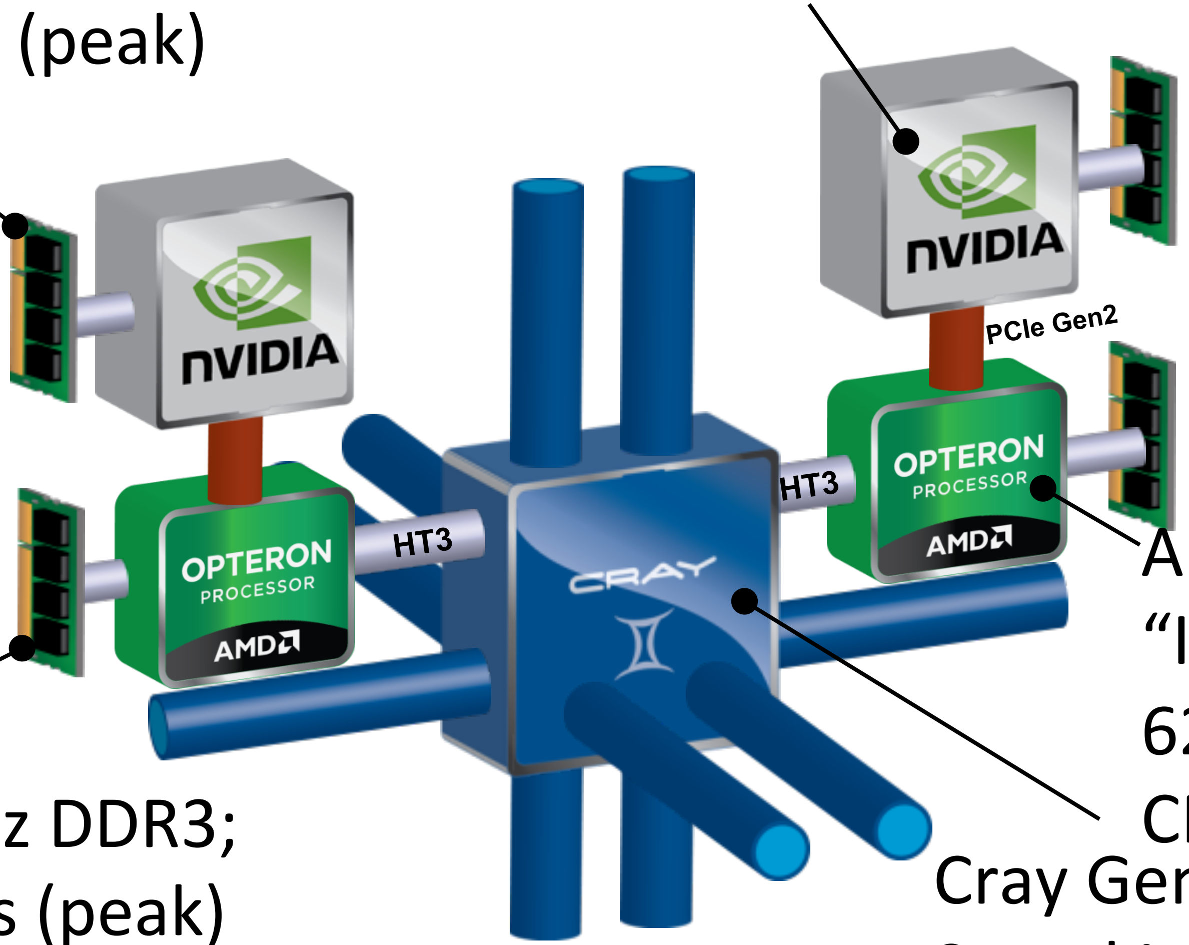
Cray XK7 Architecture

Cray XK7 Architecture

NVIDIA Kepler GPU

6GB GDDR5;
250 GB/s (peak)

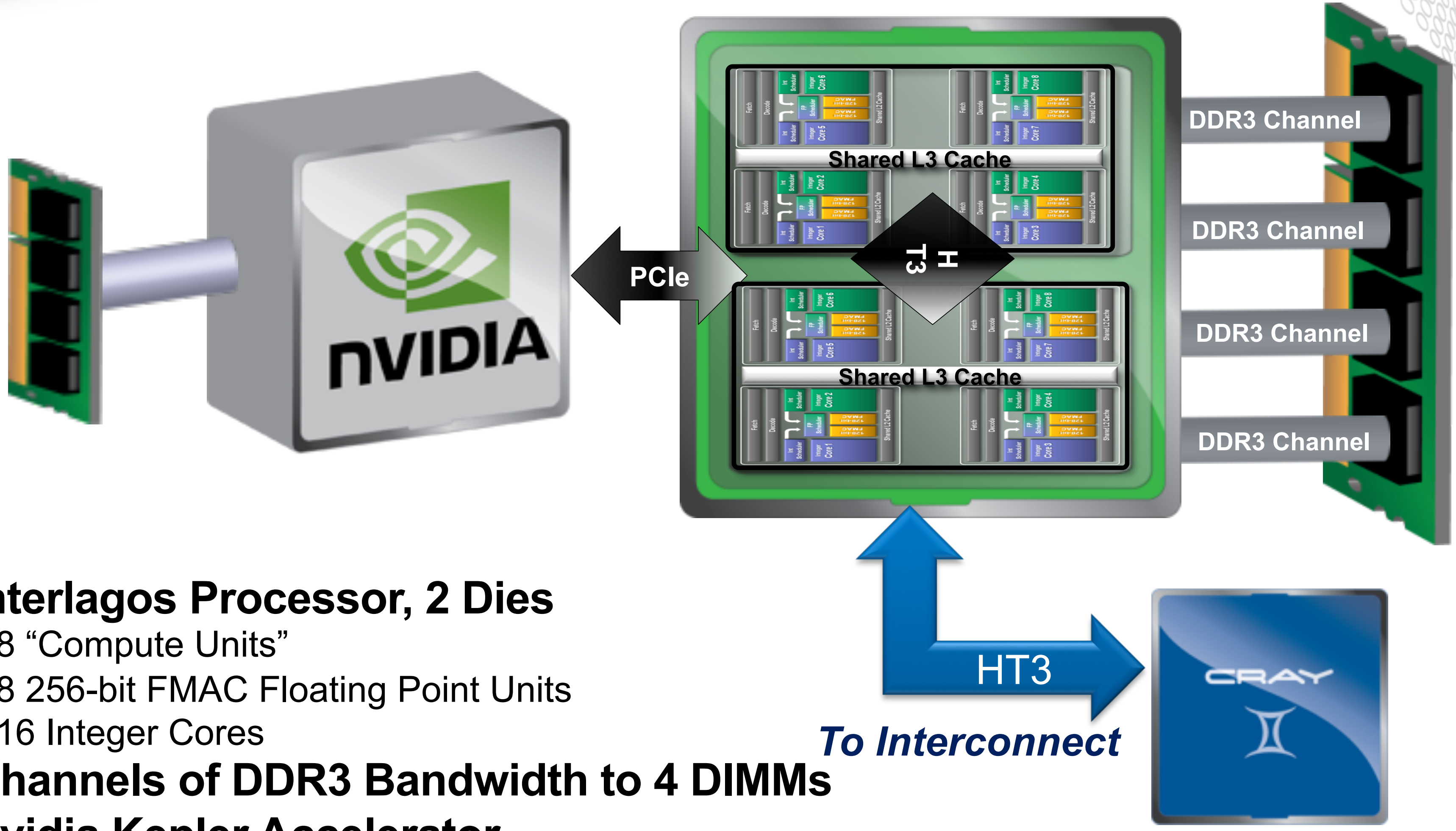


AMD
“Interlagos”
6200 Series
CPU

1600 MHz DDR3;
51.2 GB/s (peak)

Cray Gemini High
Speed Interconnect

XK7 Node Details



- 1 Interlagos Processor, 2 Dies
  - 8 “Compute Units”
  - 8 256-bit FMAC Floating Point Units
  - 16 Integer Cores
- 4 Channels of DDR3 Bandwidth to 4 DIMMs
- 1 Nvidia Kepler Accelerator
  - Connected via PCIe Gen 2

AMD Interlagos Single vs. Dual-Stream

- **Dual-stream mode allows for 16 threads of execution per CPU**
  - 16 MPI ranks
  - 16 OpenMP threads
  - Some combination between
- **Two threads share a 256-bit FPU**
  - Single FP scheduler determines how best to share
- **This is aprun's default behavior on most systems.**

- **Single-stream mode places 1 thread of execution per compute unit (maximum 8)**
  - 8 MPI ranks
  - 8 OpenMP threads
  - Some combination between
- **Each thread fully owns a 256-bit FPU**
  - AVX256 instructions required
- **This mode has same peak FP and memory performance**
  - 2X FLOPS & Bandwidth per thread
- **This can be enabled in aprun with –j1 flag**

AMD Interlagos Single vs. Dual-Stream

- **Dual-stream mode allows for 16 threads of execution per CPU**
  - 16 MPI ranks
  - 16 OpenMP threads
  - Some c
- **Two threads sh FPU**
  - Single F how best to sha
- **This is aprun default behavior on most systems.**

- **Single-stream mode places 1 thread of execution per m unit (maximum 8)**
  - reads
  - tion between
- **lly owns a**
  - 56 instructions required
- **This mode has same peak FP and memory performance**
  - 2X FLOPS & Bandwidth per thread
- **This can be enabled in aprun with –j2 flag**

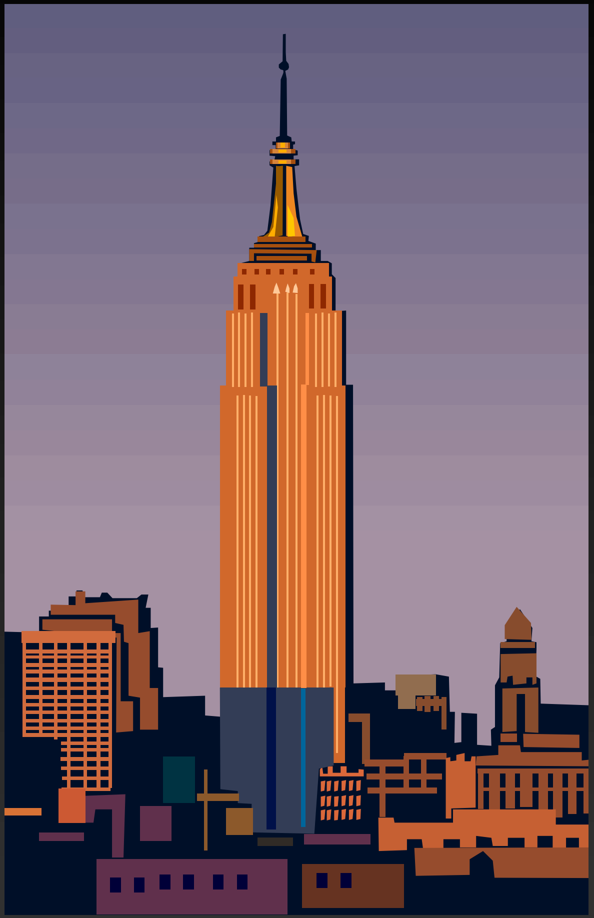
You have to experiment for yourself.

How to Think Like a GPU

You've been hired to paint a building

You’ve been hired to paint a building

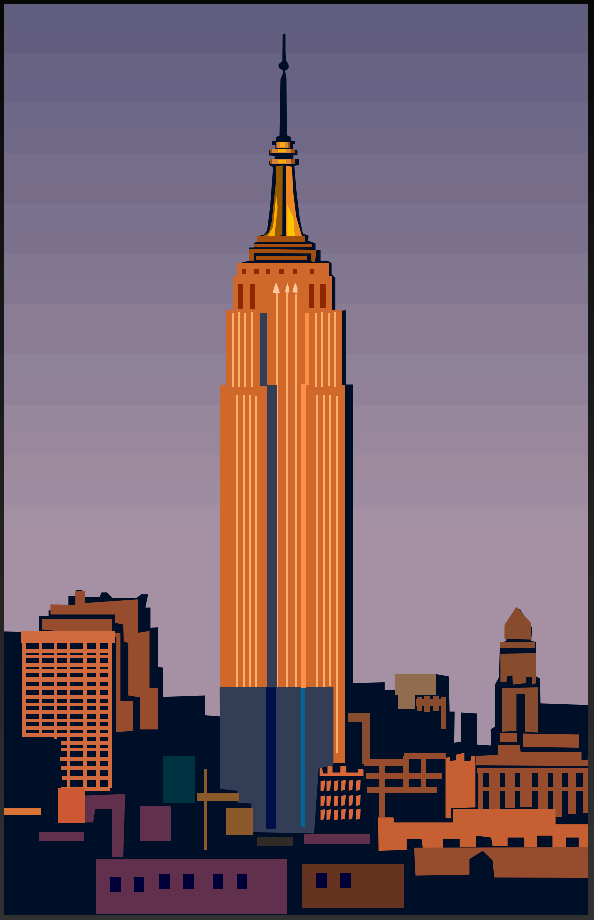
(A Big Building)



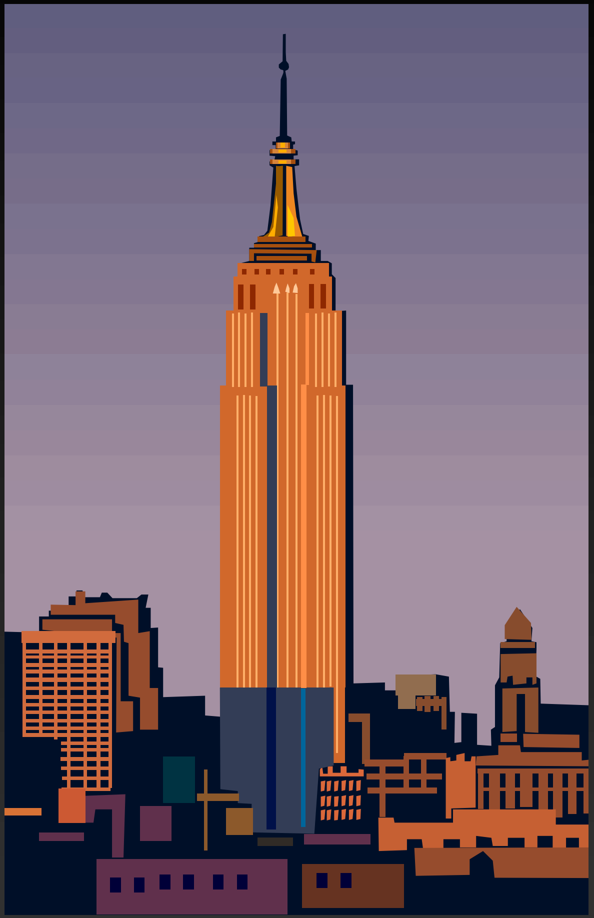
How can 1 painter paint faster?

1. Paint faster
   - One person’s arm can only move so fast
2. Paint wider
   - If you can use more rollers at once, you can cover more area, but there’s a limit to how many you can hold
3. Minimize trips to paint bucket
   - A paint tray can be kept close by, but it can only realistically be so big

In order to paint it quickly, you keep your roller and paint close by and roll as quickly as possible



But, there's a limit to how quickly you can roll and how much paint you can keep near by.



So you hire some help.

A well-organized team can paint nearly 4X faster.

What if, instead of buying more paint cans and wider rollers, you hire even more painters?

Now each painter is slower, but…

If we have enough painters, there will always be someone painting, so this won’t matter.

Thread Performance vs. Throughput

- CPUs optimize for maximum performance from each thread.
  - Fast clocks
  - Big caches

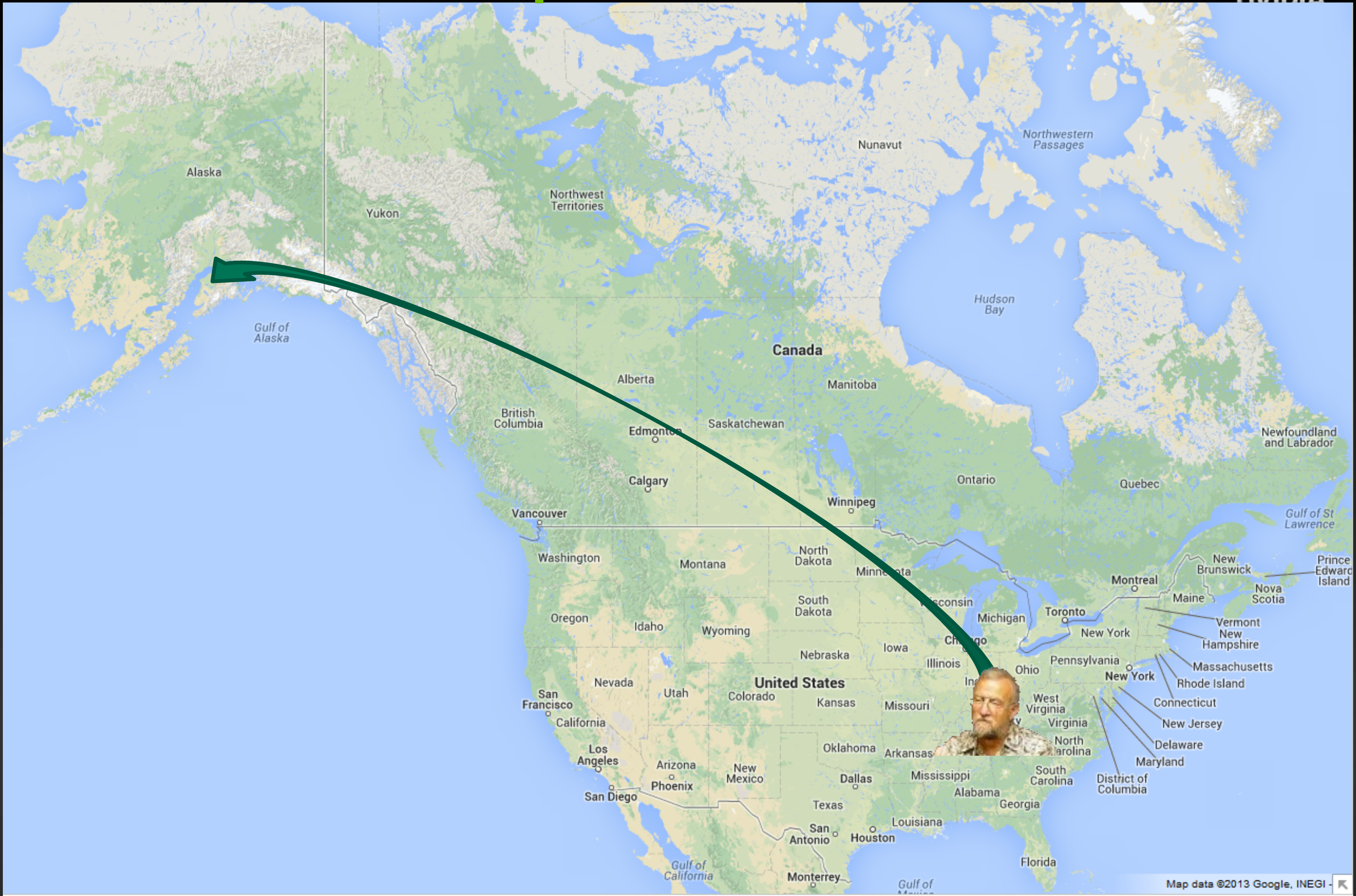
- GPUs optimize for maximum throughput.
  - Slower threads and smaller caches
  - Lots of threads active at once.

Glossary

You'll hear these terms throughout the rest of this talk, so let's relate them to the example.

Example	CUDA	OpenACC
Painter	Thread	Worker
Group of Painters	Thread Block	Gang
Number of rollers	Warp (always 32)	Vector Length
Total area to paint	Grid	Number of Gangs

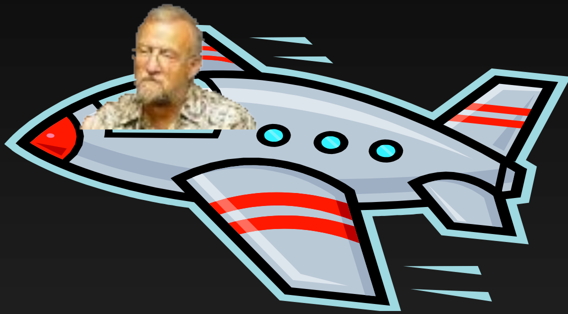
Another Example



Latency vs. Throughput

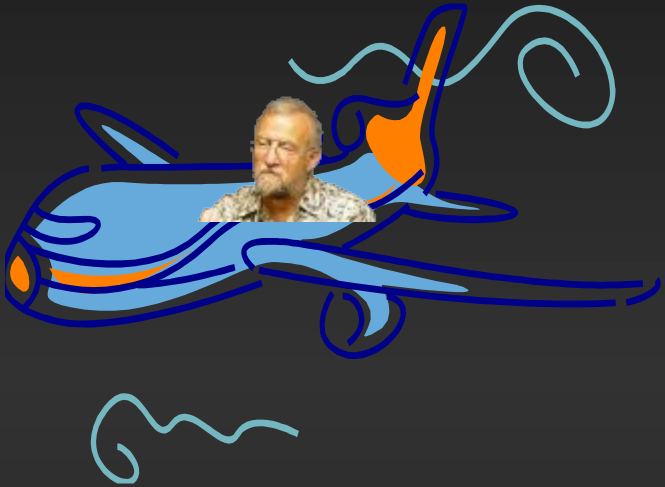
F-22 Rapter

- 1500 mph
- Knoxville to Anchorage Alaska in 2:10
- Seats 1



Boeing 737

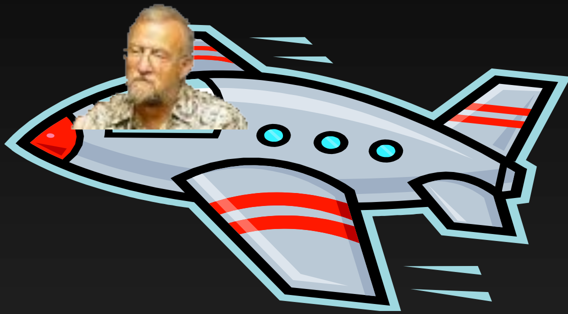
- 485 mph
- Knoxville to Anchorage Alaska in 6:45
- Seats 200



Latency vs. Throughput

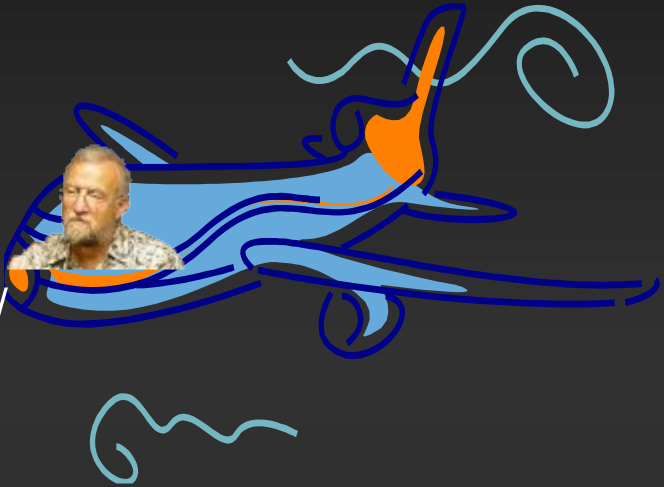
F-22 Rapter

- Latency (Time to transport 1 person) – 2:10
- Throughput – 1 / 2.16 hours = 0.46 people/ hr.
- Time to transport 200 people – 92 hours



Boeing 737

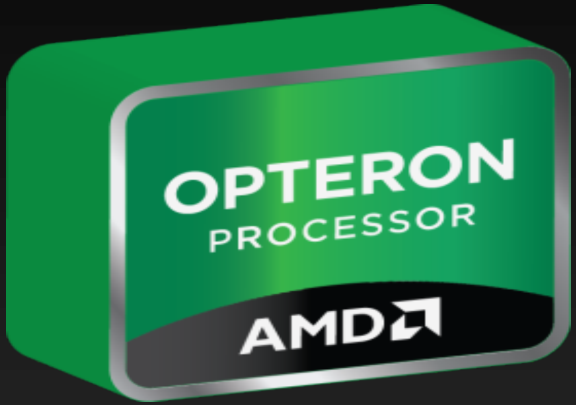
- Latency (Time to transport 1 person) – 6:45
- Throughput – 200 / 6.75 hours = 29.6 people/ hr.
- Time to transport 200 people – 6:45



Latency vs. Throughput

AMD Opteron

- Optimized for low latency
- For when time to complete an individual operation matters



NVIDIA Kepler

- Optimized for high throughput
- For when time to complete an operation on a lot of data matters