Initial Project Overview

SOC10101 Honours Project

An Investigation into improving multi-GPU applications with Data compression

Project Content and Milestones

This project aims to research the viability of compressing data on a Graphics Processing Unit(GPU) before sending it to another GPU to reduce the transfer time and bandwidth utilisation. An application will be developed to implement and test the performance of different methods of compression and data transference.
The resulting data will be used to analyse the suitability of implementing compression methods into existing GPU work, such as realtime rendering, or distributed general purpose computation.

The Main Deliverables:

- An implemented application to test the various methods of compression and to output performance data.
- Documented findings and test results of different implementation methods
- A report into the technical possibilities and limitations of compression and data transfer using GPUs.
- A report into on the viability in using compression in actual applications.

The Target Audience for the Deliverables:

An increase in data throughput between GPUs will be beneficial for any technology that uses more than one GPU for processing data. This is a common situation as GPU performance scales well when used in configurations with multiple cards. Currently this is used extensively in the field of general data processing, encryption, video rendering/encoding, CAD applications and real time 3d Rendering.

The Work to be Undertaken:

- Preliminary Investigation into existing work and possible uninvestigated methods.
- Design and Implement a software framework for testing methods with a standard set of tests..
- Research and add different techniques for transferring data and compression into the framework.
- Record and compare all results from framework.
- Evaluate the possibility of implementing any methods into an existing application.

Additional Information / Knowledge Required:

Thorough knowledge of the OpenCl Api, including some information on how it operates behind the scenes will be needed to effectively plan, implement, and test during this project.
General knowledge of compression algorithms will also be needed, this will mainly be limited to the performance characteristics of each stage of the compression process for each different algorithm.
hardware

Information Sources that Provide a Context for the Project:

The Open Computing Language (OpenCL) framework can be used for writing programs that execute across heterogeneous platforms, particularly for deploying work to GPUs, this will be used in this project. While other standards and technologies exist, they are either proprietary or locked to a specific hardware vendor. OpenCL allows for inter-GPU communication and will be the main technology used in this project. Integrating OpenCL into other applications that need to use the GPU can be a complex and even unstable/unpedicatanle endeavour, for example communicating between an OpenGL graphics rendering context, which can only target one gpu. This requires duplication of state and (possibly all) data.

The graphics technology company AMD developed a proprietary technology named Mantle. This allowed fine control of AMD GPU hardware including useful extras like the ability to dispatch commands from separate processing threads and delegate work to certain GPUs. The work done on Mantle has been used as a basis for and then merged into the forthcoming open graphics standard “Vulkan”, developed by the Khronos Group. Vulkan will supercede OpenGL with the ability to have lower level access to the hardware during rendering.

These new features in Vulkan will allow for multi-card communication during the standard graphics pipeline without needing to resort to OpenCL workarounds. This project will be able to go ahead with either technology, and can easily be moved from one to the other depending on availability. Essentially this project is built on a pre-existing software stack that is undergoing a massive changes, therefore much of the ground covered by this project will be in areas with few documented prior implementations. The main source of information for this project will be from understanding how OpenCL/Mantle/Vulkan is implemented on the software and hardware platforms. Including any documentation available on how the hardware manufactures designed the GPU hardware so new approaches can be theorised and performance issues can be solved without access to the inner workings of the proprietary black-box systems of the GPU. There are many published guides from Manufacturers (e.g AMD, NVIDIA) on best practises to programming on each of their systems and architectures, which will be invaluable to this project.

The Importance of the Project:

Until very recently, the APIs provided for 3D rendering gave little control over how instructions were executed and where data was stored. This abstraction of the specifics has allowed Gpu manufactures to support APIs like Opengl and DirectX across many different devices and software platforms. The 3D Rendering and video games industry is now moving to a more low-level set of APIs to get a finer degree of control of the hardware with the aim to regain some of the performance lost when using a generic high level API. This is going to bring a new set of challenges for the industry. There is precedent for this type of API on games consoles, where the application has full control over all aspects of the hardware, but these are specific cases on a specific pre-defined set of hardware.

Writing 3D application to make use of more than one graphics card has not been possible on the PC platform until now, this means that there is now a large set of research that needs to be done quickly on how to best divide up work and how and when to share data between cards.

This research project will provided useful data for general computing tasks, there are many ways to share data between GPUs and the different APIs available (e.g CUDA, OpenCL) and hardware (AMD, NVIDIA, Intel) each have their own unique methods for data transfer. Work has been done in this area for compressing data before sending to other nodes in a system, but these are cases specific to a certain application, I.e only a certain type of data was compressed.

This project aims to test out compression methods on a variety of data types while keeping in mind the constraints of the applications that it may be useable for, i.e compression and transfer must be quick enough to allow the card to render a frame at an acceptable rate and quality. This will provide highly useful data to the games industry as goes through it’s current changes in rendering technologies and will

also provide insight into methods of optimising general computing algorithms and programs for general purpose processing. Depending on the release and availability of the some specific APIs, this project will make use of the most cutting edge technology available, or failing this it will provide good recommendations of areas to research when the technology does arrive.

The Key Challenge(s) to be Overcome

The primary challenge to overcome will be confirming that any findings discovered during this project are not the result of a quirk in the hardware or driver software. This could include false positive results, or false negatives due to an unaccounted for factor in the experiment. This can be overcome with rigorous testing procedures and by deploying and testing on multiple platforms with multiple hardware configurations, using equipment from different hardware vendors.