PROJECT-2 REPORT

In this project, we were expected to develop a program using MPI framework to calculate conditional probabilities of bigrams. The project consists of two parts which differ in merging method.

First, we read the whole data and bigram test word files from “input_file” and “test_file” given as command line arguments. Then we called the necessary function either “master” or “worker” according to the merging method provided by the argument. With given size P, we assigned the number of processes to variable “n_rank” and also declared another variable “rank” to keep track of the rank of current process. If rank is 0, current process is master process, otherwise it is worker process.

Our program consists of two main functions which are “master” and “worker”. “master” function does the required operations for the first part which the merge operation is done by the master process. It takes file_line and test_line as parameters. Master process reads input_line which contains the data given in input_file and distributes lines evenly to the worker processes using “send” function. Then, every worker use “recv” function to receive data from master and prints its rank and number of lines that it will handle. Every worker has its own uni_count and bi_count dictionaries which has unigrams and bigrams as keys and their number of occurrences as their values. To do this job, we wrote two functions” unigram_word_counter” and “bigram_word_counter” that take a line as parameter and count the number of every unigram and bigram words in that line, then update the uni_count and bi_count dictionaries of the current worker. Then using “send” and “recv” function, we took the dicts from each worker process and send them to the master process and assigned it to the “gathered_data”. After that, master process does the merging operation using “merger” function that takes many unigram and bigram dictionaries and returns a list named “last_arr”, consists of one final unigram dict and one final bigram dict. Finally, master process does the probability calculations. It iterates over lines of test_file and reads bigrams. Then using last_arr, calculates each conditional probability and prints it. 0th index of last_arr is a dict contains all unigram words with their number of occurrences and first index is a dict contains all bigrams. Probability calculations are done by dividing the value of current bigram by the value of unigram obtained from these dictionaries.

In our second fundamental function “workers” which fulfills the requirement 3, master process does the distributing of data to the workers evenly in the same way as the “master” function. Then, every worker process prints its rank and number of lines it will handle. Each worker counts the numbers of unigrams and bigrams using unigram_word_counter and bigram_word_counter functions, then holds the numbers within the uni_count and bi_count dicts. Differently from the previous method, each worker takes the data from the previous worker using “recv” function and after the counting operations, merges its data with the received one using the “merger_two” function. Then sends the merged data named as “new_data” to the next worker with “send” function. If current worker is the last worker before the master process, it sends the merged data to the master process and if current worker is the first worker, it only sends, doesn’t receive data since there is no worker before it. After all workers finish their job, master receives merged data named as “last_arr” from the last worker with recv function. It reads input_file and test_file, then using last_arr, calculates conditional probability for each bigram in the test_file and prints it. We calculate the probabilities with dividing the value of current bigram by the value of unigram using unigram and bigram word dictionaries that are kept in last_arr list.

To run the program, one must run such as these codes in the terminal, arguments can be changed:

```
mpiexec -n 5 python BeyzanurBektan.py --input_file data/sample_text.txt --merge_method MASTER --test_file data/test.txt
```

```
mpiexec -n 5 python BeyzanurBektan.py --input_file data/sample_text.txt --merge_method WORKERS --test_file data/test.txt
```

Our result is firstly ranks and number of lines in them. For example, if there are 1 master 3 worker process, result can be looked like that:

```
rank: 1 number of sentences: 78812
rank: 2 number of sentences: 78811
rank: 3 number of sentences: 78811
```

Secondly, program prints bigrams conditional probabilities in the test file like that:

```
The probability of pazar günü --> 0.4462962962962963
The probability of pazartesi günü --> 0.5966101694915255
The probability of karar verecek --> 0.010940919037199124
The probability of karar verdi --> 0.13216630196936544
The probability of boğaziçi üniversitesi --> 0.37272727272727274
The probability of bilkent üniversitesi --> 0.2222222222222222
```