

Lead2Passed



Lead2Passed

HOME

ALL VENDORS

★ GUARANTEE

? FAQ

TESTIMONIALS

Login / Register My Shopcart (1)

Input your exam code ...



Try before you buy

Download a free sample of any of our exam questions and answers

- ✓ Online Test Engine: Online Tool, Convenient, easy to study. Instant Online Access. Supports All Web Browsers.
- ✓ PDF format: Easy to read and print learning materials, our products are available in PDF file format.
- ✓ Desktop Test Engine: Installable Software Application. Simulates Real Exam Environment. Practice Offline Anytime.



Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.



365 Days Free Updates

Free update is available within 365 days after your purchase. After 365 days, you will get 50% discounts for updating.



Money Back Guarantee

Full refund if you fail the corresponding exam in 60 days after purchasing. And Free get any another product.



Instant Download

After Payment, our system will send you the products you purchase in mailbox in a minute after payment. If not received within 2 hours, please contact us.

<http://www.lead2passed.com>

Valid Certification Exam Dumps Materials and Study Guide -
Lead2Passed

Exam : **NCP-ADS**

Title : **NVIDIA-Certified-Professional
Accelerated Data Science**

Vendor : **NVIDIA**

Version : **DEMO**

NO.1 A data scientist is training a deep learning model on an NVIDIA GPU but is encountering out-of-memory (OOM) errors.

To optimize GPU memory usage while maintaining efficient training performance, which of the following strategies should they prioritize?

- A. Increasing batch size without adjusting the optimizer settings
- B. Using mixed precision training with automatic loss scaling
- C. Storing all training data in GPU memory at once
- D. Using single-precision (FP32) calculations for better accuracy

Answer: B

NO.2 A machine learning engineer wants to evaluate the performance of NVIDIA RAPIDS cuDF and Apache Spark for large-scale data processing on a GPU-enabled cluster.

Which of the following strategies is the most effective for obtaining a fair and comprehensive benchmark?

- A. Run Spark on a CPU cluster while running RAPIDS on a GPU to compare real-world scenarios.
- B. Limit the benchmark to small datasets since GPUs excel at parallel processing.
- C. Focus only on processing speed without considering resource consumption differences between frameworks.
- D. Execute identical ETL workflows on cuDF and Spark-RAPIDS and measure execution time and resource utilization.

Answer: D

NO.3 You are working with a large dataset that contains missing values in multiple columns. Your goal is to prepare this dataset for training a machine learning model on an NVIDIA GPU using RAPIDS. Which of the following approaches is the most efficient method to handle missing values in this scenario?

- A. Convert the dataset to a NumPy array and manually replace missing values with the mean
- B. Apply a deep learning-based imputation model before moving data to the GPU
- C. Use `fillna()` with a fixed value on the GPU using cuDF
- D. Drop all rows containing missing values using Pandas before transferring data to the GPU

Answer: C

NO.4 You are working on an accelerated data science project and need to acquire a large dataset stored in a Parquet file format and load it efficiently for GPU processing using NVIDIA RAPIDS. Which of the following approaches is the most efficient way to load the dataset into a GPU-accelerated DataFrame?

- A. `df = cudf.read_parquet("data.parquet")`
- B. `df = cudf.to_gpu(pd.read_parquet("data.parquet"))`
- C. `df = pd.read_parquet("data.parquet")`
- D. `df = cudf.read_csv("data.parquet")`

Answer: A

NO.5 You are training a large-scale random forest model on a dataset with millions of rows and hundreds of features. The training time is significantly high when using traditional CPU-based

machine learning frameworks.

Which NVIDIA technology should you use to accelerate training while maintaining compatibility with common ML frameworks like scikit-learn?

- A. NVIDIA DeepStream to preprocess tabular data and optimize random forest model execution.
- B. NVIDIA RAPIDS cuML to accelerate random forest training using GPU-optimized implementations.
- C. NVIDIA Triton Inference Server to distribute random forest model training across multiple GPUs.
- D. NVIDIA TensorRT to accelerate random forest model training by optimizing tree-based algorithms.

Answer: B

NO.6 A data scientist is working on training a deep learning model in a cloud-based environment.

The dataset is large, and model convergence is taking too long on a standard CPU instance.

To optimize performance through GPU acceleration, which of the following strategies should the data scientist implement?

- A. Use a cloud instance with multiple GPUs and enable mixed-precision training.
- B. Store all training data in RAM and load it directly to the CPU for processing.
- C. Disable CUDA and use only OpenMP to parallelize computations across CPU cores.
- D. Increase the number of CPU cores and distribute training across multiple CPU threads.

Answer: A

NO.7 You are training a machine learning model using RAPIDS cuML and need to ensure that all numeric features are standardized for better model performance.

Which of the following is the best approach for scaling data using RAPIDS?

- A. `df_scaled = df / df.max()`
- B. `df_scaled = (df - df.min()) / (df.max() - df.min())`
- C. `scaler = cuml.preprocessing.StandardScaler()`
- D. `df_scaled = scaler.fit_transform(df)`
- E. `df_scaled = df.apply(lambda x: x / np.linalg.norm(x))`

Answer: C

NO.8 When deciding whether to use GPU acceleration or a traditional CPU approach for a machine learning task, which of the following factors should be considered to determine if the data qualifies as "big data" and whether GPU acceleration is beneficial? (Select two)

- A. CPU-based machine learning methods are always more effective for small datasets, regardless of the algorithm used.
- B. The size of the dataset in terms of rows and columns is irrelevant when determining if it qualifies as big data.
- C. GPU acceleration is beneficial when the dataset can be divided into independent chunks that can be processed in parallel.
- D. The complexity of the algorithm being used plays a crucial role in deciding whether to use GPU acceleration, with more complex algorithms benefiting from parallel computation.
- E. The dataset must be over 100GB in size to qualify as big data and warrant GPU acceleration.

Answer: C, D

NO.9 You are tasked with implementing a multi-GPU data pipeline using Dask-CUDA to process large

datasets stored in Parquet format. Your goal is to achieve optimal GPU memory utilization and minimize inter-GPU communication overhead.

Which of the following approaches best aligns with these goals?

- A.** Set `dask.config.set({'distributed.worker.memory.target': 0.9})` to allocate 90% of the total CPU memory for GPU operations.
- B.** Use `dask_cudf.read_parquet()` with `split_row_groups=True` to evenly distribute data across GPUs.
- C.** Use `dask.persist()` instead of `dask.compute()` to force immediate execution of tasks before distribution to GPUs.
- D.** Use `dask.array` instead of `dask_cudf` because it provides better performance for structured tabular data.

Answer: B

NO.10 A machine learning engineer wants to evaluate the performance of NVIDIA RAPIDS cuDF and Apache Spark for large-scale data processing on a GPU-enabled cluster.

Which of the following strategies is the most effective for obtaining a fair and comprehensive benchmark?

- A.** Run Spark on a CPU cluster while running RAPIDS on a GPU to compare real-world scenarios.
- B.** Limit the benchmark to small datasets since GPUs excel at parallel processing.
- C.** Execute identical ETL workflows on cuDF and Spark-RAPIDS and measure execution time and resource utilization.
- D.** Focus only on processing speed without considering resource consumption differences between frameworks.

Answer: C

NO.11 When performing benchmarking and optimization for GPU-accelerated workflows, which of the following tools is best suited for analyzing the memory utilization and computational efficiency of deep learning models running on Nvidia GPUs?

- A.** Nvidia Nsight Compute
- B.** Nvidia TensorRT
- C.** Nvidia CUDA Profiler
- D.** Nvidia Riva

Answer: A

NO.12 Which of the following data normalization techniques is most appropriate when the dataset contains outliers, and you want to minimize the influence of those outliers on the model performance?

- A.** Log Transformation
- B.** Z-score Standardization
- C.** Min-Max Scaling
- D.** Robust Scaling

Answer: D

NO.13 You are working on a data processing pipeline using NVIDIA GPUs for accelerating computations.

You need to monitor the pipeline's performance to identify bottlenecks.

Which of the following tools or techniques can be used to efficiently recognize bottlenecks in such a GPU-accelerated pipeline? (Select two)

- A. NVIDIA DLA (Deep Learning Accelerator)
- B. NVIDIA CUDA Profiler (nvprof)
- C. NVIDIA Nsight Systems
- D. NVIDIA nvidia-smi
- E. NVIDIA TensorRT Profiling

Answer: B, C

NO.14 A data engineer is designing an Extract, Transform, Load (ETL) pipeline for a retail analytics platform that processes millions of customer transactions per day. The primary objective is to accelerate data ingestion, transformation, and storage while ensuring efficient scalability.

Which of the following approaches would be the most effective for optimizing this ETL workflow using NVIDIA-accelerated ETL tools?

- A. Use NVIDIA RAPIDS cuDF for data transformations and Dask-cuDF for parallelized processing across multiple GPUs.
- B. Perform all transformations using Pandas DataFrames and then use multiprocessing to parallelize the workload on CPUs.
- C. Implement ETL processes using only SQL-based transformations within a relational database system.
- D. Use Apache Spark with CPU-based processing instead of leveraging GPU acceleration.

Answer: A

NO.15 You are tasked with selecting the optimal data processing library for an AI project that involves handling varying dataset sizes. The project must be flexible enough to scale from small datasets (a few GBs) to large datasets (hundreds of GBs or more) using NVIDIA technologies.

Which of the following libraries would you choose for optimal performance at both small and large scales?

- A. CUDA Toolkit
- B. Dask
- C. cuDF
- D. PyTorch

Answer: C

NO.16 You are working with a large dataset containing millions of high-resolution images for a deep learning project. The dataset needs to be processed efficiently on a GPU before training a model. Which NVIDIA technology is best suited for preprocessing, augmenting, and efficiently loading the dataset into memory?

- A. NVIDIA DALI (Data Loading Library) to accelerate data loading and preprocessing on the GPU.
- B. NVIDIA Triton Inference Server to preprocess the dataset before model training.
- C. NVIDIA Nsight Compute to optimize image dataset processing.
- D. NVIDIA RAPIDS cuDF to transform image data into tabular format for analysis.

Answer: A

NO.17 You are working with a dataset containing billions of records stored in a Parquet file. You need to load this dataset efficiently into an NVIDIA-accelerated RAPIDS environment for feature engineering.

Which of the following is the best approach?

- A.** Use `pandas.read_parquet()` to load the dataset and then convert it to a cuDF DataFrame
- B.** Load the Parquet file into Dask and then convert it into a cuDF DataFrame for parallel processing
- C.** Convert the dataset into a CSV format and use `cudf.read_csv()` to load it into RAPIDS
- D.** Load the Parquet file directly into a cuDF DataFrame using `cudf.read_parquet()`

Answer: D

NO.18 You are building a predictive model for retail sales forecasting and need a dataset that includes historical sales transactions, customer demographics, and external economic indicators (e.g., inflation rate, unemployment rate).

Which of the following datasets would be the most appropriate for your model?

- A.** A dataset with global temperature trends over the past decade
- B.** A dataset of product reviews and customer sentiments from an e-commerce website
- C.** A public dataset of annual GDP per country
- D.** A dataset containing transaction history and customer profiles from a retail company

Answer: D

NO.19 You are working with a dataset containing hundreds of millions of records, and you need to perform ETL operations such as filtering, joins, and aggregations. Given the dataset size, which NVIDIA-accelerated library should you use to achieve optimal performance?

- A.** Pandas, as it is widely used and supports all common DataFrame operations, even for very large datasets.
- B.** cuDF, as it provides GPU-accelerated DataFrame operations similar to Pandas, allowing for efficient processing of large datasets.
- C.** NumPy, because it is optimized for numerical computing and offers better performance for handling tabular data.
- D.** cuPy, because it provides GPU-accelerated array operations, making it the best option for processing tabular data.

Answer: B

NO.20 You are optimizing a data pipeline for a large-scale machine learning project using NVIDIA RAPIDS and Apache Spark. The pipeline performs many expensive shuffle operations.

Which of the following is the most effective method to reduce shuffle and improve performance using NVIDIA technologies?

- A.** Use RAPIDS cuDF to perform in-memory data processing on GPU before shuffling to avoid network communication.
- B.** Store data in HDFS before performing shuffle operations to reduce GPU memory overhead.
- C.** Use RAPIDS cuDF to repartition the data in the GPU memory after each shuffle.
- D.** Implement a custom shuffle partitioning scheme using NVIDIA DALI for more control over data partitioning during shuffle operations.

Answer: A

NO.21 You are working on a large-scale graph analysis problem that involves computing the shortest paths between nodes in a massive social network dataset. You decide to leverage NVIDIA RAPIDS cuGraph for accelerated computation.

Which of the following cuGraph functions should you use?

- A. `cugraph.k_truss()`
- B. `cugraph.pagerank()`
- C. `cugraph.sssp()`
- D. `cugraph.label_propagation()`

Answer: C

NO.22 Which of the following methods are commonly used to handle missing data in data analysis? (Select two)

- A. Using a fixed constant value for imputation
- B. Imputing with the mean or median of the feature
- C. Using the mode to impute missing values
- D. Removing rows with missing data
- E. Ignoring missing data entirely

Answer: B, D

NO.23 You are working with cloud-based GPUs to process a large dataset (terabytes in size) stored in Parquet format. One column represents a unique identifier (e.g., product ID), and it contains only positive integers ranging from 1 to 100,000.

Which of the following data types provides the best balance of memory efficiency and performance?

- A. `float32`
- B. `int8`
- C. `float64`
- D. `uint16`

Answer: D

NO.24 You are working on an MLOps pipeline that involves loading a large dataset for training a deep learning model on an NVIDIA GPU. Before training, you need to ensure that the dataset fits within the available GPU memory.

Which of the following commands in Python using the pandas and numpy libraries can correctly determine the memory size of a dataset?

- A. `df.info(memory_usage='deep')`
- B. `np.array(df).nbytes`
- C. `df.memory_usage(deep=True).sum()`
- D. `sys.getsizeof(df)`

Answer: C

NO.25 A data scientist is using NVIDIA RAPIDS to perform statistical analysis as part of exploratory data analysis (EDA) on a dataset containing millions of product reviews. They need to compute basic

descriptive statistics such as mean, median, and variance efficiently.

Which of the following methods is the most appropriate for performing these calculations on GPUs?

- A. Use NumPy's statistical functions, such as `numpy.mean()` and `numpy.var()`
- B. Convert the dataset into a PyTorch tensor and use PyTorch's statistical methods
- C. Use cuDF's built-in statistical functions like `.mean()`, `.median()`, and `.var()`
- D. Use a traditional SQL database to compute statistics and then transfer results to the GPU

Answer: C

NO.26 A data scientist is analyzing large-scale sensor readings from an industrial IoT system and wants to visualize high-frequency time-series data efficiently.

Which approach using NVIDIA technologies would be the most effective for interactive visualization of this dataset?

- A. Apply TensorRT to compress the time-series data and use PyTorch's native visualization functions.
- B. Process the data using RAPIDS cuDF, then convert it to CSV and visualize it with Excel charts.
- C. Use RAPIDS cuDF with Datashader to render large time-series data on the GPU efficiently.
- D. Use NVIDIA OptiX to ray-trace the time-series dataset for visualization.

Answer: C

NO.27 You are tasked with optimizing the performance of a large-scale data science project that involves deep learning models on a cloud infrastructure. Your organization is using GPUs for model training.

Which of the following strategies would be the most effective in optimizing GPU performance for data science tasks? (Select two)

- A. Overclock the GPU to achieve higher computational speeds and improve training times.
- B. Use a single cloud instance with the largest GPU available to ensure maximum performance.
- C. Optimize GPU performance by limiting the number of threads running on each GPU.
- D. Use larger batch sizes to make better use of GPU memory during model training.
- E. Utilize multi-GPU training to parallelize the workload, reducing training time.

Answer: D, E

NO.28 You are working with a large time-series dataset consisting of millions of records and want to efficiently visualize trends over time using NVIDIA technologies. The dataset is stored as a cuDF DataFrame, and you need to generate an interactive line plot with minimal performance overhead.

Which of the following is the best approach to achieve this goal?

- A. Use the hvPlot library with RAPIDS cuDF to directly render the time-series data interactively
- B. Convert the cuDF DataFrame to a Pandas DataFrame and plot using Matplotlib
- C. Load the data into a Spark DataFrame and visualize using Apache Zeppelin
- D. Use the Bokeh library to plot the time-series data from a cuDF DataFrame directly

Answer: A

NO.29 A data engineering team is tasked with processing terabytes of log data every hour using an ETL pipeline. Due to the large data volume, they need a scalable GPU-accelerated solution that can distribute data processing across multiple GPUs.

Which approach best meets their needs?

- A.** Use Dask-cuDF to distribute cuDF DataFrame operations across multiple GPUs, enabling parallel ETL processing.
- B.** Process data using Pandas, then export the results to a CSV file for GPU-accelerated analytics.
- C.** Use NumPy for data transformations before converting the dataset into cuDF for final storage.
- D.** Use cuDF alone for processing log data, as it provides optimal performance on a single GPU.

Answer: A

NO.30 You are building a real-time recommendation system that processes high-frequency transactional data from millions of users.

The system must:

- Ingest and preprocess data efficiently
- Perform similarity computations for user-item recommendations
- Scale to handle rapid incoming transactions

Which of the following NVIDIA technologies is the best choice for this use case?

- A.** NVIDIA Triton Inference Server
- B.** RAPIDS cuGraph
- C.** NVIDIA NVTabular
- D.** CUDA Kernels with Custom C++ Code

Answer: C