

10 - Tarefas e acessos sincronizado a dados

Super Computação 2018/2

Igor Montagner, Luciano Soares

Na aula expositiva vimos uma série de diretivas do OpenMP para sincronização de threads e uma técnica para sincronização no acesso a dados (locks).

Vamos trabalhar em um mini projeto nesta aula: fazer um programa que recebe dois argumentos: um diretório e uma string e busca nos arquivos deste diretório esta string. A saída do programa é uma lista de pares

arquivo:linha

onde a string foi encontrada. Um script que baixa uma lista de arquivos texto para teste está no github junto com o restante dos materiais.

Neste roteiro iremos usar a biblioteca *Boost*. Ela contém uma série de módulos que abstraem detalhes de implementação e facilitam a construção de programas de alto nível em C++. Neste exemplo iremos usar a *Boost-filesystem* para trabalhar com caminhos e listagem arquivos. Veja abaixo um exemplo de como listar todos arquivos na pasta atual. Mais detalhes podem ser encontrados na documentação do *Boost-filesystem* (https://www.boost.org/doc/libs/1_44_0/libs/filesystem/v3/doc/index.htm).

```
#include <iostream>
#include "boost/filesystem.hpp"

using namespace boost::filesystem;

int main(int argc, char **argv) {

    path p(".");

    for (auto &entry : directory_iterator(p)) {
        std::cout << entry.path().string() << is_regular_file(entry.path()) << "\n";
    }

    return 0;
}
```

Para abrir os arquivos e buscar em seu conteúdo vocês devem usar a biblioteca padrão (*fstream* - [docs](#)) e funções de `std::string`.

Paralelização

Após criar uma versão sequencial de seu programa você deverá paralelizá-los usando *tasks* do OpenMP. Se necessário, você pode usar locks ou `critical` para sincronizar o acesso a recursos no programa.

Restrição: a saída deverá estar agrupada por arquivo e em ordem crescente das linhas.

Exercício: Compare a velocidade de execução de seu programa sequencial com a versão paralela. Houve ganho de desempenho?

Exercício: Tente aumentar o número de threads do OpenMP para o dobro do número de processadores disponíveis. Melhorou o desempenho?