Concurrency and Parallelism. Block II Parallelism
Assignment 3: Domain decomposition: similarity vector for DNA datasets

Spring 2023

Similarity vector for DNA datasets

Datasets of DNA sequences

- M text sequences
- Each one with N bases (characters A,C,G,T,N)

Similarity vector

- One vector for two datasets of DNA sequences
- Metric that indicates how similar two sequences are, namely the $i - th$ sequence of each dataset
    - M results: both datasets must have the same number of sequences (M)
    - SIMPLIFICATION: all sequences have the same number of bases (N)

Similarity vector for DNA datasets

Sequential code

```
int main(int argc, char *argv[]){
  int *data1, *data2, *result;
  data1 = (int *) malloc(M*N*sizeof(int));
  data2 = (int *) malloc(M*N*sizeof(int));
  result = (int *) malloc(M*sizeof(int));

  /* Initialize Matrices */
  ...

  for(i=0;i<M;i++) {
    result[i]=0;
    for(j=0;j<N;j++) {
      result[i] += base_distance(data1[i*N+j], data2[i*N+j]);
    }
  }

  free(data1); free(data2); free(result);
```

Similarity vector for DNA datasets

Sequential code: `similarity.c`

Domain decomposition

- Divide the matrices among p processes, each one with $rows = M/p$ rows (for simplicity, start with the scenario where the number of processes is multiple of M: $M \mod p = 0$).
- Each task will be in charge of calculating $\frac{M}{p}$ rows of the result vector.
- Later, modify the code to consider the general case, not only multiples of M.

Similarity vector for DNA datasets

Parallel approach

- SPMD implementation.
- Process 0 performs the initialization of the matrices.
- Data is distributed to all the processes using collective operations.
- Gather the result using collective operations.
- I/O (printf) is performed by process 0.
- Print separately the communication time and the computation time of each process.

Domain decomposition: matrix-vector product

- Assigned points: 0.75
- Deadline: May 3rd
- It must be done by the same couples as P1 and P2
- Defense in the practical lecture