NVIDIA.NCP-ADS.v2026-08-14.q103

Exam Code:	NCP-ADS
Exam Name:	NVIDIA-Certified-Professional Accelerated Data Science
Certification Provider:	NVIDIA
Free Question Number:	103
Version:	v2026-08-14
# of views:	713
# of Questions views:	1191
https://www.dumpsfiles.com/files/NVIDIA/NCP-ADS/NVIDIA.NCP-ADS.v2026-08-14.q103	

NEW QUESTION: 1
You are implementing a GPU-accelerated ETL pipeline that involves joining two large datasets:
Dataset A: A cuDF DataFrame with 10 million customer records.
Dataset B: A cuDF DataFrame with 100 million transaction records.
The goal is to efficiently perform a join operation to link customer details with transaction data, ensuring that the pipeline remains scalable and performant.
Which of the following is the best approach to optimize the join operation using NVIDIA RAPIDS?
A. Convert both DataFrames to Pandas before performing the join for compatibility
B. Perform the join operation entirely in a CPU-based Spark environment for better stability
C. Use cuDF's .merge() function and ensure both DataFrames have the correct index before joining
D. Store the transaction data as a CSV file and perform joins using SQL queries before loading it into cuDF
Answer: (SHOW ANSWER)

NEW QUESTION: 2
A data scientist is working with a 50 TB dataset consisting of structured logs from IoT devices. The data needs to be cleaned, transformed, and aggregated before training a machine learning model.
Which of the following frameworks would be the most efficient choice for distributed data processing?
A. Python multiprocessing module
B. SQLite
C. Pandas

D. Apache Spark with RAPIDS Accelerator
Answer: D (LEAVE A REPLY)

NEW QUESTION: 3
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. GPU acceleration is beneficial when the dataset can be divided into independent chunks that can be processed in parallel.
B. The dataset must be over 100GB in size to qualify as big data and warrant GPU acceleration.
C. CPU-based machine learning methods are always more effective for small datasets, regardless of the algorithm used.
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 size of the dataset in terms of rows and columns is irrelevant when determining if it qualifies as big data.
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 4
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 = scaler.fit_transform(df)
C. scaler = cuml.preprocessing.StandardScaler()
D. df_scaled = (df - df.min()) / (df.max() - df.min())
E. df_scaled = df.apply(lambda x: x / np.linalg.norm(x))
Answer: C (LEAVE A REPLY)

NEW QUESTION: 5
You need to benchmark GPU-accelerated data science frameworks across both cloud-based and on-premise GPU setups.
Which of the following is the most effective strategy for ensuring consistent and reliable benchmarking results?
A. Running each benchmark multiple times and averaging the results to account for variance in execution time.
B. Using mixed cloud and on-premise GPU instances to test various hardware types and configurations.
C. Running benchmarks in isolated virtual machines to simulate real-world multi-tenant cloud environments.

D. Prioritizing synthetic benchmarks over real-world workloads to get a more controlled performance measurement.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 6
You are processing a dataset of high-resolution images for deep learning training and want to optimize image loading, augmentation, and transformation on an NVIDIA GPU.
Which of the following statements correctly describes the role of NVIDIA DALI (Data Loading Library) in accelerating data preparation?
A. NVIDIA DALI only supports preprocessing images and does not include any functionality for video or text data.
B. NVIDIA DALI does not support data augmentation and is only used for raw data loading.
C. NVIDIA DALI can offload image preprocessing tasks to the GPU, reducing CPU bottlenecks and improving training throughput.
D. NVIDIA DALI is an alternative to cuDF for GPU-accelerated tabular data processing.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 7
Which of the following actions can you perform using DLProf to analyze a deep learning model's performance?
A. Visualize GPU memory utilization over time
B. Automatically adjust the learning rate based on the model's convergence
C. Increase batch size to improve accuracy
D. Modify the training dataset during model execution
Answer: (SHOW ANSWER)

NEW QUESTION: 8
Which NVIDIA technology is specifically designed for accelerating deep learning workloads in the cloud?
A. NVIDIA A100
B. TensorRT
C. NVIDIA Tesla
D. NVIDIA Jetson
Answer: A (LEAVE A REPLY)

NEW QUESTION: 9
You are working on a data science project using NVIDIA RAPIDS on a multi-GPU system. To ensure reproducibility and avoid software versioning conflicts, which of the following is the best approach for managing dependencies?
A. Install all required packages globally on the system using pip install without a virtual environment.

B. Avoid dependency management frameworks and rely on manual tracking of package versions using a text file.
C. Use a Conda environment with RAPIDS-compatible versions of libraries installed using conda install
-c rapidsai -c nvidia.
D. Use a manually compiled CUDA installation alongside system-installed Python libraries to manage GPU dependencies.
Answer: (SHOW ANSWER)

NEW QUESTION: 10
A data scientist is working with datasets ranging from hundreds of megabytes to several terabytes and needs to select the most efficient NVIDIA-accelerated data processing library for optimal memory management and performance.
Which approach is best for selecting the appropriate library for different dataset sizes?
A. Use Pandas for all dataset sizes since Pandas has built-in multi-threading optimizations.
B. Always use Dask regardless of dataset size since Dask automatically scales from small to large datasets.
C. Use cuDF for small datasets and Dask-cuDF for larger datasets that do not fit in a single GPU's memory.
D. Use RAPIDS cuML for handling all data processing tasks, regardless of dataset size.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 11
You are working with a large dataset in a GPU-accelerated environment, and one of the columns, revenue, contains numeric values representing the annual revenue for companies. The revenue values are in the billions of dollars.
Which of the following is the most memory-efficient data type for the revenue column in a cuDF DataFrame?
A. df['revenue'] = df['revenue'].astype('float64')
B. df['revenue'] = df['revenue'].astype('float32')
C. df['revenue'] = df['revenue'].astype('uint32')
D. df['revenue'] = df['revenue'].astype('int64')
Answer: A (LEAVE A REPLY)

NEW QUESTION: 12
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 TensorRT to accelerate random forest model training by optimizing tree-based algorithms.
B. NVIDIA Triton Inference Server to distribute random forest model training across multiple GPUs.
C. NVIDIA DeepStream to preprocess tabular data and optimize random forest model execution.
D. NVIDIA RAPIDS cuML to accelerate random forest training using GPU-optimized implementations.
Answer: (SHOW ANSWER)

NEW QUESTION: 13
A data science team is developing a machine learning pipeline requiring specific CUDA, cuDNN, and RAPIDS versions for compatibility across environments. They need a framework to manage dependencies and version conflicts.
Which approach is best for managing software dependencies using NVIDIA technologies?
A. Manually installing each package and its dependencies using pip
B. Using a single system-wide installation of CUDA and forcing all projects to use the same version
C. Using Conda with NVIDIA Conda channels to manage CUDA and cuDNN dependencies
D. Using only virtual environments (venv) without managing GPU dependencies separately
Answer: C (LEAVE A REPLY)

NEW QUESTION: 14
You are working on a data science project where you need to process a large dataset containing
500 million records. You want to determine whether GPU acceleration would significantly improve performance.
Which of the following factors best indicates that you should use an accelerated computing solution like RAPIDS?
A. The dataset consists of simple arithmetic operations on a few columns and can be processed using vectorized NumPy operations.
B. The dataset is heavily structured but mainly requires text-based analysis using regex-based search and manipulation.
C. The dataset is a structured table with less than 100,000 records and can be handled efficiently with a Pandas DataFrame.
D. The dataset has high-dimensional sparse features and requires complex operations such as nearest neighbor search and clustering.
Answer: (SHOW ANSWER)

NEW QUESTION: 15

You are working on a large-scale social network analysis project using cuGraph. Your goal is to identify the most influential individuals in the network based on their connectivity and reach.
Which of the following cuGraph algorithms would be the most appropriate choice for this task?
A. Connected Components
B. Single Source Shortest Path (SSSP)
C. PageRank
D. Louvain Clustering
Answer: (SHOW ANSWER)

NEW QUESTION: 16
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. cuDF, as it provides GPU-accelerated DataFrame operations similar to Pandas, allowing for efficient processing of large datasets.
B. NumPy, because it is optimized for numerical computing and offers better performance for handling tabular data.
C. Pandas, as it is widely used and supports all common DataFrame operations, even for very large datasets.
D. cuPy, because it provides GPU-accelerated array operations, making it the best option for processing tabular data.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 17
You are tasked with designing and implementing a benchmark to compare the performance of different deep learning frameworks, including TensorFlow, PyTorch, and JAX, using NVIDIA GPUs.
Which of the following is the most effective approach to ensure an accurate and fair comparison?

A. Compare training times only without considering throughput, power efficiency, or memory utilization.
B. Use mixed precision (FP16) training only in TensorFlow to maximize performance while keeping other frameworks at FP32.
C. Run each framework with default settings to compare their out-of-the-box performance without any optimizations.
D. Ensure identical hardware configurations, dataset preprocessing, and model architectures while leveraging NVIDIA's Nsight Systems and DLProf for analysis.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 18
A machine learning engineer is working on a multi-GPU workload using Dask and RAPIDS to process a massive dataset efficiently. However, they notice that GPU utilization is not optimal, and data transfer between GPUs is slowing down computation.
What is the best approach to minimize data transfer overhead and maximize parallel efficiency?
A. Use NVLink to enable high-bandwidth inter-GPU communication and dask_cuda.LocalCUDACluster
B. Load the dataset into a Pandas DataFrame and distribute computations across GPUs manually
C. Use Dask's single-threaded scheduler to avoid multi-threading overhead
D. Transfer data between GPUs using NumPy's .copy() function
Answer: A (LEAVE A REPLY)

NEW QUESTION: 19
A data scientist wants to process a large dataset using multiple GPUs on an NVIDIA-supported system. They decide to use Dask to enable parallelism.
Which of the following steps is most essential for leveraging Dask for multi-GPU scaling?
A. Convert all data into Pandas DataFrames before distributing computations
B. Use dask.array or dask.dataframe with dask_cuda.CUDACluster to distribute computations across multiple GPUs
C. Use ThreadPoolExecutor to manage parallel computations on GPUs
D. Train a deep learning model on a single GPU first, then switch to Dask for scaling
Answer: B (LEAVE A REPLY)

NEW QUESTION: 20
Which of the following is the most efficient way to implement data parallelism using Dask for multi- GPU scaling on an Nvidia platform?
A. Use Dask with the dask_cuda package to distribute computation across GPUs, ensuring that each GPU is responsible for a portion of the data, while using distributed.Client to connect the GPU workers.

B. Use Dask with a single GPU, distributing data across multiple workers within the GPU without considering GPU memory limitations.
C. Use Dask on a single GPU machine with no GPU-specific optimization, treating the system as if it were CPU-only.
D. Use Dask with the dask_gpu package to assign data chunks across GPUs manually, without utilizing the distributed.Client.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 21
You are processing a large dataset using NVIDIA Dask-cuDF to distribute GPU-accelerated computation across multiple nodes. Users report inconsistent execution times, with some jobs taking significantly longer than expected.
Which of the following actions would best help diagnose the performance bottleneck?
A. Use Dask's dashboard and NVTX markers to analyze task execution times and GPU utilization.
B. Reduce the number of Dask workers to minimize parallel execution overhead.
C. Switch to using Pandas with Dask to compare execution speed differences.
D. Limit GPU memory usage to force more frequent spilling to disk and observe performance differences.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 22
A data scientist is working with large-scale datasets in a RAPIDS AI pipeline and needs to efficiently process and organize the data while leveraging GPU acceleration.
Which of the following approaches best ensures optimized processing and memory management when using NVIDIA technologies?
A. Process data using Apache Spark on CPU before transferring results to GPU for model training.
B. Write intermediate data to disk as CSV files before reading them back into RAPIDS AI to avoid memory overflow.
C. Store data in a pandas DataFrame first and then convert it to cuDF only when GPU operations are needed.
D. Use cuDF to store and process data in GPU memory, leveraging its vectorized operations for transformations and aggregations.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 23
A machine learning engineer wants to deploy a GPU-accelerated inference model in a containerized environment while ensuring compatibility with NVIDIA libraries.
Which of the following is the best approach for managing dependencies?

A. Run Docker containers without any special configurations, as Docker automatically detects and utilizes GPUs.
B. Disable the --gpus flag when running Docker containers, as RAPIDS AI libraries do not require explicit GPU selection.
C. Use the official NVIDIA Docker base images (nvidia/cuda) and install RAPIDS AI libraries within the container to ensure GPU compatibility.
D. Install GPU drivers directly inside the container instead of on the host system to avoid dependency conflicts.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 24
Which of the following is the most efficient method for processing big data in a distributed environment using NVIDIA technologies?
A. Using a single-node GPU-based solution for data processing.
B. Utilizing a distributed data processing framework like Apache Spark with GPU acceleration from the NVIDIA RAPIDS library.
C. Relying on traditional CPU-only processing frameworks for big data tasks.
D. Using cloud-based, non-GPU infrastructure for data processing
Answer: B (LEAVE A REPLY)

NEW QUESTION: 25
A data scientist is preparing a dataset containing numerical features with varying scales and distributions. Some features range from 0 to 1, while others have values ranging from -5000 to 5000. The dataset will be used in a machine learning model that relies on gradient-based optimization.
What is the best approach to standardizing the data to ensure uniformity across features?
A. Keep the features as they are since machine learning models can handle varying scales naturally
B. Normalize the dataset by dividing each feature by its maximum absolute value
C. Apply z-score normalization (standardization) to scale features to have a mean of 0 and a standard deviation of 1
D. Use min-max scaling to transform all features into the range [0,1]
Answer: (SHOW ANSWER)

NEW QUESTION: 26
You are working with a dataset where numerical features have different scales. To ensure uniformity across features, you decide to standardize the data using NVIDIA RAPIDS cuML.
Which of the following methods correctly standardizes the data in a GPU-accelerated manner?
A. df = (df - df.mean()) / df.std()

B. df = df.apply(lambda x: (x - x.mean()) / x.std(), axis=1)
C. 1. scaler = cuml.preprocessing.StandardScaler() 2. df = scaler.fit_transform(df)
D. df = (df - df.min()) / (df.max() - df.min())
Answer: C (LEAVE A REPLY)

NEW QUESTION: 27
A data scientist is working with a large dataset that contains string-based numeric values that need to be converted to floating-point numbers for further analysis. The dataset is stored as a cuDF DataFrame, and the scientist needs to ensure the conversion is performed optimally on a GPU.
Which of the following is the best method for converting string-based numeric values to floating-point numbers using NVIDIA-accelerated processing?
A. Convert the cuDF DataFrame to a Pandas DataFrame first, then apply astype(float) and convert it back to cuDF.
B. Use pandas.to_numeric() since pandas automatically handles type conversion.
C. Use cudf.DataFrame.astype(float) to convert string values to floating-point numbers efficiently on a GPU.
D. Use NumPy's astype(float) method after converting the cuDF DataFrame into a NumPy array.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 28
Which of the following best describes the purpose of the NVIDIA TensorRT library?
A. Manages GPU resources for deep learning models
B. Optimizes and accelerates inference of trained models
C. Provides hardware abstraction for AI model development
D. Accelerates training of neural networks
Answer: B (LEAVE A REPLY)

NEW QUESTION: 29
A machine learning engineer is tasked with optimizing an image classification model on a cloud platform. The engineer must select a GPU-accelerated instance that balances cost and performance while ensuring compatibility with frameworks like TensorFlow and PyTorch.
Which instance configuration is the most appropriate choice?
A. A CPU-only instance with 128 GB of RAM for increased data processing speed.
B. A single-core CPU instance with high disk I/O throughput for faster data loading.
C. A cloud instance with NVIDIA A100 GPUs and NVLink support.
D. A cloud instance with integrated graphics rather than dedicated NVIDIA GPUs.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 30
You are building a large-scale AI training pipeline that requires efficient storage and retrieval of structured and unstructured datasets across multiple GPUs.
Which of the following is the best NVIDIA technology to organize and manage datasets at scale?
A. NVIDIA Morpheus for accelerating dataset indexing and retrieval in AI pipelines.
B. NVIDIA Magnum IO for high-performance I/O and dataset storage optimization.
C. NVIDIA Nsight Systems for managing dataset storage and retrieval performance.
D. NVIDIA Clara Imaging for storing structured and unstructured datasets efficiently.
Answer: (SHOW ANSWER)

NEW QUESTION: 31
You are working on a medium-sized dataset (~500,000 rows, 20 columns) and need to perform fast exploratory data analysis (EDA) with filtering, aggregations, and transformations.
Which of the following Python libraries would be the most efficient choice for this task?
A. Vaex
B. PySpark
C. Dask
D. Pandas
Answer: D (LEAVE A REPLY)

NEW QUESTION: 32
You are building an MLOps pipeline for a predictive model that uses tabular data with both categorical and numerical features.
To ensure efficient data processing and optimal model training on an NVIDIA GPU, which of the following data types would be most suitable for a categorical feature representing different product categories?
A. String
B. Int64
C. Int32
D. Float64

Answer: C (LEAVE A REPLY)

NEW QUESTION: 33
You are a data scientist working on a large-scale deep learning project. Your team needs to train a deep neural network (DNN) on terabytes of image data using NVIDIA GPUs in the cloud.
Which of the following approaches will maximize GPU utilization and optimize training time in an NVIDIA- accelerated cloud environment?
A. Avoid using mixed precision training and rely solely on full-precision (FP32) computations for numerical stability.
B. Use NVIDIA NGC containers with TensorFlow or PyTorch and enable NCCL for multi-GPU communication.
C. Use Jupyter Notebooks on a local machine with minimal GPU resources and transfer model checkpoints manually to the cloud for training.
D. Select low-cost, consumer-grade GPUs available in the cloud marketplace to reduce costs.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 34
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. df.memory_usage(deep=True).sum()
C. np.array(df).nbytes
D. sys.getsizeof(df)
Answer: B (LEAVE A REPLY)

NEW QUESTION: 35
A data scientist is preprocessing a dataset containing several types of features:
A timestamp column storing millisecond-resolution timestamps.
A column with binary categorical values (Yes/No).
A column containing large continuous numerical values.
A column containing product category codes ranging from 0 to 5000.
Which of the following data type choices is the most optimal for maximizing GPU processing efficiency using NVIDIA cuDF?
A. Use float64 for timestamps, int8 for binary categorical values, float32 for continuous numerical values, and int32 for product category codes.

B. Use string data type for timestamps, int32 for binary values, float16 for continuous numerical values, and int64 for product category codes.
C. Store timestamps as int64, encode binary values as float16, use float64 for continuous numerical values, and use int8 for product category codes.
D. float32 is the best choice for large continuous numerical values, balancing precision and GPU efficiency.
E. Convert timestamps to datetime64[ms], encode binary values as bool, use float32 for continuous values, and int16 for product category codes.
F. Product category codes (range: 0-5000) fit within int16 (which can hold values from -32,768 to
32,767), making it more memory-efficient than int32.
G. Binary categorical values (Yes/No) should be stored as bool, which takes up minimal space.
Answer: E (LEAVE A REPLY)

NEW QUESTION: 36
You are working on a data science project that involves processing large-scale financial transaction data. You want to optimize data manipulation operations using NVIDIA's RAPIDS cuDF.
Which of the following approaches best leverages NVIDIA technologies for efficient data manipulation?
A. Load the dataset into a Pandas DataFrame and use multi-threading to speed up operations.
B. Use cuDF DataFrames for data manipulation and rely on GPU-accelerated functions like .groupby(),
.merge(), and .applymap().
C. Preprocess the data using Apache Spark's CPU-based DataFrame API before transferring it to a GPU for machine learning.
D. Convert the dataset into a SQLite database and execute SQL queries to perform data transformations.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 37
A data scientist is using NVIDIA RAPIDS cuDF to process a large dataset of customer transactions.
The dataset contains numerical, categorical, and timestamp-based features.
To optimize memory usage and performance on NVIDIA GPUs, which approach should they take when selecting data types?
A. Convert all timestamp features into object (string) format to maintain readability and ensure compatibility with GPU processing.

B. Avoid downcasting integer columns, as lower-bit integer types (e.g., int8) are not supported in GPU- accelerated computations.
C. Convert categorical variables into cuDF categorical data types and downcast numerical columns to the smallest possible precision without losing information.
D. Store all numerical columns as float64 to preserve maximum precision, even if lower precision suffices.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 38
You are processing large-scale datasets in Dask-cuDF and observe that your computation involves excessive data shuffling, which slows down performance. You decide to implement data caching to reduce shuffle overhead.
Which of the following best describes a technique to reduce shuffle costs in a Dask-cuDF workflow?
A. Disabling lazy execution using dask.config.set(lazy=False) forces all computations to execute eagerly, thereby avoiding shuffle.
B. Persisting intermediate results in distributed GPU memory using df.persist() ensures that Dask does not recompute partitions, reducing shuffle overhead.
C. Splitting the DataFrame into multiple smaller DataFrames and recomputing each separately reduces shuffle operations in Dask-cuDF.
D. Using .compute() on each partition immediately forces Dask to materialize results and store them in memory, ensuring shuffle operations are avoided.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 39
A data scientist is using RAPIDS cuML to build a predictive model on a large dataset containing numerical and categorical features.
To optimize feature engineering for accelerated GPU processing, which of the following is the best approach?
A. Convert all numerical features to float64 to maximize precision for feature transformation.
B. Convert all categorical variables to string format for efficient processing in cuDF.
C. Apply one-hot encoding on high-cardinality categorical features directly in cuDF without any transformation.
D. Normalize numerical features using min-max scaling implemented with cuDF, leveraging GPU acceleration.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 40
Which of the following are key advantages of using cuGraph for analyzing graph data in GPU- accelerated environments? (Select two)

A. cuGraph does not support distributed graph processing and is only suitable for single-node systems.
B. cuGraph can efficiently handle larger graphs than traditional CPU-based methods, providing significant performance improvements.
C. cuGraph only works on small-scale graph datasets that can fit into memory.
D. cuGraph supports various graph algorithms, including PageRank, shortest path, and community detection, leveraging GPU parallelism.
E. cuGraph only works with cloud-based computing environments and is not optimized for local GPUs.
Answer: (SHOW ANSWER)

NEW QUESTION: 41
You are working on a large-scale graph analysis project using NVIDIA cuGraph for accelerated computations. Your dataset consists of millions of nodes and edges representing social network interactions. You need to efficiently compute PageRank while minimizing memory usage.
Which of the following techniques would be the most effective?
A. Use Pandas DataFrames to manage the graph structure and leverage Pandas indexing for efficient queries.
B. Manually unroll loops and precompute node ranks in a NumPy array before feeding them into cuGraph.
C. Use cuGraph's sparse matrix representation to store graph data and perform computations efficiently.
D. Convert the graph into an adjacency list format and store it in Python dictionaries for faster lookups.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 42
A research team is analyzing a large transportation network and wants to identify groups of locations that are closely connected based on travel routes.
Which NVIDIA cuGraph function would be the most effective for this task?
A. Use cuGraph's pagerank() to rank the most connected locations.
B. Apply cuGraph's bfs() (Breadth-First Search) to find clusters of locations.
C. Use cuGraph's louvain() function to detect communities within the graph.
D. Use cuGraph's shortest_path() to compute all shortest paths and derive clusters manually.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 43
You have developed a deep learning model using TensorFlow and trained it on an NVIDIA A100 GPU. The model is deployed in production and serves real-time inference requests.

However, the inference latency is high, and you need to optimize performance without retraining the model.
Which of the following approaches is the most effective for optimizing inference performance using NVIDIA technologies?
A. Convert the model to ONNX format and use TensorRT for inference optimization.
B. Enable mixed precision training and retrain the model to improve inference speed.
C. Implement data augmentation techniques to improve inference efficiency.
D. Reduce the batch size to decrease computational overhead and improve latency.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 44
You are setting up a GPU-accelerated data science environment on a cloud-based instance that utilizes NVIDIA GPUs.
To ensure compatibility between CUDA, RAPIDS, and Python libraries, which of the following is the most effective approach for managing dependencies and avoiding version conflicts?
A. Use Conda environments with version-pinned dependencies to create an isolated environment for RAPIDS and CUDA
B. Install RAPIDS, PyTorch, TensorFlow, and other libraries in the same Conda base environment for simplified access
C. Use a virtual machine with all dependencies pre-installed to avoid any software conflicts
D. Manually install required libraries in a system-wide Python environment using pip install without version specifications
Answer: A (LEAVE A REPLY)

NEW QUESTION: 45
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 of product reviews and customer sentiments from an e-commerce website
B. A dataset with global temperature trends over the past decade
C. A dataset containing transaction history and customer profiles from a retail company
D. A public dataset of annual GDP per country
Answer: C (LEAVE A REPLY)

NEW QUESTION: 46
A machine learning engineer is benchmarking a GPU-accelerated pipeline for data preprocessing and training. The pipeline consists of large-scale data transformations using RAPIDS cuDF and training a model with TensorFlow. The engineer wants to ensure that GPU utilization is maximized throughout the workflow.

Which approach is most effective in achieving this?
A. Reduce the dataset size to fit entirely in GPU memory, even if it leads to data loss.
B. Precompute all features on the CPU before loading them into the GPU for training.
C. Use a single GPU to avoid overhead associated with multi-GPU synchronization.
D. Pipeline execution should be asynchronous, overlapping data preprocessing with model training.
Answer: (SHOW ANSWER)

NEW QUESTION: 47
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. Use dask.array instead of dask_cudf because it provides better performance for structured tabular data.
B. Set dask.config.set({'distributed.worker.memory.target': 0.9}) to allocate 90% of the total CPU memory for GPU operations.
C. Use dask.persist() instead of dask.compute() to force immediate execution of tasks before distribution to GPUs.
D. Use dask_cudf.read_parquet() with split_row_groups=True to evenly distribute data across GPUs.
Answer: (SHOW ANSWER)

NEW QUESTION: 48
You are working with a social network dataset containing millions of user interactions and need to identify influential users based on their connectivity and interactions.
Which approach using NVIDIA's cuGraph library is the most appropriate for this task?
A. Apply cuGraph's K-Means clustering to group users with similar connectivity patterns.
B. Use cuGraph's DBSCAN clustering to detect communities in the social network.
C. Run a breadth-first search (BFS) on the entire graph to find the most influential users.
D. Use cuGraph's PageRank algorithm to rank users based on their importance in the network.

Answer: D (LEAVE A REPLY)

NEW QUESTION: 49
A data scientist is setting up a RAPIDS AI environment for a machine learning project that requires CUDA-enabled libraries and specific package versions to avoid conflicts.
Which of the following approaches best ensures a stable and reproducible environment while leveraging NVIDIA technologies?
A. Install CUDA and RAPIDS AI libraries using OS-level package managers such as apt or yum instead of Conda or Docker.
B. Use Conda with the rapidsai channel to create an isolated environment that includes cuDF, cuML, and other GPU-accelerated libraries.
C. Use a system-wide Python installation and install packages globally to ensure consistency across different users.
D. Manually install each required package with pip to ensure the latest versions are used without interference from Conda.
Answer: (SHOW ANSWER)

NEW QUESTION: 50
You are working with a large dataset containing 500 million records stored as a parquet file. Your task is to perform data filtering, aggregation, and transformation as efficiently as possible.
Which of the following approaches would be the best choice for accelerated data manipulation using NVIDIA technologies?
A. Use multiprocessing in Python to parallelize pandas operations across CPU cores
B. Convert the dataset to a NumPy array and use standard Python loops
C. Load the dataset into pandas and use apply() for transformations
D. Use RAPIDS cuDF to load, filter, and transform the dataset on the GPU
Answer: (SHOW ANSWER)

NEW QUESTION: 51
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 RAPIDS cuDF to transform image data into tabular format for analysis.
C. NVIDIA Triton Inference Server to preprocess the dataset before model training.
D. NVIDIA Nsight Compute to optimize image dataset processing.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 52
You are working with a cuDF DataFrame and need to convert a column named sales from float64 to int32 to save memory.
Which of the following is the correct and most efficient way to perform this conversion in cuDF?
A. df['sales'] = df['sales'].astype('int32')
B. df['sales'] = df['sales'].to_numeric('int32')
C. df['sales'].convert_dtypes('int32')
D. df['sales'].apply(lambda x: int(x))
Answer: (SHOW ANSWER)

NEW QUESTION: 53
You are working with a large dataset containing customer transactions and want to perform exploratory data analysis (EDA) efficiently. Given the dataset's size, you decide to use NVIDIA RAPIDS to accelerate the process.
Which of the following approaches is the most effective for conducting EDA using NVIDIA technologies?
A. Perform SQL queries on a CPU-based database for initial data analysis before GPU acceleration
B. Use RAPIDS cuDF to perform fast dataframe operations and visualize results with cuXfilter
C. Use TensorFlow and Keras to preprocess the data before performing EDA
D. Load the dataset into Pandas and use Matplotlib for visualization
Answer: (SHOW ANSWER)

NEW QUESTION: 54
You are implementing a Dask-based solution for distributed data parallelism across a multi-GPU system.
Which configuration steps would ensure effective use of GPUs for parallel computation? (Select two)
A. Use Dask's Cluster class with the distributed scheduler and specify CPU cores only for GPU workloads
B. Create a LocalCUDACluster and manually specify the GPUs you want to use for each Dask worker
C. Use dask_cuda's LocalCUDACluster with proper GPU memory management to handle multiple GPUs
D. Use dask_cudf to convert DataFrame computations into GPU-accelerated operations using cuDF
E. Use dask_cuda's LocalCUDACluster and let Dask automatically allocate GPUs without any configuration

Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 55
You are working on a financial fraud detection system using NVIDIA RAPIDS cuML. You have a large time-series dataset of transaction amounts over time, and you need to identify anomalies such as unusual spikes in transactions.
Which of the following approaches is the most appropriate for detecting anomalies using NVIDIA technologies?
A. Apply cuML's PCA-based anomaly detection to detect outliers in the principal component space
B. Run standard Pandas-based anomaly detection methods on CPU for better precision
C. Train a deep learning model with TensorFlow and deploy it using NVIDIA Triton Inference Server for anomaly detection
D. Use cuML's DBSCAN clustering to identify anomalies based on density differences
Answer: (SHOW ANSWER)

NEW QUESTION: 56
You are developing an accelerated ETL workflow that requires data transformations such as filtering, aggregating, and joining large datasets. You decide to leverage NVIDIA GPUs to accelerate the transformation phase of your ETL pipeline.
Which of the following approaches will provide the greatest performance improvements when working with large-scale tabular datasets?
A. Performing transformations using SQL-based queries on CPU
B. Relying on traditional pandas for in-memory transformations
C. Using RAPIDS cuDF to perform transformations on a GPU
D. Using TensorFlow for data transformation tasks
Answer: C (LEAVE A REPLY)

NEW QUESTION: 57
You are working with a large dataset containing millions of rows, and you need to store it efficiently for fast read and write operations while maintaining compatibility with CuDF and pandas.
Which of the following file formats is the best choice for efficient columnar storage and GPU-accelerated processing?
A. TXT
B. CSV
C. JSON
D. Parquet
Answer: D (LEAVE A REPLY)

NEW QUESTION: 58

A data scientist is analyzing sales data for an e-commerce company that experiences strong seasonal trends (e.g., increased sales during holiday seasons). The goal is to accurately forecast future sales using GPU-accelerated data science techniques.
Which approach would be the most effective?
A. Use a Seasonal Autoregressive Integrated Moving Average (SARIMA) model and optimize it using NVIDIA RAPIDS.
B. Compute a rolling average and manually adjust for seasonality based on previous peak sales months.
C. Apply k-Means clustering to identify seasonal patterns and extrapolate future values.
D. Train a Decision Tree model using cuML to classify future sales trends.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 59
Which of the following is the best approach for performing benchmarking and optimizing GPU- accelerated workflows for MLOps using Nvidia technologies?
A. Use TensorRT to optimize deep learning models by converting them into highly optimized inference engines, allowing faster execution with lower latency.
B. Use Nvidia's nvprof tool to profile GPU resource usage and identify bottlenecks, then adjust the batch size to optimize throughput.
C. Use Nvidia's nsight tools to benchmark only the model training phase and ignore the inference phase, as training is the primary bottleneck.
D. Rely exclusively on the nvidia-smi tool for monitoring GPU utilization and memory usage across multiple GPUs without making any other performance adjustments.
Answer: (SHOW ANSWER)

NEW QUESTION: 60
A data scientist is using an NVIDIA RAPIDS-based data processing pipeline on a GPU cluster. They notice that the pipeline is not performing as expected and suspect a bottleneck.
Which of the following approaches would best help identify the source of the bottleneck?
A. Rely on nvidia-smi alone to check GPU utilization without considering other profiling tools.
B. Disable GPU parallelism and run computations on the CPU to compare performance.
C. Increase the batch size of data being processed without analyzing memory utilization.
D. Use nsys (NVIDIA Nsight Systems) to profile GPU kernel execution and memory transfer times.
Answer: (SHOW ANSWER)

NEW QUESTION: 61

You are a data scientist working on a large-scale deep learning project that requires significant computational resources. You have the option to run your workloads on a cloud-based GPU instance.
Which of the following statements best describes a key benefit of using cloud-based GPUs for your workload?
A. Cloud-based GPUs enable scalable resource allocation, allowing you to dynamically increase or decrease GPU instances as needed.
B. Cloud-based GPUs provide consistent and predictable performance, identical to on-premise dedicated GPUs.
C. Cloud-based GPUs are always more cost-effective than on-premise GPUs, regardless of workload size and duration.
D. Cloud-based GPUs eliminate all data transfer bottlenecks and latencies when training models on large datasets.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 62
A company is processing large log files from a cloud application, accumulating over 5TB of data daily. The data processing pipeline must be GPU-accelerated to extract insights quickly.
Which of the following is the most effective approach to handle high-volume log processing using NVIDIA technologies?
A. Use cuDF with explicit memory management to load and process the entire dataset into a single GPU.
B. Leverage Dask-cuDF to distribute the dataset across multiple GPUs, ensuring efficient parallel processing.
C. Store logs as Pandas DataFrames and use multiprocessing to parallelize operations across CPU cores.
D. Use RAPIDS cuML for performing log file processing, taking advantage of its optimized ML algorithms.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 63

You are working with a 10-terabyte dataset containing structured and unstructured data. Your goal is to perform ETL (Extract, Transform, Load) operations efficiently while leveraging GPU acceleration for distributed processing.
Which of the following frameworks would be the best choice for handling this workload?
A. RAPIDS + Dask for distributed GPU-accelerated ETL
B. Hadoop MapReduce
C. Apache Spark with its default CPU-based execution
D. Pandas with multiprocessing
Answer: A (LEAVE A REPLY)

NEW QUESTION: 64
A data engineer is tasked with processing a 5 TB dataset stored in Apache Parquet format. The dataset consists of user activity logs and needs to be filtered, aggregated, and processed for feature engineering before training an ML model. The engineer is deciding between Dask and Apache Spark.
Which statement best describes a key difference between the two frameworks?
A. Dask is a more lightweight solution, often preferred for Python-centric workflows, whereas Spark provides broader ecosystem integrations and supports SQL-like operations natively.
B. Spark cannot utilize GPUs, whereas Dask has built-in GPU acceleration by default.
C. Dask is optimized for in-memory processing, while Spark requires disk-based storage for computation.
D. Spark is better suited for structured and semi-structured data, while Dask excels at unstructured data processing.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 65
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 a traditional SQL database to compute statistics and then transfer results to the GPU
B. Convert the dataset into a PyTorch tensor and use PyTorch's statistical methods
C. Use NumPy's statistical functions, such as numpy.mean() and numpy.var()
D. Use cuDF's built-in statistical functions like .mean(), .median(), and .var()
Answer: (SHOW ANSWER)

NEW QUESTION: 66

You are tasked with processing a large dataset of 100 million records for a deep learning project using NVIDIA technologies. You need to determine the most efficient data processing library for this task to maximize performance and reduce processing time.
Which of the following libraries is best suited for this task?
A. pandas
B. PySpark
C. Dask
D. cuDF
Answer: (SHOW ANSWER)

NEW QUESTION: 67
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. Min-Max Scaling
B. Robust Scaling
C. Z-score Standardization
D. Log Transformation
Answer: (SHOW ANSWER)

NEW QUESTION: 68
Which of the following techniques are commonly used to identify and acquire datasets for machine learning and data science projects? (Select three)
A. Web scraping
B. Public APIs
C. Manual data entry
D. Crowdsourcing
E. Data augmentation
Answer: (SHOW ANSWER)

NEW QUESTION: 69
You are working on a large-scale data processing pipeline that involves multi-GPU acceleration using Dask. The dataset is too large to fit into the memory of a single GPU, so you decide to distribute the workload across multiple GPUs using Dask-CUDA.
Which of the following steps is necessary to implement efficient data parallelism across multiple GPUs in a Dask-based workflow?
A. Use dask.dataframe.read_parquet() to load the dataset and let Dask automatically distribute computations across multiple GPUs.
B. Explicitly initialize a LocalCUDACluster with multiple workers, ensuring each worker is assigned a single GPU.

C. Convert Dask DataFrames into Pandas DataFrames to leverage GPU acceleration through Pandas' built-in multi-threading support.
D. Use the DaskExecutor from RAPIDS cuML to offload data processing tasks to distributed GPU workers.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 70
A data scientist is deploying a deep learning model to production using an NVIDIA GPU-powered inference server. They want to ensure low latency and high throughput while maintaining model accuracy.
Which of the following deployment strategies would be most effective?
A. Running inference on a CPU instead of a GPU to reduce power consumption
B. Deploying a model without batching to ensure real-time responses
C. Using NVIDIA Triton Inference Server with model ensemble optimization
D. Disabling mixed-precision inference to avoid accuracy loss
Answer: (SHOW ANSWER)

NEW QUESTION: 71
A data scientist is working on a machine learning model for fraud detection. Due to the limited size of the dataset, they decide to generate synthetic data using NVIDIA RAPIDS AI and cuDF.
Which of the following approaches is the most efficient and effective for generating synthetic data while ensuring compatibility with RAPIDS AI workflows?
A. Use cuDF DataFrame operations to create new synthetic samples by applying random transformations (e.g., noise injection, permutation) to the existing dataset.
B. Train a generative adversarial network (GAN) using PyTorch and then use the generated samples in RAPIDS AI without any additional processing.
C. Use numpy and pandas to generate synthetic data, then convert the DataFrame to cuDF before using it in RAPIDS AI workflows.
D. Use the RAPIDS cuML library to directly generate synthetic tabular data with controlled statistical properties.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 72
Which of the following Nvidia technologies is primarily used for performing benchmarking and optimizing GPU-accelerated deep learning workflows, especially focusing on model training performance?
A. Nvidia Nsight Systems
B. Nvidia CUDA
C. Nvidia Triton Inference Server
D. Nvidia DeepStream SDK

Answer: A (LEAVE A REPLY)

NEW QUESTION: 73
A data scientist is working on a dataset where the numerical features have different ranges, and they need to ensure uniformity across features before training a machine learning model.
Which of the following approaches, utilizing NVIDIA technologies, would best achieve this goal?
A. Apply cuML's RobustScaler() to center the data using median and scale using the interquartile range.
B. Use cuML's PCA to directly remove the need for standardization by reducing dimensionality.
C. Use cuML's StandardScaler() to transform the features to have zero mean and unit variance.
D. Apply cuDF's normalize() function to scale each feature between 0 and 1.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 74
A data scientist is analyzing a large time-series dataset containing stock price movements of thousands of companies over a decade. The dataset is stored as a cuDF DataFrame and contains millions of rows. The scientist wants to visualize trends and patterns interactively while leveraging GPU acceleration.
Which of the following approaches is the most efficient for visualizing this time-series data?
A. Convert the cuDF DataFrame to Pandas and use matplotlib for plotting.
B. Use cuXfilter with a cuDF DataFrame to generate interactive visualizations directly on the GPU.
C. Use matplotlib with plt.plot() while applying df.to_pandas() to convert data.
D. Use seaborn with a sampled subset of the dataset to generate line plots.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 75
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. cuDF
B. Dask
C. PyTorch
D. CUDA Toolkit

Answer: (SHOW ANSWER)

NEW QUESTION: 76
You are working on an MLOps workflow that loads a dataset into GPU memory for model training using RAPIDS cuDF. Before performing transformations, you want to verify that the dataset will fit into available GPU memory.
Which of the following methods provides the most accurate estimate of dataset memory consumption in a RAPIDS cudf.DataFrame?
A. cudf_df.memory_usage(deep=True).sum()
B. cudf_df.__sizeof__()
C. cudf_df.memory_usage().sum()
D. cudf_df.to_pandas().memory_usage(deep=True).sum()
Answer: C (LEAVE A REPLY)

NEW QUESTION: 77
You are analyzing a dataset that contains missing values.
Which of the following techniques is most appropriate when dealing with missing numerical data in a dataset, ensuring minimal impact on model performance?
A. Replacing missing values with a constant value (e.g., zero)
B. Removing rows with missing values
C. Using k-nearest neighbors (KNN) imputation
D. Replacing missing values with the mean of the column
Answer: C (LEAVE A REPLY)

NEW QUESTION: 78
You are working on a data science project that requires augmenting a dataset using synthetic data.
You are utilizing cuDF and NVIDIA RAPIDS to speed up the data generation process.
Which of the following methods is the most effective way to generate synthetic data using cuDF in a RAPIDS workflow?
A. Use cudf.DataFrame.sample() to duplicate random rows and create synthetic data.

B. Use cuDF to manipulate data distributions and generate new data points based on existing features.
C. Use cudf.DataFrame.applymap() to create new synthetic features through complex mathematical functions.
Answer: (SHOW ANSWER)

NEW QUESTION: 79
A team of data scientists needs to deploy a machine learning model that depends on specific versions of CUDA and TensorFlow, ensuring it runs consistently across different machines without manually configuring each system.
Which of the following approaches best ensures consistency while leveraging NVIDIA GPUs?
A. Running the model in a local Python virtual environment and copying dependencies manually
B. Using NVIDIA Docker (nvidia-docker) to containerize the model and manage GPU dependencies
C. Using Docker without GPU support and relying on CPU fallback when running TensorFlow
D. Compiling all dependencies into the host machine and using system-wide installations
Answer: B (LEAVE A REPLY)

NEW QUESTION: 80
You are working on a structured dataset of around 10GB and need to perform exploratory data analysis (EDA), feature engineering, and filtering operations efficiently using NVIDIA technologies. The dataset fits into a single GPU's memory.
Which data processing library should you use to achieve the best performance?
A. Spark with RAPIDS Accelerator
B. cuDF
C. pandas
D. Dask DataFrame with Dask-CUDA
Answer: B (LEAVE A REPLY)

NEW QUESTION: 81
A financial analyst wants to create an interactive GPU-accelerated dashboard to visualize stock price movements in real-time.
Which NVIDIA-supported tool is best suited for this purpose?
A. Use Plotly Dash with RAPIDS cuDF to create an interactive GPU-powered dashboard.
B. Convert the stock price dataset into a NumPy array and visualize it using Seaborn's line plot.
C. Precompute the time-series visualization with Dask and display it in a static HTML page.
D. Rely on Matplotlib to generate static plots and update them every minute with a loop.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 82
You are working with a GPU-based cloud environment and need to optimize the memory usage for a dataset that contains a column item_id representing unique product IDs. The item_id values are large integers, and there are over 10 million distinct product IDs.
Which of the following is the most memory-efficient data type choice for this column?
A. df['item_id'] = df['item_id'].astype('int64')
B. df['item_id'] = df['item_id'].astype('int32')
C. df['item_id'] = df['item_id'].astype('string')
D. df['item_id'] = df['item_id'].astype('float64')
Answer: (SHOW ANSWER)

NEW QUESTION: 83
You are setting up a GPU-accelerated data science environment that includes NVIDIA RAPIDS, PyTorch, TensorFlow, and other libraries for machine learning and data processing.
Given that these frameworks have different dependencies and version requirements, what is the best approach to avoid software conflicts while ensuring reproducibility across multiple environments?
A. Manually download and compile each library from source to guarantee compatibility across all versions.
B. Install all packages globally using pip on the system-wide Python installation to ensure consistency.
C. Use a single Docker container with the latest versions of all dependencies installed system-wide.
D. Use Conda to create isolated virtual environments for each project and install dependencies via conda-forge or NVIDIA channels.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 84
You are tasked with optimizing the performance of an MLOps pipeline that uses GPU-accelerated workflows. After running initial benchmarks, you notice that the training time is higher than expected, despite the use of multiple GPUs.
What are the best strategies to optimize the GPU-accelerated workflow in this case? (Select two)
A. Disable gradient accumulation when using multi-GPU setups to increase communication efficiency.
B. Increase the batch size to better utilize the multiple GPUs and reduce the number of updates to the model during training.

C. Ensure that the model is distributed evenly across GPUs to prevent some GPUs from being underutilized.
D. Ensure efficient multi-GPU communication and synchronization strategies, such as using NCCL for distributed training.
E. Reduce the number of GPUs used and focus on fine-tuning the hyperparameters for optimal performance on a single GPU.
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 85
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. Implement a custom shuffle partitioning scheme using NVIDIA DALI for more control over data partitioning during shuffle operations.
C. Use RAPIDS cuDF to repartition the data in the GPU memory after each shuffle.
D. Store data in HDFS before performing shuffle operations to reduce GPU memory overhead.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 86
A data science team wants to leverage GPU acceleration for detecting anomalies in a massive IoT sensor dataset that is continuously streaming.
Which of the following NVIDIA-supported methods would be the best for handling real-time anomaly detection in a high-throughput environment?
A. Use RAPIDS cuML's GPU-accelerated DBSCAN to cluster the sensor readings and classify outliers as anomalies.
B. Utilize NVIDIA Morpheus with deep learning-based anomaly detection to process high-throughput log data efficiently.
C. Implement a GPU-accelerated spectral residual anomaly detection model using NVIDIA Merlin for rapid feature selection.
D. Deploy a TensorRT-optimized Transformer model to process the time-series data in parallel for real- time anomaly detection.
Answer: (SHOW ANSWER)

NEW QUESTION: 87
You are processing a multi-terabyte dataset in CuDF and want to optimize query performance and storage efficiency.

Which approach should you follow to ensure that the dataset remains efficiently partitioned and easily accessible?
A. Split the dataset into multiple files based on a logical partitioning key, such as a date column.
B. Store the entire dataset in a single large file to minimize the number of files in the directory.
C. Use pandas for large-scale processing instead of CuDF since pandas is more stable for big data processing.
D. Convert all numeric columns to float64 for higher precision, even if float32 is sufficient.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 88
You are working on a large dataset (several terabytes in size) and need to perform data preprocessing, filtering, and transformations before training a machine learning model. Given the dataset size and the requirement to optimize for GPU acceleration using NVIDIA technologies, which of the following is the most appropriate data processing library to use?
A. Dask DataFrame with Dask-CUDA
B. pandas
C. Modin with Ray backend
D. NumPy with CuPy acceleration
Answer: A (LEAVE A REPLY)

NEW QUESTION: 89
A data scientist is working on a machine learning model for fraud detection. Due to the limited size of the dataset, they decide to generate synthetic data using NVIDIA RAPIDS AI and cuDF.
Which of the following approaches is the most efficient and effective for generating synthetic data while ensuring compatibility with RAPIDS AI workflows?
A. Manually generate random values using Python's built-in random module and load them into a cuDF DataFrame.
B. Use numpy and pandas to generate synthetic data, then convert the DataFrame to cuDF before using it in RAPIDS AI workflows.
C. Use the RAPIDS cuML library to directly generate synthetic tabular data with controlled statistical properties.
D. Use cuDF DataFrame operations to create new synthetic samples by applying random transformations (e.g., noise injection, permutation) to the existing dataset.
Answer: (SHOW ANSWER)

NEW QUESTION: 90
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. Perform all transformations using Pandas DataFrames and then use multiprocessing to parallelize the workload on CPUs.
B. Use NVIDIA RAPIDS cuDF for data transformations and Dask-cuDF for parallelized processing across multiple GPUs.
C. Use Apache Spark with CPU-based processing instead of leveraging GPU acceleration.
D. Implement ETL processes using only SQL-based transformations within a relational database system.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 91
You are working with a large dataset containing missing values, and you need to clean and preprocess the data efficiently.
Which of the following methods provides the best performance when handling missing values in a GPU- accelerated EDA workflow using RAPIDS?
A. Ignore missing values since GPU acceleration can handle incomplete data without performance degradation.
B. Drop all missing values using df.dropna() in Pandas before using RAPIDS.
C. Use pandas.DataFrame.fillna() on the dataset before converting it to a cuDF DataFrame.
D. Convert the dataset to a cuDF DataFrame and use cudf.DataFrame.fillna() to fill missing values.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 92
You are training a deep learning model for image classification and want to optimize its hyperparameters, including learning rate, batch size, and number of layers.
Which of the following techniques is the most effective for efficiently searching through a high- dimensional hyperparameter space?

A. Gradient Descent
B. Bayesian Optimization
C. Random Search
D. Grid Search
Answer: (SHOW ANSWER)

NEW QUESTION: 93
A data scientist needs to process a dataset containing 10 million records, performing transformations and exploratory data analysis (EDA). The processing needs to be efficient but does not require high- performance multi-GPU execution.
Which of the following libraries provides the best balance between usability and performance?
A. Dask DataFrame, since it automatically parallelizes computations even when the dataset fits in memory.
B. cuDF, since GPU acceleration will still provide a speedup even for moderately sized datasets.
C. Spark DataFrame, as it is optimized for distributed processing and scales well even for 10 million records.
D. Pandas, as it provides a simple API and works well for datasets that fit within system memory.
Answer: (SHOW ANSWER)

NEW QUESTION: 94
You are consulting for a retail company that collects data from daily sales transactions, customer interactions, and inventory tracking across multiple locations. They are unsure whether their dataset qualifies as big data and which processing method would be most suitable.
Which of the following characteristics best indicate that the dataset requires big data processing and acceleration techniques?
A. The dataset contains more than 1 million records, making it impossible to process using pandas
B. The dataset includes a mix of numerical and categorical variables, requiring additional preprocessing
C. The dataset is stored in multiple relational database tables, making querying inefficient
D. The dataset exceeds the memory capacity of a single machine and requires distributed or GPU- accelerated processing
Answer: (SHOW ANSWER)

NEW QUESTION: 95
In a typical MLOps pipeline, which of the following practices are essential to ensuring robust deployment and monitoring of machine learning models in production? (Select two)

A. Use of manual intervention for every model update to ensure accuracy.
B. Continuous integration and continuous deployment (CI/CD) pipelines for model updates.
C. Automated hyperparameter tuning during inference to optimize model performance.
D. Post-deployment data drift detection to assess model performance degradation.
Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 96
When scaling a distributed data processing framework using NVIDIA GPU technology for big data processing, which of the following factors is most critical to optimize performance?
A. Maximizing the amount of data transferred between GPUs for faster processing.
B. Ensuring the proper configuration of GPU resources across all nodes in the distributed system.
C. Limiting the number of GPU nodes used in the cluster to avoid complexity.
D. Using more CPU cores to handle computation-heavy tasks.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 97
You are working with cuGraph to analyze a large social network dataset where users are represented as nodes, and their connections (friendships) are represented as edges. You need to determine the most efficient way to store and process the graph in cuGraph for high-performance analytics.
Which of the following graph representations is best suited for efficient processing in cuGraph?
A. Compressed Sparse Row (CSR) format
B. Adjacency List representation using Python dictionaries
C. Edge List stored in a Pandas DataFrame
D. Incidence Matrix representation
Answer: A (LEAVE A REPLY)

NEW QUESTION: 98
A data scientist is training a deep learning model on an NVIDIA GPU and wants to profile the model to identify performance bottlenecks. The scientist chooses to use NVIDIA DLProf.
Which of the following steps is the most effective way to profile the model using DLProf?
A. Modify the training script to manually insert timing functions for each layer and compare execution times.
B. Run the model using dlprof --mode profile to collect performance metrics and generate a report.
C. Rely on general CPU profiling tools like perf and gprof to analyze GPU performance.
D. Use nvprof instead of DLProf since it provides more detailed profiling for deep learning workloads.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 99
You are tasked with implementing data caching to reduce shuffle in an accelerated machine learning pipeline using NVIDIA technologies. You need to cache intermediate results after a shuffle operation in a distributed setting.
Which of the following is the best approach to minimize shuffle overhead and maximize performance?
A. Cache the data in GPU memory using RAPIDS cuDF for faster access, and leverage GPU-based partitioning to reduce shuffle size.
B. Implement Spark's default disk caching to store shuffle results, allowing the GPU to access disk data directly.
C. Use DALI to perform preprocessing and cache the output before the shuffle operation, thereby eliminating the need for shuffle.
D. Use RAPIDS cuDF to cache the shuffled data on disk and then re-load it from disk during subsequent stages.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 100
You are working on a data science project that requires processing a large-scale dataset stored in CSV format. The dataset contains hundreds of millions of rows, and you want to load it efficiently into NVIDIA RAPIDS cuDF for accelerated processing on a GPU.
Which of the following approaches is the most optimal way to load the dataset?
A. import dask_cudf 2. df = dask_cudf.read_csv("large_dataset.csv")
B. import cudf 2. df = cudf.read_csv("large_dataset.csv", chunksize=100000)
C. import pandas as pd 2. df = pd.read_csv("large_dataset.csv")
D. import cudf 2. df = cudf.DataFrame.from_pandas(pd.read_csv("large_dataset.csv"))
Answer: A (LEAVE A REPLY)

NEW QUESTION: 101
You are analyzing a large-scale transportation network using cuGraph and notice that query times are longer than expected when running graph algorithms.
What is the best way to optimize graph processing performance using GPU-accelerated tools?
A. Convert the graph to CSR (Compressed Sparse Row) format before running computations to improve memory efficiency.
B. Use cugraph.to_directed() to convert the graph into a directed format, which improves GPU parallelism.
C. Store the graph in COO (Coordinate List) format instead of CSR (Compressed Sparse Row) format for faster traversal.

D. Use cugraph.filter_unconnected_nodes() to remove unconnected nodes before processing.
Answer: (SHOW ANSWER)

NEW QUESTION: 102
A research team is analyzing large-scale social interactions and wants to identify strongly connected communities within a massive graph dataset using NVIDIA's cuGraph library. Which method would be the most efficient for this task?
A. Apply cuGraph's Dijkstra's algorithm to find the shortest paths between all nodes and group them into communities.
B. Apply cuGraph's Label Propagation Algorithm (LPA) to divide the graph into communities without predefining the number of clusters.
C. Use cuGraph's Louvain method to detect hierarchical communities based on modularity optimization.
D. Run the cuGraph PageRank algorithm and classify nodes with high scores as community leaders.
Answer: (SHOW ANSWER)

NEW QUESTION: 103
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_csv("data.parquet")
B. df = cudf.read_parquet("data.parquet")
C. df = pd.read_parquet("data.parquet")
D. df = cudf.to_gpu(pd.read_parquet("data.parquet"))
Answer: (SHOW ANSWER)