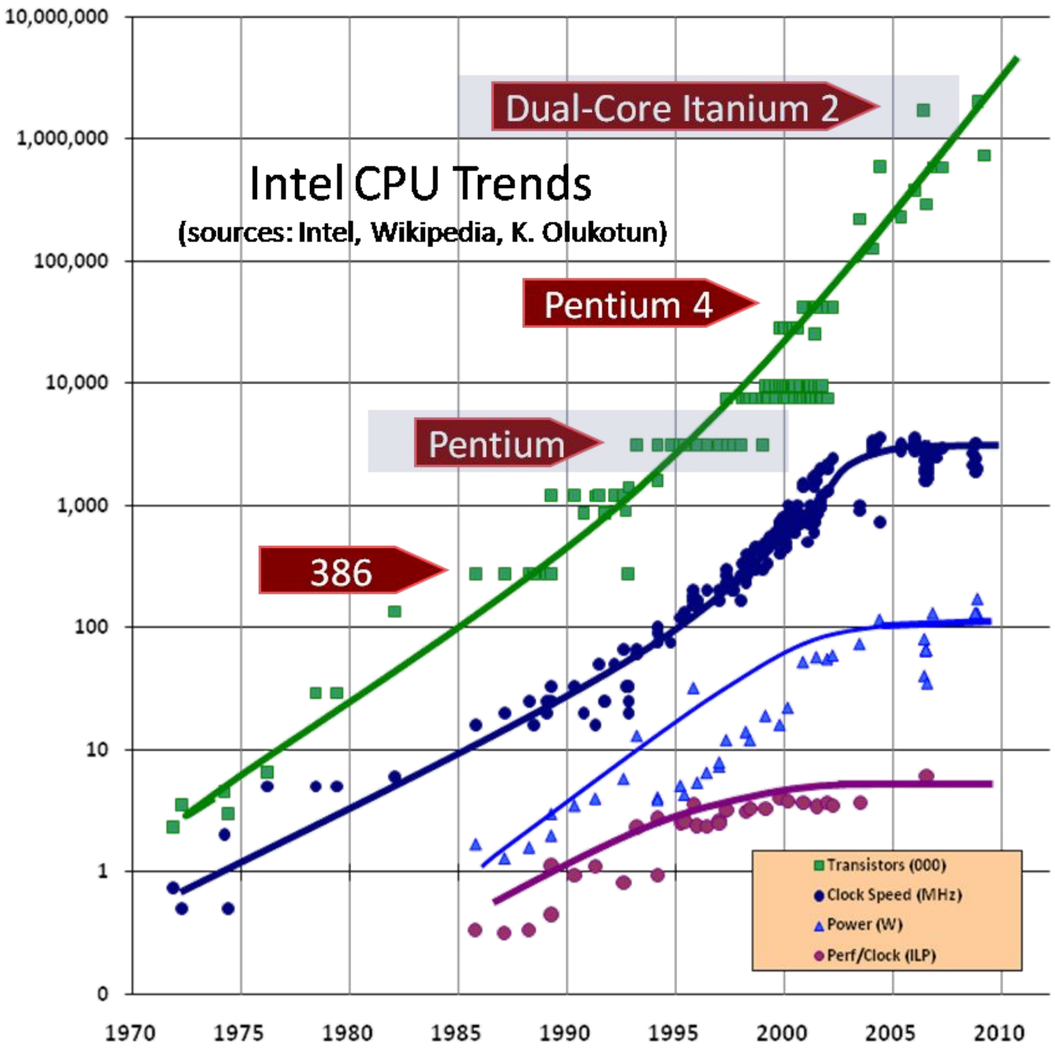
Parallel Programming with OpenCL for Fun and Profit

Dr Gordon Inggs
Amazon Web Services EC2

Why heterogeneous computing?

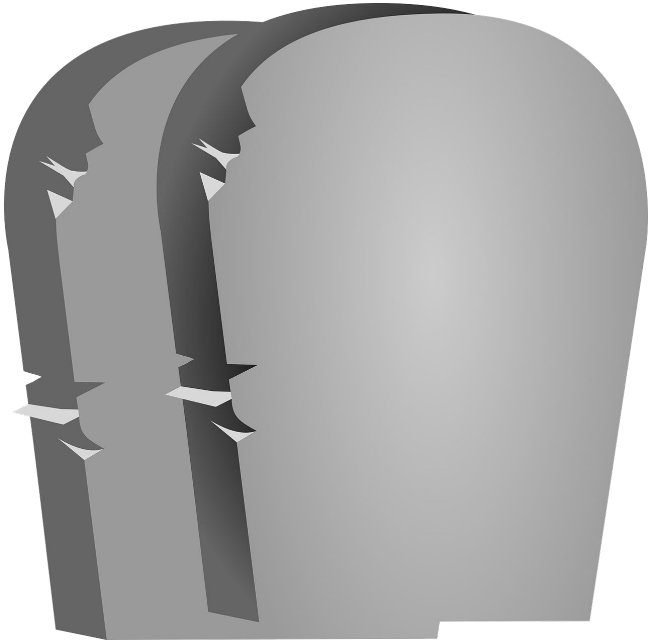
Why computers are becoming stranger (and why that's a good thing)



Source:
Herb Sutter,
The Free Lunch is Over

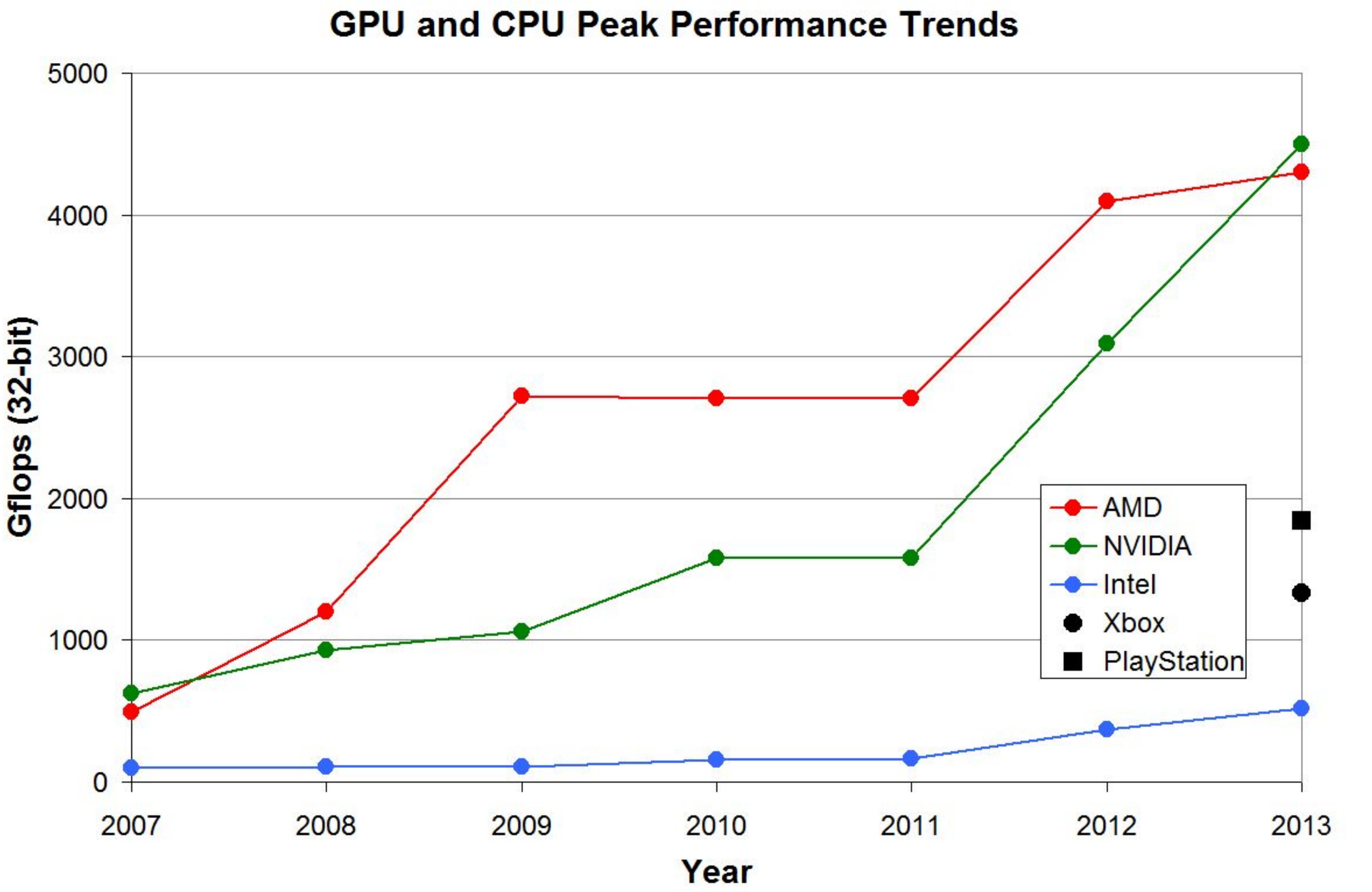
The End of CPU Frequency Scaling

- 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 ...



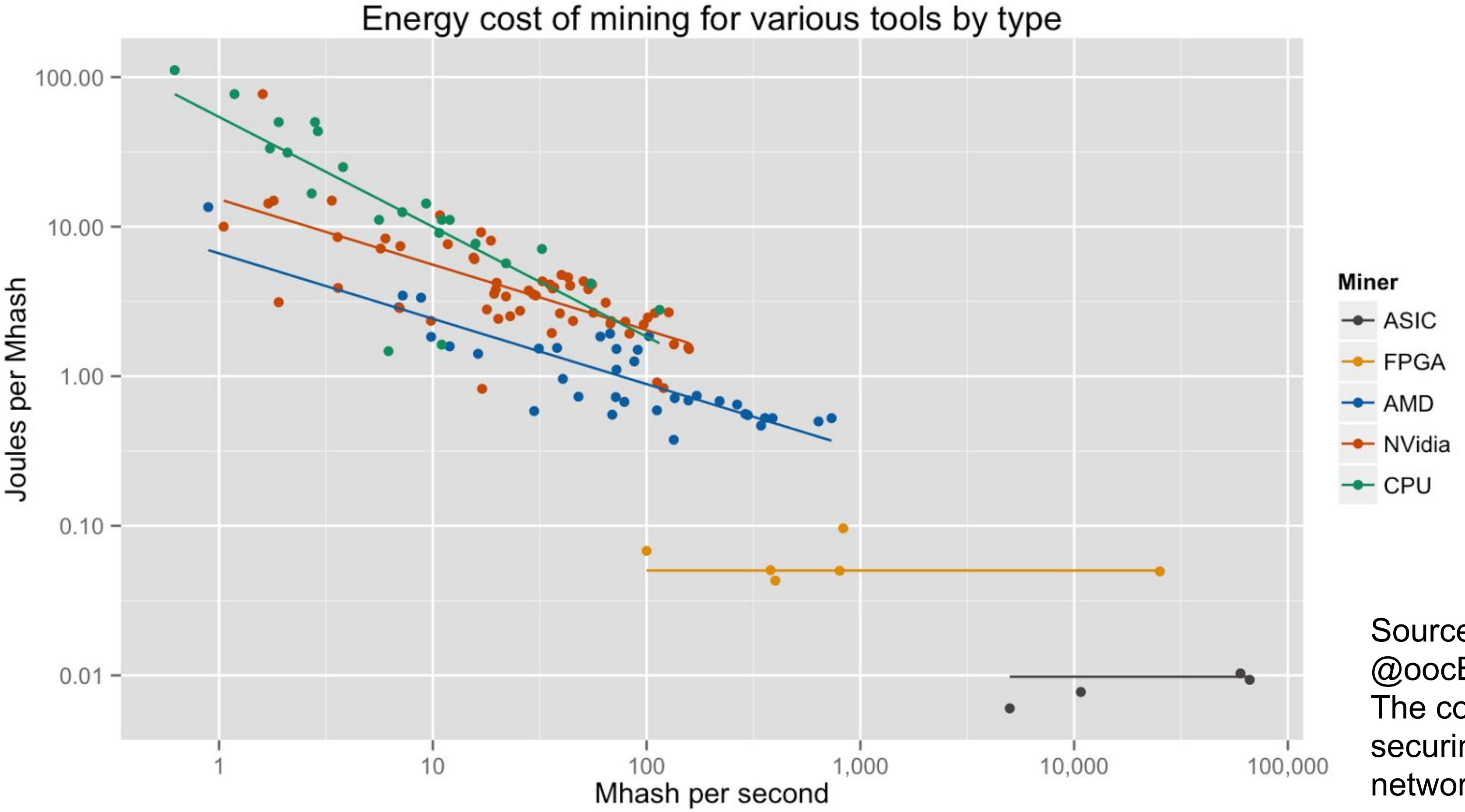
Let’s look at the alternatives

Performance



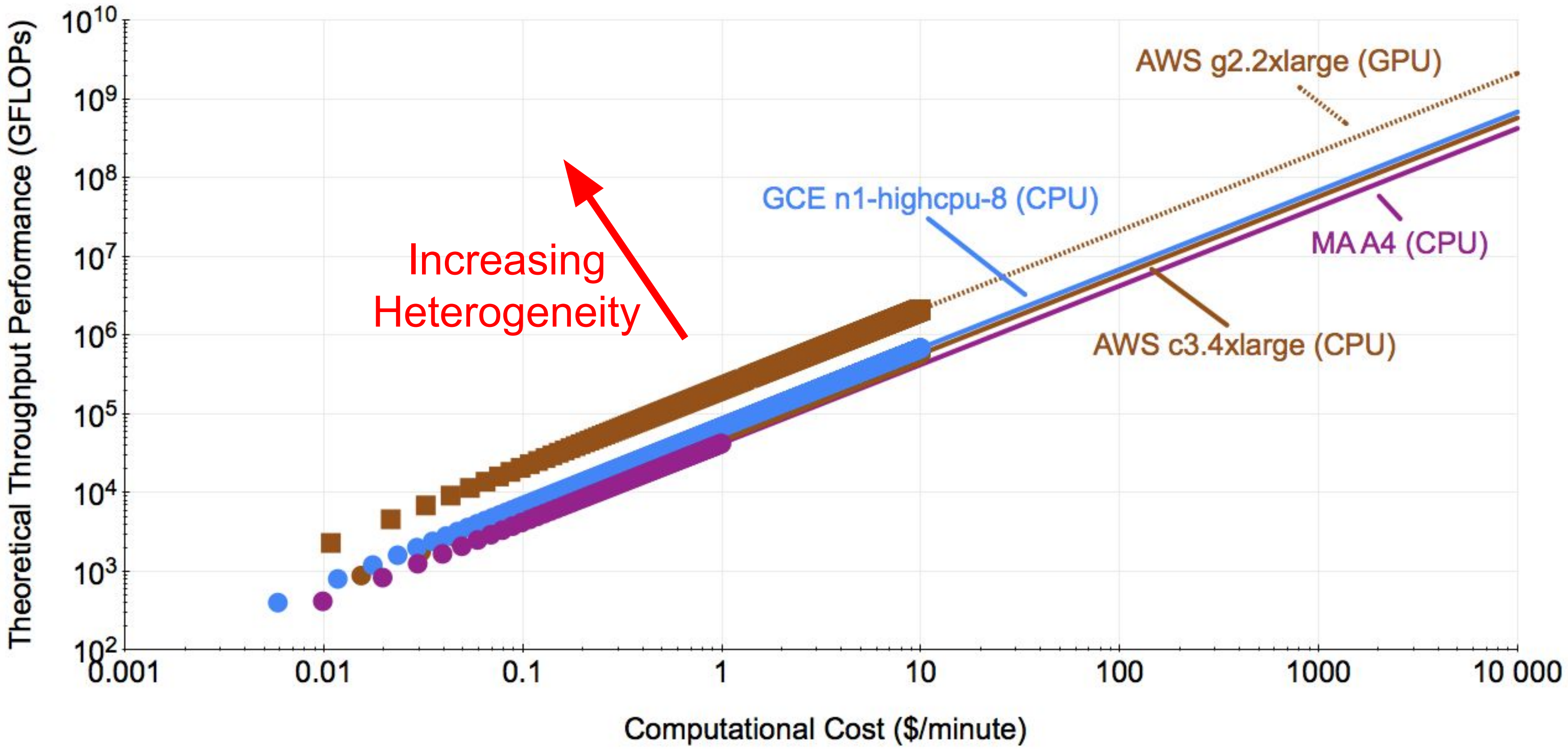
Source: Scott Feller, GPU Acceleration

Energy



Source: @oocBlog, The cost of securing the network

Cost



(... as of March 2015)

Heterogeneous Computing
sounds great,
but…

The Challenges of Heterogeneous Computing^{TM}

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

3D Incorporated Adobe ALTERA amazon.com AXELL CORPORATION AXIS COMMUNICATIONS BASEMARK BLIZZARD BROADCOM cadence CANONICAL

CEVA codeplay cognivue COLUMBIA UNIVERSITY Continental COREAVI DASSAULT SYSTEMES DMP

ENTROPIC ERICSSON ETRI freescale semiconductor FUTUREMARK CORPORATION Daijin Google harmonic HUAWEI IBM

Imperial College London ITRI Industrial Technology Research Institute itseez KDAB KISHONTI KNU KYUNGPOOK NATIONAL UNIVERSITY LG Linaro LUCASFILM Ltd

BOARD OF PROMOTERS

AMD ARM Apple SONY SAMSUNG EPIC GAMES

KHRONOS GROUP

Over 100 members worldwide
any company is welcome to join

intel VIVANTE NOKIA NVIDIA Imagination QUALCOMM

MARVELL matrox MEDIATEK Mentor Graphics Microsoft MIT Lincoln Laboratory mobica Movidius mozilla MULTICOREWARE 日本大学

NEC Nintendo OSU Oregon State Oculus VR OXIDE Panasonic PIXAR POLITECNICO DI MILANO RENESAS Rockwell Collins

서울대학교 SEOUL NATIONAL UNIVERSITY Silicon Studio socionext SPREADTRUM symbio SYNOPSYS TAKUMI TAMPERE UNIVERSITY OF TECHNOLOGY TU WIEN TECHNISCHE UNIVERSITÄT WIEN

TES Electronic Solutions TEXAS INSTRUMENTS THINCI Think Silicon tobii TOSHIBA TRANSGAMING unity University of BRISTOL UNIVERSITY OF CAMBRIDGE

University of Windsor VALVE VeriSilicon VIA Visteon vmware WARGAMING.NET XILINX zSpace

source: Khronos Group

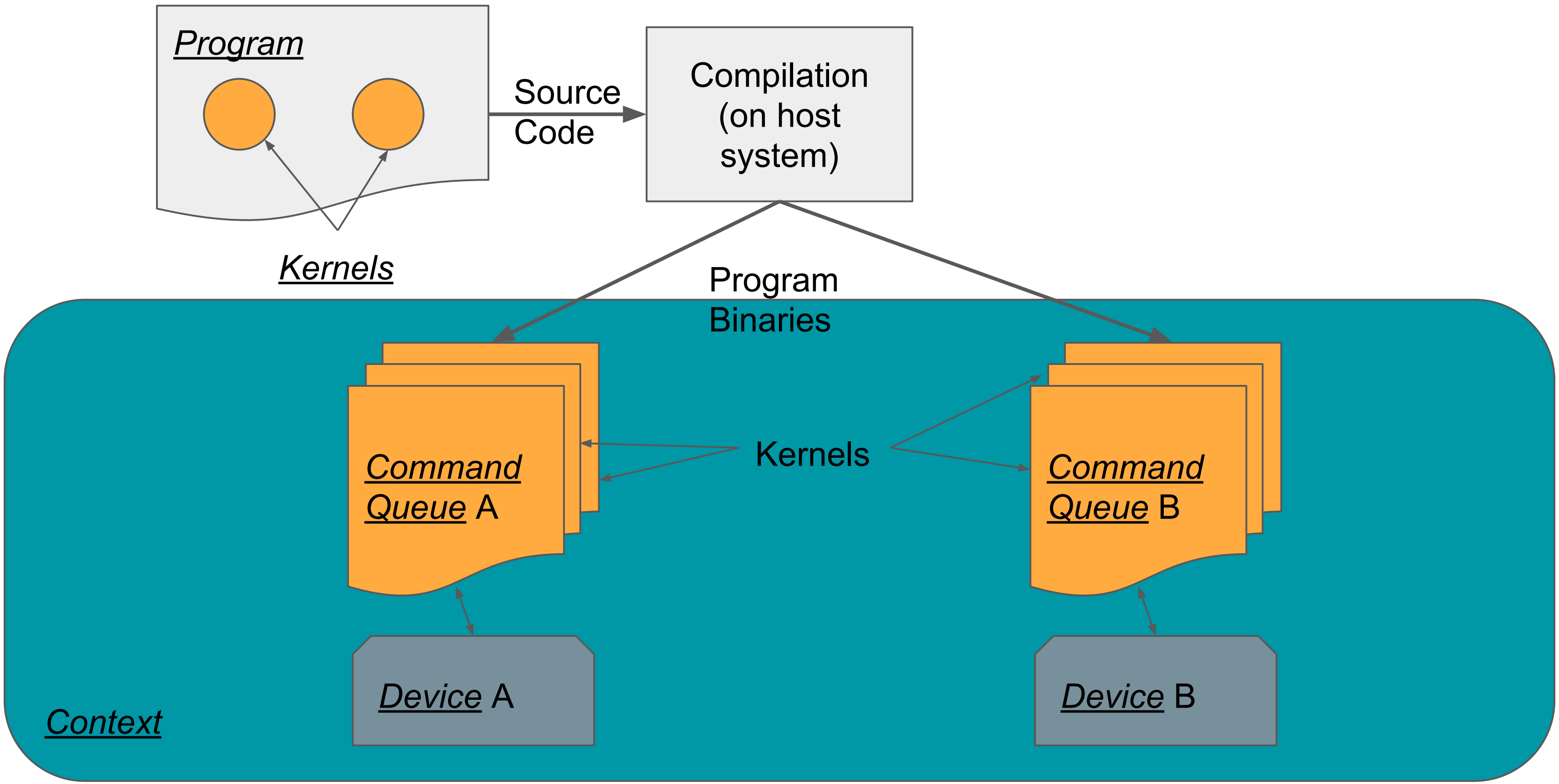
What is OpenCL?

- ‘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 - <u>not performance portability</u>

- 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

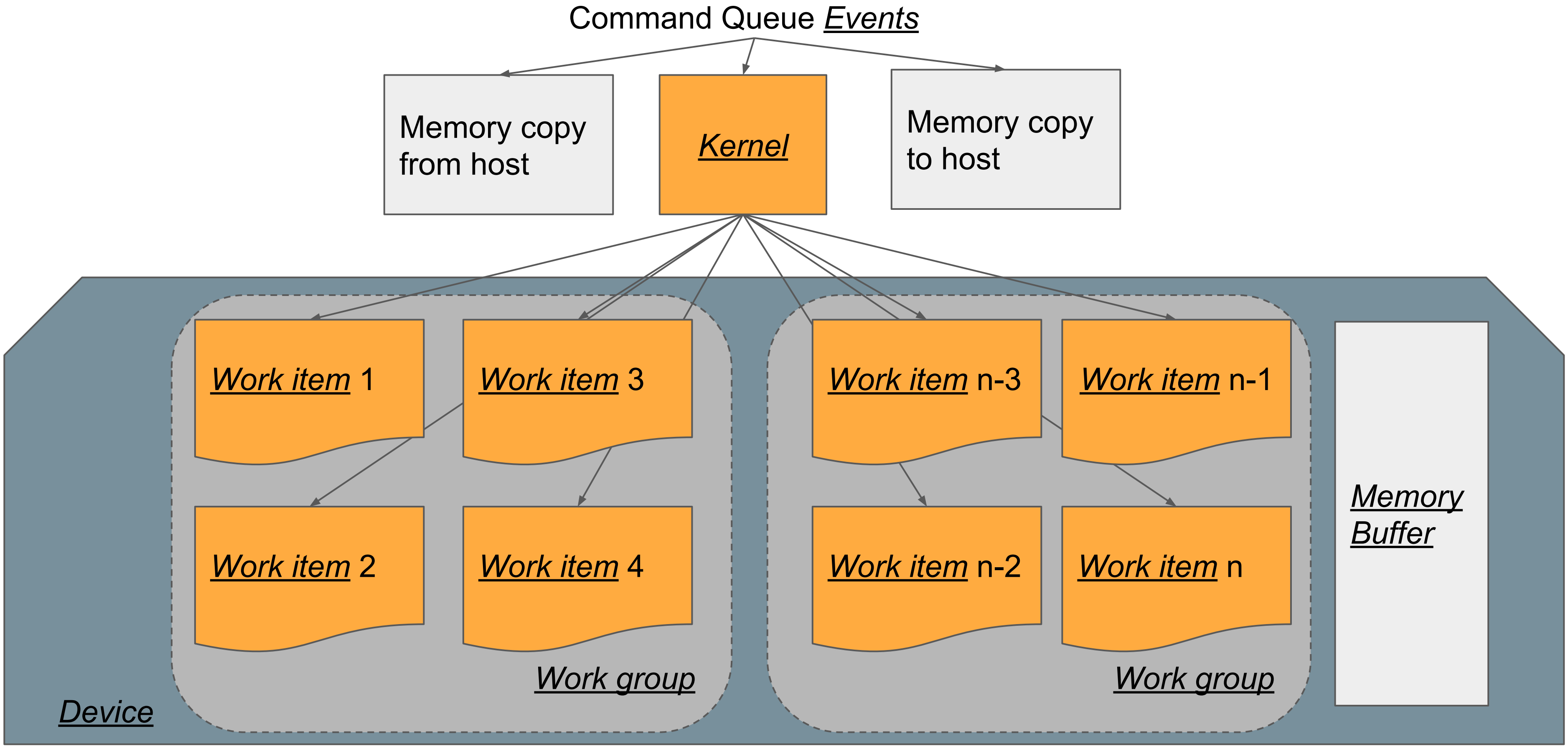
   (C without dynamic memory allocation)

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

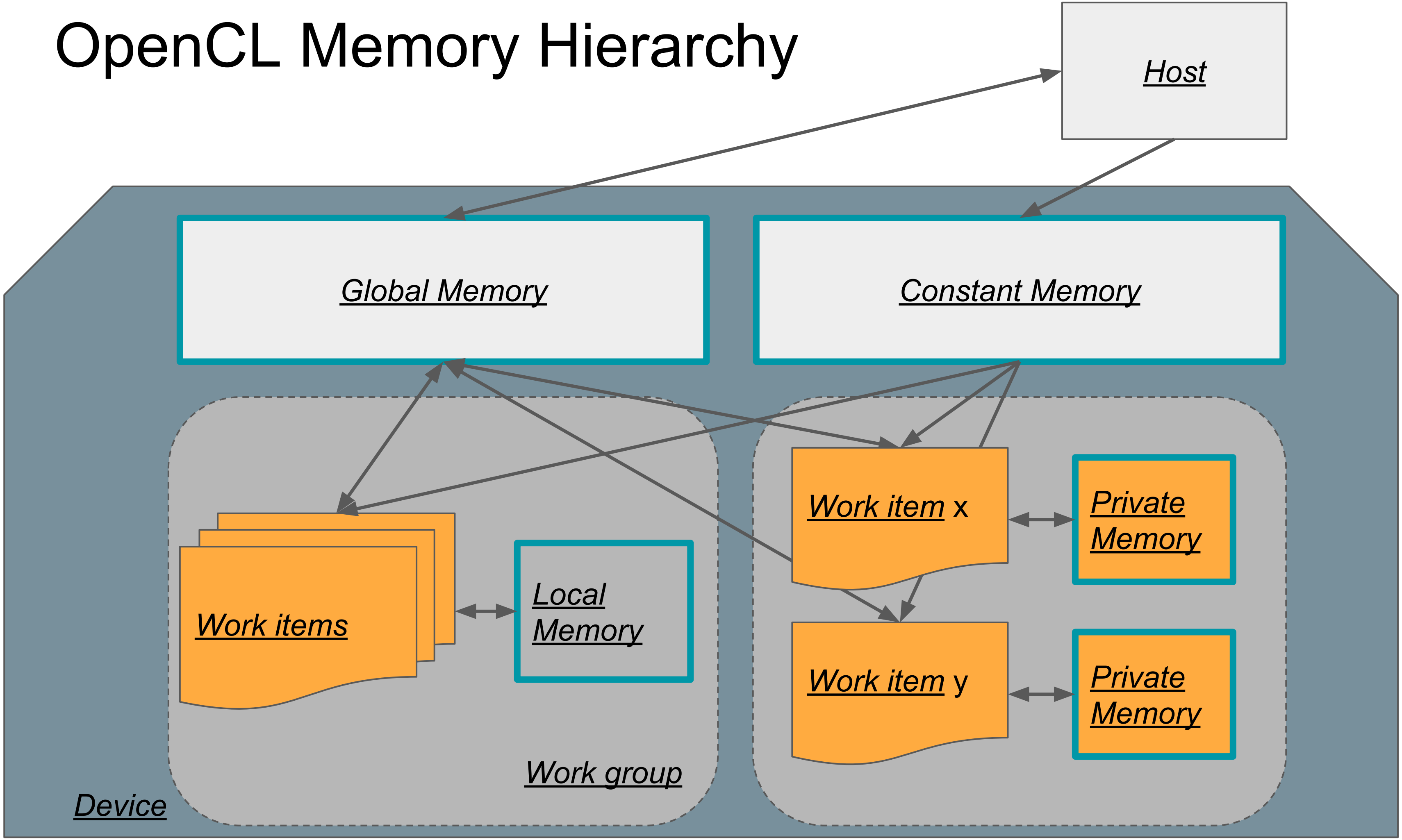
Tea!

Module 2
Moving bits around

In this module...

- How to 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



Global Memory and Events

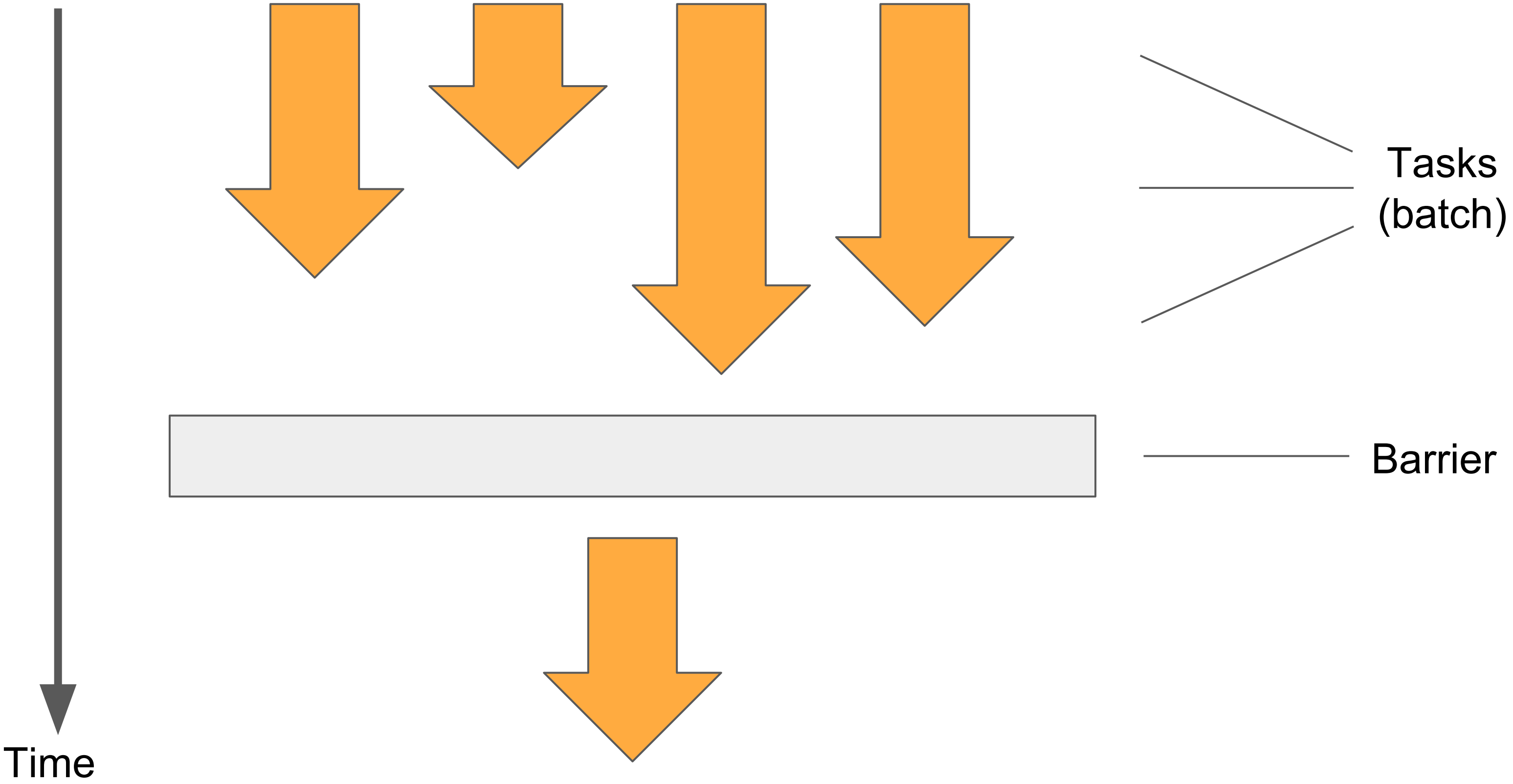
- Roughly equivalent to RAM

- Global Memory Flags:
  - READ_ONLY
  - WRITE_ONLY
  - READ_WRITE

- Events = synchronisation between device and host

Let’s manipulate some memory!

Batch Synchronous Processing



Let’s manipulate some more memory!

Short Problem

- Perform matrix multiplication (for a fixed size) using global and local 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...

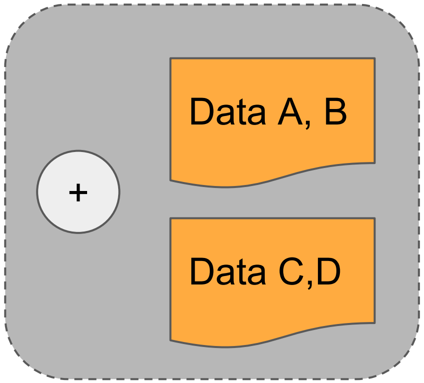
- How to inspect the properties of our devices

- How to express task vs data parallelism

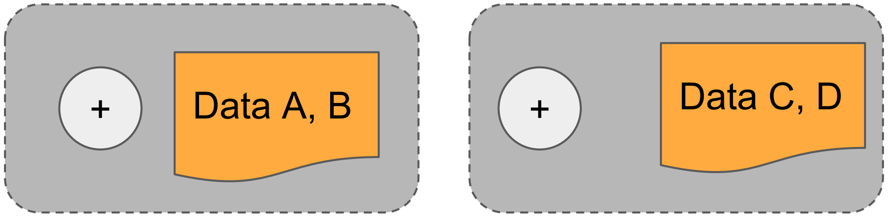
Let’s inspect our devices!

Task vs Data Parallelism

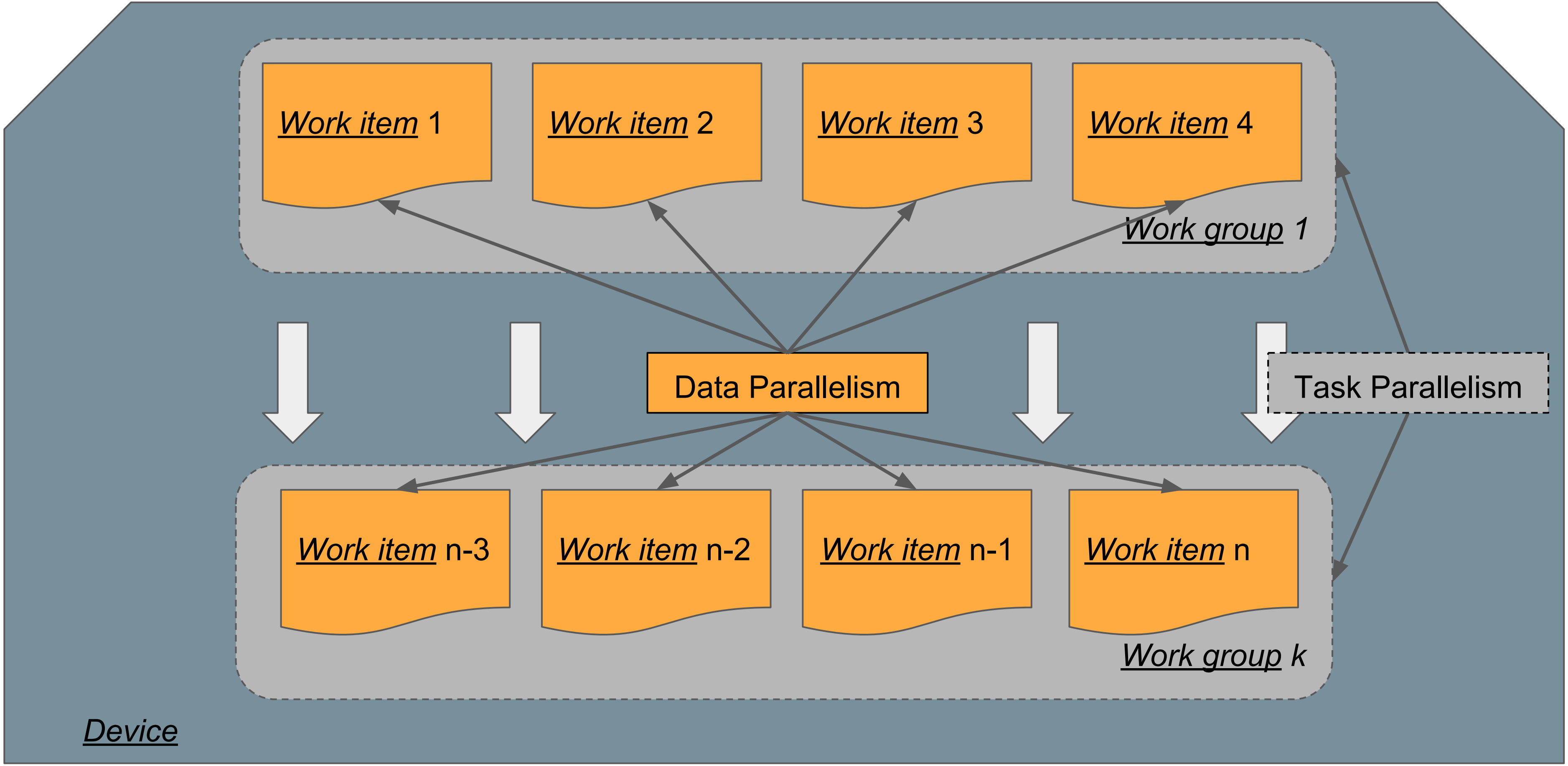
- Flynn’s Taxonomy - SIMD vs MIMD



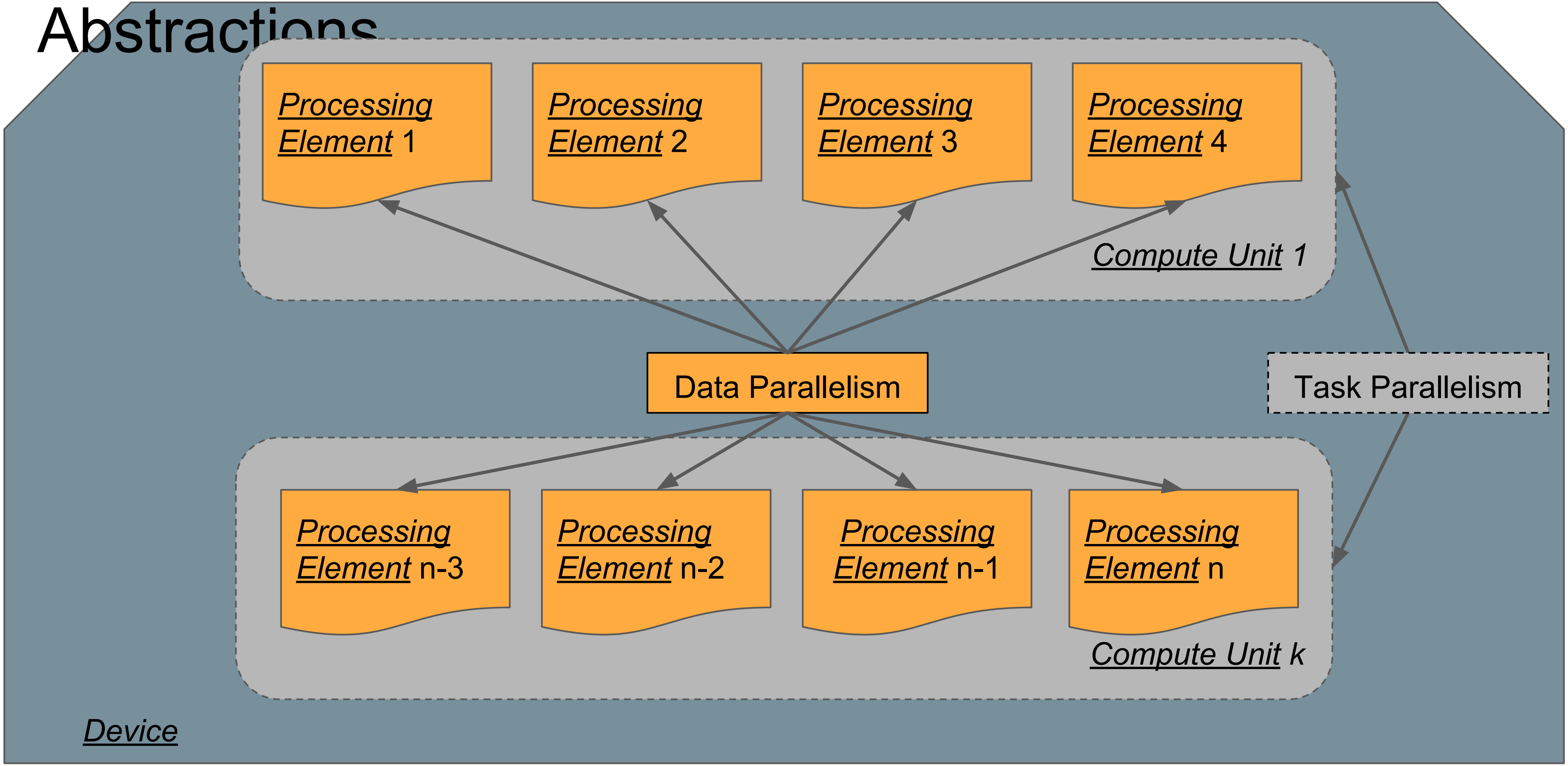
- Broadly:
    - Multicore CPUs => MIMD
    - GPUs => SIMD



OpenCL Task vs Data Parallelism Abstractions



OpenCL Task vs Data Parallelism Device Abstractions



Let’s do a lot at once!

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 Workshop Challenge

- Apply a 100x100 Gaussian blur to the following three images:
  - Fabio
  - Lion
  - World

- The convolution must be done within an OpenCL kernel
  - Submission as a Jupyter notebook, using provided template

- Score:
  - 80% - (2 x sum of sequential CPU implementations latencies) / (sum of your implementations' latencies)
  - 20% - accompanying write up (in Jupyter notebook format)

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!