

Μάριος Κωνσταντίνος Κωνσταντάκης, A.M: 3219.
csd3219@csd.uoc.gr

Γενικά, νομίζω ότι η παρούσα εφαρμογή, επειδή προσομοιώνει μία επιδημία, όπου τα δεδομένα αλλάζουν σταδιακά, μέρα με τη μέρα, είναι φύση μη παράλληλο πρόγραμμα, αφού για την ημέρα N πρέπει πρώτα να υπολογίσω την προηγούμενη N-1, και για αυτήν την N-2 κλπ. Ακόμα και αν, έβαζα όλο τον κώδικα στην main, χωρίς άλλες συναρτήσεις, θεωρώ πως δεν μπορεί να τρέξει εύκολα ταυτόχρονα 2 συναρτήσεις με πολλαπλά threads εύκολα, και το αποδεικνύει αυτό η κατακόρυφη αύξηση του χρόνου εκτέλεσης μόλις βάλαμε τα threads στο παιχνίδι. Έχω busy wait, δηλαδή. Είτε γίνει υλοποίηση με OpenMP ή pthreads.

Για τους αρχικά μολυσμένους κόμβους έφτιαξα 1 δικό μου μικρό αρχείο (MyInfectedNodes.txt), με διάφορα τυχαία νούμερα πχ 1, 2, 666, 420 κλπ. Το 2ο αρχείο (infectedNodes.txt) το έκανα generate μέσω του infectedPeopleRandomGenerator.c με τυχαίους αριθμούς στο εύρος των μεγεθών των δοσμένων testfiles. Βγήκαν 4664.

Ο γράφος μοντελοποιείται από μία απλά συνδεδεμένη λίστα από λίστες επαφών/γειτόνων του κάθε κόμβου (μέσω του πεδίου neighbors στο struct Κομνος), η οποία γεμίζει σαν στοίβα, από το τέλος προς την αρχή. (last in, first out-LIFO)

Σειριακό πρόγραμμα:

```
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 30 infectedNodes.txt
The program has been running for 6.050034 seconds, with N=30, testfile=facebook_combined.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 p2p-Gnutella24.txt 30 infectedNodes.txt
The program has been running for 35.469748 seconds, with N=30, testfile=p2p-Gnutella24.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 Email-Enron.txt 30 infectedNodes.txt
The program has been running for 329.074615 seconds, with N=30, testfile=Email-Enron.txt.
```

```
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 180 infectedNodes.txt
The program has been running for 6.053654 seconds, with N=180, testfile=facebook_combined.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 p2p-Gnutella24.txt 180 infectedNodes.txt
The program has been running for 34.652245 seconds, with N=180, testfile=p2p-Gnutella24.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 Email-Enron.txt 180 infectedNodes.txt
The program has been running for 331.520358 seconds, with N=180, testfile=Email-Enron.txt.
```

```
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 365 infectedNodes.txt
The program has been running for 6.341121 seconds, with N=365, testfile=facebook_combined.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 p2p-Gnutella24.txt 365 infectedNodes.txt
The program has been running for 35.296259 seconds, with N=365, testfile=p2p-Gnutella24.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 Email-Enron.txt 365 infectedNodes.txt
The program has been running for 343.600518 seconds, with N=365, testfile=Email-Enron.txt.
```

Παράλληλο πρόγραμμα:

```
gcc -g -fopenmp ask1.c -pthread -o ask1
export OMP_NUM_THREADS=1
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 30 infectedNodes.txt
The program has been running for 235.566629 seconds, with N=30, testfile=facebook_combined.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 p2p-Gnutella24.txt 30 infectedNodes.txt
Segmentation fault
csd3219@anafi:~/hy342/ask1$ ./ask1 Email-Enron.txt 30 infectedNodes.txt
Segmentation fault
```

```
gcc -g -fopenmp ask1.c -pthread -o ask1
export OMP_NUM_THREADS=2
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 180 infectedNodes.txt
Segmentation fault
csd3219@anafi:~/hy342/ask1$ ./ask1 p2p-Gnutella24.txt 180 infectedNodes.txt
Segmentation fault
csd3219@anafi:~/hy342/ask1$ ./ask1 Email-Enron.txt 180 infectedNodes.txt
Segmentation fault
```

```
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 365 infectedNodes.txt
The program has been running for 260.098108 seconds, with N=365, testfile=facebook_combined.txt.

csd3219@anafi:~/hy342/ask1$ make
gcc -g -fopenmp ask1.c -pthread -o ask1
export OMP_NUM_THREADS=4
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 365 infectedNodes.txt
Segmentation fault
csd3219@anafi:~/hy342/ask1$ ./ask1 facebook_combined.txt 365 infectedNodes.txt
The program has been running for 266.689187 seconds, with N=365, testfile=facebook_combined.txt.

csd3219@anafi:~/hy342/ask1$ ./ask1 p2p-Gnutella24.txt 365 infectedNodes.txt
Segmentation fault
csd3219@anafi:~/hy342/ask1$ ./ask1 p2p-Gnutella24.txt 365 infectedNodes.txt
Segmentation fault
csd3219@anafi:~/hy342/ask1$ ./ask1 Email-Enron.txt 365 infectedNodes.txt
The program has been running for 10090.949233 seconds, with N=365, testfile=Email-Enron.txt.
```

```
cat /proc/cpuinfo | grep processor | wc -l
```

Με αυτή την εντολή μπορώ να δω πόσους επεξεργαστές έχει το μηχάνημα Debian που τρέχω την άσκηση.

Όπως βλέπουμε, από την παρακάτω εικόνα είναι 6 οι πυρήνες του μηχανήματος naxos.

```
csd3219@naxos:~/hy342/ask1$ ./epidemicSimulator facebook_combined.txt 30 infectedNodes.txt
The program has been running for 238.911594 seconds, with N=30, testfile=facebook_combined.txt.

csd3219@naxos:~/hy342/ask1$ ./epidemicSimulator p2p-Gnutella24.txt 30 infectedNodes.txt
Segmentation fault
csd3219@naxos:~/hy342/ask1$ ./epidemicSimulator p2p-Gnutella24.txt 30 infectedNodes.txt
Segmentation fault
csd3219@naxos:~/hy342/ask1$ ./epidemicSimulator Email-Enron.txt 30 infectedNodes.txt
The program has been running for 10406.727030 seconds, with N=30, testfile=Email-Enron.txt.

csd3219@naxos:~/hy342/ask1$ cat /proc/cpuinfo | grep processor | wc -l
6
csd3219@naxos:~/hy342/ask1$ ./epidemicSimulator facebook_combined.txt 180 infectedNodes.txt
The program has been running for 233.071476 seconds, with N=180, testfile=facebook_combined.txt.

csd3219@naxos:~/hy342/ask1$ ./epidemicSimulator p2p-Gnutella24.txt 180 infectedNodes.txt
Segmentation fault
```

```
csd3219@naxos:~/hy342/ask1$ ./epidemicSimulator Email-Enron.txt 180 infectedNodes.txt
The program has been running for 10161.087142 seconds, with N=180, testfile=Email-Enron.txt.
```

1) Το average speedup είναι η μέση τιμή (average) του $\text{speedup} \leq 1 / ((1-p) + p/n)$, όπου $n=6$ processors, p είναι τα παράλληλα παρτ του προγράμματος, και το $1-p$ τα σειριακά.

Ο παραλληλισμός με 1 thread, $p=1$ μας δίνει $\text{average speedup} = \text{avg}(\text{speedup} \leq 1 / ((1-p) + p/n)) = \text{avg}(1/(1-1) + 1/6) = \text{avg}(1/0 + 1/6) = \text{avg}(6)$.

Ο παραλληλισμός με 2 threads, $p=2$ λοιπόν μας δίνει $\text{average speedup} = \text{avg}(\text{speedup} \leq 1 / ((1-p) + p/n)) = \text{avg}(1/(1-2) + 2/6) = \text{avg}(1/-1 + 1/3) = \text{avg}(1/(-2/3)) = \text{avg}(-3/2)$.

Ο παραλληλισμός με 4 threads, $p=4$ μας δίνει $\text{average speedup} = \text{avg}(\text{speedup} \leq 1 / ((1-p) + p/n)) = \text{avg}(1/(1-4) + 4/6) = \text{avg}(1/-3 + 2/3) = \text{avg}(1/-7/3) = \text{avg}(-3/7)$.