



Prof: Nilton Luiz Queiroz Junior
Disciplina: Programação Concorrente (5205 – Turma 1)

Segundo Trabalho Prático

Objetivo: O trabalho tem como objetivo a implementação de uma aplicação paralela, utilizando troca de mensagens e a elaboração de um relatório, fazendo análises de desempenho da aplicação implementada.

Instruções:

- O trabalho poderá ser feito em grupos de 1 a 3 membros;
- A implementação da aplicação deverá ser feita na linguagem C/C++ utilizando a biblioteca *MPI*;
- O trabalho tem valor de 0 a 10 e corresponde à segunda avaliação periódica da disciplina;
- Deverão ser entregues os seguintes itens:
 1. O código fonte do trabalho;
 2. Um passo a passo de como compilar e executar o programa;
 - Como deve ser fornecida a entrada para o programa, qual sistema operacional a ser usado, e caso seja usado algum outro recurso além da biblioteca MPI, informar qual recurso foi usado, como instalar e como configurar, etc;
 3. Um relatório, de 4 a 6 páginas, fazendo análise de desempenho composto por:
 - Título;
 - Resumo;
 - Introdução;
 - Referencial teórico;
 - Descrever superficialmente técnicas que foram usadas na paralelização do algoritmo:
 - Por exemplo:
 - Descrever os conceitos de memória distribuída, mecanismos de sincronização utilizados, etc;
 - Desenvolvimento;
 - Explicar como funciona a aplicação e como ela foi paralelizada;
 - Se foi usado algum paradigma de interação entre processos dentro os vistos em sala;
 - Ambiente experimental e experimentos realizados;
 - Descrever hardware e sistema operacional utilizado;
 - Descrever detalhes dos experimentos que foram realizados;
 - Tamanho das entradas, quantidade de processos, parâmetros do algoritmo, quantidade de processadores, etc;
 - Análise e discussão dos resultados;
 - Extrair métricas tais como speedup, entre outras vistas em sala;
 - Se desejável colocar gráficos (ou tabelas) com speedup por quantidade de processos;



Prof: Nilton Luiz Queiroz Junior
Disciplina: Programação Concorrente (5205 – Turma 1)

- Caso os experimentos sejam realizados em mais de um processador, descrever também os efeitos do uso de mais processadores no desempenho da aplicação;
- Conclusões;
- Os itens acima deverão ser entregues via moodle, em um arquivo compactado que deve seguir o padrão:
 - NomeAluno1_RAXXXXXX_NomeAluno2_RAXXXXXX_NomeAluno3_RAXXXXXX.zip;
 - O formato do arquivo para submissão poderá ser .ZIP, .RAR ou .TAR
- A data limite para a entrega do trabalho é dia 19/12/2017;

Descrição: Neste trabalho deverá ser implementada uma versão simplificada do algoritmo K-NN, fazendo a paralelização dos cálculos realizados sobre os dados da base de conhecimento, onde a classe será dada somente pelo vizinho mais próximo, ou seja, deverá ser implementado o algoritmo K-NN para K=1. Além disso, não é necessário fazer a retenção do novo conhecimento obtido. Deverão ser usadas as bases de treino e de teste que foram usadas no primeiro trabalho. E assim como no primeiro trabalho, se o grupo desejar poderá avaliar a taxa de acerto do método na tarefa de classificação poderá usar as classes do arquivo teste, porém, assim como no primeiro trabalho, o objetivo é calcular o speedup que se pode alcançar no método.

Algumas observações sobre a implementação:

- Todos os processos poderão ter acesso aos arquivos de treino, porém o arquivo de teste deverá ser acessado somente por um processo;
 - A ideia é que o processo que lê o arquivo de teste mande o exemplo para todos os outros processos;
- A distribuição de tarefas entre os processos pode ser estática ou dinâmica, isso fica a critério do grupo.

Problemas com Trabalhos COPIADOS:

Quem copiar terá o trabalho anulado (zerado), seja de outra dupla ou da internet.
Quem fornecer a cópia também terá o trabalho anulado (zerado).