



Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação

Proposta de Trabalho de Graduação.

LFD - MPI: Uma Ferramenta para Desenvolvimento de Aplicações Paralelas Utilizando MPI

Aluno: Rafael Rocha da Silva.

Orientador: Nelson Souto Rosa

Recife
Abril de 2015

Resumo

Este trabalho tem como objetivo projetar e implementar uma ferramenta de apoio ao desenvolvimento de aplicações paralelas baseadas no padrão MPI (Message-Passing Interface). A ferramenta proposta, chamada LFD-MPI (*Lighweight Formal Development in MPI*), será projetada para auxiliar a implementação de aplicações MPI com o apoio de métodos formais e geração automática de código. As técnicas formais serão utilizadas na verificação de propriedades desejáveis de aplicações MPI, como o deadlock.

Palavras-chave: MPI, aplicações paralelas, deadlock.

Sumário

1. Motivação	1
2. Objetivos	2
3. Estrutura do Trabalho	3
4. Cronograma	4
5. Possíveis Avaliadores	5
Assinaturas	6
Referências Bibliográficas	7

Capítulo 1

Motivação

Atualmente um grupo significativo de programas paralelos são implementados utilizando MPI (Message Passing Interface) [2]. MPI foi especialmente desenvolvido para ajudar no desenvolvimento de aplicações portáteis e eficientes, e como mencionado em [3], tem sido adotado em todas as áreas científicas que necessitam de processamento paralelo.

Apesar de sua popularidade, o desenvolvimento de aplicações corretas utilizando MPI é uma tarefa complexa, visto que essas aplicações estão sujeitas a um grande número de erros relacionados ao uso de parâmetros inválidos, uso inadequado dos recursos MPI, e também deadlock. A identificação de alguns destes erros é um desafio até mesmo no desenvolvimento de aplicações simples, uma vez que MPI fornece poucas informações que ajudam na sua correta utilização.

Várias soluções tem sido desenvolvidas para detectar erros em MPI [4,5]. Essas soluções tipicamente interceptam as invocações realizadas pela aplicação e em seguida checam sua corretude. Elas geralmente checam comportamentos indesejados através da execução da aplicação. Essa estratégia é útil, porém erros potenciais podem ser apenas identificados quando a aplicação já está completamente desenvolvida.

Capítulo 2

Objetivo

O objetivo deste trabalho é desenvolver a ferramenta LFD-MPI (*Lightweight Formal Development in MPI*). Esta ferramenta tem como objetivo auxiliar o desenvolvimento de aplicações paralelas utilizando MPI desde seu estágio inicial. Para isso, ela fornecerá um editor gráfico onde os desenvolvedores poderão planejar a estrutura de seus programas (garantindo um maior nível de abstração uma vez que o desenvolvedor não terá que manusear especificações MPI) e ainda checar propriedades como a presença de deadlock, tornando a aplicação mais robusta.

Capítulo 3

Estrutura do Trabalho

Introdução:

O primeiro capítulo do trabalho irá apresentar uma contextualização sobre aplicações paralelas assim como a motivação da criação da ferramenta que é proposta nesse projeto.

Descrição da Ferramenta:

O segundo capítulo dará uma descrição da ferramenta, especificando seu funcionamento, sua arquitetura e quais componentes ela possuirá assim como a função de cada uma dessas partes.

Desenvolvimento:

A partir deste capítulo, serão especificados cada um dos componentes da ferramenta e uma abstração de suas respectivas implementações.

Tecnologias utilizadas:

No terceiro capítulo serão definidos os frameworks utilizados para implementação da ferramenta “LFD-MPI”. O estudo dessas ferramentas será estendido por todo o trabalho.

Conclusão:

Neste capítulo serão levantadas as considerações finais sobre o trabalho.

Capítulo 4

Cronograma

As atividades desse trabalho serão executadas de acordo com o seguinte cronograma:

Atividades	Abril				Maio				Junho				Julho			
Definição do framework																
Estudo e desenvolvimento front-end/back-end																
Início do desenvolvimento da ferramenta e relatório																
Definição da estrutura XML																
Linkagem com ferramentas externas																
Gerar semi-implementação em C																
Elaboração das conclusões, finalização do relatório e dos slides																
Defesa																

Capítulo 5

Possíveis Avaliadores

São possíveis avaliadores do trabalho descrito nessa proposta:

- Paulo Romero Martins Maciel.

Assinaturas

Nelson Souto Rosa
Orientador

Rafael Rocha da Silva
Aluno

Referências Bibliográficas

- [1] ROSA, N.; WAGNER, A.: A Lightweight Approach to Formally Develop and Verify MPI Applications, Technical Report, 2015.
- [2] Forum, M.P.I.: MPI: A Message-Passing Interface Standard Version 3.0 (September 2012).
- [3] Vakkalanka, S.: Efficient Dynamic Verification Algorithms for MPI Applications. Ph.D. thesis, Salt Lake City, UT, USA (2010), aAI3413092
- [4] Vakkalanka, S., Vo, A., Gopalakrishnan, G., Kirby, R.M.: Precise dynamic analysis for slack elasticity: Adding buffering without adding bugs. In: Proceedings of the 17th European MPI Users' Group Meeting Conference on Recent Advances in the Message Passing Interface. pp. 152–159. EuroMPI'10, Springer-Verlag, Berlin, Heidelberg (2010), <http://dl.acm.org/citation.cfm?id=1894122.1894143>
- [5] Hilbrich, T., Protze, J., Schulz, M., Supinski, B.R.d., Muller, M.S.: MPI runtime error detection with MUST: Advances in deadlock detection. Scientific Programming 21(3), 109–121 (2013)