Guidelines for using AI Cluster (with Sample Script)

Welcome to the IIT Kharagpur AI Cluster! Please adhere to the guidelines below to ensure efficient system utilization.

Accessing the KGP AI Cluster: Use the following SSH command to connect:

```
ssh <username>@dgxmaster.iitkgp.ac.in
```

The AI cluster comprises five NVIDIA DGX H100 systems, each equipped with eight H100 Tensor Core GPUs and inbuilt high-speed NVMe storage, ensuring efficient AI training and inference. A dedicated master node, powered by an Intel Skylake Xeon CPU with 40 cores, manages the cluster and provides a centralized home directory accessible to all users. This setup is housed at the Paramshakti data centre and delivers a robust and scalable environment for advanced AI research and enterprise applications.

Job Submission Guidelines:

1. **Partition and Node mapping:**
   - The cluster consists of five partitions, each linked to a specific node. Use --partition=dgx1 for the dgx1 node, --partition=dgx2 for the dgx2 node, and similarly for the other nodes.
   - Each node is equipped with 8 GPUs. Users can utilize a maximum of 8 GPUs on their assigned node anytime.
   - Users have been assigned access to a dedicated partition. Jobs submitted by a user will only run on their assigned node.
   - ***Select the appropriate partition when submitting jobs based on your node assignment***.
2. **QoS Restrictions:**
   - Each partition supports four distinct QoS (Quality of Service) policies: gpu2, gpu4, gpu6, and gpu8.
   - Each QoS has defined constraints:

QoS Name	Max Jobs/User	Max CPUs	Min GPUs	Max GPUs	Max Wall Time
gpu2	2	56	1	2	72 hours (3 days)
gpu4	2	56	3	4	48 hours (2 days)
gpu6	1	112	5	6	24 hours (1 day)
gpu8	1	112	7	8	12 hours

- Maximum running jobs are two per user account.
- Jobs not meeting these requirements will remain pending with the reason displayed in the squeue command.

3. **Job Constraints:**
   - Jobs must be submitted to a **single node** (#SBATCH --nodes=1).
   - The number of tasks per node (--ntasks-per-node) and GPUs (--gres=gpu) must align with assigned limits.
   - Users must specify the correct QoS specifications and partition in their job script.

Data Management:

1. **Temporary Storage:**
   - Each user is allocated **50 GB** of storage in /home/<username>.
   - Job-related data should be stored in /raid/<username>, with a **1TB quota per user**.
   - In the job submission script, **dgxxxx** should be changed to **raid**, as jobs must be submitted from the home directory; direct submission from compute nodes (dgx00x) will not work.
   - Each node's RAID directory is mounted on the master node as **/dgx00x**, allowing users to access their RAID storage via `cd /dgx00x/<username>/` from the master node.
   - On the compute nodes (dgx00x), the same directory is accessible as **/raid/username**, ensuring seamless access across the cluster.
2. **Backup Policy:**
   - Transfer your data to your local system upon job completion to avoid auto-deletion.
   - User is responsible for archiving their data. The cluster administration is not liable for any data loss.
3. **Auto-Deletion Policy:**
   - Data in /raid/<username> is automatically deleted after **30 days**.
   - Regular backups are essential to prevent data loss.

SAMPLE BATCH SCRIPT

```
#!/bin/bash
#SBATCH --job-name=job_name                         # Job name
#SBATCH --output=/raid/<username>/output_%j.txt     # Standard output
#SBATCH --error=/raid/<username>/error_%j.txt       # Standard error
#SBATCH --partition=dgx2          # Partition name (modify as per node allocation)
#SBATCH --qos=gpu2                # QoS (must match the number of gpu)
#SBATCH --nodes=1                 # Use 1 node
#SBATCH --ntasks-per-node=2       # Number of tasks per node (max 8)
#SBATCH --gres=gpu:2              # Number of GPUs (max 8)
#SBATCH --time=24:00:00           # Adjust time based on GPU usage limits in the QOS

# Navigate to the working directory (raid is mounted as dgx00x in the master node)
cd /raid/<username>/

# Load necessary modules (optional)
module load cuda/12.4
module load python/3.10
# Activate your Python environment
source /home/<username>/miniconda3/bin/activate <env>

# Execute the training script
python train_model.py
```

Key Notes:

1. Replace <username> with your actual username.
2. Ensure all paths (/raid/<username> and /home/<username>) are correctly set.
3. Use /raid/<username> for intermediate job outputs and /home/<username> for error logs or configurations.
4. Select the correct partition and QoS based on **node allocation** and **GPU usage**
5. Users must adhere to QoS policies when submitting jobs to avoid scheduling delays or rejections.
6. Several packages are available as modules in the cluster, which can be accessed using the command ‘module avail’ and loaded using the command ‘*module load <package>*’. If additional packages are required, they can be installed in the user's home directory.

For further assistance or inquiries, please contact the system administrator through the Institute intercom: **84604** or email at shaktisupport@iitkgp.ac.in.

COMMON ERRORS IN THE SCRIPT

Example 1: Mismatch between `--gres=gpu` and `--qos` (Incorrect GPU Count)

```
#!/bin/bash
#SBATCH --job-name=job1
#SBATCH --output=/raid/<username>/output_%j.txt
#SBATCH --error=/raid/<username>/error_%j.txt
#SBATCH --partition=dgx2
#SBATCH --qos=gpu2                    # QoS allows max 2 GPUs
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=2
#SBATCH --gres=gpu:4                  # Requesting 4 GPUs, but gpu2 allows max 2
#SBATCH --time=24:00:00

cd /raid/<username>/
source /home/<username>/miniconda3/bin/activate <env>
python train_model.py
```

Error Explanation

- The gpu2 QoS only allows up to 2 GPUs, but the script requests 4 GPUs (`--gres=gpu:4`).
- Slurm will reject this job due to exceeding the allowed GPU count.
- Fix: Change `--gres=gpu:4` to `--gres=gpu:2`.

Example 2: Wall Time Exceeds QoS Limit

```
#!/bin/bash
#SBATCH --job-name=job2
#SBATCH --output=/raid/<username>/output_%j.txt
#SBATCH --error=/raid/<username>/error_%j.txt
#SBATCH --partition=dgx2
#SBATCH --qos=gpu4   # gpu4 has max wall time of 48 hours
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=2
#SBATCH --gres=gpu:3
#SBATCH --time=72:00:00  # Exceeding max wall time of 48 hours for gpu4

cd /raid/<username>/
source /home/<username>/miniconda3/bin/activate <env>
python train_model.py
```

Error Explanation

- The gpu4 QoS has a maximum wall time of 48 hours, but the script requests 72 hours (`--time=72:00:00`).
- The job will be rejected or truncated to the max allowed wall time.
- Fix: Reduce `--time` to 48:00:00 or select an appropriate QoS.

Example 3: Using the Wrong RAID Directory Path

```
#!/bin/bash
#SBATCH --job-name=job5
#SBATCH --output=/dgx002/<username>/output_%j.txt  # Incorrect directory
#SBATCH --error=/dgx002/<username>/error_%j.txt   # Incorrect directory
#SBATCH --partition=dgx2
#SBATCH --qos=gpu2
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=2
#SBATCH --gres=gpu:2
#SBATCH --time=24:00:00

cd /dgx002/<username>/  # Incorrect path
source /home/<username>/miniconda3/bin/activate <env>
python train_model.py
```

Error Explanation

- On the compute node (dgx00x), the RAID directory should be accessed as /raid/username/, not /dgx00x/username/.

Example 4: Insufficient GPUs for Selected QoS

```
#!/bin/bash
#SBATCH --job-name=job4
#SBATCH --output=/raid/<username>/output_%j.txt
#SBATCH --error=/raid/<username>/error_%j.txt
#SBATCH --partition=dgx2
#SBATCH --qos=gpu6   # gpu6 requires min 5 GPUs
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=2
#SBATCH --gres=gpu:4  # Only 4 GPUs requested, but gpu6 requires at least 5
#SBATCH --time=24:00:00

cd /raid/<username>/
source /home/<username>/miniconda3/bin/activate <env>
python train_model.py
```

Error Explanation

- The gpu6 QoS requires at least 5 GPUs, but the script requests only 4.
- The job will be rejected because it does not meet the minimum GPU requirement.
- Fix: Increase `--gres=gpu:4` to `--gres=gpu:5` or change QoS to gpu4.

Example 5: Submitting the Job from a Compute Node Instead of Home Directory

Incorrect job submission:

```
sbatch /raid/<username>/train_script.sh
```

<u>**Error Explanation**</u>

- Jobs must be submitted from the home directory, but here, the user submits from /raid/ on the compute node.
- Submission from dgx002 (or any dgx00x node) will fail because Slurm requires job submission from the master node.

Fix: Run the job submission from the home directory:

```
cd ~
sbatch train_script.sh
```

=============xxx================