Ranking

The rai *ranking* command is now active. The *ranking* command calculates the fastest sum of both Layer Runtimes for all valid runs for each student and then ranks each student from fastest to slowest. *Rankings* are updated immediately with each valid run. A valid run requires no profiling and a batch size of 10,000.

Below is an example of *ranking* results. Notice that your netid and "-->" indicate your rank and time.

Ranking:

```
+-------------+------+--------------+
|     YOU     | RANK | FASTEST (MS) |
+-------------+------+--------------+
|             |    0 | 1.040769s    |
|             |    1 | 1.051001s    |
|             |    2 | 1.06379s     |
|             |    3 | 1.167731s    |
|             |    4 | 1.188924s    |
| aschuh3 --> |    5 | 1.204951s    |
+-------------+------+--------------+
```

Note: We began collecting *ranking* statistics on 4/15/2021 at 11:38 AM. Prior runs will not be included in *ranking*.

Exclusive Queue

We currently have 4 servers, each with 2 GPUs, which each run up to 6 simultaneous jobs. This allows up to a total of 24 simultaneous jobs which is perfect for handling a large number of jobs efficiently. However, profiling results are impacted significantly when other programs compete for resources.

To address this issue, the exclusive queue is now active and should be used for profiling only. Three of the servers will handle the default rai_amd64_ece408 queue, allowing 18 simultaneous runs. The fourth server will now handle the rai_amd64_ece408_exclusive queue which will allow 2 simultaneous runs (1 on each GPU). The rai api allows you to specify the queue with -q <<queue name>>. Note, the default queue is rai_amd64_ece408 so the -q option is only required when using the rai_amd64_ece408_exclusive queue.

Example:

rai -p <<address of your project code>> -q rai_amd64_ece408_exclusive

There will be times when submitted jobs may not run immediately because of a large queue. This is particularly true when using the rai_amd64_ece408_exclusive queue. Below are a few tips when working with a large queue backlog.

- You can check the current queue size by adding "queued" to the end of your rai command. The rai command must still include -p <<address of your project code>> and will default to returning the rai_amd64_ece408 queue size. Include -q rai_amd64_ece408_exclusive to see the current rai_amd64_ece408_exclusive queue size.
- After your job is submitted (waiting for server), you can kill the rai command and even turn off your computer. A server will handle your queued job in its time and post the results to the database. Later you can run the rai history command. If the job has run it will be included in your history request and the results can downloaded through the associated URL.

Note: As needed we may dynamically move other machines to the rai_amd64_ece408_exclusive queue.