COSC 462 Fall 2024 : Term Project

Date: Nov 4, 2024 **Total points: 140** **Due: 11:59 PM, Dec 1, 2024**

Consider two $n \times n$ matrices A and B, both of whose elements are all ones. Let $C = A \times B$. Clearly, all the elements of C are equal to n.

(70 points) **Cannon's Algorithm**
The task is to implement Cannon's algorithms to compute the matrix $C = A \times B$ in parallel using a $p \times p = P$ array of processes where $p \leq n$.

(i) (5 points) Write a code that generates two $n \times n$ matrices, A and B, of ones where n is a user input.

(ii) (10 points) Modify the above code such that it accepts a processor count P and the array configuration $p \times p$, partitions each matrix into $\frac{n}{p} \times \frac{n}{p}$ sub-matrices and sends each sub-matrix to the appropriate processor in the $p \times p$ array.

(iii) (35 points) With $n \times n = 1024 \times 1024$, compute C on $P = 1\ (1 \times 1), 16\ (4 \times 4), 64\ (8 \times 8)$ and $256\ (16 \times 16)$ processes using Cannon's algorithm (5 points for $P = 1$ and 10 points each for $P = 16, 64$ and 256).

(iv) (5 points) Plot a graph of P vs. computation time.

(v) (5 points) Plot a graph of P vs. communication time.

(vi) (5 points) Plot a graph of P vs. total parallel execution time.

(vii) (5 points) Plot a graph of P vs. speedup, $S(P)$.

(70 points) **DNS Algorithm**
For this problem, use the same matrices A and B that you generated in the previous problem. The task is to implement the DNS algorithm to compute the matrix $C = A \times B$ in parallel using a $p \times p \times p = P$ array of processes where $p \leq n$.

(i) (50 points) With $n \times n = 1024 \times 1024$, compute C on $P = 1\ (1 \times 1 \times 1), 8\ (2 \times 2 \times 2)$, $64\ (4 \times 4 \times 4)$ and $512\ (8 \times 8 \times 8)$ processes using the DNS algorithm (5 points for $P = 1$ and 15 points each for $P = 8, 64$ and 512).

(vi) (5 points) Plot a graph of P vs. computation time.

(vii) (5 points) Plot a graph of P vs. communication time.

(viii) (5 points) Plot a graph of P vs. total parallel execution time.

(ix) (5 points) Plot a graph of P vs. speedup, $S(P)$.