Final Sample 2021 [CLOSED BOOK]

25 marks (1 hour 20 min)

1. Write a single sentence about any one aspect of MPI that you liked? [1 mark]

2. In one or two sentences, explain the difference between MPI_Send and MPI_Isend? [1 mark]

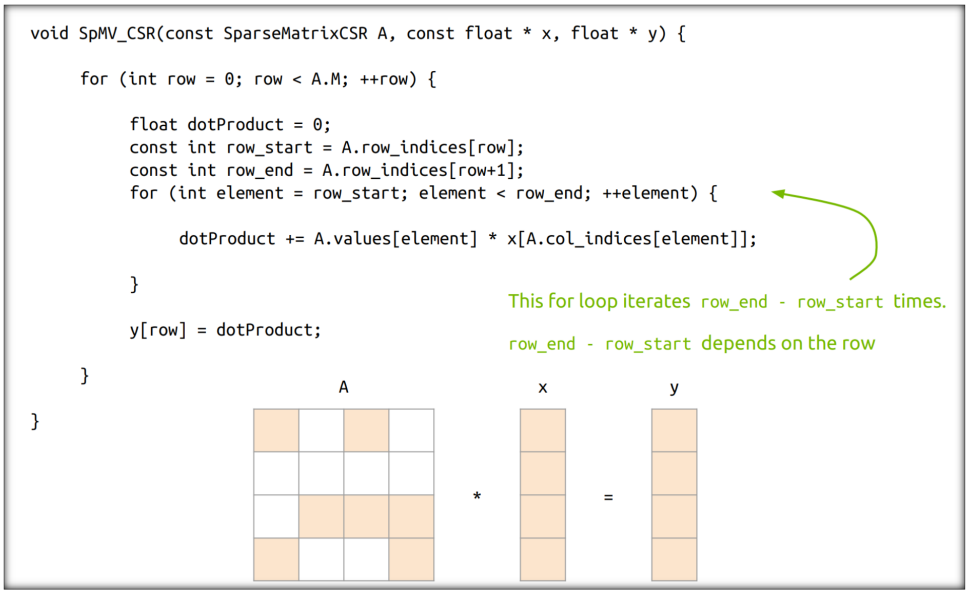
3. Provide an intuitive explanation (formal definition is also fine) of what efficiency is in terms of P and N, where P is the number of processors and N is the problem size, in or two sentences [1 mark]

4. OpenMP provides multiple mechanisms to control the number of threads. In one or two sentences explain one such method. [1 mark]

5. Explain the difference between latency and throughput [1 mark]

6. COO vs. CSR vs. CSC [5 marks]

a. How would you implement CSR, CSC, and COO-based MPI (no need to write code)? How would you distribute the work across multiple nodes? Name the MPI primitives that you will use and why?(e.g., send, recv, broadcast reduction)? Do you need MPI barrier? Compare the advantages and disadvantages of each format with respect to an MPI-based SpMV implementation.



8. Write all data dependences in the following codes (concise representation is fine for part B) [5 marks]

a)
```
a = b + c
b = b*2
b = b - 2;
d = b + a;
c = (a>b)? a:b;
```

b)
```
for (i =3; i<10; i++ )
    for (j =3; j<20; j++ )
        for(k=20; k<42; k++)
            A[i][j] += C[i][k] / D[k][j]
            B[i][j][k] = B[i-1][j][k] + E[i][j][k]
            F[i][j][k] = F[i][j-1][k] + E[i][j][k]
            G[i][j][k] = G[i][j][k-1] + E[i][j][k]
```

9. Explain 2D Summa algorithm and Cannon algorithm for matrix multiplication and summarize their differences [5 marks]

10. Filtering [5 marks]

a) Assume that you are given a dense matrix of size M*N where most elements are zero. Let's assume that the dense matrix is too big to fit in any single node; hence it is partitioned into chunks of size M*N/P and distributed among P processors (in other words the first M rows are in processor 0, the next set of M rows are in processor 1 and so on). Assume that M is perfectly divisible by P. Let's assume that the CSR format of this array can fit in a single node. Write code to convert the dense array to CSR format using MPI. The final result should be in node 0.

b) Explain your design considerations and potential performance problems with your code.

Note: For part a) you are only required to write code for the parallel part. You can skip other parts like initialization, printing, etc.