HW1
Jingli Kong

Task 3 writeup:

The customized partition I use is the default hash function which hashes all the (business_id, num_reviews) tuples into an integer to be mapped to the number of partitions. The execution time is roughly the same (nearly exactly) if I call my custom partition with the same number of partitions as the default partitioner decides to use (27 in my case).

The main differences occur when I call the task3_customized.py with extremely high (100+) partitions or extremely low partitions (<27).
If we don't create enough partitions to the number of cores on our machine we cannot parallelize the task as efficiently because we are not utilizing all our cores, therefore the tasks run longer. If we create too many partitions especially when we exceed the number of cores we have on our computer the communication between all these partitions increases the time it takes to run the program/query. But this only really occurs when we create extremely high number of partitions

Also, it's important to note that when we chose to partition data we have to make sure that this partition can fit on the local memory of the node if it cannot then the disk io operations would be extremely expensive. So we need to have all the data we have partitioned among the nodes, but each data partition needs to fit on each node's memory.