UC San Diego

CSE 291: Operating Systems in Datacenters

Amy Ousterhout

Nov. 14, 2023

UC San Diego

Agenda for Today

- Reminders
- GPUs and TPUs overview
- TensorFlow discussion

Reminders

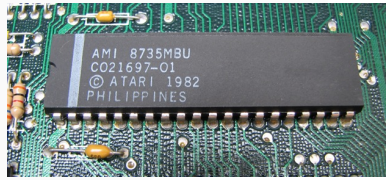
- Project check-ins this week
  - Sign up on Canvas
  - Be prepared to talk about your progress so far:
    - What have you learned?
    - What are you struggling with?
    - You can use the whiteboard or show diagrams or graphs
    - No need for a formal presentation
- No office hours this week

UC San Diego

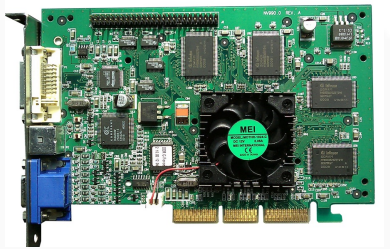
GPUs

History of GPUs

- Originally designed to create images to display
  - 1970s: video processors for arcade games
  - 1980s: graphics processors for PCs
  - 1990s: 3D graphics
    - 1999: “the world’s first GPU”
  - 2000s: more programmability
- Applied to general purpose compute tasks
  - GPGPUs
  - Linear algebra (2003)
  - Scientific computing
  - Mining bitcoin (today)



Atari ANTIC microprocessor

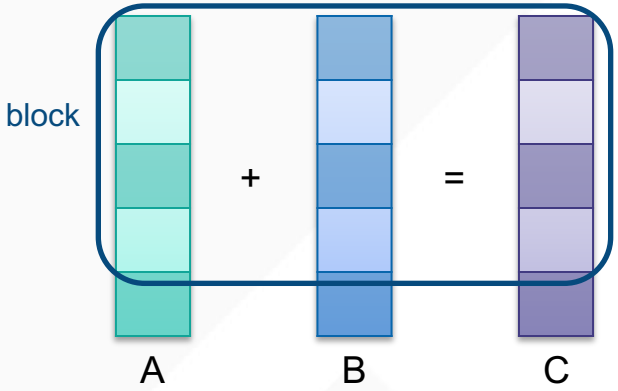


Nvidia GeForce 256

https://commons.wikimedia.org/wiki/File:ANTIC_chip_on_an_Atari_130XE_motherboard.jpg
https://commons.wikimedia.org/wiki/File:RTX_3090_Founders_Edition!.jpg

Data Parallelism

- GPUs are designed for data-parallel tasks
- Example: add two arrays/vectors



Sequential (e.g., on a CPU):

```
void sequential_add(int n, float *A, float *B, float *C) {
    for (int i = 0; i < n; i++)
        C[i] = A[i] + B[i];
}
```

Parallel (e.g., on a GPU):

```
void parallel_add(int n, float *A, float *B, float *C) {
    int i = thread_index;
    if (i < n)
        C[i] = A[i] + B[i];
}
```

Systems Research on GPUs

- How should we program GPUS?
  - CUDA, OpenCL, etc.
- How can we process packets on GPUs?
  - PacketShader, SSLShader
- How can we schedule and manage memory on GPUs?
  - TimeGraph, PTask, TensorFlow
- How can we share GPUs across multiple apps?
- How can use use GPUs to accelerate ML workloads?
  - TensorFlow

UC San Diego

TPUs

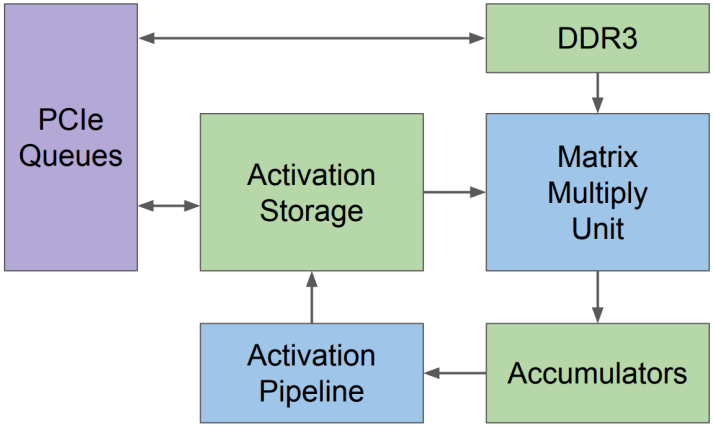
The Creation of TPUs

- 2013: machine learning was consuming more and more CPU cycles
  - Especially Deep Neural Networks (DNNs)
  - Very expensive
- Google set out to create a custom chip
  - Domain-Specific Architecture (DSA)
- Created the Tensor Processing Unit (TPU)
  - Used internally starting in 2015
  - Announced publicly in 2016
  - Codesigned with TensorFlow

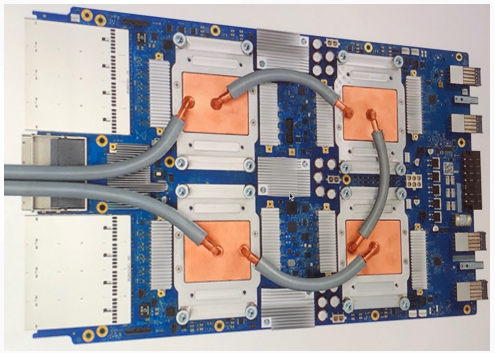
Year	Version	Training?	Inference?
2015	TPUv1		✓
2017	TPUv2	✓	✓
2018	TPUv3	✓	✓
2020	TPUv4i		✓
2021	TPUv4	✓	✓

What is a TPU?

- Coprocessor connected via PCIe
- Primarily matrix multiplication and activations
- Optimized for 99th % performance
  - No caches, context switching, out-of-order execution, etc.
- Lower precision than CPUs
  - E.g., 8-bit multiplication
- Used for: improving search results, AlphaGo, etc.
- 30-80x better performance/watt than CPUs and GPUs (TPUv1)



TPUv1



https://www.hc32.hotchips.org/assets/program/conference/day2/HotChips2020_ML_Training_Google_Norrie_Patil.v01.pdf
https://upload.wikimedia.org/wikipedia/commons/b/be/Tensor_Processing_Unit_3.0.jpg

Research on TPUs

- How to use TPUs for large-scale machine learning?
  - TensorFlow, OSDI 2016
- How to improve TPU performance?
  - “In-Datacenter Performance Analysis of a Tensor Processing Unit”, ISCA 2017
  - “Ten Lessons From Three Generations Shaped Google’s TPUv4i”, ISCA 2021
  - “TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings”, ISCA ‘23

UC San Diego

TensorFlow Discussion