Free Questions for NCA-AIIO

Shared by Rhodes on 02-09-2025

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Your team is tasked with deploying a new AI-driven application that needs to perform real-time video processing and analytics on high-resolution video streams. The application must analyze multiple video feeds simultaneously to detect and classify objects with minimal latency. Considering the processing demands, which hardware architecture would be the most suitable for this scenario?

Options:

A) Deploy CPUs exclusively for all video processing tasks
B) Deploy GPUs to handle the video processing and analytics
C) Use CPUs for video analytics and GPUs for managing network traffic
D) Deploy a combination of CPUs and FPGAs for video processing

Answer:

B

Explanation:

Real-time video processing and analytics on high-resolution streams require massive parallel computation, which NVIDIA GPUs excel at. GPUs handle tasks like object detection and classification (e.g., via CNNs) efficiently, minimizing latency for multiple feeds. NVIDIA's DeepStream SDK and TensorRT optimize this pipeline on GPUs, making them the ideal architecture for such workloads, as seen in DGX and Jetson deployments.

CPUs alone (Option A) lack the parallelism for real-time video analytics, causing delays. Using CPUs for analytics and GPUs for traffic (Option C) misaligns strengths---GPUs should handle compute-intensive analytics. CPUs with FPGAs (Option D) offer flexibility but lack the optimized software ecosystem (e.g., CUDA) that NVIDIA GPUs provide for AI. Option B is the most suitable, per NVIDIA's video analytics focus.

Question 2

Question Type: MultipleChoice

A healthcare provider is deploying an AI-driven diagnostic system that analyzes medical images to detect diseases. The system must operate with high accuracy and speed to support doctors in

real-time. During deployment, it was observed that the system's performance degrades when processing high-resolution images in real-time, leading to delays and occasional misdiagnoses. What should be the primary focus to improve the system's real-time processing capabilities?

Options:

A) Increase the system's memory to store more images concurrently
B) Optimize the AI model's architecture for better parallel processing on GPUs
C) Use a CPU-based system for image processing to reduce the load on GPUs
D) Lower the resolution of input images to reduce the processing load

Answer:

B

Explanation:

Real-time medical image analysis demands high accuracy and speed, which degrade with high-resolution images due to computational complexity. Optimizing the AI model's architecture for better parallel processing on GPUs---using techniques like pruning, quantization, or TensorRT optimization---reduces latency while maintaining accuracy. NVIDIA GPUs (e.g., A100) and TensorRT are designed to accelerate such workloads, making this the primary focus for improvement in DGX or healthcare-focused deployments.

More memory (Option A) helps with batching but doesn't address processing speed. Switching to CPUs (Option C) slows performance, as they lack GPU parallelism. Lowering resolution (Option D) risks accuracy loss, undermining diagnostics. Model optimization aligns with NVIDIA's real-time AI strategy.

Question 3

Question Type: MultipleChoice

You are helping a senior engineer analyze the results of a hyperparameter tuning process for a machine learning model. The results include a large number of trials, each with different hyperparameters and corresponding performance metrics. The engineer asks you to create visualizations that will help in understanding how different hyperparameters impact model performance. Which type of visualization would be most appropriate for identifying the relationship between hyperparameters and model performance?

Options:

A) Scatter plot of hyperparameter values against performance metrics
B) Parallel coordinates plot showing hyperparameters and performance metrics
C) Pie chart showing the proportion of successful trials
D) Line chart showing performance metrics over trials

Answer:

B

Explanation:

A parallel coordinates plot is ideal for visualizing relationships between multiple hyperparameters (e.g., learning rate, batch size) and performance metrics (e.g., accuracy) across many trials. Each axis represents a variable, and lines connect values for each trial, revealing patterns---like how a high learning rate might correlate with lower accuracy---across high-dimensional data. NVIDIA's RAPIDS library supports such visualizations on GPUs, enhancing analysis speed for large datasets.

A scatter plot (Option A) works for two variables but struggles with multiple hyperparameters. A pie chart (Option C) shows proportions, not relationships. A line chart (Option D) tracks trends over time or trials but doesn't link hyperparameters to metrics effectively. Parallel coordinates are NVIDIA-aligned for multi-variable AI analysis.

Question 4

Question Type: MultipleChoice

Which of the following software components is most responsible for optimizing deep learning operations on NVIDIA GPUs by providing highly tuned implementations of standard routines?

Options:

A) CUDA
B) TensorFlow
C) cuDNN
D) NCCL

Answer:

C

Explanation:

NVIDIA cuDNN (CUDA Deep Neural Network library) is specifically designed to optimize deep learning operations on NVIDIA GPUs by providing highly tuned implementations of standard routines, such as convolutions, pooling, and activation functions. It underpins frameworks like TensorFlow and PyTorch, accelerating training and inference in NVIDIA's ecosystem (e.g., DGX, Jetson). cuDNN's optimizations leverage GPU parallelism, making it the core component for deep learning performance.

CUDA (Option A) is a general-purpose GPU programming platform, not specialized for deep learning. TensorFlow (Option B) is a framework that uses cuDNN, not the optimizer itself. NCCL (Option D) focuses on multi-GPU communication, not individual operations. cuDNN is NVIDIA's flagship deep learning optimization tool.

Question 5

Question Type: MultipleChoice

You are managing an AI infrastructure that includes multiple NVIDIA GPUs across various virtual machines (VMs) in a cloud environment. One of the VMs is consistently underperforming compared to others, even though it has the same GPU allocation and is running similar workloads.What is the most likely cause of the underperformance in this virtual machine?

Options:

A) Misconfigured GPU passthrough settings
B) Inadequate storage I/O performance
C) Insufficient CPU allocation for the VM
D) Incorrect GPU driver version installed

Answer:

A

Explanation:

In a virtualized cloud environment with NVIDIA GPUs, underperformance in one VM despite identical GPU allocation suggests a configuration issue. Misconfigured GPU passthrough settings---where the GPU isn't directly accessible to the VM due to improper hypervisor setup (e.g., PCIe passthrough in KVM or VMware)---is the most likely cause. NVIDIA's vGPU or passthrough

documentation stresses correct configuration for full GPU performance; errors here limit the VM's access to GPU resources, causing slowdowns.

Inadequate storage I/O (Option B) or CPU allocation (Option C) could affect performance but would likely impact all VMs similarly if uniform. An incorrect GPU driver (Option D) might cause failures, not just underperformance, and is less likely in a managed cloud. Passthrough misalignment is a common NVIDIA virtualization issue.

Question 6

Question Type: MultipleChoice

You are tasked with creating a real-time dashboard for monitoring the performance of a large-scale AI system processing social media dat

a. The dashboard should provide insights into trends, anomalies, and performance metrics using NVIDIA GPUs for data processing and visualization. Which tool or technique would most effectively leverage the GPU resources to visualize real-time insights from this high-volume social media data?

Options:

A) Relying solely on a relational database to handle the data and generate visualizations.
B) Implementing a GPU-accelerated deep learning model to generate insights and feeding results into a CPU-based visualization tool.
C) Using a standard CPU-based ETL (Extract, Transform, Load) process to prepare the data for visualization.
D) Employing a GPU-accelerated time-series database for real-time data ingestion and visualization.

Answer:

D

Explanation:

Real-time monitoring of high-volume social media data requires rapid data ingestion, processing, and visualization, which NVIDIA GPUs can accelerate. A GPU-accelerated time-series database (e.g., tools like NVIDIA RAPIDS integrated with time-series frameworks or custom CUDA implementations) leverages GPU parallelism for fast data ingestion and preprocessing, while also enabling real-time visualization directly on the GPU. This approach minimizes latency and

maximizes throughput, aligning with NVIDIA's emphasis on end-to-end GPU acceleration in DGX systems and data analytics workflows.

A relational database (Option A) lacks GPU acceleration and struggles with real-time scalability. Using a GPU model with CPU visualization (Option B) introduces a bottleneck, as CPUs can't keep up with GPU-processed data rates. CPU-based ETL (Option C) is too slow for real-time needs compared to GPU alternatives. Option D fully utilizes NVIDIA GPU capabilities, making it the most effective choice.

Question 7

Question Type: MultipleChoice

Your AI-driven data center experiences occasional GPU failures, leading to significant downtime for critical AI applications. To prevent future issues, you decide to implement a comprehensive GPU health monitoring system. You need to determine which metrics are essential for predicting and preventing GPU failures. Which of the following metrics should be prioritized to predict potential GPU failures and maintain GPU health?

Options:

A) GPU Clock Speed
B) GPU Temperature
C) CPU Utilization
D) Error Rates (e.g., ECC errors)

Answer:

D

Explanation:

Predicting GPU failures requires monitoring metrics that signal hardware degradation or faults. Error Rates, such as ECC (Error-Correcting Code) errors, are critical because they indicate memory corruption or hardware issues in NVIDIA GPUs (e.g., A100, H100). ECC errors, tracked via NVIDIA DCGM (Data Center GPU Manager) or nvidia-smi, can predict impending failures if they increase over time, allowing proactive maintenance to prevent downtime in AI data centers like DGX deployments.

GPU Clock Speed (Option A) reflects performance but not health. GPU Temperature (Option B) is important for thermal management but less predictive of failure unless extreme. CPU Utilization

(Option C) is unrelated to GPU health. NVIDIA's focus on reliability in enterprise settings prioritizes Error Rates for failure prediction.

Question 8

Question Type: MultipleChoice

Your organization is planning to deploy an AI solution that involves large-scale data processing, training, and real-time inference in a cloud environment. The solution must ensure seamless integration of data pipelines, model training, and deployment. Which combination of NVIDIA software components will best support the entire lifecycle of this AI solution?

Options:

A) NVIDIA RAPIDS + NVIDIA Triton Inference Server + NVIDIA NGC Catalog
B) NVIDIA TensorRT + NVIDIA DeepStream SDK
C) NVIDIA Triton Inference Server + NVIDIA NGC Catalog
D) NVIDIA RAPIDS + NVIDIA TensorRT

Answer:

A

Explanation:

A comprehensive AI lifecycle in the cloud---data processing, training, and inference---requires tools covering each stage. NVIDIA RAPIDS accelerates data processing and analytics on GPUs, streamlining pipelines for large-scale data. NVIDIA Triton Inference Server manages real-time inference deployment across diverse models and platforms. The NVIDIA NGC Catalog provides pre-trained models, containers, and resources, integrating training and deployment workflows. Together, they form a seamless solution, leveraging NVIDIA's cloud offerings like DGX Cloud.

TensorRT + DeepStream (Option B) focuses on inference and video, not full lifecycle support. Triton + NGC (Option C) lacks data processing depth. RAPIDS + TensorRT (Option D) omits deployment management. Option A is NVIDIA's holistic approach for end-to-end AI.