



INSTITUTO NACIONAL DE PESQUISA ESPACIAL – INPE

PROVA OBJETIVA

TG13

AMBIENTE DE SUPERCOMPUTAÇÃO E HPC (HIGH PERFORMANCE COMPUTING)



SUA PROVA

- Além deste caderno contendo **45 (quarenta e cinco)** questões objetivas, você receberá do fiscal de prova o cartão de respostas;
- As questões objetivas têm **5 (cinco)** opções de resposta (A, B, C, D e E) e somente uma delas está correta.



TEMPO

- Você dispõe de **4 (quatro) horas** para a realização da prova;
- **2 (duas) horas** após o início da prova, é possível retirar-se da sala, sem levar o caderno de questões;
- A partir dos **30 (trinta) minutos** anteriores ao término da prova é possível retirar-se da sala **levando o caderno de questões**.



NÃO SERÁ PERMITIDO

- Qualquer tipo de comunicação entre os candidatos durante a aplicação da prova;
- Anotar informações relativas às respostas em qualquer outro meio que não seja no caderno de questões e nas folhas de textos definitivos;
- Levantar da cadeira sem autorização do fiscal de sala;
- Usar o sanitário ao término da prova, após deixar a sala.



INFORMAÇÕES GERAIS

- Verifique se seu caderno de questões está completo, sem repetição de questões ou falhas e também confira seu cargo. Caso tenha recebido caderno de cargo **diferente** do impresso em seu cartão de respostas, o fiscal deve ser **obrigatoriamente** informado para o devido registro na ata da sala;
- Confira seus dados pessoais, especialmente nome, número de inscrição e documento de identidade e leia atentamente as instruções para preencher o cartão de respostas;
- Para o preenchimento do cartão de respostas, use somente caneta esferográfica, fabricada em material transparente, com tinta preta ou azul;
- Assine seu nome apenas no(s) espaço(s) reservado(s) no cartão de respostas;
- Reserve tempo suficiente para o preenchimento do seu cartão de respostas. O preenchimento é de sua responsabilidade e **não será permitida a troca do cartão de respostas em caso de erro cometido pelo candidato**;
- Para fins de avaliação, serão levadas em consideração apenas as marcações realizadas no cartão de respostas;
- A FGV coletará as impressões digitais dos candidatos na lista de presença;
- Os candidatos serão submetidos ao sistema de detecção de metais quando do ingresso e da saída de sanitários durante a realização das provas.

Boa Prova!

CONHECIMENTOS ESPECÍFICOS

1

A computação em nuvem proporciona aos usuários acesso a uma capacidade computacional que se adapta à sua demanda, frequentemente superando o que seria economicamente viável adquirir de forma local.

Uma das características que viabiliza a computação em nuvem consiste na abstração dos recursos computacionais físicos dos lógicos.

Essa característica pode ser denominada

- (A) elasticidade.
- (B) redundância.
- (C) escalabilidade.
- (D) virtualização.
- (E) descentralização.

2

A computação em nuvem oferece diversos modelos de serviços.

Nesse contexto, analise as afirmativas a seguir.

- I. No modelo IaaS, é possível oferecer ao usuário um serviço para desenvolvimento de aplicativos de modo simplificado, a partir de blocos pré-determinados.
- II. No modelo SaaS, os provedores do serviço muitas vezes são os próprios desenvolvedores, o que facilita a customização do aplicativo.
- III. Um serviço no modelo PaaS pode ser utilizado para oferecer um serviço do tipo SaaS, ao passo que um serviço no modelo PaaS pode ser desenvolvido através de um serviço do tipo IaaS.

Está correto o que se afirma em

- (A) I, apenas.
- (B) III, apenas.
- (C) I e II, apenas.
- (D) II e III, apenas.
- (E) I, II e III.

3

Uma rede de dados para computação de alto desempenho (em inglês *High Performance Computing* ou HPC) é uma infraestrutura projetada para as demandas de comunicação dos dados de sistemas de computação de alto desempenho, tendo, portanto, características muito peculiares.

Considerando essas características, o componente que podemos citar como fundamental para o desempenho de uma rede de dados HPC é a(o)

- (A) velocidade do processador nos nós de computação.
- (B) largura de banda da rede.
- (C) tipo de gabinete dos servidores.
- (D) capacidade de armazenamento dos nós de computação.
- (E) equilíbrio entre as métricas de *precision* e *recall*.

4

Uma rede de dados para HPC (*High Performance Computing*) possui alguns requisitos mais exigentes em relação às redes convencionais. Tais requisitos tem reflexos nas configurações das redes de dados HPC. Nesse contexto, assinale a opção que descreve corretamente uma caracterização da rede de dados para HPC.

- (A) A largura de banda da rede não é relevante, desde que os nós de computação sejam poderosos o suficiente.
- (B) O tipo de cabo de rede não afeta significativamente o desempenho da rede.
- (C) É crucial considerar a topologia da rede para minimizar a latência e maximizar o *throughput*.
- (D) A segurança da rede não é uma preocupação em ambientes HPC.
- (E) O tipo de disco utilizado para armazenamento não é relevante.

5

Armazenamento de alta performance pode ser uma exigência quando tratamos de alguns sistemas para *Big Data*, Inteligência Artificial e Computação de Alto Desempenho (HPC). Esse tipo de armazenamento possui algumas características particulares, que o distingue dos demais tipos de armazenamento.

Diante desse contexto, assinale a opção que apresenta um benefício chave dos sistemas de armazenamento de dados de alta performance.

- (A) Baixa capacidade de armazenamento, pois com um menor volume de dados o desempenho será privilegiado.
- (B) Latência alta, pois assim haverá mais tempo para o sincronismo entre as transmissões dos dados.
- (C) Alta taxa de transferência de dados, ou seja, uma grande quantidade de dados pode ser transmitida ou recebida em um curto espaço de tempo.
- (D) Limitações de escalabilidade, o que faz com que o sistema tenha que lidar com uma menor quantidade de nós.
- (E) Falta de redundância de dados, fazendo com que o gerenciador do sistema não precise se preocupar com o cálculo de tais redundâncias.

6

Para avaliar e medir o desempenho em computação de alto desempenho (HPC), uma variedade de métricas e benchmarks são utilizados para caracterizar diferentes aspectos do sistema. Entre as métricas de HPC, *speedup* pode ser considerada uma das mais importantes.

No contexto de métricas e análises de desempenho para HPC, assinale a opção que define corretamente *speedup*.

- (A) O tempo necessário para um programa paralelo ser executado em um único processador.
- (B) A medida do tempo de execução de um programa sequencial em comparação com o tempo de execução de sua versão paralela.
- (C) A medida do tempo de execução de um programa paralelo em comparação com o tempo de execução de outro programa paralelo.
- (D) O número de *threads* ou processos que um programa paralelo pode executar simultaneamente.
- (E) O tempo necessário para inicializar um programa paralelo.

7

As *threads* e os processos são conceitos fundamentais em sistemas operacionais e programação concorrente.

Assinale a opção que descreve uma diferença importante no conceito entre processos e *threads*, incluindo aspectos de comunicação e sincronização.

- (A) *Threads* podem compartilhar recursos e memória, enquanto processos têm seus próprios espaços de endereço separados.
- (B) Processos são mais eficientes em termos de comunicação e sincronização do que *threads*.
- (C) *Threads* podem ser distribuídas em diferentes computadores, enquanto processos estão limitados a um único computador.
- (D) Processos podem executar em paralelo em diferentes núcleos de CPU, enquanto *threads* são sempre executadas sequencialmente em um único núcleo.
- (E) *Threads* podem iniciar novos *threads*, enquanto processos não podem iniciar novos processos.

8

A sincronização entre processos é um conceito fundamental em sistemas operacionais e programação concorrente. Assinale a opção que descreve um mecanismo comum de sincronização entre processos em sistemas operacionais.

- (A) Compartilhamento de memória.
- (B) Escalonamento preemptivo.
- (C) Barreiras.
- (D) Filas de mensagens.
- (E) Monitoramento de recursos.

9

Na computação paralela, onde várias tarefas ou processos são executados simultaneamente em diferentes núcleos de processador, a sincronização entre processos é essencial para garantir que essas tarefas cooperem e compartilhem recursos de forma eficiente e correta.

No contexto de um sistema de computação paralela, assinale a opção que apresenta um método de sincronização entre processos que garante que apenas um processo possa acessar uma seção crítica de código por vez.

- (A) Semáforo binário.
- (B) Barreiras.
- (C) Mutex.
- (D) Memória compartilhada.
- (E) Filas de mensagens.

Atenção: o enunciado a seguir refere-se às duas próximas questões.

OpenMP é um ambiente de programação paralela adequado para escrever programas paralelos a serem executados em sistemas de memória compartilhada. É uma coleção de interfaces de programação de aplicativos que inclui diretivas de compilador, funções de suporte e variáveis de ambiente, permitindo que os programadores explorem e controlem o paralelismo durante a execução de um programa.

10

No contexto das variáveis utilizadas em programação paralela com *OpenMP*, analise as afirmativas a seguir e assinale (V) para a verdadeira e (F) para a falsa.

- () A cláusula *shared(list)* especifica que cada variável da lista é compartilhada por todas *threads* do time, isso é, todas as *threads* compartilham uma mesma cópia da variável.
- () Por padrão, variáveis declaradas antes de um bloco paralelo são compartilhadas.
- () A cláusula *private(list)* especifica que cada variável da lista é privada por todas *threads* do time, isso é, cada *thread* tem sua própria cópia da variável.
- () Por padrão, variáveis declaradas dentro de um bloco paralelo são privadas.

As afirmativas são, respectivamente,

- (A) V – V – V – V.
- (B) F – V – F – V.
- (C) V – F – V – F.
- (D) V – V – F – V.
- (E) F – F – F – F.

11

Com isto em mente, a cláusula que especifica o número de *threads* que devem ser executadas em um bloco estruturado em paralelo é

- (A) `omp_get_thread_num()`.
- (B) `collapse(n)`.
- (C) `omp_get_num_threads()`.
- (D) `reduction(operator:variable)`.
- (E) `num_threads`.

12

Em computação paralela os loops paralelos são muito úteis, permitindo que várias iterações de um loop sejam executadas simultaneamente por diferentes *threads*, aproveitando assim ao máximo os recursos de processamento disponíveis em sistemas paralelos. Isso é especialmente importante para acelerar o processamento de tarefas computacionais intensivas, dividindo o trabalho entre múltiplos núcleos de processamento.

Neste contexto, assinale a opção que apresenta o argumento utilizado no padrão *OpenMP* para combinar múltiplos *loops* em um único, permitindo sua execução em paralelo.

- (A) `reduction(op:var)`.
- (B) `collapse(n)`.
- (C) `nowait`.
- (D) `final`.
- (E) `critical`.

13

No contexto da programação paralela com OpenMP, as variáveis declaradas fora de um bloco paralelo são compartilhadas automaticamente, enquanto as variáveis declaradas dentro de um bloco paralelo são tratadas como privadas por padrão. Também existe a possibilidade de uso de cláusulas adequadas para garantir o correto compartilhamento ou privacidade das variáveis entre as threads paralelas.

A esse respeito, analise as afirmativas a seguir e assinale (V) para verdadeira e (F) para a falsa.

- () A cláusula *shared(list)* especifica que cada variável da lista é compartilhada por todas threads do time, isso é, cada thread tem sua própria cópia da variável.
- () A cláusula *firstprivate(list)* especifica que cada variável da lista é privada por todas threads do time, mas são inicializadas com o valor que continha no momento em que a região paralela foi encontrada.
- () A cláusula *private(list)* especifica que cada variável da lista é privada por todas threads do time, isso é, todas as threads compartilham uma mesma cópia da variável.

As afirmativas são, respectivamente,

- (A) V – V – V.
- (B) F – F – F.
- (C) V – V – F.
- (D) F – V – F.
- (E) V – F – V.

14

Sistemas paralelos e distribuídos desempenham um papel crucial na computação moderna, oferecendo soluções para lidar com desafios cada vez mais complexos e demandas crescentes por desempenho, escalabilidade e confiabilidade. A importância desses sistemas reside em sua capacidade de processar grandes volumes de dados e executar tarefas computacionais intensivas de forma eficiente e rápida. Existem várias soluções que evoluíram até os sistemas paralelos modernos, que podem ser resumidos em três tipos predominantes:

- I. Sistemas de memória compartilhada.
- II. Sistemas distribuídos.
- III. Sistemas de tempo real.
- IV. Unidades de processamento gráfico (GPU).
- V. Sistemas embarcados.

Está correto o que se afirma em

- (A) I, II e III, apenas.
- (B) II, III e IV, apenas.
- (C) I, III e V, apenas.
- (D) I, II e IV, apenas.
- (E) II, IV e V, apenas.

15

A programação com memória compartilhada refere-se a um modelo de programação em que múltiplos threads ou processos compartilham o mesmo espaço de memória, permitindo que eles acessem e modifiquem os mesmos dados. Isso geralmente é implementado em sistemas *multicore* ou multiprocessadores, onde várias unidades de processamento têm acesso simultâneo à memória principal.

Para evitar condições de corrida, isto é, uma situação em que o resultado depende do tempo preciso de acessos de leitura e escrita ao mesmo local na memória principal, ao trabalhar com memória compartilhada, podem ser utilizados mecanismos de exclusão mútua, como o uso de

- (A) semáforos ou *versioning*.
- (B) transações ou algoritmos *wait-free*.
- (C) semáforos ou acesso atômico.
- (D) *lazy synchronization* ou transações.
- (E) acesso atômico ou algoritmos *wait-free*.

16

Um modelo multiprocessado é um modelo de computação paralela que se baseia no modelo de máquina de acesso aleatório e generaliza o mesmo. Existem basicamente três tipos diferentes de modelos multiprocessado e cada um desses modelos possui um número de unidades de processamento p e difere na organização das memórias e na forma como as unidades de processamento acessam essas memórias.

Assinale a opção que indica o modelo que consiste em unidades de processamento e módulos de memória acessados por uma rede de interconexão comum, sem memórias locais, permitindo o acesso uniforme às memórias quando não há acessos coincidentes.

- (A) DRAM.
- (B) PRAM.
- (C) LMM.
- (D) SAN.
- (E) MMM.

17

Sistemas paralelos e distribuídos desempenham um papel crucial na computação moderna, oferecendo soluções para lidar com desafios cada vez mais complexos e demandas crescentes por desempenho, escalabilidade e confiabilidade.

Sobre o uso de memória distribuída ou compartilhada, analise as afirmativas a seguir e assinale (V) para a verdadeira e (F) para a falsa.

- () Em relação ao tempo da comunicação, as memórias distribuídas apresentam vantagem majoritariamente.
- () Em relação à um número crescente de processadores que podem cooperar eficientemente, os computadores de memória distribuída apresentam mais vantagens.
- () Quando o número de processadores deve ser alto (mais que 8) para reduzir o tempo de execução, a velocidade da comunicação se torna um fator crucial para a performance.
- () *OpenCL* é uma biblioteca de especificação de passagem de mensagem muito útil para a implementação de paralelismo.

As afirmativas são, respectivamente,

- (A) V – F – V – F.
- (B) F – V – V – F.
- (C) V – V – F – V.
- (D) V – F – F – V.
- (E) F – V – V – V.

18

Leia o trecho a seguir.

É uma linguagem de programação paralela de plataforma cruzada usada para programar diferentes tipos de processadores em computadores pessoais, servidores, dispositivos móveis e plataformas embarcadas. Ela define uma linguagem de programação e uma interface de programação de aplicativos para controlar a plataforma e executar programas em dispositivos de computação, como CPUs e GPUs.

Assinale a opção que indica a tecnologia descrita no trecho acima.

- (A) OpenCL.
- (B) MPI.
- (C) Fortran.
- (D) OpenMP.
- (E) Assembly.

19

É construída com células que armazenam dados como carga em capacitores. A presença ou ausência de carga em um capacitor é interpretada como um binário 0 ou 1. Como os capacitores possuem uma tendência natural para descarregar, esta memória exige atualização (refresh) periódica para manter o dado armazenado.

Assinale a opção que apresenta o tipo de memória que possui as características descritas no trecho acima.

- (A) DRAM.
- (B) ROM.
- (C) Flash.
- (D) EEPROM.
- (E) CMOS.

20

Um sistema de memória semicondutora é suscetível a falhas, as quais podem ser classificadas como erros permanentes ou não permanentes. Entretanto, a maioria dos sistemas de memória modernos dispõe de lógica para detecção e correção de erros, sendo o código de Hamming um dos exemplos mais simples para esse fim. A respeito do Código de Hamming, analise as afirmativas a seguir.

- I. K bits de paridade são acrescentados a uma palavra de M bits, gerando uma nova palavra de comprimento $M + K$ bits.
- II. Todos os bits de verificação ocupam posições que são potência de 2 na palavra. Os bits de dados são encontrados nas demais posições.
- III. Os bits de verificação são calculados por meio do emprego da operação lógica AND.

Está correto o que se afirma em

- (A) I, apenas.
- (B) I e II, apenas.
- (C) I e III, apenas.
- (D) II e III, apenas.
- (E) I, II e III.

21

Um tecnólogo de alto desempenho pode ser designado para atuar na manutenção e melhoria do armazenamento de dados no INPE. Uma das técnicas que pode ser empregada para essa missão é o RAID (*Redundant Array of Independent Disks*), a qual está relacionada à redundância, tolerância a falhas, aprimoramento do desempenho, escalabilidade, integridade dos dados, dentre outros.

Diante desse cenário, assinale a opção que corresponde ao nível de RAID que apresenta como características a execução e armazenamento de dois cálculos de paridade distintos em blocos separados em discos diferentes e é o mais indicado para aplicações de tarefas críticas.

- (A) RAID 0.
- (B) RAID 1.
- (C) RAID 3.
- (D) RAID 5.
- (E) RAID 6.

22

Com relação aos modos de endereçamento de memória principal, cache e registradores presentes em um computador, analise as afirmativas a seguir e assinale (V) para a verdadeira e (F) para a falsa.

- () O endereçamento direto é a forma mais simples de uma instrução especificar um operando, uma vez que a parte da instrução referente ao endereço contém o operando propriamente dito ao invés de seu endereço.
- () No endereçamento imediato, a maneira para determinar um operando na memória é fornecer seu endereço completo.
- () Referenciar a memória sem a necessidade de ter um endereço de memória completo na instrução é uma possibilidade gerada pelo endereçamento indireto de registrador.

As afirmativas são, respectivamente,

- (A) F – F – F.
- (B) V – F – V.
- (C) F – F – V.
- (D) F – V – V.
- (E) V – V – F.

23

Leia o trecho a seguir.

Existe uma estratégia, sobretudo para sistemas operacionais antigos, para subdividir a memória virtual endereçável, permitindo que a mesma seja vista pelo programador como espaços múltiplos de endereços ou blocos, os quais possuem tamanho variável, proporcionando maior comodidade para organizar programas e dados. Além disso, apresenta como características a simplificação do tratamento de estruturas de dados dinâmicas, do compartilhamento de processos e permite melhor gestão da segurança, uma vez que o administrador do sistema pode atribuir privilégios de acesso de forma adequada a cada um dos blocos.

Assinale a opção que corresponde à técnica descrita acima.

- (A) Journaling.
- (B) Segmentação.
- (C) Swapping.
- (D) Paginação.
- (E) Polonesa Reversa.

24

A prática da computação científica pode se tornar mais eficiente quando ferramentas especializadas são utilizadas. Entre elas, uma das principais são os sistemas operacionais voltados para computação científica.

Diante do exposto, assinale a opção que apresenta a característica que é uma vantagem dos sistemas operacionais para computação científica.

- (A) Eles são projetados apenas para uso em ambientes de escritório, como processamento de texto e navegação na *web*.
- (B) Eles oferecem suporte exclusivo para jogos e entretenimento multimídia.
- (C) Eles são otimizados para lidar com cargas de trabalho intensivas em computação, como modelagem matemática, simulação e análise de dados.
- (D) Eles são compatíveis apenas com hardware de baixo desempenho, como processadores simples e memória limitada.
- (E) Eles não têm suporte para desenvolvimento de software personalizado ou programação de baixo nível.

25

A arquitetura de um computador define como seus componentes são organizados e a forma da execução de instrução e fluxos de dados por ocasião de seu funcionamento.

Diante do exposto, assinale a opção que melhor descreve um aspecto da arquitetura escalar de computadores.

- (A) Na arquitetura escalar, cada instrução é dividida em múltiplos estágios de execução, incluindo busca, decodificação, execução e escrita de resultados.
- (B) A arquitetura escalar permite que múltiplos processadores compartilhem eficientemente uma única unidade de controle centralizada.
- (C) Os processadores escalares são limitados a executar apenas uma instrução por ciclo de *clock*, sem a capacidade de aproveitar múltiplos pipelines de execução.
- (D) Na arquitetura escalar, a largura do barramento de dados não afeta o desempenho do processador devido à sua natureza altamente paralela.
- (E) A arquitetura escalar é projetada exclusivamente para aplicações de baixa demanda computacional, como tarefas de escritório e navegação na *web*.

26

Na computação de alto desempenho, a aplicabilidade influencia diretamente no tipo de arquitetura a ser utilizada.

Nesse contexto, assinale a opção que descreve corretamente um aspecto da arquitetura vetorial de computadores.

- (A) Na arquitetura vetorial, cada instrução é executada em uma única unidade de processamento, limitando assim o paralelismo de dados.
- (B) Os computadores com arquitetura vetorial são projetados exclusivamente para lidar com operações aritméticas simples, como adição e subtração.
- (C) Em computadores com arquitetura vetorial, as instruções são projetadas para operar em múltiplos elementos de um vetor simultaneamente, permitindo um alto grau de paralelismo de dados.
- (D) Os computadores com arquitetura vetorial geralmente têm um conjunto limitado de instruções e não oferecem suporte a operações complexas.
- (E) Na arquitetura vetorial, os vetores de dados são organizados de forma dinâmica, permitindo que o processador ajuste automaticamente o tamanho do vetor conforme necessário.

27

Na execução de um programa computacional, os dados e instruções podem ser tratados de forma individual ou múltipla, dependendo da arquitetura do computador.

Assinale a opção que descreve corretamente uma diferença entre as arquiteturas SIMD (*Single instruction - Multiple Data*) e MIMD (*Multiple instruction - Multiple Data*).

- (A) Na arquitetura SIMD, cada unidade de processamento executa instruções diferentes em conjunto de dados diferentes, enquanto na arquitetura MIMD, todas as unidades de processamento executam a mesma instrução em conjunto de dados diferentes.
- (B) A arquitetura SIMD é mais flexível e escalável do que a arquitetura MIMD.
- (C) Na arquitetura MIMD, todas as unidades de processamento devem estar sincronizadas para executar instruções em conjunto, enquanto na arquitetura SIMD, as unidades de processamento podem executar instruções de forma assíncrona.
- (D) A arquitetura MIMD é mais eficiente para tarefas altamente paralelizáveis, como processamento de imagens e vídeo, enquanto a arquitetura SIMD é mais adequada para tarefas com dependências complexas entre os dados.
- (E) Na arquitetura SIMD, uma única instrução é aplicada a múltiplos dados simultaneamente, enquanto na arquitetura MIMD, múltiplas instruções são executadas em múltiplos conjuntos de dados independentes.

28

Com a evolução tecnológica, os computadores passaram da arquitetura de processamento sequencial para arquitetura de processamento paralelo.

Assinale a opção que descreve corretamente uma característica da arquitetura de processamento paralelo.

- (A) A arquitetura de processamento paralelo é caracterizada pela execução sequencial de instruções em um único núcleo de processador, garantindo alta eficiência computacional.
- (B) Em arquiteturas de processamento paralelo, múltiplos núcleos de processador operam de forma independente, executando diferentes tarefas simultaneamente.
- (C) Na arquitetura de processamento paralelo, apenas uma instrução é executada por vez, mas em múltiplos conjuntos de dados, aumentando assim a eficiência de processamento.
- (D) A arquitetura de processamento paralelo não é adequada para lidar com problemas complexos, pois requer uma alta complexidade de programação e sincronização entre os núcleos de processador.
- (E) Arquiteturas de processamento paralelo não são capazes de lidar com grandes conjuntos de dados de forma eficiente, pois a comunicação entre os núcleos de processador é limitada.

29

Com a necessidade do aumento da capacidade de processamento dos computadores, muitos mecanismos para alcançar este objetivo foram desenvolvidos. Entre eles, podemos citar a paralelização através de aceleradores.

Assinale a opção que descreve corretamente um aspecto da paralelização usando aceleradores.

- (A) A paralelização usando aceleradores envolve exclusivamente a utilização de múltiplos núcleos de CPU em um único processador para aumentar o desempenho computacional.
- (B) A aceleração é alcançada principalmente através da utilização de caches maiores nos aceleradores, reduzindo assim o tempo de acesso à memória principal.
- (C) A paralelização usando aceleradores depende da utilização de unidades de processamento gráfico (GPUs) para executar tarefas computacionais de forma massivamente paralela.
- (D) A paralelização usando aceleradores é menos eficaz do que a utilização de múltiplos processadores em paralelo, pois os aceleradores têm menor capacidade de processamento.
- (E) A aceleração usando aceleradores é limitada pela dependência de instruções, tornando difícil a utilização eficiente de múltiplos núcleos de processamento.

30

Uma das formas mais conhecidas para classificar arquiteturas de computadores é através da taxonomia de Flynn.

Nesse contexto, assinale a opção que descreve corretamente a Taxonomia de Flynn.

- (A) A Taxonomia de Flynn classifica os computadores de acordo com a quantidade de memória cache disponível em cada núcleo de processador.
- (B) De acordo com a Taxonomia de Flynn, os sistemas são classificados com base na quantidade de *threads* que podem ser executadas simultaneamente em um processador.
- (C) A Taxonomia de Flynn classifica os sistemas de acordo com a quantidade de instruções por ciclo que cada processador é capaz de executar.
- (D) De acordo com a Taxonomia de Flynn, os sistemas são classificados com base no número de instruções executadas e no número de dados processados simultaneamente.
- (E) A Taxonomia de Flynn classifica os sistemas de acordo com o número de fluxos de instruções e dados independentes que podem ser executados simultaneamente.

31

A Computação de Alto Desempenho (HPC) implementa vários mecanismos adicionais em relação à computação tradicional, sendo um deles, a administração de filas de processamento.

Diante do exposto, assinale a opção que indica o papel da administração de filas de processamento em sistemas de Computação de Alto Desempenho (HPC).

- (A) Garantir que cada processo tenha acesso exclusivo a todos os recursos do sistema para maximizar o desempenho.
- (B) Distribuir equitativamente os recursos do sistema entre os processos para garantir um uso eficiente e justo.
- (C) Minimizar a utilização da CPU para reduzir o consumo de energia e resfriamento.
- (D) Alocar recursos apenas para os processos que solicitam prioridade máxima, independentemente de sua carga de trabalho.
- (E) Limitar o número total de processos ativos para evitar sobrecarga nos sistemas de armazenamento.

32

A Computação de Alto Desempenho (HPC) implementa vários mecanismos adicionais em relação à computação tradicional, sendo um deles, o gerenciamento de contas de usuários.

Diante do exposto, a função principal do gerenciamento de contas de usuários em um ambiente de Computação de Alto Desempenho (HPC) é:

- (A) facilitar o acesso irrestrito a todos os recursos do sistema para todos os usuários.
- (B) garantir que cada usuário tenha acesso exclusivo a todos os recursos do sistema.
- (C) controlar e limitar o acesso dos usuários aos recursos do sistema de acordo com suas necessidades e permissões.
- (D) permitir que os usuários compartilhem suas contas para facilitar a colaboração em projetos.
- (E) ignorar completamente o gerenciamento de contas, uma vez que todos os usuários têm acesso total aos recursos do sistema.

33

Algumas diferenças funcionais podem ser percebidas entre a computação tradicional e a Computação de Alto Desempenho (HPC). Uma dessas funções é o gerenciamento de *jobs*.

Nesse contexto, uma função importante do gerenciamento de *jobs* em um ambiente de Computação de Alto Desempenho (HPC) é:

- (A) definir as preferências de design de software para os *jobs* a serem executados.
- (B) garantir que todos os *jobs* tenham a mesma prioridade de execução para manter a equidade entre os usuários.
- (C) alocar eficientemente recursos de computação, como CPU e memória, para maximizar a utilização do sistema.
- (D) monitorar exclusivamente o status dos *jobs* após a conclusão da execução.
- (E) reduzir o número de *jobs* submetidos pelos usuários para simplificar o gerenciamento do sistema.

34

Um dos principais componentes de um sistema para Computação de Alto Desempenho (HPC) é o gerenciador de *jobs*. Nesse contexto, um dos mais conhecidos é o SLURM. Em um ambiente de Computação de Alto Desempenho (HPC), assinale a opção que indica a principal função do SLURM no gerenciamento de *Jobs*.

- (A) Priorizar aleatoriamente a execução de *jobs* para promover a equidade entre os usuários.
- (B) Alocar recursos de forma eficiente, como CPU, memória e dispositivos de armazenamento, para os *jobs* submetidos.
- (C) Limitar o número de *jobs* que um usuário pode submeter por dia, independentemente de suas necessidades computacionais.
- (D) Realizar exclusivamente o monitoramento do status dos *jobs* após a conclusão da execução.
- (E) Ignorar completamente o gerenciamento de *jobs*, deixando os usuários responsáveis por controlar seus próprios processos.

35

O controle de quotas é um importante mecanismo na Computação de Alto Desempenho (HPC). A principal finalidade do controle de quotas em um ambiente de Computação de Alto Desempenho (HPC) é

- (A) maximizar o uso de recursos, permitindo que os usuários utilizem quantidades ilimitadas de CPU, memória e armazenamento.
- (B) garantir um acesso equitativo aos recursos do sistema, evitando que um único usuário ou projeto monopolize os recursos disponíveis.
- (C) limitar o acesso apenas aos usuários administradores, impedindo que usuários regulares utilizem os recursos do sistema.
- (D) aumentar os custos operacionais do sistema, permitindo que os usuários utilizem quantidades excessivas de recursos.
- (E) reduzir a disponibilidade do sistema, priorizando o uso de recursos para projetos específicos em detrimento de outros.

36

Dada sua complexidade, o monitoramento é uma prática fundamental na Computação de Alto Desempenho.

Assinale a opção que indica a principal função das ferramentas de monitoramento de filas de processamento em um ambiente de Computação de Alto Desempenho (HPC).

- (A) Gerar relatórios detalhados sobre o desempenho individual de cada nó de computação do cluster.
- (B) Monitorar a utilização da rede de comunicação entre os nós do cluster.
- (C) Fornecer informações sobre o estado das filas de processamento, como o número de *jobs* na fila e o tempo de espera estimado para execução.
- (D) Realizar a manutenção preventiva dos sistemas de armazenamento em disco do cluster.
- (E) Controlar a temperatura e o consumo de energia de cada nó de computação para otimizar a eficiência energética.

37

Vários dos componentes da Computação de Alto Desempenho merecem ser monitorados, dada a relevância deles para o bom funcionamento desse tipo de sistema.

Em um ambiente de Computação de Alto Desempenho (HPC), assinale a opção que apresenta um aspecto importante no monitoramento de armazenamento.

- (A) Verificar a temperatura ambiente do data center para garantir condições ideais de operação dos dispositivos de armazenamento.
- (B) Monitorar a integridade dos dados armazenados para identificar e corrigir erros de leitura/gravação.
- (C) Acompanhar a largura de banda da rede para garantir que os dados sejam transferidos eficientemente entre os sistemas de armazenamento.
- (D) Analisar a utilização da CPU para otimizar o desempenho dos sistemas de armazenamento.
- (E) Verificar a versão do sistema operacional instalado nos dispositivos de armazenamento para garantir compatibilidade com os aplicativos executados no cluster.

38

O gerenciamento dos *jobs* é uma importante característica na Computação de Alto Desempenho (HPC).

Assinale a afirmativa que apresenta os aspectos essenciais a serem observados por ocasião do monitoramento de *jobs* em um ambiente de Computação de Alto Desempenho (HPC).

- (A) Apenas o *status* dos *jobs*, já que isso é suficiente para garantir que as tarefas estejam sendo executadas corretamente.
- (B) O tempo de execução dos *jobs*, pois isso é o único aspecto que realmente importa para os usuários.
- (C) O consumo de energia dos nós de processamento, pois isso influencia diretamente nos custos operacionais do sistema.
- (D) Utilização de recursos, tempo de execução, prioridades e políticas de escalonamento, notificações/alertas e histórico/estatísticas.
- (E) Apenas a prioridade dos *jobs*, pois isso determina a ordem de execução das tarefas.

39

O gerenciamento de memória é um dos itens mais críticos na Computação de Alto Desempenho (HPC), para a garantia da sua boa performance.

Assinale a afirmativa que apresenta os aspectos essenciais a serem observados por ocasião do monitoramento de memória em um ambiente de Computação de Alto Desempenho (HPC).

- (A) Verificar a utilização de memória, a fragmentação de memória, a alocação dinâmica de memória, o uso de swap e a temperatura dos módulos de memória.
- (B) Acompanhar o uso de memória física, a quantidade de memória compartilhada, a taxa de transmissão de dados entre os módulos de memória, a eficiência de alocação de memória e a disponibilidade de atualizações de firmware para os módulos de memória.
- (C) Observar o consumo de memória em relação ao número de processadores, a velocidade de acesso à memória, a latência de acesso e a largura de banda de memória.
- (D) Monitorar a utilização de memória, vazamentos de memória, uso de swap, fragmentação de memória e alocação dinâmica de memória.
- (E) Acompanhar o uso de memória em relação à utilização da CPU, o consumo de memória por processo, a quantidade de memória compartilhada, a eficiência do algoritmo de paginação e a temperatura do processador.

40

A rede desempenha um papel fundamental na Computação de Alto Desempenho, sendo que seu mal funcionamento pode degradar a performance de suas execuções.

Assinale a afirmativa que apresenta os aspectos essenciais a serem observados por ocasião do monitoramento de rede em um ambiente de Computação de Alto Desempenho (HPC).

- (A) A latência de rede entre os nós do cluster, a disponibilidade da rede, a segurança da rede, a quantidade de pacotes transmitidos e a temperatura dos switches de rede.
- (B) A utilização da largura de banda, o tráfego de rede entre o cluster e outros sistemas, a latência de rede, os erros de rede e a segurança da rede.
- (C) Apenas a segurança da rede, garantindo que não haja violações de políticas de segurança ou atividades maliciosas.
- (D) Apenas observar a disponibilidade da rede e a quantidade de pacotes transmitidos, pois isso é suficiente para garantir a conectividade dos sistemas.
- (E) Focar somente na utilização da largura de banda da rede, ignorando outros aspectos que possam afetar o desempenho e a segurança do sistema.

41

CUDA é uma arquitetura de computação paralela desenvolvida pela NVIDIA. Nesse contexto, assinale a opção que apresenta uma característica importante da programação com CUDA.

- (A) CUDA é uma linguagem de programação de alto nível que simplifica o desenvolvimento de *software* para GPUs, mas não oferece suporte para paralelismo.
- (B) A programação com CUDA requer o uso exclusivo de CPUs devido à sua arquitetura específica.
- (C) A programação com CUDA envolve a escrita de *kernels*, que são funções executadas em paralelo por múltiplos threads em uma GPU.
- (D) CUDA é uma linguagem proprietária desenvolvida apenas pela AMD para suas GPUs.
- (E) A programação com CUDA é restrita a aplicativos de processamento de texto e imagem e não pode ser utilizada para cálculos científicos ou de alto desempenho.

42

MPI, ou Interface de Passagem de Mensagens (Message Passing Interface), é um conjunto de especificações, bibliotecas e rotinas para comunicação entre processos em sistemas distribuídos e paralelos. Diante do exposto, assinale a opção que apresenta um atributo importante da programação com MPI.

- (A) MPI é uma linguagem de programação de alto nível que simplifica o desenvolvimento de *software* para comunicação entre processos em um único núcleo de CPU.
- (B) A programação com MPI é restrita a ambientes de computação paralela baseados apenas em CPUs, sem suporte para aceleração por GPU.
- (C) MPI é uma biblioteca de comunicação que permite a troca de dados entre processos em um sistema distribuído, como um *cluster* de computadores.
- (D) MPI é uma linguagem proprietária desenvolvida apenas pela NVIDIA para suas GPUs.
- (E) A programação com MPI é adequada apenas para aplicativos de computação sequencial e não oferece benefícios significativos para sistemas paralelos.

43

CUDA e MPI são duas tecnologias amplamente utilizadas em Computação de Alto Desempenho. Assinale a alternativa que apresenta a principal diferença entre CUDA e MPI.

- (A) CUDA e MPI são linguagens de programação concorrentes, cada uma projetada para tarefas específicas de processamento paralelo em GPUs e CPUs, respectivamente.
- (B) CUDA e MPI são ambas linguagens de programação para programação paralela, mas CUDA é específica para programação em GPUs, enquanto MPI é utilizada para comunicação entre processos em sistemas distribuídos.
- (C) CUDA e MPI são bibliotecas de comunicação, cada uma projetada para permitir a troca de dados entre processos em sistemas distribuídos, como clusters de computadores.
- (D) CUDA e MPI são abordagens alternativas para a programação paralela em CPUs, com CUDA focada em processamento gráfico e MPI em processamento de dados.
- (E) CUDA e MPI são tecnologias complementares, onde CUDA é utilizada para processamento paralelo em GPUs e MPI é utilizada para comunicação entre processos em sistemas distribuídos.

44

Uma forma de aumentar a capacidade computacional é montar e utilizar um *cluster* HPC. Assinale a afirmativa que apresenta uma característica essencial de um *cluster* HPC.

- (A) Cada nó em um *cluster* HPC geralmente possui uma única CPU de alta potência.
- (B) Os *clusters* HPC são limitados a uma única localização física e não podem se estender por várias instalações.
- (C) A interconexão de alta velocidade entre os nós é fundamental para garantir o desempenho do *cluster*.
- (D) *Clusters* HPC são projetados exclusivamente para aplicativos de baixa intensidade de computação.
- (E) *Clusters* HPC não podem ser escalonados horizontalmente para atender a demandas de computação crescentes.

45

A computação científica tem peculiaridades complementares que a diferencia da computação tradicional. Nesse contexto, assinale a afirmativa que apresenta uma camada adicional comum em um sistema operacional para computação científica em comparação com um sistema operacional tradicional.

- (A) Ambientes de *desktop* e interfaces gráficas de usuário para facilitar o uso do sistema.
- (B) Sistemas de arquivos de propósito geral para armazenamento de dados, sem otimizações para acesso rápido em aplicativos científicos.
- (C) Gerenciamento de recursos avançado, como sistemas de filas de trabalho e agendadores de tarefas.
- (D) Suporte nativo para jogos e entretenimento, incluindo drivers de gráficos otimizados para videogames.
- (E) Nenhuma camada adicional é necessária, pois um sistema operacional tradicional pode atender às necessidades de computação científica.

Realização

