

A Produção do Conhecimento na Engenharia da Computação

**Ernane Rosa Martins
(Organizador)**

 **Atena**
Editora
Ano 2019

Ernane Rosa Martins
(Organizador)

A Produção do Conhecimento na Engenharia da Computação

Atena Editora
2019

2019 by Atena Editora
Copyright © Atena Editora
Copyright do Texto © 2019 Os Autores
Copyright da Edição © 2019 Atena Editora
Editora Executiva: Prof^a Dr^a Antonella Carvalho de Oliveira
Diagramação: Rafael Sandrini Filho
Edição de Arte: Lorena Prestes
Revisão: Os Autores

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores. Permitido o download da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

Conselho Editorial

Ciências Humanas e Sociais Aplicadas

Prof. Dr. Álvaro Augusto de Borba Barreto – Universidade Federal de Pelotas
Prof. Dr. Antonio Carlos Frasson – Universidade Tecnológica Federal do Paraná
Prof. Dr. Antonio Isidro-Filho – Universidade de Brasília
Prof. Dr. Constantino Ribeiro de Oliveira Junior – Universidade Estadual de Ponta Grossa
Prof^a Dr^a Cristina Gaio – Universidade de Lisboa
Prof. Dr. Deyvison de Lima Oliveira – Universidade Federal de Rondônia
Prof. Dr. Gilmei Fleck – Universidade Estadual do Oeste do Paraná
Prof^a Dr^a Ivone Goulart Lopes – Istituto Internazionale delle Figlie de Maria Ausiliatrice
Prof^a Dr^a Juliane Sant’Ana Bento – Universidade Federal do Rio Grande do Sul
Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense
Prof^a Dr^a Lina Maria Gonçalves – Universidade Federal do Tocantins
Prof^a Dr^a Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Prof^a Dr^a Paola Andressa Scortegagna – Universidade Estadual de Ponta Grossa
Prof. Dr. Urandi João Rodrigues Junior – Universidade Federal do Oeste do Pará
Prof^a Dr^a Vanessa Bordin Viera – Universidade Federal de Campina Grande
Prof. Dr. Willian Douglas Guilherme – Universidade Federal do Tocantins

Ciências Agrárias e Multidisciplinar

Prof. Dr. Alan Mario Zuffo – Universidade Federal de Mato Grosso do Sul
Prof. Dr. Alexandre Igor Azevedo Pereira – Instituto Federal Goiano
Prof^a Dr^a Daiane Garabeli Trojan – Universidade Norte do Paraná
Prof. Dr. Darllan Collins da Cunha e Silva – Universidade Estadual Paulista
Prof. Dr. Fábio Steiner – Universidade Estadual de Mato Grosso do Sul
Prof^a Dr^a Girlene Santos de Souza – Universidade Federal do Recôncavo da Bahia
Prof. Dr. Jorge González Aguilera – Universidade Federal de Mato Grosso do Sul
Prof. Dr. Ronilson Freitas de Souza – Universidade do Estado do Pará
Prof. Dr. Valdemar Antonio Paffaro Junior – Universidade Federal de Alfenas

Ciências Biológicas e da Saúde

Prof. Dr. Gianfábio Pimentel Franco – Universidade Federal de Santa Maria
Prof. Dr. Benedito Rodrigues da Silva Neto – Universidade Federal de Goiás
Prof.^a Dr.^a Elane Schwinden Prudêncio – Universidade Federal de Santa Catarina
Prof. Dr. José Max Barbosa de Oliveira Junior – Universidade Federal do Oeste do Pará
Prof.^a Dr.^a Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Prof.^a Dr.^a Raissa Rachel Salustriano da Silva Matos – Universidade Federal do Maranhão
Prof.^a Dr.^a Vanessa Lima Gonçalves – Universidade Estadual de Ponta Grossa
Prof.^a Dr.^a Vanessa Bordin Viera – Universidade Federal de Campina Grande

Ciências Exatas e da Terra e Engenharias

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará
Prof.^a Dr.^a Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista

Conselho Técnico Científico

Prof. Msc. Abrãao Carvalho Nogueira – Universidade Federal do Espírito Santo
Prof.^a Dr.^a Andreza Lopes – Instituto de Pesquisa e Desenvolvimento Acadêmico
Prof. Msc. Carlos Antônio dos Santos – Universidade Federal Rural do Rio de Janeiro
Prof.^a Msc. Jaqueline Oliveira Rezende – Universidade Federal de Uberlândia
Prof. Msc. Leonardo Tullio – Universidade Estadual de Ponta Grossa
Prof. Dr. Welleson Feitosa Gazel – Universidade Paulista
Prof. Msc. André Flávio Gonçalves Silva – Universidade Federal do Maranhão
Prof.^a Msc. Renata Luciane Polsaque Young Blood – UniSecal
Prof. Msc. Daniel da Silva Miranda – Universidade Federal do Pará

Dados Internacionais de Catalogação na Publicação (CIP) (eDOC BRASIL, Belo Horizonte/MG)	
P964	A produção do conhecimento na engenharia da computação [recurso eletrônico] / Organizador Ernane Rosa Martins. – Ponta Grossa, PR: Atena Editora, 2019. Formato: PDF Requisitos de sistema: Adobe Acrobat Reader Modo de acesso: World Wide Web Inclui bibliografia ISBN 978-85-7247-339-2 DOI 10.22533/at.ed392192405 1. Computação – Pesquisa – Brasil. 2. Sistemas de informação gerencial. 3. Tecnologia da informação. I. Martins, Ernane Rosa. CDD 004
Elaborado por Maurício Amormino Júnior – CRB6/2422	

Atena Editora
Ponta Grossa – Paraná - Brasil
www.atenaeditora.com.br
contato@atenaeditora.com.br

APRESENTAÇÃO

Segundo o dicionário Aurélio a Engenharia é a “Arte de aplicar conhecimentos científicos e empíricos e certas habilitações específicas à criação de estruturas, dispositivos e processos que se utilizam para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas. A Engenharia de Computação é definida como o ramo da engenharia que se caracteriza pelo projeto, desenvolvimento e implementação de sistemas, equipamentos e dispositivos computacionais segundo uma visão integrada de hardware e software, apoiando-se em uma sólida base matemática e conhecimentos de fenômenos físicos.

Este livro, possibilita conhecer algumas das produções do conhecimento no ramo da Engenharia da Computação, que abordam assuntos extremamente importantes, tais como: as transformações sofridas nos processos de projeto desde a implementação das ferramentas digitais; o armazenamento, indexação e recuperação de formulários digitais; a reabilitação motora assistida por computadores; a reflexão acerca do realismo e da representação visual em jogos digitais; os padrões de players em ambientes virtuais; as soluções tecnológicas relevantes usadas em países africanos; a complexa relação existente entre jogos digitais e o humano; a dinâmica da comunicação de um grupo de Facebook criado em um processo de urbanismo bottom-up; o estado da arte das pesquisas e estudos acadêmicos acerca dos elementos visuais contidos na interface de jogos digitais; as estratégias de design que integrem tecnologia computacional digital a artefatos e instalações para a interação de visitantes em museus; os jogos que abordam o tema de mitologia e religião.

Deste modo, espero que este livro seja um guia para os Engenheiros de Computação auxiliando-os em assuntos relevantes da área, fornecendo conhecimentos que podem permitir especificar, conceber, desenvolver, implementar, adaptar, produzir, industrializar, instalar e manter sistemas computacionais, bem como perfazer a integração de recursos físicos e lógicos necessários para o atendimento das necessidades informacionais, computacionais e da automação de organizações em geral. Por fim, agradeço a todos que contribuíram de alguma forma para a construção desta obra e desejo a todos os leitores, novas e significativas reflexões sobre os temas abordados.

Ernane Rosa Martins

SUMÁRIO

CAPÍTULO 1	1
VIRTUAL REALITY AS A TOOL TO REGAIN TACTUAL PROCEDURES IN DIGITAL DESIGN	
Tales Lobosco	
DOI 10.22533/at.ed3921924051	
CAPÍTULO 2	15
UMA PROPOSTA DE SISTEMA DE BUSCA PARA RECUPERAÇÃO DE FORMULÁRIOS DIGITAIS	
Afonso Henrique Anastácio Calábria	
Talles Brito Viana	
DOI 10.22533/at.ed3921924052	
CAPÍTULO 3	26
REVISÃO SISTEMÁTICA: APLICABILIDADE DO MS KINECT EM REABILITAÇÃO MOTORA	
Tiago Pereira Remédio	
Alexandro José Baldassin	
DOI 10.22533/at.ed3921924053	
CAPÍTULO 4	43
REFLEXÕES ACERCA DO REALISMO E DA REPRESENTAÇÃO VISUAL EM GAMES	
TENDÊNCIAS DE MERCADO E JOGOS AAA	
Ana Carolina Generoso de Aquino	
Rosane de Fatima Antunes Obregon	
Heitor Dias Couto	
DOI 10.22533/at.ed3921924054	
CAPÍTULO 5	52
PLAYER GAME DATA MINING FOR PLAYER CLASSIFICATION	
Bruno Almeida Odierna	
Ismar Frango Silveira	
DOI 10.22533/at.ed3921924055	
CAPÍTULO 6	62
INTERAÇÃO DA TECNOLGIA NA ÁFRICA	
Welington dos Santos Ayres	
DOI 10.22533/at.ed3921924056	
CAPÍTULO 7	69
INTEGRAÇÃO DA TECNOLOGIA CUDA AO MODELO DE PREVISÃO DO TEMPO ETA	
Henrique Gavioli Flores	
Alex Lima de Mello	
Marcelo Trindade Rebonatto	
Carlos Amaral Hölbig	
DOI 10.22533/at.ed3921924057	
CAPÍTULO 8	84
HORIZONTES DA PESQUISA EM CULTURA DE GAMES SOB A ESTÉTICA DA PRODUÇÃO	
Nilson Valdevino Soares	
Luís Carlos Petry	
Guilherme Sousa Vieira	

Ana Carolina Simões de Freitas Cabral
Felipe Blanco
Saulo de Oliveira Machado
José Guilherme dos Santos Gomes

DOI 10.22533/at.ed3921924058

CAPÍTULO 9 100

HORIZONTAL DIALOGUES AND OPEN DATA: THE COMMUNICATION SPACES OF BOTTOM-UP URBANISM.

José Eduardo Calijuri Hamra

DOI 10.22533/at.ed3921924059

CAPÍTULO 10 115

ELEMENTOS VISUAIS EM JOGOS DIGITAIS: UMA REVISÃO SISTEMÁTICA DA LITERATURA.

Ana Carolina Generoso de Aquino

Rosane de Fatima Antunes Obregon

DOI 10.22533/at.ed39219240510

CAPÍTULO 11 131

MEDIAÇÃO DE CONTEÚDO E TECNOLOGIA DIGITAL EM MUSEUS: ESTRATÉGIAS PROJETAIS PARA ENRIQUECIMENTO DA EXPERIÊNCIA DO VISITANTE.

Diego Enéas Peres Ricca

Clíce de Toledo Sanjar Mazzilli

DOI 10.22533/at.ed39219240511

CAPÍTULO 12 151

BRINCANDO COM OS DEUSES: A VIABILIDADE DA DISSEMINAÇÃO DA CULTURA FOLCLÓRICA E POPULAR AFRO-BRASILEIRA EM JOGOS DIGITAIS.

Igor Rocha dos Santos

Marcos Wendell S. de O. Santos

Larissa Cardillo Acconcia Dias

Maurício Acconcia Dias

DOI 10.22533/at.ed39219240512

CAPÍTULO 13 166

A OBRA DANTESCA E SEMIOSES DA CULTURA DE JOGOS DE VIDEOGAME: REFLEXOS EM QUESTÕES DE LETRAMENTO

Caio Túlio Olímpio Pereira da Costa

Leandro Paz da Silva

DOI 10.22533/at.ed39219240513

CAPÍTULO 14 176

A BATTLING BEHAVIOR ANALYSIS OF SHOOTER GAMES BOTS BASED ON THE BARTLE'S PLAYER TYPES AND FINITE STATE MACHINES

Felipe Oviedo Frosi

Isabel Cristina Siqueira da Silva

DOI 10.22533/at.ed39219240514

SOBRE O ORGANIZADOR..... 194

DA TECNOLOGIA CUDA AO MODELO DE PREVISÃO DO TEMPO ETA

Henrique Gavioli Flores

Universidade de Passo Fundo

Passo Fundo – Rio Grande do Sul

Alex Lima de Mello

Universidade de Passo Fundo

Passo Fundo – Rio Grande do Sul

Marcelo Trindade Rebonatto

Universidade de Passo Fundo

Passo Fundo – Rio Grande do Sul

Carlos Amaral Hölbig

Universidade de Passo Fundo

Passo Fundo – Rio Grande do Sul

RESUMO: Modelos de previsão numérica do tempo fazem uso de uma grande quantidade de dados e, devido à alta complexidade das operações realizadas, torna-se demorada a obtenção do resultado. O Eta é um modelo de previsão numérica do tempo utilizado no Brasil pelo CPTEC/INPE que auxilia na tomada de decisão em diversas áreas de atuação. Este trabalho teve por objetivo realizar parte da computação existente no modelo usando GPU. Os resultados obtidos foram validados, onde foi encontrada uma redução do tempo de execução em casos específicos. A implementação foi realizada com sucesso, permitindo o uso de GPUs na execução do modelo Eta.

PALAVRAS-CHAVE: CUDA, Modelo ETA,

Processamento Paralelo

INTEGRATION OF CUDA TECHNOLOGY TO THE ETA FORECAST MODEL

ABSTRACT: Numerical weather prediction models need a large amount of data and, due to the complexity of the operations executed, it becomes slow to obtain a result. Eta is a numerical weather prediction model used in Brazil by the CPTEC/INPE which assists on decision-making in several areas of expertise. The objective of this work was to perform part of the matrix operations of the model using the GPU. The results obtained were validated, where a reduction of the execution time was found in specific cases. The implementation was successful, allowing the use of GPUs in the execution of Eta Model.

KEYWORDS: CUDA, ETA Model, Parallel Processing

1 | INTRODUÇÃO

Modelos de previsão numérica do tempo (PNT) são software que fazem uso de modelos matemáticos com dados da atmosfera e dos oceanos visando prever a meteorologia de uma região por um determinado período de tempo (Lynch, 2008). Os PNT fazem uso de sistemas

de equações diferenciais com base em conceitos físicos, químicos e movimentos de fluídos, usando sistemas de coordenadas geográficas que dividem o planeta em uma grade de três dimensões. Cada célula desta grade calcula dados de vento, transferência de calor, radiação solar, umidade relativa e hidrologia da superfície. Essas informações interagem com as células vizinhas para prever as condições da área determinada no futuro.

O modelo Eta é um PNT de fenômenos atmosféricos regional, que prevê o tempo para uma região determinada, voltado para o uso de pesquisa e decisões operacionais. Este modelo se baseia na utilização de coordenadas verticais, que permanecem aproximadamente horizontais em áreas montanhosas, tornando-o adequado para o estudo de regiões topográficas íngremes, como, por exemplo, a cordilheira dos Andes (Chou et al. 2014).

Devido à quantidade de informações e a complexidade dos cálculos realizados pelo modelo Eta, a sua execução se torna custosa e relativamente demorada até a obtenção do resultado final da previsão para a área proposta.

Uma alternativa atrativa na busca de uma solução que reduza o tempo de processamento do modelo Eta é o uso de GPUs, em virtude de sua organização e pela tendência atual das aplicações de alto desempenho. CUDA (NVIDIA A, 2017) foi escolhida para realizar a implementação de uma solução com dois níveis de paralelismo, o primeiro, já implementado no modelo, dividindo a área a ser processada entre múltiplos computadores com MPI, e o segundo utilizando GPUs para efetuar cálculos.

2 | MODELO ETA

O modelo Eta é descendente do modelo de previsão Hydrometeorological Institute and Belgrade University (HIBU) e seu desenvolvimento começou a ser realizado na década de 70 por Mesinger e Janjic (INPE, 1995). Com o passar do tempo o código fonte passou por diversas alterações, como, por exemplo, o uso do estilo Arakawa para advecção horizontal em 1984, a reescrita do uso da coordenada Eta em 1988, entre diversos upgrades realizados. Esse software, atualmente, é utilizado por diversos países (INPE, 1995). No Brasil, o Eta é utilizado no Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) do Instituto Nacional de Pesquisa Espaciais (INPE). Ele é utilizado operacionalmente desde 1997 para previsões meteorológicas, e desde 2002 para previsões climáticas, abrangendo a América do Sul (Chou et al., 2014).

O modelo é alimentado com os dados referentes à área a ser coberta e à data da previsão. Os dados de condições iniciais para o modelo são obtidos por meio de análise estatística, na qual a estimativa inicial é ajustada de acordo com as observações do horário da análise (Chou, 2016). Ao final do processamento, é obtido o resultado da previsão para o período de tempo delimitado de, no mínimo, seis (6) horas, ou o número de horas desejada pelo usuário. Quanto maior o número de horas a serem

previstas, menor será a precisão da resposta do modelo.

O modelo Eta é atualmente escrito em Fortran 90, com alguns trechos de bibliotecas escritos em Fortran 77. A paralelização do modelo é realizada com a tecnologia Message Passing Interface (MPI) (Pacheco, 1997), especificamente a implementação MPICH, que busca alto desempenho.

Os processos criados para a execução do modelo são definidos como tarefas de previsão ou servidores de entrada e saída (I/O) sendo que a quantidade de cada é ajustada pelo usuário. Os servidores de I/O são responsáveis por armazenar os resultados obtidos pela execução do modelo, enquanto as tarefas de previsão pelo cômputo do modelo.

A área a ser processada pelo Eta é mapeada por uma matriz, com a quantidade de elementos no eixo vertical e horizontal definida nos parâmetros do experimento sendo realizado. A matriz é sobreposta no mapa terrestre, tendo como ponto central a latitude e longitude informadas. O tamanho de cada elemento da matriz é ajustado pela resolução do modelo, em quilômetros. A área total a ser processada é dividida em subáreas, nas quais cada tarefa de previsão é responsável por uma.

Após cada fase de cálculos, os dados são enviados para o servidor de I/O responsável por gravar as informações em disco enquanto as tarefas de previsão prosseguem para a iteração seguinte. Caso exista mais de um servidor de I/O, na próxima iteração será utilizado um diferente. Se for definido zero (nenhum) servidor de I/O, o armazenamento será efetuado pelas próprias tarefas de previsão, onde elas gravam dados em um mesmo arquivo.

Para que o processo de execução do modelo Eta seja melhor visualizado e compreendido foi montado o diagrama apresentado na Figura 1.

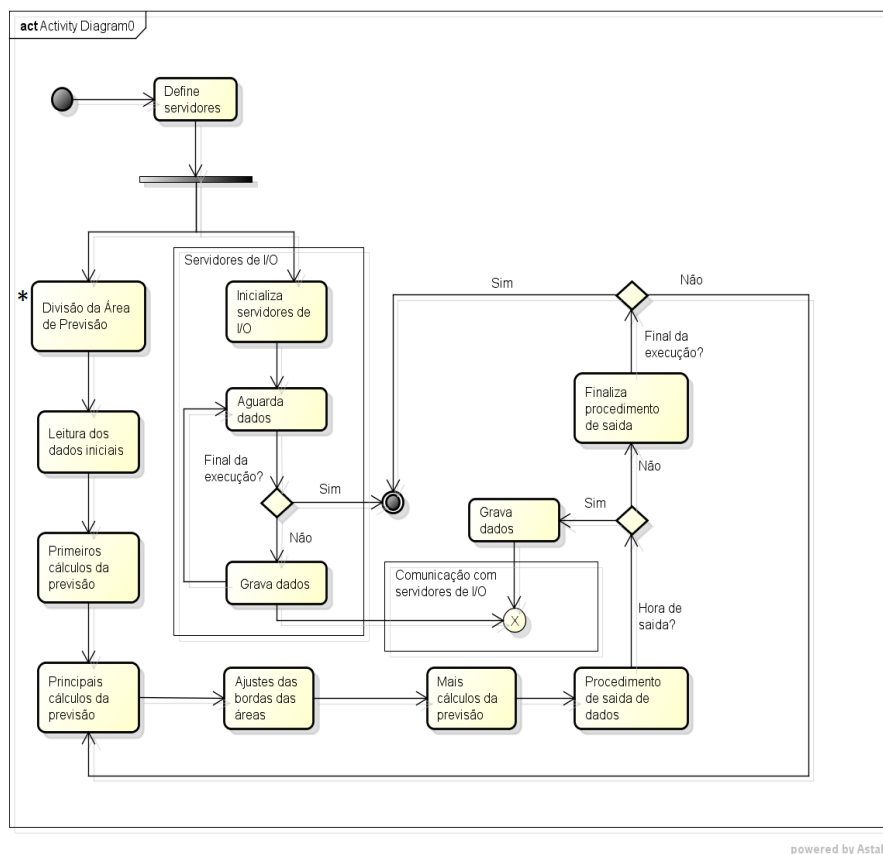


Figura 1. Diagrama de execução do modelo Eta.

Ao analisar o diagrama da Figura 1, pode-se visualizar o processo de divisão da área de previsão, desde a definição de sua área inicial, a inicialização dos servidores de I/O, os cálculos realizados pelo modelo e a comunicação da área a ser computada em cada processo. A Figura 2 complementa a Figura 1, onde pode-se visualizar outras etapas do processamento executado pelo modelo Eta no processo de “Divisão da Área de Previsão”, marcado por um * na Figura 1.

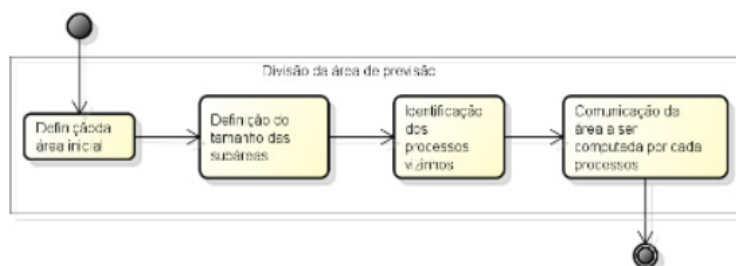


Figura 2. Diagrama parcial do Eta representando a divisão da área de previsão.

O processo de execução do Eta, apresentado nas Figuras 1 e 2, começa com dois fluxos distintos. Um dos fluxos realiza o controle dos servidores de I/O e tem como função principal salvar as informações geradas dos cálculos. Já o outro realiza a divisão da área a ser computada entre os processos de computação (Figura 2). Após a divisão, os processos realizam os cálculos da previsão e enviam os resultados para

os servidores de I/O.

3 | TRABALHOS RELACIONADOS

Foi realizada uma busca na literatura com o objetivo de encontrar trabalhos relacionados ao tema. Existem diversos modelos de simulação e previsão semelhantes ao Eta. Um exemplo é o *Ocean-Land-Atmosphere Model* (OLAM) (Marshall et al., 1997). Além da semelhança pela função de ser um PNT, outra similaridade com o Eta é o uso da tecnologia MPI em sua implementação, assim como, a utilização de vários níveis de processamento (Schepke; Maillard, 2012), onde foram utilizados CUDA e OpenMP.

O trabalho apresentado por Asunción et al. (2012) usa uma abordagem semelhante ao OLAM, onde aplica-se dois níveis de paralelismo, numa solução híbrida. São utilizados MPI e CUDA com o objetivo de melhorar o método de Roe (Asunción et al., 2012) na implementação de um sistema de águas rasas.

Na literatura é possível ainda encontrar diversos trabalhos que buscam explorar o paralelismo usando vários níveis. No trabalho de Dias et al. (2012) são citadas diversas aplicações que integram OpenMP com MPI, OpenMP com CUDA e MPI com CUDA.

Alguns exemplos de trabalhos, das mais diversas áreas, que exploram técnicas de paralelismo combinando MPI e CUDA são:

- Multiplicação de matrizes esparsas para o pré-condicionamento do método dos gradientes conjugados (Oyarzun et al., 2014).
- Implementação e otimização usando paralelismo de equações esparsas e mínimo quadrado (Huang et al. 2012).
- Implementação de um método eficiente para o cálculo de coeficiente de correlação de Pearson em matrizes (Kijispongse et al., 2011).
- Implementação de algoritmos de recuperação de senha (Apostal et al. 2012).
- Análise da performance de uma implementação do LU benchmark (Pennycook et al. 2011).
- Combinação de OpenMPI e StarSs (Perez et al., 2008) para a criação da tecnologia OmpSs (Bueno et al., 2012).
- Algoritmo de mecânica de fluídos explorando a heterogeneidade de hardware com as tecnologias OpenMP, MPI e OpenACC (Huismann et al., 2015).
- Ambientes computacionais utilizando vários níveis de paralelismo combinando CPU e GPU são apresentados nos trabalhos de Kim et al. (2017) e Mamalis et al. (2016).

Os resultados obtidos pelos trabalhos de Oyarzun et al. (2014), Huang et al. (2012), Kijispongse et al. (2011), Huismann et al. (2015), Schepke e Maillard (2012), Apostal et al. (2012), Dias et al. (2012), Pennycook et al. (2011) e Asunción et al.

(2012) são semelhantes. Após as implementações fazendo uso da GPU foram obtidas reduções nos tempos de processamento em relação aos software que não fazem uso da GPU.

Na Tabela 1 é apresentada uma sumarização dos trabalhos relacionados citados, sendo elas o ano de publicação, a tecnologia utilizada para implementação, os níveis de paralelismo e a linguagem de programação.

Trabalhos	Ano	Tecnologias envolvidas	Linguagem
(Pennycook et al. 2011)	2011	MPI, CUDA	Fortran 77
(Asunción et al. 2012)	2011	MPI, CUDA	C++
(Kijisipongse et al. 2011)	2011	MPI, CUDA	C
(Huang et al. 2012)	2012	MPI, CUDA	C
(Bueno et al. 2012)	2012	OpenMPI, StartSs	C, Fortran
(Schepke and Maillard 2012)	2012	OpenMP, CUDA, MPI	Fortran 90
(Apostal et al. 2012)	2012	MPI, CUDA	C
(Dias et al. 2012)	2012	MPI, CUDA	C, Fortran
(Oyarzun et al. 2014)	2014	MPI, CUDA	C
(Huismann et al. 2015)	2015	OpenMP, MPI, OpenACC	Fortran
(Mamalis et al 2016)	2016	OpenMP, CUDA	C
(Kim et Al. 2017)	2017	MPI, CUDA	Fortran

Tabela 1. Tabela sumarizadora de trabalhos, tecnologias e conclusões em relação ao desempenho da aplicação

Na Tabela 1 é possível verificar que nos últimos sete anos as tecnologias como MPI e CUDA são utilizadas para explorar o paralelismo nas mais diversas áreas, sendo implementados dois ou mais níveis de paralelismo. Pode-se perceber que há um forte movimento em busca da exploração de GPUs (com CUDA) nos modelos citados.

4 | MODELO PROPOSTO

Com o objetivo de criar um modelo de pesquisa que busca explorar GPUs no Eta foram realizadas análises no código do mesmo, buscando pontos onde o uso de GPUs poderia ser benéfico. Devido à grande quantidade de informações com que o Eta trabalha, foi possível encontrar nos códigos trechos com uma complexidade chegando até $O(n^4)$.

A fim de manter a implementação de MPI já presente no modelo Eta, foi elaborado um modelo de paralelismo do código que tem por objetivo delegar parte da computação realizada pelas tarefas MPI para a GPU (Figura 3).

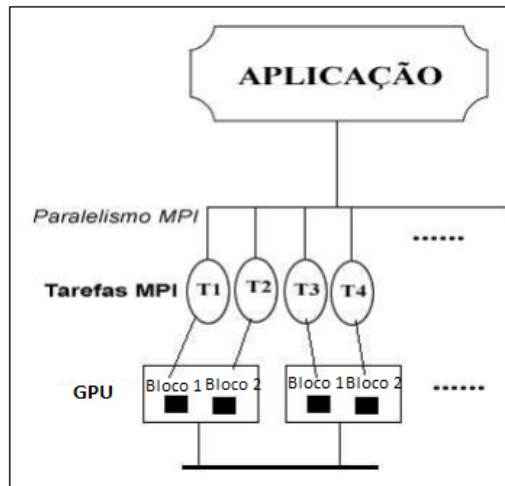


Figura 3. Modelo de solução com múltiplas tarefas MPI processando em diversas GPUs.

Com a utilização desse modelo de processamento (Figura 3), o Eta agora é um PNT que possui dois níveis de paralelismo: o primeiro com MPI, dividindo a área a ser computada entre diferentes processos; o segundo com CUDA, realizando parte dos cálculos de complexidade elevada.

5 | IMPLEMENTAÇÃO

A fim de agregar GPUs ao processamento do modelo Eta, utilizou-se a tecnologia CUDA, em virtude de ser intensamente citada (Tabela 1). Com essa adição, além de ampliar o hardware utilizado no processamento, será buscada uma redução no tempo de processamento.

Dos pontos de paralelismo identificados na criação do modelo, foram escolhidos três para a implementação com CUDA. Eles realizam multiplicação entre valores de três matrizes, tarefa favorável à execução pela GPU.

Na implementação de aplicações com uso de GPUs, deve-se escrever códigos para execução na GPUs. Pela nomenclatura usada pela Nvidia, esses blocos de código são denominados “kernel”. Na GPU o processamento ocorre em threads, que são agrupadas em blocos. O agrupamento de blocos de threads, com um determinado kernel, é chamado de grid.

Durante a execução, deve-se definir o número de blocos e a quantidade de threads que cada bloco vai computar. Dentro de cada bloco, cada thread possui um identificador único (ID) para cada dimensão, sendo eles para uso com a linguagem Fortran `threadIdx%x`, `threadIdx%y` e `threadIdx%z`, para as dimensões X, Y e Z, respectivamente.

Os índices da matriz são calculados a partir da ID de cada thread. As threads que possuem ID menor que o início ou maior que o final da matriz são finalizadas, restando uma quantidade igual ao número de elementos da matriz.

O modelo Eta é composto por, aproximadamente, 70.000 linhas, divididas em

114 códigos fontes. Destes códigos, identificou-se que os que consumiam mais tempo de execução foram as sub-rotinas VTADV, com aproximadamente 16% do tempo de execução, e o DIVHOA, com o tempo de aproximadamente 10%. Após uma análise, escolheu-se trabalhar a implementação no código fonte VTADV, cuja versão original possui 934 linhas de código e, com a adição da exploração da GPU, passou para 1127 linhas, pois este realiza mais operações matriciais, explorando assim o ponto forte do processamento em GPU.

Foram implementados três *kernels* dentro da sub-rotina VTADV, onde os principais cálculos de previsão do tempo do modelo Eta são realizados. Cada *kernel* possui um código cujo resultado é igual ao executado sem o uso de GPU. Essa execução é reproduzida para cada processo MPI criado, podendo ser computadas diversas partes da matriz ao mesmo tempo na GPU.

Os kernels criados no modelo Eta foram nomeados A, B e C. O kernel A é responsável por realizar a multiplicação de duas matrizes de três dimensões elemento por elemento e a multiplicação do valor resultado com um vetor de uma dimensão com N elementos, sendo sua complexidade $O(m*n*p)$. Nesses casos, “m” se refere a quantidade de linhas da matriz, “n” a quantidade de colunas e “p” o número de elementos localizados na altura da matriz. O kernel B, realiza a operação de multiplicação de uma matriz de três dimensões com um vetor. O resultado dessa operação é multiplicado com uma matriz de três dimensões. Após essa operação é realizada a soma de três matrizes. A complexidade destas operações é $O(m*n*p)$. Já o kernel C realiza uma multiplicação de duas matrizes, índice a índice, simplesmente, com complexidade $O(n*m*p)$.

Durante uma iteração da execução do modelo um dos kernel (A) é executado duas vezes em dois pontos diferentes do processamento do VTADV. Os kernels B e C são executados apenas uma vez. Para cada execução dos kernels é copiada da memória do computador para a GPU a área a ser processada e as variáveis, vetores ou matrizes de cada cálculo. Após o processamento das informações, o resultado é copiado do device (GPU) e atribuído para a matriz de resultado da implementação clássica e o fluxo de execução normal é retomado.

6 | EXPERIMENTOS REALIZADOS

Com o objetivo de validar as modificações realizadas no modelo Eta e verificar o desempenho obtido foram realizados testes com três tamanhos de áreas diferentes, mantendo a resolução de 10 km para cada uma das áreas (Figura 4).

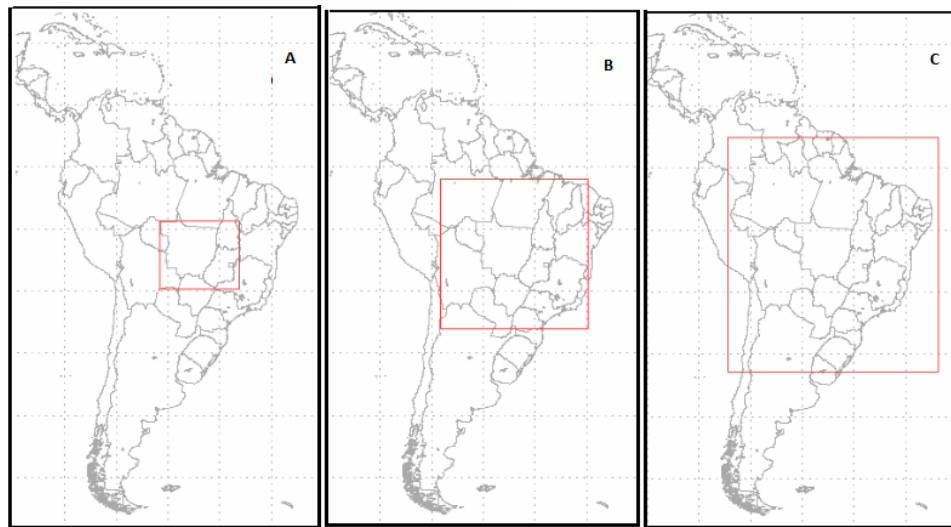


Figura 4. Visualização das áreas processadas.

Na Figura 4 pode-se visualizar as três áreas de testes onde a implementação foi validada. Foram denominadas áreas “pequena”, “média” e “grande”, Figuras 4A, 4B e 4C respectivamente. A área “pequena” possui 101x159 elementos, a área “média” 181x349 elementos e a área “grande” 251x581 elementos. Para todas as áreas foram realizados testes com três diferentes quantidades de horas de previsão, sendo estas seis (6), 12 e 24 horas.

Para cada combinação de horas de previsão e área a ser processada foram utilizados diferentes números de processos executados em cada computador, sendo eles com, 40, 60 e 80 processos.

Foram realizadas 20 repetições de cada combinação de área processada, horas de previsão e números de processos, sendo retirados o maior e o menor valor de tempo de processamento. Então, é calculada a média dos demais valores, medida a ser considerada na avaliação do desempenho dos testes.

O ambiente computacional de testes fez uso de 20 computadores com CPU Intel Core i7-3770 3,40 GHz contendo 4 núcleos físicos e 4 lógicos; 8 Gb de RAM; OS Ubuntu 16.04 LTS 64 bits; GPU GeForce GT 630 possuindo 384 CUDA cores, 2 Gb de memória, interface 64 bit-DDR3, com 14.4 Gb/s de largura de banda (NVIDIA B. 2017). Este ambiente se localiza na sala 9 do Laboratório Central de Informática (LCI) da Universidade de Passo Fundo. Além destas máquinas, foi utilizado um 21º computador como servidor de I/O, CPU Intel Core i7 920 2,66 GHz, 4 cores e 8 *threads*; 8 Gb de RAM; OS Ubuntu 16.04 LTS 64 bits (NVIDIA C, 2017). A rede de comunicação é de tecnologia Gigabit Ethernet.

As GPUs utilizadas possuem limite de 1024 *threads* por bloco, dimensões máximas de 1024 e 1024, 64 para x, y e z, respectivamente, e *wrap size* de 32, significando que independentemente do tamanho de bloco, o número de *threads* sempre será múltiplo de 32 (NVIDIA B. 2017).

Para a compilação do modelo Eta fazendo uso de CUDA foi utilizado o compilador

PGI edição comunitária, na versão 16.5. Esta edição do compilador é gratuita no período de um ano (PGI. 2017).

7 | RESULTADOS E ANÁLISE

Na realização dos experimentos foi utilizado um laboratório computacional de ensino. Este ambiente foi ocupado por aproximadamente 5 dias (123 horas) em diferentes momentos, para a realização dos experimentos.

As informações dos resultados podem ser visualizadas nas Tabelas 2, 3 e 4, onde são apresentados para as áreas grande, média e pequena, respectivamente. As tabelas são estruturadas pelo número de processos em relação às horas de previsão, com informações se foi ou não utilizado CUDA. São apresentados os resultados da média das execuções junto do desvio padrão em parênteses e o resultado do Teste-T (Valor-p), que representa se houve ou não uma diferença significativa (resultado abaixo de 0,05).

		Horas de previsão					
		6		12		24	
P	Cuda	Média (DP)	Valor-p	Média (DP)	Valor-p	Média (DP)	Valor-p
80	Não	299,20 (6,22)	0,029	582,87 (3,74) ⁺	0,001	1170,16 (8,58)	0,065
80	Sim	303,32 (5,88)		588,04 (5,39) ⁺		1165,36 (7,36)	
60	Não	302,69 (3,79)	0,117	577,09 (4,05)	0,031 [*]	1118,55 (6,89)	0,014
60	Sim	300,54 (4,63)		573,35 (6,24)		1111,89 (9,26)	
40	Não	275,36 (4,62)	0,006	544,04 (5,38)	0,640	985,32 (5,40)	0,000
40	Sim	271,46 (3,89)		543,23 (5,51)		998,59 (5,99)	

Tabela 2. Resultados dos tempos de Execução da área Grande.

* valor desconsiderado para comparações entre médias.

+ valores a uma distribuição não normal

Na Tabela 2 são mostrados os resultados dos experimentos para a área grande onde é possível notar que houve variações significativas com relação ao uso de CUDA, para os casos de 60 processos com 12 e 24 horas de previsão e com 40 processos com 6 horas de previsão. Os casos utilizando 80 processos com seis horas, e 40 processos com 24 horas obtiveram resultados significativos, porém estes aumentam o tempo de execução total do modelo fazendo uso da tecnologia CUDA.

Na Tabela 3 é possível observar que os resultados com uso de CUDA onde há redução do tempo de execução não apresentaram diferenças significativas. Onde ocorreram diferenças significativas nesta tabela houve piora no desempenho do quando realizada a execução com a integração de CUDA.

		Horas de previsão					
		6		12		24	
P	Cuda	Média (DP)	Valor-p	Média (DP)	Valor-p	Média (DP)	Valor-p
80	Não	183,58 (3,67)	0,001	357,67 (2,69)	0,003	700,14 (4,46)	0,033
80	Sim	187,24 (2,86)		361,06 (4,06)		704,06 (6,52)	
60	Não	172,32 (5,12) ⁺	0,000	332,19 (3,47)	0,270	656,24 (7,19)	0,603
60	Sim	178,31 (2,80)		330,77 (4,51)		655,18 (5,49)	
40	Não	148,06 (3,16)	0,001	285,24 (4,21)	0,188	545,44 (6,79)	0,000*
40	Sim	152,63 (4,69)		283,59 (3,52)		571,61 (3,97)	

Tabela 3. Resultados dos tempos de Execução da área Média.

* valor desconsiderado para comparações entre médias

+ valores a uma distribuição não normal

Para a área pequena, os resultados podem ser visualizados na Tabela 4 onde apresentam uma diferença significativa em relação ao uso de CUDA nos casos de 24 horas de previsão com 80 processos e 12 horas de previsão com 60 processos. Outros resultados significativos mostram uma piora do desempenho (tempo de execução) do modelo com a implementação que faz uso de CUDA, que podem ser observados nos casos de 60 processos com seis horas, 40 processos com seis e 24 horas de previsão.

		Horas de previsão					
		6		12		24	
P	Cuda	Média (DP)	Valor-p	Média (DP)	Valor-p	Média (DP)	Valor-p
80	Não	102,41 (3,33)	0,367	191,67 (6,55)	0,140	388,51 (3,67)	0,010
80	Sim	103,26 (2,47)		194,65 (5,93)		385,47 (3,44)	
60	Não	88,52 (1,87)	0,032*	169,56 (4,83)	0,018	334,22 (3,42)	0,556
60	Sim	90,58 (3,72)		166,31 (3,28)		333,46 (4,57)	
40	Não	73,76 (2,74)	0,001	144,49 (5,46)	0,406	279,31 (5,94)	0,000
40	Sim	77,39 (3,64)		143,15 (4,59)		289,22 (4,21)	

Tabela 4. Resultados dos tempos de Execução da área Pequena.

* valor desconsiderado para comparações entre médias

Os resultados dos tempos de execução podem ser visualizados na Figura 5, onde cada faceta do gráfico é uma combinação do tamanho da área e o número de horas de previsão (6h, 12h, 24h). O eixo Y apresenta a quantidade de tempo em segundos da média das execuções e o eixo X apresenta o número de processos. A cor indica se a tecnologia CUDA estava ou não sendo utilizada na execução do modelo.

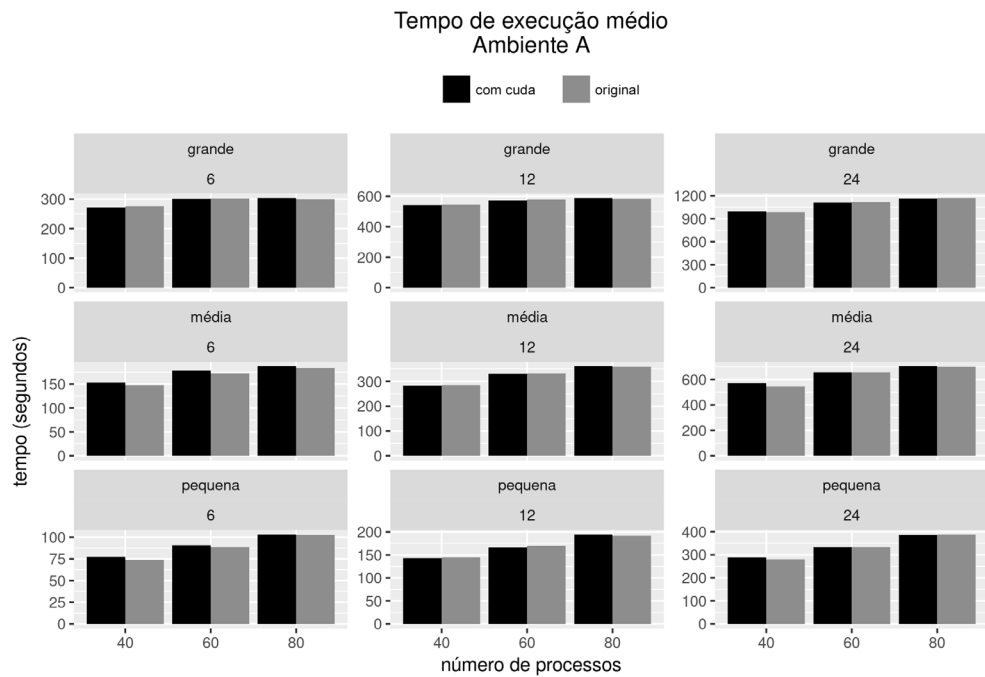


Figura 5. Gráfico de facetas representado a média dos resultados das execuções.

A Figura 5 apresenta os tempos de execução do modelo Eta com CUDA e sem as modificações. A integração do uso de GPU foi realizada com sucesso, obtendo uma melhora de desempenho em alguns casos. A utilização do modelo Eta com CUDA apresentou melhores resultados com um número menor de *threads* por computador, e em execuções para um maior período de tempo. Destaca-se que em todas as execuções realizadas, a qualidade dos resultados obtidos com exploração de GPUs que foram idênticos a execução do modelo originalmente estudado, ou seja, os resultados são os mesmos.

Um resultado inesperado obtido é que o aumento do número de processos do modelo frequentemente implica o aumento do tempo de execução do modelo, em outras palavras, o uso de um número maior de processos é ineficiente, sendo o *overhead* dos processos maior que o ganho da paralelização. Um fator que pode levar a esses resultados é a característica do modelo Eta, uma vez que o mesmo possui um forte sincronismo na sua execução (Figura 1). Esses resultados reforçam a exploração de hardware com maior capacidade de processamento, como são as GPUs, porém em menor número.

A evolução do tempo de execução foi linear em relação ao aumento do número de horas a serem processadas, o que era esperado, porém, o mesmo não pode ser dito do aumento do tempo de execução em relação ao crescimento da área. A diferença da área pequena para a área média é de cerca de quatro vezes e a diferença da área média para a grande é de cerca de duas vezes, entretanto o aumento do tempo de execução não seguiu estes fatores. Tanto da área pequena para média, quanto da área média para grande, o crescimento do tempo de execução foi de menos de duas vezes.

A tecnologia CUDA no modelo Eta foi integrada com sucesso e seus testes demonstraram que pode ser útil, sendo que em alguns casos de testes resultaram em diferenças significativas de tempo de execução, tanto para mais quanto para menos. Estes resultados abrem um gama de possibilidades para novos trabalhos visando melhorar o desempenho do modelo fazendo uso dessa nova possibilidade de paralelismo presente no modelo Eta.

8 | CONCLUSÃO

O modelo Eta apresentou-se complexo no seu entendimento e nas modificações realizadas em seu código. A integração da tecnologia CUDA, visando delegar parte do processamento do modelo para a GPU, se deu através da escrita de *kernels*, junto ao código com maior percentual de tempo de processamento, em pontos de maior complexidade computacional. O resultado obtido das execuções apresenta variações significativas com a integração de CUDA no ambiente, por vezes com melhor desempenho, mas sempre mantendo a exatidão do resultado final.

O uso da tecnologia CUDA no PNT Eta mostrou-se funcional, vantajosa em alguns casos e uma ferramenta adequada, devido seu modelo de processamento de dados, para ser incorporada em mais partes do PNT Eta. Mesmo num modelo extenso e complexo como o Eta, conclui-se que o uso de GPUs é viável e pode ser vantajoso em relação a desempenho.

Como trabalhos futuros, encontra-se em desenvolvimento uma implementação do modelo Eta com integração de GPUs modificando todo o código fonte VTADV par ser executado na GPU. Esse código ocupa aproximadamente 16% do tempo total de processamento do modelo, dessa forma, possui potencial para produzir melhores resultados no desempenho do modelo, reduzindo o tempo de execução. Além disso, outras tecnologias de exploração de GPUs, como OpenCL podem ser exploradas e agregadas ao modelo Eta.

REFERÊNCIAS

APOSTAL, D.; FOERSTER, K.; CHATTERJEE, A.; and DESELL, T. **Password recovery using mpi and cuda**. In: HIGH PERFORMANCE COMPUTING (HiPC), 9edition, Anais ... Pune: IEEE Press, 2012. p. 1-9.

ASUNCIÓN, M.; MANTAS, J.; CASTRO, M.; FERNANDEZ-NIETO, E. **An MPI-CUDA implementation of an improved roe method for two-layer shallow water systems**. Journal of Parallel and Distributed Computing, Elsevier, v.72, n.1, p.1065-1074. 2012.

BUENO, J.; PLANAS, J.; DURAN, A.; BADIA, R.; MARTORELL, X.; AYGUADÉ E.; LABARTA, J. **Productive programming of gpu clusters with ompSs**. In: PARALLEL & DISTRIBUTED PROCESSING SYMPOSIUM (IPDPS), 26th, Anais ... Shanghai: IEEE International. 2012. p. 557-568.

CHOU, S. C.; LYRA, A.; MOURÃO, C.; DEREZYNKI, C.; PILOTTO, I.; GOMES, J.; BUSTAMANTE, J.; TAVARES, P.; SILVA, A.; RODRIGUES, D.; CAMPOS, D.; CHAGAS, D.; SUEIRO, G.; SIQUEIRA,

- G.; NOBRE, P.; MARENGO, J. **Evaluation of the Eta Simulations Nested in Three Global Climate Models**. American Journal of Climate Change, v.3, n.5, p.438-454. 2014.
- CHOU, S. C. **Modelo Regional Eta**. Disponível em: <<http://climanalise.cptec.inpe.br/~rclimanl/boletim/cliesp10a/27.html>> Acesso em: Outubro 2016.
- DIAS, J.; MUNOZ-CARO, C.; NIÑO, A. **A Survey of Parallel Programming Models and Tools in the Multi and Many-Core Era**. IEEE Transactions on Parallel and Distributed Systems, v.23, n.8, p.1369-1386. 2012.
- HUANG, H.; WANG, L.; LEE, E.; CHEN, P. (2012). **An mpi-cuda implementation and op-timization for parallel sparse equations and least squares (lsqr)**. Procedia Computer Science, v.9, p.76-85. 2012.
- HUISMANN, I.; STILLER, J.; FRÖHLICH, J. **Two-level parallelization of a fluid mechanics algorithm exploiting hardware heterogeneity**. Computers & Fluids, v.117, p.114-124. 2015.
- INPE. **Model I Eta Model**. 1995. Disponível em: <<http://etamodel.cptec.inpe.br/history/>>. Acesso em: Outubro. 2016.
- KIJSIPONGSE, E.; U-RUEKOLAN, S.; NGAMPHIW, C.; TONGSIMA, S. **Efficient large pearson correlation matrix computing using hybrid MPI/CUDA**. In: 2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE). Anais ... Nakhon Pathom: IEEE Press, p. 237-241. 2011.
- KIM, B.; JUNG, J.; MIN, H.; HEO, J.; JUNG, H. **Performance Evaluations of Multiple GPUs based on MPI Environments**. In: Proceedings of the International Conference on Research in Adaptive and Convergent Systems. Anais ... Krakow: ACM Press, p.303-304. 2017.
- LYNCH, P. **The origins of computer weather prediction and climate modeling**. Journal of Computational Physics. v.227, n.7, p.3431-3444. 2008.
- MACHADO, V. L. **Análise do impacto da utilização da previsão do tempo corrigida pelo método Model Output Calibration em modelos de doenças da maçã**. Dissertação (Mestrado). Programa de Pós-graduação em Computação Aplicada (PPGCA). Universidade de Passo Fundo (UPF). 2017.
- MAMALIS, B.; PERLITIS, M. **A Hybrid Parallelization Scheme for Standard Simplex Method based on CPU/GPU Collaboration**. In: Proceedings of the 20th Pan-Hellenic Conference on Informatics. Anais ... Patras: ACM Press, p1-6. 2016.
- MARSHALL, J.; ADCROFT, A.; HILL, C.; PERELMAN, L.; HEISEY, C. **A finite-volume incompressible navier-stokes model for studies of ocean on parallel computers**. Geophysical Journal. v.139, n.1, p.123-130. 1999.
- NCEI. **Global Forecast System (GFS)**. 2003. Disponível em: <<https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs>>. Acesso em: Março. 2017.
- NVIDIA A. **CUDA Zone**. Disponível em: <<https://developer.nvidia.com/cuda-zone>>. Acesso em: Setembro. 2017.
- NVIDIA B. **NVIDIA GeForce GT 630**. 2017. Disponível em: <<http://www.nvidia.com.br/object/geforce-gt-630-br.html#pdpContent=2>> Acesso em: Setembro. 2017.
- NVIDIA C. **Placa de vídeo GTX 770 com GPU Boost 2.0**. 2017. Disponível em: <<http://www.nvidia.com.br/object/geforce-gtx-770-br.html#pdpContent=2>> Acesso em: Setembro. 2017.
- OYARZUN, G.; BORRELL, R.; GOROBETS, A.; OLIVA, A. **Mpi-cuda sparse matrix-vector multiplication for the conjugate gradient method with an approximate in-verse preconditioner**. Computers & Fluids. v.92, p.244-252. 2014.

PACHECO, Peter S. **Parallel programming with MPI**. San Francisco: Morgan Kaufmann, 1997. 418p.

PENNYCOOK, S.; HAMMOND, S.; JARVIS, S. **Performance analysis of a hybrid MPI/CUDA implementation of the NASLU benchmark**. ACM SIGMETRICS Performance Evaluation Review. v.38, n.4, p.23-29. 2011.

PEREZ, J. M.; BADIA, R. M.; LABARTA, J. **A dependency aware task-based programming environment for multi-core architectures**. In: IEEE International Conference on Cluster Computing. Anais ... Tsukuba: IEEE Press. p. 142-151. 2008.

PGI. **PGI Compiler & Tools**. 2017. Disponível em: < <https://www.pgroup.com/> > Acesso em: Setembro 2017.

SCHEPKE, C; MAILLARD, N. **Exploring Multi-level Parallelism in Atmospheric Applications**. In: Symposium on Computer Systems, 13th. Anais ... Petropolis: WSCAD-SSC. p.41-48. 2012.

SOBRE O ORGANIZADOR

ERNANE ROSA MARTINS - Doutorado em andamento em Ciência da Informação com ênfase em Sistemas, Tecnologias e Gestão da Informação, na Universidade Fernando Pessoa, em Porto/Portugal. Mestre em Engenharia de Produção e Sistemas, possui Pós-Graduação em Tecnologia em Gestão da Informação, Graduação em Ciência da Computação e Graduação em Sistemas de Informação. Professor de Informática no Instituto Federal de Educação, Ciência e Tecnologia de Goiás - IFG (Câmpus Luziânia), ministrando disciplinas nas áreas de Engenharia de Software, Desenvolvimento de Sistemas, Linguagens de Programação, Banco de Dados e Gestão em Tecnologia da Informação. Pesquisador do Núcleo de Inovação, Tecnologia e Educação (NITE), certificado pelo IFG no CNPq.

Agência Brasileira do ISBN
ISBN 978-85-7247-339-2

