

HIGH PERFORMANCE PROGRAMMING: MULTICORE, CLUSTERS E GPU

Avaliação dos trabalhos de implementação

A atividade avaliativa da disciplina consiste em paralelizar (escrever uma versão para execução em paralelo) do problema passado no semestre usando a biblioteca OpenMP, MPI ou Cuda. Deve-se marcar o tempo no algoritmos sequencial e no paralelo no mesmo ponto de processamento. Usar os dados de entrada do problema passado.

O que deve ser entregue de cada trabalho solicitado:

- O(s) arquivo(s) fontes da solução (arquivos-fonte makefile, README, ...)
- Um relatório impresso.

Itens a serem abordados no relatório. Pode ser entendido como seções do relatório

- Identificação do(s) responsável(eis);
- Descrição de forma sucinta de como o trabalho foi implementado;
- Dificuldades encontradas (se houverem);
- Recursos de programação paralela utilizados e o motivo da escolha;
- Análise dos resultados dos testes;
- Considerações finais.

Dicas para o relatório

- O relatório deve conter apenas uma página. Pode-se usar uma segunda página para anexos. Caso seja necessário, pode-se usar texto em dupla coluna;
- A formatação deve ser: folha A4, margens de 1,5 cm, espaçamento simples, fonte Times New Roman, tamanho 12 (em negrito nos títulos). O título principal deve ser em tamanho 14, negrito. Abaixo dele o(s) nome(s) do(s) aluno(s), em seguida o endereço de contato;
- É livre o uso de figuras e gráficos quando necessário;
- Tentar não repetir o enunciado, optando por descrição com detalhes técnicos;
- Os resultados intermediários (se houverem) devem ser entregues em folha anexa e se possível discutidos.

Observações referentes ao programa:

- Deve-se produzir apenas um código que será executado para qualquer número de processos/threads;
- Caso se tente executar com um número de processos/threads diferente do enunciado, o programa deve mostrar uma mensagem (apenas uma) de erro e não executar;
- Usar no máximo 4 núcleos de cada máquina para a obtenção de resultados e cada execução paralela deve usar o menor número de máquinas físicas possível;
- O código deve estar funcional para 2, 4, 8 ou 16 threads no caso de OpenMP. No caso de uso do MPI com 2, 4, 8 16 e 32 processos no caso de MPI. O uso em CUDA é de livre escolha dos alunos as configurações a serem testada;
- Pode-se mudar a forma de implementação do algoritmo sequencial passado, porém não é permitida modificação na complexidade ou no algoritmo usado.