CS596 - Assignment3

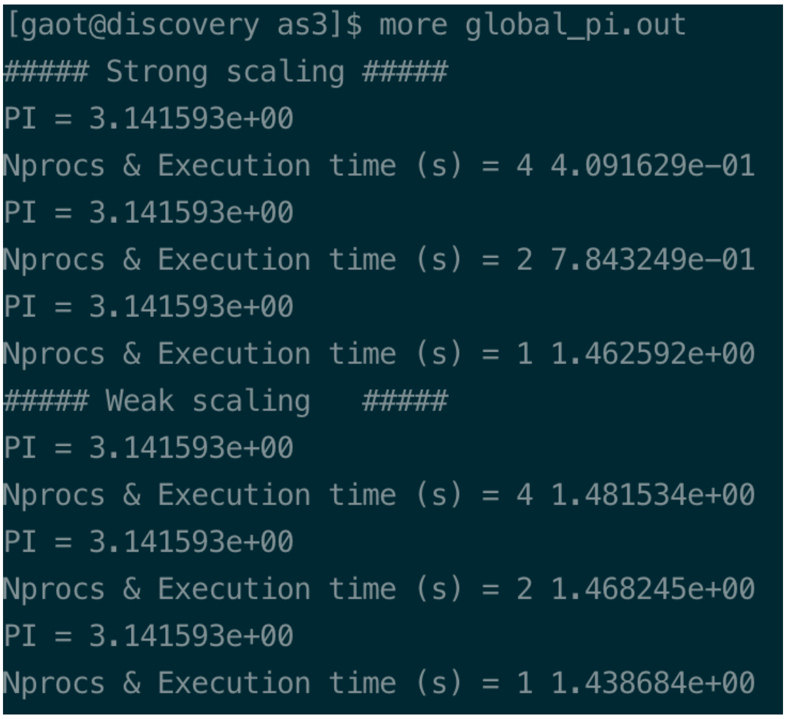
Part I: Programming

global_pi.c

```c
#include "mpi.h"
#include <stdio.h>
#define NBIN 1000000000

int nprocs;  /* Number of processes */
int myid;    /* My rank */

double global_sum(double partial) {
  /* Write your hypercube algorithm here */
  MPI_Status status;
  int bitValue,partner;
  double mydone, hisdone;

  mydone = partial;

  for(bitValue=1; bitValue<nprocs; bitValue*=2){
    partner = myid ^ bitValue;
    //send mydone to partner;
    MPI_Send(&mydone,1,MPI_DOUBLE,partner,bitValue,MPI_COMM_WORLD);
    //receive hisdone from partner;
    MPI_Recv(&hisdone,1,MPI_DOUBLE,partner,bitValue,MPI_COMM_WORLD, &status);
    mydone += hisdone;
  }
  return mydone;
}

int main(int argc, char *argv[]) {
  double partial;
  double cpu1, cpu2;
  long long i;
  double step, x, sum=0.0, pi;

  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &myid);
  MPI_Comm_size(MPI_COMM_WORLD, &nprocs);

  cpu1 = MPI_Wtime();

  step = 1.0 / NBIN;
  for (i = myid; i < NBIN; i += nprocs) {
    x = (i+0.5)*step;
    sum += 4.0/(1.0+x*x);
  }

  partial = sum * step;
  pi = global_sum(partial);

  cpu2 = MPI_Wtime();

  if (myid == 0) {
    printf("PI = %le\n",pi);
    printf("Nprocs & Execution time (s) = %d %le\n",nprocs,cpu2-cpu1);
  }

  MPI_Finalize();
  return 0;
}
```

Part II: Scalability

Perform the entire scaling tests and get result



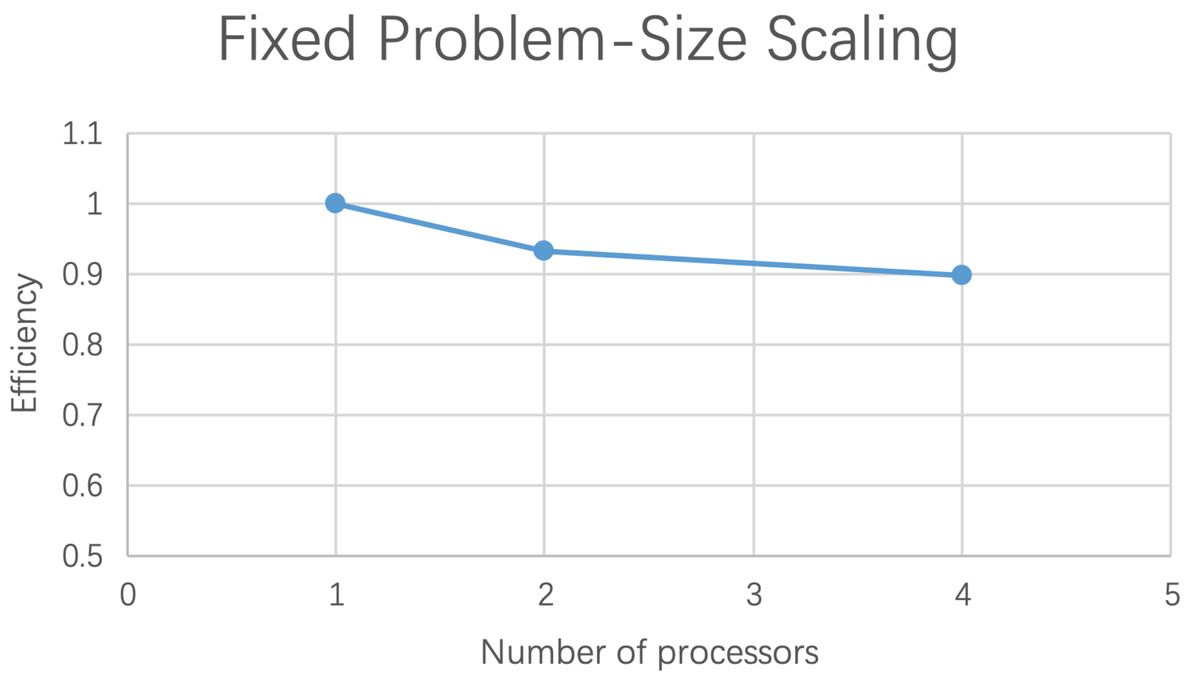
```
[gaot@discovery as3]$ more global_pi.out
##### Strong scaling #####
PI = 3.141593e+00
Nprocs & Execution time (s) = 4 4.091629e-01
PI = 3.141593e+00
Nprocs & Execution time (s) = 2 7.843249e-01
PI = 3.141593e+00
Nprocs & Execution time (s) = 1 1.462592e+00
##### Weak scaling   #####
PI = 3.141593e+00
Nprocs & Execution time (s) = 4 1.481534e+00
PI = 3.141593e+00
Nprocs & Execution time (s) = 2 1.468245e+00
PI = 3.141593e+00
Nprocs & Execution time (s) = 1 1.438684e+00
```

Fixed problem-size scaling

Calculate Effciency

Nproc	T	Sp	Ep
1	1.46E+00	1.00E+00	1.00E+00
2	7.84E-01	1.86E+00	9.32E-01
4	4.09E-01	3.57E+00	8.94E-01

Plot the fixed problem-size parallel efficiency as a function of P.

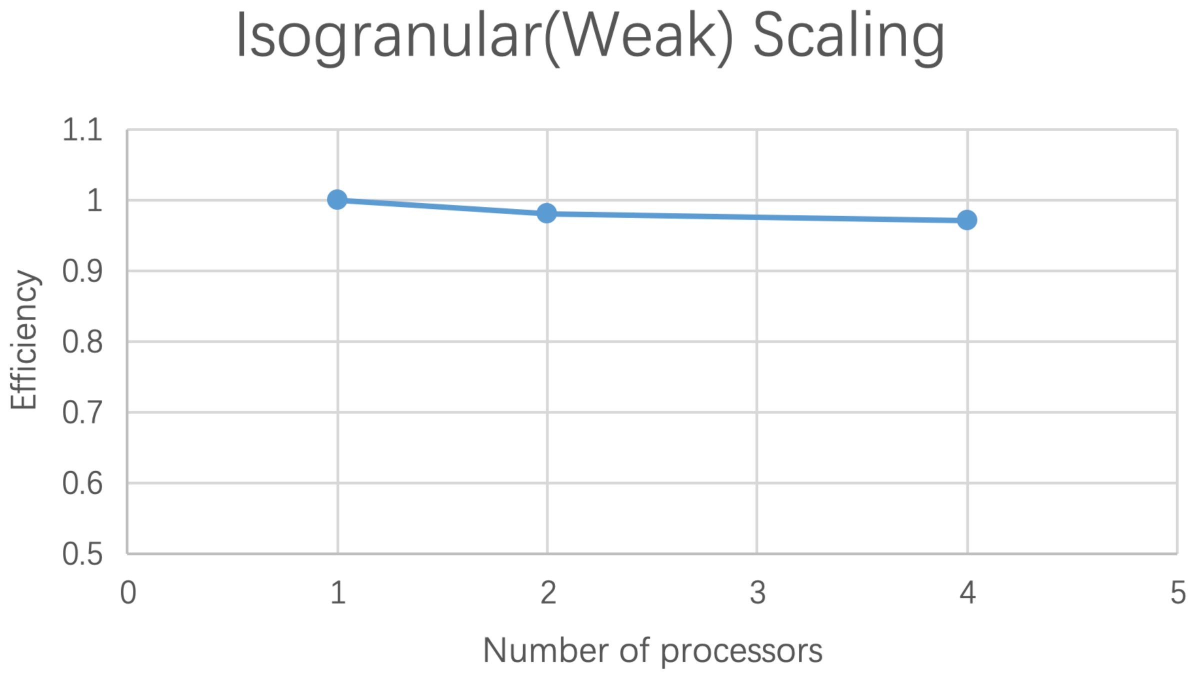


Isogranular scaling

Calculate Effciency

Nproc	T	Ep
1	1.44E+00	1.00E+00
2	1.47E+00	9.80E-01
4	1.48E+00	9.71E-01

Plot the isogranular parallel efficiency as a function of P


Isogranular(Weak) Scaling
Efficiency
1.1
1
0.9
0.8
0.7
0.6
0.5
0
1
2
3
4
5
Number of processors