(ray-conda-env) rino@rino-Z370-HD3:~/Desktop/Open_Cluster_AI_Station_beta/cluster_matrix$ python cluster_matrix_v1.py

🚀 INITIALIZING CLUSTER MATRIX DISTRIBUTION SYSTEM

📋 VALIDATING NODE CONFIGURATION...

✅**Node configuration validated:** 4 nodes configured

✅**Percentage distribution validated:** 1.000000

🌐 CONFIGURING NETWORK SETTINGS...

Head Node Ethernet IP: 192.168.2.100

Head Node WiFi IP: 192.168.50.113

Head Node Ports: PULL=7779, PUSH=7780

Worker Node Ports: PULL=5557, PUSH=5558

Cluster Barrier Port: 7790

📁 CONFIGURING STORAGE PATHS...

Local Paths:

- **RAM Results:** /dev/shm/matrix_results/
- **Disk Folder:** matrix_shards/
- **RAM Folder:** /dev/shm/matrix_shards/
- **Project Dir:** /home/rino/Desktop/Open_Cluster_AI_Station_beta/cluster_matrix/

Remote Paths:

- **Disk Folder:** matrix_shards/

- **RAM Folder:** /dev/shm/matrix_shards/
- **RAM Results:** /dev/shm/matrix_results/
- **Project Dir:** /home/rino/Desktop/Open_Cluster_AI_Station_beta/cluster_matrix/

📊 INITIALIZING INSTANCE VARIABLES...

Matrix Name: big_matrixA

Split Matrix: True

Dimension: 0

📂 CREATING LOCAL DIRECTORIES...

✅ All required directories already exist

🔌 SETTING UP ZEROMQ CONNECTIONS...

Connecting to 3 unique nodes...

✅ Connected to worker node 192.168.2.101:5557

✅ Connected to worker node 192.168.2.104:5557

✅ Connected to worker WiFi 192.168.3.13:5557

✅ Connected to worker WiFi 192.168.3.243:5557

✅ Connected to worker WiFi 192.168.3.165:5557

✅ Connected to head node (self) 192.168.2.100:7779

Total sockets in pool: 3

🔄 SETTING UP CLUSTER BARRIER/ACK RECEIVER...

✅ Python frontend ACK receiver bound to port 7790

📡 CREATING REMOTE DIRECTORIES ON WORKER NODES...

Sending command: mkdir -p /home/rino/Desktop/Open_Cluster_AI_Station_beta/cluster_matrix/matrix_shards/ /dev/shm/matrix_shards/ /dev/shm/matrix_results/

✅ Directory creation command sent to 192.168.2.101

✅ Directory creation command sent to 192.168.2.104

✅ Directory creation command sent to 192.168.2.100

✅ Created 4 shards according to node percentages

Node 0: shard shape torch.Size([4000, 20000])

Node 1: shard shape torch.Size([4000, 20000])

Node 2: shard shape torch.Size([1000, 20000])

Node 3: shard shape torch.Size([1000, 20000])

Starting distribution of 4 shards to 3 unique nodes

Processing shard 0 for node 192.168.2.100

Head node: Saving to DISK=matrix_shards/big_matrixA_shard_0.bin

Head node: Saving to RAM=/dev/shm/matrix_shards/big_matrixA_shard_0.bin

Saving matrix to binary file: matrix_shards/big_matrixA_shard_0.bin

Converting input to numpy array...

Input is PyTorch tensor: shape=torch.Size([4000, 20000]), dtype=torch.float32, device=cpu

Converted to CPU float32 numpy array

Final numpy array: shape=(4000, 20000), dtype=float32

Converting to 4D format...

2D (4000, 20000) -> 4D (1, 1, 4000, 20000)

Writing binary file...

Wrote ndim: 4

Dimensions: 1 × 1 × 4000 × 20000

Wrote 80,000,000 float32 elements

File saved successfully

File size: 320,000,020 bytes

Expected size: 320,000,020 bytes

✓ File size verification passed

Memory usage: 305.18 MB

Save completed: matrix_shards/big_matrixA_shard_0.bin

Saving matrix to binary file: /dev/shm/matrix_shards/big_matrixA_shard_0.bin

Converting input to numpy array...

Input is PyTorch tensor: shape=torch.Size([4000, 20000]), dtype=torch.float32, device=cpu

Converted to CPU float32 numpy array

Final numpy array: shape=(4000, 20000), dtype=float32

Converting to 4D format...

2D (4000, 20000) -> 4D (1, 1, 4000, 20000)

Writing binary file...

Wrote ndim: 4

Dimensions: 1 × 1 × 4000 × 20000

Wrote 80,000,000 float32 elements

File saved successfully

File size: 320,000,020 bytes

Expected size: 320,000,020 bytes

✓ File size verification passed

Memory usage: 305.18 MB

Save completed: /dev/shm/matrix_shards/big_matrixA_shard_0.bin

Added RAM path to file list

Processing shard 1 for node 192.168.2.100

Head node: Saving to DISK=matrix_shards/big_matrixA_shard_1.bin

Head node: Saving to RAM=/dev/shm/matrix_shards/big_matrixA_shard_1.bin

Saving matrix to binary file: matrix_shards/big_matrixA_shard_1.bin

Converting input to numpy array...

Input is PyTorch tensor: shape=torch.Size([4000, 20000]), dtype=torch.float32, device=cpu

Converted to CPU float32 numpy array

Final numpy array: shape=(4000, 20000), dtype=float32

Converting to 4D format...

2D (4000, 20000) -> 4D (1, 1, 4000, 20000)

Writing binary file...

Wrote ndim: 4

Dimensions: 1 × 1 × 4000 × 20000

Wrote 80,000,000 float32 elements

File saved successfully

File size: 320,000,020 bytes

Expected size: 320,000,020 bytes

✓ File size verification passed

Memory usage: 305.18 MB

Save completed: matrix_shards/big_matrixA_shard_1.bin

Saving matrix to binary file: /dev/shm/matrix_shards/big_matrixA_shard_1.bin

Converting input to numpy array...

Input is PyTorch tensor: shape=torch.Size([4000, 20000]), dtype=torch.float32, device=cpu

Converted to CPU float32 numpy array

Final numpy array: shape=(4000, 20000), dtype=float32

Converting to 4D format...

2D (4000, 20000) -> 4D (1, 1, 4000, 20000)

Writing binary file...

Wrote ndim: 4

Dimensions: 1 × 1 × 4000 × 20000

Wrote 80,000,000 float32 elements

File saved successfully

File size: 320,000,020 bytes

Expected size: 320,000,020 bytes

✓ File size verification passed

Memory usage: 305.18 MB

Save completed: /dev/shm/matrix_shards/big_matrixA_shard_1.bin

Added RAM path to file list

Processing shard 2 for node 192.168.2.101

Remote node 192.168.2.101: Beginning distribution

Step 1: Saving locally to matrix_shards/big_matrixA_shard_2.bin

Saving matrix to binary file: matrix_shards/big_matrixA_shard_2.bin

Converting input to numpy array...

Input is PyTorch tensor: shape=torch.Size([1000, 20000]), dtype=torch.float32, device=cpu

Converted to CPU float32 numpy array

Final numpy array: shape=(1000, 20000), dtype=float32

Converting to 4D format...

2D (1000, 20000) -> 4D (1, 1, 1000, 20000)

Writing binary file...

Wrote ndim: 4

Dimensions: 1 × 1 × 1000 × 20000

Wrote 20,000,000 float32 elements

File saved successfully

File size: 80,000,020 bytes

Expected size: 80,000,020 bytes

✓ File size verification passed

Memory usage: 76.29 MB

Save completed: matrix_shards/big_matrixA_shard_2.bin

Step 2: Sending file to remote node 192.168.2.101

📤 Sent file big_matrixA_shard_2.bin to 192.168.2.101

✅ Received ACK 1/1

✅ All ACKs received!

Step 3: Sending copy command to remote

Added remote RAM path to file list: /dev/shm/matrix_shards/big_matrixA_shard_2.bin

Processing shard 3 for node 192.168.2.104

Remote node 192.168.2.104: Beginning distribution

Step 1: Saving locally to matrix_shards/big_matrixA_shard_3.bin

Saving matrix to binary file: matrix_shards/big_matrixA_shard_3.bin

Converting input to numpy array...

Input is PyTorch tensor: shape=torch.Size([1000, 20000]), dtype=torch.float32, device=cpu

Converted to CPU float32 numpy array

Final numpy array: shape=(1000, 20000), dtype=float32

Converting to 4D format...

2D (1000, 20000) -> 4D (1, 1, 1000, 20000)

Writing binary file...

Wrote ndim: 4

Dimensions: 1 × 1 × 1000 × 20000

Wrote 20,000,000 float32 elements

File saved successfully

File size: 80,000,020 bytes

Expected size: 80,000,020 bytes

✓ File size verification passed

Memory usage: 76.29 MB

Save completed: matrix_shards/big_matrixA_shard_3.bin

Step 2: Sending file to remote node 192.168.2.104

📤 Sent file big_matrixA_shard_3.bin to 192.168.2.104

✅ Received ACK 1/1

✅ All ACKs received!

Step 3: Sending copy command to remote

Added remote RAM path to file list: /dev/shm/matrix_shards/big_matrixA_shard_3.bin

Distribution complete: 4 shards saved and distributed

🚀 INITIALIZING CLUSTER MATRIX DISTRIBUTION SYSTEM

📋 VALIDATING NODE CONFIGURATION...

✅ **Node configuration validated:** 4 nodes configured

✅ **Percentage distribution validated:** 1.000000

🌐 CONFIGURING NETWORK SETTINGS...

Head Node Ethernet IP: 192.168.2.100

Head Node WiFi IP: 192.168.50.113

Head Node Ports: PULL=7779, PUSH=7780

Worker Node Ports: PULL=5557, PUSH=5558

Cluster Barrier Port: 7790

📁 CONFIGURING STORAGE PATHS...

Local Paths:

- **RAM Results:** /dev/shm/matrix_results/
- **Disk Folder:** matrix_shards/
- **RAM Folder:** /dev/shm/matrix_shards/
- **Project Dir:** /home/rino/Desktop/Open_Cluster_AI_Station_beta/cluster_matrix/

Remote Paths:

- **Disk Folder:** matrix_shards/
- **RAM Folder:** /dev/shm/matrix_shards/
- **RAM Results:** /dev/shm/matrix_results/
- **Project Dir:** /home/rino/Desktop/Open_Cluster_AI_Station_beta/cluster_matrix/

📊 INITIALIZING INSTANCE VARIABLES...

Matrix Name: big_matrixB

Split Matrix: False

Dimension: 0

📂 CREATING LOCAL DIRECTORIES...

✅All required directories already exist

🔌 SETTING UP ZEROMQ CONNECTIONS...

Connecting to 3 unique nodes...

✅Connected to worker node 192.168.2.101:5557

✅Connected to worker node 192.168.2.104:5557

✅Connected to worker WiFi 192.168.3.13:5557

✅Connected to worker WiFi 192.168.3.243:5557

✅Connected to worker WiFi 192.168.3.165:5557

✅Connected to head node (self) 192.168.2.100:7779

Total sockets in pool: 3

🔄 SETTING UP CLUSTER BARRIER/ACK RECEIVER...

✅ACK receiver already exists on port 7790

📡 CREATING REMOTE DIRECTORIES ON WORKER NODES...

Sending command: mkdir -p /home/rino/Desktop/Open_Cluster_AI_Station_beta/cluster_matrix/matrix_shards/ /dev/shm/matrix_shards/ /dev/shm/matrix_results/

✅Directory creation command sent to 1