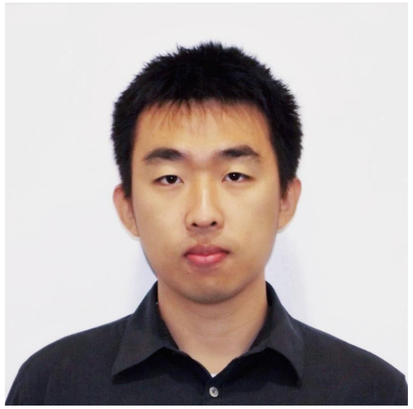
Tutorial 2: AI Kernel Programming

Andrew Adams
Adobe Research

Tri Dao
Princeton, Together AI

Sharad Vikram
Google

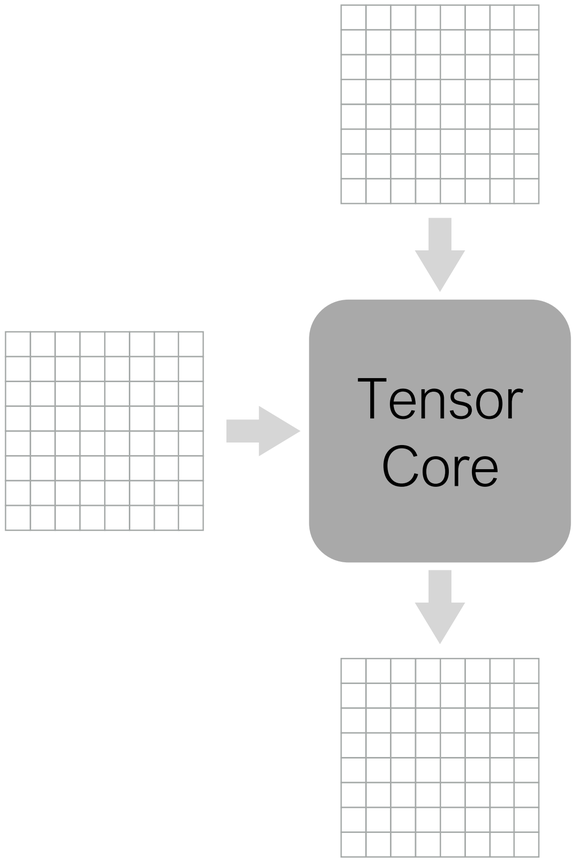


Zhihao Jia
CMU

Fred Kjolstad, Stanford University

Modern AI Hardware

Matrix Multiply



Vector Processors

Vector Core

Vector Core

Vector Core

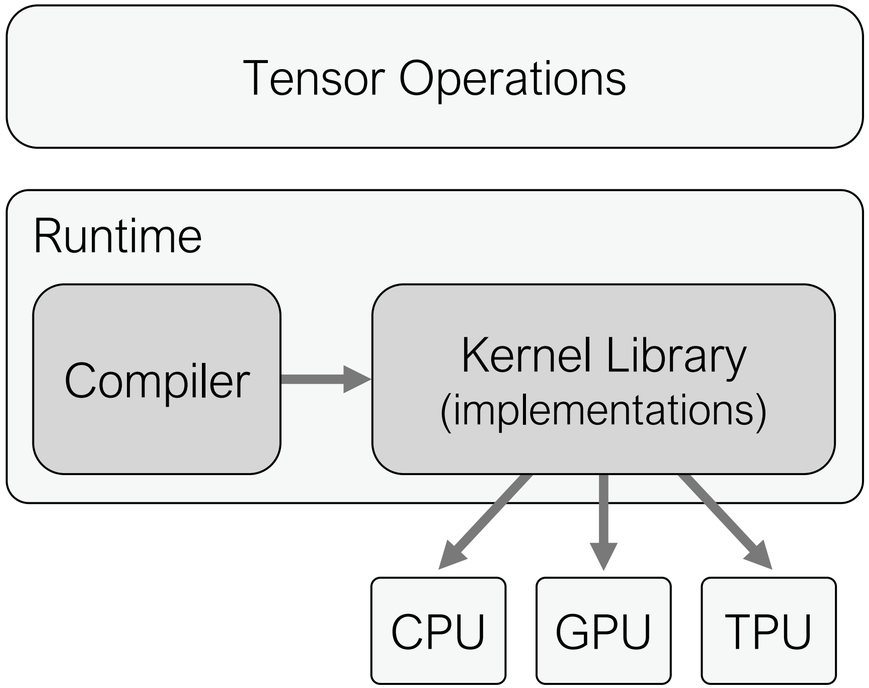
Vector Core

Data Movement HW

Tensor Memory Accelerator

Memory

ML Frameworks and Libraries



Tensor Frameworks

PyTorch

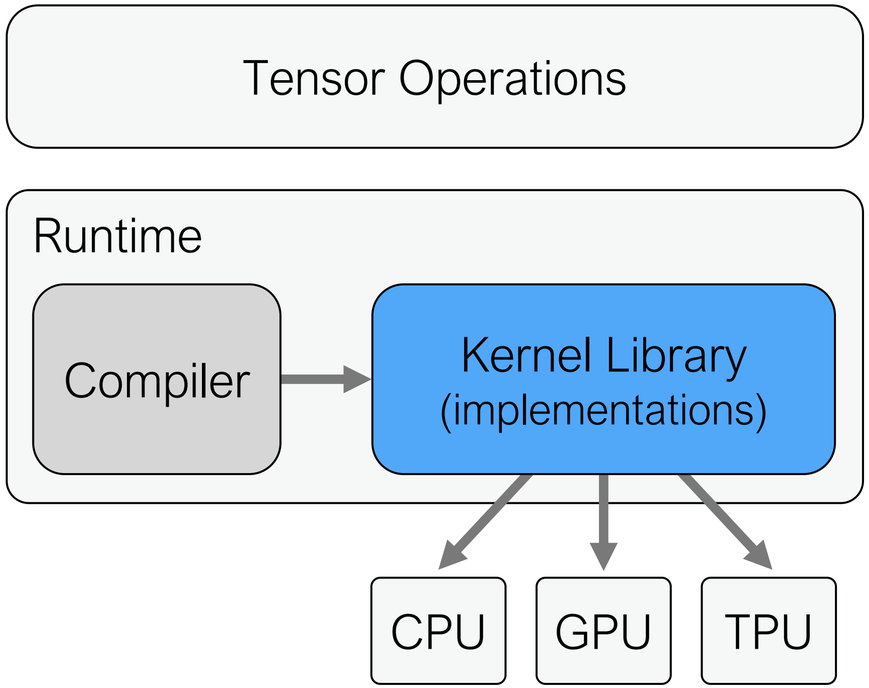
Other Frameworks

DEEP GRAPH LIBRARY

PyG

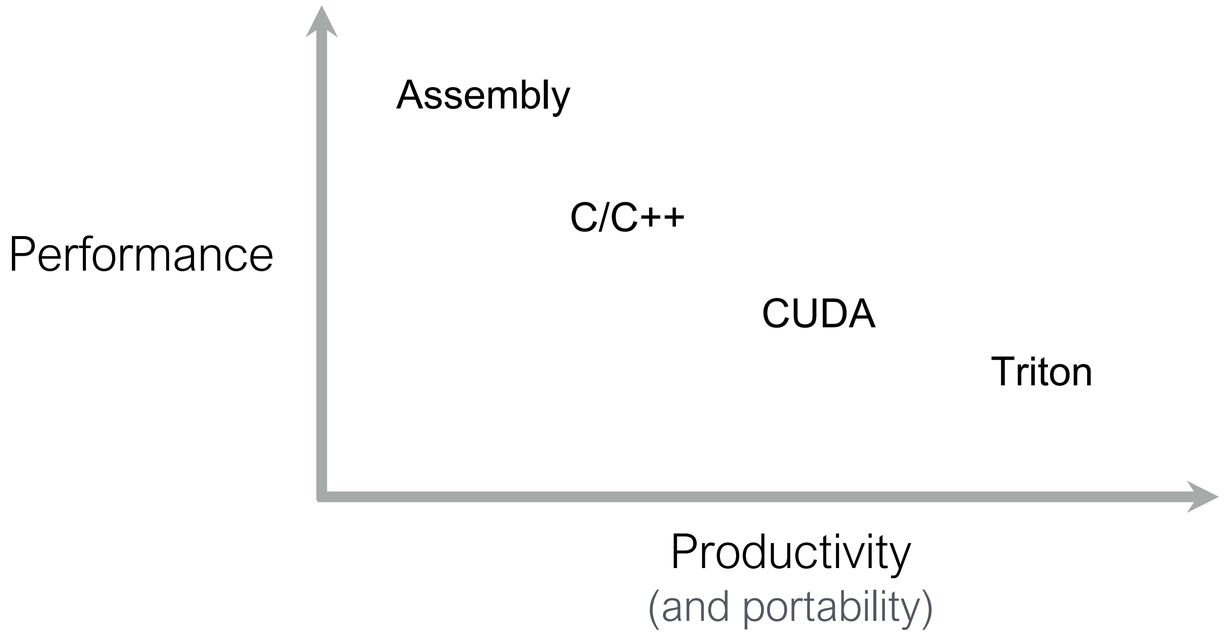
Vector Databases

Role of Hand-Written Kernels



- A good performance engineer get great performance
- Get started before you have a compiler
- But expensive and changing hardware requires rewrite

Kernel Languages



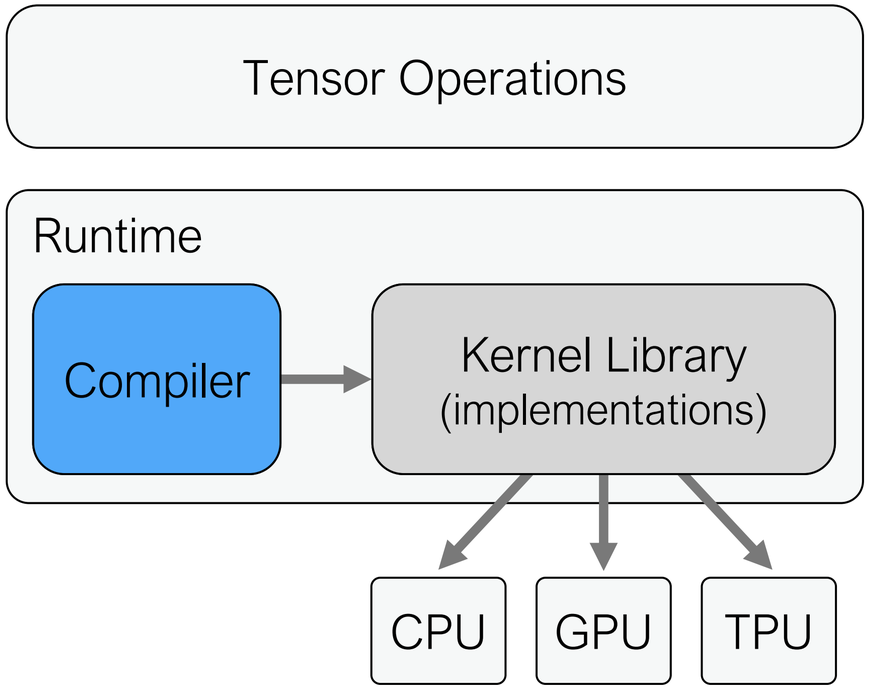
(other: Pallas, NKI, HIP, and SYCL)

Key Ideas

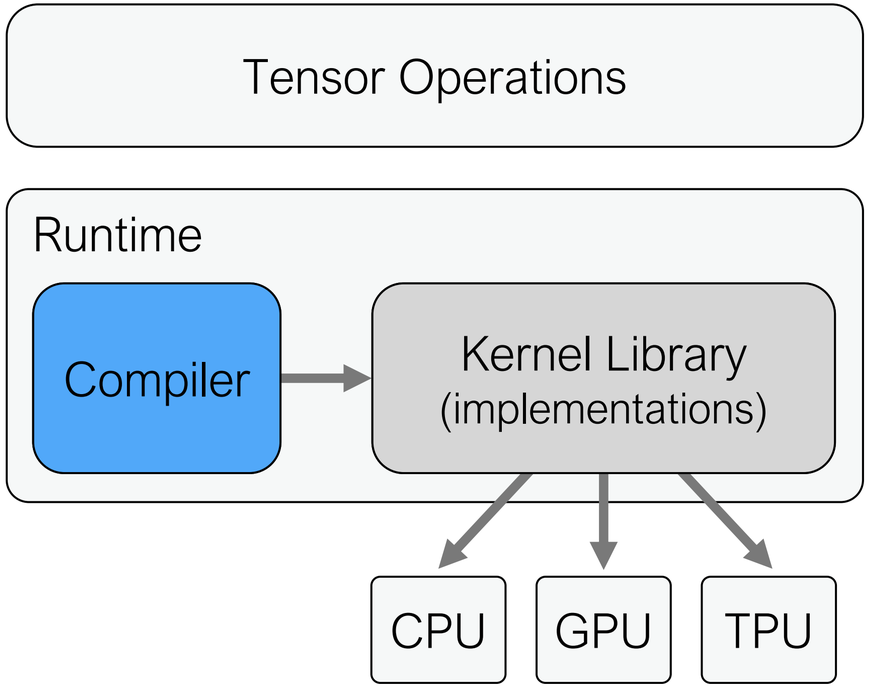
- Stream kernels for structured parallelism
- Tiles instead of scalars
- Scheduling languages
- Data structure polymorphism

Machine-Oriented instead of Collection-Oriented

Role of Compilers



Role of Compilers



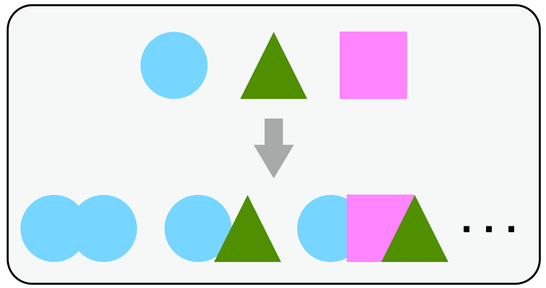
Productivity

Too many kernels to write by hand

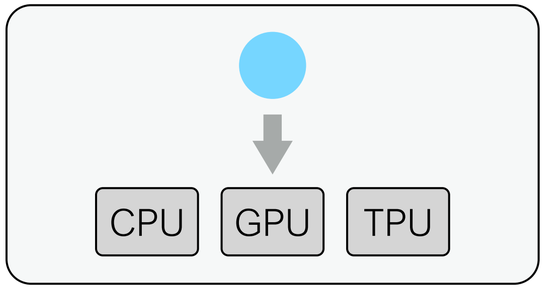
Library Size

compiler size < library size

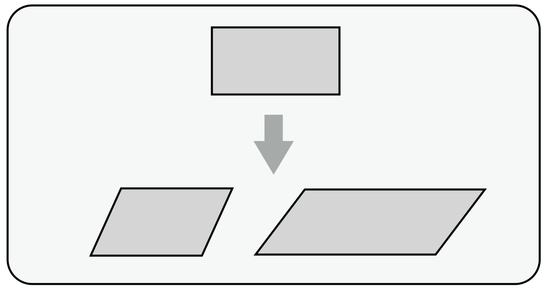
Fusion



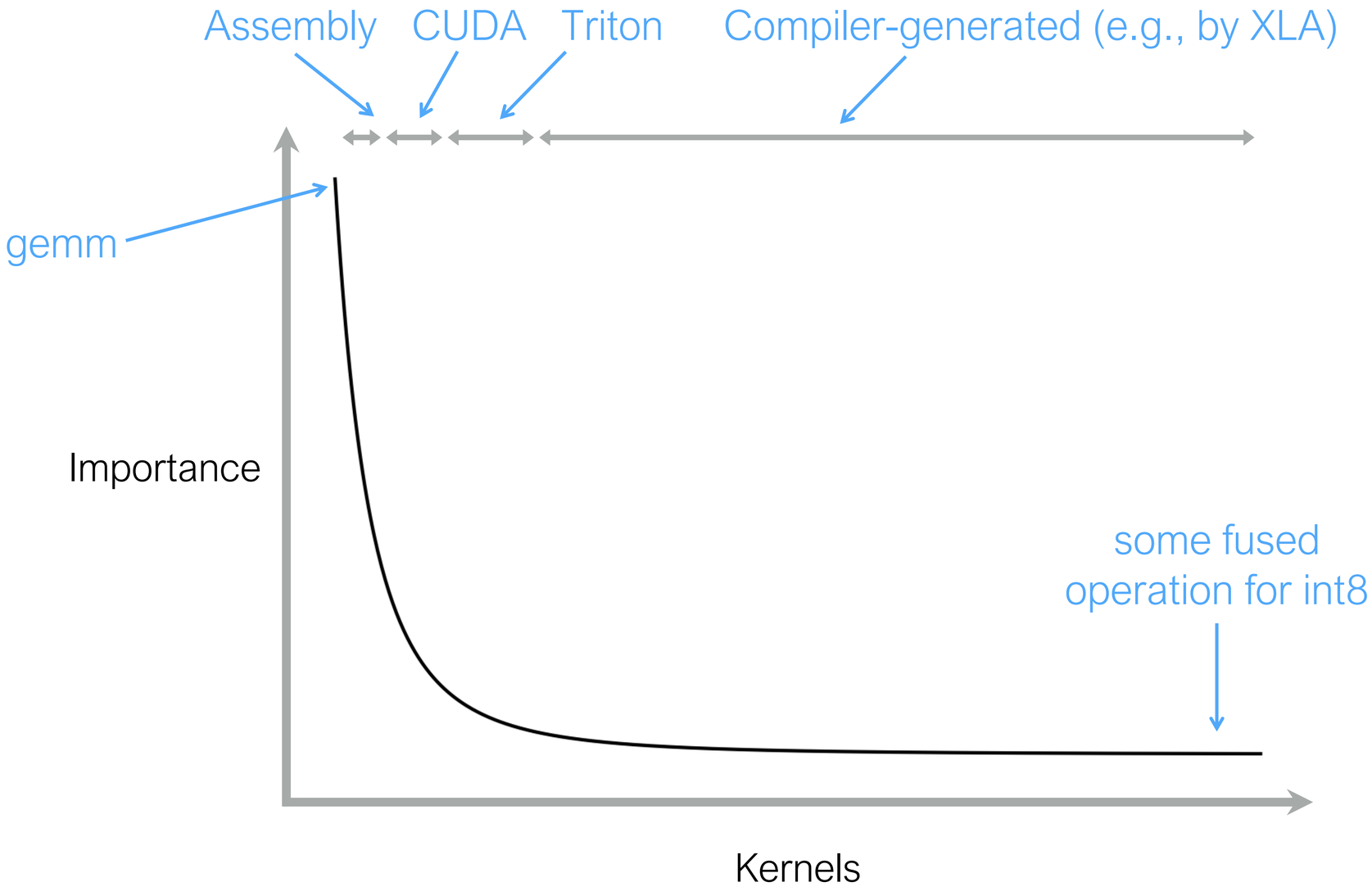
Portability



Pipelining



Performance–Productivity Tradeoff



Tutorial Schedule

1:30 – 1:45	Introduction	Fredrik Kjolstad Stanford	
1:45 – 2:30	Decoupling Performance from Correctness with User-Schedulable Kernel Languages	Andrew Adams Adobe Research	
2:30 – 3:30	Pallas: Using JAX to write custom kernels for GPUs and TPUs	Sharad Vikram Google	
3:30 – 4:00	**Coffee Break**		
4:00 – 4:45	Domain specific languages for GPU kernels and automatic kernel authoring with LLMs	Tri Dao Princeton, Together AI	
4:45 – 5:30	Programming techniques for implementing ML models on GPUs	Zihao Jia CMU