report

2022_28379 협동과정 인공지능 전공 임성준

▼ mpi 1번

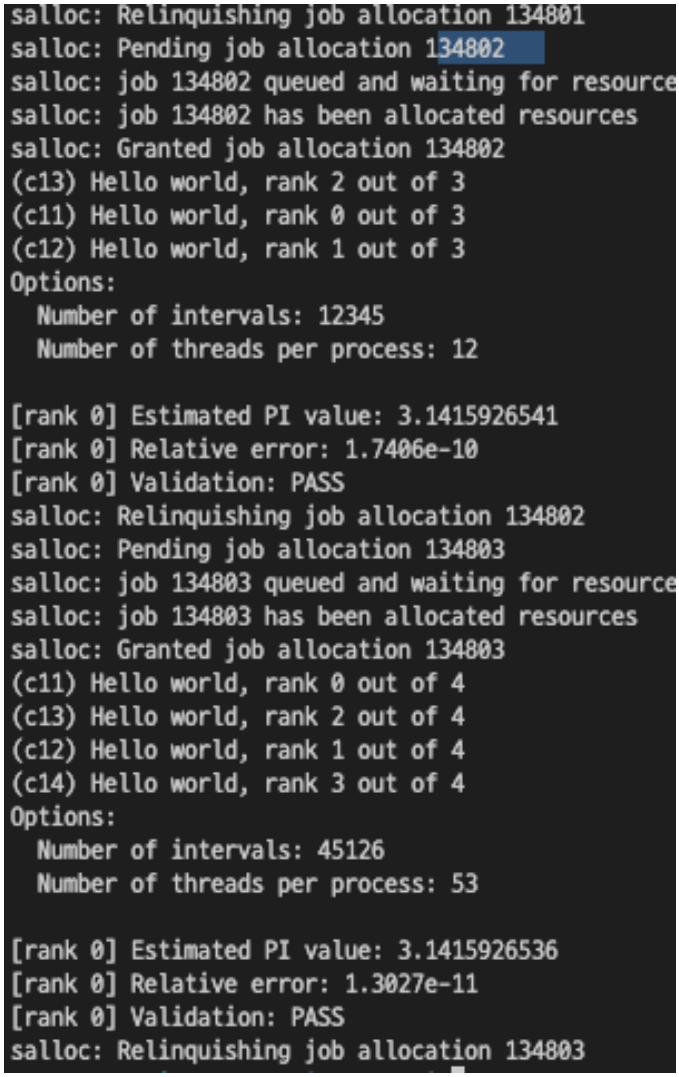
- 자신의 병렬화 방식에 대한 설명. MPI를 이용해 여러 노드(프로세스)에 작업을 어떻게 배분하는지 위주로 설명.
    - rank를 기준으로 interval을 프로세스 별로 나누고 각 프로세스에서는 openMP를 사용해서 스레드로 또 나누어 처리했습니다. 마지막으로 각 프로세스의 결과를 reduce 함수를 통해 sum해서 최종적인 값을 구했습니다.
    - (아래는 코드입니다.)

```c
double riemannsum(int num_intervals, int mpi_rank, int mpi_world_size,
                  int threads_per_process) {
  double pi = 0;
  double h = 1.0 / (double)num_intervals;

  // TODO: Parallelize the code using mpi_world_size processes (1 process per
  // node.
  // In total, (mpi_world_size * threads_per_process) threads will collaborate
  // to compute the Riemann sum.
  double sum = 0;
  int part = num_intervals / mpi_world_size;
  int remainder = num_intervals % mpi_world_size;

  if (mpi_rank == 0) {
    for (int i = 1; i < part; i++) {
      double x = h * ((double)i - 0.5);
      sum += h * f(x);
    }
    if(1){
      int start_index;
      if(mpi_world_size > num_intervals){
        start_index = mpi_world_size * part + 1;
      }else{
        start_index = mpi_world_size * part;
      }
      for (int i = start_index; i <= mpi_world_size * part + remainder; i++) {
        double x = h * ((double)i - 0.5);
        sum += h * f(x);
      }
    }
  }
  else{
    #pragma omp parallel for reduction(+:sum)
    for(int i = mpi_rank * part; i < mpi_rank * part + part; i++){
      double x = h * ((double)i - 0.5);
      sum += h * f(x);
    }
  }

  MPI_Barrier(MPI_COMM_WORLD);
  MPI_Reduce(&sum, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);
  // Rank 0 should return the estimated PI value
  // Other processes can return any value (don't care)
  return pi;
}
```

- 병렬화를 해보면 통신 시간 등 때문에 프로그램 실행시간이 오히려 늘어나는 것을 경험할 것이다. MPI를 이용한 병렬화를 하기에 적합한 f는 어떤 특징을 가져야 하겠는가? (Hint: 뼈대 코드에 주어진 f 는 어떤 것인가?)

    - f를 통해 한번에 결과를 구할 수 있고 구한 결과를 더할 수 있어야 한다.
- validation 모두 valid합니다.

```
shpc123@login0:~/hw4/riemannsum$ ./run_validation.sh
salloc: Granted job allocation 134794
(c11) Hello world, rank 0 out of 1
Options:
  Number of intervals: 120
  Number of threads per process: 16

[rank 0] Estimated PI value: 3.1415984406
[rank 0] Relative error: 1.8421e-06
[rank 0] Validation: PASS
salloc: Relinquishing job allocation 134794
salloc: Granted job allocation 134795
(c12) Hello world, rank 0 out of 2
(c13) Hello world, rank 1 out of 2
Options:
  Number of intervals: 50
  Number of threads per process: 32

[rank 0] Estimated PI value: 3.1416259869
[rank 0] Relative error: 1.0610e-05
[rank 0] Validation: PASS
salloc: Relinquishing job allocation 134795
salloc: Pending job allocation 134796
salloc: job 134796 queued and waiting for resources
salloc: job 134796 has been allocated resources
salloc: Granted job allocation 134796
(c11) Hello world, rank 0 out of 3
(c13) Hello world, rank 2 out of 3
(c12) Hello world, rank 1 out of 3
Options:
  Number of intervals: 1
  Number of threads per process: 5

[rank 0] Estimated PI value: 3.2000000000
[rank 0] Relative error: 1.8592e-02
[rank 0] Validation: PASS
salloc: Relinquishing job allocation 134796
salloc: Pending job allocation 134797
salloc: job 134797 queued and waiting for resources
salloc: job 134797 has been allocated resources
salloc: Granted job allocation 134797
(c11) Hello world, rank 0 out of 4
(c13) Hello world, rank 2 out of 4
(c12) Hello world, rank 1 out of 4
(c14) Hello world, rank 3 out of 4
Options:
  Number of intervals: 5040
  Number of threads per process: 9

[rank 0] Estimated PI value: 3.1415926569
[rank 0] Relative error: 1.0443e-09
[rank 0] Validation: PASS
salloc: Relinquishing job allocation 134797
salloc: Pending job allocation 134798
salloc: job 134798 queued and waiting for resources
salloc: job 134798 has been allocated resources
salloc: Granted job allocation 134798
(c11) Hello world, rank 0 out of 1
Options:
  Number of intervals: 2202
  Number of threads per process: 3

[rank 0] Estimated PI value: 3.1415926708
[rank 0] Relative error: 5.4706e-09
[rank 0] Validation: PASS
salloc: Relinquishing job allocation 134798
salloc: Granted job allocation 134799
(c12) Hello world, rank 0 out of 2
(c13) Hello world, rank 1 out of 2
Options:
  Number of intervals: 1367
  Number of threads per process: 15

[rank 0] Estimated PI value: 3.1415926982
[rank 0] Relative error: 1.4195e-08
[rank 0] Validation: PASS
salloc: Relinquishing job allocation 134799
salloc: Pending job allocation 134800
salloc: job 134800 queued and waiting for resources
salloc: job 134800 has been allocated resources
salloc: Granted job allocation 134800
(c11) Hello world, rank 0 out of 3
(c13) Hello world, rank 2 out of 3
(c12) Hello world, rank 1 out of 3
Options:
  Number of intervals: 325
  Number of threads per process: 26

[rank 0] Estimated PI value: 3.1415934425
[rank 0] Relative error: 2.5113e-07
[rank 0] Validation: PASS
salloc: Relinquishing job allocation 134800
salloc: Pending job allocation 134801
salloc: job 134801 queued and waiting for resources
salloc: job 134801 has been allocated resources
salloc: Granted job allocation 134801
(c12) Hello world, rank 1 out of 4
(c14) Hello world, rank 3 out of 4
(c11) Hello world, rank 0 out of 4
(c13) Hello world, rank 2 out of 4
Options:
  Number of intervals: 64
  Number of threads per process: 31

[rank 0] Estimated PI value: 3.1416129986
[rank 0] Relative error: 6.4760e-06
[rank 0] Validation: PASS
```



▼ mpi 2번

- 자신의 병렬화 방식에 대한 설명.
  - 이전과 같은 방법으로 row를 프로세스 별로 나눠주고 각 프로세스에서는 openMP를 통해 스레드 별로 또 처리하였습니다. 작업량이 process size로 나누어 떨어지지 않는 경우에는 rank 0에 나머지 작업을 추가로 처리하도록 구현했습니다.
  - 아래는 제 코드입니다.

```
void matmul(const float *A, const float *B, float *C, int M, int N, int K,
            int threads_per_process, int mpi_rank, int mpi_world_size) {

  // // TODO: FILL_IN_HERE
  int interval = M / mpi_world_size;
  int remainder = M % mpi_world_size;
  float r;

  if (mpi_rank == 0){
    MPI_Bcast(A,M*K,MPI_FLOAT,0,MPI_COMM_WORLD);
    MPI_Bcast(B,K*N,MPI_FLOAT,0,MPI_COMM_WORLD);

    // interval 처리
    for(int i = 1; i < ld_size; i++){
      MPI_Recv(C + i * erval, interval * N, MPI_FLOAT, i, i, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    }
    for (int k=0; k<K; k++) {
      #pragma omp parallel for
      for (int i=0; i<interval; i++) {
          r = *(A + i*K + k);
          for (int j=0; j<N; j++){
            *(C + i*N + j) += r * *(B + j + k*N);
          }
      }
    }

    // residual 처리
    if(remainder != 0){
      for (int k=0; k<K; k++) {
        #pragma omp parallel for
        for (int i=interval * mpi_world_size; i<interval * mpi_world_size + remainder; i++) {
            r = *(A + i*K + k);
            for (int j=0; j<N; j++){
              *(C + i*N + j) += r * *(B + j + k*N);
            }
        }
      }

    }
  }
  else if(mpi_rank != 0)// interval * (N + 1)
  {
    MPI_Bcast(A,M*K,MPI_FLOAT,0,MPI_COMM_WORLD);
    MPI_Bcast(B,K*N,MPI_FLOAT,0,MPI_COMM_WORLD);

    for (int k=0; k<K; k++){
      #pragma omp parallel for
      for (int i=mpi_rank * interval; i< mpi_rank * interval + interval; i++) {
          r = *(A + i*K + k);
          for (int j=0; j<N; j++){
            *(C + i*N + j) += r * *(B + j + k*N);
          }
      }
    }
    MPI_Send(C + mpi_rank * N * interval, interval * N, MPI_FLOAT, 0, mpi_rank, MPI_COMM_WORLD);
  }
}
```

- 자신이 사용한 통신 패턴에 대한 설명. (e.g., Scatter, Gather, Broadcast 등)
  - 하나 하나 옮길 수도 있지만 성능 기준을 넘어서 그냥 한번에 옮겨주는 MPI_Bcast방법을 통해서 A,B행렬을 master process를 제외한 프로세스에 옮겼

고 각 프로세스에서는 작업을 마치고 MPI_Send를 통해서 다시 master process에 결과를 보냈습니다. 이를 MPI_Recv를 반복문을 돌리면서 master process에서 받았습니다.

1) valid 모두 valid합니다.


shpc123@login0:~/hw4/matmul$ ./run_validation.sh
salloc: Pending job allocation 134590
salloc: job 134590 queued and waiting for resources
salloc: job 134590 has been allocated resources
salloc: Granted job allocation 134590
(c11) Hello world, rank 0 out of 4
(c13) Hello world, rank 2 out of 4
(c12) Hello world, rank 1 out of 4
(c14) Hello world, rank 3 out of 4
Options:
Problem size: M = 831, N = 538, K = 2304
Number of threads per process: 26
Number of iterations: 1
Print matrix: off
Validation: on
[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.056225 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.056225 sec
[rank 0] Avg. throughput: 36.640773 GFLOPS
salloc: Relinquishing job allocation 134590
salloc: Pending job allocation 134593
salloc: job 134593 queued and waiting for resources
salloc: job 134593 has been allocated resources
salloc: Granted job allocation 134593
(c12) Hello world, rank 0 out of 2
(c13) Hello world, rank 1 out of 2
Options:
Problem size: M = 3305, N = 1864, K = 3494
Number of threads per process: 9
Number of iterations: 1
Print matrix: off
Validation: on
[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.155410 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.155410 sec
[rank 0] Avg. throughput: 277.007374 GFLOPS
salloc: Relinquishing job allocation 134593
salloc: Pending job allocation 134596
salloc: job 134596 queued and waiting for resources
salloc: job 134596 has been allocated resources
salloc: Granted job allocation 134596
(c12) Hello world, rank 0 out of 1
Options:
Problem size: M = 618, N = 3102, K = 1695
Number of threads per process: 16
Number of iterations: 1
Print matrix: off
Validation: on
[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.026150 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.026150 sec
[rank 0] Avg. throughput: 248.518765 GFLOPS
salloc: Relinquishing job allocation 134596
salloc: Pending job allocation 134597
salloc: job 134597 queued and waiting for resources
salloc: job 134597 has been allocated resources
salloc: Granted job allocation 134597
(c11) Hello world, rank 0 out of 3
(c13) Hello world, rank 2 out of 3
(c12) Hello world, rank 1 out of 3
Options:
Problem size: M = 1876, N = 3453, K = 3590
Number of threads per process: 30
Number of iterations: 1
Print matrix: off
Validation: on
[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.170357 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.170357 sec
[rank 0] Avg. throughput: 273.020269 GFLOPS
salloc: Relinquishing job allocation 134597
salloc: Pending job allocation 134599
salloc: job 134599 queued and waiting for resources
salloc: job 134599 has been allocated resources
salloc: Granted job allocation 134599
(c11) Hello world, rank 0 out of 3
(c12) Hello world, rank 1 out of 3
(c13) Hello world, rank 2 out of 3
Options:
Problem size: M = 1228, N = 2266, K = 1552
Number of threads per process: 16
Number of iterations: 1
Print matrix: off
Validation: on
[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.049887 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.049887 sec
[rank 0] Avg. throughput: 173.136622 GFLOPS

```
salloc: Relinquishing job allocation 134599
salloc: Pending job allocation 134602
salloc: job 134602 queued and waiting for resou
salloc: job 134602 has been allocated resources
salloc: Granted job allocation 134602
(c13) Hello world, rank 0 out of 2
(c14) Hello world, rank 1 out of 2
Options:
  Problem size: M = 3347, N = 171, K = 688
  Number of threads per process: 2
  Number of iterations: 1
  Print matrix: off
  Validation: on

[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.018538 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.018538 sec
[rank 0] Avg. throughput: 42.481490 GFLOPS
salloc: Relinquishing job allocation 134602
salloc: Pending job allocation 134604
salloc: job 134604 queued and waiting for resou
salloc: job 134604 has been allocated resources
salloc: Granted job allocation 134604
(c11) Hello world, rank 0 out of 3
(c12) Hello world, rank 1 out of 3
(c13) Hello world, rank 2 out of 3
Options:
  Problem size: M = 3583, N = 962, K = 765
  Number of threads per process: 8
  Number of iterations: 1
  Print matrix: off
  Validation: on

[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.030943 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.030943 sec
[rank 0] Avg. throughput: 170.430862 GFLOPS
salloc: Relinquishing job allocation 134604
salloc: Pending job allocation 134606
salloc: job 134606 queued and waiting for resou
salloc: job 134606 has been allocated resources
salloc: Granted job allocation 134606
(c11) Hello world, rank 0 out of 1
Options:
  Problem size: M = 2962, N = 373, K = 1957
  Number of threads per process: 30
  Number of iterations: 1
  Print matrix: off
  Validation: on

[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.021604 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.021604 sec
[rank 0] Avg. throughput: 200.163673 GFLOPS
salloc: Relinquishing job allocation 134606
salloc: Pending job allocation 134608
salloc: job 134608 queued and waiting for resou
salloc: job 134608 has been allocated resources
salloc: Granted job allocation 134608
(c11) Hello world, rank 0 out of 4
(c12) Hello world, rank 1 out of 4
(c13) Hello world, rank 2 out of 4
(c14) Hello world, rank 3 out of 4
Options:
  Problem size: M = 3646, N = 2740, K = 3053
  Number of threads per process: 9
  Number of iterations: 1
  Print matrix: off
  Validation: on

[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.154446 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.154446 sec
[rank 0] Avg. throughput: 394.955861 GFLOPS
salloc: Relinquishing job allocation 134608
salloc: Pending job allocation 134609
salloc: job 134609 queued and waiting for resou
salloc: job 134609 has been allocated resources
salloc: Granted job allocation 134609
(c11) Hello world, rank 0 out of 3
(c12) Hello world, rank 1 out of 3
(c13) Hello world, rank 2 out of 3
Options:
  Problem size: M = 1949, N = 3317, K = 3868
  Number of threads per process: 26
  Number of iterations: 1
  Print matrix: off
  Validation: on

[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.179927 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.179927 sec
[rank 0] Avg. throughput: 277.956790 GFLOPS
```

2) performance: 715GFLOPS나왔습니다.

```
shpc123@login0:~/hw4/matmul$ ./run_performance.
salloc: Pending job allocation 134570
salloc: job 134570 queued and waiting for resou
salloc: job 134570 has been allocated resources
salloc: Granted job allocation 134570
(c11) Hello world, rank 0 out of 4
(c12) Hello world, rank 1 out of 4
(c13) Hello world, rank 2 out of 4
(c14) Hello world, rank 3 out of 4
Options:
  Problem size: M = 8192, N = 8192, K = 4096
  Number of threads per process: 32
  Number of iterations: 10
  Print matrix: off
  Validation: on

[rank 0] Initializing matrices...Done!
[rank 0] Warming up...
[rank 0] Calculating...(iter=0) 0.683846 sec
[rank 0] Calculating...(iter=1) 0.715141 sec
[rank 0] Calculating...(iter=2) 0.903678 sec
[rank 0] Calculating...(iter=3) 0.687797 sec
[rank 0] Calculating...(iter=4) 0.712965 sec
[rank 0] Calculating...(iter=5) 0.904192 sec
[rank 0] Calculating...(iter=6) 0.697587 sec
[rank 0] Calculating...(iter=7) 0.781605 sec
[rank 0] Calculating...(iter=8) 0.901266 sec
[rank 0] Calculating...(iter=9) 0.691133 sec
Validating...
Result: VALID
[rank 0] Avg. time: 0.767921 sec
[rank 0] Avg. throughput: 715.901644 GFLOPS
salloc: Relinquishing job allocation 134570
```