

[Painel do utilizador](#) / [Minhas disciplinas](#) / [CPeD-LEI](#) / [Aula 04 - 25/03 e 29/03](#) / [TPC04 - Até Modelos de Programação Paralela Parte 1](#)

Iniciada	quarta-feira, 30 de março de 2022 às 22:22
Estado	Terminada
Terminada	quarta-feira, 30 de março de 2022 às 22:44
Tempo gasto	21 minutos 51 segundos
Nota	12,6/18,0
Nota	14,0 num máximo de 20,0 (70%)

Pergunta **1**

Correta

Nota: 1,0 em 1,0

Qual a ordem correta desde o nível de abstração mais baixo até ao mais elevado?

- a. Modelos de máquinas paralelas, Modelos computacionais paralelos, Modelos de arquiteturas paralelas, Modelos de programação paralela
- b. Modelos de programação paralela, Modelos computacionais paralelos, Modelos de arquiteturas paralelas, Modelos de máquinas paralelas
- c. Modelos computacionais paralelos, Modelos de programação paralela, Modelos de máquinas paralelas, Modelos de arquiteturas paralelas
- d. Modelos de máquinas paralelas, Modelos de arquiteturas paralelas, Modelos computacionais paralelos, Modelos de programação paralela ✓
- e. Modelos de programação paralela, Modelos de máquinas paralelas, Modelos computacionais paralelos, Modelos de arquiteturas paralelas

Pergunta **2**

Parcialmente correta

Nota: 0,5 em 1,0

Sobre **paralelização de programas**, faça a correspondência adequada:

Escalonamento	Decomposição das computações	✗
Load balancing	Atribuição das tarefas aos processos	✓
Minimizar <i>overheads</i> tais como custos de comunicação	Orquestração e Mapeamento dos processos aos cores	✓
Especificação das dependências das tarefas	Atribuição das tarefas aos processos	✗

Pergunta **3**

Correta

Nota: 1,0 em 1,0

Sobre *loops* paralelos, faça a correspondência adequada:

As instruções sem inter-dependência são separadas e podem ser paralelizadas

Loop distribuído



Todas as instruções podem ser executadas paralelamente

Paralelismo DOALL

Pergunta **4**

Correta

Nota: 1,0 em 1,0

Nos processadores multi-núcleo com desenho hierárquico, os cores partilham diversas caches.

Selecione uma opção:

- Verdadeiro
- Falso

Pergunta **5**

Correta

Nota: 1,0 em 1,0

O uso nas CPUs de palavras de instrução muito longas, ou VLIW, como um mecanismo de melhoria do desempenho através do processamento paralelo de instruções, requer que os compiladores reconheçam quais instruções são independentes umas das outras e as organize na *instruction word*.

Selecione uma opção:

- Verdadeiro
- Falso

Pergunta **6**

Correta

Nota: 1,0 em 1,0

Assinale a alternativa menos correta:

- a. O processamento de uma instrução não é afetado pelo tamanho do chip da CPU
- b. Devido à crescente disparidade entre a redução do tempo de acesso à RAM e o aumento da frequência de relógio da CPU, o tempo de acesso à RAM contado em ciclos de relógio aumentou entre 1990 e 2003 mais de 30 vezes
- c. A distância entre os componentes de um chip é uma preocupação dos projetistas
- d. O aumento do número de transístores por área de um chip provoca maior desperdício de energia
- e. O *overclock* pode causar que uma instrução demore mais ciclos de relógio para executar

Pergunta **7**

Incorreta

Nota: 0,0 em 1,0

A integração de caches no chip da CPU não melhora o problema da barreira de memória (*Memory Wall*), porque continua sendo memória com desvantagem no tempo de acesso em relação à velocidade da CPU.

Selecione uma opção:

- Verdadeiro ✘
- Falso

Pergunta **8**

Correta

Nota: 1,0 em 1,0

A granularidade G de uma *task* é definida como a razão entre tempo de computação (uso da CPU) e tempo de comunicação (*IO*):

$$G = \frac{T_{\text{computação}}}{T_{\text{comunicação}}}$$

As *tasks* podem então ser classificadas, umas em relação às outras, como *fine-grained*, *medium-grained*, ou *coarse-grained*, conforme esta relação vá do valor mais pequeno para o maior. Assim, quanto maior a necessidade de comunicação, mais "fina" a granularidade.

Considere as 3 hipóteses de processamento de uma imagem com 100 pixels:

1. Usamos 100 unidades de processamento, cada uma processando 1 pixel
2. Usamos 50 unidades de processamento, cada uma processando 2 pixels
3. Usamos 2 unidades de processamento, cada uma processando 50 pixels

Faça a associação correta de cada hipótese com o nível de granularidade correspondente:

- | | | |
|------------|----------------|---|
| Hipótese 3 | Coarse-grained | ✓ |
| Hipótese 1 | Fine-grained | ✓ |
| Hipótese 2 | Medium-grained | ✓ |

Pergunta **9**

Correta

Nota: 1,0 em 1,0

Sobre a forma como as *tasks* de uma computação decomposta se comunicam, faça a associação correta:

- | | | |
|---------------------------------------|-----------------------|---|
| Sistemas de <i>distributed memory</i> | Comunicação pela rede | ✓ |
| Sistemas de <i>shared memory</i> | Variáveis na memória | ✓ |

Pergunta **10**

Incorreta

Nota: 0,0 em 1,0

Sobre o *hyperthreading*, assinale a alternativa menos correta:

- a. Aumenta a utilização dos recursos de processamento
- b. A melhoria no tempo de execução depende da arquitetura da CPU e independe das aplicações
- c. Bloqueios num processador lógico podem ser devidos a: *cache-misses*, predição errada para *branches*, dependências entre instruções, *hazards* de *pipelining*
- d. O *hyperthreading* pode melhorar o tempo de processamento de uma aplicação em 15% ou mais ✘
- e. Bloqueios num processador lógico permitem que outro processador lógico utilize os recursos de processamento

Pergunta **11**

Correta

Nota: 1,0 em 1,0

Na paralelização de ciclos *for*, a complexidade resultante (e portanto o ganho de eficiência) é afetada pela interdependência das operações realizadas em cada iteração.

Selecione uma opção:

- Verdadeiro ✔
- Falso

Pergunta **12**

Parcialmente correta

Nota: 0,7 em 1,0

Sobre a criação de *threads*, faça a associação correta:

Um *thread* existente lança um novo *thread*

Criação de um *pool* de *threads* no início da execução

Parbegin-Parend

Modo assíncrono

Um novo *thread* é criado cada vez que um utilizador se liga ao servidor

O processamento aguarda que um *thread* termine para continuar as computações

Modo síncrono

Fork	✔
Criação estática	✔
Fork	✘
SPMD	✔
Criação dinâmica	✔
Criação dinâmica	✘
SIMD	✔

Pergunta **13**

Incorreta

Nota: 0,0 em 1,0

Se tem um programa *CPU-bound* e vai utilizar a biblioteca *multiprocessing*, quantos processos deve lançar?

- a. 1000
- b. Não devo usar o *multiprocessing*, e sim *threads* com um programa *CPU-bound*.
- c. Tantos quanto o número de cores de CPUs do computador
- d. O máximo possível
- e. 1

Pergunta **14**

Correta

Nota: 1,0 em 1,0

Sobre programação distribuída e programação paralela, faça a correspondência adequada:

Geralmente usa sistemas operativos diferentes

Programação distribuída ✓

A meta é processar o problema total o mais rápido possível

Programação paralela ✓

Um problema é particionado por vários processadores

Programação paralela ✓

Geralmente usa sistemas operativos iguais

Programação paralela ✓

Sistemas autónomos diferentes

Programação distribuída ✓

Usa servidores remotos que podem por sua vez dividir o trabalho para vários processadores

Programação distribuída ✓

Pergunta **15**

Incorreta

Nota: 0,0 em 1,0

Assinale a alternativa menos correta relativamente a sistemas multi-processadores com espaço de endereçamento partilhado:

- a. A distinção entre acesso à memória global e acesso à memória local é implícita ao modelo utilizado
- b. Os processadores conectam-se à memória global através de uma rede
- c. As operações de sincronização são implícitas ao modelo utilizado
- d. Usa variáveis partilhadas para troca de dados através de operações coordenadas de leitura e escrita
- e. Há uma memória global e podem haver várias memórias locais a cada um dos processadores



Pergunta **16**

Correta

Nota: 1,0 em 1,0

A computação paralela comercial existe desde o início da história da computação.

Selecione uma opção:

- Verdadeiro
- Falso ✓

Pergunta **17**

Parcialmente correta

Nota: 0,4 em 1,0

Sobre memória, faça a correspondência adequada:

Maior área por kilobyte	DRAM	✗
RAM	SRAM	✗
Menor tempo de acesso	DRAM	✗
Cache	SRAM	✓
Menor custo por kilobyte	DRAM	✓

Pergunta **18**

Correta

Nota: 1,0 em 1,0

Sobre a **granularidade** das aplicações de computação paralela, faça a correspondência adequada:

Ao aumentar o tamanho da palavra de memória, o processamento é realizado com menos instruções

Paralelismo bit-level



O servidor web faz multi-threading e usa cores físicos diferentes para atender diferentes utilizadores

Paralelismo task-level



Numa multiplicação de matrizes, divide-se as matrizes em blocos menores que são multiplicados em paralelo

Paralelismo instruction-level



[← Auto-avaliação sobre Modelos e Bibliotecas de Programação Paralela Parte 1 \(cópia\)](#)

Ir para...

[Slides da disciplina Computação de Alto Desempenho sobre Modelos de Computação Paralela ►](#)