UNISC UNIVERSIDADE DE SANTA CRUZ DO SUL	**Departamento de Engenharias, Arquitetura e Computação** Disciplina de Programação Paralela e Distribuída **Prof.: Geancarlo Abich** **Trabalho Prático 1 – 15/04/2023**

Especificação do Trabalho 1

O exercício pode ser feito em grupo de no máximo **4 pessoas**. Lembrando: ambos os integrantes entregam a tarefa no moodle, mencionando o nome de todos os integrantes.

Apresentação

Este trabalho tem por objetivo explorar o paralelismo em uma aplicação de rede neural convolucional (RNC) CIFAR-10 desenvolvida em C/C++ disponível dentro dos exemplos da biblioteca CMSIS-NN para microcontroladores Arm.

Requisitos

Cada grupo deve desenvolver versões da CNN CIFAR-10 que execute em pelo menos 4 threads em paralelo ou o número máximo de núcleos físicos da sua máquina host considerando: biblioteca pthreads, diferentes compiladores, variação no número de taredas/threads, 1000 imagens como entrada.

Cada grupo deve encontrar os pré requisitos do trabalho:

- Biblioteca POSIX-threads (PThreads)
- Compiladores GCC e CLANG
- Gerar 1000 entradas para a RNC (100 imagens diferentes e de cada tipo, podem usar python para isso).

Repositório CMSIS: https://github.com/ARM-software/CMSIS_5/tree/5.7.0/ .

Tarefas

Etapa 1)

1. Extrair a RNC do código e compilar para a arquitetura da sua máquina host.
2. Calcular o tempo de execução da RNC na sua máquina Host.
3. Gerar 1000 entradas para a RNC (100 imagens diferentes de cada tipo, podem usar python para isso) e alterem o código para suportar essa entrada de dados, ou seja, cada execução deve processar todas as entradas.

Etapa 2)

1. Paralelizar com Pthreads a execução da RNC para que cada inferência (detecção de padrões) execute em 1 thread.
2. Casos com pelo menos 1, 2, 3 e 4 threads (ou até o máximo de núcleos físicos da sua máquina host) e verifiquem quantas inferências são feitas por segundo para cada uma das configurações.

3. Utilizar 2 compiladores: GCC (v10 ou maior) e CLANG (v10 ou maior).

Recomendações para desenvolvimento do trabalho: Utilizem ponteiros para apontar para as imagens de cada inferência, isso faz com que reduza os requisitos de memória para a implementação. Além disso, extraiam os valores dos píxels RGB de cada imagem (como está feito na versão inicial) para reduzir o escopo de memória utilizado. Utilizem contadores para contar quantas imagens de cada tipo foram reconhecidas e printem estes valores ao final da execução. Capturar o tempo de execução das versões sequencial e paralela, calculando quantas inferências são feitas por segundo. Ex.: digamos que para processar as 1000 imagens a RNC execute durante 28 segundos, logo temos que são reconhecidas 35,7 imagens por segundo (inferencias). Ao final comparem os tempos de cada versão (paralela e sequencial) e cada compilador (GCC e CLANG) e compare os ganhos de desempenho. Podem extrair os tempos tanto para o número máximo de núcleos de processamento quanto para o número máximo de threads suportada por sua máquina host e verifiquem se quando excedemos o número de threads por núcleo ainda atingimos algum ganho de desempenho. Atenção: utilizem as funcionalidades da biblioteca pthreads (mutexes e semaforos) para controlar o número máximo de threads simultaneas, ou seja, limitar o número de threads sendo executadas para cada um dos casos paralelos (1, 2, 3, 4 threads ou até o número máximo de núcleos físicos da sua máquina host).

Entrega

O trabalho deverá ser realizado individualmente ou em grupo de no máximo 4 pessoas. Cada grupo deve entregar:

1. O projeto contendo
   a. Os códigos utilizados para todas as versões, comentados e bem detalhados.
   b. Os makefiles para o projeto.
   c. Os scripts de geração das imagens.
2. Relatório no formato ABNT/SBC (modelo disponível no moodle)
3. Apresentação do Trabalho funcionando (todas as versões) e análise dos ganhos.

Recomendações para a escrita do artigo: Introdução (contextualizando o atual cenário da complexidade de aplicações computacionais e de machine learning e a necessidade da paralelização), Fundamentação teórica/Background (descrevendo as principais arquiteturas paralelas e bibliotecas de paralelismo, bem como sobre redes neurais convolucionais), Estudo de caso (contemplando a explicação sobre análise do desempenho da aplicação RNC CIFAR-10 sequencial, detalhes da implementação usando as bibliotecas de paralelismo e makefile, cenários dos testes realizados, análise de desempenho). Conclusões (fechamento do artigo) e Referências.

Prazos

1. Entradas geradas e versão sequencial (single thread) através de entrega do projeto parcial ou com um vídeo do funcionamento printando os outputs de todas as entradas **14/04/2023**.

2. Projeto (Código + makefiles com todas as versões), apresentação e relatório **20/04/2023**.