

Anna Victoria Cabrera Rondon Oikawa

Escalonamento Global Orientado à Carga de Trabalho para Aplicações Científicas

Florianópolis

2017

Anna Victoria Cabrera Rondon Oikawa

Escalonamento Global Orientado à Carga de Trabalho para Aplicações Científicas

Monografia submetida ao Programa de Graduação em Ciência da Computação para a obtenção do Grau de Bacharel.

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística
Ciência da Computação

Orientador: Márcio Bastos Castro
Coorientador: Laércio Lima Pilla

Florianópolis

2017


FOLHA DE APROVAÇÃO DE PROPOSTA DE TCC

Acadêmico(s)	Anna Victoria Cabrera Rondon Oikawa
Título do trabalho (subtítulo)	Escalonamento Global Orientado à Carga de Trabalho para Aplicações Científicas
Curso	Ciência da Computação/INE/UFSC
Área de Concentração	Sistemas de Computação

Instruções para preenchimento pelo **ORIENTADOR DO TRABALHO**:

- Para cada critério avaliado, assinale um X na coluna SIM apenas se considerado aprovado. Caso contrário, indique as alterações necessárias na coluna Observação.

Critérios	Aprovado				Observação
	Sim	Parcial	Não	Não se aplica	
1. O trabalho é adequado para um TCC no CCO/SIN (relevância / abrangência)?	✓				
2. O título do trabalho é adequado?	✓				
3. O tema de pesquisa está claramente descrito?	✓				
4. O problema/hipóteses de pesquisa do trabalho está claramente identificado?	✓				
5. A relevância da pesquisa é justificada?	✓				
6. Os objetivos descrevem completa e claramente o que se pretende alcançar neste trabalho?	✓				
7. É definido o método a ser adotado no trabalho? O método condiz com os objetivos e é adequado para um TCC?	✓				
8. Foi definido um cronograma coerente com o método definido (indicando todas as atividades) e com as datas das entregas (p.ex. Projeto I, II, Defesa)?	✓				
9. Foram identificados custos relativos à execução deste trabalho (se houver)? Haverá financiamento para estes custos?	✓				
10. Foram identificados todos os envolvidos neste trabalho?	✓				
11. As formas de comunicação foram definidas (ex: horários para orientação)?	✓				
12. Riscos potenciais que podem causar desvios do plano foram identificados?	✓				
13. Caso o TCC envolva a produção de um software ou outro tipo de produto e seja desenvolvido também como uma atividade realizada numa empresa ou laboratório, consta da proposta uma declaração (Anexo 3) de ciência e concordância com a entrega do código fonte e/ou documentação produzidos?				✓	

Avaliação	<input checked="" type="checkbox"/> Aprovado <input type="checkbox"/> Não Aprovado		
Professor Responsável	Márcio Bastos Castro	16/11/2017	
Orientador externo			

Resumo

Aplicações científicas desenvolvidas em grandes centros de pesquisa necessitam de alto poder computacional para que possam obter resultados precisos. Para que se possa explorar ao máximo todos os recursos computacionais e de memória disponíveis, faz-se o uso de plataformas de Computação de Alto Desempenho (*High Performance Computing* - HPC) em conjunto com interfaces de Computação Paralela. Uma aplicação HPC pode ser decomposta em tarefas paralelas, onde cada uma possui cargas computacionais diferentes de acordo com suas características individuais. Tais diferenças resultam em um desbalanceamento de carga entre os recursos computacionais, causando um impacto negativo no desempenho e escalabilidade da aplicação. Assim, para evitar o desperdício computacional e de energia, são aplicados algoritmos de escalonamento global para que essas tarefas em desequilíbrio sejam redistribuídas e o desempenho da aplicação tenha uma melhoria significativa.

O objetivo deste trabalho é propor a automatização da escolha de algoritmos de escalonamento global para aplicações científicas HPC, fazendo o uso de Aprendizagem de Máquina (*Machine Learning* - ML) para auxiliar na tomada de decisão do melhor algoritmo a ser escolhido. Essa decisão será feita por um meta-escalonador que será desenvolvido e leva em conta características de cada cenário particular. Além de não existir um algoritmo ideal para encontrar uma distribuição de tarefas ótima que funcione todas as situações, não há uma forma de identificar qual algoritmo é o mais indicado para uma dada aplicação executando em uma dada plataforma e em um dado instante do tempo. Portanto, é um problema não trivial que afeta aplicações científicas atualmente e, assim, o estudo e implementação da solução deste problema será o foco deste trabalho.

Para dar início à execução do projeto, será realizado um trabalho de pesquisa com a finalidade de caracterizar *escalonadores globais*, juntamente com a caracterização de *aplicações científicas para fins de escalonamento global*, e por fim, das *plataformas paralelas para fins de escalonamento global*. Após a obtenção e compreensão desses conhecimentos, o estado da arte que compõe o contexto deste trabalho terá formado a base necessária para a implementação do protótipo de um *meta-escalonador adaptativo*. Por fim, será necessário efetuar a validação e análise do protótipo desenvolvido, uma vez que este fará uso de um algoritmo de ML para auxiliar na tomada de decisão, e portanto, será preciso verificar se os resultados obtidos estão de acordo com aqueles esperados. Dessa forma, a coerência da escolha do melhor algoritmo para um determinado cenário – que é composto por características particulares – será usada como métrica de avaliação do meta-escalonador.

Palavras-chaves: HPC. CAD. balanceamento de carga. aprendizagem de máquina. escalonamento global.

Lista de abreviaturas e siglas

HPC	High Performance Computing
CAD	Computação de Alto Desempenho
ML	Machine Learning

Sumário

1	PROJETO	9
1.1	Introdução	9
1.2	Objetivos	12
1.3	Método de Pesquisa	13
2	PLANEJAMENTO	15
2.1	Cronograma	15
2.2	Recursos Humanos	16
2.3	Custos	17
2.4	Comunicação	17
2.5	Riscos	19
	REFERÊNCIAS	21

1 Projeto

Este capítulo apresenta o tema de pesquisa do Trabalho de Conclusão de Curso, o escopo no qual o problema em questão será tratado e a justificativa do projeto. Na Seção 1.1 é apresentado o contexto e a motivação para a realização do trabalho. Na Seção 1.2 são introduzidos os objetivos gerais e específicos, as restrições e premissas existentes, a lista de marcos e os critérios de aceite do projeto. Na Seção 1.3, são abordados os métodos de pesquisa envolvidos para alcançar a solução proposta.

1.1 Introdução

Um dos principais elementos que contribuem para o avanço da Ciência é o desempenho das aplicações utilizadas em grandes centros de pesquisa. Diversas áreas no contexto de Pesquisas Espaciais, por exemplo, necessitam de simulações numéricas complexas e de alto desempenho para que se possa alcançar resultados precisos e em grande escala (PILLA, 2017). Aplicações paralelas de dinâmica molecular, por exemplo, têm o balanceamento de carga como um aspecto crucial para um bom desempenho em máquinas paralelas de larga escala (BATHELE; KALE; S., 2009).

Assim, para satisfazer essa necessidade de alto poder computacional, plataformas de Computação de Alto Desempenho (*High Performance Computing* – HPC) são utilizadas em conjunto com linguagens e interfaces de computação paralela. Um dos elementos que levam a complexidade no escalonamento de aplicações é a dependência dessas tarefas, visto que é necessário estimar quando uma delas será iniciada e encerrada para que as tarefas dependentes a ela possam iniciar execução (RODAMILANS, 2009).

Atualmente, há um crescimento constante e significativo na quantidade de núcleos nos processadores, e conseqüentemente, as plataformas mais recentes acabam por conter centenas de milhares de núcleos. Mas para que se possa extrair todo o potencial desse aumento do número de núcleos, é preciso adaptar e otimizar as aplicações que farão uso deles. Por conta disso, o desenvolvimento de programas paralelos possui extrema importância para o uso das plataformas computacionais atuais visto que o paralelismo de processadores e plataformas tendem a aumentar (PILLA; MENESES, 2015).

Uma vez que os tipos de aplicações que executam em ambientes de larga escala possuem um objetivo complexo, diversos elementos estão envolvidos de acordo com as características únicas de cada aplicação e dos cenários que as mesmas estão sendo executadas. Assim, a solução para atingir esse objetivo é decompor a aplicação principal em tarefas menores, onde cada tarefa realiza um trabalho específico e diferente das outras tarefas, para

que depois, seja possível fazer a união desses resultados menores para compor o resultado final. Deve-se levar em consideração que nem sempre é possível prever a quantidade de trabalho que cada tarefa receberá antes da aplicação iniciar a sua execução. Uma irregularidade de carga poderá surgir das tarefas que realizam diferentes papéis na aplicação, além da possibilidade da carga mudar dinamicamente ao longo da execução (PILLA, 2014).

Em grande parte das situações, o *overhead* relacionado às diferenças de carga nos processadores é o fator dominante responsável por encerrar suas respectivas execuções em momentos diferentes. Alguns processadores podem permanecer ociosos enquanto outros estão sobrecarregados, e como cada tarefa possui um objetivo diferente, a consequência é que cada uma delas terá uma carga computacional diferente. Assim, pode-se dizer que uma aplicação é desbalanceada se um número significativo de nodos computacionais dependem que outros terminem algum tipo de trabalho, resultando em desperdício de recursos e de energia (RASHID; BANICESCU; CARINO, 2008; FREITAS; PILLA, 2017).

Aplicações científicas como simulações de partículas têm cargas de trabalho imprevisíveis, isto é, variam de acordo com o fluxo de computação. Para atingir um alto desempenho nessas classes de aplicação, é necessário que a carga de trabalho seja distribuída entre os processadores dinamicamente. Mas os requisitos na escolha de algoritmo variam de acordo com a natureza de cada contexto. Ou seja, requisitos para uma aplicação de simulação de partículas serão diferente daqueles escolhidos para aplicações como Refinamento Adaptativo de Malhas (*Adaptive Mesh Refinement* - AMR) (CHAUBE; CARINO; BANICESCU, 2007).

Selecionar um algoritmo de escalonamento eficiente entre os disponíveis atualmente para alcançar o balanceamento de carga no contexto de aplicações que são executadas em ambientes imprevisíveis é uma tarefa não trivial. O balanceamento de carga pode ser necessário em diferentes partes da aplicação e cada uma dessas partes pode precisar de outros algoritmos de escalonamento para atingir um desempenho ótimo. Uma maneira eficiente de escalar essas aplicações é realizando uma distribuição de cargas de trabalho, mas é possível que efeitos indesejados surjam com esse processo (RASHID; BANICESCU; CARINO, 2008; FREITAS; PILLA, 2017). Isso resulta em comportamentos irregulares observados dinamicamente, levando a um desbalanceamento de carga dos recursos computacionais e a um custo maior de comunicação. Dessa forma, o desempenho e escalabilidade da aplicação são comprometidos visto que processadores podem ficar ociosos, o que acarreta um desperdício computacional e de energia.

Os algoritmos de escalonamento global usados em balanceadores de carga gerenciam a distribuição das tarefas decompostas da aplicação entre os recursos disponíveis em um ambiente paralelo. O principal ponto é que encontrar uma distribuição de tarefas ótima é um problema NP-Difícil e por conta disso, existem algoritmos que são baseados em diferentes heurísticas para serem usados em diferentes aplicações científicas (LEUNG, 2004).

Assim, deve-se ter em mente que cada algoritmo de escalonamento global tem como foco diferentes objetivos e leva em consideração as características específicas da plataforma computacional alvo.

Para ilustrar esse fato, pode-se ter a situação em que um escalonador tem como prioridade reduzir o consumo de potência de uma plataforma, enquanto outro concentra seu foco em reduzir o tempo de execução das iterações da aplicação (PILLA, 2017). Logo, *não há um algoritmo que seja ideal para todas as situações*, uma vez que cada aplicação científica é executada sob diferentes condições, sejam elas o tipo da plataforma, o objetivo que se quer atingir, as características que compõem a aplicação, entre outros elementos que a faz particular. Além disso, mesmo que um algoritmo cumpra com o objetivo para o qual foi desenvolvido, pode ocorrer de outro aspecto ser impactado negativamente.

Seja uma aplicação que é caracterizada por possuir tarefas com alto dinamismo de carga e tem como objetivo *migrar uma maior quantidade de tarefas* para atingir um equilíbrio. O sobrecusto do processo de migração de tarefas pode acabar prejudicando o desempenho a ponto de que a redistribuição de trabalho não tenha um bom resultado, do ponto de vista geral da aplicação. É importante ressaltar que identificar qual algoritmo é o mais indicado para uma dada aplicação que será executada em uma dada plataforma e em um dado instante do tempo é um *problema não trivial que afeta aplicações científicas atualmente* (PILLA, 2017). Um algoritmo de escalonamento escolhido estaticamente pode realizar um bom trabalho no início da execução da aplicação, mas posteriormente tornar-se inapropriado. Uma entidade mais inteligente é necessária para selecionar dinamicamente algoritmos de escalonamento ao longo da execução (RASHID; BANICESCU; CARINO, 2008).

Assim, nota-se a necessidade de um meta-escalonador que realize uma análise individual para cada aplicação científica de forma adaptativa. Esse meta-escalonador levará em conta as diferentes características que compõem o contexto onde a aplicação está sendo executada, sejam elas o instante do tempo, aspectos da plataforma paralela da execução, ou características da própria aplicação. É esperado que tratando do problema do escalonamento global no contexto de aplicações científicas, estas terão um aumento significativo no desempenho e acabarão por produzir resultados mais precisos e em maior escala. Isso abre portas para novas áreas que poderão ser exploradas ou obter respostas mais próximas da realidade das aplicações que atualmente não podem explorar os recursos disponíveis que não estão sendo aproveitados. Claramente pode-se notar o grande impacto no avanço da Ciência que o resultado do projeto pode trazer, uma vez que após a implementação deste trabalho, poderá ser estudado e aperfeiçoado em trabalhos futuros.

1.2 Objetivos

O principal desafio é gerenciar a execução de uma aplicação científica para que se possa explorar ao máximo os recursos de uma plataforma paralela e então, atingir desempenho e escalabilidade (PILLA, 2014). Portanto, uma vez que o problema do escalonamento global foi identificado no contexto de aplicações científicas e seu impacto negativo no avanço da Ciência, este *Trabalho de Conclusão de Curso* tem como objetivo a implementação de um meta-escalonador adaptativo orientado à carga de trabalho das tarefas que compõem uma dada aplicação científica que executará em plataformas paralelas. O meta-escalonador fará uso de técnicas de *Aprendizagem de Máquina* uma vez que a tomada de decisão ocorrerá de forma adaptativa para cada cenário diferente em aplicações diferentes e por consequência, conjunto de características diferentes.

Além disso, para que os resultados obtidos tenham sucesso, é necessário compreender e definir os conceitos necessários envolvidos no escopo deste trabalho. Logo, têm-se como objetivos específicos:

1. Definição das características que afetam o desempenho das aplicações científicas nos aspectos de desbalanceamento de carga e comunicações custosas;
2. Definição de métricas relacionadas ao desempenho de núcleos de processamento e transferência de dados;
3. Definição de métricas para a compreensão e comparação de algoritmos de escalonamento global.

Por fim, o objetivo final deste projeto é a realizar uma análise dos resultados obtidos das execuções do meta-escalonador. Métricas serão definidas para verificar se, dado um cenário de execução composto por características individuais, a tomada de decisão do melhor algoritmo de escalonamento será feita de fato da melhor forma. Isto é, se o algoritmo escolhido satisfaz as necessidades de cada aplicação que o executará. Assim, pode-se obter comparações dos resultados obtidos do trabalho implementado com o estado da arte, verificando as melhorias significativas que essa nova abordagem poderá trazer.

Restrições:

- A caracterização de aplicações científicas para fins de escalonamento global deve ser definida antes da implementação do meta-escalonador;
- A caracterização de plataformas paralelas para fins de escalonamento global deve ser definida antes da implementação do meta-escalonador;

- A caracterização de escalonadores globais deve ser definida antes da implementação do meta-escalonador.

Premissas:

- O notebook para realização do trabalho estará disponível;
- O orientador e coorientador estarão disponíveis para reuniões periódicas;
- Disponibilidade de água;
- Disponibilidade de luz;
- Disponibilidade de Internet.

Marcos:

- Entrega do resumo em TCC I: 2ª semana de Junho/2018;
- Entrega do projeto em fase inicial: 2ª semana de Junho/2018;
- Entrega da primeira versão da monografia em TCC II: 2ª semana de Outubro/2018;
- Defesa da monografia: 1ª semana de Novembro/2018;
- Entrega da versão final da monografia em TCC II: 2ª semana de Novembro/2018.

Crítérios de aceite:

- Aprovação da banca avaliadora;
- Aprovação do orientador;
- Aprovação do coorientador;
- Conformidade da monografia com as normas definidas pela instituição;
- Prazos cumpridos.

1.3 Método de Pesquisa

Inicialmente, o trabalho possuirá caráter teórico, envolvendo o estudo do estado da arte e a caracterização de aplicações científicas, plataformas paralelas e escalonamento global, todos estes no contexto de aplicações científicas que envolvem balanceamento de carga.

Precedendo esses conceitos, será realizado um estudo de *aprendizagem de máquina, balanceamento de carga* e *Charm++* para que uma base com fortes fundamentos seja estabelecida, fornecendo o conhecimento e entendimento necessário para a realização do projeto.

O desenvolvimento prático do protótipo do meta-escalonador fará uso do sistema de programação paralela *Charm++*, tendo C++ como a principal linguagem de programação. No contexto de balanceamento de carga, a base para o desenvolvimento envolverá conceitos como o *Modelo de Programação de Objetos Migráveis*, o qual é suportado por sistemas de execução adaptativos. Assim, o uso de *Charm++* facilita a implementação pois segue um paradigma de programação paralela orientado a objetos, permitindo paralelismo de dados e de tarefas e que usa a troca de mensagens assíncronas para o desenvolvimento das aplicações (ILLINOIS, 2017).

Na etapa de encerramento, ocorrerá a validação e análise dos resultados obtidos de diferentes execuções do meta-escalonador implementado. A forma que essa etapa será cumprida envolve a realização de experimentos e testes do meta-escalonador em plataformas paralelas, para que assim, haja a possibilidade da análise dos resultados obtidos. Será levado em conta a decisão do algoritmo de acordo com o contexto onde a aplicação estará sendo executada, e então, conclusões poderão ser inferidas com base nas análises realizadas.

2 Planejamento

Este capítulo contém o planejamento do projeto onde cada plano de gerenciamento se encontra em diferentes seções. A Seção 2.1 apresenta o cronograma que deverá ser seguido ao longo da realização deste trabalho e as atividades planejadas. A Seção 2.2 apresenta os recursos humanos envolvidos. A Seção 2.3 apresenta os custos estimados para a execução do projeto. A Seção 2.4 apresenta o gerenciamento da comunicação entre as partes envolvidas. A Seção 2.5 apresenta os riscos identificados e suas respectivas estratégias.

2.1 Cronograma

As atividades previstas no projeto estão descritas abaixo:

- **A1: Estudo da fundamentação teórica.** Nesta etapa será realizada a revisão de artigos e materiais pertinentes à literatura que será utilizada.
- **A2: Revisão do estado da arte e prática.** Nesta etapa será tratada a revisão do estado da arte envolvido no contexto do trabalho, a fim de fortalecer a base do conhecimento necessária para a realização do mesmo.
- **A3: Elaboração da proposta.** Nesta etapa será apresentada a proposta de solução do problema em questão, bem como suas estratégias de implementação.
- **A4: Escrita do relatório do TCC I.** Nesta etapa será realizada a escrita do relatório do TCC I. A entrega deste documento está prevista para a segunda semana do mês de junho.
- **A5: Implementação da proposta.** Nesta etapa será realizada a implementação da proposta da solução apresentada.
- **A6: Realização de experimentos.** Nesta etapa será efetuada a fase de testes do projeto, envolvendo a análise dos resultados e comportamentos obtidos da execução do código desenvolvido.
- **A7: Escrita do rascunho do TCC II.** Nesta etapa será realizada a escrita do rascunho do TCC II. A entrega deste documento está prevista para a segunda semana do mês de outubro.
- **A8: Preparação da defesa pública.** Nesta etapa será realizada a preparação da apresentação oral do conteúdo deste trabalho para a defesa pública.

- **A9: Defesa pública.** Nesta etapa será realizada a defesa do projeto desenvolvido. Pretende-se realizar a defesa pública do trabalho na primeira semana do mês de novembro.
- **A10: Correções e entrega da versão final do TCC.** Nesta etapa serão realizadas as correções e os ajustes da monografia e a entrega final do documento. A entrega da versão final está prevista para a quarta semana do mês de novembro.

A Figura 1 apresenta o cronograma previsto para a realização das atividades descritas anteriormente, o que inclui atividades durante o primeiro e segundo semestres de 2018.

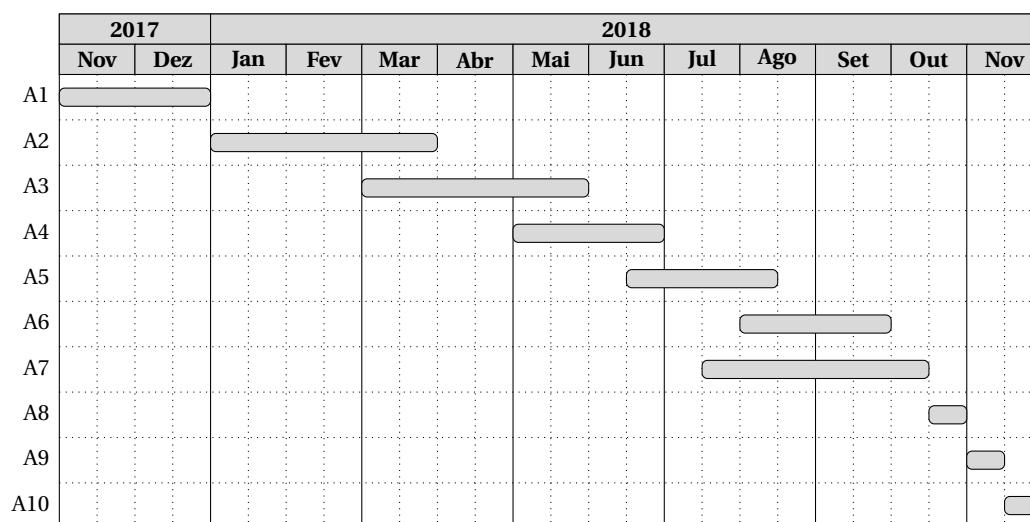


Figura 1 – Cronograma de atividades.

2.2 Recursos Humanos

Papel	Nome
Orientador	Márcio Bastos Castro
Coorientador	Laércio Lima Pilla
Coordenador	Renato Cislacghi
Membro da Banca I	
Membro da Banca II	
Autor	Anna Victoria Cabrera Rondon Oikawa

2.3 Custos

Estimativas para Recursos Humanos					
Nome	Data Início	Data Fim	Hora/Mês	Valor/Hora	Custo Total
Orientador	25/09/2017	01/12/2018	4	R\$ 71,00	R\$ 3976,00
Coorientador	25/09/2017	01/12/2018	4	R\$ 71,00	R\$ 3976,00
Coordenador	01/09/2017	01/12/2018	1	R\$ 102,00	R\$ 1530,00
Membro da Banca I	01/11/2018	01/12/2018	1	R\$ 60,00	R\$ 60,00
Membro da Banca II	01/11/2018	01/12/2018	1	R\$ 60,00	R\$ 60,00
Subtotal estimativas para recursos humanos					R\$ 9602,00

Estimativas para Recursos Não Humanos					
Descrição	Data Início	Data Fim	Quantidade	Valor Unitário	Custo Total
CDs para o código desenvolvido	11/18	11/18	2	R\$ 4,00	R\$ 8,00
Impressão para o relatório	06/18	06/18	3	R\$ 20,00	R\$ 60,00
Subtotal estimativas para recursos não humanos					R\$ 68,00

2.4 Comunicação

O que precisa ser comunicado	Reuniões periódicas com o Orientador e Coorientador
Emissor	Anna Victoria Cabrera Rondon Oikawa
Receptor	Márcio Bastos Castro, Laércio Lima Pilla
Comunicação	Reuniões periódicas com o Orientador e Coorientador para acompanhamento do projeto
Forma de comunicação	Pessoalmente
Frequência ou Quando	Quinzenalmente

O que precisa ser comunicado	Entrega da Proposta
Emissor	Anna Victoria Cabrera Rondon Oikawa
Receptor	Renato Cislighi
Comunicação	Entrega da proposta completa do TCC
Forma de comunicação	Sistema de TCC
Frequência ou Quando	Única vez

O que precisa ser comunicado	Entrega do Relatório em TCC I
Emissor	Anna Victoria Cabrera Rondon Oikawa
Receptor	Renato Cislighi
Comunicação	Entrega da primeira parte da monografia
Forma de comunicação	Sistema de TCC
Frequência ou Quando	Única vez

O que precisa ser comunicado	Entrega da primeira versão da monografia completa em TCC II
Emissor	Anna Victoria Cabrera Rondon Oikawa
Receptor	Renato Cislighi
Comunicação	Entrega da primeira versão da monografia completa
Forma de comunicação	Sistema de TCC
Frequência ou Quando	Única vez

O que precisa ser comunicado	Defesa do TCC
Emissor	Anna Victoria Cabrera Rondon Oikawa
Receptor	Márcio Bastos Castro, Laércio Lima Pilla, Membro da Banca I, Membro da Banca II
Comunicação	Realização da defesa do TCC aos membros da banca.
Forma de comunicação	Pessoalmente
Frequência ou Quando	Única vez

O que precisa ser comunicado	Entrega da versão final da monografia
Emissor	Anna Victoria Cabrera Rondon Oikawa
Receptor	Renato Cislighi
Comunicação	Entrega da monografia com os ajustes feitos após a defesa
Forma de comunicação	Sistema de TCC
Frequência ou Quando	Única vez

2.5 Riscos

Nome	Probabilidade	Impacto	Exposição	Estratégia	Ações de Prevenção
Notebook danificado	Baixa	Alto	Média	Mitigar	Manter o dispositivo longe de lugares úmidos, quentes e evitar o impacto do mesmo.
Problemas de saúde do Orientando	Média	Baixo	Baixa	Aceitar	-
Indisponibilidade do Orientador/Coorientador	Média	Médio	Média	Aceitar	-

Referências

- BATHELE, A.; KALE, L.; S., K. Dynamic topology aware load balancing algorithms for molecular dynamics applications. In: *Proceedings of the 23rd international conference on Supercomputing*. [S.l.: s.n.], 2009.
- CHAUBE, R.; CARINO, R.; BANICESCU, I. Effectiveness of a dynamic load balancing library for scientific applications. In: *Sixth International Symposium on Parallel and Distributed Computing*. [S.l.: s.n.], 2007.
- FREITAS, V. M. C. T.; PILLA, L. L. Comparando diferentes ordenações de tarefas em balanceamento de carga distribuído. In: *XVII Escola Regional de Alto Desempenho*. [S.l.: s.n.], 2017.
- ILLINOIS, U. of. *Parallel Languages/Paradigms: Charm++*. 2017. Disponível em: <<http://charm.cs.illinois.edu/research/charm>>.
- LEUNG, J. Y. T. *Handbook of scheduling: algorithms, models, and performance analysis*. [S.l.]: Chapman and Hall/CRC, 2004.
- PILLA, L. *Escalonamento Global Adaptativo para Aplicações Científicas*. 2017.
- PILLA, L. L. *Topology-Aware Load Balancing for Performance Portability over Parallel High Performance Systems*. Tese (Doutorado) — Université de Grenoble, 2014.
- PILLA, L. L.; MENESES, E. *Programação Paralela em Charm++*. 2015.
- RASHID, M.; BANICESCU, I.; CARINO, R. L. Investigating a dynamic loop scheduling with reinforcement learning approach to load balancing in scientific applications. In: *International Symposium on Parallel and Distributed Computing*. [S.l.: s.n.], 2008.
- RODAMILANS, C. B. *Análise de desempenho de algoritmos de escalonamento de tarefas em grids computacionais usando simuladores*. Dissertação (Mestrado) — Universidade de São Paulo, 2009.