4.2. Amdahl's Law (15 points)

This problem is concerned with the speedup S of solving a problem. The efficiency E of this parallel approach can be calculated with E = S/p where p is the number of processing elements.

1. Consider a parallel program for which serial work comprises 8%. Given that the speedup is given by

   $$S = \frac{p}{B \times p + (1-B)},$$

   where B denotes the fraction of serial work to be done (i.e. B = 8%), calculate the maximum number p of processing elements such that the parallel efficiency E is at least 60%.

2. Consider a matrix $A \in \mathrm{IR}^{n \times n}$. A smoothing algorithm calculates the mean of each coefficient a_{ij} and its four neighbours before the new value a'_{ij} is stored at position (i, j) again—one single smoothing calculation takes t_{smooth} time. For a parallel approach with p processing elements the matrix A shall now be divided into p parts, with each part consisting of n/p rows and n columns. For calculating the mean at the border of each part, each processing element has to exchange its border values with its direct neighbours — a single send/receive of one value takes t_{comm} time. The communications between different pairs of processing elements can be performed in parallel, but within the same pair of processing elements, the communications must be in serial. Calculate the maximum amount p of processing elements as a function of n, t_{smooth} and t_{comm} so that the parallel efficiency E is at least 50%.

3. Explain the difference between Amdahl's Law and Gustafson's model. (Hint: Consider the different definitions of the sequential part of a parallel program). Discuss which model is best for measuring the performance of parallel algorithms.

By Jiahui Tang

Question 1.

Since $B = 8\%$, and

$$S = \frac{p}{B * p + (1 - B)}$$

we have

$$S = \frac{p}{0.08 * p + 0.92}$$

And to derive for Efficiency E, we get

$$E = \frac{S}{p} = \frac{p}{p * (0.08 * p + 0.92)} = \frac{1}{(0.08 * p + 0.92)} \geqslant 60\%$$

Thus, we have

$$1 \geqslant 0.6 * (0.08 * p + 0.92) = 0.048p + 0.552$$

$$0.448 \geqslant 0.048p$$

$$9.333 \geqslant p$$

Thus, the maximum number of processor is **9** to acheive at least 60% efficiency.

Question 2.

1 Proccessor:

For $n * n$ matrix, if we one processor, the smooth algorithm time is

$$T(n,1) = T_{smooth} * n^2$$

P Proccessor:

As the matrix is partitioned into P parts for p processing elements, each elements with $\frac{n}{p}$ rows and n columns. The communication happened at border of two processors at row level could also be done in parallel, it takes a time of communicating one row of border's elements, which is

$$n * T_{comm}$$

As smooth algorithm could be done in parallel, the time it takes to compute is

$$T_{smooth} * n * n/p$$

Thus, in total, the time it takes is:

$$T(n,p) = n * T_{comm} + \frac{T_{smooth} * n^2}{p}$$

For Efficiency E we get an at least 50%,

$$E = \frac{S}{p}$$

Where S the speed up is

$$S = \frac{T(n,1)}{T(n,p)} = \frac{T_{smooth} * n^2}{n * T_{comm} + \frac{T_{smooth} * n^2}{p}}$$

Thus,

$$E = \frac{S}{p} = \frac{T_{smooth} * n^2}{p * n * T_{comm} + p * \frac{T_{smooth} * n^2}{p}}$$

$$E = \frac{S}{p} = \frac{T_{smooth} * n}{p * T_{comm} + T_{smooth} * n} \geqslant 50\%$$

we can get the expression for maximum amount of p:

$$\frac{n * T_{smooth}}{T_{comm}} \geqslant p$$

Question 3

Amdahl's Law keeps the problem size fixed, and states that potential program speed up is defined by the fraction of code (c) that could be parallelized. And speedup is limited by this sequential code. Even a small fraction of sequential code will be the bottonneck that greatly limit potential speedup.

Thus, under Amdahl's Law, the speed up is under Strong Scaling

$$S_T : \lim_{p \to \infty} S_T = \frac{1}{1-c}$$

However, in Gustafson's Law, it states larger system should be used to solve larger problems. And ideally there should be scaled problem size but not a fixed problem size, where there will be a fixed amount of parallel work per processor. Thus, when we increase the problem size, the number of processing units are also increased to keep the fraction of time the code is executed in parallel constant. In this case, the sequential part's execution time also increases with N, which now becomes a measure of the problem size. I think this is the key difference in terms of different definition of sequential part of a parallel program. In Amdahl's Law, the sequential part is fixed size, and in Gustafson's Law, the sequential part is fixed proportion.

Under Gustafson's Law, it's Weak Scaling, where problem size per processor stays the same as more processors are added.

$$S_T(n,p) = 1 - c + c * p$$

To select the best model for measuring performance of parallel programs, Amdahl's Law should be used when the algorithm is designed to reduce execution time and run same problem faster; Gustafson's Law should be used when the algorihtm is designed to run a larger size problem in the same amount of time, keeping problem size per processor staying the same when more processors are added in.

```
In [ ]:
```