NVIDIA
NCP-AII
NVIDIA AI Infrastructure

Exams Boost

Boost Up Your Career

For More Information – Visit link below:

https://www.examsboost.com/

Product Version

✓ Up to Date products, reliable and verified.
✓ Questions and Answers in PDF Format.

Latest Version: 6.0

Question: 1

A GPU in your AI server consistently overheats during inference workloads. You've ruled out inadequate cooling and software bugs.
Running 'nvidia-smi' shows high power draw even when idle. Which of the following hardware issues are the most likely causes?

A. Degraded thermal paste between the GPU die and the heatsink.
B. A failing voltage regulator module (VRM) on the GPU board, causing excessive power leakage.
C. Incorrectly seated GPU in the PCIe slot, leading to poor power delivery.
D. A BIOS setting that is overvolting the GPU.
E. Insufficient system RAM.

Answer: A,B,C

Explanation:
Degraded thermal paste loses its ability to conduct heat effectively. A failing VRM can cause excessive power draw and heat generation. An incorrectly seated GPU can cause instability and poor power delivery, leading to overheating. Overvolting in BIOS will definitely cause overheating. While insufficient RAM can cause performance issues, it is less likely to lead to overheating.

Question: 2

You are monitoring a server with 8 GPUs used for deep learning training. You observe that one of the GPUs reports a significantly lower utilization rate compared to the others, even though the workload is designed to distribute evenly. 'nvidia-smi' reports a persistent "XID 13" error for that GPU. What is the most likely cause?

A. A driver bug causing incorrect workload distribution.
B. Insufficient system memory preventing data transfer to that GPU.
C. A hardware fault within the GPU, such as a memory error or core failure.
D. An incorrect CUDA version installed.
E. The GPU's compute mode is set to 'Exclusive Process'.

Answer: C

Explanation:
XID 13 errors in 'nvidia-smi' typically indicate a hardware fault within the GPU. Driver bugs or memory issues would likely cause different error codes or system instability across multiple GPUs. CUDA version mismatch might prevent the application from running altogether, but is less likely to lead to a specific XID error on a single GPU. Exclusive Process mode will lead to it being used by a different process but not necessarily cause that XID error.

Question: 3

You notice that one of the fans in your GPU server is running at a significantly higher RPM than the others, even under minimal load. ipmitool sensor' output shows a normal temperature for that GPU. What could be the potential causes?

A. The fan's PWM control signal is malfunctioning, causing it to run at full speed.
B. The fan bearing is wearing out, causing increased friction and requiring higher RPM to maintain airflow.
C. The fan is attempting to compensate for restricted airflow due to dust buildup.
D. The server's BMC (Baseboard Management Controller) has a faulty temperature sensor reading, causing it to overcompensate.
E. A network connectivity issue is causing higher CPU utilization, leading to increased system-wide heat.

Answer: A,B,C

Explanation:
A malfunctioning PWM control signal, worn fan bearings, or restricted airflow can all cause a fan to run at higher RPMs. While a faulty BMC sensor could be a cause, the question states that 'ipmitool sensor' shows a normal temperature. Network connectivity issues are less likely to cause an isolated fan to run high, if the GPU temperature is normal.

Question: 4

After upgrading the network card drivers on your A1 inference server, you experience intermittent network connectivity issues, including packet loss and high latency. You've verified that the physical connections are secure. Which of the following steps would be most effective in troubleshooting this issue?

A. Roll back the network card drivers to the previous version.
B. Check the system logs for error messages related to the network card or driver.
C. Run network diagnostic tools like 'ping', 'traceroute', and 'iperf3' to assess the network performance.
D. Reinstall the operating system.
E. Update the server's BIOS.

Answer: A,B,C

Explanation:
Rolling back drivers is a quick way to revert to a known working state. Checking system logs will provide valuable information about driver errors or network issues. Network diagnostic tools will quantify the network performance and help isolate the problem. Reinstalling the OS is drastic and should be a last resort. Updating the BIOS is unlikely to resolve driver-related network issues unless specifically recommended for the network card.

Question: 5

Your deep learning training job that utilizes NCCL (NVIDIA Collective Communications Library) for multi-GPU communication is failing with "NCCL internal error, unhandled system error" after a recent CUDA update. The error occurs during the 'all reduce' operation.
What is the most likely root cause and how would you address it?

A. Incompatible NCCL version with the new CUDA version. Update NCCL to a version compatible with the installed CUDA version.
B. Insufficient shared memory allocated to the CUDA context. Increase the shared memory limit using 'cudaDeviceSetLimit(cudaLimitSharedMemory, new_limity.
C. Firewall rules blocking inter-GPU communication. Configure the firewall to allow communication on the NCCL-defined ports (typically 8000-8010).
D. Faulty network cables used for inter-node communication (if the training job spans multiple servers). Replace the network cables with certified high-speed cables.
E. GPU Direct RDMA is not properly configured. Check 'dmesg' for errors and ensure RDMA is enabled.

Answer: A

Explanation:
NCCL relies on specific CUDA versions. An incompatibility after a CUDA update is the most probable cause. Insufficient shared memory is less likely to cause a system error within NCCL. Firewall rules usually manifest as connection refused errors. Faulty network cables affect inter-node communication, not intra-node. While RDMA issues can cause problems, they typically don't present as 'unhandled system error' immediately after a CUDA update, and are more likely if RDMA was working previously.

Question: 6

You are deploying a new A1 inference service using Triton Inference Server on a multi-GPU system. After deploying the models, you observe that only one GPU is being utilized, even though the models are configured to use multiple GPUs. What could be the possible causes for this?

A. The model configuration file does not specify the 'instance_group' parameter correctly to utilize multiple GPUs.
B. The Triton Inference Server is not configured to enable CUDA Multi-Process Service (MPS).
C. Insufficient CPU cores are available for the Triton Inference Server, limiting its ability to spawn multiple inference processes.
D. The models are not optimized for multi-GPU inference, resulting in a single GPU bottleneck.
E. The GPUs are not of the same type and Triton cannot properly schedule across them.

Answer: A,B

Explanation:
The 'instance_group' parameter in the model configuration dictates how Triton distributes the model across GPUs. Without proper configuration, it may default to a single GPIJ. CUDA MPS allows multiple

CUDA applications (in this case, Triton inference processes) to share a single GPU, improving utilization. Insufficient CPU cores or non-optimized models could limit performance, but wouldn't necessarily restrict usage to a single GPIJ. While dissimilar GPIJs can affect performance, Triton will attempt to schedule across them if configured correctly.

Question: 7

An A1 server exhibits frequent kernel panics under heavy GPU load. 'dmesg' reveals the following error: 'NVRM: Xid (PCl:0000:3B:00): 79, pid=..., name=..., GPU has fallen off the bus.' Which of the following is the least likely cause of this issue?

A. Insufficient power supply to the GPIJ, causing it to become unstable under load.
B. A loose or damaged PCle riser cable connecting the GPU to the motherboard.
C. A driver bug in the NVIDIA drivers, leading to GPU instability.
D. Overclocking the GPU beyond its stable limits.
E. A faulty CPU.

Answer: E

Explanation:
The error message GPU has fallen off the bus strongly suggests a hardware-related issue with the GPU's connection to the motherboard or its power supply. Insufficient power, a loose riser cable, driver bugs and overclocking can all lead to this. A faulty CPU, while capable of causing system instability, is less directly related to the GPIJ falling off the bus and therefore the least likely cause in this specific scenario.

Question: 8

You are using GPU Direct RDMA to enable fast data transfer between GPUs across multiple servers. You are experiencing performance degradation and suspect RDMA is not working correctly. How can you verify that GPU Direct RDMA is properly enabled and functioning?

A. Check the output of 'nvidia-smi topo -m' to ensure that the GPUs are connected via NVLink and have RDMA enabled.
B. Examine the 'cimesg' output for any errors related to RDMA or InfiniBand drivers.
C. Use the 'ibstat command to verify that the InfiniBand interfaces are active and connected.
D. Run a bandwidth benchmark using a tool like or to measure the RDMA throughput.
E. Ping the other servers to ensure network connectivity.

Answer: B,C,D

Explanation:
'dmesg' will show errors during RDMA driver initialization. Sibstat' confirms the InfiniBand interface status. Benchmarking with or validates the actual RDMA throughput. 'nvidia-smi topo -m' shows the topology but not necessarily active RDMA. Pinging only verifies basic network connectivity, not RDMA functionality.

Question: 9

You're debugging performance issues in a distributed training job. 'nvidia-smi' shows consistently high GPU utilization across all nodes, but the training speed isn't increasing linearly with the number of GPUs. Network bandwidth is sufficient. What is the most likely bottleneck?

A. Inefficient data loading and preprocessing pipeline, causing GPUs to wait for data.
B. NCCL is not configured optimally for the network topology, leading to high communication overhead.
C. The learning rate is not adjusted appropriately for the increased batch size across multiple GPUs.
D. The global batch size has exceeded the optimal point for the model, reducing per-sample accuracy and slowing convergence.
E. CUDA Graphs is not being utilized.

Answer: A,B,C,D

Explanation:
If GPUs are highly utilized but scaling is poor, the bottleneck is likely not GPU compute itself. Inefficient data pipelines mean GPUs spend time idle waiting for data. Suboptimal NCCL configurations result in communication overhead negating the benefit of more GPUs. Incorrect learning rate with larger batch size will impact covergence. Batch sizes can affect convergence and model effectiveness. While CUDA Graphs improves performance, the other answers are more pertinent to the question.

Question: 10

You have a server equipped with multiple NVIDIA GPUs connected via NVLink. You want to monitor the NVLink bandwidth utilization in real-time. Which tool or method is the most appropriate and accurate for this?

A. Using 'nvidia-smi' with the '—display=nvlink' option.
B. Parsing the output of *nvprof during a representative workload.
C. Utilizing DCGM (Data Center GPU Manager) with its NVLink monitoring capabilities.
D. Monitoring network interface traffic using 'iftop' or 'tcpdump' .
E. Using 'gpustat' .

Answer: C

Explanation:
DCGM is specifically designed for monitoring and managing GPUs in data centers, including detailed NVLink statistics in real time.
'nvidia-smi —display=nvlink' provides a snapshot, not real-time data. 'nvprof is a profiling tool and not ideal for continuous monitoring. 'iftop' and 'tcpdump' monitor network traffic, not NVLink. 'gpustat' does not offer the granular NVLink data of DCGM.

Thank You for Trying Our Product

For More Information – Visit link below:

https://www.examsboost.com/

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ **90 Days Free Updates**

- ✓ **Money Back Pass Guarantee**

- ✓ **Instant Download or Email Attachment**

- ✓ **24/7 Live Chat Support**

- ✓ **PDF file could be used at any Platform**

- ✓ **50,000 Happy Customer**