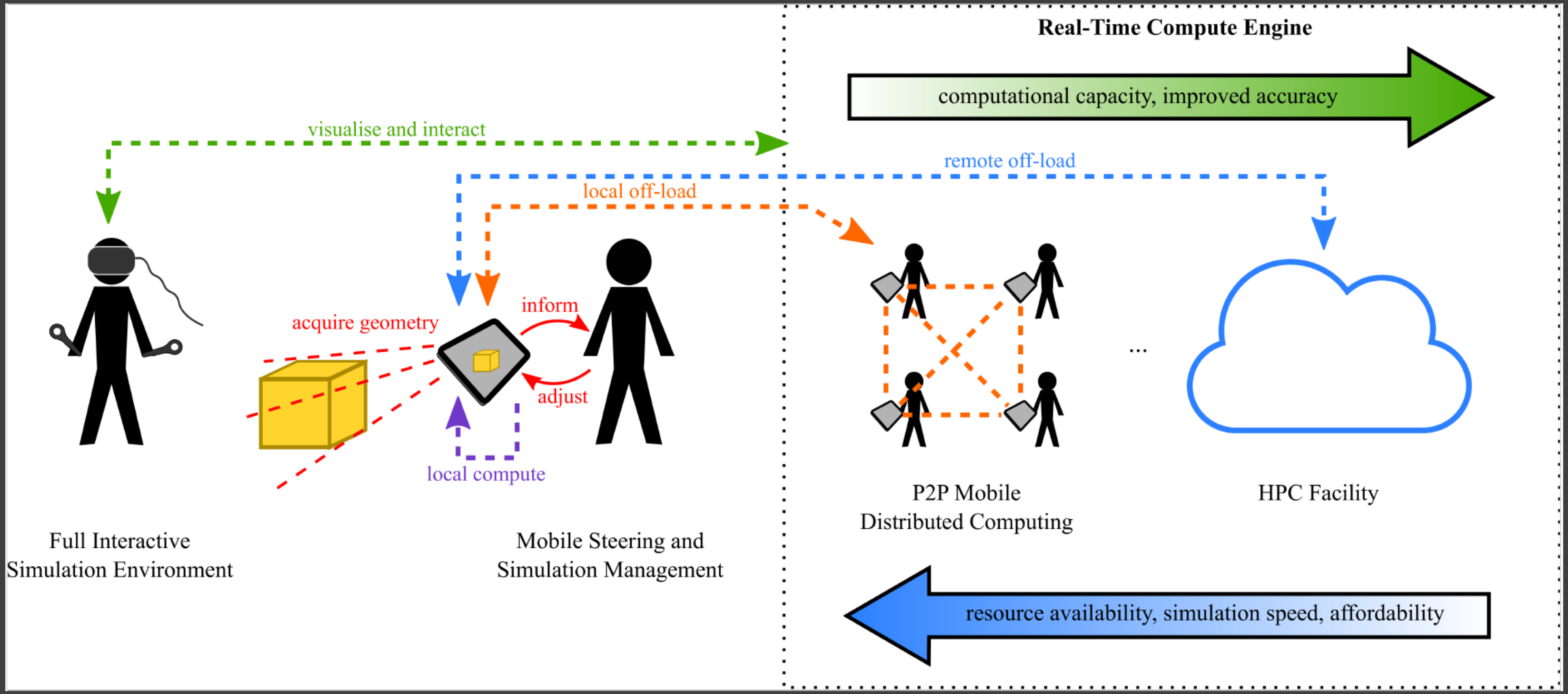
Using Mobile Devices for Engineering Simulation

Adrian R.G. Harwood
School of Mechanical, Aerospace & Civil Engineering
The University of Manchester, UK

Long-Term Research Question

- How can we use accessible, affordable distributed computing resources to provide accelerated, on-demand, interactive modelling and simulation for a wide range of applications?

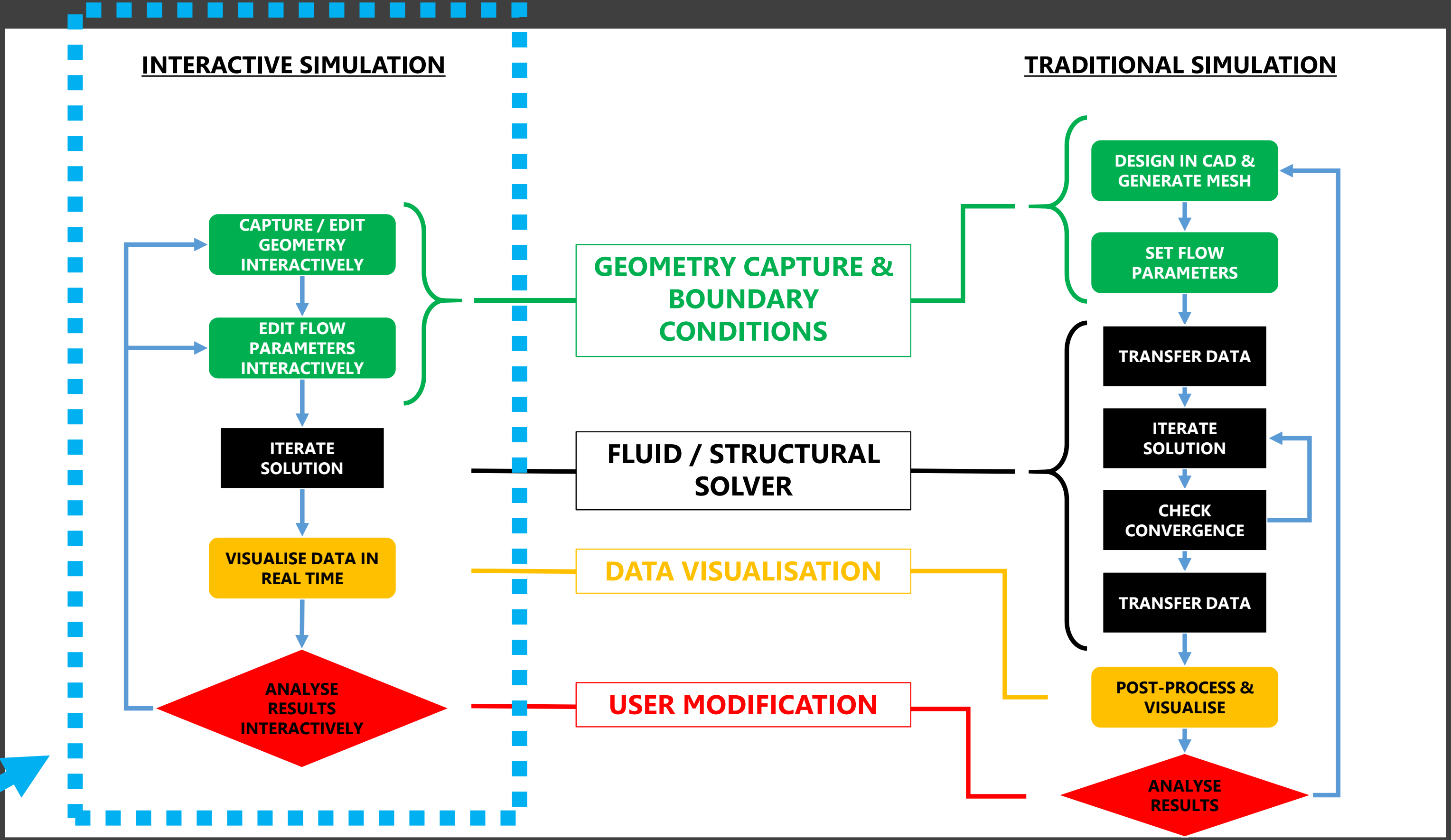
Work So Far



- Proposal of an interactive simulation infrastructure which extends existing capabilities to new use-modes[1].
- Affordable modelling and simulation "on-the-go".

[1] Adrian R.G. Harwood, Petra Wenisch, and Alistair J. Revell. A Real-Time Modelling and Simulation Platform for Virtual Engineering Design and Analysis. In *Proceedings of 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7), 11-15 June 2018, Glasgow, UK*. ECCOMAS, 2018.

Short-Term Aim

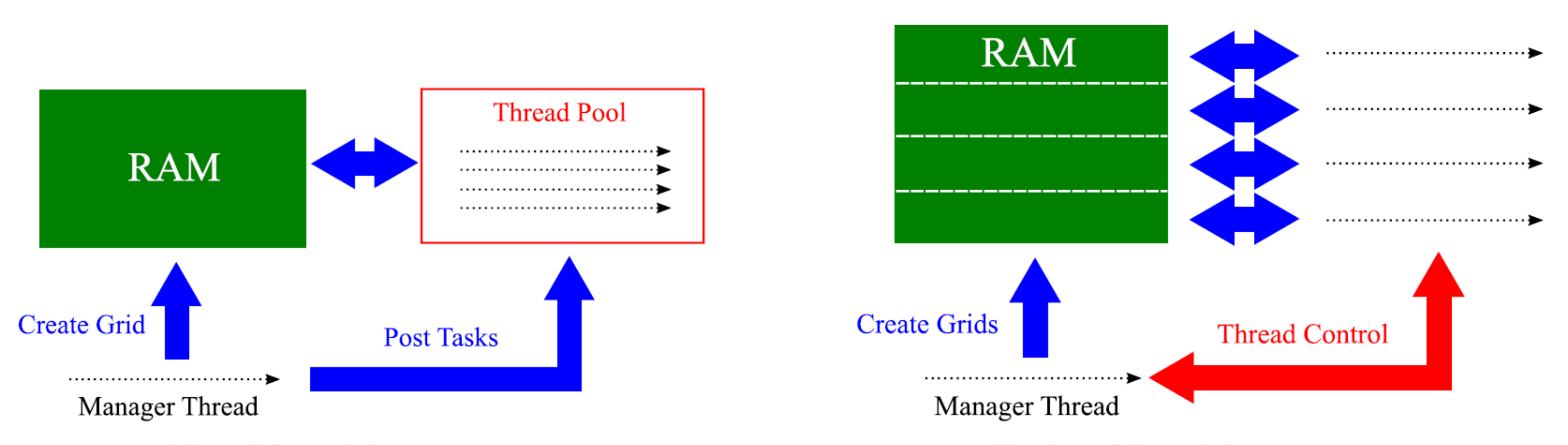


Interactive simulation in particular

- How best can we leverage a mobile cluster through new or existing software frameworks to provide feasible, scalable simulation?
- What practical potential do these devices offer given appropriate algorithms?

The University of Manchester

Mobile CPUs

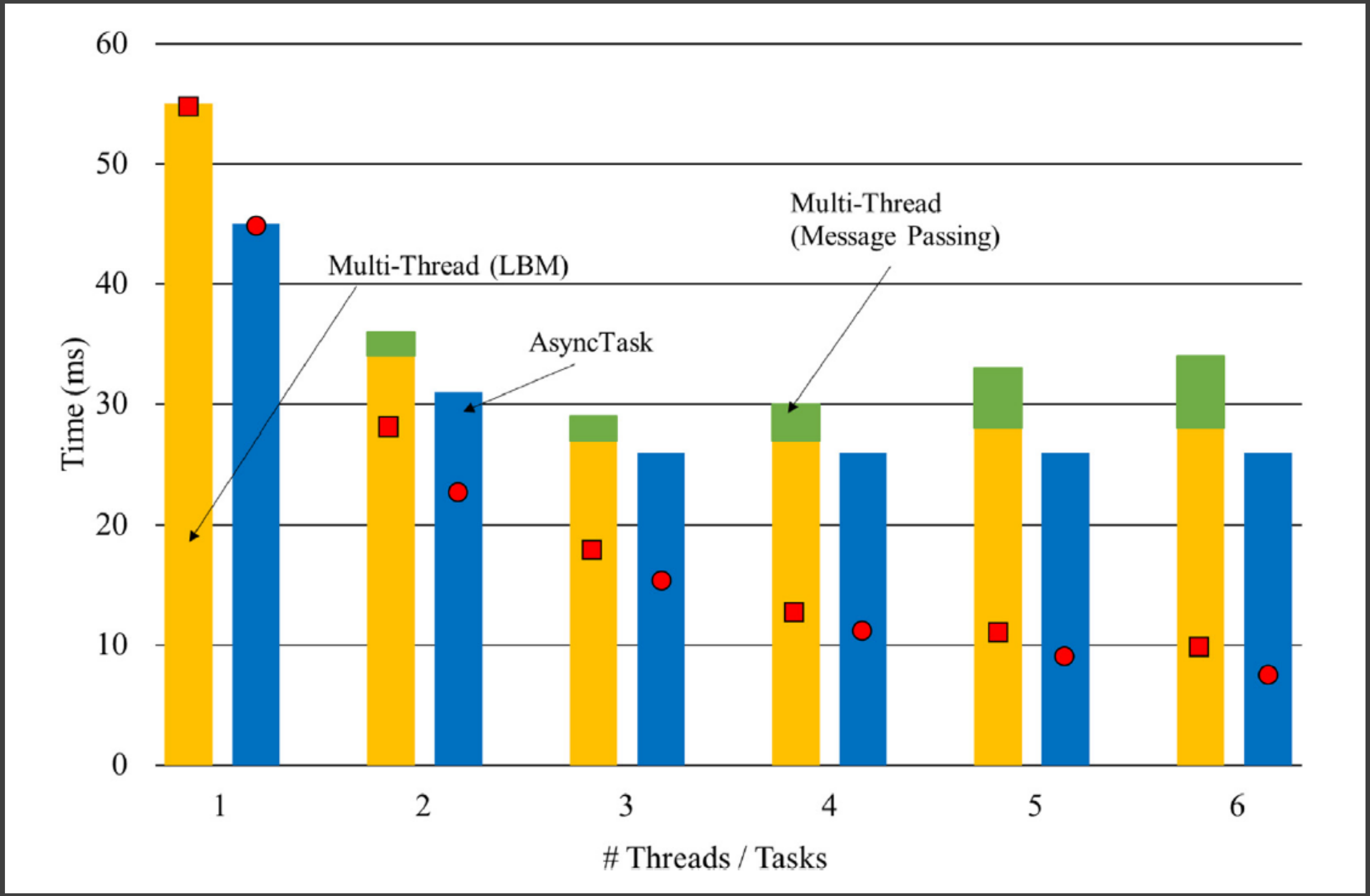


(a) Task-based design concept
(b) Thread-based design concept

- Development of CPU parallelisation strategies suitable for Android-based mobile devices.
- Implementation of interactive CFD simulation based on Lattice-Boltzmann Method using mobile hardware[2].

[2] Adrian R.G. Harwood and Alistair J. Revell. Parallelisation of an interactive lattice-Boltzmann method on an Android-powered mobile device. Advances in Engineering Software, 104(1):38–50, 2017.

Mobile CPUs



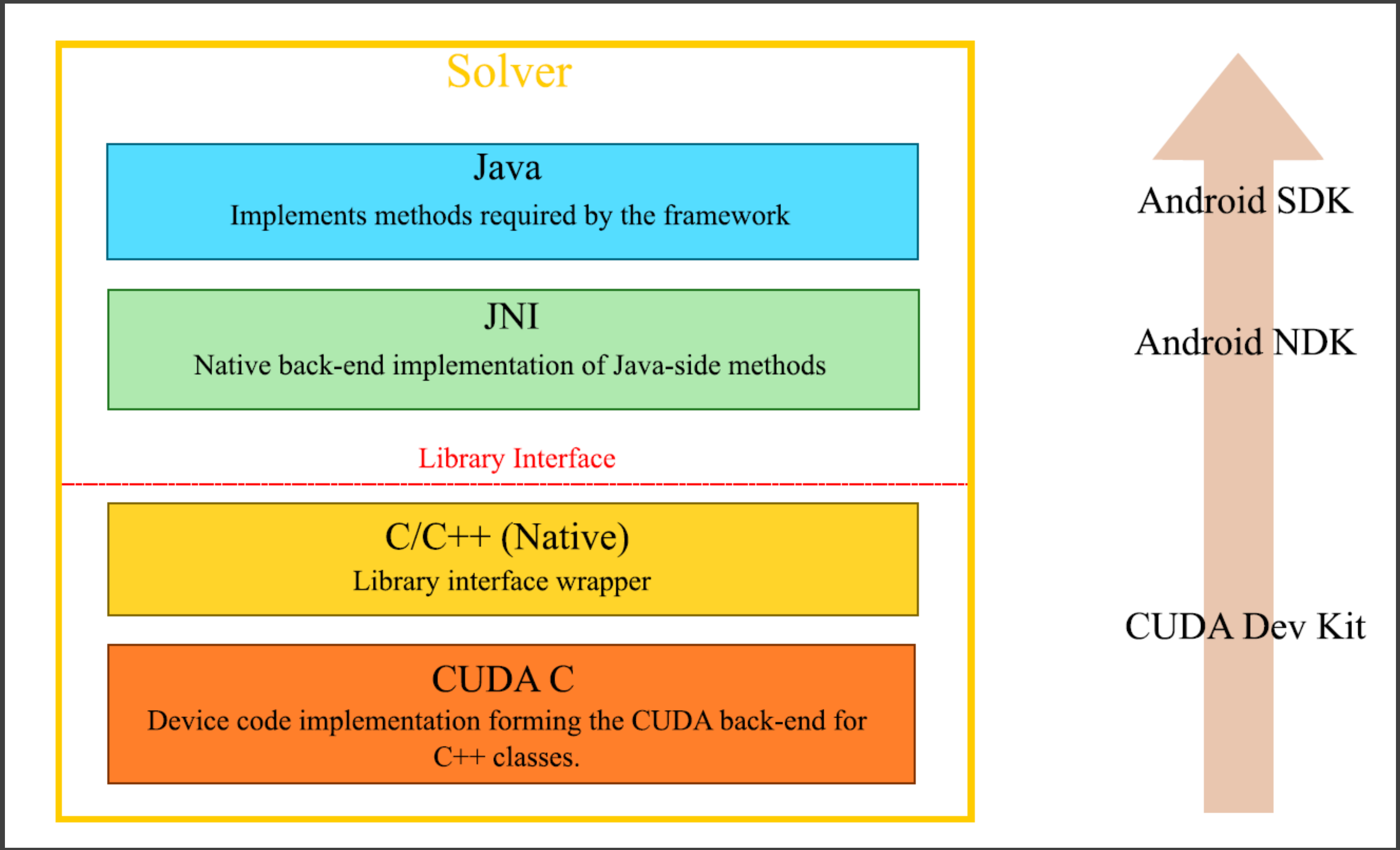
[2] Adrian R.G. Harwood and Alistair J. Revell. Parallelisation of an interactive lattice-Boltzmann method on an Android-powered mobile device. Advances in Engineering Software, 104(1):38–50, 2017.

- Demonstrates good scalability but very limited throughput on mobile CPU driven by power-efficient design ~1 MLUPS peak performance[2].

The University of Manchester

MANCHESTER 1824

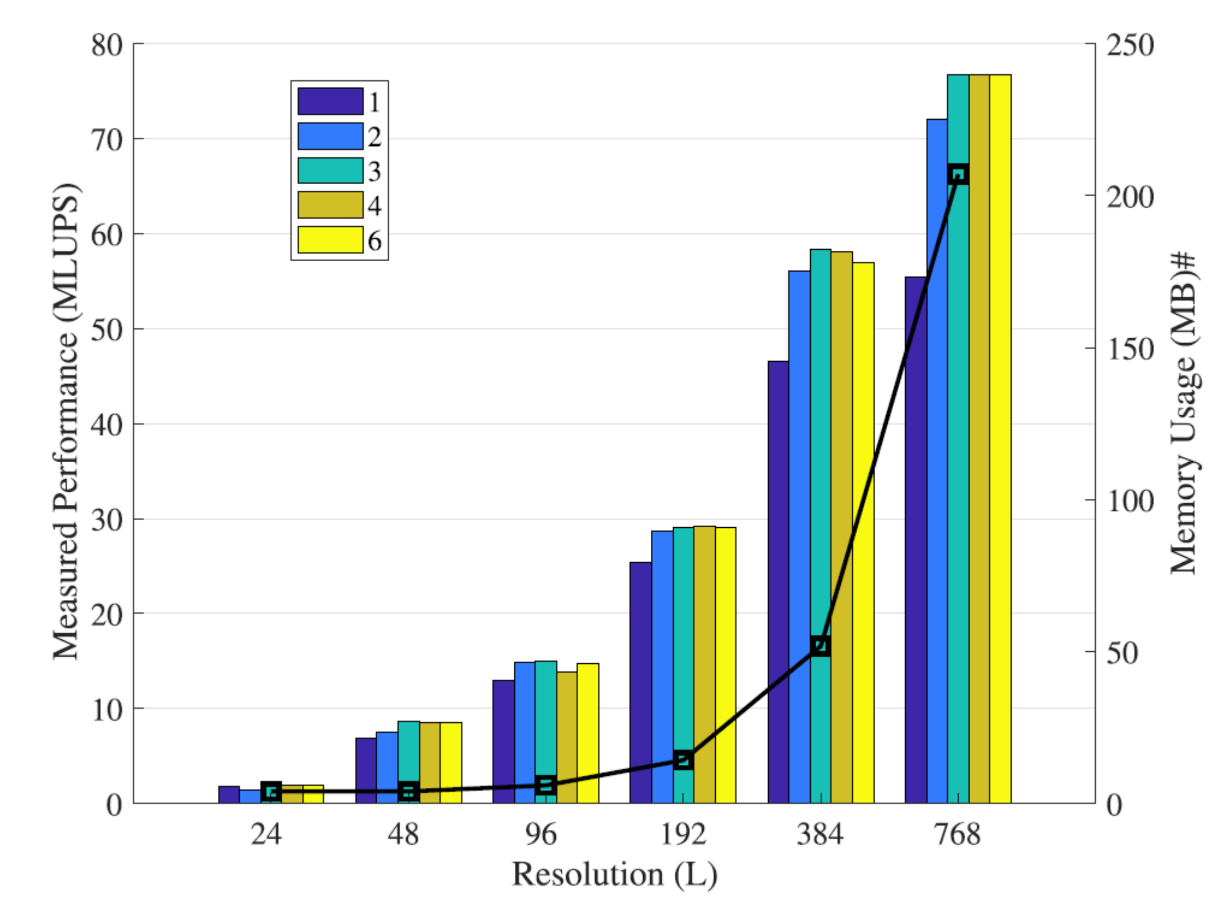
Mobile GPUs



[3] Adrian R.G. Harwood and Alistair J. Revell. Interactive flow simulation using Tegra-powered mobile devices. Advances in Engineering Software, 115(Supplement C):363 – 373, 2018.

- Development of CUDA-based GPU implementation of interactive CFD simulation based on Lattice-Boltzmann Method[3].

Mobile GPUs



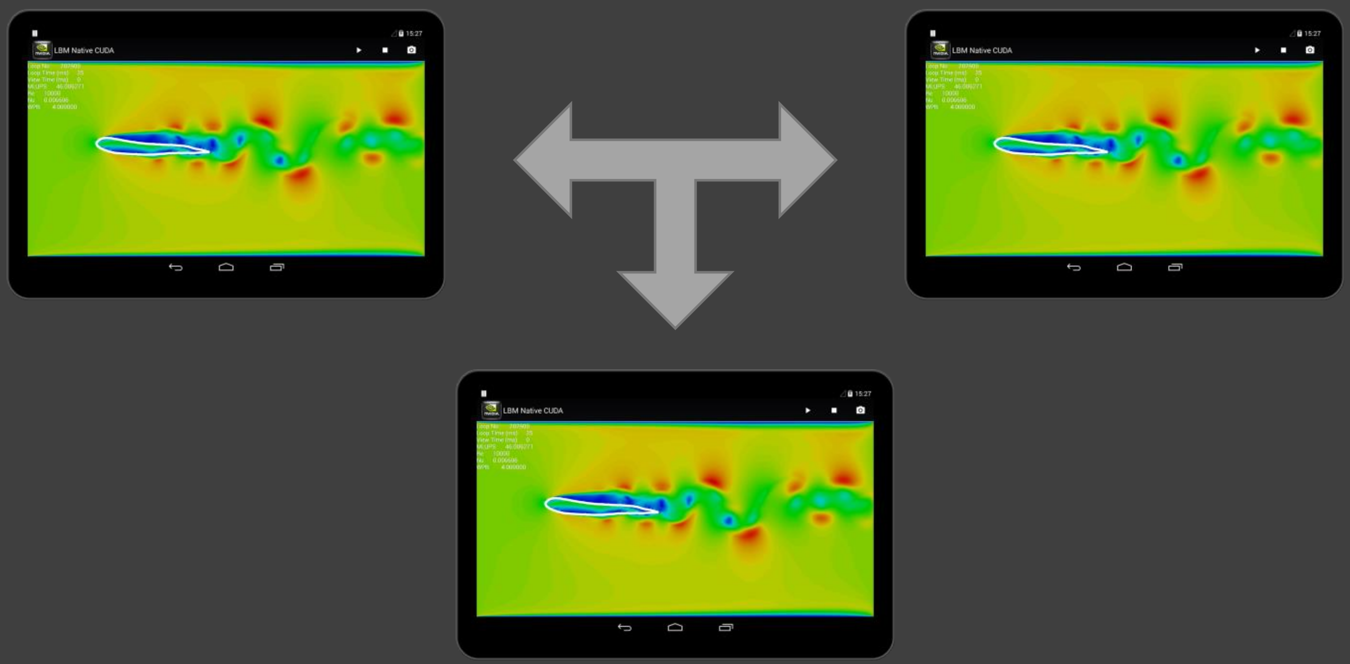
- Scales very well and GPU parallelisation offered 13.4x throughput and 22.5x better power consumption of for the CPU test case[3].

[3] Adrian R.G. Harwood and Alistair J. Revell. Interactive flow simulation using Tegra-powered mobile devices. Advances in Engineering Software, 115(Supplement C):363 – 373, 2018.

MANCHESTER 1824

The University of Manchester

Lots of mobile GPUs

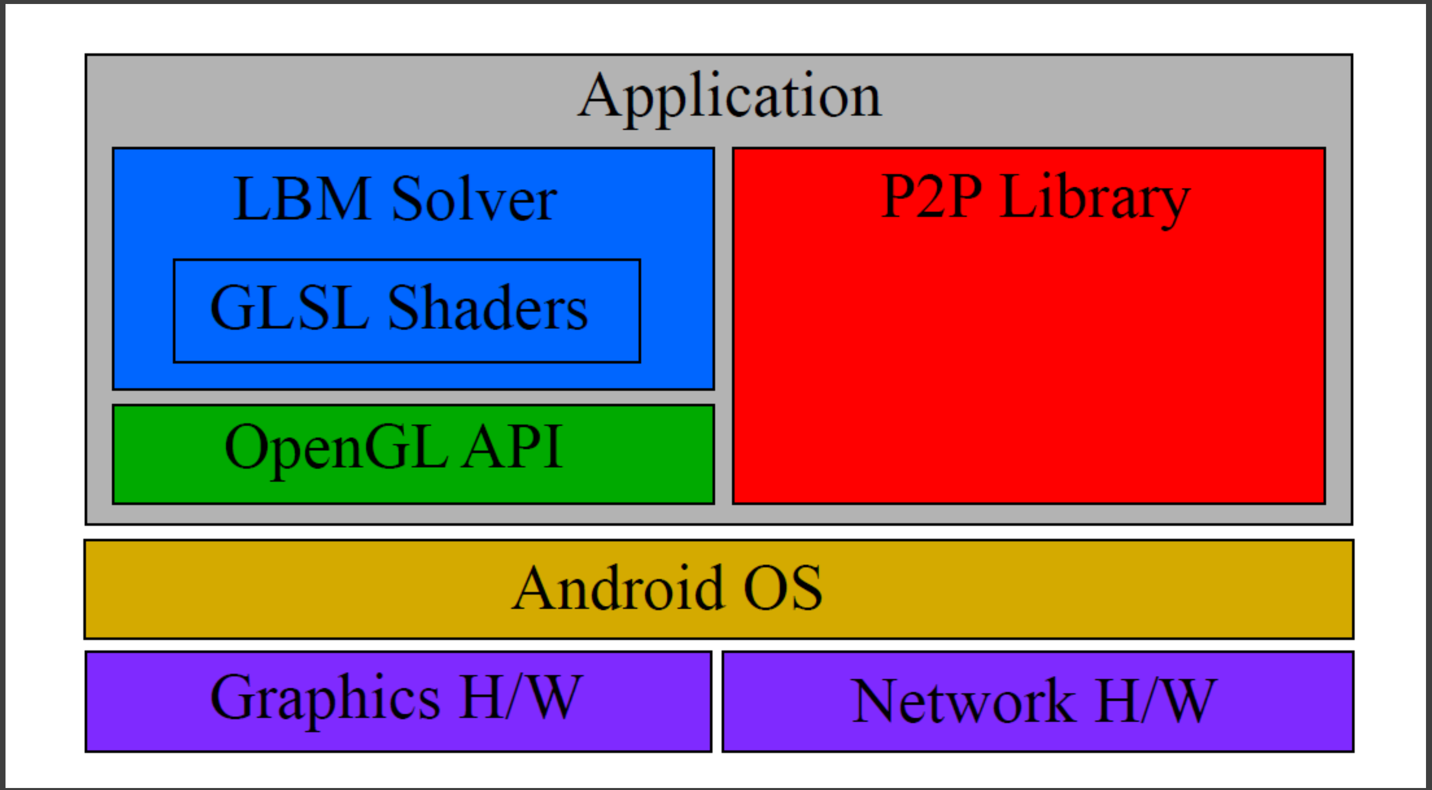


- Improve generality of implementation by increasing hardware support.
- Explore inter-device communication capabilities and develop suitable strategies.
- Determine scalability of solution for more practical applications.

Approach

- OpenGL-based implementation of LBM solver using compute shaders (since OpenGL 4.3, OpenGL ES 3.1).
  - Not restricted to NVIDIA hardware.
  - Recent standards offer better support for GPGPU computing that they used to.
  - Close integration with graphics makes coupling with visualisation more intuitive.
- Inter-device communication using WiFi-Direct Peer-to-Peer networking.
  - Highest bandwidth and range compared with Bluetooth and NFC.
  - P2P limits "server" load.

Architecture

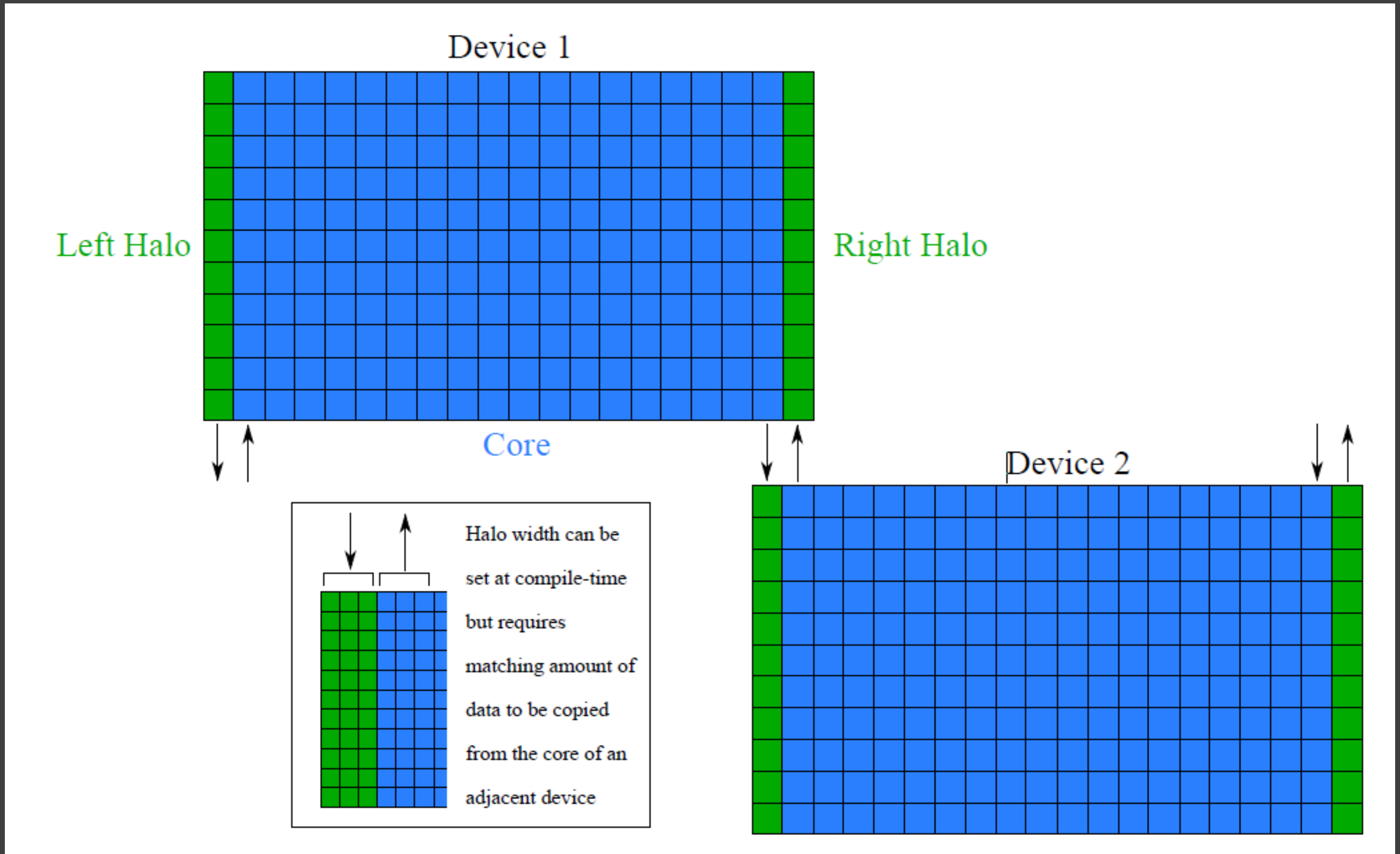


- Split into two main packages[4].
- P2P Library developed for generic application with defined interface exposed to LBM solver.
- LBM Solver built around OpenGL API.

[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

The University of Manchester

Architecture



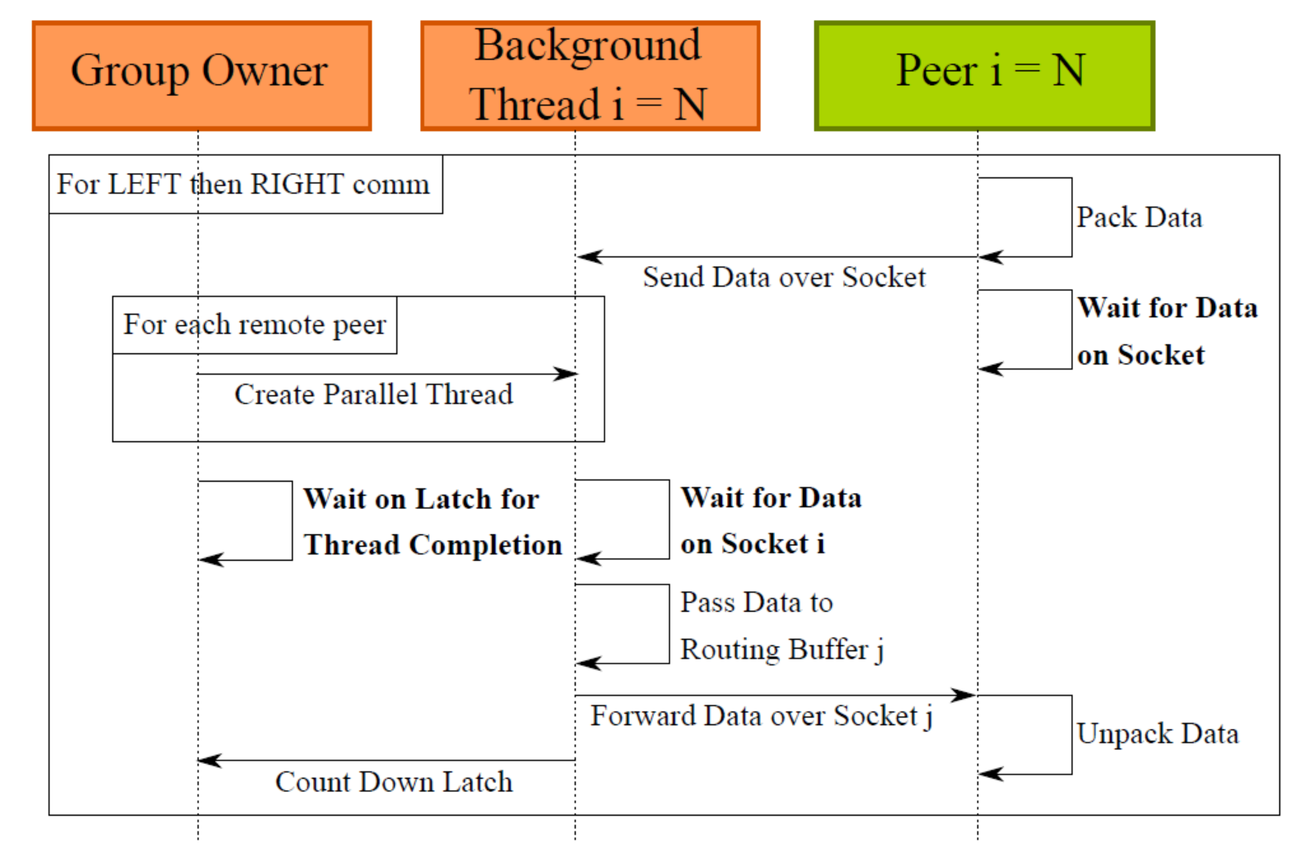
[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

- Solver grids have a halo region for passing data.
- Halo width can be set at compile-time.
- Allows *communication frequency* to traded for *amount of data to be passed*.

MANCHESTER 1824

The University of Manchester

Initial (Naïve) Comm Design



- P2P communication relies on WiFiP2pManager components of the Android SDK.
- Principally based on multi-threaded, asynchronous observer patterns.
- Data passed using Java Sockets.

[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

The University of Manchester

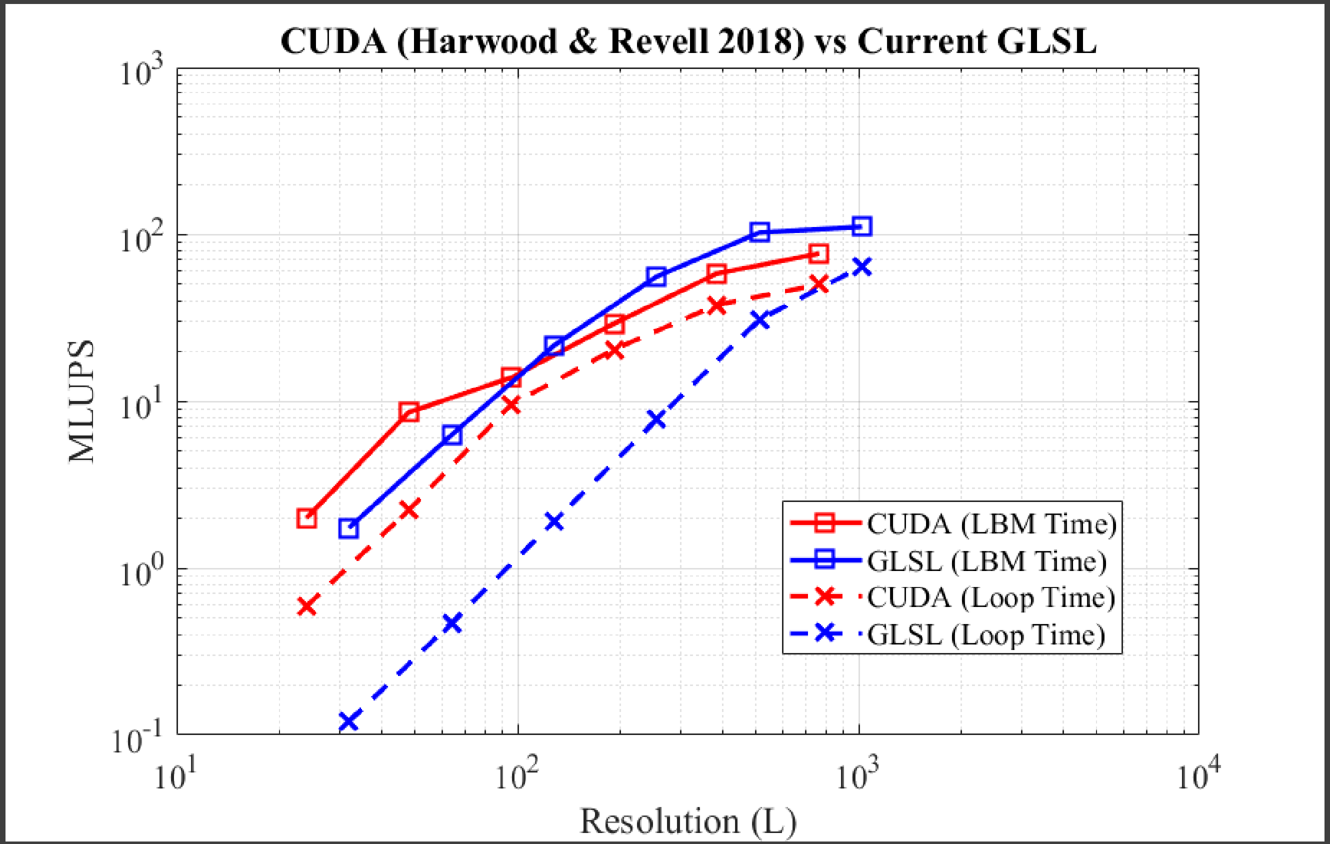
MANCHESTER 1824

Testing

1. Compare single-device performance to CUDA implementation.
2. Compare multi-device performance to single-device performance.
3. Examine scalability of proposed communication algorithm.

The University of Manchester

Single Device Performance



- In CUDA implementation[3], compute and drawing handled by separate threads – shorter loop time.
- In current work, two-pass render technique used on single GLThread.

[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

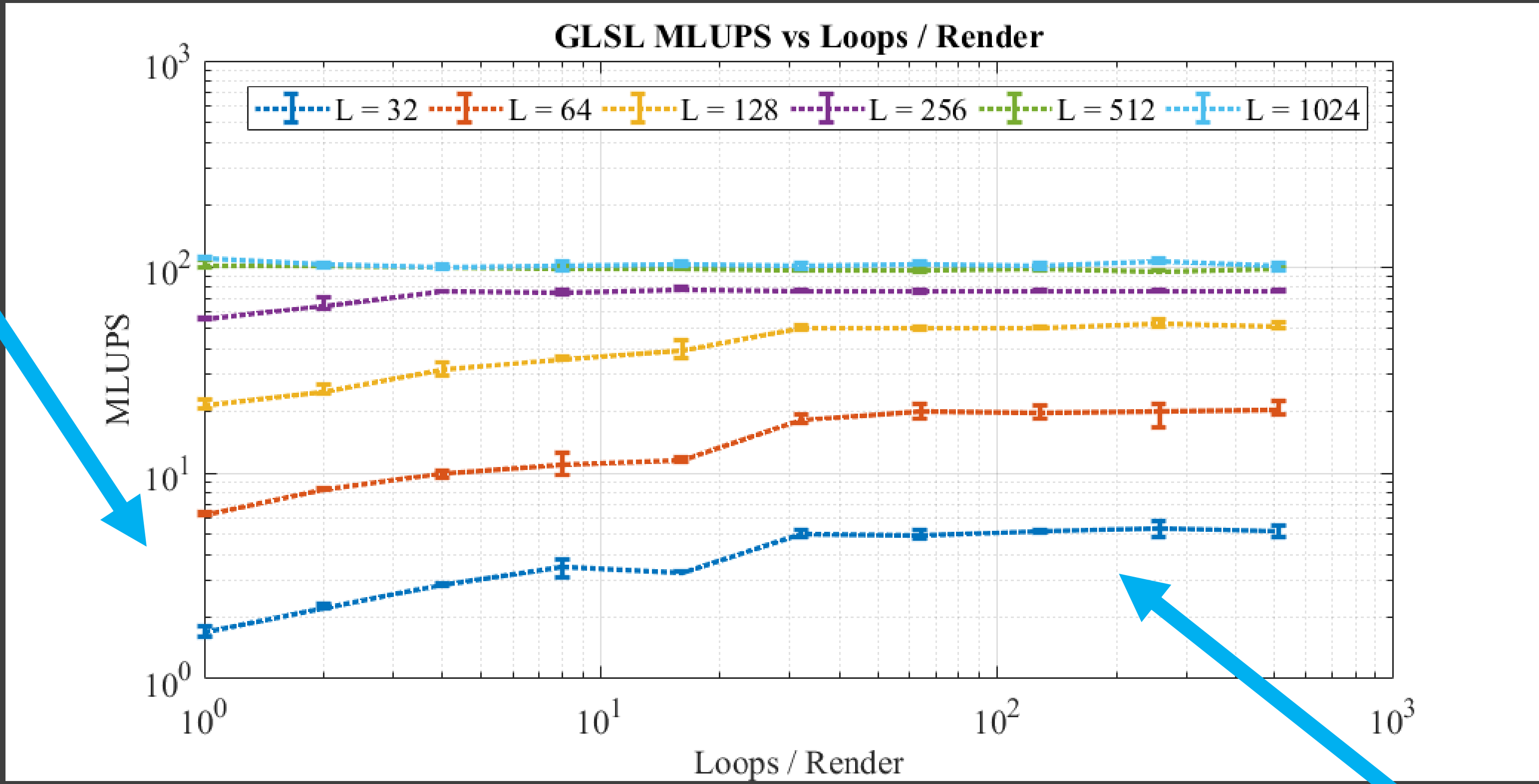
[3] Adrian R.G. Harwood and Alistair J. Revell. Interactive flow simulation using Tegra-powered mobile devices. Advances in Engineering Software, 115(Supplement C):363 – 373, 2018.

MANCHESTER 1824

The University of Manchester

Single Device Performance

Limited by rendering



Limited by LBM

- Previous work updated display every iteration.
  - In reality only need to ensure 24fps.
- Effect on throughput is not insignificant.
- Clear evidence of GPU saturation.

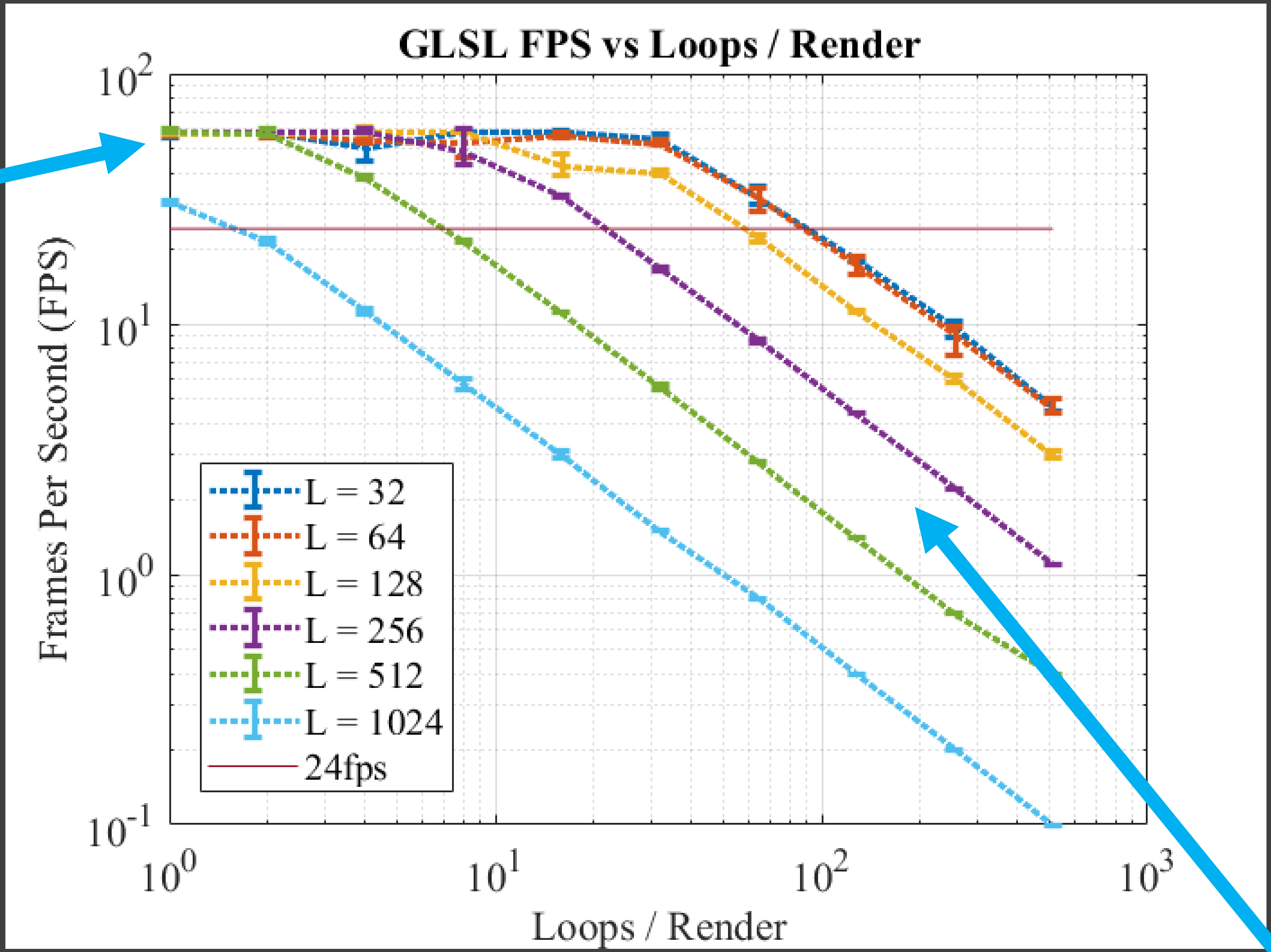
[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

The University of Manchester

MANCHESTER 1824

GLThread max frequency

Single Device Performance



LBM load dominates

[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

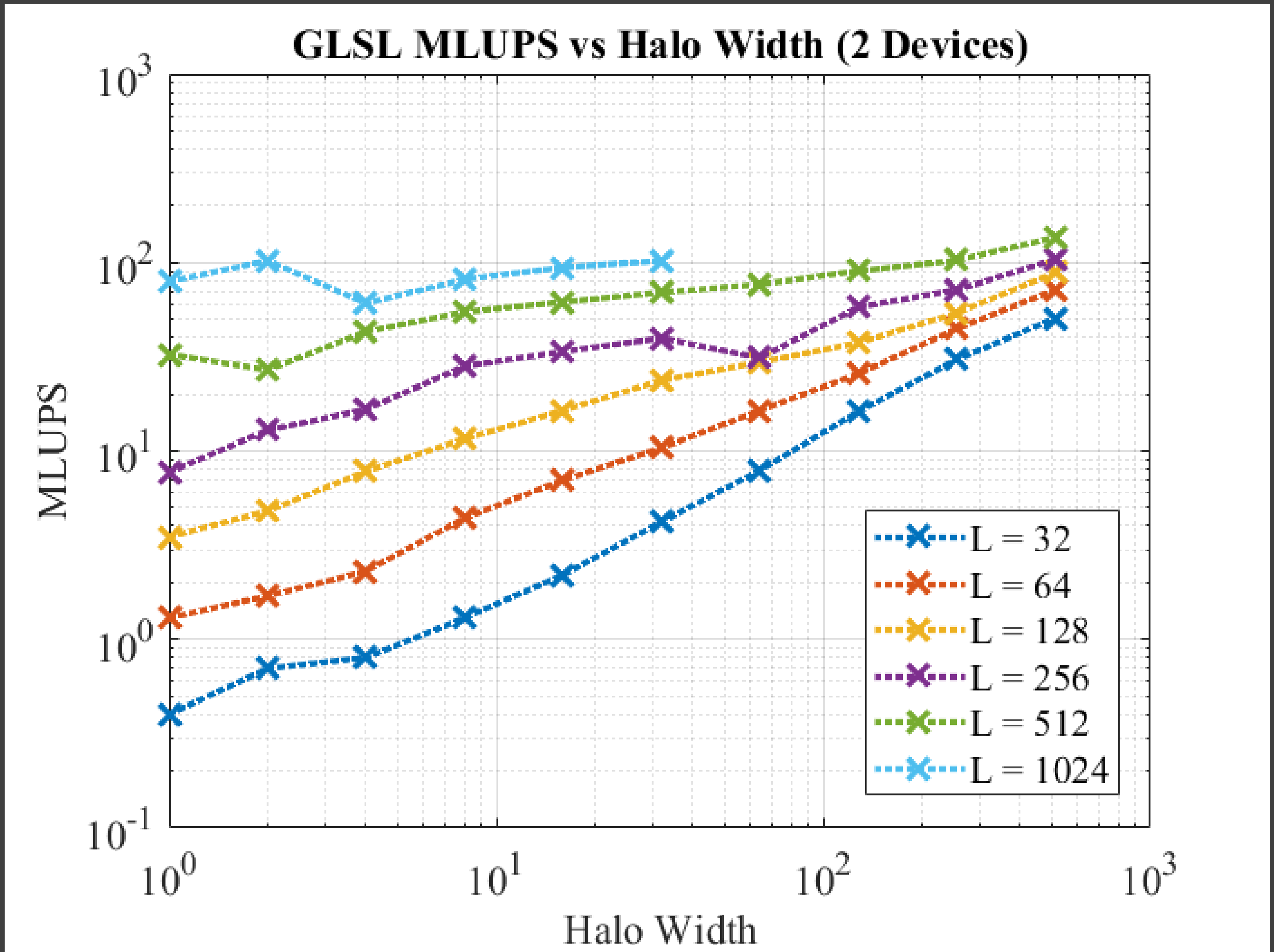
- GLThreads are restricted by the OS to run at 60 Hz which limits potential at lower resolutions where it could run faster.

The University of Manchester

MANCHESTER 1824

Loops per Render chosen from serial test case

Multi-Device Performance

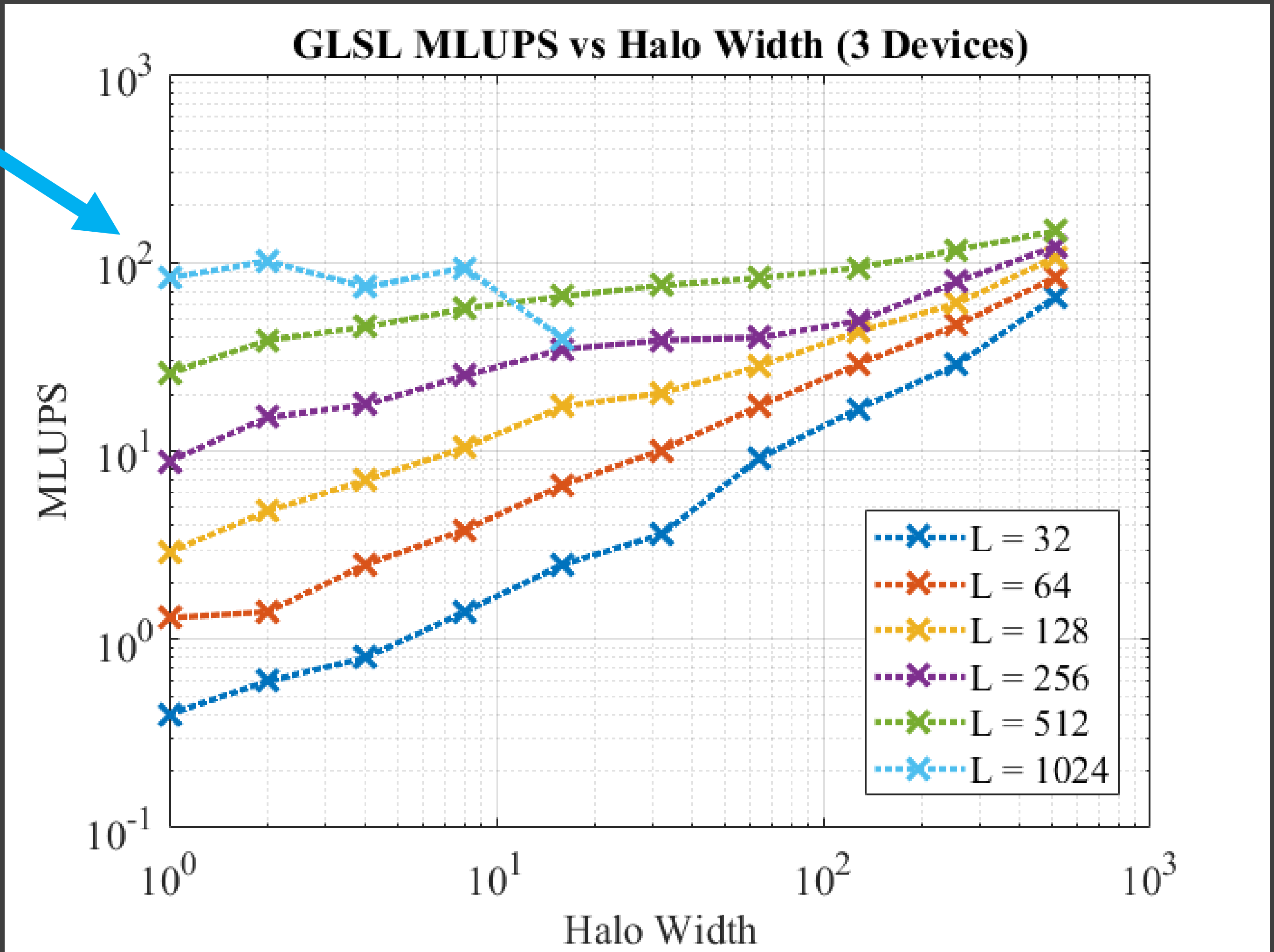


[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

- Small halo has small memory footprint but frequent exchange of small amounts of data.
- Throughput improved with infrequent comms.

Little benefit at the moment

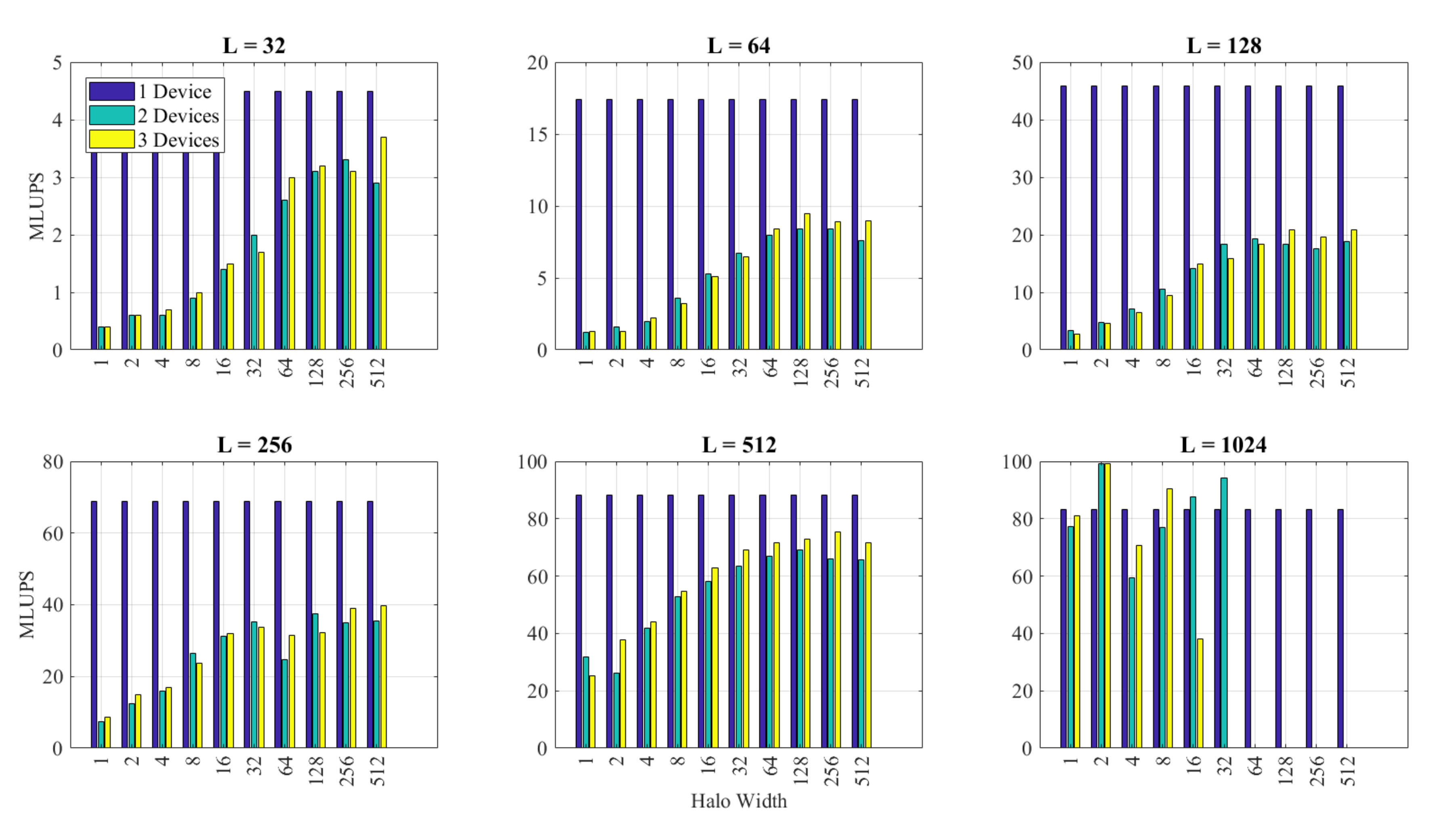
Multi-Device Performance



[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

- 2 Device and 3 Devices look very similar.
- Added cost of communication cancels out throughput improvement by adding devices.

Multi-Device Performance



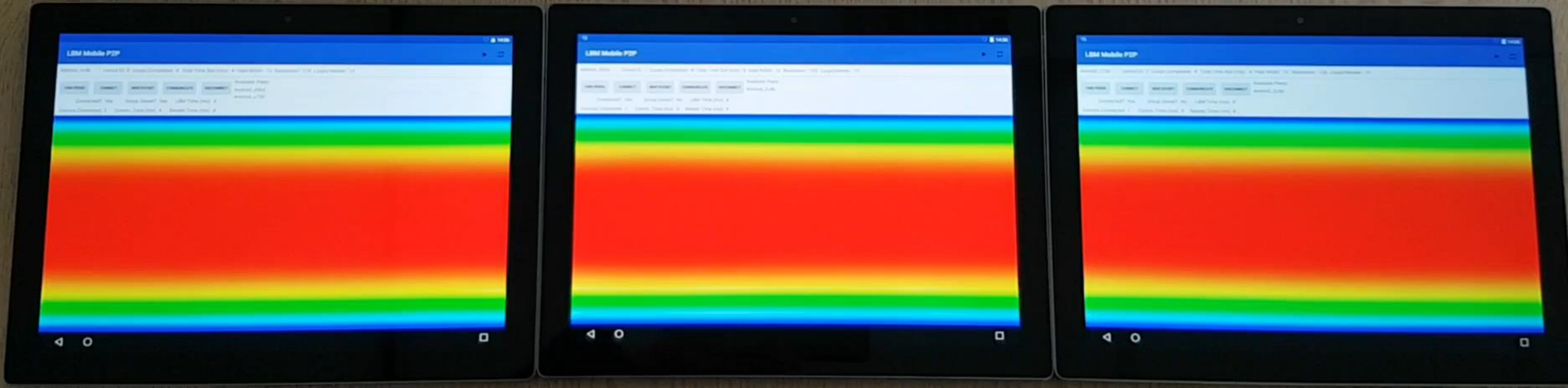
[4] Adrian R.G. Harwood. Interactive flow simulation using a peer-to-peer network of mobile GPUs, In Preparation.

- For larger cases, possible to recover serial performance even with unoptimized communication.

MANCHESTER 1824

The University of Manchester

- 2D channel flow is the base simulation.
- Touch response for adding geometry.
- Total calculation size ~110,000 cells spread across 3 devices.
- Android 8.0-powered Pixel C tablets.



Achievements

1. Generalisation of an efficient GPU-LBM implementation suitable for OpenGL ES and heterogenous devices.

2. Successful implementation of P2P communication strategy capable of proving the multi-device concept.

3. Established that even a unoptimised algorithm can maintain throughput of a single device as more devices are added.

Recognised Challenges

1. Realising an efficient GPU-GPU communication using a P2P configuration.
2. Implementing and establishing scalability over large numbers of devices.
3. Managing limitations of OpenGL ES vs Desktop GL and associated GLSL support.

Proposed Solutions

1. Realising an efficient GPU-GPU communication using a P2P configuration.
   - Use sockets between device pairs directly.
   - Don't buffer the data before/after sending/receiving.

2. Implementing and establishing scalability over large numbers of devices.
   - Explore heterogenous testing within research group.

3. Managing limitations of OpenGL ES vs Desktop GL and associated GLSL support.
   - Cannot really influence standard or vendor support for it but can consider alternatives such as Vulkan / OpenCL.

So should you use mobile devices?

- It depends…
  - Application
  - Numerical Methods Used
  - Speed Requirements
  - Accuracy Requirements
  - Hardware Suitability
  - Required Expertise to Implement
- Existing cluster computing solutions might be a better alternative e.g. Condor depending on your needs
- But…mobile is an available, affordable, low-power option with interaction to boot!