MANCHESTER
1824

The University of Manchester

Heterogeneous Managed Runtime Systems: A Computer Vision Case Study

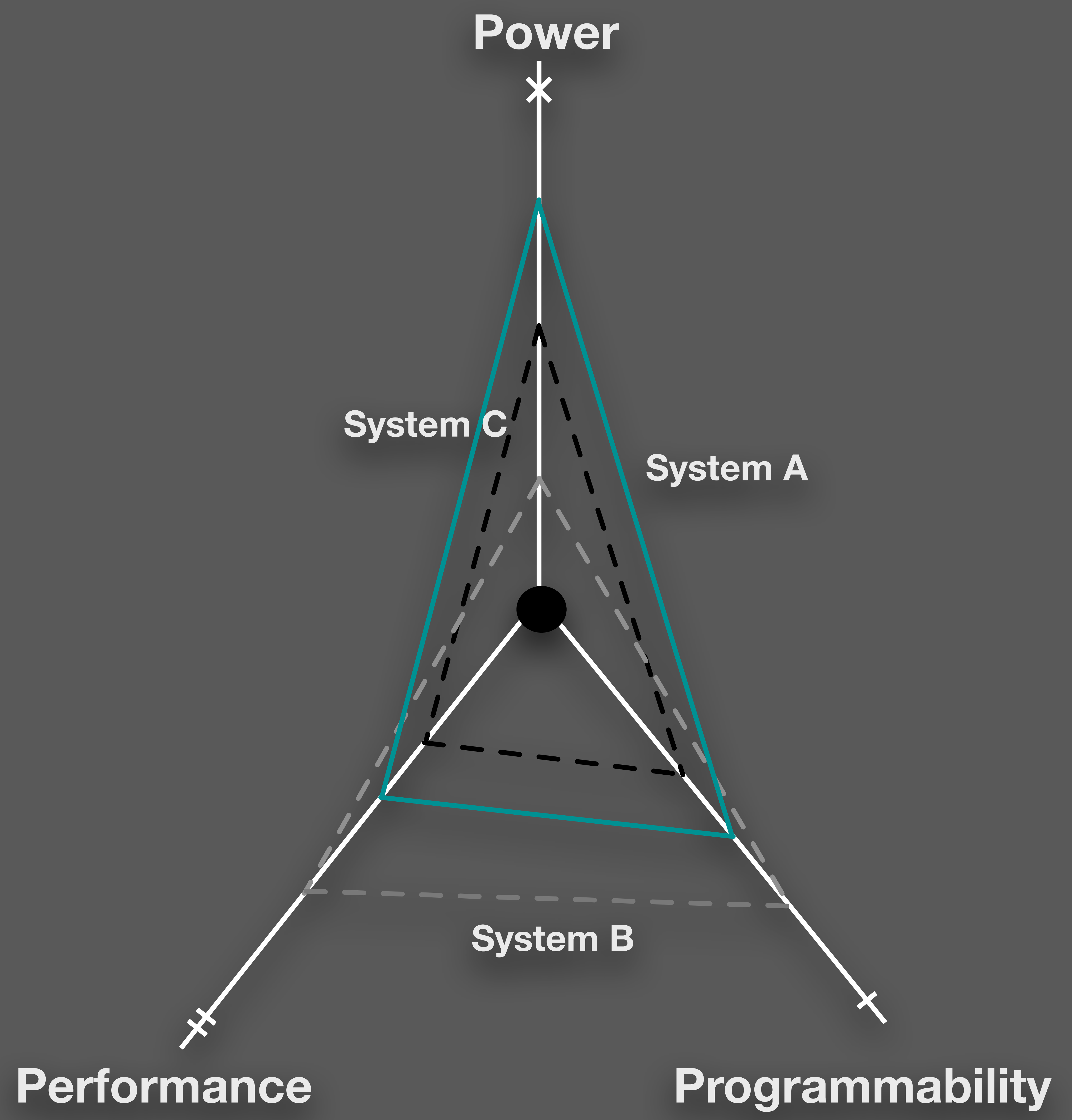
Christos Kotselidis, James Clarkson,
Andrey Rodchenko, Andy Nisbet, John Mawer, Mikel Luján
Advanced Processor Technologies Group

VEE, 08 April 2017

Computing Challenges

- More performance
- Less power
- Complex applications (NN, CV, etc.)
- Diversity of:
  - Hardware devices
  - Programming models
  - Programming languages

Pareto-optimal Point



Challenge Accepted

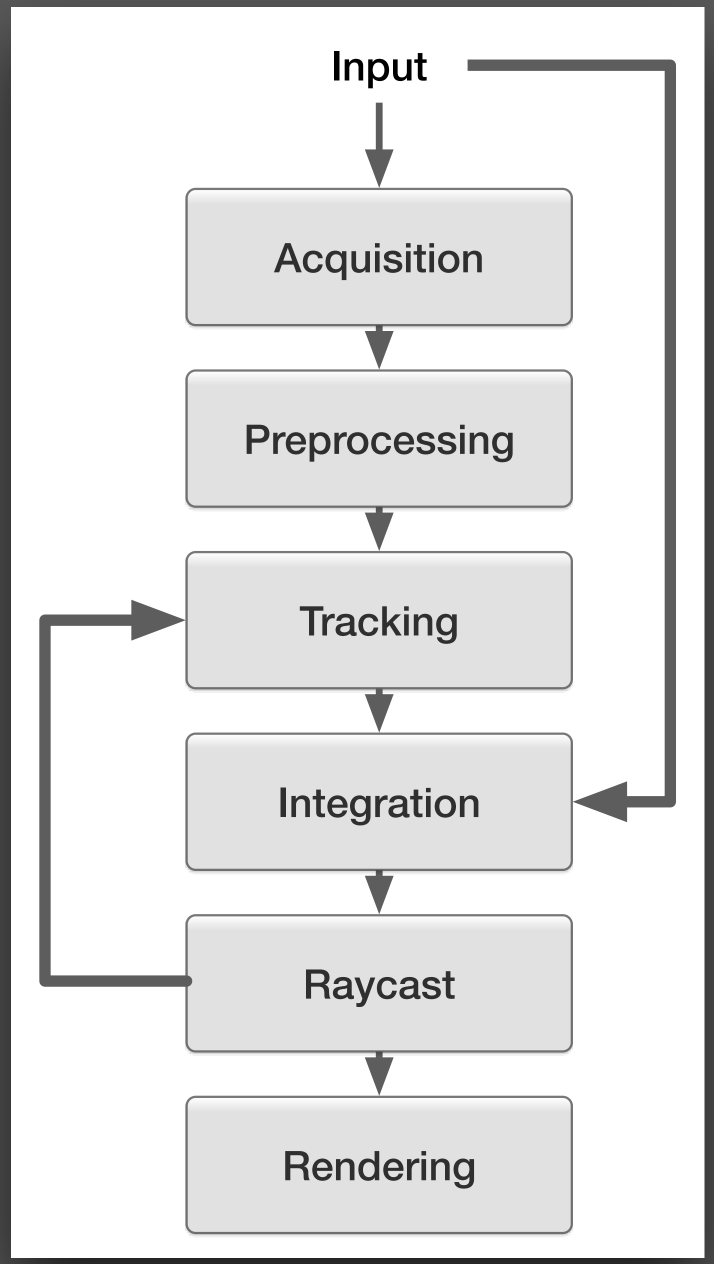
MANCHESTER 1824
The University of Manchester

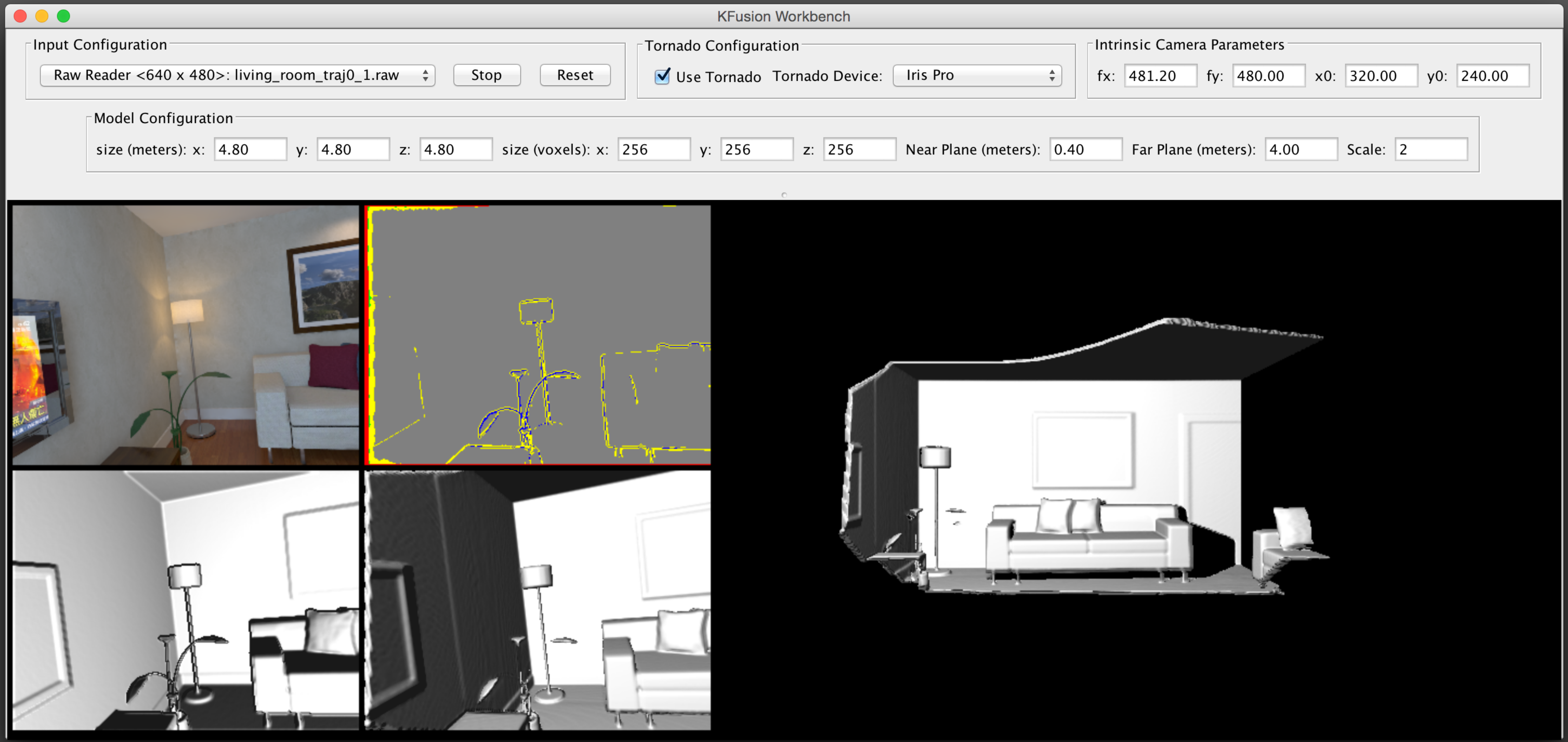
- “Write-once-run-everywhere”
- Exploit heterogeneous hardware
- Choose a demanding application
- Meet the QoS of selected application
- Generalize solution

Kinect Fusion

- 3D space reconstruction (RGB-D)
- Complex multi-kernel pipeline
  - 540-1620 kernels/second
  - SLA of 30 FPS
- Cutting edge robotics application
- Deploy across many combinations of platforms and accelerators

Kinect Fusion





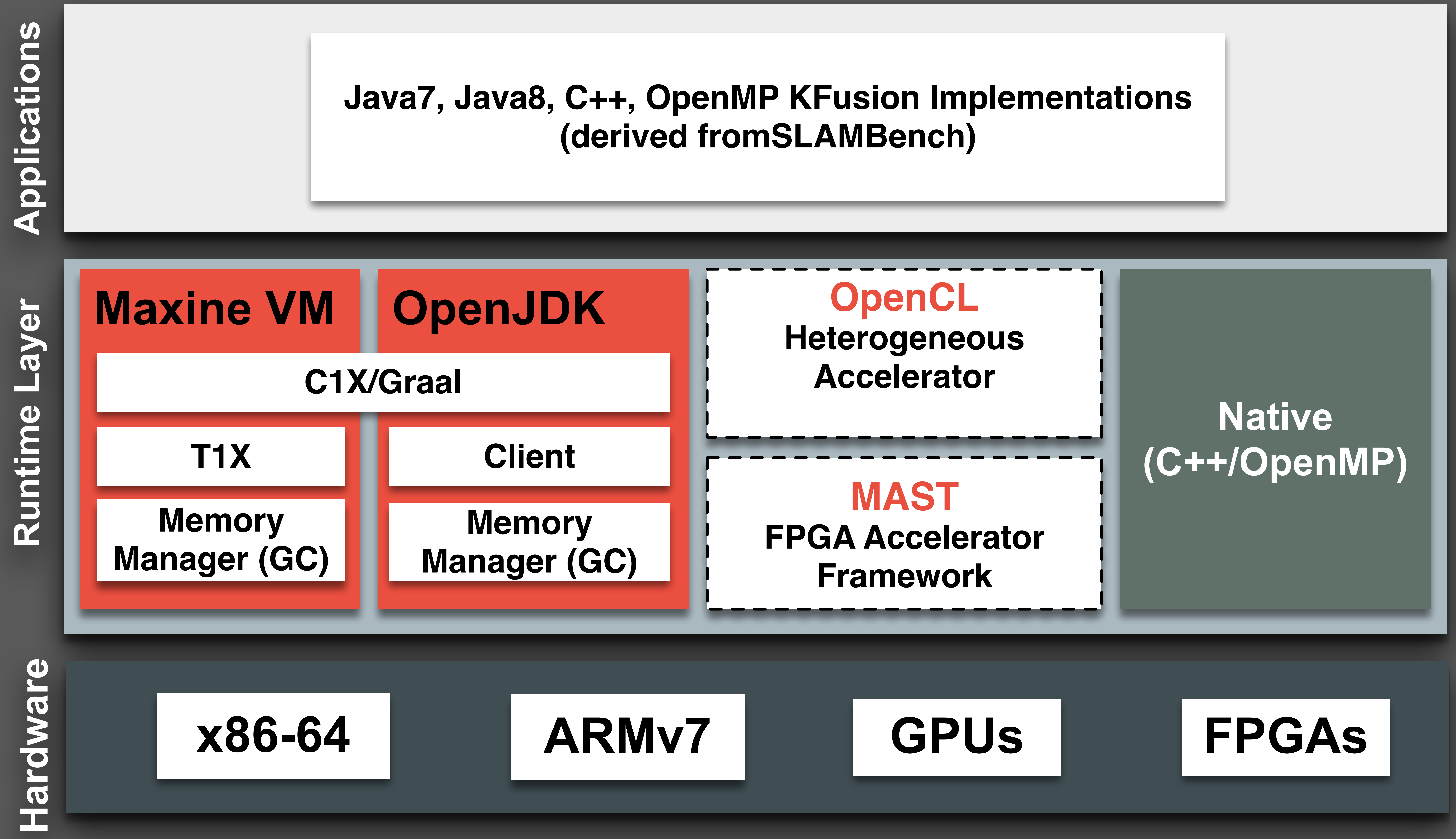
Platform | Objectives

The University of Manchester

- Application portability
- Increased productivity
- Hardware / Device diversity
- High performance
- Easy experimentation and prototyping

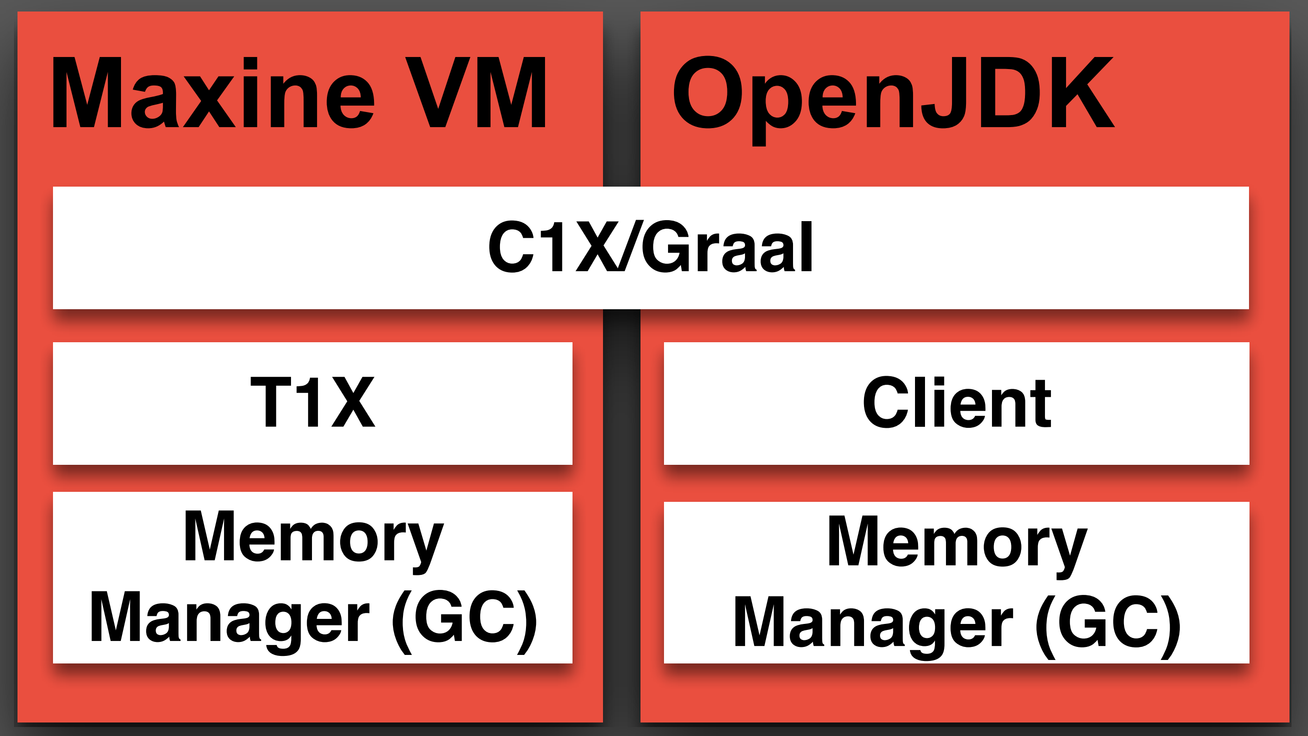
Platform | Floor Plan

MANCHESTER
1824
The University of Manchester



Platform | Maxine VM and OpenJDK

- Graal Integration and Interoperability
- Research vs. Industrial VMs
- Flexibility vs. Performance



- Maxine VM Improvements
  - Enable Profile-guided Optimisations
  - ARM Port (32bit transition)
  - Stability and Performance Improvements
  - x86-64: 1.64x over baseline Maxine VM 0.57x over HotSpot C2
  - ARMv7: SpecJVM2008, KFusion 100% passrate
    2.3x and 3.3x slower against HotSpot C1 and C2

Platform | OpenCL Acceleration

- OpenCL Acceleration Module
- Based on Graal/OpenJDK
- API, Compiler, Runtime
- Exploits any OpenCL Compatible Device
- Successor of JACC [1]

[1] James Clarkson, Christos Kotselidis, Gavin Brown, Mikel Lujan. ***Boosting Java Performance using GPGPUs***.
In 30th International Conference on Architecture of Computing Systems (ARCS), 2017.

Platform | OpenCL Acceleration

MANCHESTER
1824
The University of Manchester

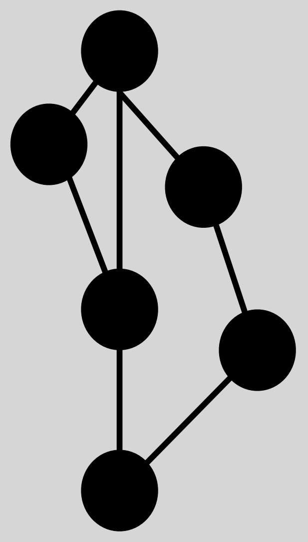
OpenCL/Java API

```
preprocessingGraph = new TaskGraph()
  .streamIn(depthImageInput)
  .add(ImagingOps::mm2metersKernel,
    scaledDepthImage,
depthImageInput, scalingFactor)
  .add(ImagingOps::bilateralFilter,
    pyramidDepths[0],
scaledDepthImage,
    gaussian, eDelta, radius)
  .mapAllTo(deviceMapping);
```

- Users create Task Graphs with our OpenCL API.

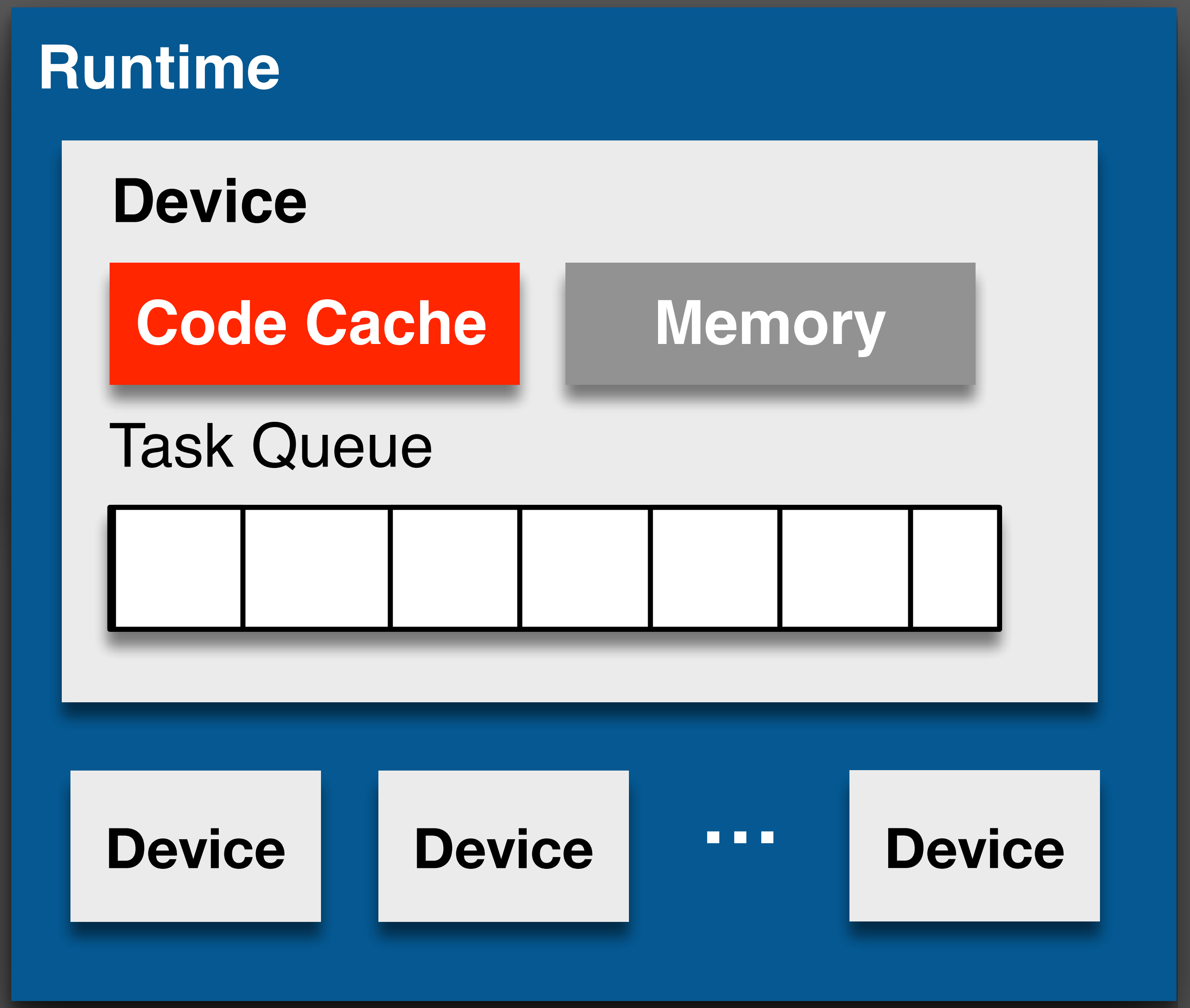
Task Graph

Graph Optimizer



- The compiler expands graphs to include data movement.
- Graph is optimized to remove redundant data transfers.

Optimized Graph



- Runtime schedules tasks on devices.

Platform | FPGA Acceleration

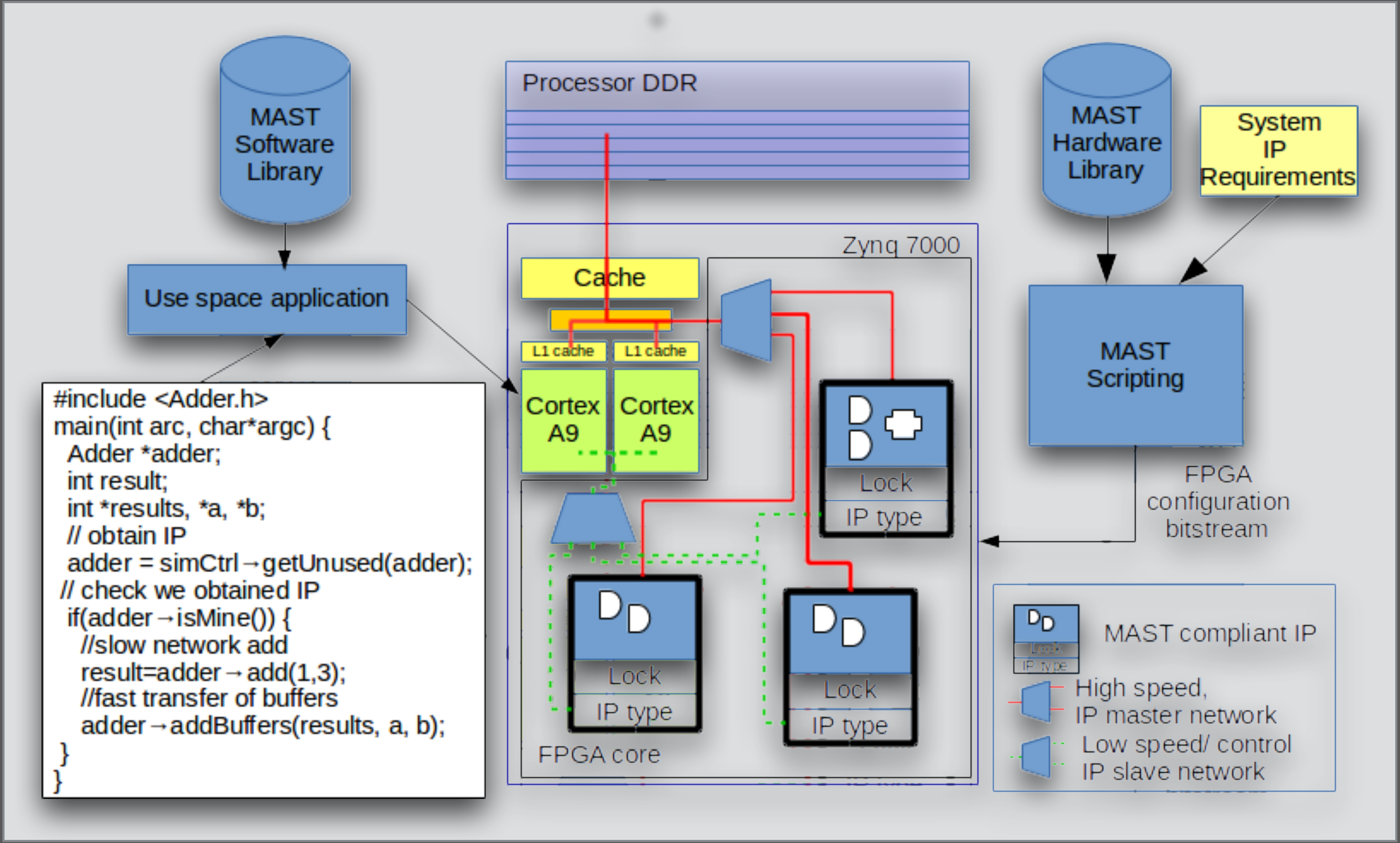
MANCHESTER 1824

The University of Manchester

- MAST: Modular Accelerator and Simulation Technology
- HW/SW Libraries for FPGA Acceleration (C++, BlueSpec)
- Thread, Process, OS Concurrency
- Acceleration through IP Blocks
- Simulation [2] via MAMBO [3] Dynamic Binary Instrumentation
- Currently Implemented in ARMv7 Xilinx Zynq SoC

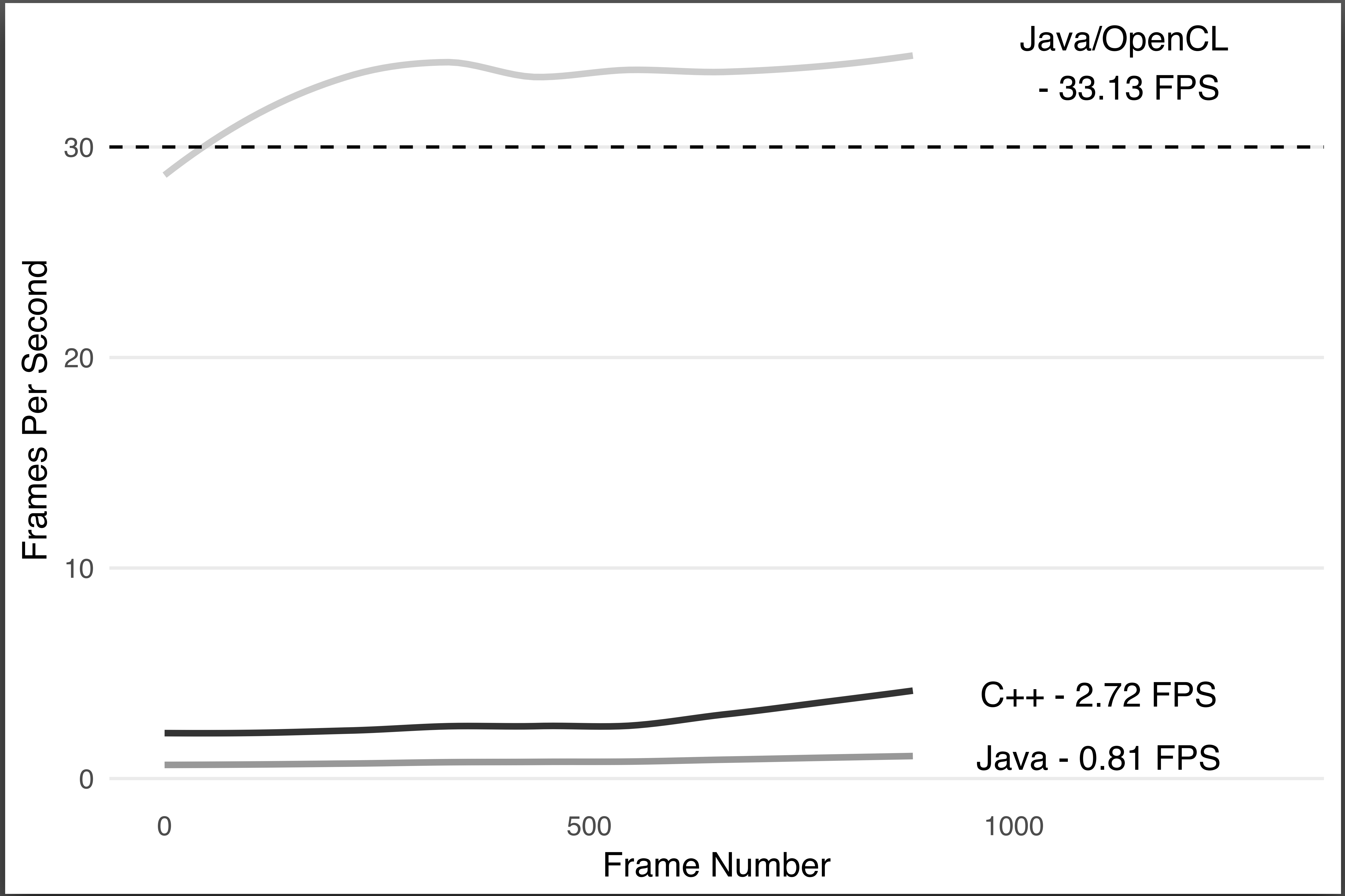
[2] John Mawer, Oscar Palomar, Cosmin Gorgovan, Andy Nisbet, Will Toms and Mikel Lujan.
The potential of dynamic binary modification and CPU/FPGA SoCs for simulation*,* In FCCM 2017.
[3] Cosmin Gorgovan, Amanieu d'Antras, and Mikel Luján.
MAMBO: A low-overhead dynamic binary modification tool for ARM. ACM TACO. 13, 1, Article 14 (April 2016).

Platform I FPGA Acceleration



Results I OpenCL Acceleration

The University of Manchester



OpenJDK/Graal, Intel Xeon E5-2620@GHz, 32GB RAM, 24 HTs, NVIDIA Tesla K20m, OpenCL 1.2

Results | FPGA Acceleration

- Targeting preprocessing stage
  - Image scaling from mm to meters
  - Bilateral filter to produce a filtered scaled image
- Merging of two kernels

VM	No FPGA Acceleration	With FPGA Acceleration	Speedup
Maxine VM	2.20	0.05	43x
OpenJDK	0.66	0.03	22x

OpenJDK/Maxine,Xilinx Zynq 706, ARMv7 Cortex A9, 1GB RAM, MAST FPGA

Conclusions

- Future Challenges of Computing
  - Performance | Power | Programmability | +++
  - Is there a Magic Bullet?
- VM Approach
  - “Write-once-run-everywhere”
  - Extend VMs for the heterogeneous world
- Our Approach
  - Demonstrate feasibility with proof-of-concept
  - Target both Industrial and Research VMs
  - High Performance through OpenCL/FPGA Acceleration
  - Use complex CV application as a driver
- Future Work
  - Device Diversity
  - Join Mast and OpenCL Accelerator
  - Extend API through more real-world use cases

MANCHESTER
1824
The University of Manchester

Thank you!

Questions?

christos.kotselidis@manchester.ac.uk

https://github.com/beehive-lab