Device Architecture) (1)

- NVIDIA's programming platform for GPGPU (General-Purpose computing on Graphics Processing Units).
- Designed for heterogeneous computing: some functions run on the CPU and others on the GPU. Programs typically written in C or C++ with annotations for GPU execution.

Typical CUDA program flow

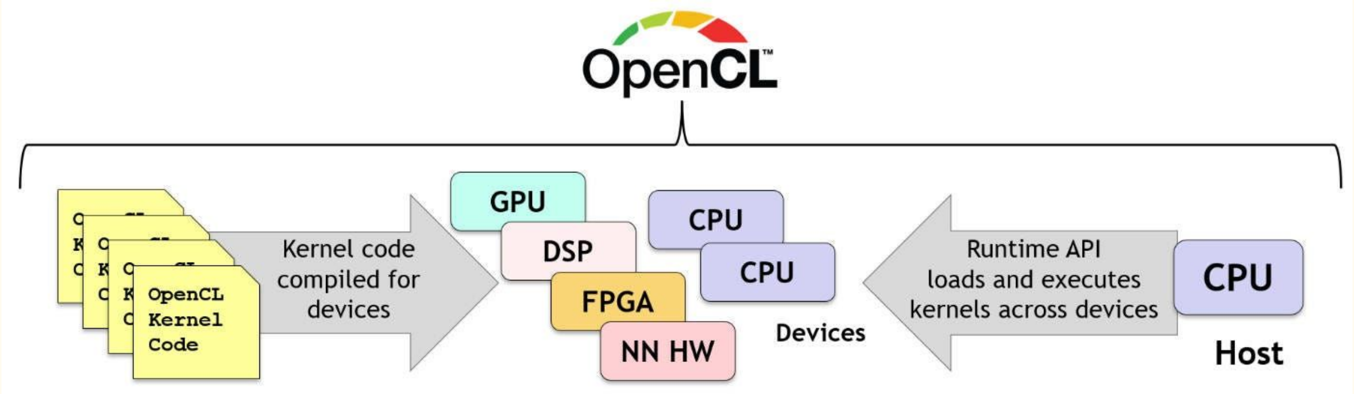


CUDA(Compute Unified Device Architecture) (2)

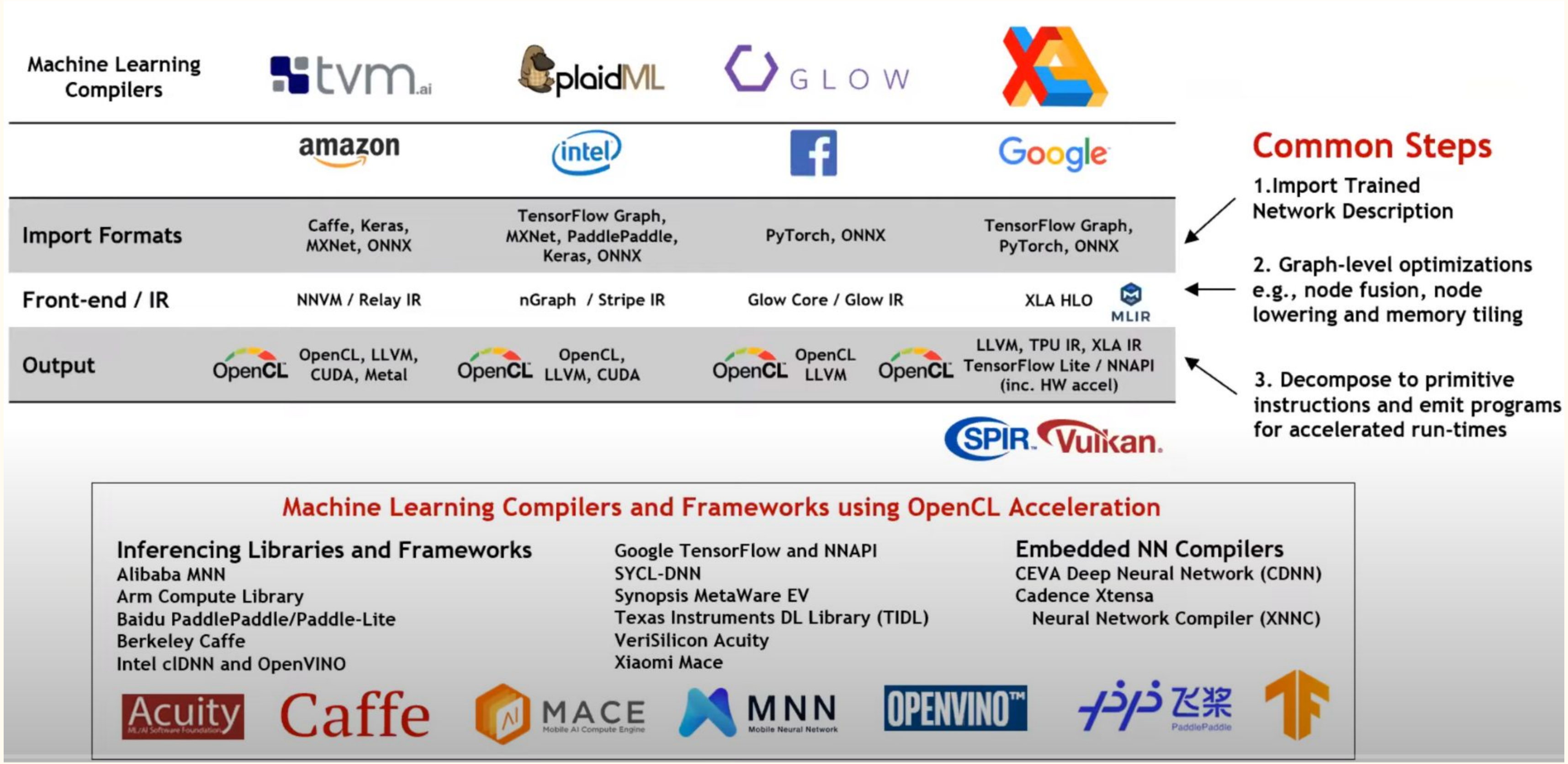


OpenCL (Open Computing Language) (1)

- Cross-Platform Framework: Enables execution of code across different hardware platforms.
- Supports Heterogeneous Systems: Works with various computing devices including CPUs, GPUs, DSPs, and FPGAs.
- Write Once, Run Anywhere: Programs written in OpenCL can run efficiently across multiple hardware architectures supports the OpenCL standard.



OpenCL (Open Computing Language) (2)

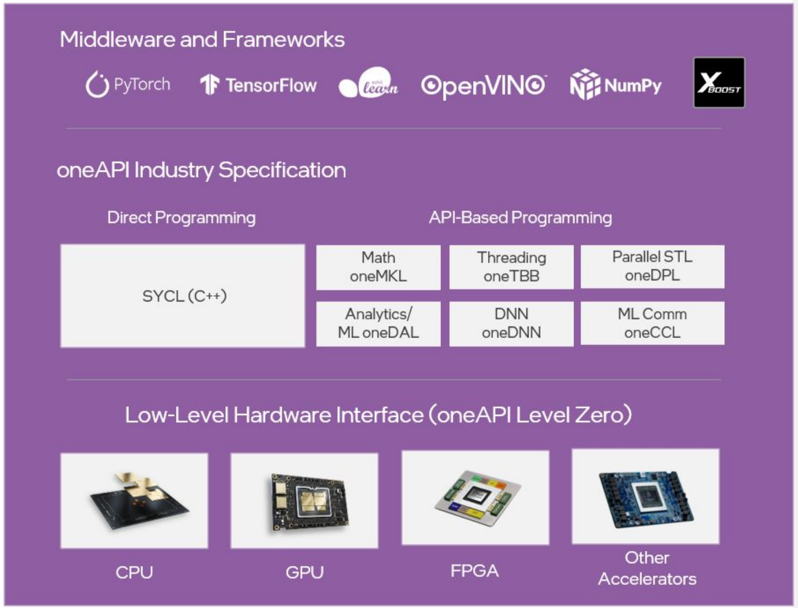


Intel oneAPI (1)

- A unified programming model designed to streamline development across Intel's diverse hardware (CPUs, GPUs, FPGAs, AI accelerators).
- Core Language - DPC++ : Based on ISO C++ and SYCL(a cross-platform abstraction layer that allows algorithms to switch between hardware accelerators—such as CPUs, GPUs, and FPGAs) standards, DPC++ is aimed at leveraging the parallel computing potential of Intel's hardware. `DPC++` = `C++` + `SYCL` + `community extensions`
- Workload Categorization: Classifies workloads into scalar, vector, matrix, and spatial domains to optimize computing across different Intel processors.

Intel oneAPI (2)

- Intel is pitching oneAPI as a viable CUDA alternative.
- There's a growing trend of migrating to oneAPI for its promise of eliminating vendor lock-in and fostering a diverse computing environment.
- Performance Gains: Adopters of oneAPI often report enhanced performance compared to previous implementations.



Performance improvement

- Martínez et al. [12] explored the potential of Intel's oneAPI as a unified programming model for machine learning applications, with a focus on revamping the Caffe framework.
- They performed a comparative performance study for two crucial layers, softmax and convolution, between the original Caffe implementation (with CUDA) and oneAPI implementation on both CPU and GPU platforms.
- Softmax Layer: The oneAPI version showed a 7x speedup on Intel GPUs compared to the CPU baseline and a 1.75x improvement over NVIDIA A100 GPUs.
- Convolution Layer: While GPU comparisons were limited due to compatibility issues(with NVIDIA GPUs), the oneAPI implementation on Intel CPUs outperformed native Caffe implementations for larger datasets.

DL load distribution for Heterogeneous computing

- Let's take the example of Convolutional Neural Network(CNN), a popular image recognition/classification deep learning model.
- CPU:
  - Data Management: Preprocesses input data (decoding, normalization, augmentation).
  - Orchestration: Manages training loop (batch prep, epoch management).
  - Parameter Updates: Updates model weights post-backpropagation.
- GPU:
  - Layer Processing: Speeds up computation of CNN layers (convolutions, activations).
  - Backpropagation: Efficiently computes gradients, enhancing learning speed.
- FPGA:
  - Optimized Operations: Executes custom, efficient CNN operations.
  - Inference: Supports real-time model deployment with low latency.
- TPU:
  - Matrix Operations: Specializes in large-scale matrix computations.
  - Efficient Training/Inference: Boosts both training and inference speeds.

Heterogeneous Computing for Deep Learning (1)

Malita *et al.* [3] explores challenges and advancements in hardware acceleration for deep learning:

- Computational components of Deep Neural Networks (DNNs)
  - Fully connected layers, convolutional layers, pooling layers, softmax layers
  - Significant computational intensity
  - Efficient acceleration for optimal performance very important for DL
- State of the art hardware solutions
  - Reviewed various state-of-the-art hardware solutions - discuss various architecture
    - Intel's Many Integrated Core (MIC) Processors
    - NVIDIA's Graphics Processing units (GPUs)
    - Google's Tensor Processing Units (TPUs)
  - Discussed: performance characteristics, energy efficiency, limitations

Heterogeneous Computing for Deep Learning (2)

Malita *et al.* [3] explores challenges and advancements in hardware acceleration for deep learning:

- Limitations of Specific ASICs like TPU
  - TPUs offer significant computational power → lack flexibility needed for DL
  - Issues regarding:
    - Flexibility, resource utilization, memory hierarchy, architectural suitability
  - Understand limitations to develop strategies to mitigate challenges
- Conclusion and Future Directions
  - Need for innovative architectural designs able to
    - Adapt to evolving landscape of deep learning
    - While optimizing for performance and energy efficiency

AI for Heterogeneous Computing (1)

- Heterogeneous computing is a necessity for AI
- Can AI in turn be used to optimize heterogeneous computing?
  - Implementation: AI for Compiler
    - As seen previously, EdgeCortix's MERA
  - Predict quantifiers: Power, energy use, performance, etc.
    - Useful for understanding system impacts
    - Making design decisions
  - Designing: Finding optimal configuration
    - AI can find a optimal configuration faster or with less energy use

AI for Heterogeneous Computing (2)

Approach 1: Memeti *et al.* [1] use of AI to optimize use of system.

Enumeration and Measurements (ENuM)

vs.

Memeti *et al.* [1] method AI Heuristics with Machine Learning (AML)

ENuM

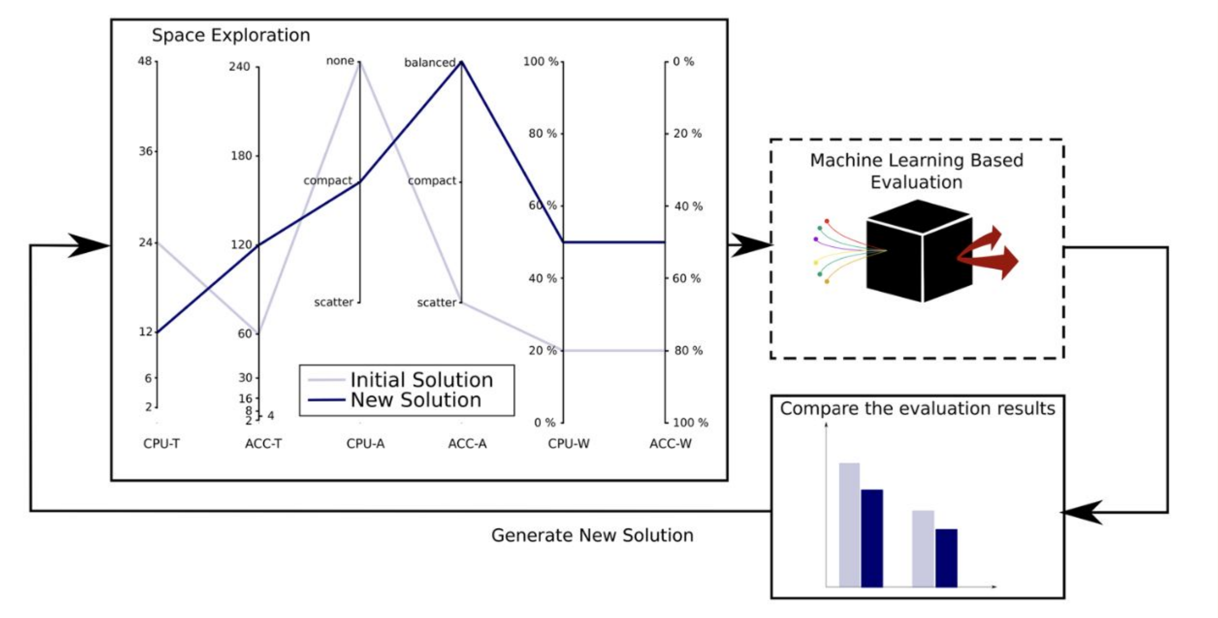
- Brute-force search
- Evaluates every possible option
- Makes a decision based on those options

AML

- Aims to find near optimal system configuration
- To optimize use of heterogeneous system:
    - AI heuristic search in combination with machine learning

AI for Heterogeneous Computing (3)

Approach 1: Memeti *et al.* [1] use of AI to optimize use of system.



- Parameter space
  - Heuristic search as guide
  - Simulated annealing to conduct exploration
- Decision tree regression
  - Supervised Machine Learning model
  - Evaluate system configuration

Results: AML is 1300 times faster than ENuM and achieves similar energy efficiency after only evaluating around 7% of possible configurations.

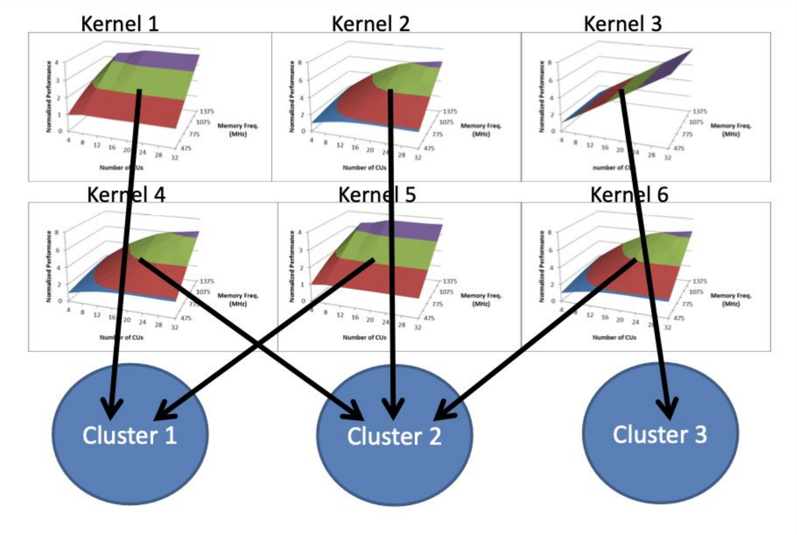
AI for Heterogeneous Computing (4)

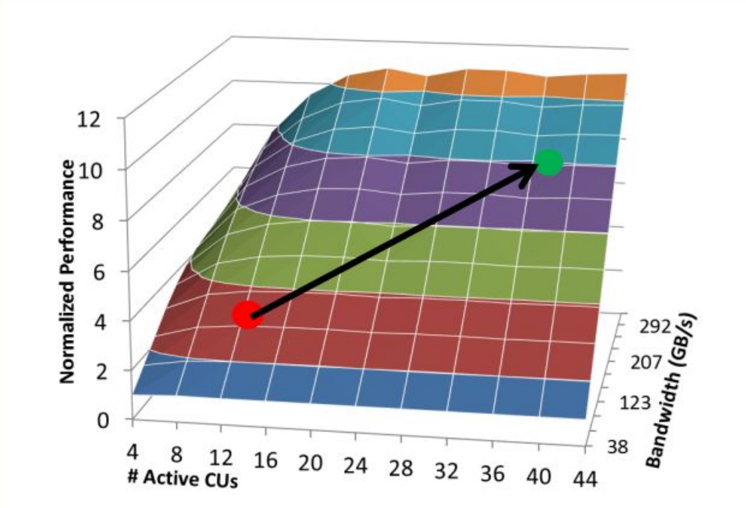
Approach 2: Greathouse *et al.* [2] use AI to predict performance or power.

- Predict an applications performance or power on various heterogeneous systems.
- Designed to lessen cost of simulation for industry professionals
  - When picking system configuration
- Created dataset
  - Running various applications on different combinations of hardware
  - Storing the data
- Fully connected neural network
  - Linear input layer
  - Sigmoid functions for hidden and output layers

AI for Heterogeneous Computing (5)

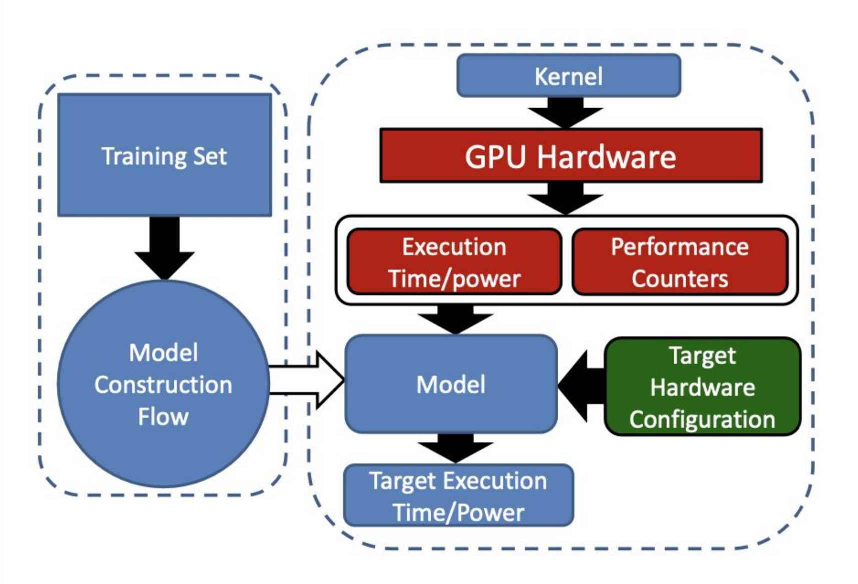
- Kernels are trained and clustered
  - Grouped with similar kernels
  - Kernel with highest value is chosen as cluster representative
  - Within each kernel is a scaling curve





AI for Heterogeneous Computing (6)

- Model is ready to use
  - Application is inserted
  - Cluster is picked based on closest match to application.
  - Scaling curve predicts performance or power of kernel
    - Based on desired configuration
- Results: Use of system is very effective
  - Does not require heavy computations
  - Performance: Model is 85% accurate
  - Power: Model is 90% accurate



Future of Heterogeneous Computing

- Challenges for wider adoption
  - Programming Complexity: writing for various processor types
  - Memory Management: everything is sharing a memory, needs to be effectively managed
  - Task Scheduling: scheduling is done across all processors, needs to be efficient
- Potential for further accelerating AI research and applications
  - Tons of potential with the continued development of AI
    - As AI pushes further it will require more and more from heterogeneous systems
  - Heterogeneous computing being used more push for development
    - Also increase capabilities of heterogeneous computing for AI research and applications

Conclusion

- Heterogeneous Computing
  - Supports the high computational demands of AI processes
  - Allows for scalability
  - Reduces power consumption
- Artificial Intelligence
  - Allows for reduction in time and monetary costs
  - Using AI for design, implementation, prediction of quantifiers
- Infrastructure and Frameworks
  - A lot of organizations are working in the space of developing Heterogeneous Computing ecosystem because of it's promising future.

References

1. Memeti, S., Pllana, S. Optimization of heterogeneous systems with AI planning heuristics and machine learning: a performance and energy aware approach. Computing 103, 2943-2966 (2021). https://doi.org/10.1007/s00607-021-01017-6

2. J. L. Greathouse and G. H. Loh, "Machine Learning for Performance and Power Modeling of Heterogeneous Systems," 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Diego, CA, USA, 2018, pp. 1-6, doi: 10.1145/3240765.3243484.

3. Mihaela Maliţa, George Vlăduţ Popescu, & Ştefan, G. M. (2019). Heterogeneous Computing System for Deep Learning. Studies in Computational Intelligence, 287-319. https://doi.org/10.1007/978-3-030-31756-0_10

4. *CUDA FAQ*. NVIDIA Developer. (n.d.). https://developer.nvidia.com/cuda-faq

5. *CUDA in Action* - Research & Apps. NVIDIA Developer. (n.d.). https://developer.nvidia.com/cuda-action-research-apps

6. Dasgupta, S. (2024, January 2). *AI drives the software-defined heterogeneous computing era*. Energy-Efficient AI Processors and Acceleration. https://www.edgecortix.com/en/blog/ai-drives-the-software-defined-heterogeneous-computing-era#:~:text=Heterogeneous%20Computing%20in%20an%20AI%20Context&text=Heterogeneity%20can%20involve%20different%20instruction,%2C%20cost%2C%20and%20greater%20flexibility

References

7. *Specifications | oneAPI*. (n.d.). OneAPI.io. https://www.oneapi.io/spec/
8. *Accelerating Machine Learning with OpenCL*. (2022, May 11). The Khronos Group. https://www.khronos.org/events/accelerating-machine-learning-with-opencl
9. Liu, X., Ounifi, H.-A., Gherbi, A., Li, W., & Cheriet, M. (2019). A hybrid GPU-FPGA based design methodology for enhancing machine learning applications performance. *Journal of Ambient Intelligence and Humanized Computing*, *11*(6), 2309–2323. https://doi.org/10.1007/s12652-019-01357-4
10. *Heterogeneous System Architecture*. (2023, December 29). Wikipedia. https://en.wikipedia.org/wiki/Heterogeneous_System_Architecture#
11. *Deep Learning Software*. (n.d.). NVIDIA Developer. https://developer.nvidia.com/deep-learning-software
12. Martínez, P. A., Peccerillo, B., Bartolini, S., García, J. M., & Bernabé, G. (2022). Applying Intel's oneAPI to a machine learning case study. *Concurrency and Computation: Practice and Experience*, *34*(13). https://doi.org/10.1002/cpe.6917

Thank you!

Questions?