****

**Project 1**

**Performance Evaluation of a Single Core**

*Relatório*

Computação Paralela

4º Ano do Mestrado Integrado em Engenharia Informática e Computação

**Elementos do Grupo:**

Diogo Vaz Nunes - 201108019 - ei11065@fe.up.pt

Mike Santos Pinto – 201103127 – ei11050@fe.up.pt

7 de Abril de 2015

# **Índice**

[**Descrição do Problema** 3](#_Toc415956671)

[**Algoritmos** 4](#_Toc415956672)

[**Demonstração e Análise de Resultados** 5](#_Toc415956673)

[**Conclusão** 6](#_Toc415956674)

# **Descrição do Problema**

A tecnologia evolui a um rácio temporal muito alto, as diferenças com as tecnologias de há duas décadas atrás são muito significativas. Com a abordagem do processamento *multi core* ao invés do *single core* com frequências superiores, é necessário entender quais as vantagens e desvantagens, em termos de processamento, em ter mais ou menos núcleos e a influência da frequência de relógio na resolução de certos problemas.

O objetivo deste projeto é estudar, através do cálculo de multiplicação de matrizes, qual o impacto no desempenho do processador tendo em conta a hierarquia da memória no acesso a grande quantidade de dados.

# **Algoritmos**

**Tipo de algoritmo**

O algoritmo utilizado no cálculo do produto matricial é um algoritmo *naïve*.

Consiste na multiplicação de cada linha da primeira matriz com uma coluna da segunda matriz obtendo-se um valor da matriz resultado, este processo repete-se até que a matriz resultado esteja totalmente calculada (ver fig.1).

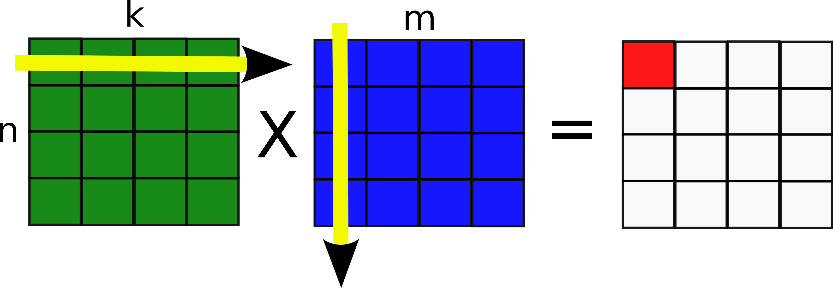


Figura 1

**1ª Versão – algoritmo *naïve* ‘normal’**

O algoritmo é essencialmente igual ao descrito em termos de implementação. Isto dá origem a uma ordem espacial *S(n3)* e uma ordem temporal *O(2n3)* no cálculo de matrizes quadradas.

**2ª Versão – algoritmo *naïve* ‘otimizado’**

Esta versão tem em conta a arquitetura da máquina e como é feito o acesso a memória, especificamente como é feito o acesso à memória cache e o carregamento dos dados entre esta memória e a principal.

O cálculo deixa de ser feito em linha por coluna e é realizado linha por linha, cada elemento da primeira matriz é multiplicado pela linha correspondente da segunda matriz.

Desta forma, o algoritmo possui uma ordem espacial *S(n3)* e uma ordem temporal *O(n3)* no cálculo de matrizes quadradas. Realça-se que a otimização pouco ou nada altera estas variáveis, a otimização feita faz-se notar nos tempos de acesso a memória.

# **Demonstração e Análise de Resultados**

# **Conclusão**