Of course! Based on the files you have in your directory, here is a step-by-step guide to compile the code, run the benchmarks, generate the graphs, and package the final results.

Follow these commands in your terminal from the directory shown in your screenshot.

Prerequisites

Before you begin, make sure you have:

1. An MPI implementation installed (like Open MPI or MPICH).
2. Python 3 installed.
3. The necessary Python libraries: pandas, matplotlib, and seaborn. If you don't have them, you can install them with pip:

```
pip install pandas matplotlib seaborn
```

Step 1: Compile the C Source Code

You have a `Makefile` which automates the compilation process. Simply run the `make` command. This will use `mpicc` to compile all your `.c` files and create the corresponding executables: `linear_exchange`, `ring`, `rabenseifner`, and `non-pipeline-allreduce`.

```
make
```

After this step, you will see four new executable files in your directory.

Step 2: Run the Benchmark Script

The `run_benchmarks.sh` script is designed to run all the compiled programs with the various process counts and buffer sizes.

First, make the script executable:

```
chmod +x run_benchmarks.sh
```

Now, run the script. This will execute all the benchmark tests and save the timing results into a file named `latency_results.csv`.

```
./run_benchmarks.sh
```

Note: This step will take some time to complete as it runs many simulations. You will see output in your terminal for each test being performed.

Step 3: Generate the Latency Graphs

Once the benchmark script is finished and you have the `latency_results.csv` file, you can generate the graphs using the Python script.

Run the `generate_latency_graphs.py` script:

```
python3 generate_latency_graphs.py
```

This will read the `latency_results.csv` file and create a new folder named `latency_graphs`. Inside this folder, you will find all the `.png` image files for your plots.

Step 4: Package the Graphs into a Zip File

The final step is to create a zip archive containing all the generated graphs.

Run the following command to create `latency_comparison_graphs.zip`:

```
zip -r latency_comparison_graphs.zip latency_graphs/
```

Summary of Commands

Here are all the commands in order:

1. `make`
2. `chmod +x run_benchmarks.sh`
3. `./run_benchmarks.sh`
4. `python3 generate_latency_graphs.py`
5. `zip -r latency_comparison_graphs.zip latency_graphs/`

After completing these steps, you will have successfully compiled the code, run the experiments, and generated all the required deliverables.