

Name: Tran Thi Thoa

ID: s1242006

Concurrent and distributed systems

Exercise 7.1: The benefits of Asynchronous processing.

We have 2 computers, connected with a 100 Mbit/s link. One array block consists of 500000 integers (each one has 32 bits).

We estimate the speedup achieved by using asynchronous send/receive operations comparing to the ordinary send/receive processing when sorting 100 array blocks.

Therefore the time for sending/receiving 1 array in this situation is:

$$T = \frac{100 \cdot 10^6}{500000 \cdot 32} = 6.25(s)$$

The time for sorting 1 array block is $t = 0.08(s)$ (using my own laptop).

+ Using synchronous send/receive operation, we need the time to run the program is

$$T_1 = (2 * T + t) * 100 (s) = 1258$$

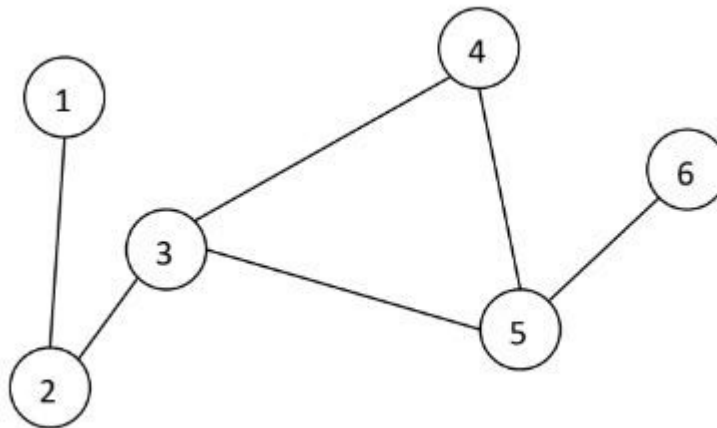
+ Using asynchronous send/receive operation, because the time for send/receive process is much bigger than the time for sorting an array, we need the time to run the program is:

$$T_2 = 200 * T (s) = 1250$$

So the speedup of the asynchronous operation compared to the ordinary one is:

$$\text{Speedup} = T_1/T_2 = 1.0064 \text{ times}$$

Exercise 7.2: Define Topology



]

procID	neighbors	nbrsSum
1	2	1
2	1,3	3
3	2,4,5	6
4	3,5	8
5	3,4,6	11
6	5	12

Index = {1,3,6,8,11,12}

Edges = {2,1,3,2,4,5,3,5,3,4,6,5}

Exercise 7.4: Money Space

Implement a shared bank account system in Javaspace. There is a shared bank account (initially empty) and 2 user processes, each has to add 500000 money units to the account.

Consider the shared bank account system is the shared space in Javaspace, the account that we have to add the money units into it is an object in this shared space and 2 users are 2 parallel processes that can read and take an object from the shared space.

Therefore, the pseudocode for that program using Javaspace is:

Initiate the account object Acc in shared space which has value = 0;

Initiate 2 user processes;

int count = 500000;

int x;

Process 1:

```
x = read(Acc);
for(count = 500000, count > 0; count - -) {
    x ++;
}
count = 500000;
write(x, Acc);
```

Process 2:

```
x = read(Acc);
for(count = 500000, count > 0; count - -) {
    x ++;
}
count = 500000;
write(x, Acc);
```