



Evento	Salão UFRGS 2019: FEIRA DE INOVAÇÃO TECNOLÓGICA DA UFRGS - FINOVA
Ano	2019
Local	Campus do Vale - UFRGS
Título	Implementação de Arquitetura Dedicada à Execução Paralela de Programas Funcionais Puros
Autor	RICARDO DE ARAÚJO COELHO
Orientador	GABRIEL LUCA NAZAR

RESUMO

[máximo duas páginas]

TÍTULO DO PROJETO: Implementação de Arquitetura Dedicada à Execução Paralela de Programas Funcionais Puros

Aluno: Ricardo de Araujo Coelho

Orientador: Gabriel Luca Nazar

RESUMO DAS ATIVIDADES DESENVOLVIDAS PELO BOLSISTA

O acelerador ACQuA(Active Call Queue Architecture) - arquitetura de hardware dedicada - é proposto tendo o objetivo de explorar o paralelismo inerente de programas em linguagens puramente funcionais automaticamente paralelizando aplicações de função independentes. Com o suporte direto da arquitetura, um desenvolvedor pode escrever programas em linguagem puramente funcional e ter o benefício do paralelismo em aplicações de funções de forma transparente, sem necessidade de recorrer a abstrações da linguagem para explicitamente criar, sincronizar e descrever a comunicação para múltiplas threads.

Nessa fase do projeto ACQuA foi desenvolvido o suporte ao processamento de listas, tanto na construção do hardware quanto no suporte do compilador para gerar as diretivas de acesso aos seus mecanismos. Isso implicou em se estender a composição de alguns módulos em linguagem de descrição de hardware, podendo-se citar: a criação de um processador de listas dentro das unidades de processamentos(PUs); inserção de novos campos de parâmetros nas trocas de mensagens entre alguns módulos; criação de memória dedicada à manipulação de listas e estruturas de dados de registro de chamadas; mecanismos de otimização de processamento, como cache com dados de chamada de funções;

Alguns algoritmos com diferentes características foram testados para a medição de performance. Esses testes consistem na medição do tempo de execução variando-se o número de PUs. Os algoritmos são divididos em dois tipos: algoritmos sobre operação em números escalares, e operação sobre listas. Dentre os escalares, foram utilizados o cálculo de fatorial, sequência de Fibonacci e método de newton raphson. Já para os algoritmos de lista, foram utilizados o cálculo de uma lista de produtos vetoriais(dot), greatest common divisor(gcd), longest common subsequence(lcs) e quicksort.

Os algoritmos que possuem chamadas recursivas independentes foram os que resultaram em um maior ganho de processamento com múltiplos cores. Alguns algoritmos que não tem essas características como o fatorial e newton-raphson não apresentaram ganhos significativos de desempenho com o aumento de núcleos de processamento. Os resultados desses testes estão

de acordo com simulações teóricas feitas via software. Os resultados são apresentados na Figura 1 e Figura 2.

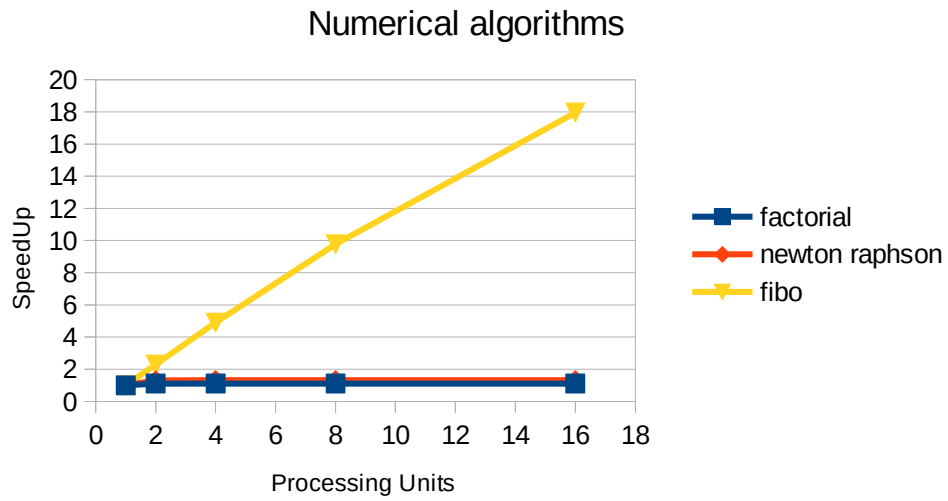


Figura 1: Medida de ganho de desempenho para os algoritmos numéricos.

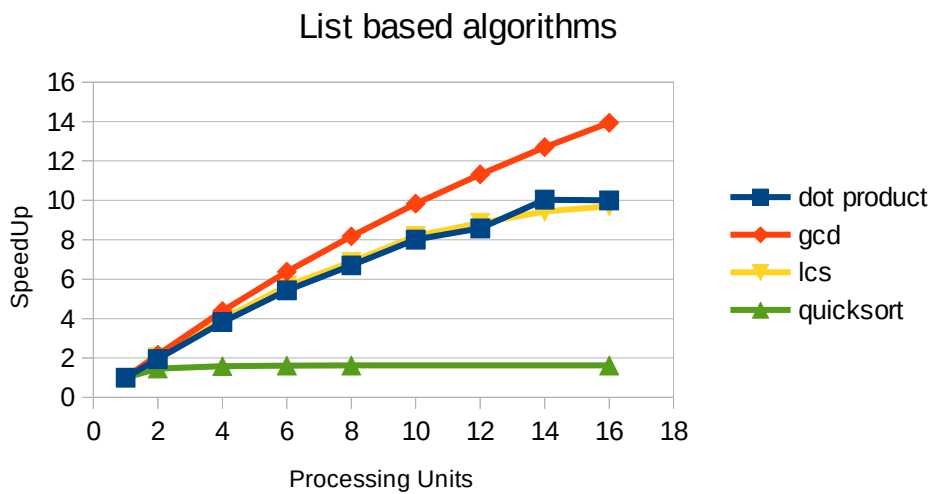


Figura 2: Medida de ganho de desempenho para os algoritmos com listas.