

Relatório de Paralelização do N-Body com MPI

Welliton Slaviero
178342@upf.br

Informações Gerais:

Este relatório apresenta uma visão objetiva da paralelização do problema N-Body utilizando MPI. A implementação foi baseada no modelo mestre-escravo, no qual o mestre distribui tarefas para os escravos, que realizam os cálculos e retornam os resultados. Essa estrutura permitiu dividir os loops do código sequencial, mantendo a lógica original e otimizando o tempo de execução.

Dificuldades encontradas:

O principal desafio foi ajustar corretamente os loops para manter a lógica semelhante ao código original sequencial, o que exigiu diversas tentativas e ajustes. Outra dificuldade relevante foi um problema técnico que ocorreu ao tentar executar o código em múltiplas máquinas no laboratório LCI. Ao distribuir a execução em quatro máquinas, surgiu um erro informando que algumas máquinas não tinham acesso ao executável. Apesar das tentativas, não consegui resolver o problema, limitando a execução ao uso de apenas uma máquina (lci-16-1-lnx, IP: 192.168.14.201).

Resultados:

Realizei testes utilizando diferentes quantidades de processos e registrei os tempos de execução. As execuções com 2 e 4 processos mostraram ganho moderado em relação ao sequencial, com 8 processos, o desempenho já era bem mais expressivo. A melhor execução ocorreu com 12 processos, atingindo o menor tempo de execução. A partir de 16 processos, o desempenho começou a piorar novamente, indicando perda de eficiência devido ao overhead de comunicação. Com os dados, desenvolvi uma função $f(x) = a/x + bx + c$ representando o tempo em função do número de processos, e utilizei Python para gerar gráficos e tabelas. Adicionei 5% de ruído gaussiano às simulações para refletir variações naturais de execução.

Principais resultados:

Tempo sequencial	Melhor tempo paralelo	Melhor speedup	Melhor eficiência
426.116928 s	82.781426 s (12p)	5.15 (12 proc.)	81.29% (2 proc.)

Análise breve:

Granularidade: Boa até 12 processos, sendo maior o tempo gasto em cálculo do que em comunicação.

Escalabilidade: Eficiente até 12 processos; após este ponto, o desempenho começa a diminuir significativamente.

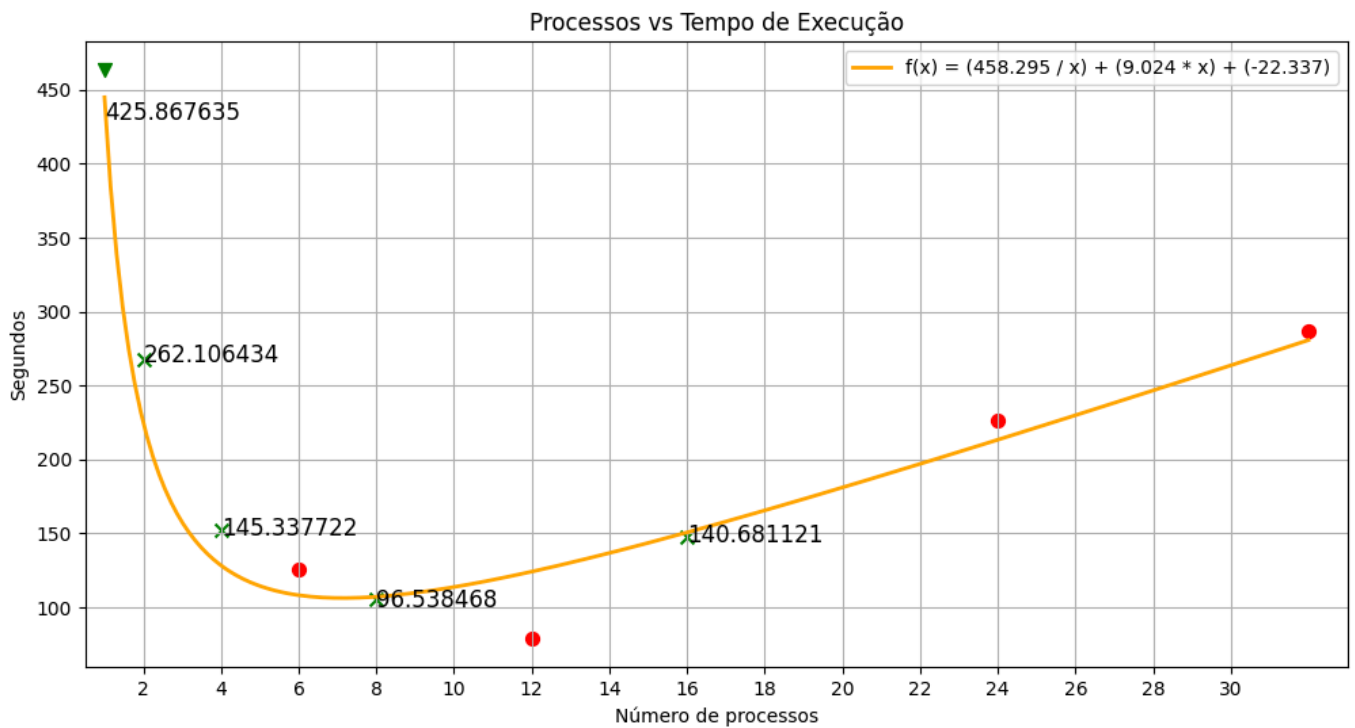
Facilidade de programação: A implementação foi relativamente simples, embora tenha exigido cuidado na sincronização e nas mensagens trocadas.

Conclusões:

A paralelização utilizando MPI proporcionou ganhos expressivos de desempenho, especialmente até 12 processos. Apesar das limitações técnicas ao tentar executar em múltiplas máquinas, o resultado final foi satisfatório. Para futuros estudos, é recomendável explorar métodos mais eficientes de comunicação, até mesmo aprimorar o uso das funções MPI, considerando uma divisão mais dinâmica das tarefas para melhorar ainda mais o desempenho.

Anexos:

Gráfico:



Resultados Intermediários:

#	Número de Processos	#	Tempo de Execução Paralelo	#	Speedup	#	Eficiência	#	Custo Computacional
	2		262,106434		1,625740053		0,8128700267		524,212868
	4		145,337722		2,931908675		0,7329771689		581,350888
	6		112,974103		3,771810678		0,628635113		677,844618
	8		96,538468		4,413959915		0,5517449894		772,307744
	12		82,781426		5,147494415		0,4289578679		993,377112
	16		140,681121		3,028956017		0,1893097511		2250,897936
	24		227,914097		1,869638314		0,07790159641		5469,938328
	32		288,240091		1,478340249		0,04619813279		9223,682912