CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Análise e Testes de Algoritmos em Sistemas de Arquiteturas Híbridas CPU/GPU

Danilo da Silva Maciel

Marília, 2015

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Análise e Testes de Algoritmos em Sistemas de Arquiteturas Híbridas CPU/GPU

Monografia apresentada ao Centro Universitário Eurípides de Marília como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Prof. MS. Maurício Duarte

Marília, 2015

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
MANTIDO PELA FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Danilo da Silva Maciel

Análise e Testes de Algoritmos em Sistemas de Arquiteturas Híbridas CPU/GPU.

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 9.0 (nove)

Orientador:Mauricio Duarte

1º.Examinador:Ildeberto de Gênova Bugatti

2º.Examinador:Fabio Piola Navarro

Marília, 02 de dezembro de 2015.

DEDICATÓRIA

Aos meus pais, Jorge e Luiza, pelo sacrifício ilimitado em todos os sentidos, que mais do que me proporcionar uma boa infância, formaram os fundamentos do meu caráter e me apontaram o caminho a seguir. Obrigado por serem a minha referência de tantas maneiras e estarem sempre presentes na minha vida de uma forma indispensável.

A minha avó Ruth por tudo o que ela fez e por tudo o que ela me proporcionou para que eu pudesse ingressar no ensino superior, obrigado por tudo sem a senhora eu não estaria finalizando mais esta etapa da minha vida.

A minha Irmã Daniela e meu Irmão Júnior que sempre estiveram dispostos a ajudar de alguma forma, seja ela na forma de palavras amigas ou conselhos que me ajudaram a permanecer firme até o final desta jornada incentivando a não desistir. Agradeço também a Joice pela paciência e pelo apoio que nos momentos que eu precisei, estendeu suas mãos para ajudar.

Muito obrigado, nunca será suficiente para demonstrar a grandeza do que recebi de vocês.

AGRADECIMENTOS

Ao Profº. Ms. Maurício Duarte, orientador, professor, amigo, pelo desprendimento ao escolher me dar apoio, muito obrigado pela dedicação e ajuda, que apesar de toda a dificuldade enfrentada no último ano se mostrou uma pessoa de fé inabalável e exemplo de superação, com certeza lembrarei pra sempre deste mestre como exemplo e espelho para minha vida.

Agradeço ao Centro Universitário Eurípides de Marília – UNIVEM pelo incentivo, apoio e infraestrutura para a graduação e o desenvolvimento e conclusão deste projeto. Agradeço também a todos os professores que sempre estiveram dispostos a ajudar e incentivar os alunos que assim como eu tiveram dificuldades durante a graduação.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

GPU - Unida de Processamento Gráfico (Graphics Processing Unit)
CPU - Unidade Central de Processamento (Central Processing Unit)
PU - Unidade de Processamento (Processing Unit)
DSP - Digital Signal Processor
FPGA - Field Programmable Gate Array
UC - Unidade de Computação,
PE - Elemento de Processamento

Resumo

Há pouco tempo atrás, a principal utilização de placas de videos (GPU - Graphics Processing Unit ou Unidade de Processamento Gráfico) era na utilização de jogos eletrônicos e processamento de videos em alta definição. Mas nos últimos anos tem se observado um aumento muito grande no campo de pesquisa e utilização de GPUs fora deste contexto, assim como a CPU (Central Processing Unit ou Unidade de Processamento Central) evoluiu , a GPU também evoluiu aumentando consideravelmente o poder computacional dos mesmos. Através deste contexto este trabalho busca tentar demonstrar através da utilização da CPU, GPU e do conjunto CPU e GPU, comprovando através de testes o eventual ganho ou não de performance na execução destes algoritmos com relação ao tempo de execução das arquiteturas envolvidas.

Palavras Chave: CPU, GPU, performance, algoritmos, execução.

Abstract

Not long ago , the main use of video cards (GPU - Graphics Processing Unit or graphics processing unit) was the use of electronic games and video processing in high definition. But in recent years has seen a very large increase in the search field and use of GPUs outside this context , as well as the CPU (central processing unit or CPU) has evolved , the GPU has also evolved greatly increasing the computational power of the same . Through this context, this work aims to try to demonstrate through the use of CPU , GPU and CPU and GPU together , demonstrating by testing the possible gain or non- performance in the execution of these algorithms with respect to time of execution of the architectures involved .

Keywords: CPU, GPU, performance, algorithms, execution.

Introdução

A evolução dos computadores permitiu que vários componentes presente em sua arquitetura evoluíssem de forma espantosa, principalmente o componente de processamento central chamado de CPU, que teve um aumento em seu poder computacional de forma impressionante nos últimos anos, os fabricantes conseguiram por um bom tempo explorar este dispositivo de forma satisfatória, mas algumas barreiras encontradas forçaram os fabricantes de hardware procurarem outras alternativas para aumentar o poder de processamento dos computadores como um todo. Com isto outro componente vem ganhando bastante, um processador que antes era dedicado ao processamento de imagens e video, a GPU está sendo utilizada para outros fins, mudando a forma de processamento tradicional.

Com estas informações em mente, o principal objetivo deste trabalho é avaliar o desempenho de execução de um algoritmo que realiza a subtração entre dois vetores explorando a CPU/GPU, CPU e somente a GPU, observando em que determinado cenário cada arquitetura se destaca.

Como a CPU e a GPU possuem certas particularidades, mas que ambos possuem alto poder de processamento, fica difícil distinguir até que ponto é vantagem a utilização de somente um deles ou até mesmo deles em conjunto, afim de processar uma quantidade de dados maior. Sendo assim os testes do algoritmo proposto foram executados nas arquiteturas da CPU, da GPU e da CPU/GPU combinadas, ficando mais claro o entendimento do cenário em que cada arquitetura consegue obter melhor desempenho de execução em relação ao tempo de processamento.

Para os testes, foi criado um Kernel OpenCL, que pode ser entendido como uma pequena unidade de execução que possui clareza em sua função e pode ser processada paralelamente[AMD], e será executada tanto na CPU como na GPU e também na CPU/GPU, sem alteração. Três vetores serão submetidos ao Kernel, que será feito a operação de subtração de um valor contido na posição de um, pelo valor que estará contido na posição do outro, o resultado será armazenado no terceiro vetor. Nas arquiteturas individuais a Unidade de processamento do dispositivo irá processar todo o vetor, mas no caso da CPU/GPU será dividido de forma que cada Unidade de Processamento processe uma parte deste vetor. Com o processamento do Kernel pelo dispositivo, será possível através de diretivas do próprio OpenCL, o resgate dos valores com relação ao tempo utilizado para o processamento do

Kernel por cada dispositivo, ficando mais fácil a comparação entre os resultados obtidos.

Organização do Trabalho

O conteúdo contido neste trabalho foi organizado em 5 Capítulos. No Capítulo 1 é feita uma introdução a Evolução dos Processadores com uma breve explicação sobre a CPU, Arquitetura Multicore, GPUs, Arquitetura Híbrida e Algoritmos Paralelos. No Capítulo 2 é realizada uma Introdução a metodologia OpenCL, realizando uma explicação sobre os modelos utilizados na arquitetura da mesma. No Capítulo 3 se encontra a Definição do Projeto com a explicação de Trabalhos Correlatos encontrados e algumas características do código dos testes realizado e as características da CPU e GPU utilizada nos mesmos. No Capítulo 4 estão relacionados os resultados obtidos nos testes realizados, com tabelas e gráficos informativos. O Capítulo 5 contém a Conclusão realizada com os resultados do Trabalho e propostas de trabalhos futuros.

Metodologia

O projeto foi divido em quatro etapas principais:

- Estudo e revisão bibliográfica sistemática do Paralelismo em CPUs e GPUs. Nessa foram pesquisados projetos e tecnologias sobre Paralelismo em CPUs e GPUs.
- Estudo e revisão bibliográfica sistemática da linguagem escolhida para a escrita do programa Host. Nessa etapa foram pesquisados e lidos trabalhos anteriores sobre OpenCL e programação paralela utilizando OpenCL.
- Análise e desenvolvimento do algoritmo que foi utilizado para os testes de processamento das duas arquiteturas.
- Implementação e validação do algoritmo no computador de testes, após a implementação os testes foram definidos e executados para cada variação de tamanho do vetor.
- Foi gerado tabelas e gráficos dos resultados obtidos, sendo possível após essa etapa a comparação entre os testes.

Capitulo 1 - A evolução dos Processadores

1.1 CPU

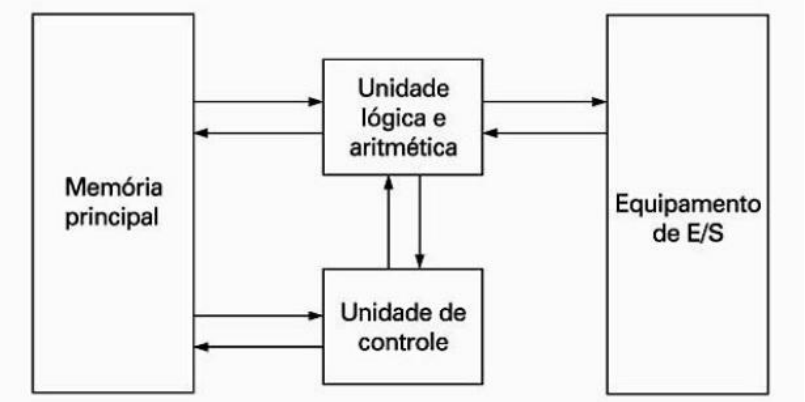
"A CPU (Central Processing Unit - Unidade Central de Processamento) é o 'cérebro' do computador. Sua função é executar programas armazenados na memória principal buscando suas instruções, examinando-as e executando-as uma após a outra".[TANEMBAUM]

A evolução dos computadores tem sido caracterizada pelo aumento da velocidade dos processadores, pela diminuição do tamanho dos componentes, pelo aumento da capacidade da memória e pelo aumento da capacidade e da velocidade de lidar com dados de entrada e saída.[STALLINGS]

A dessas afirmações fica fácil compreender que o processador é um dos principais componentes presentes em um computador, por isso ele é um dos componentes que mais evoluí rapidamente dentro deste conjunto.

A CPU é composta basicamente de unidade de controle, registradores, unidade lógica e aritmética, unidade de execução de instruções e interconexões entre estes componentes, que também são chamadas de barramento. [STALLINGS]. A figura 1 ilustra uma arquitetura básica de Von Neumann.

Figura 1 - Arquitetura de Von Neumann.

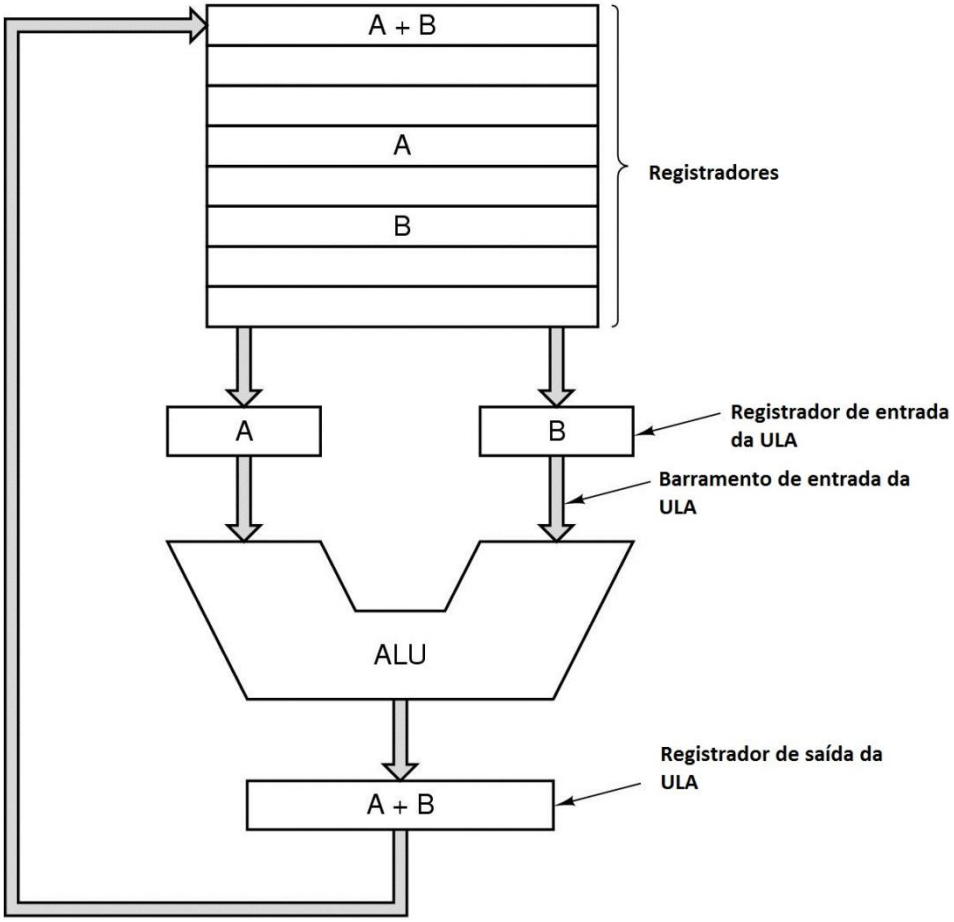


Fonte: [STALLINGS].

Uma das partes importantes de uma CPU, é denominada caminho de dados, que é composta por registradores, a ULA e diversos barramentos que conectam as peças, os

registradores alimentam os registradores A e B que irão conter as entradas para ULA realizar cálculos, este processo é chamado de ciclo de caminho de dados, ele define o que a máquina pode fazer, quanto mais rápido o ciclo, mais rápido será o processamento da máquina. [TANEMBAUM]. A figura 2 ilustra o exemplo do caminho de dados da arquitetura de Von Neumann.

Figura 2 - Caminho de dados, arquitetura de Von Neumann.



Fonte: [TANENBAUM].

O processador é bem pequeno se comparado a outros componentes de um computador dado a sua importância dentro de uma arquitetura, a CPU é composta por outros componentes menores.

Mas a primeira vez que foi possível reunir todos estes componente em um só, foi a partir de 1971, a empresa Intel desenvolveu o processador Intel 4004 o primeiro processador que continha em sua pastilha todos os componentes de uma CPU, este dia foi um marco muito importante pois, havia nascido o microprocessador.[STALLINGS]

O Intel 4004 conseguia somar números de até 4 bits e através de somas consecutivas,

conseguia realizar a operação de multiplicação. Na época era algo realmente novo, já que os processadores antecessores não tinham tanto poder de processamento. E a partir dessa inovação, inaugurou-se uma era de evolução contínua na capacidade e poder de processamento dos computadores ou de tudo o que trabalharia com processadores.[STALLINGS]

Logo após a revolução que o Intel 4004 trouxe, a Intel desenvolveu o Intel 8008 e foi lançado em 1972, apenas dois anos após o 4004. E com ele outra inovação nascia, o 8008 foi o primeiro processador a trabalhar com palavras de 8 bits, dobrando a quantidade de bits do seu antecessor o 4004, mas com isso também dobrando sua complexidade comparada a versão anterior da Intel. Mas o Intel 4004 e o Intel 8008, eram dois processadores que eram utilizados para fins específicos, ou seja, para uma determinada tarefa, mas isso começou a mudar com o próximo lançamento da Intel e em 1974 a Intel lança no mercado o Intel 8080, este sim um processador de propósitos gerais e possuía um poder de processamento de 8 bits, assim como seu antecessor, mas possuía mais instruções, era mais rápido e possuía maior capacidade de endereçamento de memória.[STALLINGS]

Antes do final dos anos 70 foram lançados os processadores de 16 bits de propósitos gerais. Em 1981 a Heawlett-Packard e a Bell Laboratories desenvolveram o primeiro processadores de 32 bits em uma única pastilha, a Intel em 1985 lançou a sua versão de processadores de 32 bits, o 80386. A tabela 1 contém algumas informações técnicas sobre a evolução dos processadores Intel.

Tabela 1 - Evolução dos Processadores Intel

Nome	Data	Transistores	Microns	Velocidade do clock	Largura de Dados	MIPS
8080	1974	6.00	6	2 Mhz	8 bits	0,64
8088	1979	29.000	3	5 Mhz	16 bits/8 bits	0,33
80286	1982	134.000	1,5	6 Mhz	16 bits	1
80386	1985	275.000	1,5	16 Mhz	32 bits	5
80486	1989	1.200.00	1	25 Mhz	32 bits	20
Pentium	1993	3.100.00	0,8	60 Mhz	32 bits/64 bits	100
Pentium II	1997	7.500.00	0,35	233 Mhz	32 bits/64 bits	300
Pentium III	1999	9.500.000	0,25	450 Mhz	32 bits/64 bits	510
Pentium 4	2000	42.000.000	0,18	1,5 Mhz	32 bits/64 bits	1,700
Pentium 4 "Prescott"	2004	125.000.000	0,09	3,6 Mhz	32 bits/64 bits	7,000
Pentium D	2005	230.000.000	90nm	2,8 Mhz / 3,2 Mhz	32 bits	
Core2	2006	152.000.000	65nm	1,33 Mhz/2,33 Mhz	32 bits	26,000

Core 2 Duo	2007	820.000.000	45nm	3 Mhz	64 bits	53,000
Core i7	2008	731.000.000	45nm	2,66 Mhz/3,2 Mhz	64 bits	76,000

Fonte: Adaptado de [FILHO].

Seguindo a lei de Moore, o poder de processamento e a quantidade de bits que o processador conseguia trabalhar, foi quase que dobrando com o passar do tempo e o custo diminuindo.

1.2 Arquitetura Multicore

Em uma arquitetura Multicore, existem mais de um núcleo de processamento de alta frequência com recursos avançados no mesmo componente. [STONE,GOHARA,SHI]

A arquitetura Multicore surgiu como uma das alternativas ao esgotamento da arquitetura tradicional de processadores. Em uma única pastilha é colocado mais de um núcleo físico de processamento. Com este tipo de arquitetura, teoricamente os sistemas operacionais podem delegar para cada núcleo, uma parte do processamento, tornado a realização da tarefa muito mais rápida.[PINTO]

Neste tipo de arquitetura, os núcleos que estão na mesma pastilha, compartilham a mesma memória cache.

> *Do ponto de vista do sistema operacional, cada núcleo de um processador multicore é interpretado como um processador lógico independente com recursos de execução distintos, de forma que cada núcleo pode executar aplicações distintas. Além disso, através do uso de técnicas de programação paralela é possível executar uma aplicação em um conjunto de núcleos paralelamente buscando reduzir o tempo de execução em relação a um processador sequencial. [PINTO]*

1.3 GPUs

A GPU que antes era para quem utilizava os computadores para fins de computação gráfica, edição de video e imagens, jogos eletrônicos. Segundo STEFANELLO et al, a GPU "passou de um simples processador gráfico para um coprocessador paralelo, capaz de executar milhares de operações simultaneamente, gerando uma grande capacidade computacional, que muitas vezes supera o poder de processamento das tradicionais CPUs".

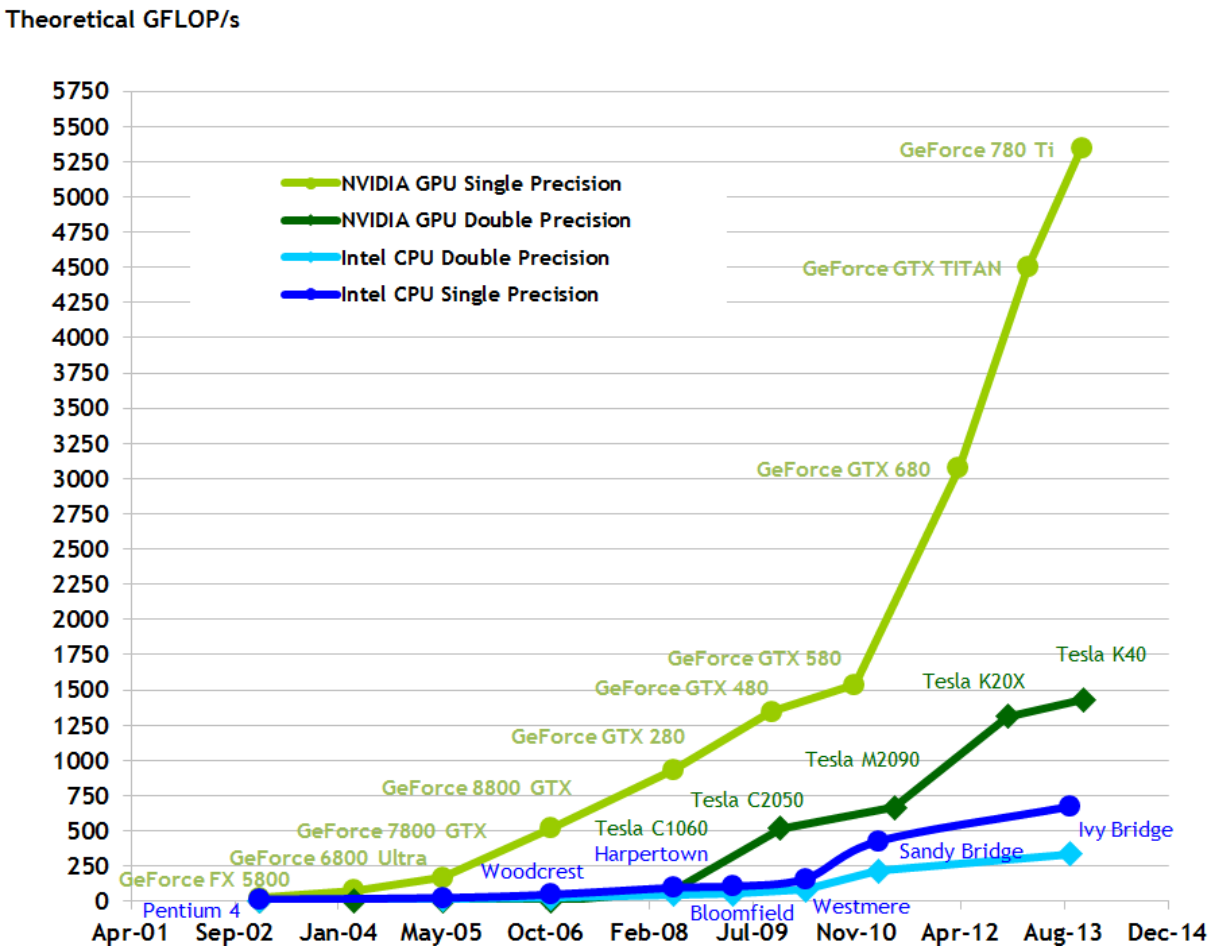
A diferença fundamental entre os processadores (CPUs) e os chipsets de vídeo (GPUs) é o fato de que as CPUs são otimizadas para cálculos sequenciais (executando aplicativos

diversos), enquanto as GPUs são otimizadas para cálculos massivamente paralelos, processando gráficos 3D.[MORIMOTO]

O grande diferencial nas GPUs ocorreu com a introdução de shaders (que são pequenos aplicativos, destinados a executarem tarefas específicas na composição das cenas) com isso as GPUS atuais podem executar código sequencial, semelhantes as instruções executadas pelas CPUs.[MORIMOTO]

A arquitetura da GPU assim como a da CPU, tem evoluído de tal forma que o ganho no processamento computacional acompanha a evolução em cada arquitetura. A figura 3 demonstra de forma gráfica, a comparação do poder de processamento das GPUs Nvidia em comparação com processadores Intel.

Figura 3 - Comparação de poder computacional entre CPUs Intel e GPUs Nvidia.



Fonte : [NVIDIA].

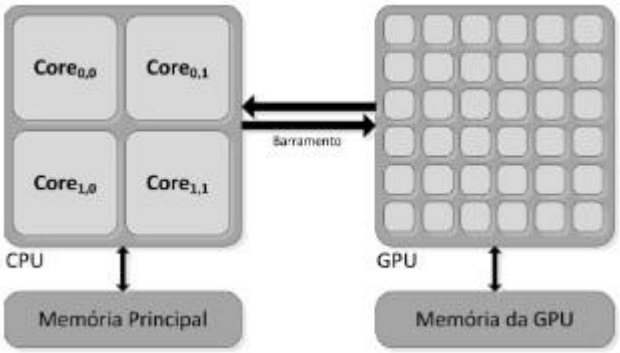
1.4 Arquitetura Híbrida

Ao analisar as duas arquiteturas CPU e GPU, é possível verificar que a união destas duas arquiteturas tem muito a oferecer, este trabalho visa testar e documentar o desempenho de processamento da união não fisicamente, mas sim virtualmente através de programação paralela destas arquiteturas.

Alguns dos motivos da adoção de arquitetura híbrida em sistemas computacionais paralelos, se dá principalmente pelos limites atingidos das arquiteturas tradicionais e pelo surgimento da arquitetura multicore.[PINTO]

A figura 4, demonstra o funcionamento da arquitetura híbrida, utilizando a arquitetura Multicore de Processadores e os núcleos da GPU. Cada arquitetura contém sua memória e a mesma não é compartilhada.

Figura 4 - Arquitetura Híbrida.



Fonte:[PINTO].

Para que as arquiteturas funcionem em conjunto, elas necessitam utilizar algoritmos que consigam aproveitar esse paralelismo de arquitetura. São o algoritmos paralelos, um modelo de programação que explora a utilização de mais de um núcleo de processamento.

1.5 Algoritmos Paralelos

Durante muito tempo os algoritmos foram escritos para serem executados de forma sequencial, “quebrando” o problema em problemas menores para facilitar sua resolução e executados um a um de forma sequencial, mas já a algum tempo está sendo disseminado o conceito de algoritmos paralelos que trabalha de forma parecida com a forma sequencial, ele

quebra o problemas em partes menores e utiliza os recursos da máquina, como a arquitetura Multicore para resolver o problema de forma que cada parte deste problema seja resolvida com independência da outra. Mas não são todos os aplicativos que são escritos para serem executados desta forma, para isso eles necessitam utilizar técnicas de paralelização como por exemplo o uso de Threads, dividindo o código a ser executado em um núcleo disponível da CPU.[KIRK, WHU]. Utilizando este tipo de algoritmo o código é dividido em partes e cada parte será executada por um núcleo diferente, visto que ao invés do código ser executado em um só núcleo, ele é dividido entre outros núcleos ganhando desempenho na execução, diminuindo a concorrência pelo uso de um único núcleo da CPU.

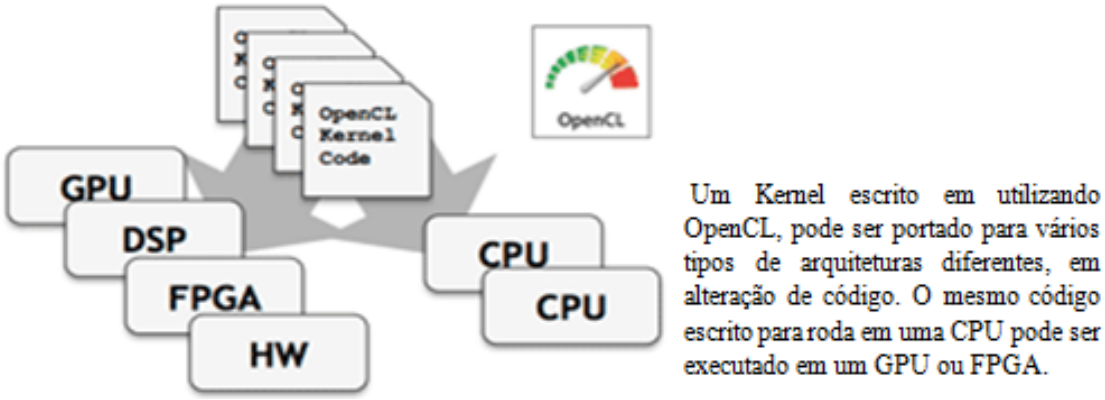
Capitulo 2 - Metodologia OpenCL

O OpenCL é um framework que foi inicialmente desenvolvido pela Apple em parceria com a Nvidia com a ideia da criação de uma plataforma versátil para o desenvolvimento de aplicações para tirar proveito do processamento dos múltiplos cores da GPU , ele foi apresentado ao Khronos Group em 2008[GRAMMELSBACHER, MEDRADO]. Segundo KHRONOS OpenCL é um padrão aberto multi plataforma para programação paralela de processadores modernos que abrange diversos dispositivos como computadores pessoais, servidores e dispositivos móveis. [KHRONOS GROUP]

Sua aplicação engloba vários segmentos como jogos, meio científico, medicina, etc. O código escrito pode ser executado em CPUs, GPUS, DSPs, FPGA sem alteração, como é ilustrado na Figura 5.

O OpenCL é baseado na linguagem C e pode ser utilizado em placas de diversos fabricantes, para isso basta que o fabricante implemente o OpenCL em seu dispositivo.

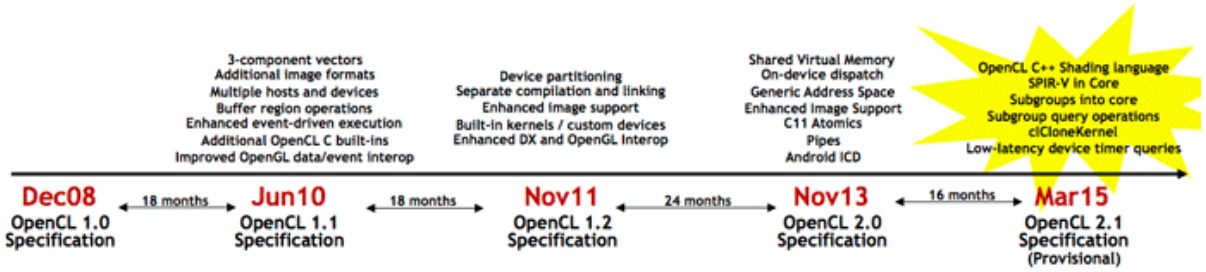
Figura 5 - Código OpenCL pode ser portável para dispositivos diferentes.



Fonte: Adaptado de [KHRONOS GROUP].

Desde que foi criado, o OpenCL sofre constantes evoluções e a cada versão que é lançada traz melhorias para os desenvolvedores e fabricantes. Na figura6 é possível ver esta evolução e suas modificações.

Figura 6 - Evolução do OpenCL.



Fonte: [KHRONOS GROUP].

O OpenCL consiste em uma API para a exploração de computação paralela em processadores heterogêneos, é uma linguagem de programação multi-plataforma com foco em um ambiente específico[KHRONOS]. O padrão OpenCL possuí as seguintes características:

- Suporta dados e tarefas com base em modelos de programação paralela.
- Utiliza um subconjunto de ISO C99 com extensões para paralelismo.
- Define os requisitos numéricos consistentes baseados em IEEE 754.
- Define um perfil de configuração para dispositivos portáteis e embarcados.
- Eficientemente interopera com OpenGL , OpenGL ES e outras APIs gráficas.

2.1 Arquitetura OpenCL

Segundo KHRONOS, o OpenCL é um padrão aberto que facilita a utilização de vários dispositivos de fabricantes diferentes, mas que possuem a implementação do OpenCL, eles podem ser agrupados por uma Plataforma específica para fornecer facilidades que ocultam a dificuldade e a necessidade de conhecimento de hardware de baixo nível, o que facilita o desenvolvimento aproveitando as características da plataforma. [KHRONOS]

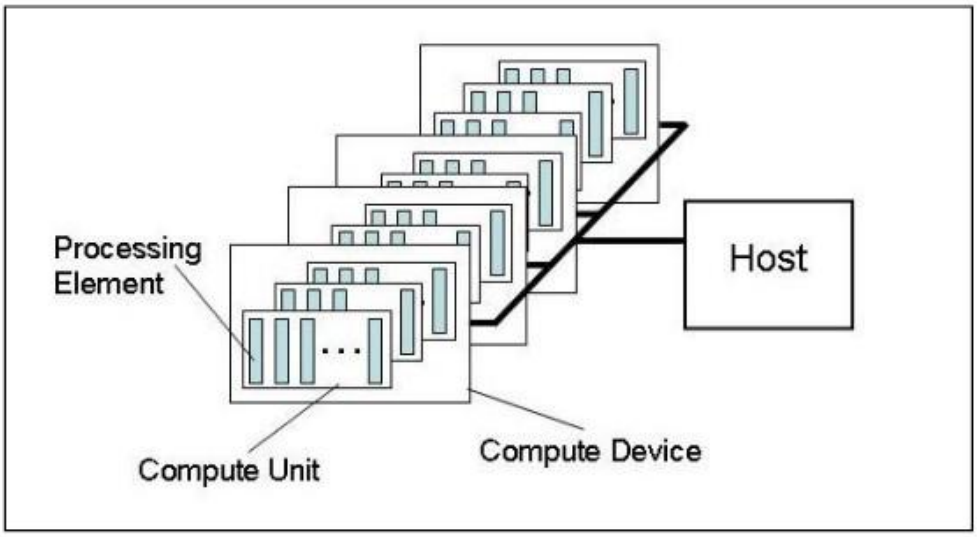
O OpenCL possui alguns modelos que possuem uma hierarquia entre eles:

- Modelo de Plataforma;
- Modelo de Memória;
- Modelo de Execução;
- Modelo de Programação.

2.2 Modelo de Plataforma

Este modelo utiliza uma unidade de processamento que irá hospedar a aplicação Host, ela necessita possuir suporte OpenCL, mas aplicação Host pode ser em outras linguagens, ela irá gerenciar os outros dispositivos que também suportem OpenCL. Os dispositivos OpenCL são classificados como unidades de computação, os agrupamentos das unidades de computação são classificados como Elementos de Processamento. O processamento no Padrão OpenCL ocorre nestes agrupamentos de dispositivos (Elementos de Processamento). [KHRONOS, 2011]. A figura 7 demonstra o Modelo de Plataforma utilizado pelo OpenCL.

Figura 7 - Modelo de Plataforma utilizado pelo OpenCL.



Fonte: [KHRONOS, 2011]

2.3 Modelo de Execução

Este modelo descreve como o OpenCL executa os programas, que ocorre em duas etapas:

- Um Kernel (Função) é executado por um ou mais dispositivos OpenCL que estejam de certa forma ligados ao Host;
- Um programa hospedeiro chamado de Host que é responsável por gerenciar o Kernel.

Segundo KIRK e HWU, como o OpenCL consegue trabalhar com uma gama extensa

de dispositivos(Devices), é necessário que se crie um contexto para identificar um ou mais dispositivos, desta forma o programa hospedeiro define o contexto do Kernel e administra seu núcleo de execução [KHRONOS].

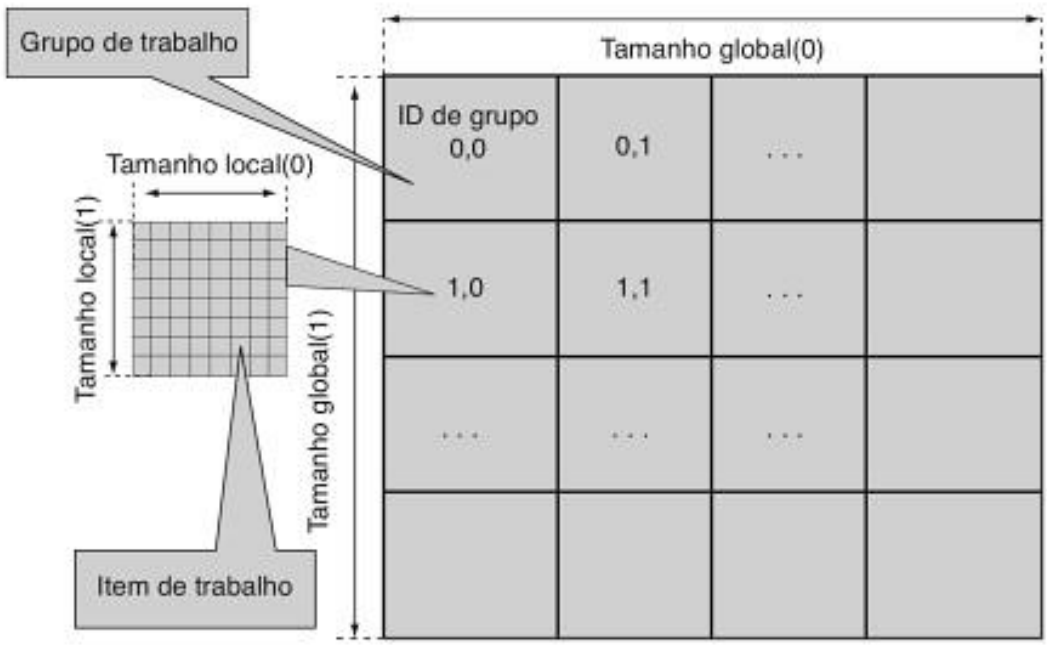
Um execução paralela no OpenCL ocorre quando o host começa disparar as funções do Kernel.[KIRK, HWU]. Quando um Kernel entra na fila de execução no dispositivo, é então criado um Espaço de Índice e cada ponto neste espaço é utilizado por uma instância do Kernel, semelhante ao funcionamento de Threads. Cada instância do Kernel neste espaço é chamado de Work-Item(Item de Trabalho), ele ganha um identificador referente ao seu ponto no Espaço de Índice, que define sua identificação global.

Os Work-Items se agrupam e formam Work-Groups(Grupos de Trabalho), este também recebe um identificador que faz o Work-Group ser único, os Work-Items recebem uma identificação dentro do Work-Groups, desta forma o Work-Item pode ser identificado através do seu identificador global ou através da combinação do identificador do Work-Groups mais o identificador do Work-Item dentro do do Work-Groups. A dimensão de um Work-Groups é relativa a quantidade de Work-Items que o Work-Groups possui, os Work-Items executam concorrentemente os elementos de processamento de um único espaço de índice que é chamado de NDRange e possuí N-Dimensões, sendo N um número inteiro(1, 2 ou 3) e que define sua dimensão.[KHRONOS].

Um NDRange é definida por uma matriz de números inteiros de comprimento N que especifica a extensão do espaço de índice em cada dimensão a partir de um índice de deslocamento F (que o valor por padrão é zero). [KHRONOS]

Os Work-Items são semelhante a Threads, os Work-Group são semelhantes a um bloco de Threads que são utilizadas pela CPU na paralelização de tarefas. A Figura 8 ilustra os conceitos de Work-Group e Work-Item.

Figura 8 - Exemplo de Work-Items(Item de trabalho) e Work-Groups(Grupo de Trabalho), paralelismo de dados.



Fonte: [KIRK, HWU].

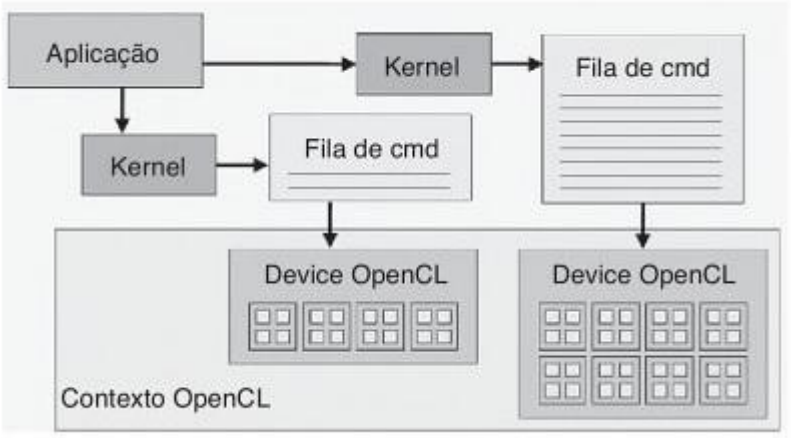
2.4 Modelo de Execução: Contexto e Fila de Comandos

O contexto do Programa Hospedeiro (Host) defina a função do Kernel e possui os seguintes recursos:

- **Device**: Coleção de Dispositivos OpenCL a disposição do Programa Hospedeiro.
- **Kernel**: São as funções OpenCL que serão executadas nos Dispositivos.
- **Objetos de Programas**: São os códigos fontes e executáveis que compõem o Kernel.
- **Objetos de Memória**: São partes da memória que ficam visíveis para o Programa Hospedeiro e para os Dispositivos permitindo comandos de acesso e leitura de dados, eles possuem valores que podem ser manipulados pelo Kernel.

O programa Host cria e manipula o contexto do Kernel através da API do OpenCL, criando uma estrutura chamada de Fila de Comando para coordenar a execução dos Kernel nos Dispositivos[KHRONOS, 2011]. A Figura 9 ilustra o conceito do gerenciamento dos dispositivos através do Contexto.

Figura 9 - Gerenciamento dos Dispositivo através do Contexto.



Fonte: [KIRK, HWU].

O funcionamento da Fila de Comando é semelhante a estrutura Fila presente na maioria das linguagens de programação, quando o Device estiver disponível para processar o próximo Kernel, o Kernel do inicio da Fila de Comando será retirado para ser processado.

Na Fila de Comando, os Kernels que serão executados são enfileirados de acordo com a chegada, mas isso não garante que serão executados na mesma ordem sendo necessário que o Host realize a sincronização para que haja consistência nos dados e na aplicação. As Filas de Comandos serão executadas nos Dispositivos sendo possível criar múltiplas filas de comandos destinados a diferentes dispositivos, neste caso o OpenCL vai executá-los simultaneamente[TAY], mas uma fila atende somente um dispositivo.

Os comandos que podem ser executados são:

- **Comando de Execução do Kernel:** Executa um comando em um Dispositivo OpenCL.
- **Comandos de Memória**: Realiza operações na Memória como transferência de valores entre Objetos, mapeamento.
- **Comandos de Sincronização**: Usado para restringir a ordem de execução de comandos.

De forma assíncrona, os comandos são executados no Host e nos Device, de dois

modos[KHRONOS, 2011]:

- **Ordem de Execução**: Os Kernels são executados na ordem em que se encontram na fila de comando e de como foram inseridos nela, serializando os comandos.
- **Fora de Ordem de Execução**: Os Kernels são processados sem uma ordem definida, mas é necessário um realizar um controle através de comandos de sincronização se o dados processados são compartilhados.

O modelo de execução do OpenCL suporta dois tipos de Kernel[KHRONOS, 2011]:

- **Kernel OpenCL:** escritos em OpenCL C e compilados com o compilador OpenCL, possui todas as implementações do OpenCL, seus mecanismos permite a criação de outros Kernels.
- **Kernel Nativos:** são implementadas nos dispositivos e acessadas como ponteiros pelo Host.

2.5 Modelo de Memória

Um Work-Item quando executa um Kernel, possuí acesso a 4 regiões da memória diferentes, Global, Constante, Local e Privada. Elas serão explicadas de forma breve nas linhas abaixo:

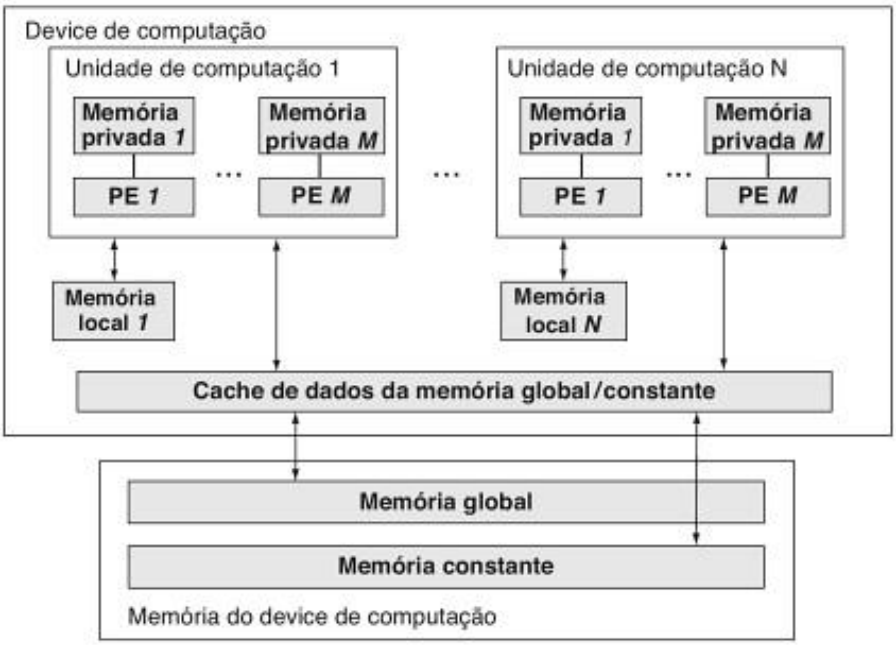
- **Memória Global:** fica disponível para todos os Work-Items em todos os Work-Groups. Os objetos nesta região da memória podem serem lidos ou escritos a qualquer momento pelos Work-Items.
- **Memória Constante:** é uma parte da memória Global que possuí um valor que não pode ser alterado durante a execução do Kernel. Sendo o seu valor definido pelo Host sendo visível a todos work-item.
- **Memória Local:** é a memória que fica disponível em um Work-Group para seus Work-Items, sendo compartilhada entre eles. Pode ser mapeada na memória do Device ou mesmo em seções da Memória Global.
- **Memória Privada:** é a memória disponível para cada Work-Item sendo restrita

somente a ele, nenhum outro Work-Item tem acesso aos dados da região privada de outro.

O modelo de memória utilizado pelo OpenCL garante a integridade dos dados na memória disponível para o Work-Groups, mas para Work-Groups diferentes não há garantia da consistência desses dados.[KHRONOS], visto que cada Work-Group possuí sua memória. Dentro de um Work-Group, seus Work-Items conseguem se sincronizar, mas o mesmo não é possível entre Work-Items de Work-Groups diferentes [KIRK, HWU].

A Figura 10 ilustra as regiões da memória e suas relações com os modelos abordados, o programa Host não aparece na ilustração. Cada Device é considerado uma ou mais Unidades de Computação(UCs), uma Unidade de Computação consiste em um ou mais Elementos de Processamento(PEs), a computação em um Device ocorre nos PEs individuais.[KIRK, HWU].

Figura 10 - Relação dos Modelos e as regiões da memória acessíveis ao Work-Item. O Host não aparece.



Fonte: [KHRONOS, 2011].

2.6 Modelo de Programação

O OpenCL permite que programadores tenha o controle do paralelismo e o controle de dados em processadores massivamente paralelos, através das extensões da linguagem e APIs de runtime[KIRK, HWU].

A linguagem utilizada no OpenCL é baseada em C99 na escrita do Kernel para os dispositivos, já o programa Host(Hospedeiro) pode ser escrito em outras linguagens[SOUZA].

O OpenCL suporta os modelos de paralelismo:

- **Paralelismo de dados:** envia instruções a múltiplos elementos que estão na memória, o espaço de índice e o modelo de execução define cada Work-Item e seu mapeamento de dados. Cada Work-Item em execução possuí seus próprios dados para manipulação, mas todos são instâncias do mesmo Kernel.
- **Paralelismo de tarefas:** diferente do paralelismo de dados, a paralelização ocorre referente as tarefas(processos), cada Work-Item instância somente um kernel, ficando independente de espaço de índice.
- **Paralelismo Híbrido:** OpenCL pode trabalhar com os dois modelos de paralelismo, criando um paralelismo híbrido.

2.7 Kernel

Segundo AMD, um Kernel pode ser classificado como uma pequena unidade de execução que possui clareza em sua função e pode ser processada paralelamente, portanto o Kernel deve conter a operação necessária para obtenção dos resultados que sejam satisfatórios para o cenário do problema. Ele possui similaridade ao uso de funções na Linguagem C.[AMD]

Na Figura 11 é demonstrado um exemplo de Kernel OpenCL e na Figura 12 a mesma funcionalidade implementada na Linguagem C, uma soma entre dois vetores com o valor sendo armazenado em um terceiro vetor.

OpenCL Kernel:

Figura 11 - Kernel OpenCL que realiza a soma entre dois vetores

```
__Kernel void somaDeVetores(__global const int* a, __global const int* b,__global int* c)
{
        int id = get_global_id(0);
        c[id] = a[id] + (b[id]);

}
```

Fonte: Elaborada pelo autor.

Mesmo código escrito em Linguagem C:

Figura 12 - Exemplo da mesma operação sendo escrita na Linguagem C.

```
void somaDeVetores(int *a, int* b, int*c, int qtde)
{
        int i;

        for(i = 0 ; i < qtde ; i++)
        {
                c[i] = a[i] + b[i];
        }


}
```

Fonte: Elaborada pelo autor.

Uma questão interessante é que a Linguagem C, faz as repetições em cada posição dos vetores controlada pela variável *qtde* que está vindo por parâmetro, mas o Kernel OpenCL não precisa de tal controle pois ele trabalha em cima dos Work-Item[WOOLEY], cada Work-Item possui seu identificador e cada Work-Item fica responsável por uma instância do Kernel.

Capitulo 3 - Definição do Projeto

Neste capítulo será apresentado trabalhos correlatos encontrados e os detalhes da implementação.

3.1 Trabalhos Correlatos

Bruno Mokan Muenchen apresenta uma comparação entre aceleradores gráficos de fabricantes diferentes utilizando a biblioteca OpenCL. Utilizando as plataformas CUDA(Nvidia) e APP(AMD), ele apresenta testes que demonstram o consumo energético e o desempenho alcançado sobre as plataformas executando Benchmarks selecionados nas placas aceleradoras Nvidia, Amd e dos processadores Intel.[MUENCHEN]. A Tabela 2 traz informações sobre este trabalho.

Stefanello et al, desenvolveram um trabalho que se concentra em testes de desempenho e eficiência de algoritmos utilizando a CPU e GPU por meio de aplicações híbridas utilizando OpenMP, MPI, CUDA e OpenCl. A Tabela 2 traz informações sobre este trabalho.

Tabela 2 - Trabalhos Correlatos

Autor	Título	Plataforma
MUENCHEN, B. M	GPGPU: COMPARAÇÃO DE DESEMPENHO DE ACELERADORES AMD, NVIDIA E INTEL UTILIZANDO A BIBLIOTECA OPENCL	Placa Gráfica AMD, Nvidia e processador Intel
STEFANELLO et al	PROGRAMAÇÃO PARALELA HÍBRIDA EM CPU E GPU: UMA ALTERNATIVA NA BUSCA POR DESEMPENHO	Servidor Intel Xeon 2.4 Ghz, 12 GB Ram, Linux, GPU Nvidia Tesla M2050

Fonte: **Elaborada pelo autor.**

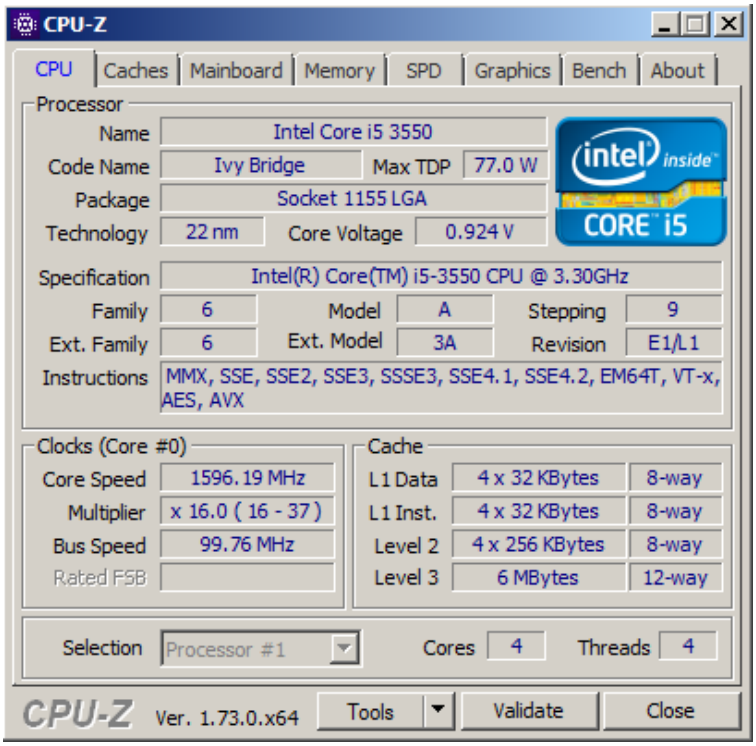
3.2 Implementação

Algumas informações sobre o computador utilizado como o processador e a placa de video, assim como a maneira que foram realizados os testes estão descritos abaixo.

Para a elaboração e execução dos testes foi utilizado o Sistema Operacional Microsoft Windows 7 Professional Edition de 64 Bits Service Pack 1.

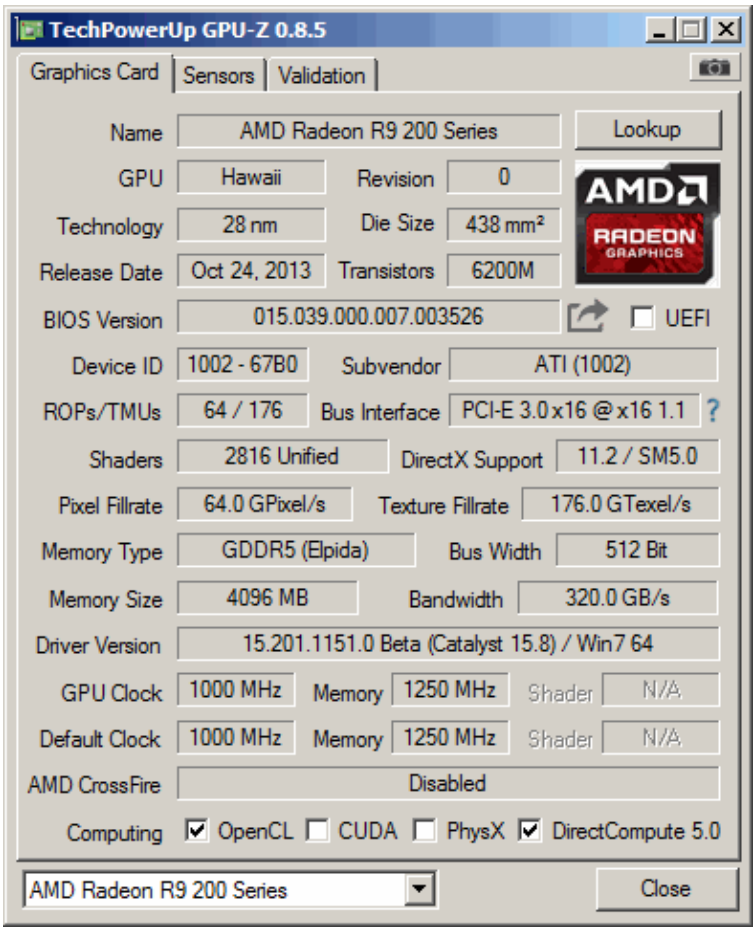
O computador utilizado para os testes possuí 8Gb de memória Ram DDR3, SSD de 480GB, a configuração da CPU e GPU foram extraídas com os software CPU-Z(www.cpuid.com/softwares/cpu-z.html) e GPU-Z(www.techpowerup.com/gpuz), a Figura 13 ilustra as informações coletadas sobre o processador(CPU) e a Figura 14 ilustra as informações coletadas sobre a placa de video(GPU).

Figura 13 - Modelo do Processador utilizado para rodar os testes.



Fonte: Elaborada pelo autor.

Figura 14 - Modelo da GPU utilizada para rodar os testes.



Fonte: **Elaborada pelo autor.**

O principal objetivo do trabalho é avaliar o desempenho da execução de um algoritmo explorando a CPU, GPU e a CPU com a GPU. Para isso foi desenvolvido um algoritmo para realizar os testes.

O algoritmo desenvolvido foi escrito na linguagem C/C++ para o Host utilizando a biblioteca OpenCL, e a plataforma escolhida foi o AMD APP, por rodar tanto na placa de video AMD de teste como no processador Intel de testes. O kernel que foi desenvolvido para o teste é de certa forma simples, mas foi elaborado com a finalidade de testar os dispositivos, a operação realizada é através de 3 vetores que o kernel recebe por parâmetro, a Figura 15 demonstra o código do Kernel utilizado.

Figura 15 - Exemplo de Kernel utilizado no trabalho.

```
__kernel void Subtracao(__global const int* a, __global const int* b,__global int* c)
 {
      int id = get_global_id(0);

      c[id] = a[id] - b[id];
 };
```

Fonte: Elaborada pelo autor.

A declaração do kernel é semelhante a uma declaração de função da Linguagem C/C++, a diferença é que utilizamos o identificador __kernel para dizer ao OpenCL que o código que segue é um kernel e deve ser processado como tal em seguida pelo nome do Kernel, a operação a ser processada pelo dispositivo OpenCL é uma subtração do vetor a pelo vetor b, o resultado será armazenado no vetor c.

Os vetores utilizados variam de tamanho a cada teste realizado. E cada Work-Item vai processar uma instância do Kernel que representa um posição vetor.

Para o teste da CPU e GPU individualmente, foram criados duas Filas de Comando, uma para cada dispositivo. Os valores de cada posição do vetor foi gerado aleatoriamente no Host e copiado para a memória do dispositivo a ser executado. Este tipo de teste foi realizado uma vez na CPU e no seu término o mesmo código foi utilizado para realizar o teste na GPU, com um mudança apenas em um dos argumentos da Função descrita a abaixo, a primeira para a ativação da CPU e a segunda para a ativação da GPU, a Figura 16 demonstra a forma como foi realizado o procedimento acima.

Figura 16 - Ativação CPU e GPU em OpenCL

```
cl_device_id  dispositivo[2];

clGetDeviceIDs(plataformaId, CL_DEVICE_TYPE_CPU, 1, &dispositivo[1], NULL);

clGetDeviceIDs(plataformaId, CL_DEVICE_TYPE_GPU, 1, &dispositivo[0], NULL);
```

Fonte: **Elaborada pelo autor.**

Para o teste da CPU e GPU foi criado um kernel para cada dispositivo que possuem a mesma operação, uma fila de comando para cada dispositivo contendo os vetores a serem processados, o tamanho de cada vetor foi alterado de acordo com o balanceamento de carga entre os dois dispositivos da forma 50/50. Afim de ser definido em que cenário é vantagem a utilização do processamento dos dois dispositivos em conjunto.

Para uma eficiente paralelização de dados, um contexto foi criado e adicionado a ele as duas filas de comando, a Figura 17 contém o trecho do código que realiza este procedimento.

Figura 17 - Criação das Filas de Comando da CPU e GPU

```
//identifica a plataforma
clGetPlatformIDs(numRequeridas ,plataformaId, &numPlataformas);

//Com a plataforma selecionada, selecionamos os dispositivos disponíveis
clGetDeviceIDs(plataformaId, CL_DEVICE_TYPE_CPU, 1, &dispositivo[1], NULL);

clGetDeviceIDs(plataformaId, CL_DEVICE_TYPE_GPU, 1, &dispositivo[0], NULL);

//cria-se o contexto com os dispositivos encontrados, dispositivoId é um vetor e os
representa
contexto = clCreateContext(0, 2, dispositivo, NULL, NULL, &erro);

//cria-se as duas filas de comando, uma para cada dispositivo
filaComando_CPU = clCreateCommandQueue(contexto, dispositivoId[1],
CL_QUEUE_PROFILING_ENABLE , &erro);

filaComando_GPU = clCreateCommandQueue(contexto, dispositivoId[0],
CL_QUEUE_PROFILING_ENABLE , &erro);

//as filas são adicionadas, e o valores dos vetores serão copiados do host para o dispositivo,
bufferA_CPU, bufferB_CPU, bufferA_GPU e bufferB_GPU são os vetores que serão
utilizados pelo Kernel

clEnqueueWriteBuffer(filaComando_CPU, bufferA_CPU, CL_TRUE, 0,
ARRAY_LENGTH_CPU * sizeof(int), hostA_CPU, 0, NULL, NULL);

clEnqueueWriteBuffer(filaComando_CPU, bufferB_CPU, CL_TRUE, 0,
ARRAY_LENGTH_CPU * sizeof(int), hostB_CPU, 0, NULL, NULL);

clEnqueueWriteBuffer(filaComando_GPU, bufferA_GPU, CL_TRUE, 0,
ARRAY_LENGTH_GPU * sizeof(int), hostA_GPU, 0, NULL, NULL);

clEnqueueWriteBuffer(filaComando_GPU, bufferB_GPU, CL_TRUE, 0,
ARRAY_LENGTH_GPU * sizeof(int), hostB_GPU, 0, NULL, NULL);
```

Fonte: **Elaborada pelo autor.**

Logo após a definição do contexto e a criação das Filas de Comando, é necessário que as Filas de Comando sejam enfileiradas nos dispositivos, a Figura 18 contém o trecho do código em que as Filas de Comando são enfileiradas nos dispositivos.

Figura 18 - Filas de comando enfileiradas nos dispositivos.

```
clEnqueueNDRangeKernel(filaComando_CPU, kernel_CPU, 1, NULL, globalsize_CPU,
NULL, 0, NULL, &evento_CPU);

clEnqueueNDRangeKernel(filaComando_CPU, kernel_GPU, 1, NULL, globalsize_CPU,
NULL, 0, NULL, &evento_CPU);
```

Fonte: Elaborada pelo autor.

Com isto o Host encaminha o Kernel para o dispositivo que cuidará da execução de todos eles.

A instrução clFinish é utilizada para a sincronização de dados, garantindo que todo o processamento necessário dos dispositivos envolvidos tenham êxito antes da finalização do processamento da Fila de Comando. Mas ele deve ser executado depois da criação das Filas de Comando de todos os dispositivos, porque ela configura a fila para que funcione corretamente de acordo com o propósito da aplicação. A Figura 19 contém o trecho do código utilizado para realizar a sincronização.

Figura 19 - Sincronização de dados

```
clFinish(filaComando_GPU);

clFinish(filaComando_CPU);
```

Fonte: Elaborada pelo autor.

Para os testes realizados, a opção escolhida foi a comparação do tempo de execução dos Kernels, e para que seja possível o cálculo do tempo a abordagem adotada foi a utilização de mecanismos disponibilizados pelo próprio OpenCL. Assim em cada Fila de Comando quando criada, é passado um evento que pode ser utilizado para consultar o status de execução da mesma, como por exemplo o tempo inicial da execução e o tempo final de execução da Fila de Comando, após a sua conclusão. A variável utilizada como para armazenar o evento precisa ser do tipo cl_event. A Figura 20 contém o trecho do código utilizado para criação do evento.

Figura 20 - Utilização de eventos na Fila de Comando.

```
cl_event evento_CPU, evento_GPU;

clEnqueueNDRangeKernel(filaComando_CPU, kernel_CPU, 1, NULL, globalsize_CPU,
NULL, 0, NULL, &evento_CPU);

clEnqueueNDRangeKernel(filaComando_GPU, kernel_GPU, 1, NULL, globalsize_GPU ,
NULL, 0, NULL, &evento_GPU);
```

Fonte: **Elaborada pelo autor.**

Para cada teste realizado, é necessário garantir que cada Work-Group termine seu processamento, através do comando clWaitForEvents é garantido que a cada processamento seja realizado até o seu término. A Figura 21 contém o trecho do código utilizado para o procedimento.

Figura 21 - Garantia que cada Work-Group termine seu processamento.

```
clWaitForEvents(1, &evento_CPU);

clWaitForEvents(1, &evento_GPU);
```

Fonte: **Elaborada pelo autor.**

Depois de terminada a execução da Fila de Comando é possível acessar os valores requeridos pelas seguintes linhas de instrução. . A Figura 22 contém o trecho do código utilizado para o procedimento.

Figura 22 - Realizando a coleta dos dados sobre o tempo de processamento do Kernel.

```
tempo = getTempo (evento);

double getTempo (cl_event event)
{
    //CL_PROFILING_COMMAND_QUEUED,
   cl_ulong start_time, end_time;


   clGetEventProfilingInfo (event, CL_PROFILING_COMMAND_SUBMIT, sizeof(cl_ulong),
            &start_time, NULL);

   clGetEventProfilingInfo (event,CL_PROFILING_COMMAND_END, sizeof(cl_ulong),
          &end_time, NULL);

   double total_time = (end_time - start_time) * 1e-6;

   return total_time;
}
```

Fonte: **Elaborada pelo autor.**

As variáveis que irão conter o tempo inicial (CL_PROFILING_COMMAND_START) e final (CL_PROFILING_COMMAND_END) da execução do Kernel, precisam ser do tipo cl_ulong e serão preenchidas na passagem delas por parâmetro na função. O resultado final (total_time_CPU e total_time_GPU) foi utilizado variáveis do tipo Double para uma maior precisão, o valor retornado é na medida de nanossegundos, sendo necessário a conversão para milissegundos para uma melhor compreensão.

Cada teste foi executado 100 vezes em cada arquitetura, os dados coletados são em função do tempo de processamento do kernel em cada arquitetura. Com isto foi calculado a média aritmética dos valores e foi calculado o valor de desvio padrão dos dados coletados, desta forma foi possível perceber a oscilação a cada teste realizado e margem de confiabilidade dos mesmos.

Capitulo 4 - Resultados

Cada teste foi executado 100 vezes, com os valores coletados foi possível realizar o cálculo da média aritmética e também foi calculado o desvio padrão amostral sobre o tempo. Essa abordagem foi utilizada porque existe sempre a utilização dos dispositivos pelo Sistema Operacional, não sendo possível executar o teste sem que algum tipo de processamento esteja ocorrendo no dispositivo fora do contexto do teste, influenciando o resultado do teste realizado. A CPU é a que mais sofre alteração como na maioria dos testes foi possível verificar com a utilização do desvio padrão, ficando a GPU de certa forma com os resultados com menos variações. Desta forma não foi considerada a utilização dos dispositivos pela Sistema Operacional, já que não é possível dedicar todo o processamento do dispositivo para o teste.

No final de cada teste foi gerado um arquivo de texto com os valores de todos os testes realizados com o adicional do desvio padrão que se encontra em anexo junto ao final do trabalho.

Os testes foram executados com o mesmo tamanho de vetor entre as arquiteturas, sendo na CPU/GPU dividida igualmente entre as duas unidades de processamento.

A operação realizada no kernel é uma subtração de um valor contido em uma posição de um vetor pelo valor contido na posição de outro vetor diferente. Os vetores são preenchidos de forma aleatória pelo próprio algoritmo.

O kernel utilizado está demonstrado na figura 23, nestes testes não há dependência de dados entre um operação e outra, o Apêndice I contém os testes realizados com dependência de dados e os resultados obtidos.

Figura 23 - Kernel utilizado para os testes sem dependência de dados.

```
__kernel void Subtracao(__global const int* a, __global const int* b,__global int* c)
{
    int id = get_global_id(0);

    c[id] = a[id] - b[id];
};
```

Fonte: Elaborada pelo autor.

4.1 Teste utilizando um vetor de 1.000 posições

O primeiro teste foi utilizado um vetor de 1.000 posições e foi obtido o resultado contido na tabela 3.

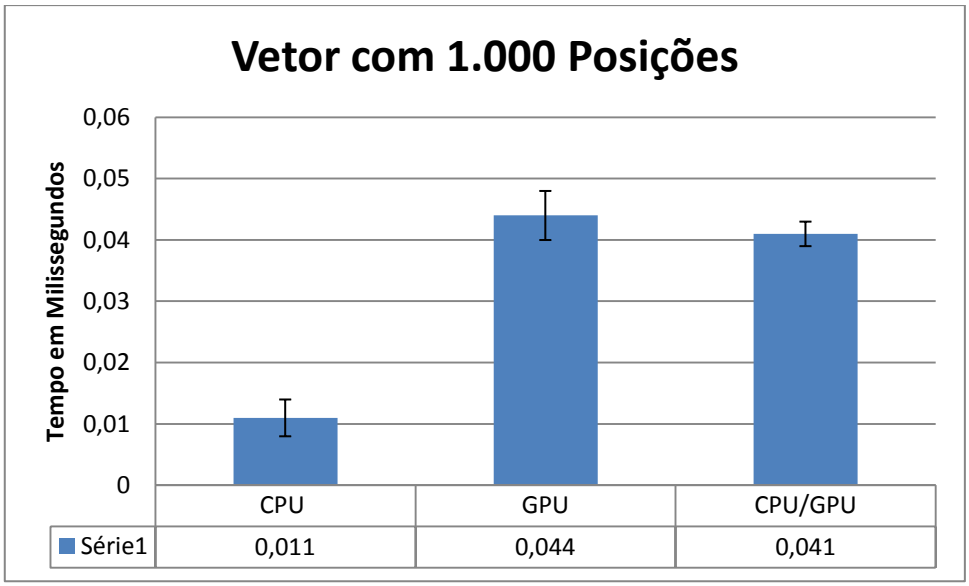
Tabela 3 - Resultado Vetor de 1.000 posições

Dispositivo	Tamanho do Vetor 1.000	Variação
CPU	0,011	0,003
GPU	0,044	0,004
CPU / GPU	0,041	0,002

Fonte: Elaborada pelo autor.

No primeiro teste realizado, a CPU conseguiu superar a GPU e a CPU/GPU. A CPU conseguiu ser aproximadamente 300% mais rápida que a GPU e aproximadamente 272% mais rápida que a CPU/GPU. Já a diferença entre a GPU e a CPU/GPU foi de aproximadamente 7%. A Figura 24 ilustra através de gráficos, os resultados obtidos.

Figura 24 - Gráfico Vetor de 1.000 Posições.



Fonte: Elaborada pelo autor.

4.2 Teste utilizando um vetor de 10.000 posições

O segundo teste foi utilizado um vetor de 10.000 posições e foi obtido o resultado contido na tabela 4.

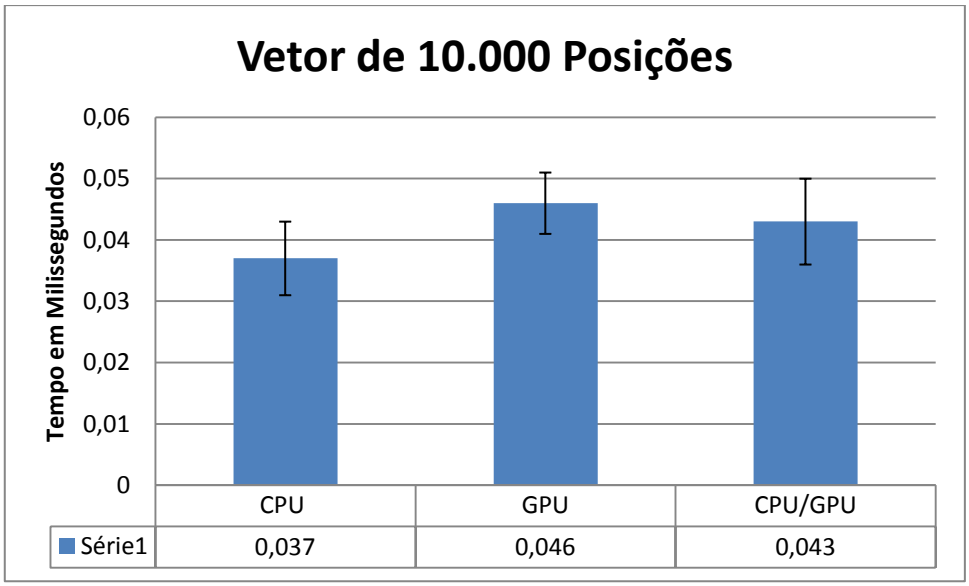
Tabela 4 -- Resultado Vetor de 10.000 posições.

Dispositivo	Tamanho do Vetor 10.000	Variação
CPU	0,037	0.006
GPU	0.046	0.005
CPU / GPU	0.043	0.007

Fonte: Elaborada pelo autor.

O teste realizado com um vetor de 10.000 posições, a CPU continua sendo mais rápida que a GPU e a CPU/GPU. A diferença entre eles foi amenizada, mais ainda existe. Neste cenário a CPU conseguiu ser aproximadamente 24% mais rápida que a GPU e aproximadamente 16% mais rápida que a CPU/GPU. A CPU/GPU conseguiu superar a GPU em aproximadamente 7% novamente. A Figura 25 ilustra através de gráficos, os resultados obtidos.

Figura 25 - Gráfico Vetor de 10.000 Posições.



Fonte: Elaborada pelo autor.

4.3 Teste utilizando um vetor de 20.000 posições

Neste foi utilizado um vetor de 20.000 posições e foi obtido o resultado contido na tabela 5.

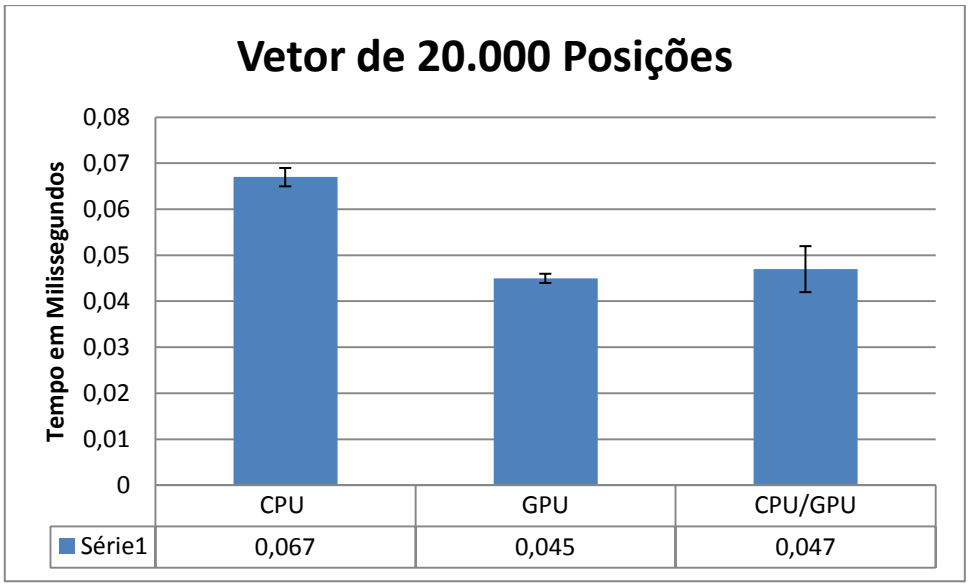
Tabela 5 - Resultado Vetor de 20.000 posições

Dispositivo	Tamanho do Vetor 20.000	Variação
CPU	0,067	0,002
GPU	0,045	0,001
CPU / GPU	0,047	0,005

Fonte: Elaborada pelo autor.

Neste teste, houve uma mudança nos resultados. A GPU começou a demonstrar que consegue trabalhar melhor com carga maiores de trabalho. A GPU superou a CPU e a CPU/GPU, chegando a ser aproximadamente 49% mais rápida que a CPU e aproximadamente 4% mais rápida que a CPU/GPU. Já a CPU/GPU superou a CPU, sendo aproximadamente 42% mais rápida que ela. Neste teste foi possível observar que a CPU/GPU está se aproximando da GPU sozinha. A Figura 26 ilustra através de gráficos, os resultados obtidos.

Figura 26 - Gráfico Vetor de 20.000 Posições.



Fonte: Elaborada pelo autor.

4.4 Teste utilizando um vetor de 22.000 posições

Como no teste de 20.000 a CPU/GPU se aproximou bastante da GPU, no quarto teste foi utilizado um vetor de 22.000 posições e foi obtido o resultado contido na tabela 6.

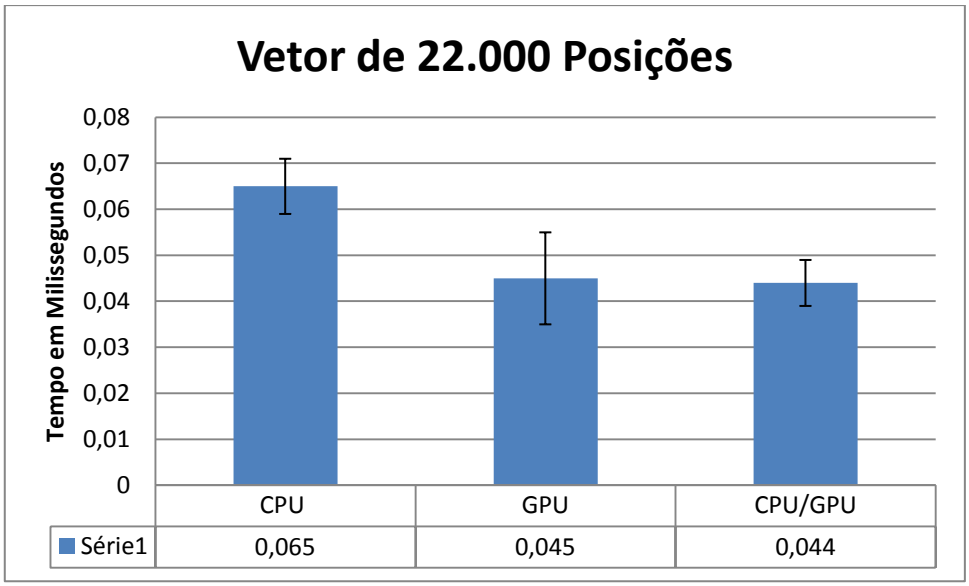
Tabela 6 - Resultado Vetor de 22.000 posições.

Dispositivo	Tamanho do Vetor 22.000	Variação
CPU	0,065	0,006
GPU	0,045	0,010
CPU / GPU	0,044	0,005

Fonte: Elaborada pelo autor.

Neste novo teste, agora com 22.000 posições, foi encontrado o ponto aproximado em que CPU/GPU superou a CPU e a GPU. A combinação CPU/GPU foi aproximadamente 35% mais rápida que CPU sozinha e aproximadamente 2% mais rápida que a GPU. A Figura 17 ilustra através de gráficos, os resultados obtidos. A Figura 27 ilustra através de gráficos, os resultados obtidos.

Figura 27 - Gráfico Vetor de 22.000 Posições.



Fonte: Elaborada pelo autor.

4.5 Teste utilizando um vetor de 24.000 posições

Como no teste de 22.000 a CPU/GPU conseguiu um tempo menor que as outras arquiteturas, o teste agora foi feito com 24.000 posições e foi obtido o resultado contido na tabela 7.

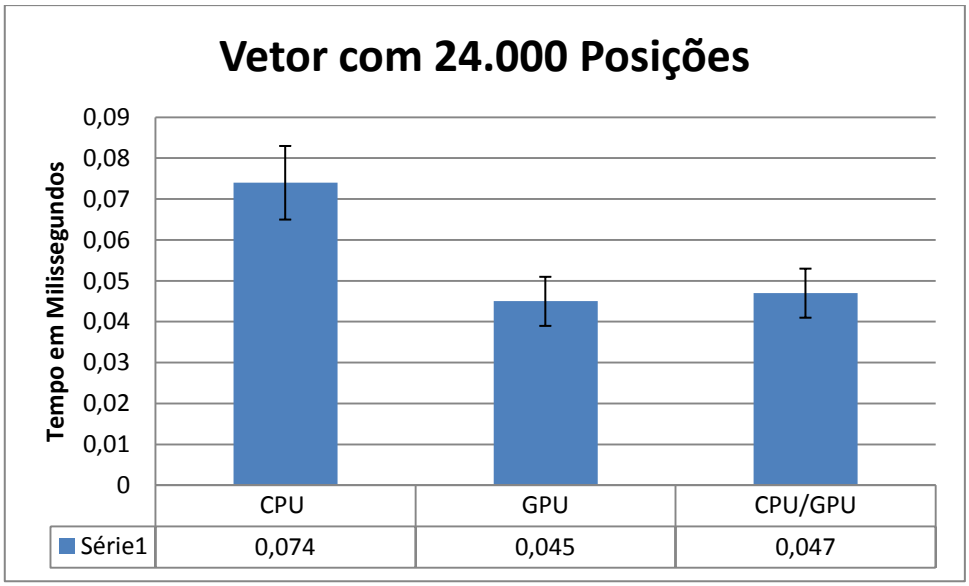
Tabela 7 - Resultado Vetor de 24.000 posições.

Dispositivo	Tamanho do Vetor 24.000	Variação
CPU	0,074	0,009
GPU	0,045	0,006
CPU / GPU	0,047	0,006

Fonte: Elaborada pelo autor.

Neste teste, a o tempo que CPU/GPU utilizou para processar o vetor, começou aumentar novamente. O que demonstra que o teste anterior se mostrou mais propício para o cenário CPU/GPU. A CPU foi a que mais demorou para processar o vetor. A Figura 28 ilustra através de gráficos, os resultados obtidos.

Figura 28 - Gráfico Vetor de 24.000 Posições.



Fonte: Elaborada pelo autor.

4.6 Teste utilizando um vetor de 100.000 posições

Para o quinto teste foi utilizado um vetor de 100.000 posições e foi obtido o resultado contido na tabela 8.

Tabela 8 - Resultado Vetor de 100.000 posições.

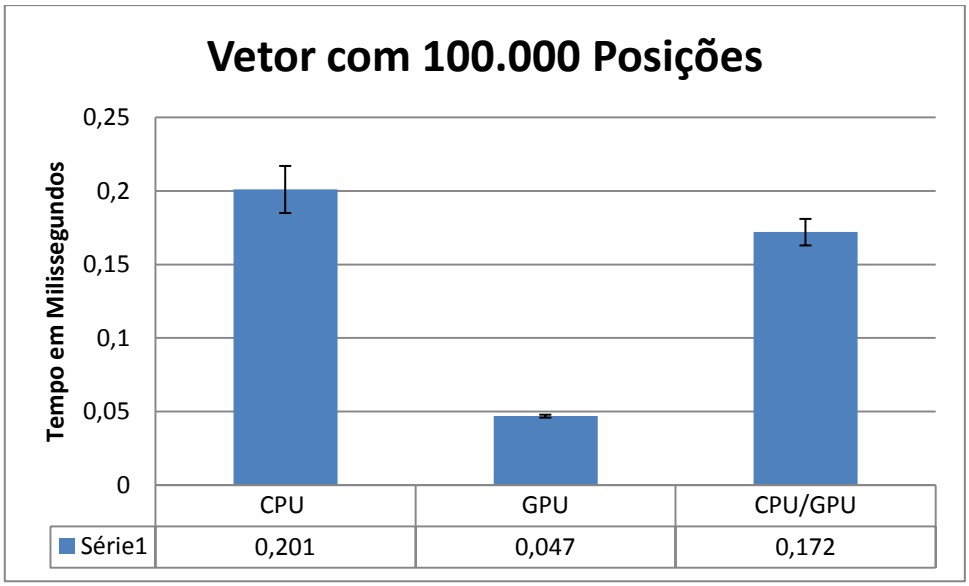
Dispositivo	Tamanho do Vetor 100.000	Variação
CPU	0,201	0,016
GPU	0,047	0,001
CPU / GPU	0,172	0,009

Fonte: Elaborada pelo autor.

Neste teste a GPU voltou a ser mais rápida que as outras arquiteturas. Ela chegou a ser aproximadamente 327% mais rápida que a CPU e aproximadamente 266% mais rápida que CPU/GPU. A CPU/GPU conseguiu ser aproximadamente 17% mais rápida que a CPU.

Novamente em um cenário que possui uma carga maior, a GPU conseguiu realizar o processamento sem que houvesse uma alteração muito grande no tempo que ela levou para processar. A Figura 29 ilustra através de gráficos, os resultados obtidos.

Figura 29 - Gráfico Vetor de 100.000 Posições.



Fonte: Elaborada pelo autor.

4.7 Teste utilizando um vetor de 1.000.000 posições

Para o sexto teste, o vetor teve um aumento proposital para que fosse possível a análise em cima das três arquitetura, o vetor utilizado possui 1.000.000 posições e foi obtido o resultado contido na tabela 9.

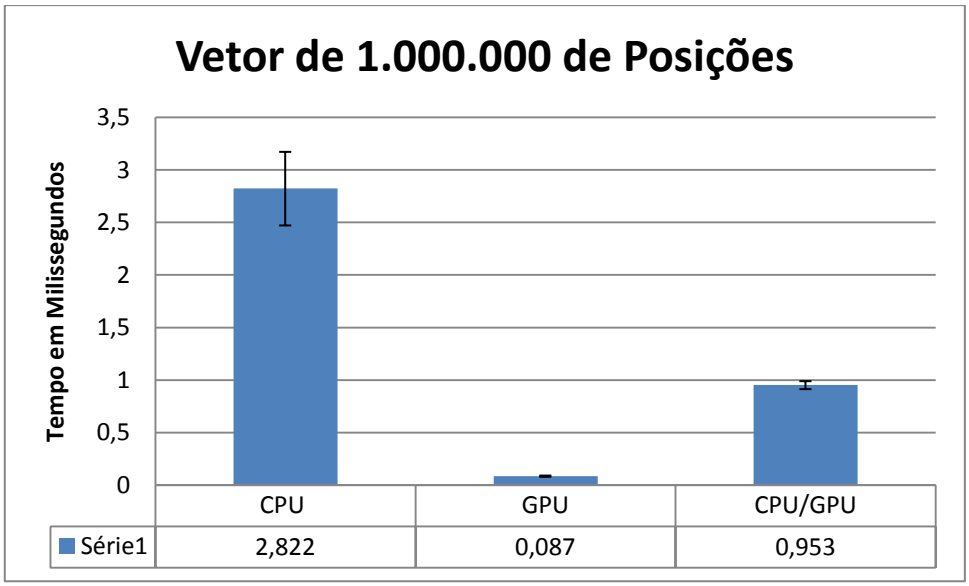
Tabela 9 - Resultado Vetor de 1.000.000 posições.

Dispositivo	Tamanho do Vetor 1.000.000	Variação
CPU	2,822	0,350
GPU	0,087	0,006
CPU / GPU	0,953	0,038

Fonte: Elaborada pelo autor.

Foi possível observar um aumento considerável no tempo da CPU para processar o vetor, a GPU por sua vez, não teve um aumento na mesma proporção e superou a CPU e a CPU/GPU. A GPU conseguiu ser aproximadamente 3.100% mais rápida que CPU e 995% mais rápida que CPU/GPU. A CPU/GPU superou em aproximadamente 196% a CPU. A Figura 30 ilustra através de gráficos, os resultados obtidos.

Figura 30 - Gráfico Vetor de 1.000.000 Posições.



Fonte: Elaborada pelo autor.

4.8 Teste utilizando um vetor de 10.000.000 posições

O último teste foi utilizado um vetor de 10.000.000 posições e foi obtido o resultado contido na tabela 10.

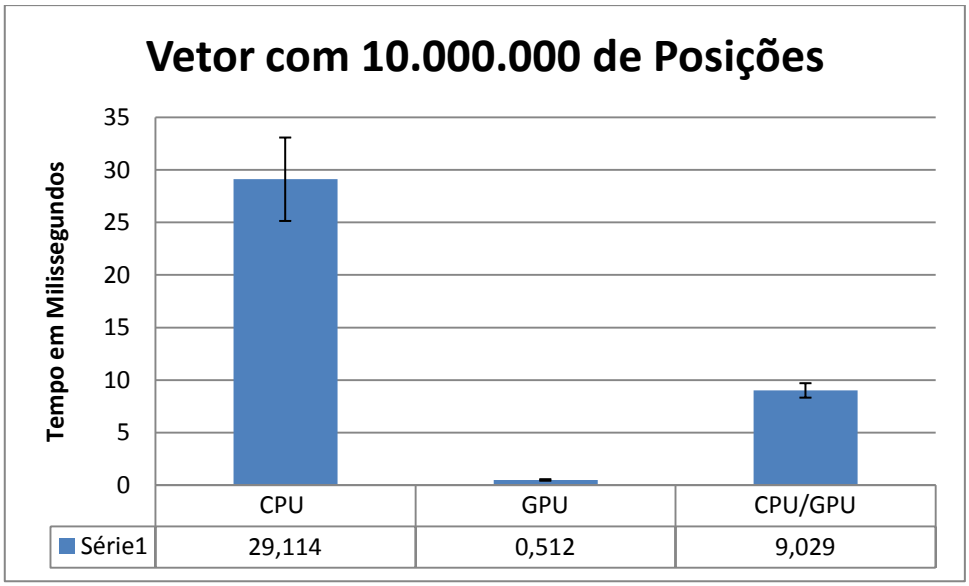
Tabela 10 - Resultado Vetor de 10.000.000 posições.

Dispositivo	Tamanho do Vetor 10.000.000	Variação
CPU	29,114	3,968
GPU	0,512	0,073
CPU / GPU	9,029	0,687

Fonte: Elaborada pelo autor.

No último teste a diferença entre a GPU e as outras duas arquiteturas somente aumentou. A GPU foi aproximadamente 5.587% mais rápida que a CPU e 1.663% mais rápida que a CPU/GPU. A CPU/GPU conseguiu ser aproximadamente 222% mais rápida que a CPU. A Figura 31 ilustra através de gráficos, os resultados obtidos.

Figura 31 - Gráfico Vetor de 10.000.000 Posições.



Fonte: Elaborada pelo autor.

Capítulo 5

5.1 Conclusão

Com o resultado dos testes fica-se entendido que vetores pequenos não são interessantes de serem processados pela GPU, já com vetores de tamanho grande fica num cenário mais interessante a utilização da GPU.

No caso da CPU ela demonstrou nos testes que consegue trabalhar muito bem com vetores pequenos mas que ao aumentar o tamanho do vetor como por exemplo acima de vinte mil posições, ela demonstra que precisa de mais tempo para processar o vetor que as outras arquiteturas, ficando definido que a para a CPU é interessante o processamento de vetores pequenos.

No caso da união das arquiteturas foi concluído que ela consegue processar os vetores propostos de forma a ficar entre a CPU e a GPU, mas que em um determinado cenário ela consegue se destacar mais que as outras arquiteturas. Por isso foi realizado mais alguns testes com vetores de tamanho próximo a vinte mil, pois neste tamanho de vetor as arquiteturas processaram o vetor com um tempo muito próximo uma da outra. Com isto chegou-se a um valor aproximado de vetor em que a CPU/GPU conseguiu processar o vetor em um tempo menor que as arquiteturas da CPU e GPU.

5.2 Proposta de trabalhos futuros

De acordo com o que foi realizado neste trabalho, foi possível verificar as diferenças entre as arquiteturas envolvidas, ficando sugerido para trabalhos futuros o balanceamento de cargas dos testes realizados, ou mesmo o aumento da complexidade do algoritmo visto que ele é de certa forma bem simples.

Referências Bibliográficas

TANENBAUM, A. S. Organização Estruturada de Computadores , 5º ed. Rio de Janeiro, Prentice-Hall,2007

STALLINGS, W. Arquitetura e Organização de Computadores, 8º ed. Rio de Janeiro, Prentice-Hall, 2008.

PINTO, G. V. Ambientes de Programação Paralela Híbrida. 2011. Dissertação. Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 2011.

GRAMMELSBACHER, A. V. , MEDRADO, J. C. C. COMPARAÇÃO DE DESEMPENHO ENTRE GPGPU E SISTEMAS PARALELOS, 2009.
Dissertação.Universidade Anhembi Morumbi, São Paulo, 2009.

KIRK, D. e HWU, W. Programando para Processadores Paralelos: uma abordagem prática à programação de GPU. Rio de Janeiro, Elsevier, 2011.

MORIMOTO, C. E. OpenCl, Cuda e Brook: Processamento usando a GPU. Disponível em : <http://www.hardware.com.br/dicas/opencl.html>, visitado em 02/07/2015.

KHRONOS GROUP.The open standard for parallel programming of heterogeneous systems.Disponível em : <https://www.khronos.org/opencl/>, visitado em 18/07/2015.

Khronos OpenCL Working Group, The OpenCL Specification version 2.1 , 2015. Disponível em < https://www.khronos.org/registry/cl/specs/opencl-2.1.pdf>, visitado em 08/08/2015.

TAY, R. OpenCL Parallel Programming Development CookBook, 1º ed. Birminghan, Packt Publishing, 2013.

SOUZA, A. M. COMPARAÇÃO DE DESEMPENHO ENTRE DIFERENTES IMPLEMENTAÇÕES DO ALGORITMO KECCAK PARA PLATAFORMAS GPGPUS UTILIZANDO OPENCL. Monografia. Universidade Univem, Marília, 2013.

MUENCHEN, B. M. GPGPU: COMPARAÇÃO DE ACELERADORES AMD, NIVIDIA E INTEL UTILIZANDO A BIBLIOTECA OPENCL. Monografia. Universidade Regional do Noroeste do Estado do Rio Grande do Sul - UNIJUI. Ijuí, 2013.

AMD. AMD Accelerated Parallel Processing. Disponível em : <http://amd-dev.wpengine.netdnacdn.com/wordpress/media/2013/07/AMD_Accelerated_Parallel_Processing_OpenCL_Programming_Guide-rev-2.7.pdf>, visitado em 10/10/2015.

WOOLLEY C., INTRODUCTION TO OPENCL. Disponível em <http://www.cc.gatech.edu/~vetter/keeneland/tutorial-2011-04-14/06-intro_to_opencl.pdf>, visitado em 03/09/2015.

FILHO O. H.B, COMPONENTES ELETRÔNICOS E UNIDADES DE MEDIDA, CONCEITOS BÁSICOS.
Disponível em < http://www.hardware.com.br/tutoriais/componentes-eletronicos-unidades-medida-conceitos-basicos/componentes-eletronicos.html>, visitado em 03/12/2015.

STEFANELLO, A. L.; MACHADO, C. C.; ROSA D., SULZBACH, M.; MOERSCHBACHER, R. W.; SARTURI, T. R. PROGRAMAÇÃO PARALELA HÍBRIDA EM CPU E GPU: UMA ALTERNATIVA NA BUSCA POR DESEMPENHO. Anais do EATI, Frederico Westphalen – RS – Brasil, ANO 3 n. 1, PÁG. 124 - 131, NOVEMBRO 2013.

STONE, J. E.; GOHARA, D.; SHI, G. OPENCL: A PARALLEL PROGRAMMING STANDART FOR HETEROGENEOUS COMPUTING SYSTEMS. IEEE, Los Alamitos, CA, USA, Volume 12 Issue 3, Pág. 66-73, Maio 2010.

NVIDIA. CUDA TOOLKIT DOCUMENTATION. Disponível em <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#axzz3tSUYoOH1>, visitado em 03/12/2015.

Apêndice A - Resultados Vetor 1.000

CPU = 1000	GPU = 1000	CPU/GPU = 1000(500/500)
==========	==========	==========
teste 1 CPU --> 0.016 ||	teste 1 GPU --> 0.065 ||	teste 1 CPU/GPU --> 0.056
teste 2 CPU --> 0.015 ||	teste 2 GPU --> 0.058 ||	teste 2 CPU/GPU --> 0.044
teste 3 CPU --> 0.014 ||	teste 3 GPU --> 0.053 ||	teste 3 CPU/GPU --> 0.042
teste 4 CPU --> 0.014 ||	teste 4 GPU --> 0.047 ||	teste 4 CPU/GPU --> 0.041
teste 5 CPU --> 0.014 ||	teste 5 GPU --> 0.043 ||	teste 5 CPU/GPU --> 0.041
teste 6 CPU --> 0.012 ||	teste 6 GPU --> 0.043 ||	teste 6 CPU/GPU --> 0.041
teste 7 CPU --> 0.024 ||	teste 7 GPU --> 0.042 ||	teste 7 CPU/GPU --> 0.041
teste 8 CPU --> 0.022 ||	teste 8 GPU --> 0.043 ||	teste 8 CPU/GPU --> 0.040
teste 9 CPU --> 0.025 ||	teste 9 GPU --> 0.043 ||	teste 9 CPU/GPU --> 0.040
teste 10 CPU --> 0.010 ||	teste 10 GPU --> 0.043 ||	teste 10 CPU/GPU --> 0.041
teste 11 CPU --> 0.010 ||	teste 11 GPU --> 0.042 ||	teste 11 CPU/GPU --> 0.040
teste 12 CPU --> 0.010 ||	teste 12 GPU --> 0.045 ||	teste 12 CPU/GPU --> 0.042
teste 13 CPU --> 0.011 ||	teste 13 GPU --> 0.043 ||	teste 13 CPU/GPU --> 0.041
teste 14 CPU --> 0.010 ||	teste 14 GPU --> 0.043 ||	teste 14 CPU/GPU --> 0.042
teste 15 CPU --> 0.009 ||	teste 15 GPU --> 0.044 ||	teste 15 CPU/GPU --> 0.042
teste 16 CPU --> 0.010 ||	teste 16 GPU --> 0.043 ||	teste 16 CPU/GPU --> 0.040
teste 17 CPU --> 0.010 ||	teste 17 GPU --> 0.042 ||	teste 17 CPU/GPU --> 0.040
teste 18 CPU --> 0.010 ||	teste 18 GPU --> 0.042 ||	teste 18 CPU/GPU --> 0.041
teste 19 CPU --> 0.010 ||	teste 19 GPU --> 0.043 ||	teste 19 CPU/GPU --> 0.041
teste 20 CPU --> 0.010 ||	teste 20 GPU --> 0.042 ||	teste 20 CPU/GPU --> 0.040
teste 21 CPU --> 0.010 ||	teste 21 GPU --> 0.042 ||	teste 21 CPU/GPU --> 0.040
teste 22 CPU --> 0.009 ||	teste 22 GPU --> 0.042 ||	teste 22 CPU/GPU --> 0.041
teste 23 CPU --> 0.009 ||	teste 23 GPU --> 0.044 ||	teste 23 CPU/GPU --> 0.040
teste 24 CPU --> 0.010 ||	teste 24 GPU --> 0.044 ||	teste 24 CPU/GPU --> 0.041
teste 25 CPU --> 0.009 ||	teste 25 GPU --> 0.042 ||	teste 25 CPU/GPU --> 0.040
teste 26 CPU --> 0.010 ||	teste 26 GPU --> 0.044 ||	teste 26 CPU/GPU --> 0.040
teste 27 CPU --> 0.010 ||	teste 27 GPU --> 0.043 ||	teste 27 CPU/GPU --> 0.040
teste 28 CPU --> 0.011 ||	teste 28 GPU --> 0.042 ||	teste 28 CPU/GPU --> 0.041
teste 29 CPU --> 0.010 ||	teste 29 GPU --> 0.042 ||	teste 29 CPU/GPU --> 0.043
teste 30 CPU --> 0.011 ||	teste 30 GPU --> 0.044 ||	teste 30 CPU/GPU --> 0.042
teste 31 CPU --> 0.011 ||	teste 31 GPU --> 0.042 ||	teste 31 CPU/GPU --> 0.041
teste 32 CPU --> 0.010 ||	teste 32 GPU --> 0.043 ||	teste 32 CPU/GPU --> 0.042
teste 33 CPU --> 0.009 ||	teste 33 GPU --> 0.044 ||	teste 33 CPU/GPU --> 0.040
teste 34 CPU --> 0.009 ||	teste 34 GPU --> 0.043 ||	teste 34 CPU/GPU --> 0.041
teste 35 CPU --> 0.010 ||	teste 35 GPU --> 0.049 ||	teste 35 CPU/GPU --> 0.040
teste 36 CPU --> 0.010 ||	teste 36 GPU --> 0.044 ||	teste 36 CPU/GPU --> 0.041
teste 37 CPU --> 0.011 ||	teste 37 GPU --> 0.046 ||	teste 37 CPU/GPU --> 0.040
teste 38 CPU --> 0.010 ||	teste 38 GPU --> 0.043 ||	teste 38 CPU/GPU --> 0.041
teste 39 CPU --> 0.009 ||	teste 39 GPU --> 0.042 ||	teste 39 CPU/GPU --> 0.040
teste 40 CPU --> 0.009 ||	teste 40 GPU --> 0.042 ||	teste 40 CPU/GPU --> 0.040
teste 41 CPU --> 0.012 ||	teste 41 GPU --> 0.043 ||	teste 41 CPU/GPU --> 0.041
teste 42 CPU --> 0.010 ||	teste 42 GPU --> 0.042 ||	teste 42 CPU/GPU --> 0.041
teste 43 CPU --> 0.015 ||	teste 43 GPU --> 0.042 ||	teste 43 CPU/GPU --> 0.040
teste 44 CPU --> 0.014 ||	teste 44 GPU --> 0.044 ||	teste 44 CPU/GPU --> 0.040
teste 45 CPU --> 0.014 ||	teste 45 GPU --> 0.042 ||	teste 45 CPU/GPU --> 0.040
teste 46 CPU --> 0.016 ||	teste 46 GPU --> 0.044 ||	teste 46 CPU/GPU --> 0.042
teste 47 CPU --> 0.017 ||	teste 47 GPU --> 0.042 ||	teste 47 CPU/GPU --> 0.041
teste 48 CPU --> 0.013 ||	teste 48 GPU --> 0.044 ||	teste 48 CPU/GPU --> 0.041
teste 49 CPU --> 0.012 ||	teste 49 GPU --> 0.042 ||	teste 49 CPU/GPU --> 0.042
teste 50 CPU --> 0.012 ||	teste 50 GPU --> 0.042 ||	teste 50 CPU/GPU --> 0.041
teste 51 CPU --> 0.011 ||	teste 51 GPU --> 0.042 ||	teste 51 CPU/GPU --> 0.040
teste 52 CPU --> 0.010 ||	teste 52 GPU --> 0.043 ||	teste 52 CPU/GPU --> 0.040
teste 53 CPU --> 0.010 ||	teste 53 GPU --> 0.042 ||	teste 53 CPU/GPU --> 0.041

teste 54 CPU --> 0.010 ||	teste 54 GPU --> 0.042 ||	teste 54 CPU/GPU --> 0.041
teste 55 CPU --> 0.010 ||	teste 55 GPU --> 0.043 ||	teste 55 CPU/GPU --> 0.040
teste 56 CPU --> 0.010 ||	teste 56 GPU --> 0.043 ||	teste 56 CPU/GPU --> 0.041
teste 57 CPU --> 0.010 ||	teste 57 GPU --> 0.053 ||	teste 57 CPU/GPU --> 0.040
teste 58 CPU --> 0.010 ||	teste 58 GPU --> 0.062 ||	teste 58 CPU/GPU --> 0.041
teste 59 CPU --> 0.010 ||	teste 59 GPU --> 0.050 ||	teste 59 CPU/GPU --> 0.040
teste 60 CPU --> 0.010 ||	teste 60 GPU --> 0.044 ||	teste 60 CPU/GPU --> 0.040
teste 61 CPU --> 0.010 ||	teste 61 GPU --> 0.042 ||	teste 61 CPU/GPU --> 0.040
teste 62 CPU --> 0.010 ||	teste 62 GPU --> 0.043 ||	teste 62 CPU/GPU --> 0.040
teste 63 CPU --> 0.010 ||	teste 63 GPU --> 0.042 ||	teste 63 CPU/GPU --> 0.042
teste 64 CPU --> 0.010 ||	teste 64 GPU --> 0.043 ||	teste 64 CPU/GPU --> 0.042
teste 65 CPU --> 0.010 ||	teste 65 GPU --> 0.042 ||	teste 65 CPU/GPU --> 0.041
teste 66 CPU --> 0.010 ||	teste 66 GPU --> 0.042 ||	teste 66 CPU/GPU --> 0.043
teste 67 CPU --> 0.010 ||	teste 67 GPU --> 0.043 ||	teste 67 CPU/GPU --> 0.041
teste 68 CPU --> 0.011 ||	teste 68 GPU --> 0.043 ||	teste 68 CPU/GPU --> 0.040
teste 69 CPU --> 0.010 ||	teste 69 GPU --> 0.042 ||	teste 69 CPU/GPU --> 0.040
teste 70 CPU --> 0.010 ||	teste 70 GPU --> 0.043 ||	teste 70 CPU/GPU --> 0.041
teste 71 CPU --> 0.010 ||	teste 71 GPU --> 0.042 ||	teste 71 CPU/GPU --> 0.040
teste 72 CPU --> 0.010 ||	teste 72 GPU --> 0.043 ||	teste 72 CPU/GPU --> 0.041
teste 73 CPU --> 0.010 ||	teste 73 GPU --> 0.041 ||	teste 73 CPU/GPU --> 0.040
teste 74 CPU --> 0.011 ||	teste 74 GPU --> 0.042 ||	teste 74 CPU/GPU --> 0.041
teste 75 CPU --> 0.010 ||	teste 75 GPU --> 0.042 ||	teste 75 CPU/GPU --> 0.040
teste 76 CPU --> 0.009 ||	teste 76 GPU --> 0.042 ||	teste 76 CPU/GPU --> 0.040
teste 77 CPU --> 0.009 ||	teste 77 GPU --> 0.044 ||	teste 77 CPU/GPU --> 0.040
teste 78 CPU --> 0.009 ||	teste 78 GPU --> 0.043 ||	teste 78 CPU/GPU --> 0.041
teste 79 CPU --> 0.009 ||	teste 79 GPU --> 0.043 ||	teste 79 CPU/GPU --> 0.040
teste 80 CPU --> 0.011 ||	teste 80 GPU --> 0.042 ||	teste 80 CPU/GPU --> 0.042
teste 81 CPU --> 0.010 ||	teste 81 GPU --> 0.044 ||	teste 81 CPU/GPU --> 0.042
teste 82 CPU --> 0.011 ||	teste 82 GPU --> 0.042 ||	teste 82 CPU/GPU --> 0.041
teste 83 CPU --> 0.010 ||	teste 83 GPU --> 0.042 ||	teste 83 CPU/GPU --> 0.042
teste 84 CPU --> 0.010 ||	teste 84 GPU --> 0.042 ||	teste 84 CPU/GPU --> 0.041
teste 85 CPU --> 0.010 ||	teste 85 GPU --> 0.042 ||	teste 85 CPU/GPU --> 0.040
teste 86 CPU --> 0.010 ||	teste 86 GPU --> 0.042 ||	teste 86 CPU/GPU --> 0.041
teste 87 CPU --> 0.010 ||	teste 87 GPU --> 0.042 ||	teste 87 CPU/GPU --> 0.040
teste 88 CPU --> 0.010 ||	teste 88 GPU --> 0.043 ||	teste 88 CPU/GPU --> 0.041
teste 89 CPU --> 0.010 ||	teste 89 GPU --> 0.042 ||	teste 89 CPU/GPU --> 0.040
teste 90 CPU --> 0.010 ||	teste 90 GPU --> 0.045 ||	teste 90 CPU/GPU --> 0.041
teste 91 CPU --> 0.009 ||	teste 91 GPU --> 0.042 ||	teste 91 CPU/GPU --> 0.040
teste 92 CPU --> 0.010 ||	teste 92 GPU --> 0.044 ||	teste 92 CPU/GPU --> 0.041
teste 93 CPU --> 0.010 ||	teste 93 GPU --> 0.043 ||	teste 93 CPU/GPU --> 0.041
teste 94 CPU --> 0.010 ||	teste 94 GPU --> 0.042 ||	teste 94 CPU/GPU --> 0.040
teste 95 CPU --> 0.010 ||	teste 95 GPU --> 0.042 ||	teste 95 CPU/GPU --> 0.040
teste 96 CPU --> 0.010 ||	teste 96 GPU --> 0.043 ||	teste 96 CPU/GPU --> 0.043
teste 97 CPU --> 0.009 ||	teste 97 GPU --> 0.042 ||	teste 97 CPU/GPU --> 0.042
teste 98 CPU --> 0.009 ||	teste 98 GPU --> 0.042 ||	teste 98 CPU/GPU --> 0.042
teste 99 CPU --> 0.011 ||	teste 99 GPU --> 0.042 ||	teste 99 CPU/GPU --> 0.041
teste 100 CPU --> 0.010 ||	teste 100 GPU --> 0.042 ||	teste 100 CPU/GPU --> 0.043

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.011 (+ ou -) 0.003	0.044 (+ ou -) 0.004	0.041 (+ ou -) 0.002

Apêndice B - Resultados Vetor 10.000

CPU = 10000	GPU = 10000	CPU/GPU = 10000(5000/5000)
==========	==========	==========
teste 1 CPU --> 0.081	teste 1 GPU --> 0.067	teste 1 CPU/GPU --> 0.104
teste 2 CPU --> 0.045	teste 2 GPU --> 0.062	teste 2 CPU/GPU --> 0.059
teste 3 CPU --> 0.038	teste 3 GPU --> 0.047	teste 3 CPU/GPU --> 0.046
teste 4 CPU --> 0.054	teste 4 GPU --> 0.050	teste 4 CPU/GPU --> 0.043
teste 5 CPU --> 0.058	teste 5 GPU --> 0.048	teste 5 CPU/GPU --> 0.042
teste 6 CPU --> 0.040	teste 6 GPU --> 0.048	teste 6 CPU/GPU --> 0.043
teste 7 CPU --> 0.055	teste 7 GPU --> 0.047	teste 7 CPU/GPU --> 0.042
teste 8 CPU --> 0.041	teste 8 GPU --> 0.059	teste 8 CPU/GPU --> 0.042
teste 9 CPU --> 0.039	teste 9 GPU --> 0.062	teste 9 CPU/GPU --> 0.042
teste 10 CPU --> 0.038	teste 10 GPU --> 0.049	teste 10 CPU/GPU --> 0.042
teste 11 CPU --> 0.050	teste 11 GPU --> 0.045	teste 11 CPU/GPU --> 0.042
teste 12 CPU --> 0.039	teste 12 GPU --> 0.047	teste 12 CPU/GPU --> 0.045
teste 13 CPU --> 0.036	teste 13 GPU --> 0.055	teste 13 CPU/GPU --> 0.045
teste 14 CPU --> 0.038	teste 14 GPU --> 0.047	teste 14 CPU/GPU --> 0.043
teste 15 CPU --> 0.035	teste 15 GPU --> 0.055	teste 15 CPU/GPU --> 0.042
teste 16 CPU --> 0.036	teste 16 GPU --> 0.046	teste 16 CPU/GPU --> 0.042
teste 17 CPU --> 0.036	teste 17 GPU --> 0.054	teste 17 CPU/GPU --> 0.042
teste 18 CPU --> 0.037	teste 18 GPU --> 0.046	teste 18 CPU/GPU --> 0.043
teste 19 CPU --> 0.037	teste 19 GPU --> 0.048	teste 19 CPU/GPU --> 0.042
teste 20 CPU --> 0.036	teste 20 GPU --> 0.045	teste 20 CPU/GPU --> 0.044
teste 21 CPU --> 0.037	teste 21 GPU --> 0.042	teste 21 CPU/GPU --> 0.044
teste 22 CPU --> 0.036	teste 22 GPU --> 0.042	teste 22 CPU/GPU --> 0.044
teste 23 CPU --> 0.036	teste 23 GPU --> 0.044	teste 23 CPU/GPU --> 0.045
teste 24 CPU --> 0.035	teste 24 GPU --> 0.042	teste 24 CPU/GPU --> 0.042
teste 25 CPU --> 0.036	teste 25 GPU --> 0.043	teste 25 CPU/GPU --> 0.042
teste 26 CPU --> 0.053	teste 26 GPU --> 0.042	teste 26 CPU/GPU --> 0.042
teste 27 CPU --> 0.036	teste 27 GPU --> 0.041	teste 27 CPU/GPU --> 0.042
teste 28 CPU --> 0.036	teste 28 GPU --> 0.041	teste 28 CPU/GPU --> 0.042
teste 29 CPU --> 0.036	teste 29 GPU --> 0.041	teste 29 CPU/GPU --> 0.042
teste 30 CPU --> 0.035	teste 30 GPU --> 0.043	teste 30 CPU/GPU --> 0.042
teste 31 CPU --> 0.036	teste 31 GPU --> 0.050	teste 31 CPU/GPU --> 0.043
teste 32 CPU --> 0.035	teste 32 GPU --> 0.043	teste 32 CPU/GPU --> 0.042
teste 33 CPU --> 0.035	teste 33 GPU --> 0.046	teste 33 CPU/GPU --> 0.042
teste 34 CPU --> 0.036	teste 34 GPU --> 0.049	teste 34 CPU/GPU --> 0.042
teste 35 CPU --> 0.036	teste 35 GPU --> 0.048	teste 35 CPU/GPU --> 0.042
teste 36 CPU --> 0.035	teste 36 GPU --> 0.044	teste 36 CPU/GPU --> 0.042
teste 37 CPU --> 0.035	teste 37 GPU --> 0.047	teste 37 CPU/GPU --> 0.042
teste 38 CPU --> 0.035	teste 38 GPU --> 0.046	teste 38 CPU/GPU --> 0.042
teste 39 CPU --> 0.037	teste 39 GPU --> 0.046	teste 39 CPU/GPU --> 0.042
teste 40 CPU --> 0.036	teste 40 GPU --> 0.045	teste 40 CPU/GPU --> 0.042
teste 41 CPU --> 0.036	teste 41 GPU --> 0.044	teste 41 CPU/GPU --> 0.041
teste 42 CPU --> 0.036	teste 42 GPU --> 0.047	teste 42 CPU/GPU --> 0.042
teste 43 CPU --> 0.035	teste 43 GPU --> 0.046	teste 43 CPU/GPU --> 0.042
teste 44 CPU --> 0.036	teste 44 GPU --> 0.045	teste 44 CPU/GPU --> 0.042
teste 45 CPU --> 0.036	teste 45 GPU --> 0.044	teste 45 CPU/GPU --> 0.042
teste 46 CPU --> 0.036	teste 46 GPU --> 0.046	teste 46 CPU/GPU --> 0.042
teste 47 CPU --> 0.035	teste 47 GPU --> 0.049	teste 47 CPU/GPU --> 0.042
teste 48 CPU --> 0.036	teste 48 GPU --> 0.045	teste 48 CPU/GPU --> 0.043
teste 49 CPU --> 0.038	teste 49 GPU --> 0.044	teste 49 CPU/GPU --> 0.042
teste 50 CPU --> 0.036	teste 50 GPU --> 0.047	teste 50 CPU/GPU --> 0.042
teste 51 CPU --> 0.036	teste 51 GPU --> 0.046	teste 51 CPU/GPU --> 0.042
teste 52 CPU --> 0.036	teste 52 GPU --> 0.046	teste 52 CPU/GPU --> 0.042
teste 53 CPU --> 0.036	teste 53 GPU --> 0.044	teste 53 CPU/GPU --> 0.043
teste 54 CPU --> 0.036	teste 54 GPU --> 0.053	teste 54 CPU/GPU --> 0.042
teste 55 CPU --> 0.035	teste 55 GPU --> 0.045	teste 55 CPU/GPU --> 0.042

teste 56 CPU --> 0.036 ||	teste 56 GPU --> 0.046 ||	teste 56 CPU/GPU --> 0.042
teste 57 CPU --> 0.037 ||	teste 57 GPU --> 0.048 ||	teste 57 CPU/GPU --> 0.042
teste 58 CPU --> 0.037 ||	teste 58 GPU --> 0.046 ||	teste 58 CPU/GPU --> 0.043
teste 59 CPU --> 0.036 ||	teste 59 GPU --> 0.048 ||	teste 59 CPU/GPU --> 0.042
teste 60 CPU --> 0.037 ||	teste 60 GPU --> 0.046 ||	teste 60 CPU/GPU --> 0.042
teste 61 CPU --> 0.036 ||	teste 61 GPU --> 0.050 ||	teste 61 CPU/GPU --> 0.042
teste 62 CPU --> 0.037 ||	teste 62 GPU --> 0.049 ||	teste 62 CPU/GPU --> 0.042
teste 63 CPU --> 0.037 ||	teste 63 GPU --> 0.048 ||	teste 63 CPU/GPU --> 0.042
teste 64 CPU --> 0.036 ||	teste 64 GPU --> 0.044 ||	teste 64 CPU/GPU --> 0.042
teste 65 CPU --> 0.038 ||	teste 65 GPU --> 0.043 ||	teste 65 CPU/GPU --> 0.042
teste 66 CPU --> 0.040 ||	teste 66 GPU --> 0.045 ||	teste 66 CPU/GPU --> 0.042
teste 67 CPU --> 0.039 ||	teste 67 GPU --> 0.044 ||	teste 67 CPU/GPU --> 0.042
teste 68 CPU --> 0.038 ||	teste 68 GPU --> 0.043 ||	teste 68 CPU/GPU --> 0.042
teste 69 CPU --> 0.036 ||	teste 69 GPU --> 0.044 ||	teste 69 CPU/GPU --> 0.059
teste 70 CPU --> 0.038 ||	teste 70 GPU --> 0.050 ||	teste 70 CPU/GPU --> 0.047
teste 71 CPU --> 0.036 ||	teste 71 GPU --> 0.044 ||	teste 71 CPU/GPU --> 0.044
teste 72 CPU --> 0.036 ||	teste 72 GPU --> 0.046 ||	teste 72 CPU/GPU --> 0.044
teste 73 CPU --> 0.044 ||	teste 73 GPU --> 0.043 ||	teste 73 CPU/GPU --> 0.042
teste 74 CPU --> 0.041 ||	teste 74 GPU --> 0.044 ||	teste 74 CPU/GPU --> 0.042
teste 75 CPU --> 0.038 ||	teste 75 GPU --> 0.045 ||	teste 75 CPU/GPU --> 0.042
teste 76 CPU --> 0.033 ||	teste 76 GPU --> 0.049 ||	teste 76 CPU/GPU --> 0.042
teste 77 CPU --> 0.033 ||	teste 77 GPU --> 0.044 ||	teste 77 CPU/GPU --> 0.042
teste 78 CPU --> 0.033 ||	teste 78 GPU --> 0.046 ||	teste 78 CPU/GPU --> 0.042
teste 79 CPU --> 0.033 ||	teste 79 GPU --> 0.047 ||	teste 79 CPU/GPU --> 0.043
teste 80 CPU --> 0.032 ||	teste 80 GPU --> 0.045 ||	teste 80 CPU/GPU --> 0.042
teste 81 CPU --> 0.033 ||	teste 81 GPU --> 0.041 ||	teste 81 CPU/GPU --> 0.042
teste 82 CPU --> 0.032 ||	teste 82 GPU --> 0.041 ||	teste 82 CPU/GPU --> 0.042
teste 83 CPU --> 0.032 ||	teste 83 GPU --> 0.041 ||	teste 83 CPU/GPU --> 0.042
teste 84 CPU --> 0.033 ||	teste 84 GPU --> 0.041 ||	teste 84 CPU/GPU --> 0.042
teste 85 CPU --> 0.032 ||	teste 85 GPU --> 0.041 ||	teste 85 CPU/GPU --> 0.042
teste 86 CPU --> 0.033 ||	teste 86 GPU --> 0.040 ||	teste 86 CPU/GPU --> 0.042
teste 87 CPU --> 0.032 ||	teste 87 GPU --> 0.041 ||	teste 87 CPU/GPU --> 0.043
teste 88 CPU --> 0.035 ||	teste 88 GPU --> 0.042 ||	teste 88 CPU/GPU --> 0.042
teste 89 CPU --> 0.032 ||	teste 89 GPU --> 0.041 ||	teste 89 CPU/GPU --> 0.042
teste 90 CPU --> 0.033 ||	teste 90 GPU --> 0.041 ||	teste 90 CPU/GPU --> 0.042
teste 91 CPU --> 0.034 ||	teste 91 GPU --> 0.051 ||	teste 91 CPU/GPU --> 0.045
teste 92 CPU --> 0.034 ||	teste 92 GPU --> 0.045 ||	teste 92 CPU/GPU --> 0.049
teste 93 CPU --> 0.033 ||	teste 93 GPU --> 0.047 ||	teste 93 CPU/GPU --> 0.045
teste 94 CPU --> 0.033 ||	teste 94 GPU --> 0.045 ||	teste 94 CPU/GPU --> 0.043
teste 95 CPU --> 0.034 ||	teste 95 GPU --> 0.046 ||	teste 95 CPU/GPU --> 0.042
teste 96 CPU --> 0.033 ||	teste 96 GPU --> 0.048 ||	teste 96 CPU/GPU --> 0.042
teste 97 CPU --> 0.032 ||	teste 97 GPU --> 0.049 ||	teste 97 CPU/GPU --> 0.042
teste 98 CPU --> 0.035 ||	teste 98 GPU --> 0.045 ||	teste 98 CPU/GPU --> 0.042
teste 99 CPU --> 0.033 ||	teste 99 GPU --> 0.046 ||	teste 99 CPU/GPU --> 0.042
teste 100 CPU --> 0.033 ||	teste 100 GPU --> 0.051 ||	teste 100 CPU/GPU --> 0.043

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.037 (+ ou -) 0.006	0.046 (+ ou -) 0.005	0.043 (+ ou -) 0.007

Apêndice C - Resultados Vetor 20.000

CPU = 20000	GPU = 20000	CPU/GPU = 20000(10000/10000)
==========	==========	==========
teste 1 CPU --> 0.069 ||	teste 1 GPU --> 0.084 ||	teste 1 CPU/GPU --> 0.093
teste 2 CPU --> 0.066 ||	teste 2 GPU --> 0.045 ||	teste 2 CPU/GPU --> 0.057
teste 3 CPU --> 0.069 ||	teste 3 GPU --> 0.043 ||	teste 3 CPU/GPU --> 0.049
teste 4 CPU --> 0.066 ||	teste 4 GPU --> 0.042 ||	teste 4 CPU/GPU --> 0.045
teste 5 CPU --> 0.067 ||	teste 5 GPU --> 0.041 ||	teste 5 CPU/GPU --> 0.044
teste 6 CPU --> 0.069 ||	teste 6 GPU --> 0.042 ||	teste 6 CPU/GPU --> 0.058
teste 7 CPU --> 0.068 ||	teste 7 GPU --> 0.041 ||	teste 7 CPU/GPU --> 0.043
teste 8 CPU --> 0.068 ||	teste 8 GPU --> 0.041 ||	teste 8 CPU/GPU --> 0.050
teste 9 CPU --> 0.068 ||	teste 9 GPU --> 0.041 ||	teste 9 CPU/GPU --> 0.046
teste 10 CPU --> 0.066 ||	teste 10 GPU --> 0.041 ||	teste 10 CPU/GPU --> 0.047
teste 11 CPU --> 0.066 ||	teste 11 GPU --> 0.041 ||	teste 11 CPU/GPU --> 0.046
teste 12 CPU --> 0.067 ||	teste 12 GPU --> 0.041 ||	teste 12 CPU/GPU --> 0.046
teste 13 CPU --> 0.067 ||	teste 13 GPU --> 0.041 ||	teste 13 CPU/GPU --> 0.046
teste 14 CPU --> 0.067 ||	teste 14 GPU --> 0.041 ||	teste 14 CPU/GPU --> 0.044
teste 15 CPU --> 0.066 ||	teste 15 GPU --> 0.041 ||	teste 15 CPU/GPU --> 0.047
teste 16 CPU --> 0.065 ||	teste 16 GPU --> 0.043 ||	teste 16 CPU/GPU --> 0.051
teste 17 CPU --> 0.067 ||	teste 17 GPU --> 0.041 ||	teste 17 CPU/GPU --> 0.051
teste 18 CPU --> 0.068 ||	teste 18 GPU --> 0.041 ||	teste 18 CPU/GPU --> 0.048
teste 19 CPU --> 0.067 ||	teste 19 GPU --> 0.041 ||	teste 19 CPU/GPU --> 0.044
teste 20 CPU --> 0.069 ||	teste 20 GPU --> 0.041 ||	teste 20 CPU/GPU --> 0.053
teste 21 CPU --> 0.068 ||	teste 21 GPU --> 0.041 ||	teste 21 CPU/GPU --> 0.046
teste 22 CPU --> 0.068 ||	teste 22 GPU --> 0.042 ||	teste 22 CPU/GPU --> 0.047
teste 23 CPU --> 0.060 ||	teste 23 GPU --> 0.041 ||	teste 23 CPU/GPU --> 0.046
teste 24 CPU --> 0.068 ||	teste 24 GPU --> 0.041 ||	teste 24 CPU/GPU --> 0.044
teste 25 CPU --> 0.066 ||	teste 25 GPU --> 0.041 ||	teste 25 CPU/GPU --> 0.051
teste 26 CPU --> 0.068 ||	teste 26 GPU --> 0.041 ||	teste 26 CPU/GPU --> 0.049
teste 27 CPU --> 0.068 ||	teste 27 GPU --> 0.041 ||	teste 27 CPU/GPU --> 0.047
teste 28 CPU --> 0.066 ||	teste 28 GPU --> 0.041 ||	teste 28 CPU/GPU --> 0.046
teste 29 CPU --> 0.066 ||	teste 29 GPU --> 0.041 ||	teste 29 CPU/GPU --> 0.047
teste 30 CPU --> 0.068 ||	teste 30 GPU --> 0.041 ||	teste 30 CPU/GPU --> 0.046
teste 31 CPU --> 0.069 ||	teste 31 GPU --> 0.041 ||	teste 31 CPU/GPU --> 0.046
teste 32 CPU --> 0.067 ||	teste 32 GPU --> 0.043 ||	teste 32 CPU/GPU --> 0.045
teste 33 CPU --> 0.065 ||	teste 33 GPU --> 0.041 ||	teste 33 CPU/GPU --> 0.044
teste 34 CPU --> 0.069 ||	teste 34 GPU --> 0.041 ||	teste 34 CPU/GPU --> 0.047
teste 35 CPU --> 0.068 ||	teste 35 GPU --> 0.042 ||	teste 35 CPU/GPU --> 0.046
teste 36 CPU --> 0.068 ||	teste 36 GPU --> 0.041 ||	teste 36 CPU/GPU --> 0.044
teste 37 CPU --> 0.075 ||	teste 37 GPU --> 0.041 ||	teste 37 CPU/GPU --> 0.047
teste 38 CPU --> 0.068 ||	teste 38 GPU --> 0.041 ||	teste 38 CPU/GPU --> 0.046
teste 39 CPU --> 0.067 ||	teste 39 GPU --> 0.041 ||	teste 39 CPU/GPU --> 0.046
teste 40 CPU --> 0.067 ||	teste 40 GPU --> 0.041 ||	teste 40 CPU/GPU --> 0.047
teste 41 CPU --> 0.068 ||	teste 41 GPU --> 0.041 ||	teste 41 CPU/GPU --> 0.046
teste 42 CPU --> 0.068 ||	teste 42 GPU --> 0.041 ||	teste 42 CPU/GPU --> 0.046
teste 43 CPU --> 0.069 ||	teste 43 GPU --> 0.041 ||	teste 43 CPU/GPU --> 0.044
teste 44 CPU --> 0.068 ||	teste 44 GPU --> 0.041 ||	teste 44 CPU/GPU --> 0.044
teste 45 CPU --> 0.068 ||	teste 45 GPU --> 0.041 ||	teste 45 CPU/GPU --> 0.047
teste 46 CPU --> 0.058 ||	teste 46 GPU --> 0.041 ||	teste 46 CPU/GPU --> 0.046
teste 47 CPU --> 0.067 ||	teste 47 GPU --> 0.041 ||	teste 47 CPU/GPU --> 0.044
teste 48 CPU --> 0.068 ||	teste 48 GPU --> 0.043 ||	teste 48 CPU/GPU --> 0.051
teste 49 CPU --> 0.069 ||	teste 49 GPU --> 0.041 ||	teste 49 CPU/GPU --> 0.049
teste 50 CPU --> 0.067 ||	teste 50 GPU --> 0.041 ||	teste 50 CPU/GPU --> 0.046
teste 51 CPU --> 0.067 ||	teste 51 GPU --> 0.041 ||	teste 51 CPU/GPU --> 0.046
teste 52 CPU --> 0.066 ||	teste 52 GPU --> 0.045 ||	teste 52 CPU/GPU --> 0.047
teste 53 CPU --> 0.067 ||	teste 53 GPU --> 0.043 ||	teste 53 CPU/GPU --> 0.044
teste 54 CPU --> 0.070 ||	teste 54 GPU --> 0.042 ||	teste 54 CPU/GPU --> 0.047
teste 55 CPU --> 0.068 ||	teste 55 GPU --> 0.041 ||	teste 55 CPU/GPU --> 0.046

teste 56 CPU --> 0.067 ||	teste 56 GPU --> 0.041 ||	teste 56 CPU/GPU --> 0.044
teste 57 CPU --> 0.067 ||	teste 57 GPU --> 0.041 ||	teste 57 CPU/GPU --> 0.047
teste 58 CPU --> 0.066 ||	teste 58 GPU --> 0.041 ||	teste 58 CPU/GPU --> 0.046
teste 59 CPU --> 0.066 ||	teste 59 GPU --> 0.041 ||	teste 59 CPU/GPU --> 0.046
teste 60 CPU --> 0.066 ||	teste 60 GPU --> 0.041 ||	teste 60 CPU/GPU --> 0.046
teste 61 CPU --> 0.060 ||	teste 61 GPU --> 0.041 ||	teste 61 CPU/GPU --> 0.044
teste 62 CPU --> 0.068 ||	teste 62 GPU --> 0.041 ||	teste 62 CPU/GPU --> 0.046
teste 63 CPU --> 0.067 ||	teste 63 GPU --> 0.041 ||	teste 63 CPU/GPU --> 0.046
teste 64 CPU --> 0.067 ||	teste 64 GPU --> 0.043 ||	teste 64 CPU/GPU --> 0.046
teste 65 CPU --> 0.066 ||	teste 65 GPU --> 0.041 ||	teste 65 CPU/GPU --> 0.044
teste 66 CPU --> 0.067 ||	teste 66 GPU --> 0.041 ||	teste 66 CPU/GPU --> 0.046
teste 67 CPU --> 0.069 ||	teste 67 GPU --> 0.041 ||	teste 67 CPU/GPU --> 0.046
teste 68 CPU --> 0.067 ||	teste 68 GPU --> 0.041 ||	teste 68 CPU/GPU --> 0.046
teste 69 CPU --> 0.067 ||	teste 69 GPU --> 0.040 ||	teste 69 CPU/GPU --> 0.046
teste 70 CPU --> 0.067 ||	teste 70 GPU --> 0.041 ||	teste 70 CPU/GPU --> 0.046
teste 71 CPU --> 0.066 ||	teste 71 GPU --> 0.041 ||	teste 71 CPU/GPU --> 0.046
teste 72 CPU --> 0.067 ||	teste 72 GPU --> 0.041 ||	teste 72 CPU/GPU --> 0.046
teste 73 CPU --> 0.067 ||	teste 73 GPU --> 0.040 ||	teste 73 CPU/GPU --> 0.046
teste 74 CPU --> 0.067 ||	teste 74 GPU --> 0.073 ||	teste 74 CPU/GPU --> 0.044
teste 75 CPU --> 0.067 ||	teste 75 GPU --> 0.073 ||	teste 75 CPU/GPU --> 0.044
teste 76 CPU --> 0.066 ||	teste 76 GPU --> 0.073 ||	teste 76 CPU/GPU --> 0.046
teste 77 CPU --> 0.067 ||	teste 77 GPU --> 0.071 ||	teste 77 CPU/GPU --> 0.046
teste 78 CPU --> 0.068 ||	teste 78 GPU --> 0.072 ||	teste 78 CPU/GPU --> 0.044
teste 79 CPU --> 0.069 ||	teste 79 GPU --> 0.072 ||	teste 79 CPU/GPU --> 0.044
teste 80 CPU --> 0.068 ||	teste 80 GPU --> 0.072 ||	teste 80 CPU/GPU --> 0.048
teste 81 CPU --> 0.069 ||	teste 81 GPU --> 0.073 ||	teste 81 CPU/GPU --> 0.047
teste 82 CPU --> 0.068 ||	teste 82 GPU --> 0.072 ||	teste 82 CPU/GPU --> 0.044
teste 83 CPU --> 0.067 ||	teste 83 GPU --> 0.061 ||	teste 83 CPU/GPU --> 0.046
teste 84 CPU --> 0.067 ||	teste 84 GPU --> 0.061 ||	teste 84 CPU/GPU --> 0.046
teste 85 CPU --> 0.066 ||	teste 85 GPU --> 0.063 ||	teste 85 CPU/GPU --> 0.044
teste 86 CPU --> 0.066 ||	teste 86 GPU --> 0.043 ||	teste 86 CPU/GPU --> 0.047
teste 87 CPU --> 0.065 ||	teste 87 GPU --> 0.042 ||	teste 87 CPU/GPU --> 0.046
teste 88 CPU --> 0.066 ||	teste 88 GPU --> 0.042 ||	teste 88 CPU/GPU --> 0.044
teste 89 CPU --> 0.066 ||	teste 89 GPU --> 0.040 ||	teste 89 CPU/GPU --> 0.046
teste 90 CPU --> 0.068 ||	teste 90 GPU --> 0.041 ||	teste 90 CPU/GPU --> 0.046
teste 91 CPU --> 0.068 ||	teste 91 GPU --> 0.041 ||	teste 91 CPU/GPU --> 0.046
teste 92 CPU --> 0.067 ||	teste 92 GPU --> 0.041 ||	teste 92 CPU/GPU --> 0.046
teste 93 CPU --> 0.067 ||	teste 93 GPU --> 0.041 ||	teste 93 CPU/GPU --> 0.044
teste 94 CPU --> 0.067 ||	teste 94 GPU --> 0.041 ||	teste 94 CPU/GPU --> 0.046
teste 95 CPU --> 0.068 ||	teste 95 GPU --> 0.041 ||	teste 95 CPU/GPU --> 0.046
teste 96 CPU --> 0.069 ||	teste 96 GPU --> 0.043 ||	teste 96 CPU/GPU --> 0.047
teste 97 CPU --> 0.067 ||	teste 97 GPU --> 0.041 ||	teste 97 CPU/GPU --> 0.047
teste 98 CPU --> 0.067 ||	teste 98 GPU --> 0.041 ||	teste 98 CPU/GPU --> 0.046
teste 99 CPU --> 0.067 ||	teste 99 GPU --> 0.041 ||	teste 99 CPU/GPU --> 0.046
teste 100 CPU --> 0.066 ||	teste 100 GPU --> 0.041 ||	teste 100 CPU/GPU --> 0.046

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.067 (+ ou -) 0.002	0.045 (+ ou -) 0.010	0.047 (+ ou -) 0.005

Apêndice D - Resultados Vetor 22.000

CPU = 22000	GPU = 22000	CPU/GPU = 22000(11000/11000)
==========	==========	==========
teste 1 CPU --> 0.073 ||	teste 1 GPU --> 0.075 ||	teste 1 CPU/GPU --> 0.088
teste 2 CPU --> 0.074 ||	teste 2 GPU --> 0.065 ||	teste 2 CPU/GPU --> 0.048
teste 3 CPU --> 0.074 ||	teste 3 GPU --> 0.045 ||	teste 3 CPU/GPU --> 0.044
teste 4 CPU --> 0.074 ||	teste 4 GPU --> 0.044 ||	teste 4 CPU/GPU --> 0.045
teste 5 CPU --> 0.074 ||	teste 5 GPU --> 0.043 ||	teste 5 CPU/GPU --> 0.044
teste 6 CPU --> 0.075 ||	teste 6 GPU --> 0.043 ||	teste 6 CPU/GPU --> 0.044
teste 7 CPU --> 0.072 ||	teste 7 GPU --> 0.043 ||	teste 7 CPU/GPU --> 0.044
teste 8 CPU --> 0.075 ||	teste 8 GPU --> 0.043 ||	teste 8 CPU/GPU --> 0.044
teste 9 CPU --> 0.065 ||	teste 9 GPU --> 0.044 ||	teste 9 CPU/GPU --> 0.044
teste 10 CPU --> 0.062 ||	teste 10 GPU --> 0.042 ||	teste 10 CPU/GPU --> 0.044
teste 11 CPU --> 0.067 ||	teste 11 GPU --> 0.047 ||	teste 11 CPU/GPU --> 0.044
teste 12 CPU --> 0.093 ||	teste 12 GPU --> 0.042 ||	teste 12 CPU/GPU --> 0.044
teste 13 CPU --> 0.066 ||	teste 13 GPU --> 0.044 ||	teste 13 CPU/GPU --> 0.044
teste 14 CPU --> 0.063 ||	teste 14 GPU --> 0.043 ||	teste 14 CPU/GPU --> 0.044
teste 15 CPU --> 0.065 ||	teste 15 GPU --> 0.044 ||	teste 15 CPU/GPU --> 0.044
teste 16 CPU --> 0.064 ||	teste 16 GPU --> 0.043 ||	teste 16 CPU/GPU --> 0.044
teste 17 CPU --> 0.064 ||	teste 17 GPU --> 0.043 ||	teste 17 CPU/GPU --> 0.043
teste 18 CPU --> 0.064 ||	teste 18 GPU --> 0.043 ||	teste 18 CPU/GPU --> 0.044
teste 19 CPU --> 0.063 ||	teste 19 GPU --> 0.043 ||	teste 19 CPU/GPU --> 0.044
teste 20 CPU --> 0.062 ||	teste 20 GPU --> 0.042 ||	teste 20 CPU/GPU --> 0.044
teste 21 CPU --> 0.062 ||	teste 21 GPU --> 0.042 ||	teste 21 CPU/GPU --> 0.043
teste 22 CPU --> 0.075 ||	teste 22 GPU --> 0.043 ||	teste 22 CPU/GPU --> 0.044
teste 23 CPU --> 0.065 ||	teste 23 GPU --> 0.043 ||	teste 23 CPU/GPU --> 0.044
teste 24 CPU --> 0.065 ||	teste 24 GPU --> 0.042 ||	teste 24 CPU/GPU --> 0.044
teste 25 CPU --> 0.062 ||	teste 25 GPU --> 0.043 ||	teste 25 CPU/GPU --> 0.044
teste 26 CPU --> 0.062 ||	teste 26 GPU --> 0.043 ||	teste 26 CPU/GPU --> 0.044
teste 27 CPU --> 0.063 ||	teste 27 GPU --> 0.046 ||	teste 27 CPU/GPU --> 0.043
teste 28 CPU --> 0.062 ||	teste 28 GPU --> 0.043 ||	teste 28 CPU/GPU --> 0.043
teste 29 CPU --> 0.062 ||	teste 29 GPU --> 0.043 ||	teste 29 CPU/GPU --> 0.043
teste 30 CPU --> 0.062 ||	teste 30 GPU --> 0.043 ||	teste 30 CPU/GPU --> 0.044
teste 31 CPU --> 0.062 ||	teste 31 GPU --> 0.044 ||	teste 31 CPU/GPU --> 0.043
teste 32 CPU --> 0.063 ||	teste 32 GPU --> 0.042 ||	teste 32 CPU/GPU --> 0.058
teste 33 CPU --> 0.063 ||	teste 33 GPU --> 0.043 ||	teste 33 CPU/GPU --> 0.046
teste 34 CPU --> 0.062 ||	teste 34 GPU --> 0.043 ||	teste 34 CPU/GPU --> 0.049
teste 35 CPU --> 0.062 ||	teste 35 GPU --> 0.042 ||	teste 35 CPU/GPU --> 0.049
teste 36 CPU --> 0.063 ||	teste 36 GPU --> 0.043 ||	teste 36 CPU/GPU --> 0.047
teste 37 CPU --> 0.062 ||	teste 37 GPU --> 0.043 ||	teste 37 CPU/GPU --> 0.044
teste 38 CPU --> 0.063 ||	teste 38 GPU --> 0.043 ||	teste 38 CPU/GPU --> 0.044
teste 39 CPU --> 0.062 ||	teste 39 GPU --> 0.043 ||	teste 39 CPU/GPU --> 0.044
teste 40 CPU --> 0.063 ||	teste 40 GPU --> 0.043 ||	teste 40 CPU/GPU --> 0.043
teste 41 CPU --> 0.063 ||	teste 41 GPU --> 0.043 ||	teste 41 CPU/GPU --> 0.044
teste 42 CPU --> 0.063 ||	teste 42 GPU --> 0.042 ||	teste 42 CPU/GPU --> 0.044
teste 43 CPU --> 0.078 ||	teste 43 GPU --> 0.044 ||	teste 43 CPU/GPU --> 0.044
teste 44 CPU --> 0.069 ||	teste 44 GPU --> 0.043 ||	teste 44 CPU/GPU --> 0.044
teste 45 CPU --> 0.064 ||	teste 45 GPU --> 0.043 ||	teste 45 CPU/GPU --> 0.044
teste 46 CPU --> 0.066 ||	teste 46 GPU --> 0.042 ||	teste 46 CPU/GPU --> 0.045
teste 47 CPU --> 0.064 ||	teste 47 GPU --> 0.044 ||	teste 47 CPU/GPU --> 0.044
teste 48 CPU --> 0.062 ||	teste 48 GPU --> 0.042 ||	teste 48 CPU/GPU --> 0.044
teste 49 CPU --> 0.063 ||	teste 49 GPU --> 0.043 ||	teste 49 CPU/GPU --> 0.043
teste 50 CPU --> 0.063 ||	teste 50 GPU --> 0.043 ||	teste 50 CPU/GPU --> 0.044
teste 51 CPU --> 0.063 ||	teste 51 GPU --> 0.042 ||	teste 51 CPU/GPU --> 0.044
teste 52 CPU --> 0.063 ||	teste 52 GPU --> 0.043 ||	teste 52 CPU/GPU --> 0.044
teste 53 CPU --> 0.063 ||	teste 53 GPU --> 0.043 ||	teste 53 CPU/GPU --> 0.044
teste 54 CPU --> 0.062 ||	teste 54 GPU --> 0.043 ||	teste 54 CPU/GPU --> 0.044
teste 55 CPU --> 0.063 ||	teste 55 GPU --> 0.043 ||	teste 55 CPU/GPU --> 0.044

teste 56 CPU --> 0.062 ||	teste 56 GPU --> 0.042 ||	teste 56 CPU/GPU --> 0.044
teste 57 CPU --> 0.063 ||	teste 57 GPU --> 0.042 ||	teste 57 CPU/GPU --> 0.043
teste 58 CPU --> 0.063 ||	teste 58 GPU --> 0.042 ||	teste 58 CPU/GPU --> 0.044
teste 59 CPU --> 0.062 ||	teste 59 GPU --> 0.044 ||	teste 59 CPU/GPU --> 0.043
teste 60 CPU --> 0.062 ||	teste 60 GPU --> 0.042 ||	teste 60 CPU/GPU --> 0.044
teste 61 CPU --> 0.061 ||	teste 61 GPU --> 0.044 ||	teste 61 CPU/GPU --> 0.044
teste 62 CPU --> 0.063 ||	teste 62 GPU --> 0.043 ||	teste 62 CPU/GPU --> 0.045
teste 63 CPU --> 0.063 ||	teste 63 GPU --> 0.044 ||	teste 63 CPU/GPU --> 0.044
teste 64 CPU --> 0.062 ||	teste 64 GPU --> 0.042 ||	teste 64 CPU/GPU --> 0.044
teste 65 CPU --> 0.064 ||	teste 65 GPU --> 0.043 ||	teste 65 CPU/GPU --> 0.044
teste 66 CPU --> 0.065 ||	teste 66 GPU --> 0.043 ||	teste 66 CPU/GPU --> 0.044
teste 67 CPU --> 0.064 ||	teste 67 GPU --> 0.043 ||	teste 67 CPU/GPU --> 0.044
teste 68 CPU --> 0.063 ||	teste 68 GPU --> 0.054 ||	teste 68 CPU/GPU --> 0.044
teste 69 CPU --> 0.061 ||	teste 69 GPU --> 0.058 ||	teste 69 CPU/GPU --> 0.044
teste 70 CPU --> 0.063 ||	teste 70 GPU --> 0.132 ||	teste 70 CPU/GPU --> 0.044
teste 71 CPU --> 0.063 ||	teste 71 GPU --> 0.047 ||	teste 71 CPU/GPU --> 0.044
teste 72 CPU --> 0.081 ||	teste 72 GPU --> 0.048 ||	teste 72 CPU/GPU --> 0.044
teste 73 CPU --> 0.084 ||	teste 73 GPU --> 0.062 ||	teste 73 CPU/GPU --> 0.044
teste 74 CPU --> 0.081 ||	teste 74 GPU --> 0.045 ||	teste 74 CPU/GPU --> 0.043
teste 75 CPU --> 0.065 ||	teste 75 GPU --> 0.044 ||	teste 75 CPU/GPU --> 0.044
teste 76 CPU --> 0.064 ||	teste 76 GPU --> 0.043 ||	teste 76 CPU/GPU --> 0.044
teste 77 CPU --> 0.065 ||	teste 77 GPU --> 0.044 ||	teste 77 CPU/GPU --> 0.043
teste 78 CPU --> 0.064 ||	teste 78 GPU --> 0.043 ||	teste 78 CPU/GPU --> 0.045
teste 79 CPU --> 0.062 ||	teste 79 GPU --> 0.045 ||	teste 79 CPU/GPU --> 0.044
teste 80 CPU --> 0.064 ||	teste 80 GPU --> 0.043 ||	teste 80 CPU/GPU --> 0.044
teste 81 CPU --> 0.062 ||	teste 81 GPU --> 0.043 ||	teste 81 CPU/GPU --> 0.044
teste 82 CPU --> 0.062 ||	teste 82 GPU --> 0.043 ||	teste 82 CPU/GPU --> 0.044
teste 83 CPU --> 0.062 ||	teste 83 GPU --> 0.044 ||	teste 83 CPU/GPU --> 0.044
teste 84 CPU --> 0.066 ||	teste 84 GPU --> 0.043 ||	teste 84 CPU/GPU --> 0.044
teste 85 CPU --> 0.064 ||	teste 85 GPU --> 0.043 ||	teste 85 CPU/GPU --> 0.044
teste 86 CPU --> 0.062 ||	teste 86 GPU --> 0.044 ||	teste 86 CPU/GPU --> 0.044
teste 87 CPU --> 0.065 ||	teste 87 GPU --> 0.043 ||	teste 87 CPU/GPU --> 0.044
teste 88 CPU --> 0.064 ||	teste 88 GPU --> 0.042 ||	teste 88 CPU/GPU --> 0.043
teste 89 CPU --> 0.065 ||	teste 89 GPU --> 0.045 ||	teste 89 CPU/GPU --> 0.043
teste 90 CPU --> 0.063 ||	teste 90 GPU --> 0.043 ||	teste 90 CPU/GPU --> 0.044
teste 91 CPU --> 0.062 ||	teste 91 GPU --> 0.043 ||	teste 91 CPU/GPU --> 0.043
teste 92 CPU --> 0.063 ||	teste 92 GPU --> 0.043 ||	teste 92 CPU/GPU --> 0.044
teste 93 CPU --> 0.062 ||	teste 93 GPU --> 0.043 ||	teste 93 CPU/GPU --> 0.044
teste 94 CPU --> 0.063 ||	teste 94 GPU --> 0.043 ||	teste 94 CPU/GPU --> 0.044
teste 95 CPU --> 0.063 ||	teste 95 GPU --> 0.044 ||	teste 95 CPU/GPU --> 0.043
teste 96 CPU --> 0.063 ||	teste 96 GPU --> 0.042 ||	teste 96 CPU/GPU --> 0.044
teste 97 CPU --> 0.066 ||	teste 97 GPU --> 0.043 ||	teste 97 CPU/GPU --> 0.043
teste 98 CPU --> 0.064 ||	teste 98 GPU --> 0.042 ||	teste 98 CPU/GPU --> 0.044
teste 99 CPU --> 0.064 ||	teste 99 GPU --> 0.043 ||	teste 99 CPU/GPU --> 0.044
teste 100 CPU --> 0.064 ||	teste 100 GPU --> 0.043 ||	teste 100 CPU/GPU --> 0.044

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.065 (+ ou -) 0.006	0.045 (+ ou -) 0.010	0.044 (+ ou -) 0.005

Apêndice E - Resultados Vetor 24.000

CPU = 24000	GPU = 24000	CPU/GPU = 24000(12000/12000)
==========	==========	==========
teste 1 CPU --> 0.078 ||	teste 1 GPU --> 0.085 ||	teste 1 CPU/GPU --> 0.091
teste 2 CPU --> 0.082 ||	teste 2 GPU --> 0.048 ||	teste 2 CPU/GPU --> 0.058
teste 3 CPU --> 0.078 ||	teste 3 GPU --> 0.044 ||	teste 3 CPU/GPU --> 0.056
teste 4 CPU --> 0.081 ||	teste 4 GPU --> 0.044 ||	teste 4 CPU/GPU --> 0.052
teste 5 CPU --> 0.081 ||	teste 5 GPU --> 0.043 ||	teste 5 CPU/GPU --> 0.047
teste 6 CPU --> 0.080 ||	teste 6 GPU --> 0.043 ||	teste 6 CPU/GPU --> 0.049
teste 7 CPU --> 0.080 ||	teste 7 GPU --> 0.043 ||	teste 7 CPU/GPU --> 0.048
teste 8 CPU --> 0.080 ||	teste 8 GPU --> 0.043 ||	teste 8 CPU/GPU --> 0.047
teste 9 CPU --> 0.083 ||	teste 9 GPU --> 0.043 ||	teste 9 CPU/GPU --> 0.051
teste 10 CPU --> 0.078 ||	teste 10 GPU --> 0.045 ||	teste 10 CPU/GPU --> 0.049
teste 11 CPU --> 0.081 ||	teste 11 GPU --> 0.044 ||	teste 11 CPU/GPU --> 0.046
teste 12 CPU --> 0.078 ||	teste 12 GPU --> 0.044 ||	teste 12 CPU/GPU --> 0.046
teste 13 CPU --> 0.076 ||	teste 13 GPU --> 0.043 ||	teste 13 CPU/GPU --> 0.045
teste 14 CPU --> 0.067 ||	teste 14 GPU --> 0.043 ||	teste 14 CPU/GPU --> 0.046
teste 15 CPU --> 0.081 ||	teste 15 GPU --> 0.043 ||	teste 15 CPU/GPU --> 0.059
teste 16 CPU --> 0.083 ||	teste 16 GPU --> 0.044 ||	teste 16 CPU/GPU --> 0.045
teste 17 CPU --> 0.085 ||	teste 17 GPU --> 0.042 ||	teste 17 CPU/GPU --> 0.045
teste 18 CPU --> 0.083 ||	teste 18 GPU --> 0.043 ||	teste 18 CPU/GPU --> 0.047
teste 19 CPU --> 0.078 ||	teste 19 GPU --> 0.042 ||	teste 19 CPU/GPU --> 0.050
teste 20 CPU --> 0.079 ||	teste 20 GPU --> 0.044 ||	teste 20 CPU/GPU --> 0.046
teste 21 CPU --> 0.079 ||	teste 21 GPU --> 0.045 ||	teste 21 CPU/GPU --> 0.051
teste 22 CPU --> 0.079 ||	teste 22 GPU --> 0.044 ||	teste 22 CPU/GPU --> 0.049
teste 23 CPU --> 0.078 ||	teste 23 GPU --> 0.044 ||	teste 23 CPU/GPU --> 0.046
teste 24 CPU --> 0.079 ||	teste 24 GPU --> 0.043 ||	teste 24 CPU/GPU --> 0.052
teste 25 CPU --> 0.079 ||	teste 25 GPU --> 0.043 ||	teste 25 CPU/GPU --> 0.045
teste 26 CPU --> 0.082 ||	teste 26 GPU --> 0.044 ||	teste 26 CPU/GPU --> 0.045
teste 27 CPU --> 0.084 ||	teste 27 GPU --> 0.045 ||	teste 27 CPU/GPU --> 0.044
teste 28 CPU --> 0.081 ||	teste 28 GPU --> 0.044 ||	teste 28 CPU/GPU --> 0.045
teste 29 CPU --> 0.079 ||	teste 29 GPU --> 0.043 ||	teste 29 CPU/GPU --> 0.045
teste 30 CPU --> 0.080 ||	teste 30 GPU --> 0.043 ||	teste 30 CPU/GPU --> 0.045
teste 31 CPU --> 0.079 ||	teste 31 GPU --> 0.043 ||	teste 31 CPU/GPU --> 0.045
teste 32 CPU --> 0.080 ||	teste 32 GPU --> 0.044 ||	teste 32 CPU/GPU --> 0.046
teste 33 CPU --> 0.106 ||	teste 33 GPU --> 0.043 ||	teste 33 CPU/GPU --> 0.050
teste 34 CPU --> 0.081 ||	teste 34 GPU --> 0.043 ||	teste 34 CPU/GPU --> 0.045
teste 35 CPU --> 0.084 ||	teste 35 GPU --> 0.043 ||	teste 35 CPU/GPU --> 0.045
teste 36 CPU --> 0.067 ||	teste 36 GPU --> 0.043 ||	teste 36 CPU/GPU --> 0.045
teste 37 CPU --> 0.068 ||	teste 37 GPU --> 0.043 ||	teste 37 CPU/GPU --> 0.045
teste 38 CPU --> 0.079 ||	teste 38 GPU --> 0.043 ||	teste 38 CPU/GPU --> 0.045
teste 39 CPU --> 0.067 ||	teste 39 GPU --> 0.042 ||	teste 39 CPU/GPU --> 0.045
teste 40 CPU --> 0.067 ||	teste 40 GPU --> 0.048 ||	teste 40 CPU/GPU --> 0.045
teste 41 CPU --> 0.068 ||	teste 41 GPU --> 0.044 ||	teste 41 CPU/GPU --> 0.045
teste 42 CPU --> 0.067 ||	teste 42 GPU --> 0.043 ||	teste 42 CPU/GPU --> 0.045
teste 43 CPU --> 0.069 ||	teste 43 GPU --> 0.046 ||	teste 43 CPU/GPU --> 0.045
teste 44 CPU --> 0.068 ||	teste 44 GPU --> 0.044 ||	teste 44 CPU/GPU --> 0.045
teste 45 CPU --> 0.070 ||	teste 45 GPU --> 0.043 ||	teste 45 CPU/GPU --> 0.045
teste 46 CPU --> 0.067 ||	teste 46 GPU --> 0.043 ||	teste 46 CPU/GPU --> 0.045
teste 47 CPU --> 0.068 ||	teste 47 GPU --> 0.043 ||	teste 47 CPU/GPU --> 0.044
teste 48 CPU --> 0.067 ||	teste 48 GPU --> 0.043 ||	teste 48 CPU/GPU --> 0.046
teste 49 CPU --> 0.067 ||	teste 49 GPU --> 0.043 ||	teste 49 CPU/GPU --> 0.053
teste 50 CPU --> 0.068 ||	teste 50 GPU --> 0.043 ||	teste 50 CPU/GPU --> 0.050
teste 51 CPU --> 0.069 ||	teste 51 GPU --> 0.043 ||	teste 51 CPU/GPU --> 0.048
teste 52 CPU --> 0.068 ||	teste 52 GPU --> 0.043 ||	teste 52 CPU/GPU --> 0.047
teste 53 CPU --> 0.088 ||	teste 53 GPU --> 0.043 ||	teste 53 CPU/GPU --> 0.050
teste 54 CPU --> 0.079 ||	teste 54 GPU --> 0.043 ||	teste 54 CPU/GPU --> 0.046
teste 55 CPU --> 0.078 ||	teste 55 GPU --> 0.042 ||	teste 55 CPU/GPU --> 0.045

teste 56 CPU --> 0.069 ||	teste 56 GPU --> 0.043 ||	teste 56 CPU/GPU --> 0.045
teste 57 CPU --> 0.068 ||	teste 57 GPU --> 0.042 ||	teste 57 CPU/GPU --> 0.045
teste 58 CPU --> 0.068 ||	teste 58 GPU --> 0.043 ||	teste 58 CPU/GPU --> 0.045
teste 59 CPU --> 0.081 ||	teste 59 GPU --> 0.045 ||	teste 59 CPU/GPU --> 0.045
teste 60 CPU --> 0.069 ||	teste 60 GPU --> 0.048 ||	teste 60 CPU/GPU --> 0.045
teste 61 CPU --> 0.068 ||	teste 61 GPU --> 0.043 ||	teste 61 CPU/GPU --> 0.045
teste 62 CPU --> 0.068 ||	teste 62 GPU --> 0.043 ||	teste 62 CPU/GPU --> 0.045
teste 63 CPU --> 0.067 ||	teste 63 GPU --> 0.042 ||	teste 63 CPU/GPU --> 0.044
teste 64 CPU --> 0.067 ||	teste 64 GPU --> 0.044 ||	teste 64 CPU/GPU --> 0.045
teste 65 CPU --> 0.081 ||	teste 65 GPU --> 0.043 ||	teste 65 CPU/GPU --> 0.044
teste 66 CPU --> 0.068 ||	teste 66 GPU --> 0.043 ||	teste 66 CPU/GPU --> 0.054
teste 67 CPU --> 0.068 ||	teste 67 GPU --> 0.043 ||	teste 67 CPU/GPU --> 0.044
teste 68 CPU --> 0.067 ||	teste 68 GPU --> 0.043 ||	teste 68 CPU/GPU --> 0.045
teste 69 CPU --> 0.067 ||	teste 69 GPU --> 0.043 ||	teste 69 CPU/GPU --> 0.045
teste 70 CPU --> 0.067 ||	teste 70 GPU --> 0.044 ||	teste 70 CPU/GPU --> 0.045
teste 71 CPU --> 0.069 ||	teste 71 GPU --> 0.043 ||	teste 71 CPU/GPU --> 0.045
teste 72 CPU --> 0.068 ||	teste 72 GPU --> 0.043 ||	teste 72 CPU/GPU --> 0.045
teste 73 CPU --> 0.104 ||	teste 73 GPU --> 0.042 ||	teste 73 CPU/GPU --> 0.045
teste 74 CPU --> 0.067 ||	teste 74 GPU --> 0.043 ||	teste 74 CPU/GPU --> 0.045
teste 75 CPU --> 0.068 ||	teste 75 GPU --> 0.045 ||	teste 75 CPU/GPU --> 0.045
teste 76 CPU --> 0.067 ||	teste 76 GPU --> 0.044 ||	teste 76 CPU/GPU --> 0.045
teste 77 CPU --> 0.103 ||	teste 77 GPU --> 0.042 ||	teste 77 CPU/GPU --> 0.045
teste 78 CPU --> 0.069 ||	teste 78 GPU --> 0.043 ||	teste 78 CPU/GPU --> 0.045
teste 79 CPU --> 0.068 ||	teste 79 GPU --> 0.043 ||	teste 79 CPU/GPU --> 0.044
teste 80 CPU --> 0.067 ||	teste 80 GPU --> 0.059 ||	teste 80 CPU/GPU --> 0.045
teste 81 CPU --> 0.067 ||	teste 81 GPU --> 0.058 ||	teste 81 CPU/GPU --> 0.045
teste 82 CPU --> 0.067 ||	teste 82 GPU --> 0.052 ||	teste 82 CPU/GPU --> 0.045
teste 83 CPU --> 0.066 ||	teste 83 GPU --> 0.051 ||	teste 83 CPU/GPU --> 0.045
teste 84 CPU --> 0.068 ||	teste 84 GPU --> 0.051 ||	teste 84 CPU/GPU --> 0.070
teste 85 CPU --> 0.071 ||	teste 85 GPU --> 0.051 ||	teste 85 CPU/GPU --> 0.049
teste 86 CPU --> 0.068 ||	teste 86 GPU --> 0.051 ||	teste 86 CPU/GPU --> 0.051
teste 87 CPU --> 0.068 ||	teste 87 GPU --> 0.050 ||	teste 87 CPU/GPU --> 0.048
teste 88 CPU --> 0.069 ||	teste 88 GPU --> 0.051 ||	teste 88 CPU/GPU --> 0.045
teste 89 CPU --> 0.067 ||	teste 89 GPU --> 0.050 ||	teste 89 CPU/GPU --> 0.045
teste 90 CPU --> 0.103 ||	teste 90 GPU --> 0.066 ||	teste 90 CPU/GPU --> 0.045
teste 91 CPU --> 0.067 ||	teste 91 GPU --> 0.044 ||	teste 91 CPU/GPU --> 0.045
teste 92 CPU --> 0.072 ||	teste 92 GPU --> 0.043 ||	teste 92 CPU/GPU --> 0.045
teste 93 CPU --> 0.095 ||	teste 93 GPU --> 0.042 ||	teste 93 CPU/GPU --> 0.045
teste 94 CPU --> 0.067 ||	teste 94 GPU --> 0.043 ||	teste 94 CPU/GPU --> 0.046
teste 95 CPU --> 0.069 ||	teste 95 GPU --> 0.042 ||	teste 95 CPU/GPU --> 0.045
teste 96 CPU --> 0.069 ||	teste 96 GPU --> 0.043 ||	teste 96 CPU/GPU --> 0.045
teste 97 CPU --> 0.067 ||	teste 97 GPU --> 0.042 ||	teste 97 CPU/GPU --> 0.045
teste 98 CPU --> 0.067 ||	teste 98 GPU --> 0.042 ||	teste 98 CPU/GPU --> 0.046
teste 99 CPU --> 0.067 ||	teste 99 GPU --> 0.042 ||	teste 99 CPU/GPU --> 0.050
teste 100 CPU --> 0.068 ||	teste 100 GPU --> 0.043 ||	teste 100 CPU/GPU --> 0.054

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.074 (+ ou -) 0.009	0.045 (+ ou -) 0.006	0.047 (+ ou -) 0.006

Apêndice F - Resultados Vetor 100.000

CPU = 100000	GPU = 100000	CPU/GPU = 100000(50000/50000)
==========	==========	==========
teste 1 CPU --> 0.224 ||	teste 1 GPU --> 0.046 ||	teste 1 CPU/GPU --> 0.190
teste 2 CPU --> 0.191 ||	teste 2 GPU --> 0.050 ||	teste 2 CPU/GPU --> 0.176
teste 3 CPU --> 0.202 ||	teste 3 GPU --> 0.050 ||	teste 3 CPU/GPU --> 0.171
teste 4 CPU --> 0.203 ||	teste 4 GPU --> 0.046 ||	teste 4 CPU/GPU --> 0.170
teste 5 CPU --> 0.218 ||	teste 5 GPU --> 0.046 ||	teste 5 CPU/GPU --> 0.170
teste 6 CPU --> 0.213 ||	teste 6 GPU --> 0.046 ||	teste 6 CPU/GPU --> 0.170
teste 7 CPU --> 0.310 ||	teste 7 GPU --> 0.046 ||	teste 7 CPU/GPU --> 0.169
teste 8 CPU --> 0.225 ||	teste 8 GPU --> 0.048 ||	teste 8 CPU/GPU --> 0.170
teste 9 CPU --> 0.221 ||	teste 9 GPU --> 0.046 ||	teste 9 CPU/GPU --> 0.171
teste 10 CPU --> 0.228 ||	teste 10 GPU --> 0.046 ||	teste 10 CPU/GPU --> 0.176
teste 11 CPU --> 0.294 ||	teste 11 GPU --> 0.046 ||	teste 11 CPU/GPU --> 0.151
teste 12 CPU --> 0.199 ||	teste 12 GPU --> 0.046 ||	teste 12 CPU/GPU --> 0.176
teste 13 CPU --> 0.199 ||	teste 13 GPU --> 0.046 ||	teste 13 CPU/GPU --> 0.172
teste 14 CPU --> 0.196 ||	teste 14 GPU --> 0.046 ||	teste 14 CPU/GPU --> 0.173
teste 15 CPU --> 0.194 ||	teste 15 GPU --> 0.046 ||	teste 15 CPU/GPU --> 0.171
teste 16 CPU --> 0.196 ||	teste 16 GPU --> 0.046 ||	teste 16 CPU/GPU --> 0.171
teste 17 CPU --> 0.195 ||	teste 17 GPU --> 0.046 ||	teste 17 CPU/GPU --> 0.173
teste 18 CPU --> 0.196 ||	teste 18 GPU --> 0.046 ||	teste 18 CPU/GPU --> 0.172
teste 19 CPU --> 0.196 ||	teste 19 GPU --> 0.046 ||	teste 19 CPU/GPU --> 0.171
teste 20 CPU --> 0.195 ||	teste 20 GPU --> 0.046 ||	teste 20 CPU/GPU --> 0.171
teste 21 CPU --> 0.197 ||	teste 21 GPU --> 0.046 ||	teste 21 CPU/GPU --> 0.173
teste 22 CPU --> 0.204 ||	teste 22 GPU --> 0.046 ||	teste 22 CPU/GPU --> 0.175
teste 23 CPU --> 0.195 ||	teste 23 GPU --> 0.046 ||	teste 23 CPU/GPU --> 0.171
teste 24 CPU --> 0.196 ||	teste 24 GPU --> 0.046 ||	teste 24 CPU/GPU --> 0.171
teste 25 CPU --> 0.197 ||	teste 25 GPU --> 0.046 ||	teste 25 CPU/GPU --> 0.171
teste 26 CPU --> 0.197 ||	teste 26 GPU --> 0.046 ||	teste 26 CPU/GPU --> 0.170
teste 27 CPU --> 0.195 ||	teste 27 GPU --> 0.046 ||	teste 27 CPU/GPU --> 0.145
teste 28 CPU --> 0.194 ||	teste 28 GPU --> 0.046 ||	teste 28 CPU/GPU --> 0.176
teste 29 CPU --> 0.196 ||	teste 29 GPU --> 0.046 ||	teste 29 CPU/GPU --> 0.172
teste 30 CPU --> 0.196 ||	teste 30 GPU --> 0.046 ||	teste 30 CPU/GPU --> 0.174
teste 31 CPU --> 0.201 ||	teste 31 GPU --> 0.046 ||	teste 31 CPU/GPU --> 0.172
teste 32 CPU --> 0.194 ||	teste 32 GPU --> 0.046 ||	teste 32 CPU/GPU --> 0.171
teste 33 CPU --> 0.197 ||	teste 33 GPU --> 0.046 ||	teste 33 CPU/GPU --> 0.169
teste 34 CPU --> 0.197 ||	teste 34 GPU --> 0.046 ||	teste 34 CPU/GPU --> 0.170
teste 35 CPU --> 0.197 ||	teste 35 GPU --> 0.046 ||	teste 35 CPU/GPU --> 0.172
teste 36 CPU --> 0.211 ||	teste 36 GPU --> 0.046 ||	teste 36 CPU/GPU --> 0.170
teste 37 CPU --> 0.205 ||	teste 37 GPU --> 0.046 ||	teste 37 CPU/GPU --> 0.170
teste 38 CPU --> 0.197 ||	teste 38 GPU --> 0.046 ||	teste 38 CPU/GPU --> 0.171
teste 39 CPU --> 0.200 ||	teste 39 GPU --> 0.047 ||	teste 39 CPU/GPU --> 0.170
teste 40 CPU --> 0.197 ||	teste 40 GPU --> 0.045 ||	teste 40 CPU/GPU --> 0.170
teste 41 CPU --> 0.202 ||	teste 41 GPU --> 0.045 ||	teste 41 CPU/GPU --> 0.169
teste 42 CPU --> 0.196 ||	teste 42 GPU --> 0.046 ||	teste 42 CPU/GPU --> 0.169
teste 43 CPU --> 0.201 ||	teste 43 GPU --> 0.046 ||	teste 43 CPU/GPU --> 0.171
teste 44 CPU --> 0.199 ||	teste 44 GPU --> 0.046 ||	teste 44 CPU/GPU --> 0.174
teste 45 CPU --> 0.201 ||	teste 45 GPU --> 0.046 ||	teste 45 CPU/GPU --> 0.173
teste 46 CPU --> 0.198 ||	teste 46 GPU --> 0.046 ||	teste 46 CPU/GPU --> 0.175
teste 47 CPU --> 0.201 ||	teste 47 GPU --> 0.046 ||	teste 47 CPU/GPU --> 0.169
teste 48 CPU --> 0.199 ||	teste 48 GPU --> 0.046 ||	teste 48 CPU/GPU --> 0.168
teste 49 CPU --> 0.199 ||	teste 49 GPU --> 0.046 ||	teste 49 CPU/GPU --> 0.175
teste 50 CPU --> 0.197 ||	teste 50 GPU --> 0.046 ||	teste 50 CPU/GPU --> 0.170
teste 51 CPU --> 0.200 ||	teste 51 GPU --> 0.047 ||	teste 51 CPU/GPU --> 0.170
teste 52 CPU --> 0.195 ||	teste 52 GPU --> 0.046 ||	teste 52 CPU/GPU --> 0.169
teste 53 CPU --> 0.198 ||	teste 53 GPU --> 0.045 ||	teste 53 CPU/GPU --> 0.170
teste 54 CPU --> 0.199 ||	teste 54 GPU --> 0.046 ||	teste 54 CPU/GPU --> 0.171
teste 55 CPU --> 0.201 ||	teste 55 GPU --> 0.046 ||	teste 55 CPU/GPU --> 0.169

teste 56 CPU --> 0.198 ||	teste 56 GPU --> 0.046 ||	teste 56 CPU/GPU --> 0.172
teste 57 CPU --> 0.202 ||	teste 57 GPU --> 0.046 ||	teste 57 CPU/GPU --> 0.179
teste 58 CPU --> 0.196 ||	teste 58 GPU --> 0.048 ||	teste 58 CPU/GPU --> 0.174
teste 59 CPU --> 0.195 ||	teste 59 GPU --> 0.051 ||	teste 59 CPU/GPU --> 0.171
teste 60 CPU --> 0.204 ||	teste 60 GPU --> 0.049 ||	teste 60 CPU/GPU --> 0.191
teste 61 CPU --> 0.197 ||	teste 61 GPU --> 0.047 ||	teste 61 CPU/GPU --> 0.177
teste 62 CPU --> 0.197 ||	teste 62 GPU --> 0.048 ||	teste 62 CPU/GPU --> 0.169
teste 63 CPU --> 0.196 ||	teste 63 GPU --> 0.048 ||	teste 63 CPU/GPU --> 0.171
teste 64 CPU --> 0.196 ||	teste 64 GPU --> 0.047 ||	teste 64 CPU/GPU --> 0.170
teste 65 CPU --> 0.197 ||	teste 65 GPU --> 0.047 ||	teste 65 CPU/GPU --> 0.169
teste 66 CPU --> 0.197 ||	teste 66 GPU --> 0.047 ||	teste 66 CPU/GPU --> 0.172
teste 67 CPU --> 0.194 ||	teste 67 GPU --> 0.047 ||	teste 67 CPU/GPU --> 0.167
teste 68 CPU --> 0.197 ||	teste 68 GPU --> 0.050 ||	teste 68 CPU/GPU --> 0.169
teste 69 CPU --> 0.193 ||	teste 69 GPU --> 0.049 ||	teste 69 CPU/GPU --> 0.171
teste 70 CPU --> 0.196 ||	teste 70 GPU --> 0.046 ||	teste 70 CPU/GPU --> 0.172
teste 71 CPU --> 0.194 ||	teste 71 GPU --> 0.046 ||	teste 71 CPU/GPU --> 0.194
teste 72 CPU --> 0.198 ||	teste 72 GPU --> 0.046 ||	teste 72 CPU/GPU --> 0.173
teste 73 CPU --> 0.194 ||	teste 73 GPU --> 0.046 ||	teste 73 CPU/GPU --> 0.173
teste 74 CPU --> 0.198 ||	teste 74 GPU --> 0.046 ||	teste 74 CPU/GPU --> 0.169
teste 75 CPU --> 0.196 ||	teste 75 GPU --> 0.049 ||	teste 75 CPU/GPU --> 0.169
teste 76 CPU --> 0.195 ||	teste 76 GPU --> 0.049 ||	teste 76 CPU/GPU --> 0.177
teste 77 CPU --> 0.195 ||	teste 77 GPU --> 0.050 ||	teste 77 CPU/GPU --> 0.142
teste 78 CPU --> 0.197 ||	teste 78 GPU --> 0.050 ||	teste 78 CPU/GPU --> 0.169
teste 79 CPU --> 0.196 ||	teste 79 GPU --> 0.048 ||	teste 79 CPU/GPU --> 0.172
teste 80 CPU --> 0.198 ||	teste 80 GPU --> 0.048 ||	teste 80 CPU/GPU --> 0.170
teste 81 CPU --> 0.193 ||	teste 81 GPU --> 0.051 ||	teste 81 CPU/GPU --> 0.173
teste 82 CPU --> 0.197 ||	teste 82 GPU --> 0.050 ||	teste 82 CPU/GPU --> 0.170
teste 83 CPU --> 0.195 ||	teste 83 GPU --> 0.049 ||	teste 83 CPU/GPU --> 0.174
teste 84 CPU --> 0.199 ||	teste 84 GPU --> 0.046 ||	teste 84 CPU/GPU --> 0.171
teste 85 CPU --> 0.194 ||	teste 85 GPU --> 0.046 ||	teste 85 CPU/GPU --> 0.169
teste 86 CPU --> 0.198 ||	teste 86 GPU --> 0.046 ||	teste 86 CPU/GPU --> 0.169
teste 87 CPU --> 0.192 ||	teste 87 GPU --> 0.046 ||	teste 87 CPU/GPU --> 0.169
teste 88 CPU --> 0.197 ||	teste 88 GPU --> 0.046 ||	teste 88 CPU/GPU --> 0.232
teste 89 CPU --> 0.195 ||	teste 89 GPU --> 0.046 ||	teste 89 CPU/GPU --> 0.171
teste 90 CPU --> 0.199 ||	teste 90 GPU --> 0.046 ||	teste 90 CPU/GPU --> 0.169
teste 91 CPU --> 0.194 ||	teste 91 GPU --> 0.046 ||	teste 91 CPU/GPU --> 0.169
teste 92 CPU --> 0.197 ||	teste 92 GPU --> 0.046 ||	teste 92 CPU/GPU --> 0.171
teste 93 CPU --> 0.194 ||	teste 93 GPU --> 0.046 ||	teste 93 CPU/GPU --> 0.174
teste 94 CPU --> 0.196 ||	teste 94 GPU --> 0.046 ||	teste 94 CPU/GPU --> 0.190
teste 95 CPU --> 0.192 ||	teste 95 GPU --> 0.046 ||	teste 95 CPU/GPU --> 0.171
teste 96 CPU --> 0.195 ||	teste 96 GPU --> 0.045 ||	teste 96 CPU/GPU --> 0.171
teste 97 CPU --> 0.191 ||	teste 97 GPU --> 0.046 ||	teste 97 CPU/GPU --> 0.171
teste 98 CPU --> 0.197 ||	teste 98 GPU --> 0.046 ||	teste 98 CPU/GPU --> 0.172
teste 99 CPU --> 0.194 ||	teste 99 GPU --> 0.048 ||	teste 99 CPU/GPU --> 0.170
teste 100 CPU --> 0.198 ||	teste 100 GPU --> 0.046 ||	teste 100 CPU/GPU --> 0.173

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.201 (+ ou -) 0.016	0.047 (+ ou -) 0.001	0.172 (+ ou -) 0.009

Apêndice G - Resultados Vetor 1.000.000

CPU = 1000000	GPU = 1000000	CPU/GPU = 1000000(500000/500000)
==========	==========	==========
teste 1 CPU --> 3.536 ||	teste 1 GPU --> 0.109 ||	teste 1 CPU/GPU --> 0.984
teste 2 CPU --> 3.562 ||	teste 2 GPU --> 0.126 ||	teste 2 CPU/GPU --> 0.915
teste 3 CPU --> 3.534 ||	teste 3 GPU --> 0.109 ||	teste 3 CPU/GPU --> 0.948
teste 4 CPU --> 3.486 ||	teste 4 GPU --> 0.108 ||	teste 4 CPU/GPU --> 0.955
teste 5 CPU --> 3.501 ||	teste 5 GPU --> 0.108 ||	teste 5 CPU/GPU --> 0.939
teste 6 CPU --> 3.491 ||	teste 6 GPU --> 0.088 ||	teste 6 CPU/GPU --> 0.979
teste 7 CPU --> 3.520 ||	teste 7 GPU --> 0.087 ||	teste 7 CPU/GPU --> 0.996
teste 8 CPU --> 3.507 ||	teste 8 GPU --> 0.086 ||	teste 8 CPU/GPU --> 0.942
teste 9 CPU --> 3.488 ||	teste 9 GPU --> 0.086 ||	teste 9 CPU/GPU --> 0.989
teste 10 CPU --> 3.482 ||	teste 10 GPU --> 0.084 ||	teste 10 CPU/GPU --> 0.940
teste 11 CPU --> 3.603 ||	teste 11 GPU --> 0.085 ||	teste 11 CPU/GPU --> 0.968
teste 12 CPU --> 3.493 ||	teste 12 GPU --> 0.085 ||	teste 12 CPU/GPU --> 0.946
teste 13 CPU --> 3.499 ||	teste 13 GPU --> 0.088 ||	teste 13 CPU/GPU --> 0.984
teste 14 CPU --> 3.494 ||	teste 14 GPU --> 0.085 ||	teste 14 CPU/GPU --> 0.953
teste 15 CPU --> 3.488 ||	teste 15 GPU --> 0.086 ||	teste 15 CPU/GPU --> 0.985
teste 16 CPU --> 3.511 ||	teste 16 GPU --> 0.086 ||	teste 16 CPU/GPU --> 0.936
teste 17 CPU --> 2.663 ||	teste 17 GPU --> 0.086 ||	teste 17 CPU/GPU --> 1.133
teste 18 CPU --> 2.643 ||	teste 18 GPU --> 0.085 ||	teste 18 CPU/GPU --> 0.941
teste 19 CPU --> 2.659 ||	teste 19 GPU --> 0.086 ||	teste 19 CPU/GPU --> 0.975
teste 20 CPU --> 3.009 ||	teste 20 GPU --> 0.085 ||	teste 20 CPU/GPU --> 0.972
teste 21 CPU --> 2.661 ||	teste 21 GPU --> 0.086 ||	teste 21 CPU/GPU --> 0.936
teste 22 CPU --> 2.774 ||	teste 22 GPU --> 0.086 ||	teste 22 CPU/GPU --> 0.989
teste 23 CPU --> 2.657 ||	teste 23 GPU --> 0.086 ||	teste 23 CPU/GPU --> 0.940
teste 24 CPU --> 2.653 ||	teste 24 GPU --> 0.087 ||	teste 24 CPU/GPU --> 0.977
teste 25 CPU --> 3.473 ||	teste 25 GPU --> 0.085 ||	teste 25 CPU/GPU --> 0.937
teste 26 CPU --> 2.635 ||	teste 26 GPU --> 0.084 ||	teste 26 CPU/GPU --> 0.991
teste 27 CPU --> 3.534 ||	teste 27 GPU --> 0.086 ||	teste 27 CPU/GPU --> 0.941
teste 28 CPU --> 2.631 ||	teste 28 GPU --> 0.087 ||	teste 28 CPU/GPU --> 0.984
teste 29 CPU --> 2.643 ||	teste 29 GPU --> 0.086 ||	teste 29 CPU/GPU --> 0.917
teste 30 CPU --> 2.645 ||	teste 30 GPU --> 0.087 ||	teste 30 CPU/GPU --> 0.984
teste 31 CPU --> 2.647 ||	teste 31 GPU --> 0.087 ||	teste 31 CPU/GPU --> 0.920
teste 32 CPU --> 2.645 ||	teste 32 GPU --> 0.086 ||	teste 32 CPU/GPU --> 0.984
teste 33 CPU --> 2.633 ||	teste 33 GPU --> 0.085 ||	teste 33 CPU/GPU --> 0.944
teste 34 CPU --> 3.471 ||	teste 34 GPU --> 0.085 ||	teste 34 CPU/GPU --> 0.935
teste 35 CPU --> 3.587 ||	teste 35 GPU --> 0.089 ||	teste 35 CPU/GPU --> 0.979
teste 36 CPU --> 2.636 ||	teste 36 GPU --> 0.088 ||	teste 36 CPU/GPU --> 0.953
teste 37 CPU --> 2.663 ||	teste 37 GPU --> 0.086 ||	teste 37 CPU/GPU --> 0.982
teste 38 CPU --> 2.647 ||	teste 38 GPU --> 0.086 ||	teste 38 CPU/GPU --> 0.917
teste 39 CPU --> 2.636 ||	teste 39 GPU --> 0.086 ||	teste 39 CPU/GPU --> 1.001
teste 40 CPU --> 2.635 ||	teste 40 GPU --> 0.085 ||	teste 40 CPU/GPU --> 0.938
teste 41 CPU --> 2.663 ||	teste 41 GPU --> 0.086 ||	teste 41 CPU/GPU --> 0.980
teste 42 CPU --> 2.629 ||	teste 42 GPU --> 0.086 ||	teste 42 CPU/GPU --> 0.919
teste 43 CPU --> 2.639 ||	teste 43 GPU --> 0.086 ||	teste 43 CPU/GPU --> 0.984
teste 44 CPU --> 2.633 ||	teste 44 GPU --> 0.087 ||	teste 44 CPU/GPU --> 0.936
teste 45 CPU --> 2.685 ||	teste 45 GPU --> 0.086 ||	teste 45 CPU/GPU --> 0.982
teste 46 CPU --> 2.640 ||	teste 46 GPU --> 0.085 ||	teste 46 CPU/GPU --> 1.093
teste 47 CPU --> 2.634 ||	teste 47 GPU --> 0.086 ||	teste 47 CPU/GPU --> 0.952
teste 48 CPU --> 2.646 ||	teste 48 GPU --> 0.084 ||	teste 48 CPU/GPU --> 0.942
teste 49 CPU --> 2.645 ||	teste 49 GPU --> 0.085 ||	teste 49 CPU/GPU --> 0.968
teste 50 CPU --> 2.624 ||	teste 50 GPU --> 0.085 ||	teste 50 CPU/GPU --> 0.930
teste 51 CPU --> 2.641 ||	teste 51 GPU --> 0.085 ||	teste 51 CPU/GPU --> 0.970
teste 52 CPU --> 2.642 ||	teste 52 GPU --> 0.085 ||	teste 52 CPU/GPU --> 0.934
teste 53 CPU --> 2.631 ||	teste 53 GPU --> 0.088 ||	teste 53 CPU/GPU --> 0.952
teste 54 CPU --> 2.634 ||	teste 54 GPU --> 0.085 ||	teste 54 CPU/GPU --> 0.935
teste 55 CPU --> 2.634 ||	teste 55 GPU --> 0.086 ||	teste 55 CPU/GPU --> 0.959

teste 56 CPU --> 2.674 ||	teste 56 GPU --> 0.086 ||	teste 56 CPU/GPU --> 0.939
teste 57 CPU --> 2.639 ||	teste 57 GPU --> 0.085 ||	teste 57 CPU/GPU --> 0.951
teste 58 CPU --> 2.631 ||	teste 58 GPU --> 0.085 ||	teste 58 CPU/GPU --> 0.929
teste 59 CPU --> 2.637 ||	teste 59 GPU --> 0.086 ||	teste 59 CPU/GPU --> 0.956
teste 60 CPU --> 2.647 ||	teste 60 GPU --> 0.084 ||	teste 60 CPU/GPU --> 0.929
teste 61 CPU --> 2.635 ||	teste 61 GPU --> 0.085 ||	teste 61 CPU/GPU --> 0.949
teste 62 CPU --> 2.633 ||	teste 62 GPU --> 0.086 ||	teste 62 CPU/GPU --> 0.931
teste 63 CPU --> 2.641 ||	teste 63 GPU --> 0.087 ||	teste 63 CPU/GPU --> 0.961
teste 64 CPU --> 2.644 ||	teste 64 GPU --> 0.085 ||	teste 64 CPU/GPU --> 0.931
teste 65 CPU --> 2.635 ||	teste 65 GPU --> 0.086 ||	teste 65 CPU/GPU --> 0.959
teste 66 CPU --> 2.631 ||	teste 66 GPU --> 0.086 ||	teste 66 CPU/GPU --> 0.937
teste 67 CPU --> 2.693 ||	teste 67 GPU --> 0.086 ||	teste 67 CPU/GPU --> 0.963
teste 68 CPU --> 2.644 ||	teste 68 GPU --> 0.089 ||	teste 68 CPU/GPU --> 0.931
teste 69 CPU --> 2.632 ||	teste 69 GPU --> 0.088 ||	teste 69 CPU/GPU --> 0.953
teste 70 CPU --> 2.631 ||	teste 70 GPU --> 0.086 ||	teste 70 CPU/GPU --> 0.904
teste 71 CPU --> 2.690 ||	teste 71 GPU --> 0.086 ||	teste 71 CPU/GPU --> 0.900
teste 72 CPU --> 2.649 ||	teste 72 GPU --> 0.085 ||	teste 72 CPU/GPU --> 0.924
teste 73 CPU --> 2.627 ||	teste 73 GPU --> 0.086 ||	teste 73 CPU/GPU --> 0.954
teste 74 CPU --> 2.636 ||	teste 74 GPU --> 0.085 ||	teste 74 CPU/GPU --> 0.927
teste 75 CPU --> 2.670 ||	teste 75 GPU --> 0.086 ||	teste 75 CPU/GPU --> 0.951
teste 76 CPU --> 2.632 ||	teste 76 GPU --> 0.086 ||	teste 76 CPU/GPU --> 0.902
teste 77 CPU --> 2.630 ||	teste 77 GPU --> 0.088 ||	teste 77 CPU/GPU --> 1.131
teste 78 CPU --> 2.641 ||	teste 78 GPU --> 0.086 ||	teste 78 CPU/GPU --> 0.943
teste 79 CPU --> 2.681 ||	teste 79 GPU --> 0.086 ||	teste 79 CPU/GPU --> 0.964
teste 80 CPU --> 2.647 ||	teste 80 GPU --> 0.084 ||	teste 80 CPU/GPU --> 0.906
teste 81 CPU --> 2.625 ||	teste 81 GPU --> 0.085 ||	teste 81 CPU/GPU --> 0.898
teste 82 CPU --> 2.633 ||	teste 82 GPU --> 0.086 ||	teste 82 CPU/GPU --> 0.923
teste 83 CPU --> 2.660 ||	teste 83 GPU --> 0.086 ||	teste 83 CPU/GPU --> 0.962
teste 84 CPU --> 2.634 ||	teste 84 GPU --> 0.086 ||	teste 84 CPU/GPU --> 0.919
teste 85 CPU --> 2.644 ||	teste 85 GPU --> 0.087 ||	teste 85 CPU/GPU --> 0.954
teste 86 CPU --> 2.667 ||	teste 86 GPU --> 0.086 ||	teste 86 CPU/GPU --> 0.925
teste 87 CPU --> 2.645 ||	teste 87 GPU --> 0.086 ||	teste 87 CPU/GPU --> 0.955
teste 88 CPU --> 2.628 ||	teste 88 GPU --> 0.086 ||	teste 88 CPU/GPU --> 0.955
teste 89 CPU --> 2.638 ||	teste 89 GPU --> 0.086 ||	teste 89 CPU/GPU --> 0.958
teste 90 CPU --> 2.662 ||	teste 90 GPU --> 0.085 ||	teste 90 CPU/GPU --> 0.927
teste 91 CPU --> 2.629 ||	teste 91 GPU --> 0.086 ||	teste 91 CPU/GPU --> 0.954
teste 92 CPU --> 2.631 ||	teste 92 GPU --> 0.085 ||	teste 92 CPU/GPU --> 0.929
teste 93 CPU --> 2.638 ||	teste 93 GPU --> 0.086 ||	teste 93 CPU/GPU --> 0.957
teste 94 CPU --> 2.667 ||	teste 94 GPU --> 0.088 ||	teste 94 CPU/GPU --> 0.904
teste 95 CPU --> 2.635 ||	teste 95 GPU --> 0.086 ||	teste 95 CPU/GPU --> 0.901
teste 96 CPU --> 2.634 ||	teste 96 GPU --> 0.086 ||	teste 96 CPU/GPU --> 0.921
teste 97 CPU --> 2.629 ||	teste 97 GPU --> 0.086 ||	teste 97 CPU/GPU --> 0.952
teste 98 CPU --> 2.648 ||	teste 98 GPU --> 0.084 ||	teste 98 CPU/GPU --> 0.943
teste 99 CPU --> 2.632 ||	teste 99 GPU --> 0.085 ||	teste 99 CPU/GPU --> 0.958
teste 100 CPU --> 2.624 ||	teste 100 GPU --> 0.086 ||	teste 100 CPU/GPU --> 0.908

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
2.822 (+ ou -) 0.350	0.087 (+ ou -) 0.006	0.953 (+ ou -) 0.038

Apêndice H - Resultados Vetor 10.000.000

CPU = 10000000	GPU = 10000000	CPU/GPU = 10000000(5000000/5000000)
==========	==========	==========
teste 1 CPU --> 26.457 ||	teste 1 GPU --> 0.524 ||	teste 1 CPU/GPU --> 8.800
teste 2 CPU --> 26.516 ||	teste 2 GPU --> 0.511 ||	teste 2 CPU/GPU --> 11.544
teste 3 CPU --> 26.404 ||	teste 3 GPU --> 0.507 ||	teste 3 CPU/GPU --> 8.807
teste 4 CPU --> 26.392 ||	teste 4 GPU --> 0.502 ||	teste 4 CPU/GPU --> 9.041
teste 5 CPU --> 26.188 ||	teste 5 GPU --> 0.508 ||	teste 5 CPU/GPU --> 8.913
teste 6 CPU --> 26.258 ||	teste 6 GPU --> 0.503 ||	teste 6 CPU/GPU --> 9.504
teste 7 CPU --> 26.399 ||	teste 7 GPU --> 0.501 ||	teste 7 CPU/GPU --> 8.941
teste 8 CPU --> 26.938 ||	teste 8 GPU --> 0.508 ||	teste 8 CPU/GPU --> 8.762
teste 9 CPU --> 26.146 ||	teste 9 GPU --> 0.502 ||	teste 9 CPU/GPU --> 8.932
teste 10 CPU --> 26.221 ||	teste 10 GPU --> 0.501 ||	teste 10 CPU/GPU --> 8.875
teste 11 CPU --> 26.334 ||	teste 11 GPU --> 0.500 ||	teste 11 CPU/GPU --> 8.768
teste 12 CPU --> 26.311 ||	teste 12 GPU --> 0.504 ||	teste 12 CPU/GPU --> 8.847
teste 13 CPU --> 34.655 ||	teste 13 GPU --> 0.503 ||	teste 13 CPU/GPU --> 8.785
teste 14 CPU --> 34.692 ||	teste 14 GPU --> 0.504 ||	teste 14 CPU/GPU --> 8.799
teste 15 CPU --> 34.671 ||	teste 15 GPU --> 0.549 ||	teste 15 CPU/GPU --> 8.934
teste 16 CPU --> 34.789 ||	teste 16 GPU --> 0.507 ||	teste 16 CPU/GPU --> 8.775
teste 17 CPU --> 36.323 ||	teste 17 GPU --> 0.527 ||	teste 17 CPU/GPU --> 8.722
teste 18 CPU --> 34.751 ||	teste 18 GPU --> 0.509 ||	teste 18 CPU/GPU --> 8.746
teste 19 CPU --> 34.891 ||	teste 19 GPU --> 0.508 ||	teste 19 CPU/GPU --> 9.188
teste 20 CPU --> 34.615 ||	teste 20 GPU --> 0.507 ||	teste 20 CPU/GPU --> 8.914
teste 21 CPU --> 34.695 ||	teste 21 GPU --> 0.502 ||	teste 21 CPU/GPU --> 9.810
teste 22 CPU --> 34.731 ||	teste 22 GPU --> 0.507 ||	teste 22 CPU/GPU --> 9.756
teste 23 CPU --> 34.662 ||	teste 23 GPU --> 0.502 ||	teste 23 CPU/GPU --> 8.794
teste 24 CPU --> 34.687 ||	teste 24 GPU --> 0.502 ||	teste 24 CPU/GPU --> 8.855
teste 25 CPU --> 26.262 ||	teste 25 GPU --> 0.509 ||	teste 25 CPU/GPU --> 8.893
teste 26 CPU --> 26.196 ||	teste 26 GPU --> 0.503 ||	teste 26 CPU/GPU --> 8.791
teste 27 CPU --> 26.159 ||	teste 27 GPU --> 0.502 ||	teste 27 CPU/GPU --> 8.992
teste 28 CPU --> 26.097 ||	teste 28 GPU --> 0.509 ||	teste 28 CPU/GPU --> 8.756
teste 29 CPU --> 26.385 ||	teste 29 GPU --> 0.503 ||	teste 29 CPU/GPU --> 8.893
teste 30 CPU --> 26.149 ||	teste 30 GPU --> 0.503 ||	teste 30 CPU/GPU --> 9.089
teste 31 CPU --> 26.393 ||	teste 31 GPU --> 0.501 ||	teste 31 CPU/GPU --> 8.826
teste 32 CPU --> 26.665 ||	teste 32 GPU --> 0.498 ||	teste 32 CPU/GPU --> 8.805
teste 33 CPU --> 29.079 ||	teste 33 GPU --> 0.503 ||	teste 33 CPU/GPU --> 8.818
teste 34 CPU --> 26.272 ||	teste 34 GPU --> 0.535 ||	teste 34 CPU/GPU --> 8.815
teste 35 CPU --> 26.186 ||	teste 35 GPU --> 0.536 ||	teste 35 CPU/GPU --> 8.863
teste 36 CPU --> 26.231 ||	teste 36 GPU --> 0.515 ||	teste 36 CPU/GPU --> 8.765
teste 37 CPU --> 26.216 ||	teste 37 GPU --> 0.513 ||	teste 37 CPU/GPU --> 8.747
teste 38 CPU --> 26.493 ||	teste 38 GPU --> 0.528 ||	teste 38 CPU/GPU --> 8.747
teste 39 CPU --> 30.049 ||	teste 39 GPU --> 0.514 ||	teste 39 CPU/GPU --> 8.977
teste 40 CPU --> 26.211 ||	teste 40 GPU --> 0.503 ||	teste 40 CPU/GPU --> 9.070
teste 41 CPU --> 26.261 ||	teste 41 GPU --> 1.227 ||	teste 41 CPU/GPU --> 8.926
teste 42 CPU --> 26.277 ||	teste 42 GPU --> 0.526 ||	teste 42 CPU/GPU --> 8.890
teste 43 CPU --> 26.424 ||	teste 43 GPU --> 0.497 ||	teste 43 CPU/GPU --> 9.230
teste 44 CPU --> 26.330 ||	teste 44 GPU --> 0.500 ||	teste 44 CPU/GPU --> 8.797
teste 45 CPU --> 26.242 ||	teste 45 GPU --> 0.501 ||	teste 45 CPU/GPU --> 8.839
teste 46 CPU --> 34.634 ||	teste 46 GPU --> 0.500 ||	teste 46 CPU/GPU --> 8.754
teste 47 CPU --> 34.764 ||	teste 47 GPU --> 0.501 ||	teste 47 CPU/GPU --> 8.864
teste 48 CPU --> 34.782 ||	teste 48 GPU --> 0.502 ||	teste 48 CPU/GPU --> 9.075
teste 49 CPU --> 34.705 ||	teste 49 GPU --> 0.499 ||	teste 49 CPU/GPU --> 8.856
teste 50 CPU --> 26.187 ||	teste 50 GPU --> 0.501 ||	teste 50 CPU/GPU --> 8.810
teste 51 CPU --> 26.245 ||	teste 51 GPU --> 0.501 ||	teste 51 CPU/GPU --> 9.497
teste 52 CPU --> 26.321 ||	teste 52 GPU --> 0.526 ||	teste 52 CPU/GPU --> 8.897
teste 53 CPU --> 26.252 ||	teste 53 GPU --> 0.500 ||	teste 53 CPU/GPU --> 8.798
teste 54 CPU --> 26.203 ||	teste 54 GPU --> 0.502 ||	teste 54 CPU/GPU --> 8.766
teste 55 CPU --> 26.400 ||	teste 55 GPU --> 0.499 ||	teste 55 CPU/GPU --> 8.898

teste 56 CPU --> 26.313 ||	teste 56 GPU --> 0.501 ||	teste 56 CPU/GPU --> 8.764
teste 57 CPU --> 34.682 ||	teste 57 GPU --> 0.501 ||	teste 57 CPU/GPU --> 8.774
teste 58 CPU --> 34.730 ||	teste 58 GPU --> 0.504 ||	teste 58 CPU/GPU --> 8.928
teste 59 CPU --> 26.203 ||	teste 59 GPU --> 0.501 ||	teste 59 CPU/GPU --> 8.852
teste 60 CPU --> 26.131 ||	teste 60 GPU --> 0.502 ||	teste 60 CPU/GPU --> 8.936
teste 61 CPU --> 26.182 ||	teste 61 GPU --> 0.521 ||	teste 61 CPU/GPU --> 9.017
teste 62 CPU --> 26.168 ||	teste 62 GPU --> 0.501 ||	teste 62 CPU/GPU --> 8.964
teste 63 CPU --> 26.238 ||	teste 63 GPU --> 0.508 ||	teste 63 CPU/GPU --> 8.815
teste 64 CPU --> 26.271 ||	teste 64 GPU --> 0.505 ||	teste 64 CPU/GPU --> 8.810
teste 65 CPU --> 26.115 ||	teste 65 GPU --> 0.498 ||	teste 65 CPU/GPU --> 8.935
teste 66 CPU --> 26.456 ||	teste 66 GPU --> 0.498 ||	teste 66 CPU/GPU --> 8.807
teste 67 CPU --> 26.364 ||	teste 67 GPU --> 0.500 ||	teste 67 CPU/GPU --> 8.783
teste 68 CPU --> 35.220 ||	teste 68 GPU --> 0.495 ||	teste 68 CPU/GPU --> 8.810
teste 69 CPU --> 34.672 ||	teste 69 GPU --> 0.494 ||	teste 69 CPU/GPU --> 8.927
teste 70 CPU --> 35.783 ||	teste 70 GPU --> 0.501 ||	teste 70 CPU/GPU --> 8.805
teste 71 CPU --> 28.547 ||	teste 71 GPU --> 0.499 ||	teste 71 CPU/GPU --> 8.799
teste 72 CPU --> 26.182 ||	teste 72 GPU --> 0.501 ||	teste 72 CPU/GPU --> 8.876
teste 73 CPU --> 26.103 ||	teste 73 GPU --> 0.500 ||	teste 73 CPU/GPU --> 8.774
teste 74 CPU --> 26.169 ||	teste 74 GPU --> 0.503 ||	teste 74 CPU/GPU --> 8.750
teste 75 CPU --> 34.740 ||	teste 75 GPU --> 0.498 ||	teste 75 CPU/GPU --> 9.073
teste 76 CPU --> 34.736 ||	teste 76 GPU --> 0.498 ||	teste 76 CPU/GPU --> 8.894
teste 77 CPU --> 34.629 ||	teste 77 GPU --> 0.499 ||	teste 77 CPU/GPU --> 8.809
teste 78 CPU --> 34.607 ||	teste 78 GPU --> 0.502 ||	teste 78 CPU/GPU --> 8.767
teste 79 CPU --> 34.729 ||	teste 79 GPU --> 0.500 ||	teste 79 CPU/GPU --> 8.891
teste 80 CPU --> 34.846 ||	teste 80 GPU --> 0.504 ||	teste 80 CPU/GPU --> 8.986
teste 81 CPU --> 26.144 ||	teste 81 GPU --> 0.499 ||	teste 81 CPU/GPU --> 8.765
teste 82 CPU --> 26.168 ||	teste 82 GPU --> 0.501 ||	teste 82 CPU/GPU --> 8.786
teste 83 CPU --> 26.168 ||	teste 83 GPU --> 0.500 ||	teste 83 CPU/GPU --> 8.943
teste 84 CPU --> 26.267 ||	teste 84 GPU --> 0.505 ||	teste 84 CPU/GPU --> 8.781
teste 85 CPU --> 28.643 ||	teste 85 GPU --> 0.500 ||	teste 85 CPU/GPU --> 8.790
teste 86 CPU --> 26.265 ||	teste 86 GPU --> 0.502 ||	teste 86 CPU/GPU --> 8.877
teste 87 CPU --> 26.153 ||	teste 87 GPU --> 0.501 ||	teste 87 CPU/GPU --> 9.092
teste 88 CPU --> 26.203 ||	teste 88 GPU --> 0.500 ||	teste 88 CPU/GPU --> 8.824
teste 89 CPU --> 26.184 ||	teste 89 GPU --> 0.499 ||	teste 89 CPU/GPU --> 8.969
teste 90 CPU --> 26.163 ||	teste 90 GPU --> 0.495 ||	teste 90 CPU/GPU --> 8.762
teste 91 CPU --> 34.663 ||	teste 91 GPU --> 0.500 ||	teste 91 CPU/GPU --> 8.759
teste 92 CPU --> 34.661 ||	teste 92 GPU --> 0.503 ||	teste 92 CPU/GPU --> 8.750
teste 93 CPU --> 26.351 ||	teste 93 GPU --> 0.499 ||	teste 93 CPU/GPU --> 8.942
teste 94 CPU --> 34.655 ||	teste 94 GPU --> 0.501 ||	teste 94 CPU/GPU --> 8.808
teste 95 CPU --> 34.645 ||	teste 95 GPU --> 0.500 ||	teste 95 CPU/GPU --> 8.748
teste 96 CPU --> 34.623 ||	teste 96 GPU --> 0.496 ||	teste 96 CPU/GPU --> 9.635
teste 97 CPU --> 26.141 ||	teste 97 GPU --> 0.501 ||	teste 97 CPU/GPU --> 10.726
teste 98 CPU --> 26.285 ||	teste 98 GPU --> 0.494 ||	teste 98 CPU/GPU --> 13.242
teste 99 CPU --> 26.242 ||	teste 99 GPU --> 0.513 ||	teste 99 CPU/GPU --> 12.811
teste 100 CPU --> 26.197 ||	teste 100 GPU --> 0.503 ||	teste 100 CPU/GPU --> 8.770

==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
29.114 (+ ou -) 3.968	0.512 (+ ou -) 0.073	9.029 (+ ou -) 0.687

Apêndice I - Resultados dos testes com dependência de dados

O Kernel utilizado os testes com dependência de dados está descrito na figura.

Figura 32 - Kernel utilizado para os testes com dependência de dados

```
__kernel void ArrayDiff(__global const int* a, __global const int* b,__global int* c)
 {
        int id = get_global_id(0);
        if(id > 0){
                c[id] = (a[id] - b[id]) + c[id-1];
        }else{
                c[id] = (a[id] - b[id]);
        }
 }";
```

Fonte: Elaborada pelo autor.

Resultado dos testes:

Vetor com 1.000 posições:

CPU = 1000	GPU = 1000	CPU/GPU = 1000(500/500)
==========	==========	==========
teste 1 CPU --> 0.012 ||	teste 1 GPU --> 0.056 ||	teste 1 CPU/GPU --> 0.062
teste 2 CPU --> 0.011 ||	teste 2 GPU --> 0.045 ||	teste 2 CPU/GPU --> 0.049
teste 3 CPU --> 0.022 ||	teste 3 GPU --> 0.051 ||	teste 3 CPU/GPU --> 0.047
teste 4 CPU --> 0.011 ||	teste 4 GPU --> 0.044 ||	teste 4 CPU/GPU --> 0.045
teste 5 CPU --> 0.023 ||	teste 5 GPU --> 0.044 ||	teste 5 CPU/GPU --> 0.047
teste 6 CPU --> 0.011 ||	teste 6 GPU --> 0.043 ||	teste 6 CPU/GPU --> 0.045
teste 7 CPU --> 0.012 ||	teste 7 GPU --> 0.042 ||	teste 7 CPU/GPU --> 0.047
teste 8 CPU --> 0.011 ||	teste 8 GPU --> 0.042 ||	teste 8 CPU/GPU --> 0.046
teste 9 CPU --> 0.011 ||	teste 9 GPU --> 0.042 ||	teste 9 CPU/GPU --> 0.046
teste 10 CPU --> 0.012 ||	teste 10 GPU --> 0.042 ||	teste 10 CPU/GPU --> 0.045
==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.014 (+ ou -) 0.005	0.045 (+ ou -) 0.005	0.048 (+ ou -) 0.005

Vetor com 10.000 posições

CPU = 10000	GPU = 10000	CPU/GPU = 10000(5000/5000)
==========	==========	==========
teste 1 CPU --> 0.029 ||	teste 1 GPU --> 0.070 ||	teste 1 CPU/GPU --> 0.094
teste 2 CPU --> 0.038 ||	teste 2 GPU --> 0.053 ||	teste 2 CPU/GPU --> 0.044
teste 3 CPU --> 0.036 ||	teste 3 GPU --> 0.093 ||	teste 3 CPU/GPU --> 0.043
teste 4 CPU --> 0.047 ||	teste 4 GPU --> 0.048 ||	teste 4 CPU/GPU --> 0.042
teste 5 CPU --> 0.033 ||	teste 5 GPU --> 0.097 ||	teste 5 CPU/GPU --> 0.063
teste 6 CPU --> 0.033 ||	teste 6 GPU --> 0.044 ||	teste 6 CPU/GPU --> 0.047
teste 7 CPU --> 0.033 ||	teste 7 GPU --> 0.043 ||	teste 7 CPU/GPU --> 0.043
teste 8 CPU --> 0.027 ||	teste 8 GPU --> 0.049 ||	teste 8 CPU/GPU --> 0.050
teste 9 CPU --> 0.028 ||	teste 9 GPU --> 0.040 ||	teste 9 CPU/GPU --> 0.048
teste 10 CPU --> 0.028 ||	teste 10 GPU --> 0.043 ||	teste 10 CPU/GPU --> 0.044
==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU

0.033 (+ ou -) 0.006	0.058 (+ ou -) 0.021	0.052 (+ ou -) 0.016

Vetor com 100.000 posições

CPU = 100000	GPU = 100000	CPU/GPU = 100000(50000/50000)
==========	==========	==========
teste 1 CPU --> 0.234 ||	teste 1 GPU --> 0.134 ||	teste 1 CPU/GPU --> 0.179
teste 2 CPU --> 0.230 ||	teste 2 GPU --> 0.064 ||	teste 2 CPU/GPU --> 0.140
teste 3 CPU --> 0.250 ||	teste 3 GPU --> 0.065 ||	teste 3 CPU/GPU --> 0.141
teste 4 CPU --> 0.327 ||	teste 4 GPU --> 0.059 ||	teste 4 CPU/GPU --> 0.143
teste 5 CPU --> 0.208 ||	teste 5 GPU --> 0.091 ||	teste 5 CPU/GPU --> 0.142
teste 6 CPU --> 0.209 ||	teste 6 GPU --> 0.098 ||	teste 6 CPU/GPU --> 0.142
teste 7 CPU --> 0.225 ||	teste 7 GPU --> 0.078 ||	teste 7 CPU/GPU --> 0.142
teste 8 CPU --> 0.323 ||	teste 8 GPU --> 0.083 ||	teste 8 CPU/GPU --> 0.141
teste 9 CPU --> 0.213 ||	teste 9 GPU --> 0.078 ||	teste 9 CPU/GPU --> 0.142
teste 10 CPU --> 0.214 ||	teste 10 GPU --> 0.050 ||	teste 10 CPU/GPU --> 0.142
==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
0.243 (+ ou -) 0.045	0.080 (+ ou -) 0.024	0.145 (+ ou -) 0.012

Vetor com 1.000.000 posições

CPU = 1000000	GPU = 1000000	CPU/GPU = 1000000(500000/500000)
==========	==========	==========
teste 1 CPU --> 1.924 ||	teste 1 GPU --> 0.202 ||	teste 1 CPU/GPU --> 1.153
teste 2 CPU --> 1.968 ||	teste 2 GPU --> 0.106 ||	teste 2 CPU/GPU --> 1.097
teste 3 CPU --> 2.780 ||	teste 3 GPU --> 0.104 ||	teste 3 CPU/GPU --> 1.103
teste 4 CPU --> 2.403 ||	teste 4 GPU --> 0.103 ||	teste 4 CPU/GPU --> 1.100
teste 5 CPU --> 1.937 ||	teste 5 GPU --> 0.104 ||	teste 5 CPU/GPU --> 1.091
teste 6 CPU --> 1.927 ||	teste 6 GPU --> 0.104 ||	teste 6 CPU/GPU --> 1.068
teste 7 CPU --> 1.953 ||	teste 7 GPU --> 0.104 ||	teste 7 CPU/GPU --> 1.084
teste 8 CPU --> 1.921 ||	teste 8 GPU --> 0.103 ||	teste 8 CPU/GPU --> 1.200
teste 9 CPU --> 1.917 ||	teste 9 GPU --> 0.112 ||	teste 9 CPU/GPU --> 1.087
teste 10 CPU --> 2.075 ||	teste 10 GPU --> 0.105 ||	teste 10 CPU/GPU --> 1.067
==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
2.081 (+ ou -) 0.287	0.115 (+ ou -) 0.031	1.105 (+ ou -) 0.041

Vetor com 10.000.000 posições

CPU = 10000000	GPU = 10000000	CPU/GPU = 10000000(5000000/5000000)
==========	==========	==========
teste 1 CPU --> 28.673 ||	teste 1 GPU --> 0.915 ||	teste 1 CPU/GPU --> 10.342
teste 2 CPU --> 28.375 ||	teste 2 GPU --> 0.639 ||	teste 2 CPU/GPU --> 10.316
teste 3 CPU --> 28.455 ||	teste 3 GPU --> 0.634 ||	teste 3 CPU/GPU --> 10.062
teste 4 CPU --> 28.635 ||	teste 4 GPU --> 0.637 ||	teste 4 CPU/GPU --> 10.268
teste 5 CPU --> 28.371 ||	teste 5 GPU --> 0.635 ||	teste 5 CPU/GPU --> 10.075
teste 6 CPU --> 28.580 ||	teste 6 GPU --> 0.652 ||	teste 6 CPU/GPU --> 12.863
teste 7 CPU --> 28.668 ||	teste 7 GPU --> 0.657 ||	teste 7 CPU/GPU --> 20.962
teste 8 CPU --> 29.000 ||	teste 8 GPU --> 0.650 ||	teste 8 CPU/GPU --> 15.948
teste 9 CPU --> 29.719 ||	teste 9 GPU --> 0.666 ||	teste 9 CPU/GPU --> 10.706
teste 10 CPU --> 37.608 ||	teste 10 GPU --> 0.665 ||	teste 10 CPU/GPU --> 10.285
==========	==========	==========
Tempo medio CPU	Tempo medio GPU	Tempo medio CPU/GPU
29.608 (+ ou -) 2.839	0.675 (+ ou -) 0.085	12.183 (+ ou -) 3.605