

Universidade Federal do ABC
Prova de MCZA020-13— Programação Paralela
Prova 1

Turma DAMCZA020-13SA
 2018.Q3

Emilio Francesquini

Nome:

RA:

Questão	Pontos	Nota
1	2½	
2	2½	
3	2	
4	3	
5	1	
Total:	11	

- A prova tem duração de **110 minutos**
- Não esqueça de preencher a caneta **todos** os campos na primeira página da folha de prova e seu nome e RA nesta folha de questões.
- Antes da resposta da questão i escreva “Questão i” em letras **garrafais**.
- As respostas às questões podem ser deixadas a lápis.
- Ao final da prova entregue tanto a folha de questões quanto a folha de prova.
- Em caso de fraude, **TODOS** os envolvidos:
 - **Receberão conceito final F (reprovado) na disciplina**
 - Serão **denunciados** à Comissão de Transgressões Disciplinares Discentes da Graduação e à Comissão de Ética da UFABC cuja punição pode resultar em **advertência, suspensão ou desligamento**, de acordo com os artigos 78-82 do Regimento Geral da UFABC e do artigo 25 do Código de Ética da UFABC.

Boa Prova!

Questão 1. Considere o computador que você utiliza no dia a dia. A hierarquia de memória do seu computador muito provavelmente é composta por alguns níveis de cache (L1, L2 e talvez L3) e pela memória principal (RAM).

- (a) (1½ pontos) Projete e descreva um experimento (isto é, elabore e detalhe os programas que você criaria e as suas utilizações) para determinar a largura de banda da conexão entre o seu processador e sua memória principal. Não se esqueça de descrever como os experimentos seriam executados e de justificar porque seu experimento funciona.
- (b) (1 ponto) Como o seu experimento poderia ser adaptado para, além de ser capaz de determinar a largura de banda entre o processador e a memória, determinar qual o tamanho da memória cache em cada um dos níveis da hierarquia?

Dica: Para testar a largura de banda entre o processador e a memória você não quer que cache interfira nas suas medições, ou seja, você não quer que haja reutilização de linhas de cache. Para testar os tamanhos das caches, você quer que haja reuso e,

conforme a quantidade de dados for aumentando, você vai perceber o reuso diminuir drasticamente (e portanto o desempenho também).

Questão 2. A lei de Amdahl pode ser aplicada em diversos contextos além do processamento paralelo. Suponha que uma aplicação seja composta por: 20% de operações com pontos flutuantes; e 80% de outras operações (estes percentuais se referem ao número de operações e não ao tempo de execução). O tempo de execução de uma operação de ponto flutuante em um processador é três vezes maior do que as demais operações. Está sob consideração pelos usuários da aplicação a compra de um novo processador com uma versão mais otimizada da unidade de ponto flutuante (todo o resto do processador permanece o mesmo) na esperança de acelerar a execução.

- (1 ponto) Formule uma versão mais geral da lei de Amdahl em termos do percentual do número de operações de uma aplicação (e não em termos de capacidade de processamento paralelo).
- (1 ponto) Quão mais rápida a unidade de ponto flutuante deste novo processador deverá ser para assegurar um aumento de desempenho de 25% no tempo de execução da aplicação?
- ($\frac{1}{2}$ ponto) Qual é o *speedup* máximo que podemos obter na aplicação apenas pela substituição da unidade de ponto flutuante?

Questão 3. (2 pontos) Em muitos casos o tempo de execução da comunicação ponto-a-ponto entre dois processadores pode ser, de maneira simplificada, descrita pela seguinte função linear:

$$T(s) = \tau + t_c \cdot s$$

Onde s é o tamanho da mensagem, τ é o tempo de inicialização (que é independente do tamanho da mensagem) e t_c é o inverso da largura de banda.

Sabendo deste fato, descreva detalhadamente como, experimentalmente, você poderia determinar os valores de τ e t_c em um computador paralelo.

Questão 4. Hipercubo.

- (2 pontos) Escreva um programa em MPI (pseudo-código, porém completo) que implementa a operação de *broadcast* apenas utilizando as funções `MPI_Send()` e `MPI_Recv()`. O programa deve usar $p = 2^k$ processadores e estes processadores devem estar organizados em uma rede com topologia de hipercubo (ou seja, todas as comunicações entre os processadores devem seguir esta topologia).
- ($\frac{1}{2}$ ponto) Quantas mensagens são enviadas por cada processador (suponha que o processador que inicia a comunicação é o 0) como resultado da chamada à função *broadcast* implementada acima?
- ($\frac{1}{2}$ ponto) Se cada comunicação leva tempo τ , quanto tempo leva desde o início do *broadcast* até o último processador receber a mensagem?

Questão 5. (1 ponto) Comente, critique, reclame, e/ou elogie a disciplina, o material, o projeto e/ou o professor. Caso deseje, sugira mudanças que você gostaria de ver na segunda metade do quadrimestre.