

Multicore e Aceleração em Hardware

Henrique Noronha Facioli - 157986

Lauro Cruz e Souza - 156175

Thiago Silva Farias - 148077

Introdução

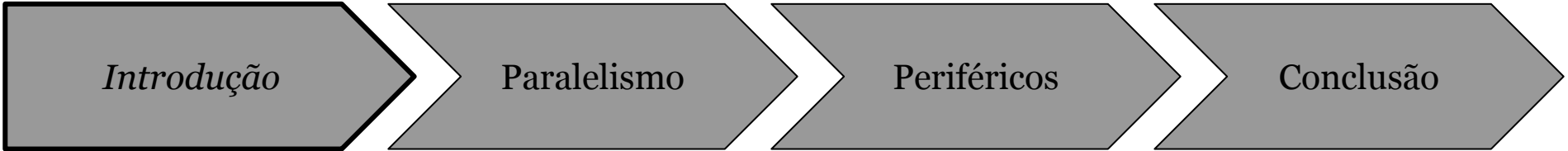
Paralelismo

Periféricos

Conclusão

FFT e KissFFT

- **FFT (Fast Fourier Transform):** é um algoritmo eficiente para se calcular a Transformada discreta de Fourier (DFT) e a sua inversa.
- Utilizada em uma vasta gama de aplicações, desde processamento digital de sinais até algoritmos para multiplicação de grandes inteiros.
- Foi usada no projeto uma implementação de FFT chamada **KissFFT**, que alia uma boa eficiência a um código relativamente fácil de ser entendido e modificado.



Experimentos

Número do Experimento	Número de Processadores	Sem/Com Acelerador
Experimento 1	1	Sem
Experimento 2	1	Com
Experimento 3	2	Sem
Experimento 4	2	Com
Experimento 5	4	Sem
Experimento 6	4	Com
Experimento 7	8	Sem
Experimento 8	8	Com

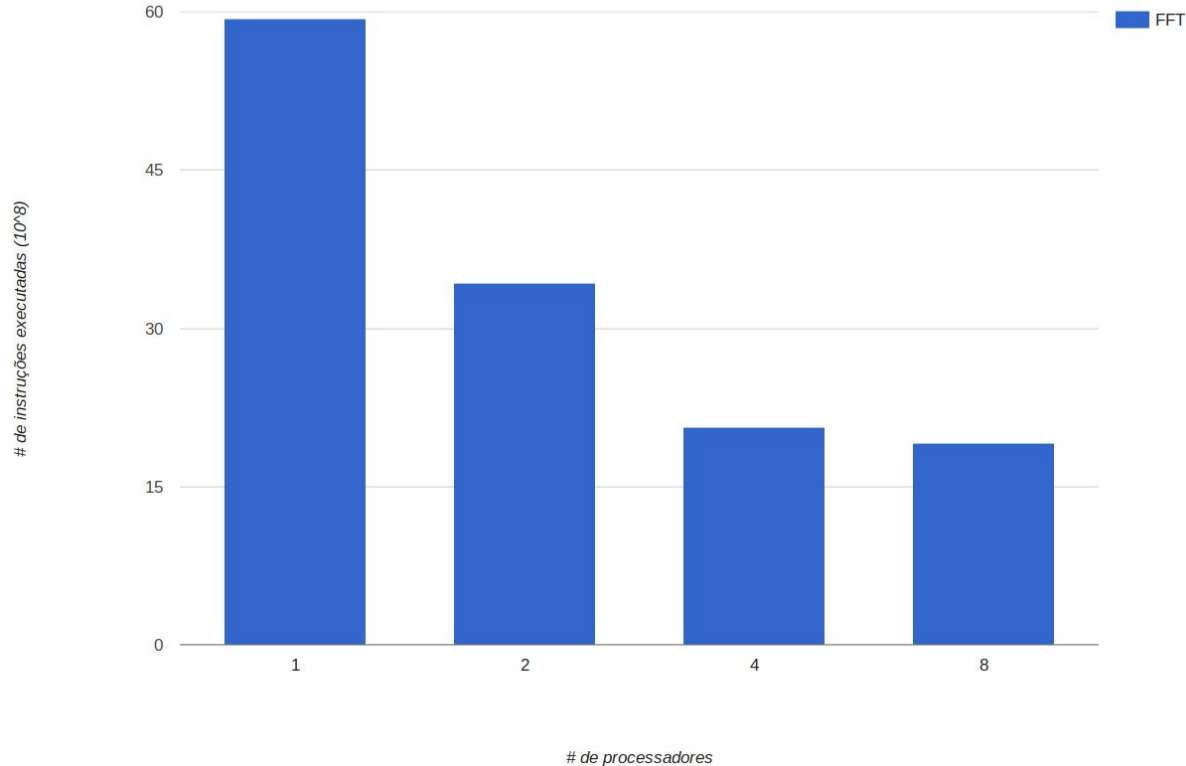
Introdução

Paralelismo

Periféricos

Conclusão

FFT - Sem aceleradores



Melhorias:

- 1 \rightarrow 2: 42.27%
- 2 \rightarrow 4: 39.71%
- 4 \rightarrow 8: 7.30 %

Má escalabilidade para
mais de 4 processadores

Introdução

Paralelismo

Periféricos

Conclusão

Acelerado de Ponto Flutuante

- Operações de ponto flutuante utilizam somente 1 única instrução.
- É escrito um número a e b em determinado endereço e a operação entre eles é lida em um outro endereço
- Permite utilização simultânea por n processadores

Escrita	$a + \text{offset}$
Escrita	$b + \text{offset}$
Leitura	$(a + \text{offset}) + (b + \text{offset})$
Leitura	$(a + \text{offset}) - (b + \text{offset})$
Leitura	$(a + \text{offset}) * (b + \text{offset})$
Leitura	$(a + \text{offset}) / (b + \text{offset})$

Introdução

Paralelismo

Periféricos

Conclusão

Operações de Ponto Flutuante

- Simplificação do uso através de API's (funções)

```
float sum_float(float a, float b, int proc);
```

```
float sub_float(float a, float b, int proc);
```

```
float mul_float(float a, float b, int proc);
```

```
float div_float(float a, float b, int proc);
```

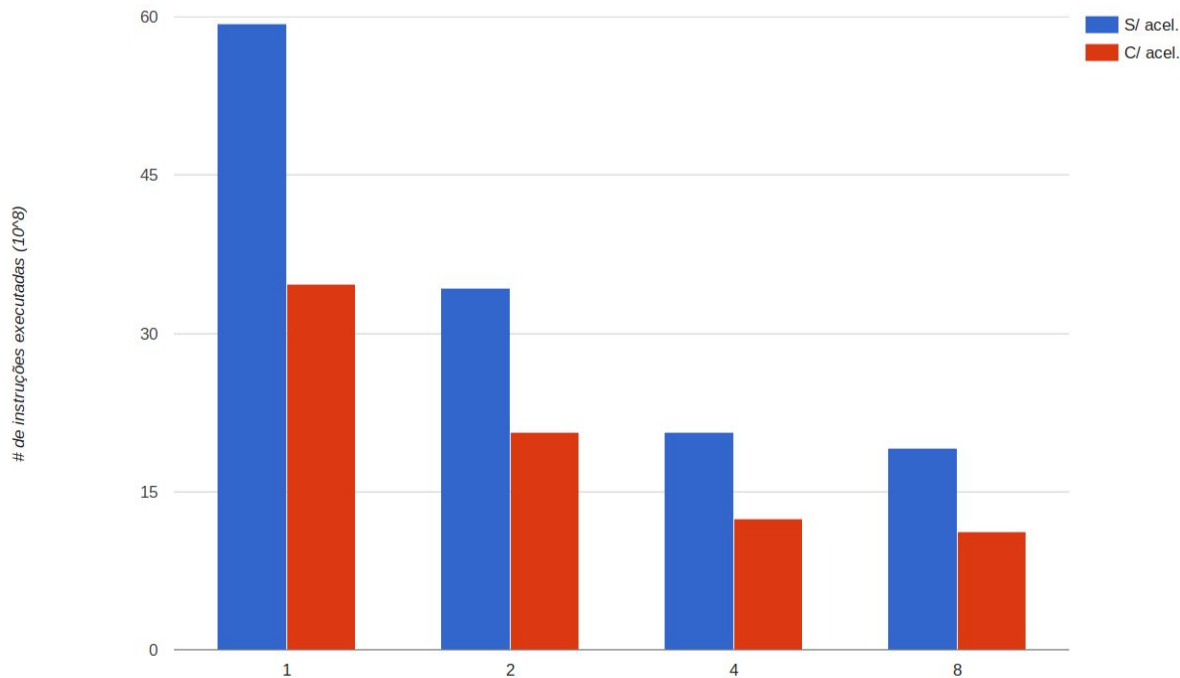
Introdução

Paralelismo

Periféricos

Conclusão

KissFFT



de processadores

Impacto de
aproximadamente 40% em
todas as simulações

- 1 \longrightarrow 41.54%
- 2 \longrightarrow 39.72%
- 3 \longrightarrow 39.77%
- 4 \longrightarrow 41.42%



Introdução

Paralelismo

Periféricos

Conclusão

- O KissFFT apresenta um ganho muito bom para 2 e 4 processadores, mas para mais processadores o ganho não é mais tão relevante, pois o sistema não foi programado para ser mais escalável.
- O impacto do uso de um acelerador é em torno de 40%, e independe do número de processadores empregado na execução.
- A configuração com o melhor resultado, considerando possíveis custos de implementação, seria aquela com 4 processadores e com uso de acelerador. Apesar de o número de instruções diminuir com o uso de 8 processadores, a queda percentual não é grande o suficiente para justificar o custo maior desse tipo de implementação.