

Trabalho de formatura supervisionado

Avaliação de linguagens de programação quanto ao desempenho e ao tamanho do código gerado no contexto da programação paralela

Motivação

Atualmente, a grande maioria dos computadores é capaz de realizar processamento e gerenciamento de memória em paralelo. Para facilitar a implementação de algoritmos paralelos, diversos modelos teóricos existem para abstrair as arquiteturas de memória e processador [1], e cada uma das linguagens de programação implementa um ou diversos desses modelos em suas bibliotecas nativas.

Proposta

O objetivo do presente trabalho é estudar o desempenho, a clareza do código e o tamanho do código gerado por diferentes linguagens de programação atualmente utilizadas na implementação de algoritmos paralelos. Acredita-se que o modelo de programação paralela implementado pela linguagem, juntamente suas características intrínsecas (ex. o paradigma de programação que utilizam, tipagem, entre outras), podem facilitar ou dificultar a implementação desses algoritmos, gerando impacto no desempenho dos programas, no tamanho e na clareza do código [2].

Os resultados deste estudo, além de contribuírem com o conhecimento geral da aplicabilidade das linguagens, pode facilitar a escolha de uma linguagem específica para ser utilizada no ensino de programação paralela no contexto universitário.

Atividades

As atividades referentes ao trabalho estão citadas e comentadas a seguir:

1. Levantamento bibliográfico: leitura inicial sobre o tema, linguagens e modelos teóricos de programação paralela. Essa etapa tem como objetivo familiarizar o aluno com as linguagens paralelas mais populares, os modelos de programação que implementam nativamente, as técnicas mais utilizadas para avaliação de performance das implementações e os critérios de avaliação normalmente utilizados para medir a clareza de um código.
2. Escolha das linguagens: com base na leitura inicial, esta etapa visa construir o *pool* de linguagens que

serão avaliadas no trabalho, segundo o paradigma de programação, as características de tipagem e o modelo de programação paralela que utilizam. Outras características, tais como a popularidade das linguagens também serão consideradas [3].

3. Familiarização com as linguagens: esta etapa tem características teóricas e práticas, envolvendo a leitura de documentação e literatura específica das linguagens a serem utilizadas, bem como a implementação de programas simples e pequenos trechos de código paralelos para familiarização com a sintaxe e as características do código.
4. Estudo aprofundado dos modelos de programação paralela: será realizado ao longo de todo o desenvolvimento do trabalho, com o objetivo de enriquecer o conhecimento do aluno a respeito do tema, levando à escrita de códigos mais claros e de melhor desempenho
5. Escolha dos algoritmos a serem avaliados: das várias rotinas que permitem paralelização, o aluno deverá, nesta etapa, escolher aqueles que serão avaliados no presente trabalho. A etapa de levantamento bibliográfico deverá trazer informações relevantes para essa escolha.
6. Implementação dos algoritmos nas diversas linguagens: constitui a "parte prática" do trabalho, que será o objeto da avaliação proposta.
7. Escolha do método de avaliação da performance e clareza do código: definição de quais técnicas e ferramentas serão utilizadas para avaliar a performance dos algoritmos implementados na etapa anterior e a clareza do código.
8. Elaboração da monografia e preparação do poster: a execução das atividades anteriores deverá fornecer material suficiente para esta etapa, que encerrará o trabalho proposto

Cronograma

A seguir encontra-se um cronograma tentativo de realização das atividades. As atividades de caráter mais teórico serão concentradas no primeiro semestre, enquanto que as atividades de cunho prático e avaliatório serão concentradas, no segundo semestre:

Atividades/Meses	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
Levantamento bibliográfico	x	x	x						
Escolha das linguagens		x	x						
Familiarização com as linguagens			x	x					
Estudo dos modelos de PP				x	x				
Implementação dos algoritmos				x	x	x	x		
Implementação do método de avaliação					x	x	x		
Elaboração da monografia e do poster						x	x	x	x

Referências

[1] Barney, B.; *Introduction to Parallel Computing*, disponível em

https://computing.llnl.gov/tutorials/parallel_comp/#Models, acessada em abril de 2017.

[2] Nanz, S.; West, S.; da Silveira, K.S.; *Examining the Expert Gap in Parallel Programming*. In: Wolf, F.; Mohr, B.; an Mey, D., eds.; Euro-Par 2013 Parallel Processing. Euro-Par 2013. Lecture Notes in Computer Science, vol 8097. Springer: Berlin, Heidelberg.

[3] TIOBE Index: <https://www.tiobe.com/tiobe-index/>, acessada em abril de 2017.