

Relatório 8

Programação de Alto Desempenho (HPC) usando MPI – Computação Paralela

Cristiano Lopes Moreira

Matrícula: 119103-0

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
02/09/2019	1	2º. Semestre de 2019		PEL_216_Relatório_8_Cristiano_Moreira.doc			1 (11)

Sumário

1.	Introdução	3
2.	Desenvolvimento teórico	3
2.1.	Descrição do problema:	5
2.2.	Algoritmo de Monte Carlo com MPI:	6
3.	Proposta de implementação	7
4.	Experimentação e Resultados	8
5.	Conclusão	11
6.	Referências	11

Aluno		RA/Matrícula	Professor	Tipo
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação
Data	Versão	Turma	Nome do arquivo	Página
02/09/2019	1	2º. Semestre de 2019	PEL_216_Relatório_8_Cristiano_Moreira.doc	2 (11)

1. Introdução

A crescente demanda por sistemas de uso em tempo real em conjunto com os limites físicos que delimitam o crescimento da capacidade de processamento dos computadores vem desde 1955, pela iniciativa da IBM com o arquiteto de computadores Gene Amdahl, impulsionando desenvolvimento de tecnologias de Programação de Alto Desempenho (HPC) e possibilitando a interconexão de múltiplos computadores e processadores com suas unidades de memórias para dividir uma tarefa e compartilhar a capacidade de cada unidade de processamento.

Dentre as técnicas HPC desenvolvidas encontra-se a técnica de computação paralela via Interface de Transmissão de Mensagens (MPI) que desenvolve um padrão de comunicação para transmitir informações entre múltiplos computadores ou processadores, com métodos para envio e recebimento, de forma síncrona ou assíncrona.

O objetivo deste trabalho é implementar a tecnologia de computação paralela utilizando MPI, verificar e quantificar seus ganhos e perdas na tarefa de cálculo numérico de integral pelo método de Monte Carlo.

2. Desenvolvimento teórico

Como já afirmado por Amdahl (1967), a décadas os profetas expressaram a alegação de que a organização de um único computador atingiu seus limites, e que avanços verdadeiramente significativos podem ser feitos apenas pela interconexão de uma multiplicidade de computadores de maneira a permitir uma solução cooperativa.

Geralmente, métodos de computação paralela têm sido usados para aumentar o desempenho do sistema ou aumentar sua disponibilidade, porém a busca por esta interconexão trilha os desafios de conflitos no uso de recursos simultâneos como memória, discos, i/o. Três pontos são limitantes no progresso da computação com

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
02/09/2019	1	2º. Semestre de 2019	PEL_216_Relatório_8_Cristiano_Moreira.doc		3 (11)

multiprocessadores: custo, facilidade de aplicação e desempenho (RODGERS, 1985).

No que tange a custos, apartado do elevado custo de memória cache (tipicamente integrada direto na CPU), que reduz os problemas de conflitos e colisões no uso de memória entre os processadores; o crescimento da oferta dos elementos de hardware nos anos entre 2017 e 2019 reduziu os valores dos elementos computacionais que não se mostra mais um grande limitante para a computação com multiprocessadores.

No limitante de facilidade de aplicação é possível verificar que as práticas de design de sistemas de hardware e software que produzem um bom desempenho de processador monolítico quando aplicadas a multiprocessadores levam a gargalos de desempenho devido à largura de banda de memória e falta de recursos de serviços do sistema. O próprio sistema operacional gera algumas dependências de serialização, o que limita o desempenho da adição de múltiplos processadores, sendo o ponto mais importante a interação física do sistema de interconexão com o sistema de memória, já que os processadores compartilham estes recursos.

Uma das técnicas utilizadas para contornar o desafio do compartilhamento dos recursos de memória é a arquitetura de endereçamento virtual de memória, mas além de adicionar uma complexidade na gestão de tabelas de indexação de memória, cria uma latência adicional e consequente Overhead de gerenciamento do paralelismo (podendo variar de acordo com o tamanho do problema e número de núcleos utilizados) que culminando em uma redução do desempenho da aplicação [4] e [5].

Outra abordagem, ainda na parte de aplicação, é a definição de arquiteturas para uso da memória e do gerenciamento das aplicações:

Estrutura de conexão de memória por

- Multiprocessadores com barramento compartilhado no tempo
- Multiprocessadores com chave de barra cruzada

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
02/09/2019	1	2º. Semestre de 2019	PEL_216_Relatório_8_Cristiano_Moreira.doc		4 (11)

- Multiprocessadores com memória multiportas

Gestão das aplicações por

- Supervisor separado em cada processador
- Sistema operacional mestre-escravo
- Sistema operacional de supervisor flutuante

De todas as formas os sistemas de multiprocessadores compartilham (memória, I/O, devices, interrupções de sistemas etc.) e recursos como memória e I/O são atribuídos dinamicamente a processos, e não permanentemente conectados aos processadores (RODGERS, 1985).

Nos estudos realizados por Amdahl (1967) ele já observava que a fração computacional responsável por gerenciamento de memória corresponde cerca de 40%; esse fato somado as instruções que são necessariamente sequencias, como I/O, geram um gargalo a todo o processamento e coloca um limite superior na aceleração

Aceleração
$$S = \frac{1}{r_s + \frac{r_p}{n}} \quad (1)$$

Onde $r_s + r_p = 1$ (2) e r_s representa a proporção da porção sequencial em um programa.

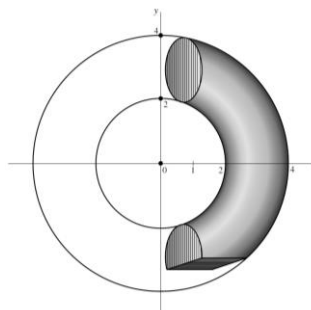
Levando (2) em (1)
$$S = \frac{1}{(1 - r_p) + \frac{r_p}{n}} \quad (3)$$

2.1. Descrição do problema:

Deseja-se calcular o volume do Toroide seccionado, proposto pelo por Press et al. (2007), utilizado a metodologia de Monte Carlo implementada com computação

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
02/09/2019	1	2º. Semestre de 2019	PEL_216_Relatório_8_Cristiano_Moreira.doc		5 (11)

paralela utilizando os padrões de MPI, e avaliar a eficiência do paralelismo em processadores multicore de 4 cores.



O objeto é definido pelas seguintes equações [3]:

$$\text{Toroide} \quad z^2 + \left(\sqrt{x^2 - y^2} - 3 \right)^2 \leq 1 \quad (4)$$

$$\text{Seccionamento} \quad x \geq 1 \text{ e } y \geq 1 \quad (5)$$

2.2. Algoritmo de Monte Carlo com MPI:

O algoritmo de integração numérica pelo método de Monte Carlo propõe sortear variáveis de sua equação ou ambiente, realizar a interação dessas variáveis sorteadas com o ambiente e/ou função que se deseja obter o resultado estatístico, anotar os resultados e repetir N vezes até que suas amostras sejam suficientes para representar o resultado. A característica essencial do processo é que evitamos lidar com múltiplas integrações e passamos a gerar amostras em uma distribuição uniforme.

Sua integração utilizando MPI é realizada pela divisão da tarefa de gerar múltiplos sorteios e cálculos entre os diversos processadores e, na sequência, centralizar as informações geradas no elemento Master que concluirá o cálculo da integral utilizando todos os dados.

Seus métodos principais são:

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
02/09/2019	1	2º. Semestre de 2019		PEL_216_Relatório_8_Cristiano_Moreira.doc			6 (11)

construtor (fx, xlo, xhi, 3, erro, sorteios): inicializa o objeto MonteCarlo com os parametros, funcao , limite inferior, limite superior, erro aceitável e quantidade de sorteios.

gera_dados (): calcula e acumula $F(x)$ e $F(x)^2$, por métodos estatísticos (estocástico);

calc (): Calculo o valor da integral e do erro utilizando as informações obtidas pelo método gera_dados();

qTpassos (): retorna à quantidade de sorteios e cálculos realizados;

ParcialFx(): retorna o valor do acumulado de $f(x)$ pelo método gera_dados();

ParcialFx2(): retorna o valor do acumulado de $f(x)^2$ pelo método gera_dados();

MPI_init(): inicializa os multiprocessos em paralelo;

MPI_reduce(): reúne e soma as variáveis de cada processo e coloca à disposição do processo Master;

MPI_finalize(): encerra todos os processos em paralelo.

3. Proposta de implementação

Pseudocódigo:

MPI_init()

gera_dados()

inicializa a semente randômica

loop no número de eventos

gera um número aleatório x,y,z entre limites

calcula $f(x,y,z)$

calcula somatoria F_{xyz}

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
02/09/2019	1	2º. Semestre de 2019		PEL_216_Relatório_8_Cristiano_Moreira.doc			7 (11)

```

        calcula somatoriaFxyz ~2
    fim loop

MPI_reduce(dados gerados)

Se task==Master
    calcula integral (dados gerados)
    retorna integral
MPI_finalize

```

4. Experimentação e Resultados

Foram executados 230 ciclos de episódios (rodadas de verificações com números aleatórios) com 100.000 números aleatórios sendo distribuída a tarefa entre processos paralelos utilizando MPI da seguinte forma: 1 acrescido em 1 até 10 processos, de 10 acrescido de 10 até 100 processos e de 100 acrescido de 50 até 250; como observação inicial do resultado comparado com a aplicação executada sem nenhuma técnica de paralelismo.

Ambiente:

Processador model name : Intel(R) Core(TM) i5 CPU U 470 @ 1.33GHz
4 cores cache size : 3072 KB

OS Linux openSUSE Leap 15.1

Resultado do cálculo do Volume do Toroide seccionado pelo método de Monte Carlo

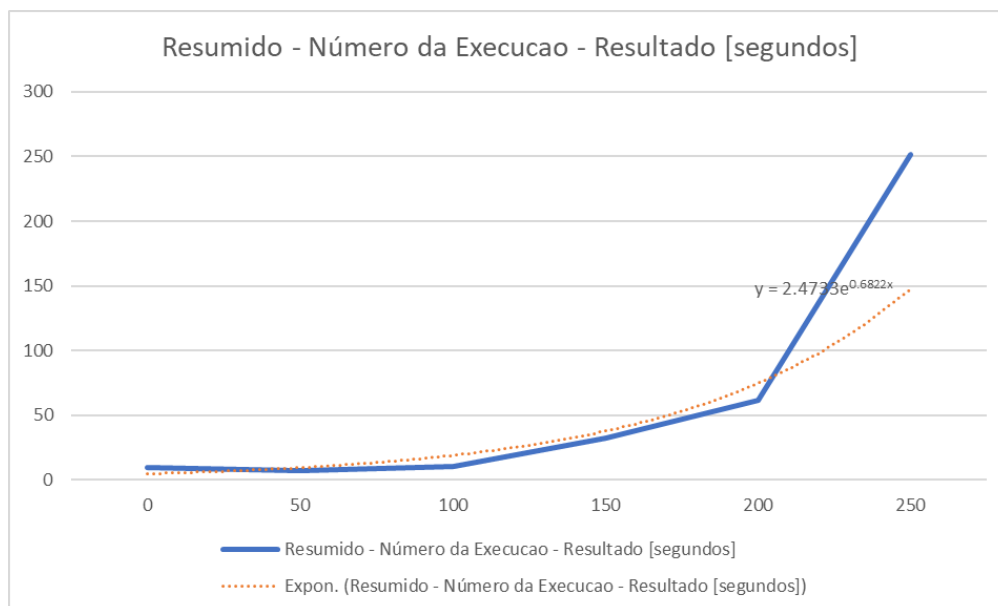
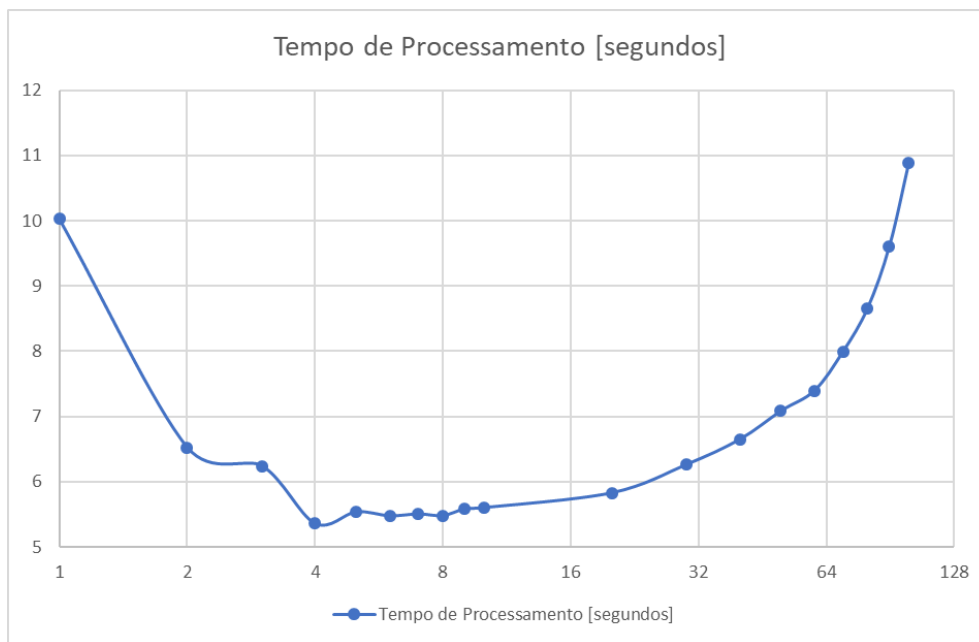
Monte Carlo		
N	Integral	Erro
100.000	22.1827200000	0.066302426305649

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
02/09/2019	1	2º. Semestre de 2019	PEL_216_Relatório_8_Cristiano_Moreira.doc		8 (11)

Tempo de execução por processos no MPI

100.000 Sorteios											
Número da Execução - Resultado [segundos]											
Processos	1	2	3	4	5	6	7	8	9	10	Média
S/ MPI	9.7	9.72	9.71	9.7	9.7	9.71	9.7	9.71	9.71	9.7	9.706
1	9.98	10.01	10.28	10.02	10.01	9.99	9.99	10	10	10	10.02
2	6.52	6.51	6.54	6.5	6.53	6.54	6.53	6.52	6.53	6.54	6.526
3	6.44	6.3	6.13	6.16	6.7	6.34	6.15	6.11	6.12	5.95	6.24
4	5.33	5.33	5.34	5.34	5.34	5.37	5.34	5.35	5.34	5.49	5.357
5	5.54	5.53	5.51	5.58	5.5	5.55	5.63	5.48	5.56	5.45	5.533
6	5.52	5.48	5.51	5.5	5.45	5.43	5.54	5.41	5.49	5.4	5.473
7	5.45	5.53	5.57	5.46	5.56	5.47	5.48	5.58	5.45	5.46	5.501
8	5.43	5.48	5.52	5.43	5.45	5.51	5.52	5.53	5.46	5.44	5.477
9	5.62	5.52	5.52	5.56	5.5	5.75	5.62	5.61	5.58	5.52	5.58
10	5.62	5.58	5.65	5.52	5.61	5.65	5.58	5.59	5.6	5.59	5.599
20	5.87	5.81	5.76	5.84	5.89	5.83	5.78	5.87	5.78	5.83	5.826
30	6.35	6.11	6.32	6.34	6.13	6.42	6.18	6.35	6.12	6.31	6.263
40	6.82	6.76	6.44	6.47	6.74	6.47	6.76	6.5	6.82	6.67	6.645
50	7.17	7.1	7.12	7.13	7.05	7.09	7.02	7.15	6.94	7.05	7.082
60	7.55	7.29	7.29	7.42	7.24	7.61	7.33	7.36	7.43	7.44	7.396
70	7.88	8.02	8.02	8.19	7.94	8.01	7.87	8.11	7.93	7.97	7.994
80	8.7	8.6	8.63	8.56	8.72	8.7	8.75	8.6	8.74	8.61	8.661
90	9.47	9.46	9.77	9.71	9.7	9.58	9.51	9.68	9.53	9.7	9.611
100	10.7	10.96	10.87	10.91	10.84	10.77	11.39	10.63	10.93	10.9	10.89
150	32.57	32.57	32.83	33.62	32.5	34.39	32.56	33.37	31.71	31.64	32.77
200	62.55	45.52	65.71	65.18	63.36	58.9	61.55	68.48	63.76	62.58	61.75
250	318	251	248	241	239	241	248	233	235	259	251

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
02/09/2019	1	2º. Semestre de 2019		PEL_216_Relatório_8_Cristiano_Moreira.doc			9 (11)



Aluno Cristiano Lopes Moreira		RA/Matrícula 119103-0	Professor Dr Reinaldo Bianchi	Tipo Relatório de implementação	
Data 02/09/2019	Versão 1	Turma 2º. Semestre de 2019	Nome do arquivo PEL_216_Relatório_8_Cristiano_Moreira.doc		Página 10 (11)

5. Conclusão

O acréscimo de processos paralelos até o máximo do número de cores do processador acelera a execução da tarefa computacional, porém não proporcional à quantidade de cores adicionadas, é observada reduções do acréscimo de aceleração a cada processador adicionado em confirmação da lei de Amdahl.

Para acréscimos de processos acima do número de cores do processador observa-se que o comportamento é contrário, desacelerando a execução da tarefa computacional de forma exponencial com a quantidade de processos adicionados, o que confirma os custos de Overhead mensurados nos trabalhos de Höfinger e Haunschmid (2017) e Oliveira et al. (2018).

6. Referências

- [1] AMDAHL, Gene M.. Validity of the single processor approach to achieving large scale computing capabilities. In: AFIPS SPRING JOINT COMPUTER CONF, 67., 1967, Atlantic City. **(Spring) Proceedings of the April 18-20**. Reston: Afips Press, 1967. v. 30, p. 483 - 485
- [2] RODGERS, David P.. Improvements in Multiprocessor System Design. In: ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE, 12., 1985, Boston. **ISCA '85 Proceedings**. Los Alamitos: Ieee Computer Society Press, 1985. p. 225 - 231.
- [3] PRESS, William H. et al. **Numerical Recipes: The Art of Scientific Computing**. 3. ed. Cambridge, Massachusetts: Cambridge University Press, 2007. Cap. 7. p. 397-401
- [4] HÖFINGER, Siegfried; HAUNSCHMID, Ernst. **Modelling parallel overhead from simple run-time records**. 2017. The Journal of Supercomputing. Disponível em: <<https://doi.org/10.1007/s11227-017-2023-9>>. Acesso em: 31 mar. 2017.
- [5] OLIVEIRA, Victor H. F. et al. Application Speedup Characterization: Modeling Parallelization Overhead and Variations of Problem Size and Number of Cores. In: ACM/SPEC INTERNATIONAL CONFERENCE ON PERFORMANCE ENGINEERING, 18., 2018, Berlin. **Proceeding ICPE '18 Companion of the 2018**. New York: Acm, 2018. p. 43 - 44

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
02/09/2019	1	2º. Semestre de 2019		PEL_216_Relatório_8_Cristiano_Moreira.doc			11 (11)