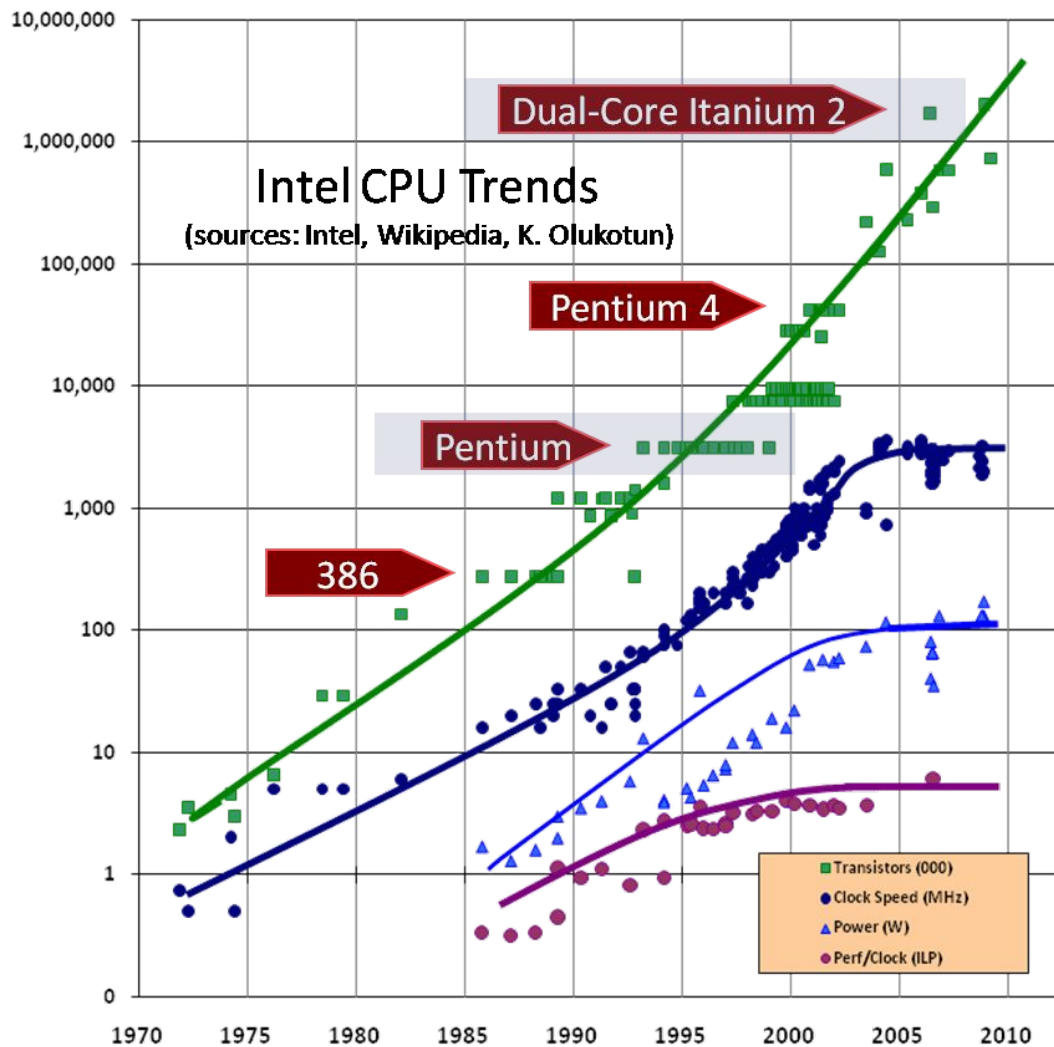


Parallel Programming with OpenCL for Fun and Profit

Dr Gordon Inggs
Amazon Web Services EC2

Why heterogeneous computing?

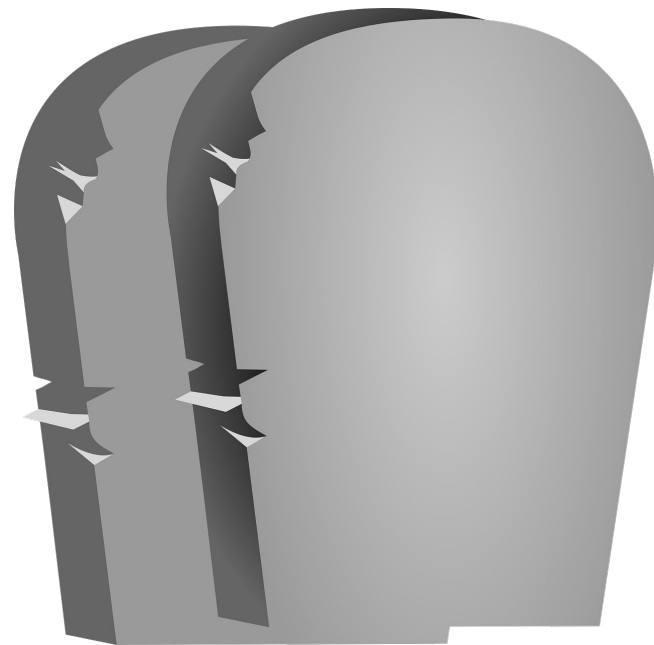
Why computers are becoming stranger
(and why that's a good thing)



Source:
Herb Sutter,
The Free Lunch is Over

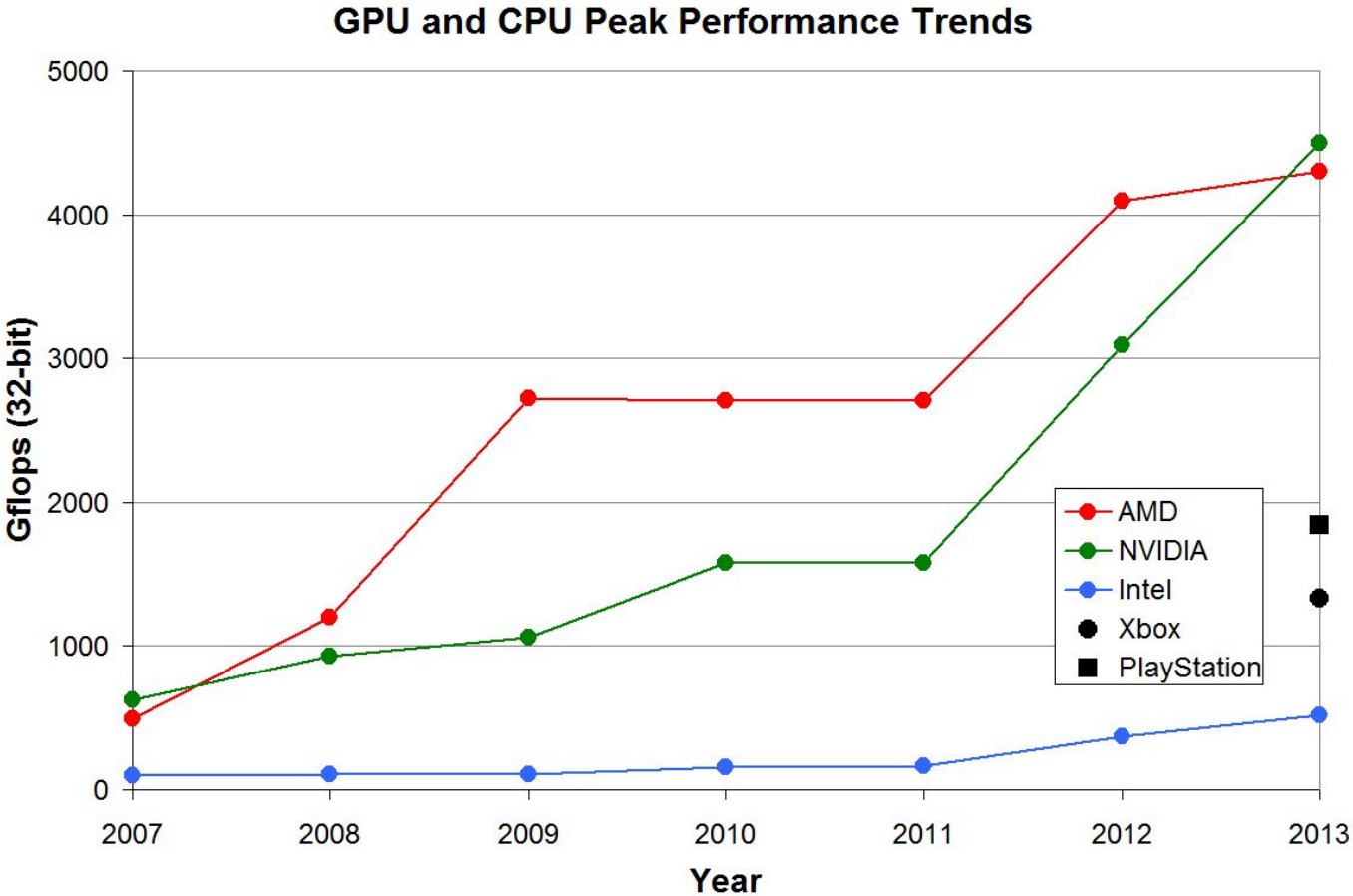
The End of CPU Frequency Scaling

- Moore's Law:
 - Transistors on the an IC doubling every ~2 years,
 - 1971 to ~2020
 - RIP
- Dennard Scaling:
 - Constant power density
 - 1974 to 2005
 - RIP
- Off-chip IO and Caching
 - Grow at a much slower rate
 - Cache hierarchy can hide this, but ...



Let's look at the alternatives

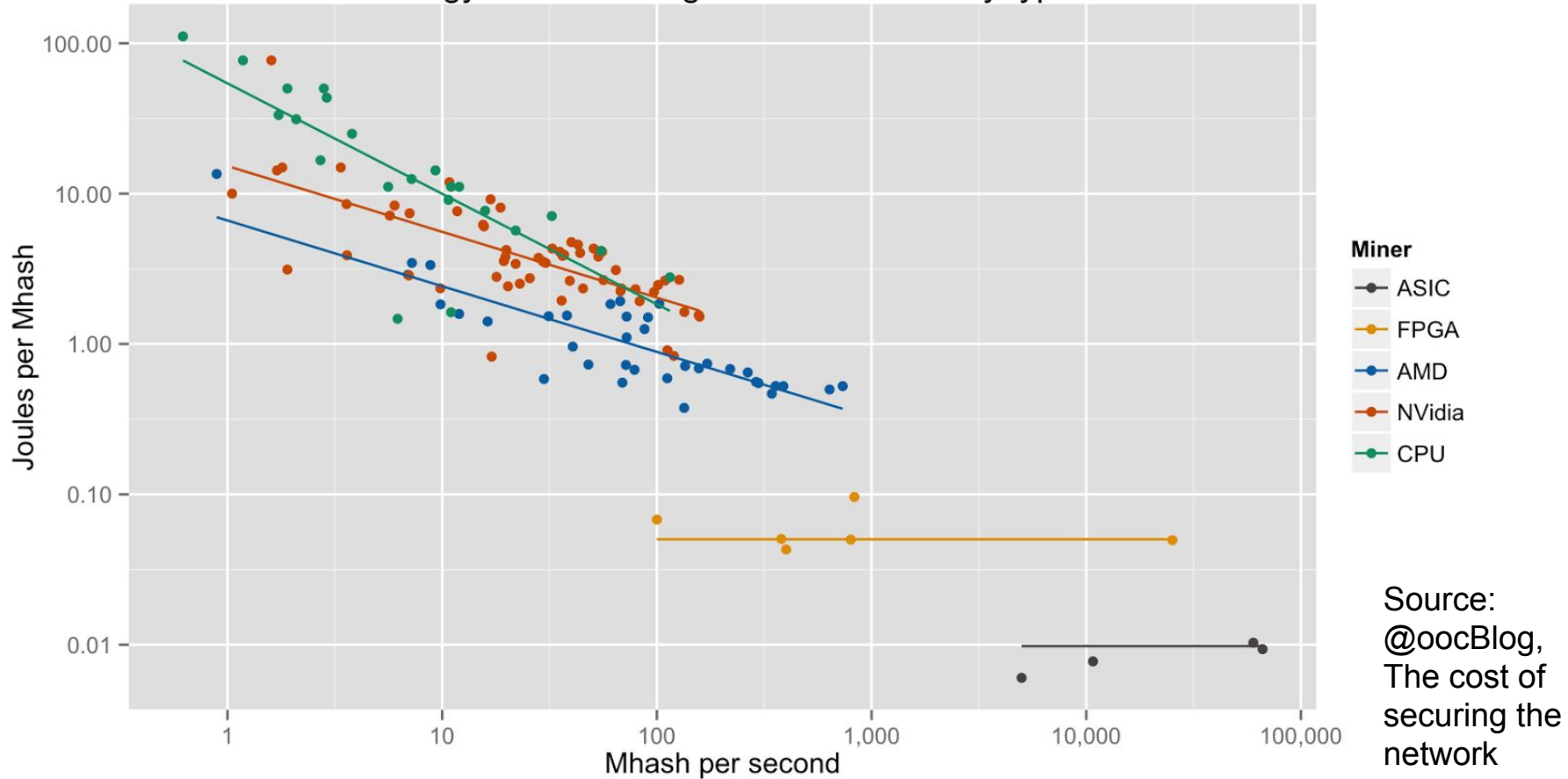
Performance



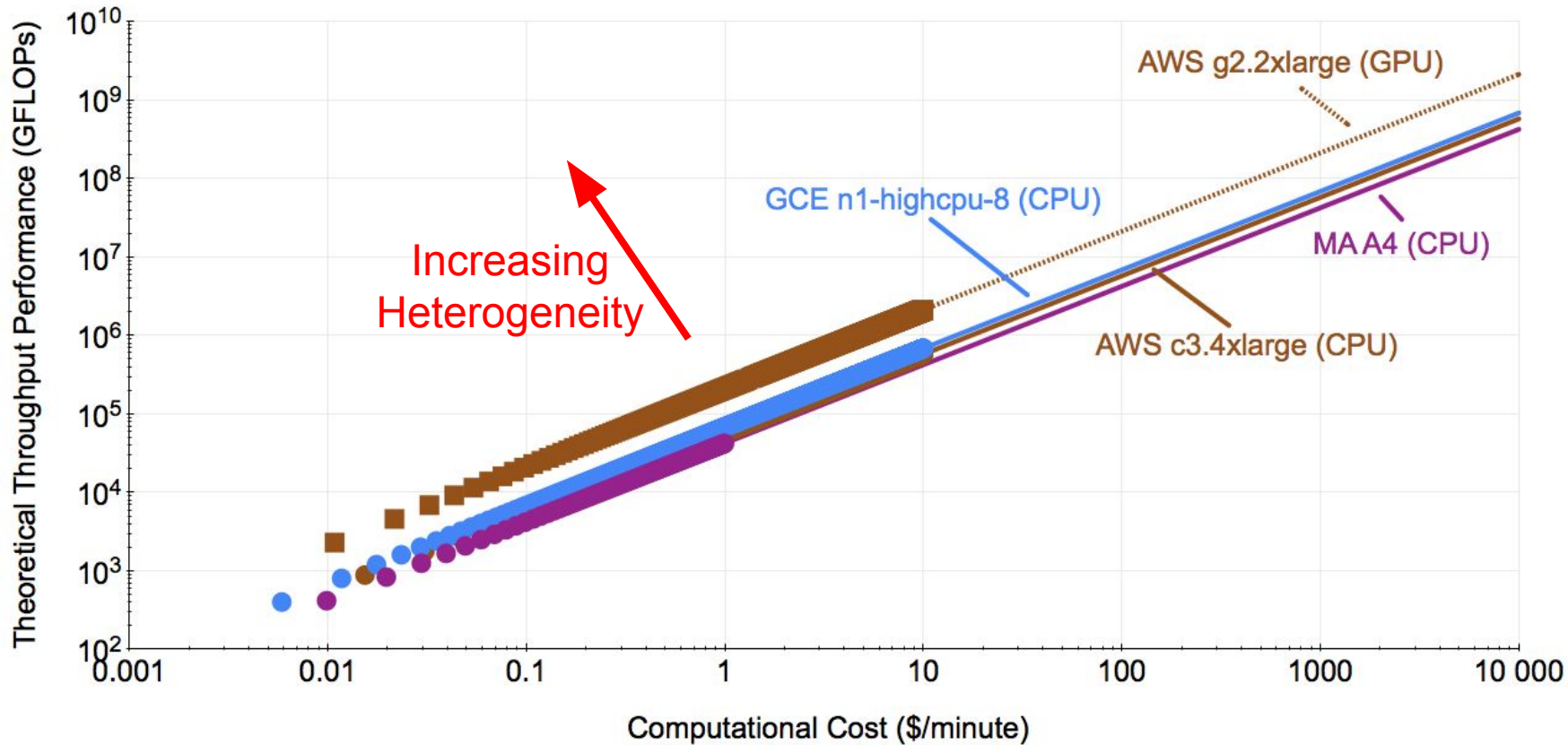
Source: Scott Feller, GPU Acceleration

Energy

Energy cost of mining for various tools by type



Cost



(... as of March 2015)

Heterogeneous Computing
sounds great,
but...

The Challenges of Heterogeneous Computing™

1. The Orientation Problem
2. The IO Problem
3. The Conceptual Problem

Workshop Goals

I want to help you address the challenges of heterogeneous computing:

- Orientation Problem
 - Program multicore CPUs and GPUs
 - Inspect device characteristics at runtime
- IO Problem
 - Transfer data efficiently
 - Use device memory hierarchy effectively
- Conceptual problem
 - Express task vs data parallelism
 - Exploit the comparative advantages of the devices available

Workshop Schedule

Arrive (9am)

- Introduction
- Module 1: Programming fancy devices with OpenCL

Tea (10:30am - 10:45am)

- Module 2: Moving bits around with OpenCL
- Module 3: Doing much task wow with OpenCL

Lunch (12:15pm - 1:00pm)

- A Challenging Time (with OpenCL)

Go Home (4pm)

Workshop Method

- You're going to code throughout
- Explain-Solve-Discuss cycle:
 - Brief Explanation
 - Short problem solving
 - Solution discussion
- Two iterations per module

Workshop Tools

We're going to be using:

- OpenCL and PyOpenCL
 - Intel and NVIDIA OpenCL SDKs and runtimes
 - Python OpenCL bindings (host language)
- Jupyter Notebooks
 - Nice way to work remotely
 - Rapid iteration
- AWS
 - Cheap Infrastructure

Module 1

Programming Fancy Devices (with OpenCL)

In this module...

- How to compile an OpenCL program (and what that is)
- How to run an OpenCL program

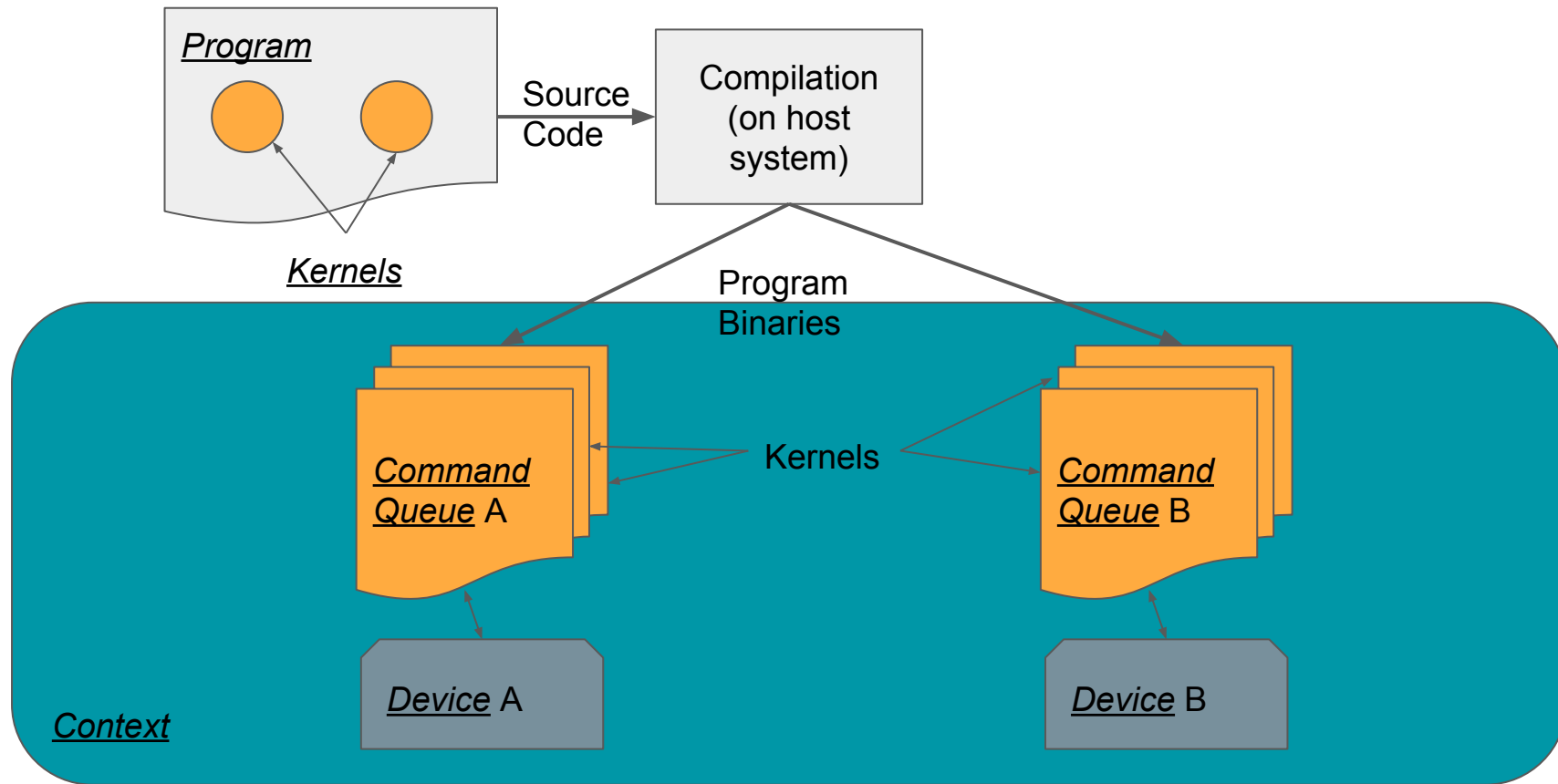


source: Khronos Group

What is OpenCL?

- ‘OpenCL™ (Open Computing Language) is the open, royalty-free standard for cross-platform, parallel programming of diverse processors found in personal computers, servers, mobile devices and embedded platforms.’
- Portable Heterogeneous Computing - not performance portability
- 2 APIs (programming and runtime) and a kernel language

OpenCL Programming API Abstractions

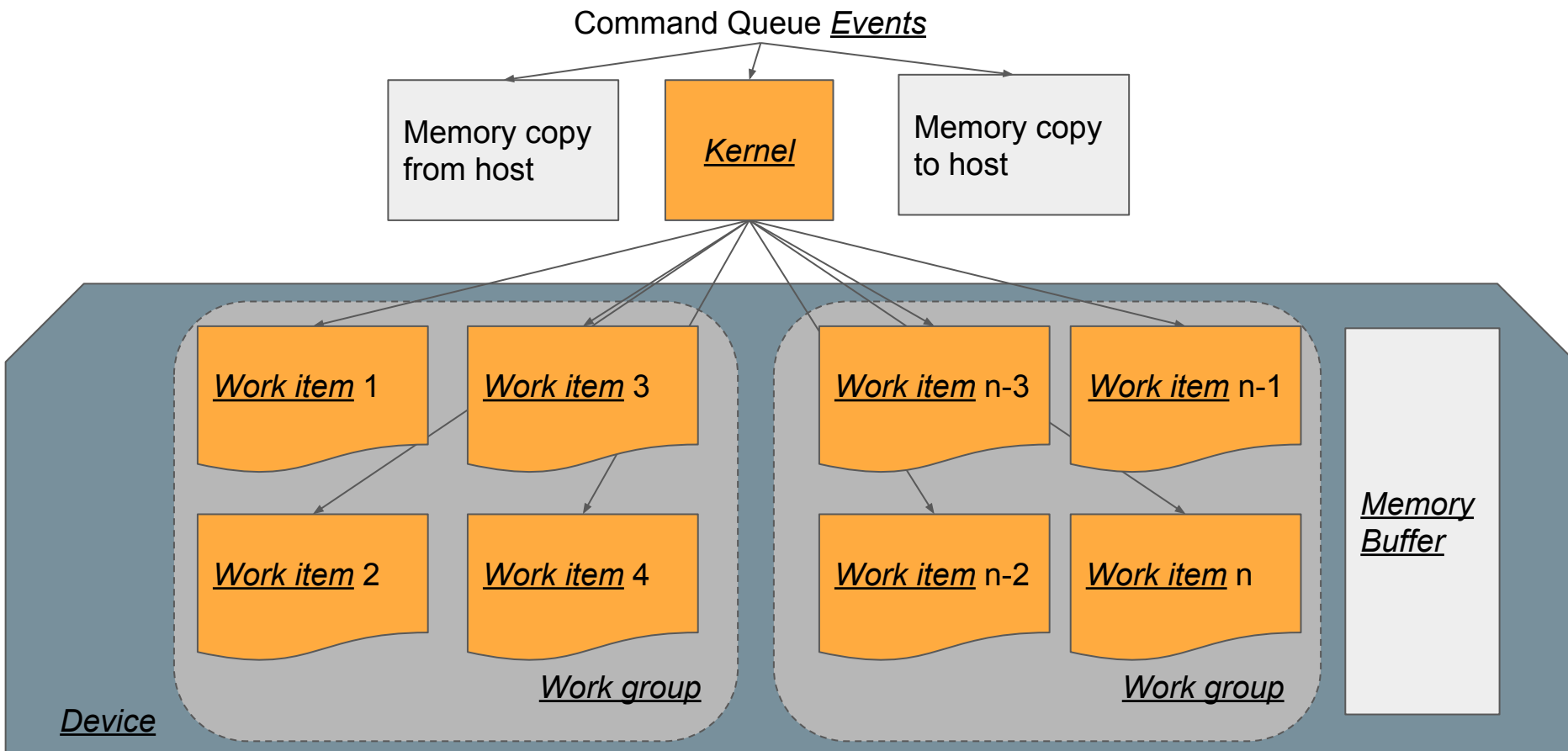


OpenCL Programming API Rules

1. Program source code must conform to OpenCL kernel code specification
(C without dynamic memory allocation)
2. One Platform (vendor implementation) per Context
3. One Command Queue per Device

Let's build a program!

OpenCL Runtime API Abstractions



OpenCL Runtime API Rules

1. The event ordering of the queue is respected, unless otherwise specified
2. Device memory can only be managed from the host
3. Each work item runs identical code and may execute in any order
4. Work item indices can have up to 3 dimension

Let's run a program!

Short Problem

- Use the Intel Platform to program the CPU to perform the same vector addition
- Compare the results - are they the same?
- Bonus: compare the timing of the two

Tea!

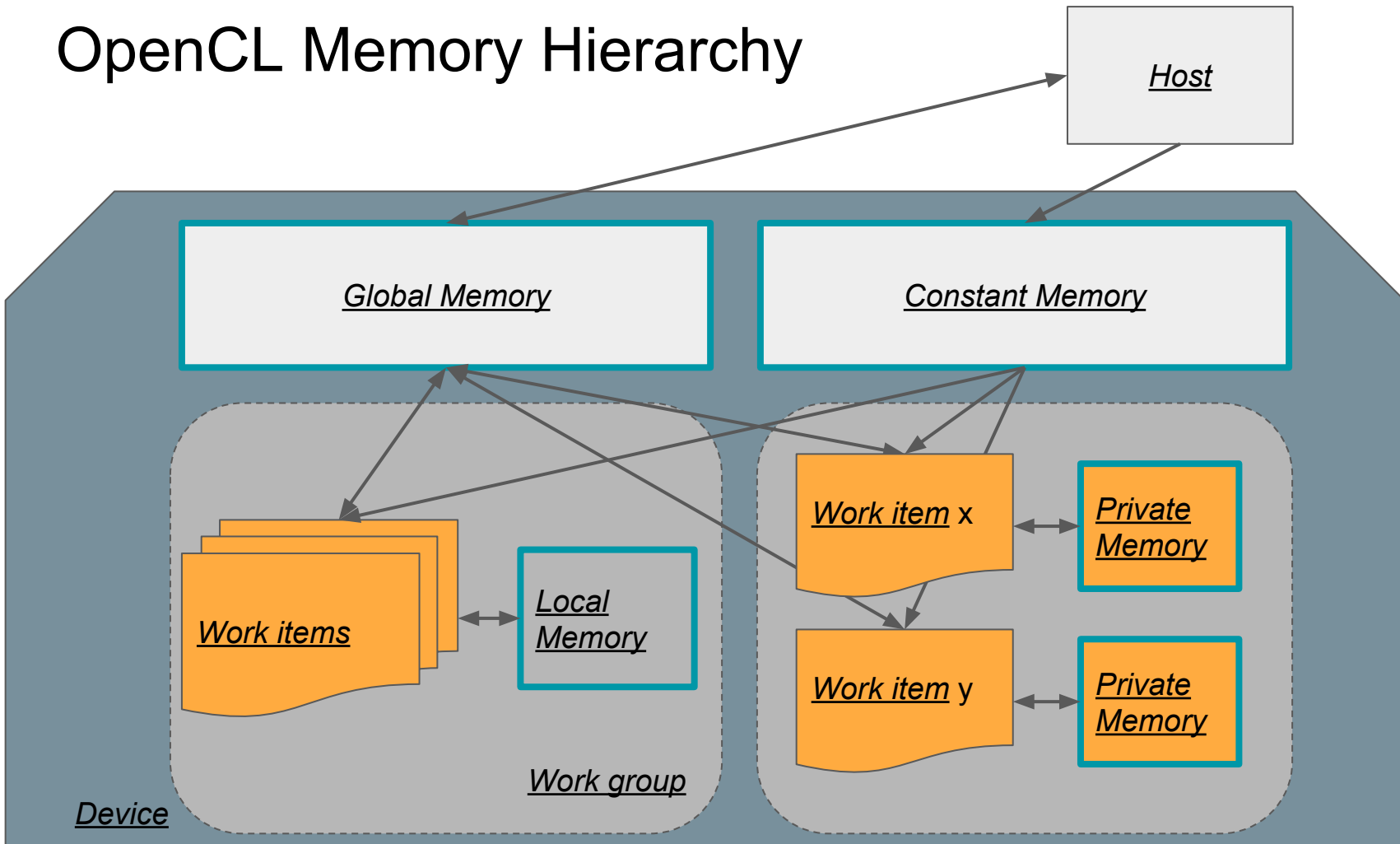
Module 2

Moving bits around

In this module...

- How to inspect the properties of our devices
- How to declare global memory buffers and using events to express dependencies
- How to declare local memory buffers and using memory fences to synchronise

OpenCL Memory Hierarchy



Let's inspect our devices!

Let's manipulate some memory!

Let's manipulate some more memory!

Short Problem

- Perform matrix multiplication (for a fixed size) using global, local and constant memories
- Measure the performance difference between the three
- Bonus: Write a few short kernels that help you work out how long a global, local and private memory read and write takes, approximately.

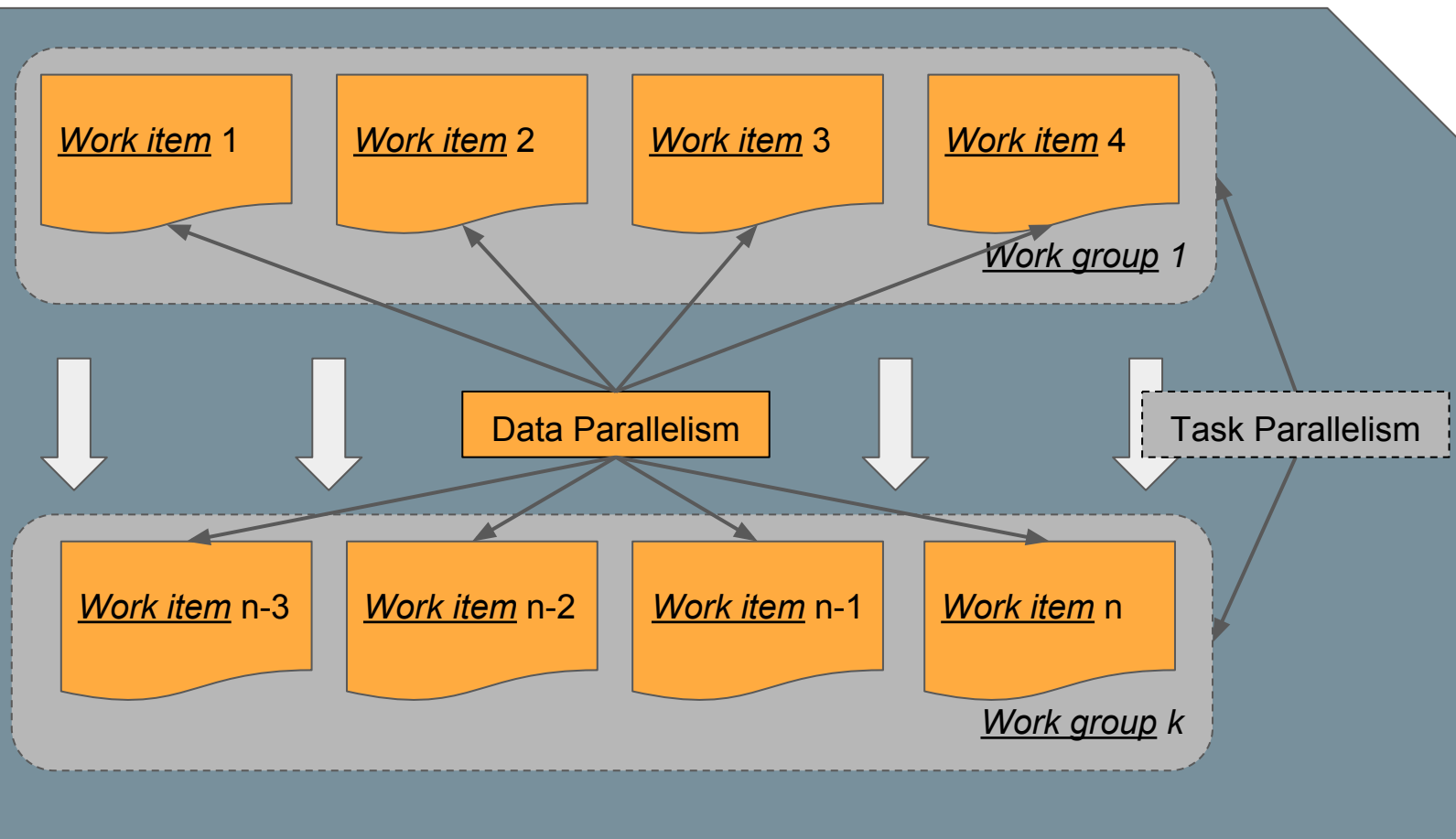
Module 3

Doing much task wow

In this module...

- How to express task vs data parallelism
- How to find the optimal workgroup size

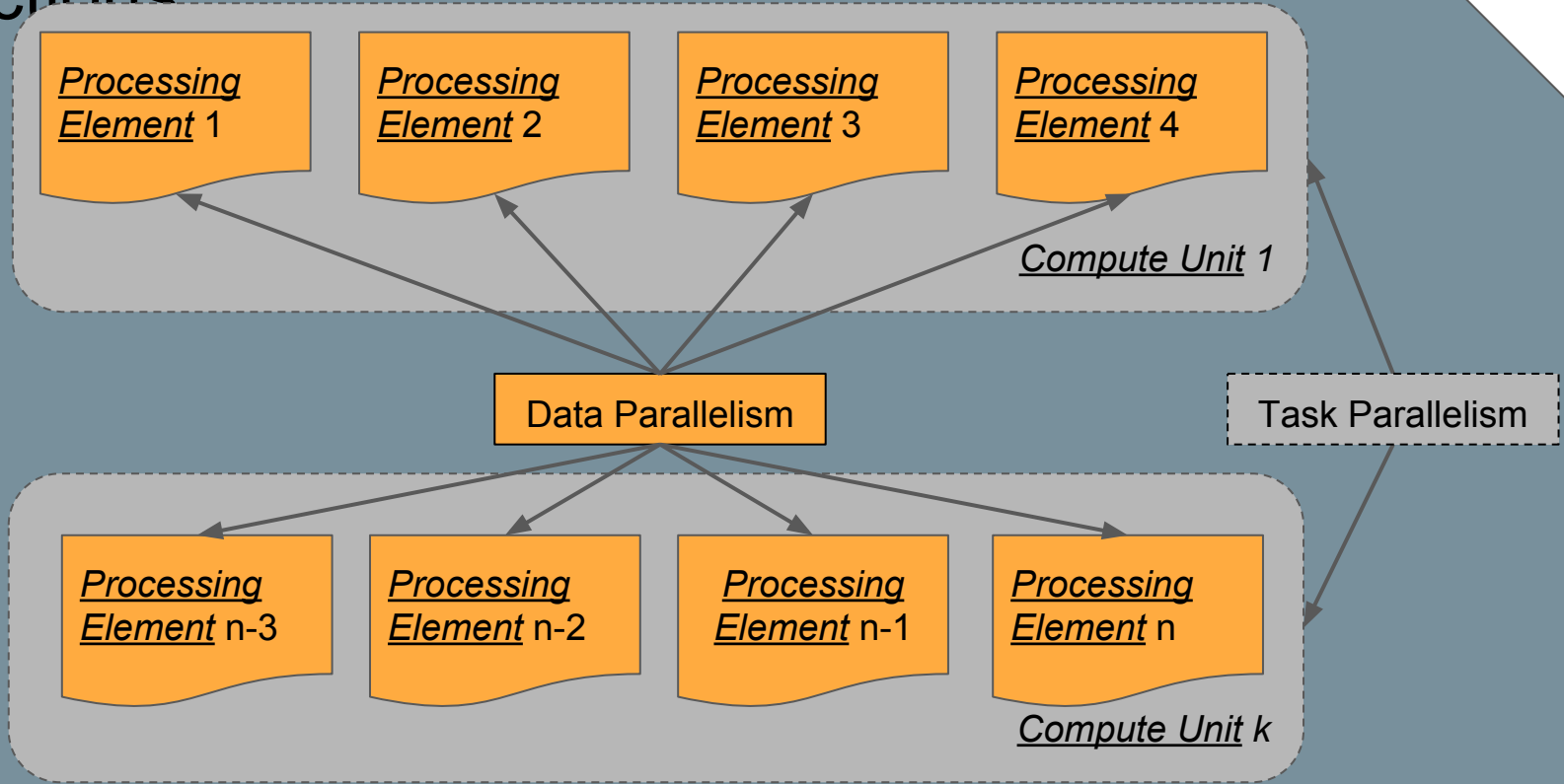
OpenCL Task vs Data Parallelism Abstractions



Let's do a lot at once!

OpenCL Task vs Data Parallelism Device

Abstractions



Device

Let's do a lot at once, efficiently!

Short Problem

- Implement any BLAS (matrix and/or vector) operation, using both the task and data parallel capabilities of the device
- Characterise the performance of your operation, and any improvements you've made
- Bonus: Look at how your operation scales

Shameless Punt

- Amazon Web Services:
 - Is Big (largest cloud computing provider in world)
 - Does interesting work (growing array of diverse services)
 - Has both technical and leadership career paths

- And,
 - ... is in Cape Town!

- Apply for Internships and Jobs



Lunch!

Challenging Time!

UCT OpenCL Challenge Workshop

- Apply a 100x100 Gaussian blur to the following three images:
 - The convolution must be done within an OpenCL kernel
 - Submission is a Jupyter notebook
- Score:
 - $80\% - (2 \times \text{sum of sequential CPU implementations latencies}) / (\text{sum of your implementations' latencies})$
 - 20% - accompanying write up (in Jupyter notebook format)
- Deadline: 23:59pm, 11 September 2016

Shameless Punt

- Amazon Web Services:
 - Is Big (largest cloud computing provider in world)
 - Does interesting work (growing array of diverse services)
 - Has both technical and leadership career paths

- And,
 - ... is in Cape Town!

- Apply for Internships and Jobs



Go *Away!*