UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA

JUDSON DOS SANTOS COSTA
MIGUEL EDUARDO DE SOUTO
WELLINGTON MEDEIROS DE ARAÚJO VAZ

ANÁLISE DE ESCALABILIDADE - PARSEC
LU CB

Relatório apresentado ao Componente Curricular de Programação Concorrente e Distribuída do Departamento de Engenharia de Computação e Automação, UFRN.
Professor: Samuel Xavier

NATAL – RN
NOVEMBRO/2019

INTRODUÇÃO

OpenMP é uma API (Application Programming Interface) para a programação paralela em sistemas de memória compartilhada. O algoritmo o qual está sendo trabalhado - LU CB - está também usando uma API para a programação paralela em sistemas de memória compartilhada, que é o Pthreads, no entanto, Pthreads requerem que o programador especifique explicitamente o comportamento de cada thread. Por outro lado, OpenMP pode permitir que o programador declare que um bloco de código deve ser executado em paralelo e o compilador faz o tratamento sobre quais serão as tarefas e quais threads executarão tais tarefas.

Pthreads, por ser baixo nível, acaba por deixar para o programador especificar todo detalhe sobre o comportamento de cada thread. OpenMP, por outro lado, permite ao compilador e ao sistema de tempo de execução determinar alguns dos detalhes dos comportamento de cada thread. Isso torna programar comportamento de processamento paralelo mais simples, usando OpenMP.

Conceitos:

- Escalável: a eficiência não diminui ao aumentar número de processos e tamanho do problema.
- Francamente escalável: a eficiência se mantém ao aumentar número de processos e tamanho do problema.
- Fortemente escalável: a eficiência se mantém ao aumentar número de processos e manter tamanho do problema constante.

RESULTADOS

Para obter os resultados, foram executadas as duas versões dos algoritmos (Pthread e OpenMP) no Supercomputador utilizando o SBATCH, com a flag de otimização O2.

PTHREADS					
EFICIÊNCIA					
PROCESSOS	TAMANHO DA MATRIZ				
	512	1024	2048	4096	8192
1	1,00	1,00	1,00	1,00	1,00
2	0,87	0,95	0,98	0,99	0,98
4	0,75	0,90	0,95	0,98	0,98
8	0,41	0,77	0,89	0,95	0,96
16	0,19	0,52	0,77	0,90	0,93
32	0,08	0,18	0,35	0,44	0,51

Tabela 01: Eficiência do algoritmo em Pthreads

OPENMP					
EFICIÊNCIA					
PROCESSOS	TAMANHO DA MATRIZ				
	512	1024	2048	4096	8192
1	1,00	1,00	1,00	1,00	1,00
2	0,94	1,00	1,00	0,99	0,99
4	0,96	1,00	0,99	1,00	0,99
8	0,99	0,97	1,00	1,00	0,99
16	1,00	1,00	0,98	0,98	0,98
32	1,00	0,99	0,94	0,95	0,94

Tabela 02: Eficiência do algoritmo em OpenMP

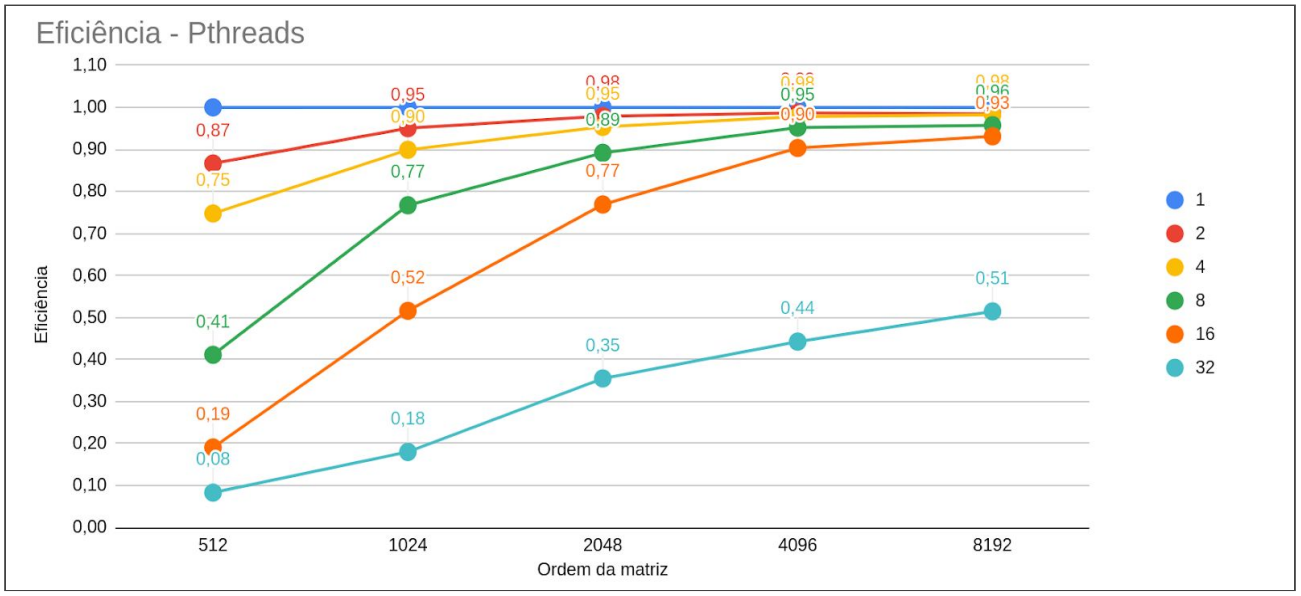

Gráfico 01: Eficiência do algoritmo em Pthreads

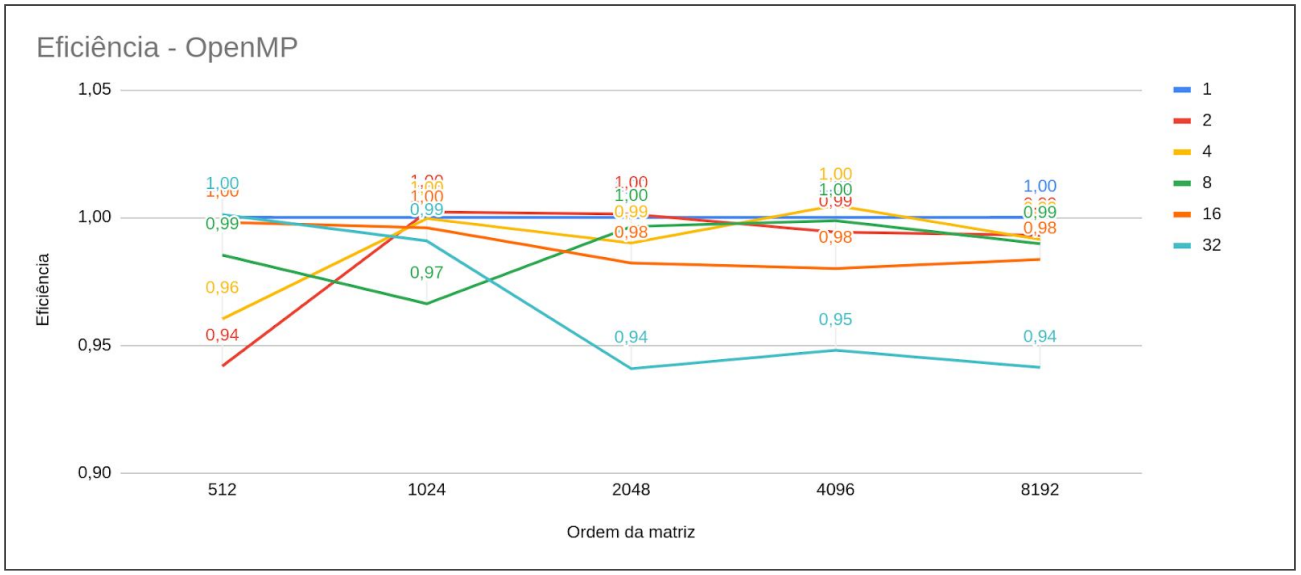

Gráfico 02: Eficiência do algoritmo em OpenMP

CONCLUSÕES

Foi realizado com êxito a portabilidade do código escrito usando Pthreads para OpenMP e foi verificado que o uso de OpenMP torna o código menos verboso e mais eficiente apesar de ser uma API de alto nível.

Após realizadas as etapas de perfilagem, vetorização, portabilidade e análise da escalabilidade dos programas escritos em Pthread e OpenMP, constatamos que a escalabilidade pode ser classificada como fracamente escalável em certos intervalos.