HW1:

(1) Program — 1000ms
in 10 processors in 120ms

a) = Speedup = $\frac{\text{old time}}{\text{New time}}$ = $\frac{1000}{120}$ = 8.33

b) Efficiency: $\frac{S}{N}$ = $\frac{8.33}{10}$ = 83.3 %

c) Overhead: T_0 = PTp − T_s = 10×120 − 1000 = 200 ms

d) T_p for 0.9 efficiency:

T_p = $\frac{1000}{0.9 \times 10}$ = 111.1 ms

(2) 8 nodes each 4 cores
1 MB data

a) 8 × 4 = 32 cores

Each node communicated with each other so $\frac{n(n-1)}{2}$

Total exchanges = $\frac{32 \times 31}{2}$ = 496 messages

b) Total Volume:

496 × 1 = 496 MB of data

c) Theoretical MIN transfer time:

$$10 \text{ Gbps} = \frac{10}{8} = 1.25 \text{ GB/s}$$

$$\text{Transfer time} = \frac{496/1024 \text{ GB}}{1.25 \text{ GBPS}} = 0.38 \text{ s}$$

(3)

Task 0 : Send (to: 1) ; Barrier();

Task 1 : Recv (from: ANY); Recv (from: 2) ; Barrier();

Task 2 : Send (to: 1); Barrier();

1. Terminates:

If happens in order $0 \rightarrow 1 \rightarrow 2$

0 sends msg to 1

1 recv msg from 0

2 sends to 1

1 recv from 2

Termination

2. Deadlocks: $2 \rightarrow 1 \rightarrow 0$

If 2 sends msg

1 can recv from any. Now 0 sends msg.

Now 0 doesnt hit barrier but 2 doesnt either. Now all tasks are blocked.

Creates a deadlock

3. Deadlock solution:

- If task 1 is changed from Recv(from:ANY) to Recv(from:1) it can receive ① & ②'s messages without causing a deadlock.
- We could replace all the Send/Recv with non-blocking Send/Recv or Isend & Irecv. This will avoid deadlocks.