High Performance LLMs From First Principles (2024)

Session 2
https://github.com/rwitten/HighPerfLLMs2024

rwitten@

Goal: learn how to achieve **high performance** for LLMs

This week: understand single chip performance

Program (write code in Jax)

Predict (roofline on napkin or spreadsheet)

Profile (run code, compare to predictions)

My Asks

Please ask lots of questions! Just raise your hand or speak up!

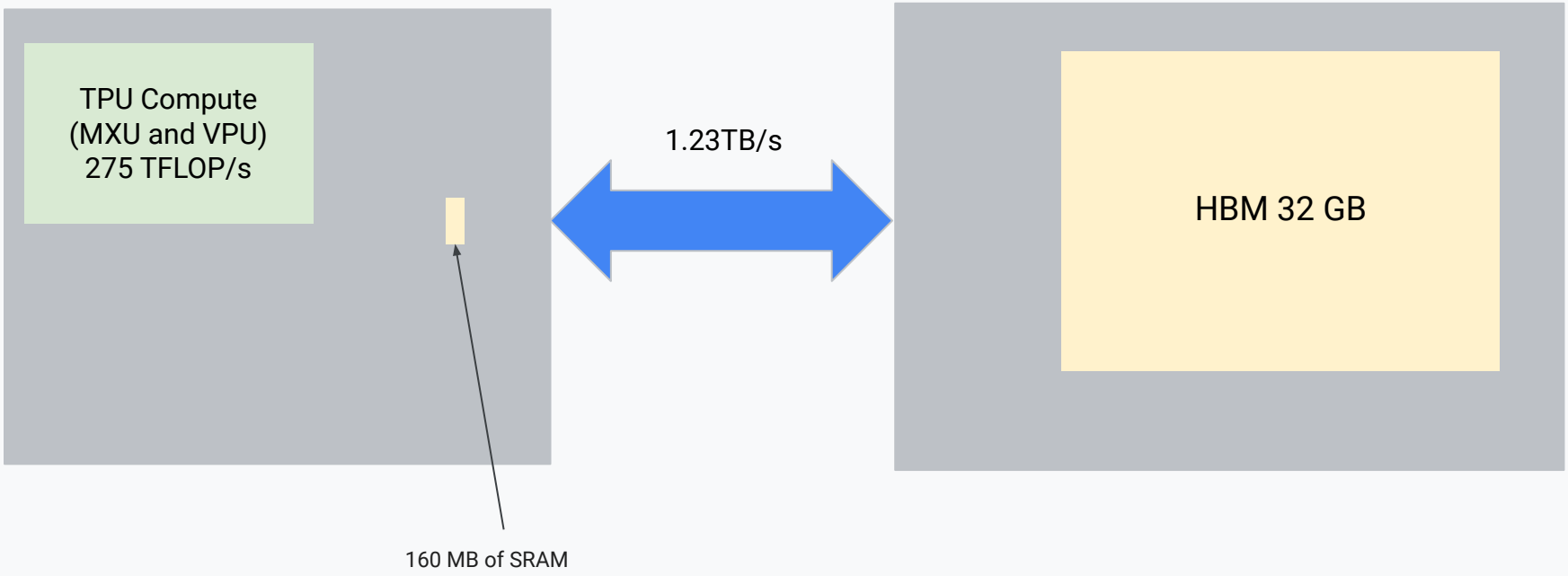
If there are topics you're interested in, message me between sessions.

Join the discord! https://discord.gg/2AWcVatVAw

Do the exercises! Give feedback, ask questions!

Website: https://github.com/rwitten/HighPerfLLMs2024

System Diagram For TPUs (v4)



Let’s Time An Addition!

- A + B
- Code Exercise And How Long Should It Take?

Let’s Time Two Additions!

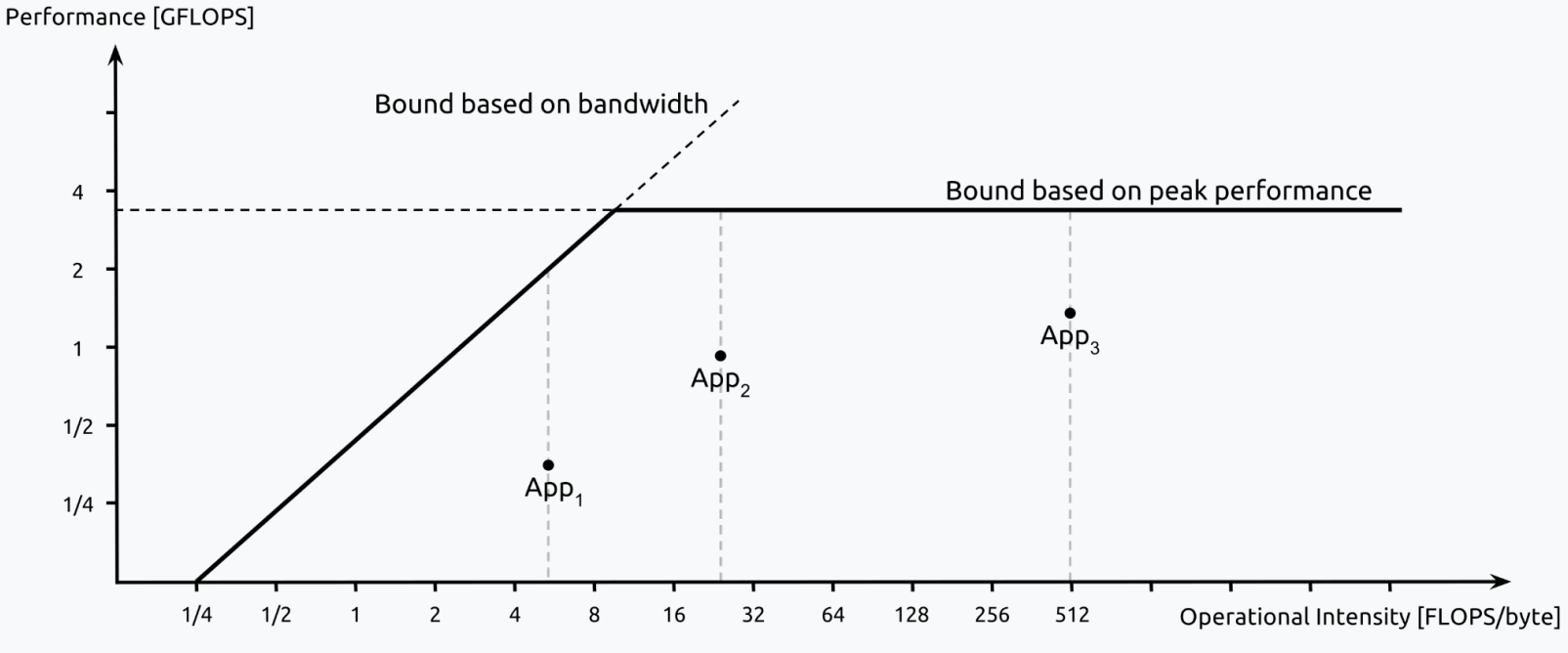
- A + B + C
- Code Exercise And How Long Should It Take?

Operator Fusion!

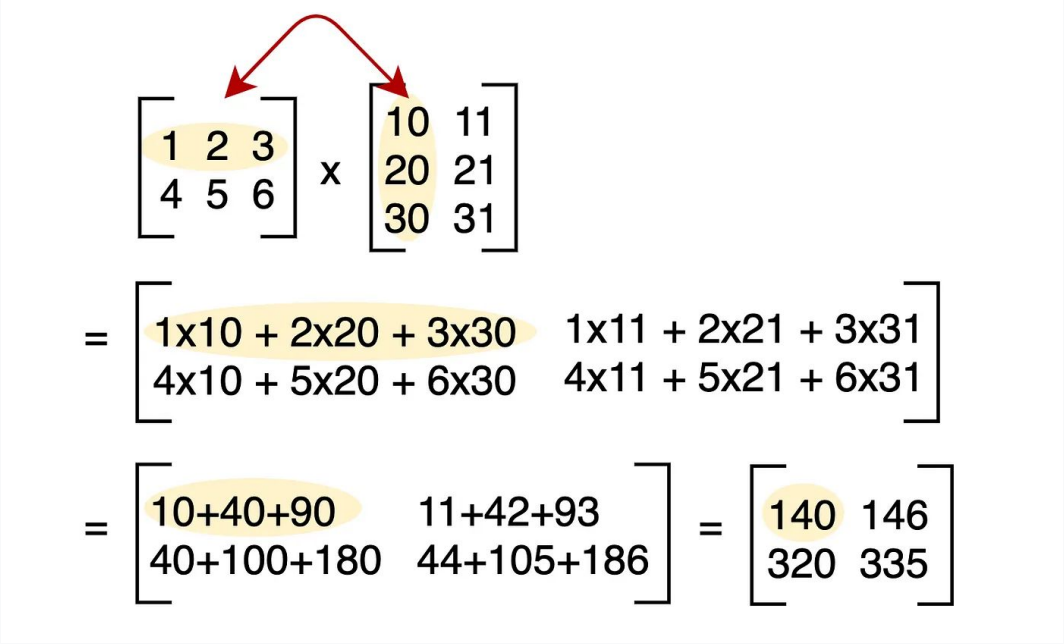
- This example is pretty simple. But as graphs get more complicated, choosing the right fusions gets subtle.
- Normal compilers (GCC, Clang, NVCC) miss a lot of good opportunities for optimization. In the first example, **add** was compiled first – a normal compiler can't fix this!
- Until ~2015, the main idea was "write bigger well optimized functions"! People would hand code and optimize bigger and bigger building blocks (see LAPACK / BLAS) until they hit diminishing returns.
- With GPUs/TPUs (and more compute per memory bandwidth) the necessary size blocks got bigger!
- PyTorch (and early TensorFlow) just did operator-by-operator dispatch. (Two calls to **add** in this case.)
- Compilers do many more optimizations: fissions, overlapping, rematerialization

Key Observations: Arithmetic Intensity & Roofline

- Compare hardware FLOPs/byte to the program's FLOPs/byte.



Matrix Multiply (matmul) Reminder



- https://towardsdatascience.com/a-complete-beginners-guide-to-matrix-multiplication-for-data-science-with-python-numpy-9274ecfc1dc6

Matmul Reminder

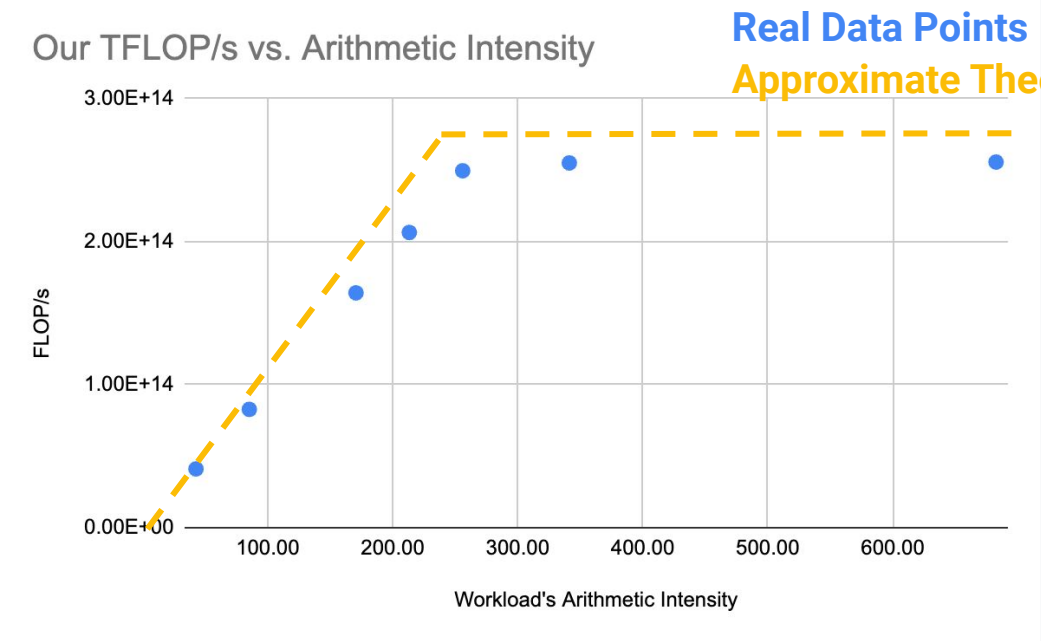
- Input A and B
  - A is (x,y)
  - B is (y,z)
- FLOPs = 2*x*y*z
- Input is 2*(x*y + y*z) bytes, assuming matmul is in bfloat16
- Output is 2*(x*z) bytes, assuming matmul is in bfloat16

Show Fusion After Matmul

- relu(A@B)

Roofline of Matmuls

- time(A@B) as dimension changes vs. roofline?



Heads up: no class next week (3/6)!
Back 3/13!

Thanks!
Ping me (rwitten@google.com) with feedback, suggested topics, etc!