Desenvolvimento de uma aplicação em Orientação a Objetos

Pedro Miguel Ferreira Tavares Carrega - 49480
Estudo Orientado em Engenharia Informática
Mestrado em Engenharia Informática
Faculdade de Ciências, Universidade de Lisboa
fc49480@alunos.fc.ul.pt

Abstract

Este relatório foi desenvolvido com o propósito de descrever o ambito do tema de tese a ser entregue no fim deste ano lectivo e detalhar o trabalho até agora realizado.

This paper was developed with the purpose to explain GPU (Graphical Processing Unit) programing and the improvements that can be made. To demonstrate this we used the Floyd-Warshall algorithm since it is highly demanding for the CPU (Central Processing Unit) due to the high number of computations required. During our research we discovered that the available materials for learning GPGPU (General Purpose Graphics Processing Unit) are quite difficult to understand for those who are looking to learn the basics of GPU programing. For that we developed this paper explaining from scratch GPGPU and some of the most significant improvements that can be made. We will start with a simple sequential solution of the algorithm in CUDA and from there go step by step till we use most of the tools and processing power that a GPU has to offer. This paper is helpful for someone who intends to learn the basics of GPGPU or could be used as a teaching guide in an IT course to introduce GPGPU to students.

Keywords ARTSOFT, ERP, OOP, C++

1 Introduction

The purpose of this paper is to teach GPU (Graphics Processing Unit) programing and the different improvements that can be made, showing the different improvements in each version so that the solution can perform even better taking advantage of the GPU optimizations. To demonstrate this, we used a well known problem that is the Floyd-Warshall algorithm. The algorithm calculates all the possible paths between two points and saves the shortest path. It was chosen because it requires a enormous amount of computations to get the final result since it is has $O(n^3)$ complexity. With a high amount of entries, that is, a big matrix and so it can be very demanding for the CPU (Central Processing Unit). This is where the GPU benefits due to its high number of threads. For this paper all the examples will be in CUDA.

Introduction to Research , December 2021, Lisbon

.

To make a method to be computed in the GPU you need to declare it with the "__global__" qualifier so that it will run on the GPU when called in the CPU. Each kernel runs a grid which is the group of blocks that are launched by the kernel, where each block is a set of threads. The grid can be represented as a one dimensional array, a two dimensional array or a three dimensional array. This can be used to improve the mapping of the work that each thread will do.

When launching a kernel it has to be declared the number of threads per block and the number of blocks that are going to be used. It should be taken into consideration the fact that blocks are composed of warps which are a set of 32 threads, so when declaring the number of blocks being used, it should be a multiple of 32 since all threads in a warp use the same program counter. Therefor if the number of threads in the blocks is not multiple of 32 there will be threads that are busy, waiting for the other threads in their warp that are actually doing some work.

The composition of the kernel grid has a huge impact on performance, with that in mind a programmer of GPGPU (General Purpose Graphics Processing Unit), in most scenarios, should aim to use the largest amount of threads per block possible.

It also needs to be considered that the GPU does not have access to the memory used by the CPU, so all pointers used as input for the GPU kernel will have to be allocated and copied to the GPU global memory before being able to access it. The pointers of the memory that were allocated need to be passed through the parameters of the kernel call so that they can be used by the GPU. In case of a primitive type, it can be passed only by the parameters of the kernel (i.e. int) and does not need to allocate memory on the GPU memory manually. After all the computations are finished, the result should be accessible by the CPU so that it can copy back the result. To do that it needs to copy the result similarly to the opposite operation but this time from the device to the host. Then the memory that was allocated in the GPU memory should be freed so there are no memory leaks just like in the CPU. In the next figure you can see an example of a input being copied to the GPU, the call of the kernel, the corresponding composition of the grid which will be represented in the (Fig.1) and then the result being copied back to the CPU memory:

Here you should motivate your work.
What is the context?

What is the problem?

Why is it important? O cliente quer

What data and methods are you thinking about using to tackle it?

How is your approach different from what was already done (if something was done).

How is this document organised?

2 Background

This section should contain any information needed to understand the problem you are tackling.

2.1 Ferramentas Utilizadas

2.2 Formação

Os primeiros três meses da minha tese foram dedicados à minha formação.

2.3 Redação da Especificação de Requisitos

3 Related Work

This is the time to do a literature review!

What is the state of the art on the topic you are working?

Previous work with same data, similar work with other data, ...

Problems tackled, data science approaches used, pros and cons, ...

Example of a reference[1]. This was a book, but all other relevant works in papers [2] could be added as well.

4 Data {if necessary}

Creio que não tenho data.
Here you should describe your data in as much detail as possible.

You can describe raw data and any pre-processing need for your work.

You can have a section on exploratory data analysis.

5 Methods

Here you should describe in as much details as possible the problem and your plan to tackle it.

What are the methods you are planning to use, or already started to use, to tackle your problem.

This should be based on related word, your understanding of the problem and eventually preliminary exploratory data analysis or preliminary results.

5.1 You decide on the subsection

6 Preliminary Results (Optional)

Se conseguir acesso ao ambiente de qualidade antes da entrega (prob not).
Here you can include preliminary results if you already have them.

7 Forthcoming Work

Here you should write the conclusion of the preliminary work and the goals for the remaining period.

Os próximos passos a realizar no projeto será a realização de uma reunião para apresentar e aprovar a especificação de requisitos. Dada a aprovação, serão definidos vários SPRINTS quinzenais com diferentes objetivos vão ser implementadas as interfaces gráficas e requisitos descritos na especificação. Uma vez implementadas, irá ser utilizado o ambiente de qualidade para realizar testes de forma a confirmar o correto comportamento das funcionalidades implementadas. Estas novas funcionalidades irão ser incluídas na nova versão da aplicação ARTSOFT, aonde será realizado tratamento de bugs que surgam nas funcionalidades implementadas.

References

[1] Leslie Lamport. 1994. *LaTeX - A Document Preparation System: User's Guide and Reference Manual, Second Edition.* Pearson / Prentice Hall, New York.

[2] A. M. Turing. 1937. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* s2-42, 1 (01 1937), 230–265. https://doi.org/10.1112/plms/s2-42.1.230 arXiv:https://academic.oup.com/plms/article-pdf/s2-42/1/230/4317544/s2-42-1-230.pdf