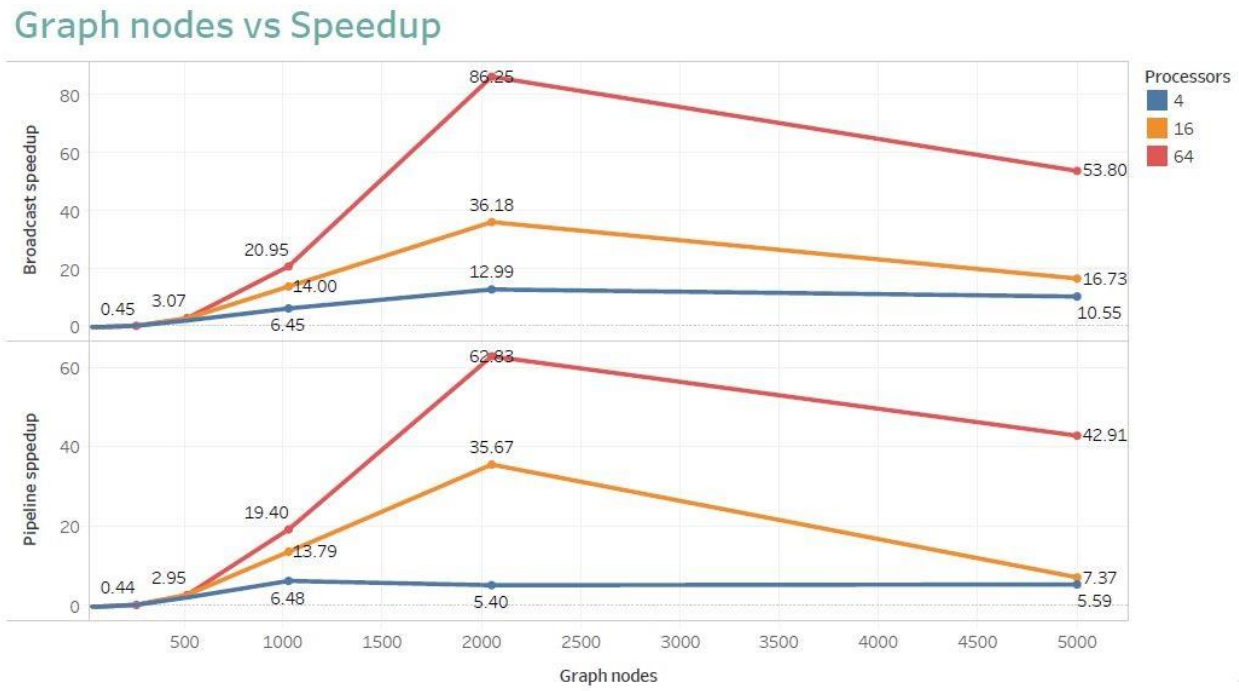
Parallel Programming Assignment 3

Shyam Kantesariya (40042715)

Question 1



Following are the observations we can derive from graph

- For smaller graphs, we don't see much speed up no matter how many processers we deploy
- Speedup increases with increase in graph size up to certain level and then starts decreasing
- For graph size close to 2000, we see a super linear speed up scenario for both types of parallel algorithms
- Though theoretically, we should be getting higher speedup for pipeline communication technique compared to Broadcast, however in practice it is totally reverse because of the way how MPI handles buffer in sync and async messages.

Question 2

Advantages:

- When a set of processors completes its work, the processors are reassigned to help evaluate other parts of the tree. This results in efficient load balancing in irregular trees.
- Guaranteed termination with no false termination as in algorithm terminates though there is some pending work.

Disadvantages:

- Once the reallocation is completed, the parent process blocks once again, awaiting another message.
- The processor is allocated to the first child, and then the parent process blocks, waiting for the child to complete.
- Uneven work distribution for unbalanced tree
- Many processors remain unassigned till the time we reach to the leaf of tree, which increases parallel cost.
- Communication happens only through parent process, which introduces delay in new work assignment if a processor is available to take up new work.

Question 3

We would prove the accuracy of algorithm by negation method, that is let's assume that the process who initiated the token received white token which is the pre-condition for termination and then prove it wrong.

For this situation to happen, the last processor in the ring should need to pass a white token. Tokens are passed in two ways, either the token is passed as-is, or it converts to black if the process is black.

Since we know that the token originates as white from the initiator processor, then if at any point the token encounters a non-idle processor, it will be turned black, and cannot be turned white again.

So either the token will encounter only white processors, meaning the program is complete, or the token will arrive black. This implies therefor that it is impossible to receive a white token unless the processes have all successfully terminated, proving this algorithm by negation.

Question 4

(a) For determining longest common subsequence, in block mapping two columns are assigned to each processor and processing elements are n/p.

If we consider nxn matrix,

Sequential time, $T_s = O(n^2)\, t_c$

Total parallel time, $T_p = (2n/p-1)\, t_c + t_s + t_w$

$E = T_s / pT_p$

$$= \frac{O(n^2)\, t_c}{n/p\,(2n/p-1)\, t_c + t_s + t_w}$$

For maximum efficiency, ts=tw=0

$$E_{max} = \frac{np}{2n/p-1}$$

For 4x4 matrix, given n=4 and p=2

$$E_{max} = \frac{4*2}{2*4/2-1} = 2.66$$

(b) If we consider block- cyclic mapping instead of block mapping in the problem defined in (a),then would not be any difference in maximum efficiency obtained because in block-cyclic also there are n/p elements and each processor will be assigned two blocks of 2x2 (in (a) two columns are assigned to each processor).

$$E_{max} = \frac{np}{2n/p-1}$$