Details behind the new WDPM

Oluwaseun Sharomi & Raymond Spiteri

Department of Computer Science
University of Saskatchewan

March 19, 2014

Outline

WDPM FORTRAN Code

- The FORTRAN code is slow;

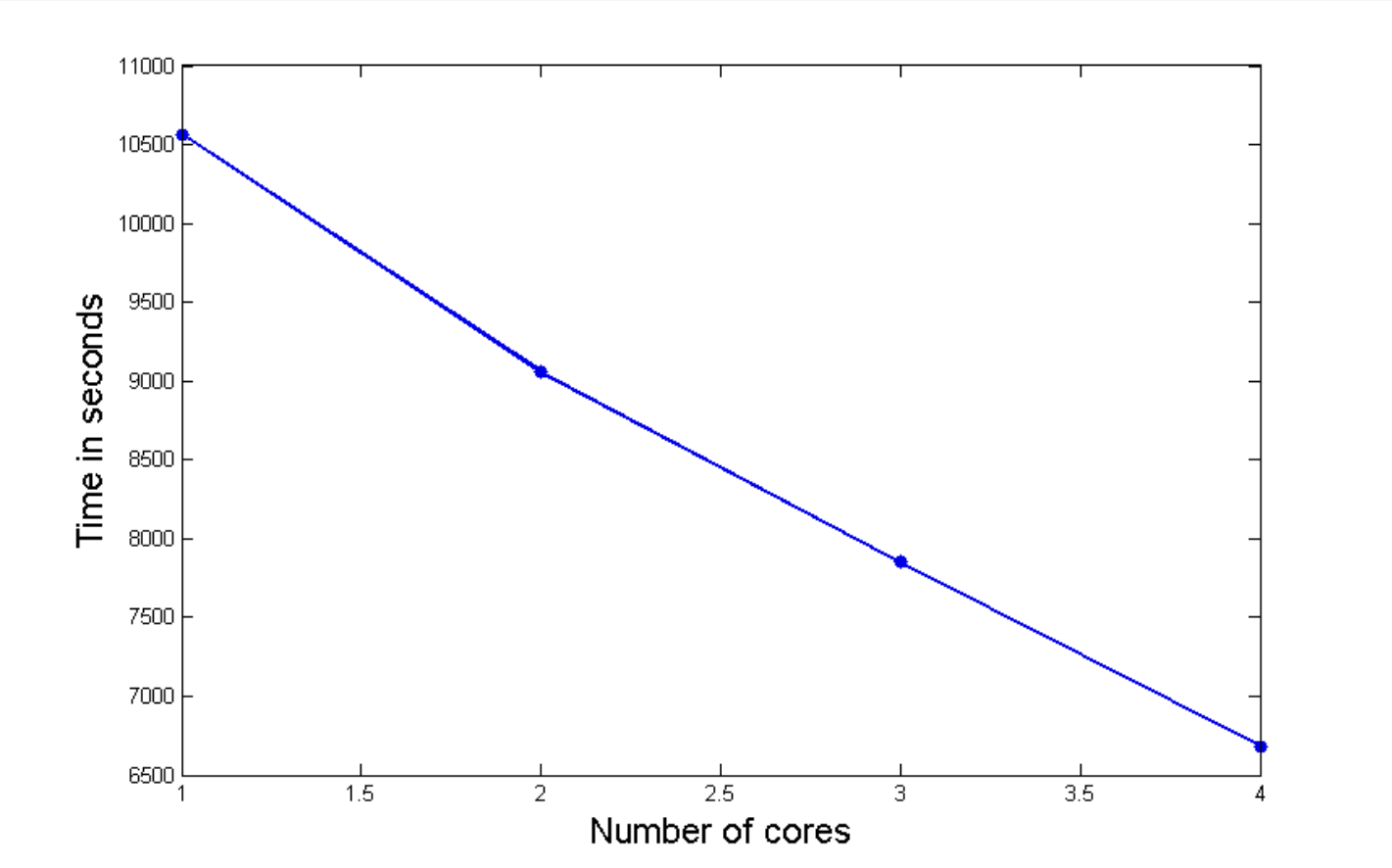

Figure: FORTRAN Timing

Goals

- Accelerate the process so that the code can run faster;
- A code that will run on a cheap computer (not necessarily Super Computers);
- A code that runs on devices with large number of cores.
- A code that runs on recent and very cheap GPUs;
- A code that is portable and does massive parallelism at the same time;
- Convert the FORTRAN code to OpenCL C code.
- Design a graphical user interface for the OpenCL C code.

Background: Acceleration Methods

- OpenMP (used for multi-core CPUs)
- MPI (used for clusters with many nodes)
- OpenCL (used for multi-core CPUs, GPUs and other accelerators)

Background: OpenCL

- Open Computing Language;
- is a framework for writing programs that execute across heterogeneous platforms;
- Some examples of platforms includes CPUs, GPUs and other processors;
- OpenCL provides parallel computing using task-based and data-based parallelism;
- OpenCL is an open standard maintained by Khronos Group;
- OpenCL is used to give non-graphical computing access for an application on a GPU.

Devices must pass the OpenCL conformance tests to support OpenCL. However, CPUs that supports at least SSE2 will do

	Version 2013 R3 Version 2014 Beta	Version XE 2013 R3
Supported Processors		
Intel® HD Graphics Family	✔	
Intel® Iris™ Graphics Family	✔	
Intel® Atom™ Processor	✔	
Intel® Core™ Processor	✔	
Intel® Xeon® Processor		✔
Intel® Xeon Phi™ Coprocessor		✔

Figure: Supported Intel Processor

Host Application Development

The host application requires six data structures

- Platform: cl_platform_id
- Device: cl_device_id
- Kernel: cl_kernel
- Program: cl_program
- Command queue: cl_command_queue
- Context: cl_context

- Platform: A platform is a data structure that identifies a vendor's implementation of OpenCL. Platforms make it possible to access devices;
- Device: Device receives kernels from the host;
- Kernel: A host application distributes kernels to devices.
- Program: The host selects kernels from a program.
- Command queue: Device receives kernels through a command queue.
- Context: An OpenCL context allows devices to receive kernels and transfer data.

OpenCL Device

- Examples include Intel CPU, Nvidia GPUs and others;
- An OpenCL device is made of compute-units;
- Each compute-unit is composed of a work-group;
- work-groups are collection of threads that can run in parallel.

Memory

The OpenCL device model identifies four address spaces but 2 are of interest to us.

- Global memory: Stores data for the entire device.
- Local memory: Stores data for the threads in a work-group.
- Threads can access local memory much faster (≈100x) than they can access global memory;

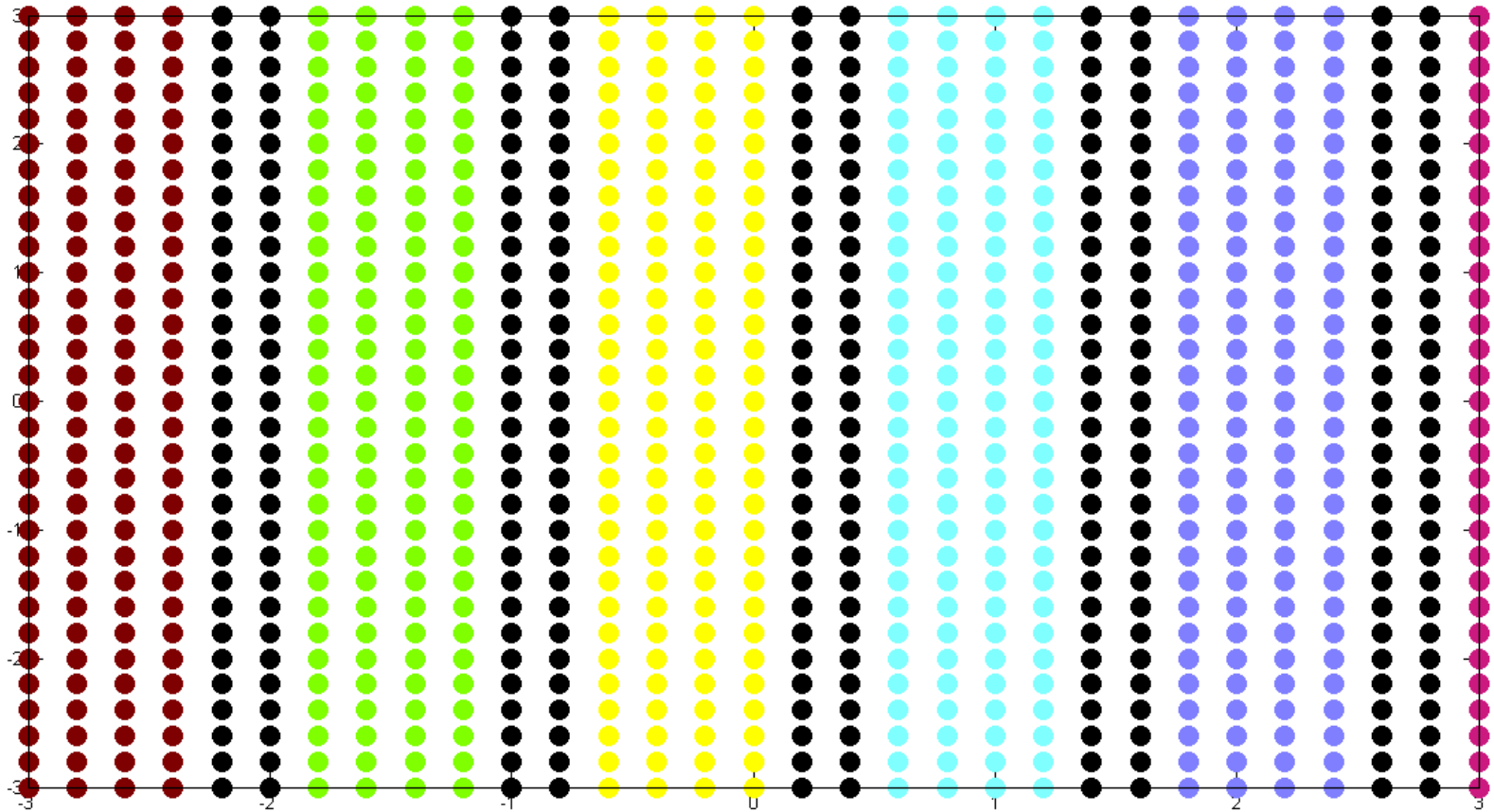

Figure: Fortran Implementation

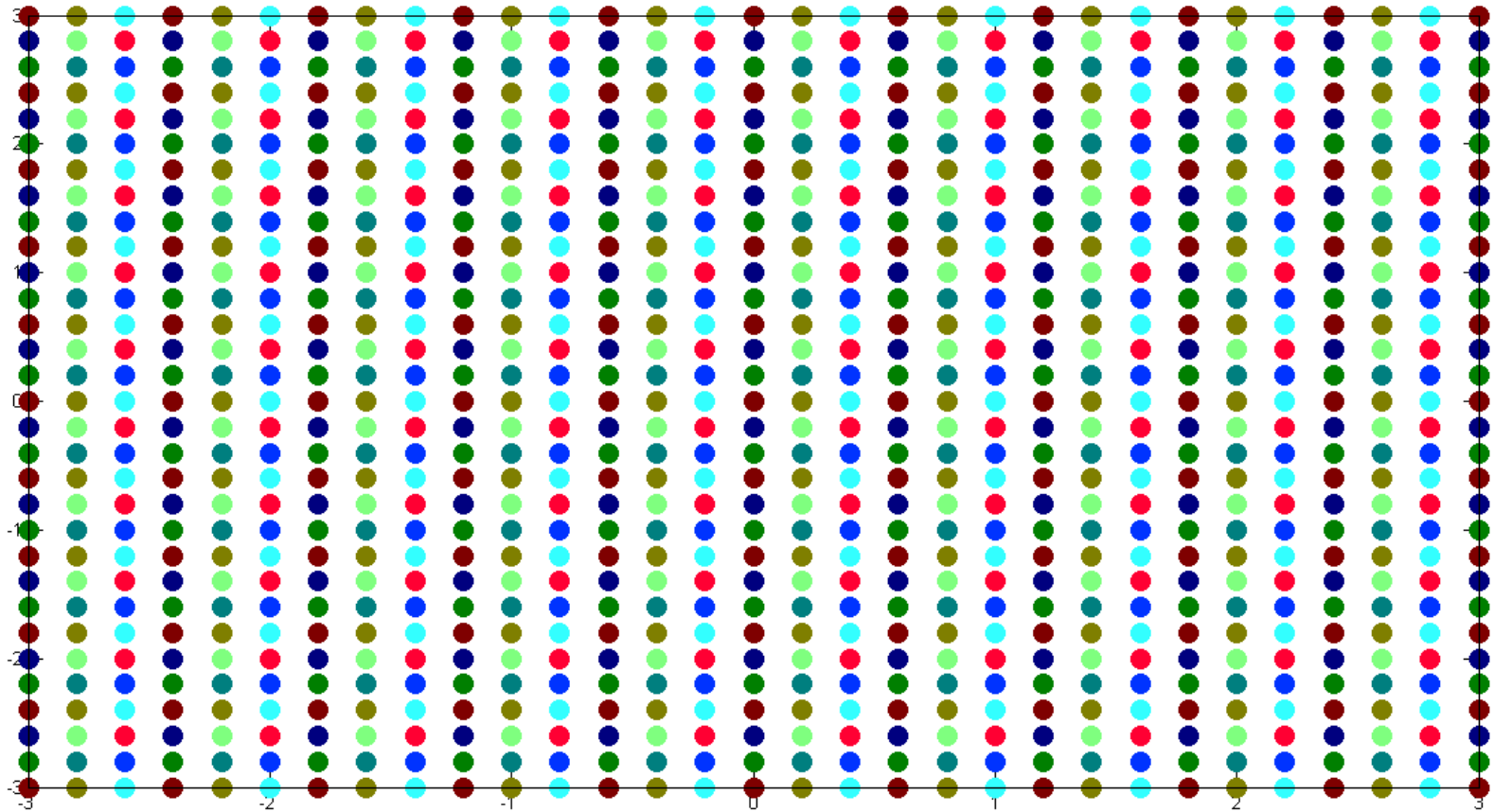

Figure: OpenCL Implementation

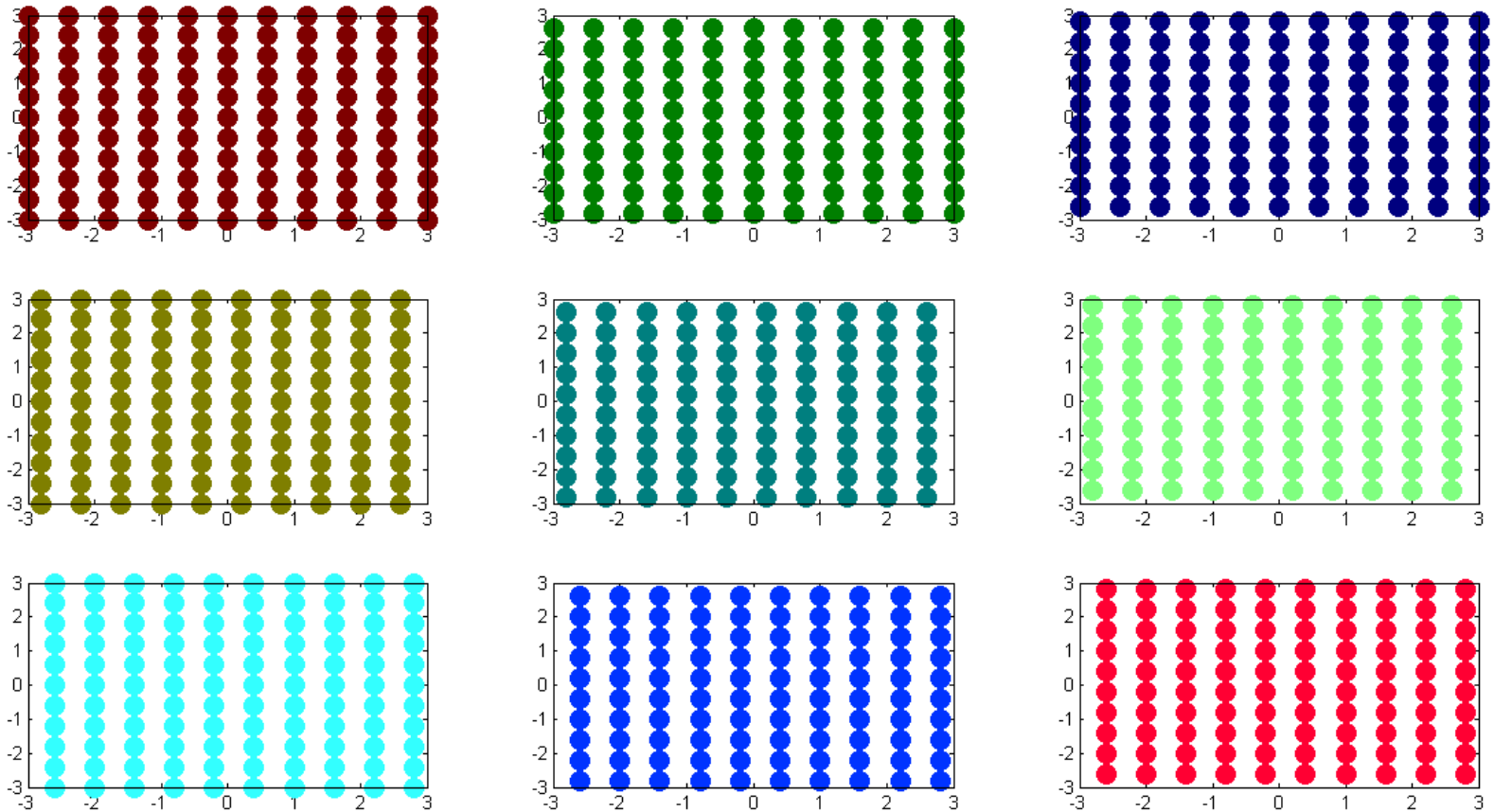

Figure: OpenCL Slices

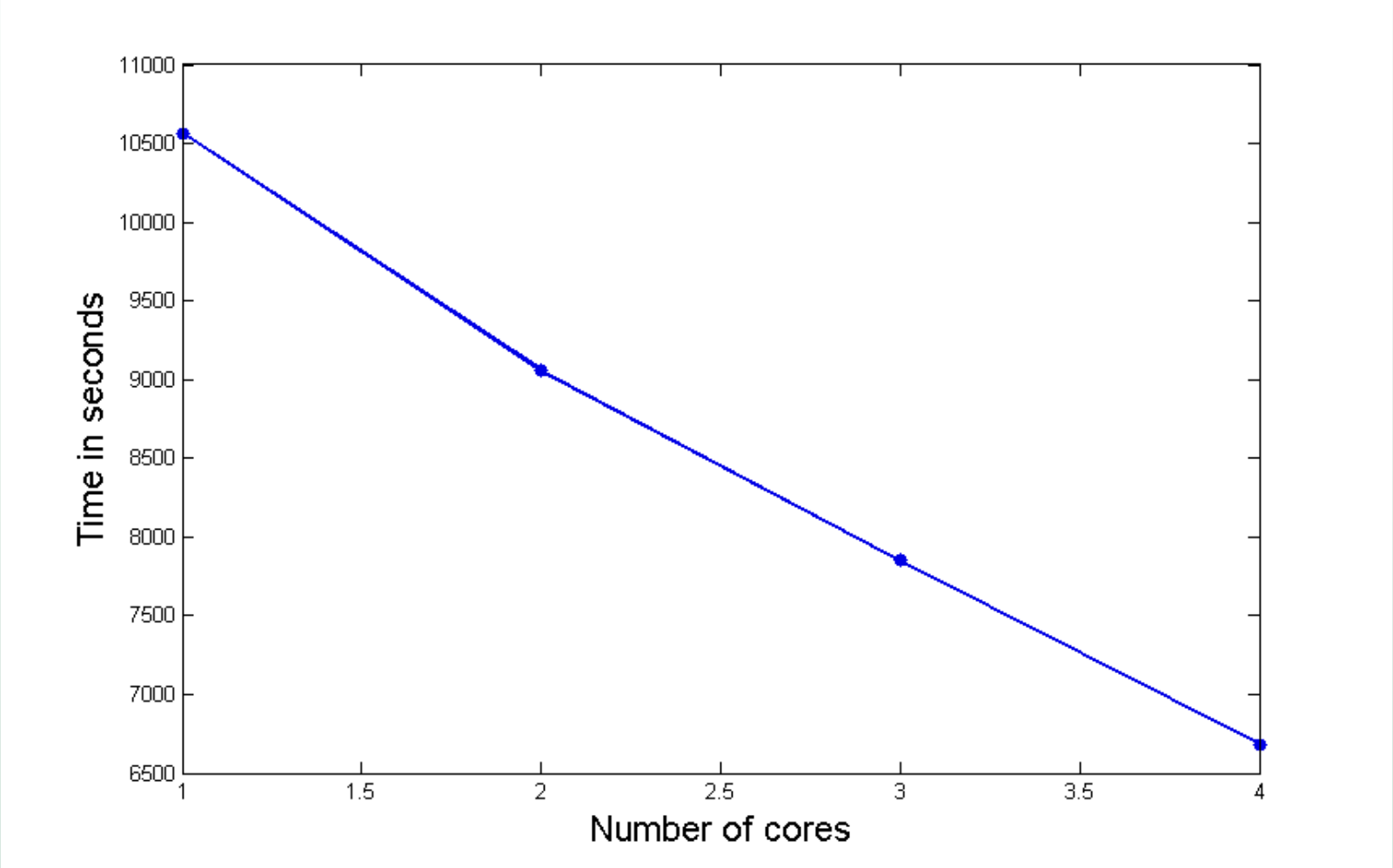

Figure: FORTRAN Timing

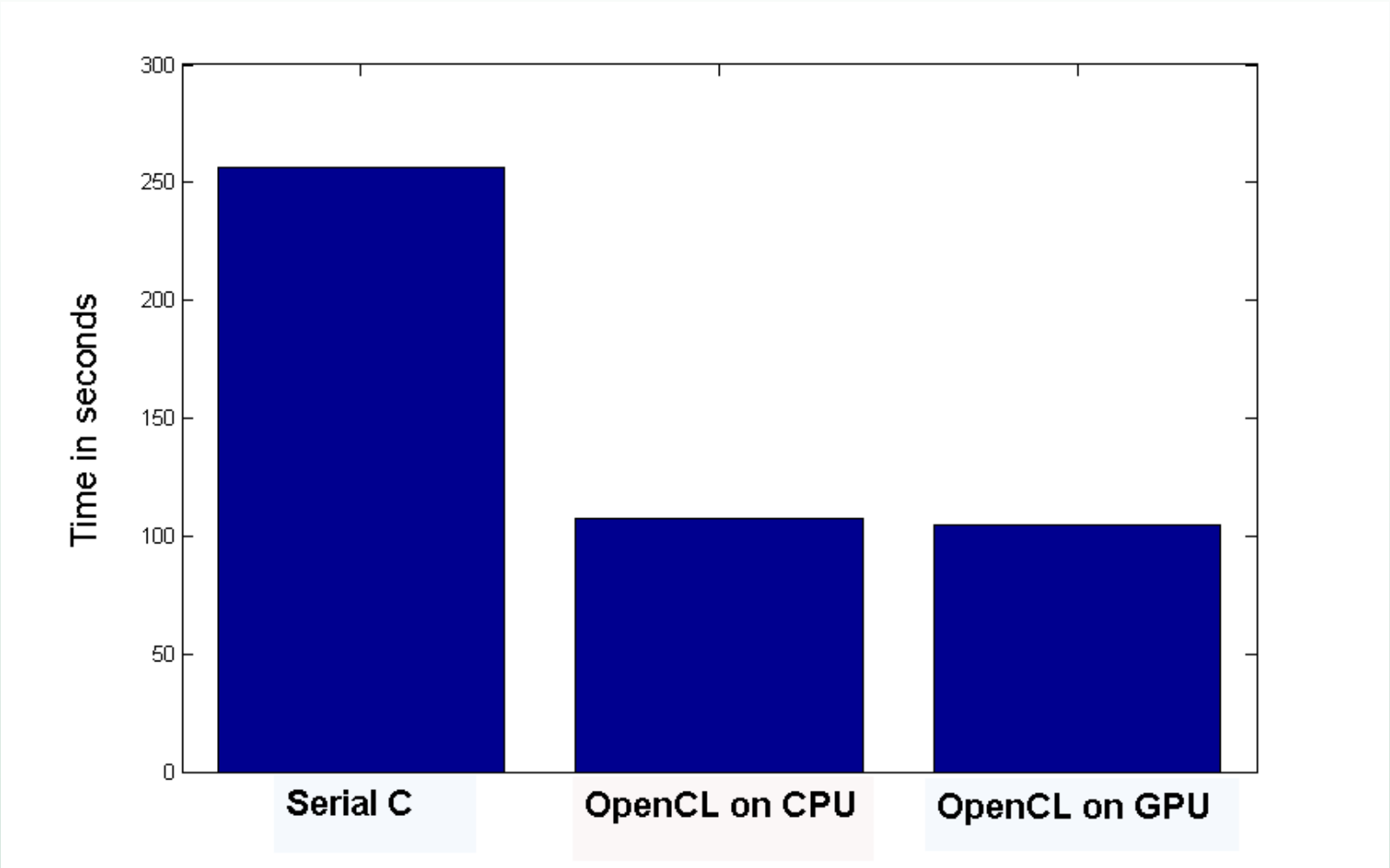

Figure: OpenCL Timing

QUESTIONS