**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**

**CENTRO DE TECNOLOGIA**

**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO**

**Paralelização e Análise de Escalabilidade na Decomposição LU**

**Discentes:** Flávio Henrique Lopes Barbosa**,**

Jaysa Keylla Siqueira Barbosa,

José Augusto Agripino de Oliveira,

Klyfton Stanley Fernandes da Silva Queiroz.

**Introdução**

A aplicação de Decomposição LU, mesmo sendo um algoritmo pequeno, em relação a sua lógica e linhas de código, requer um alto poder de processamento quando aplicada a matrizes com grandes ordens e quantidade de elementos. Esta prática tem como objetivo a otimização do código utilizado relativo ao respectivo algoritmo, bem como através da ferramenta Pascal Suite e o compilador GCC, ter-se a capacidade de analisar a escalabilidade do programa e se este é forte ou fracamente escalável, e por conseguinte seu desempenho e Eficiência.

**Metodologia**

Para fazer a análise correta da escalabilidade, precisamos fazer as devidas medições do programa sequencial e pegar o seu melhor tempo. Para isso, o NPAD foi utilizado com as entradas 4000, 6000, 8000, 12000 e 16000.

Compilamos o programa:

$ gcc -g -pg -Wall -o omp\_decomposicaoLU omp\_decomposicaoLU.c

Utilizamos um script para que fosse possível rodar em um nó de computação exclusivo:

*#!/bin/bash*

*#SBATCH --time=0-0:20*

*#SBATCB --exclusive*

*./omp\_decomposicaoLU n <ordem da matriz>*

Com o resultado adquirido a partir do gprof, podemos partir para o desenvolvimento do programa paralelo. Para fazer a paralelização foi utilizada a ferramenta OpenMp, adicionando a biblioteca omp.h. Já para realizar a análise de escalabilidade, foi modificado o tamanho do problema que o algoritmo precisa resolver. Na nossa aplicação de decomposição LU, foi feita alterando o tamanho da matriz. Para análise, foram utilizadas as ordens 4000, 6000, 8000, 12000 e 16000 para a matriz a ser decomposta. Ademais, para cada ordem, foram utilizados quatro valores de threads diferentes: 2, 4, 8 e 16 threads, a fim de observar o desempenho obtido. A máquina usada foi o supercomputador do NPAD.

Compilamos o programa:

$ gcc -g -Wall -fopenmp -o omp\_decomposicaoLU omp\_decomposicaoLU.c

Utilizamos um script para que fosse possível rodar em um nó de computação exclusivo:

*#!/bin/bash*

*#SBATCH --job-name=omp\_LU*

*#SBATCH --time=0-10:0*

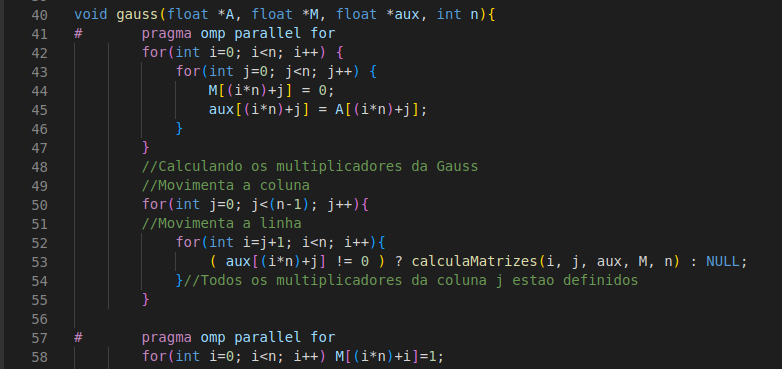
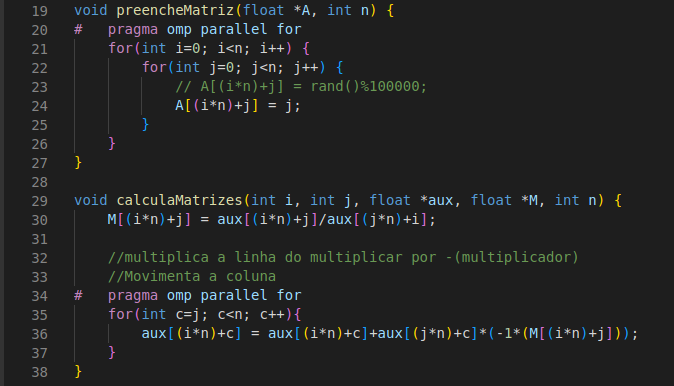
*#SBATCH --cpus-per-task=8*

*#SBATCH --hint=compute\_bound*

*#SBATCB --exclusive*

*./pascalanalyzer -c 1,2,4,8,16 -i 4000,6000,8000,12000,16000 ./omp\_decomposicaoLU -o omp\_decomposicaoLU.json*

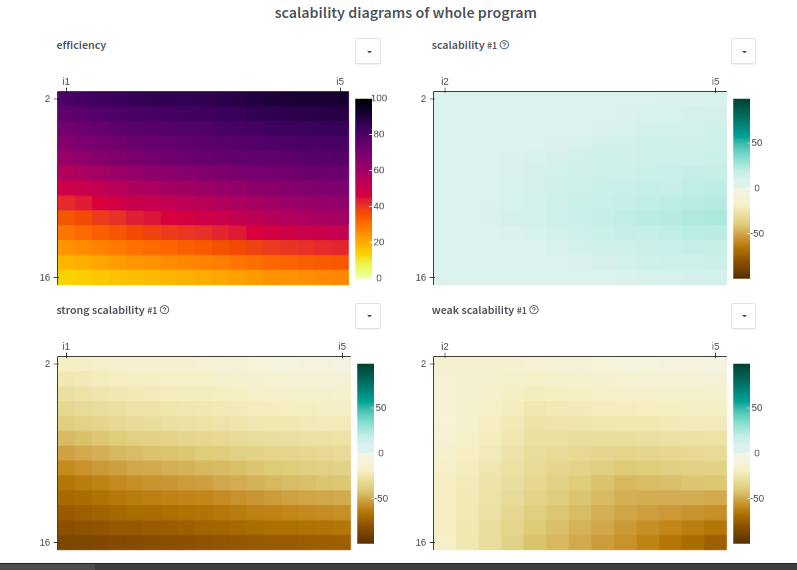
Foi utilizado o pascal analyzer para obter os resultados de escalabilidade do programa, alterando o número de cores e a ordem da matriz, que é a entrada do nosso problema. Com o resultado, um arquivo .json, conseguimos fazer proveito do pascal suite, que é uma ferramenta da UFRN para a análise da escalabilidade com o arquivo gerado a partir do pascal analyzer. A seguir estão trechos do código já paralelizado:



**Resultados**

**Análise de Escalabilidade**

A análise de escalabilidade foi feita, através da perfilagem e da ferramenta Pascal Viewer, que forneceu a possibilidade de visualizarmos, através de sua interpretação do rastreio da execução feita pelo Analyzer, as regiões de Eficiência e Escalabilidade através dos gráficos abaixo:



Executando o código com as diferentes entradas e número de threads, esses foram os resultados obtidos em tempo de execução. Foram um pouco menores que o tempo de execução sequencial, mas deveriam ter sido bem mais baixos, uma vez que o trabalho está sendo dividido entre as várias threads. Isso ocorreu devido ao erro em usar apenas 4 cores, e não as 16 exigidas pelo nosso problema, que foi definido no script de execução.

## **Conclusão**

Os resultados obtidos com a paralelização demonstram que o programa é escalável - mesmo com o erro em relação ao número de cores -, através das análise das regiões pelo Pascal Viewer, constatou-se que o programa não é fortemente ou fracamente escalável. E nesta atividade, foi possível vermos o quão poderosos são os recursos de paralelização, ferramentas e técnicas utilizadas na construção do entendimento dos temas e objetivos propostos.

É importante salientar que,mesmo com alguns desvios de percepção em relação aos métodos, os conceitos ficaram claros e a percepção em relação a execução de programas, por parte do grupo, tem uma perspectiva bem mais ampla em relação ao início do curso.