

Μετρήσεις:

N:1.000 #: 10

#thread: 1	# thread: 50	# thread: 100
0,003280	0.011338	0.016725
0,003109	0.006505	0.014911
0,002643	0.007768	0.014801
0,002274	0.007097	0.014596
0,002889	0.007438	0.014827
0,003033	0.007495	0.016199
0,003431	0.008767	0.014134
0,003324	0.007499	0.015929
0,003538	0.006720	0.019119
0,004103	0.008546	0.015947
Μέσος Όρος: 0.0031624	Μέσος Όρος: 0.0079173	Μέσος Όρος: 0.0157188

N: 100.000 #10

# thread: 1	# thread: 50	# thread: 10
0,195187	0.046510	0,055297
0,152700	0.038532	0,047865
0,182787	0.040094	0,047551
0,165218	0.039791	0,049154
0,143828	0.040526	0,048379
0,111434	0.040303	0,054867
0,177238	0.040277	0,048686
0,148422	0.039304	0,046919
0,122588	0.040382	0,049366
0,081981	0.040123	0,049085
Μέσος Όρος: 0.1481383	Μέσος Όρος: 0.0405842	Μέσος Όρος: 0.0497169

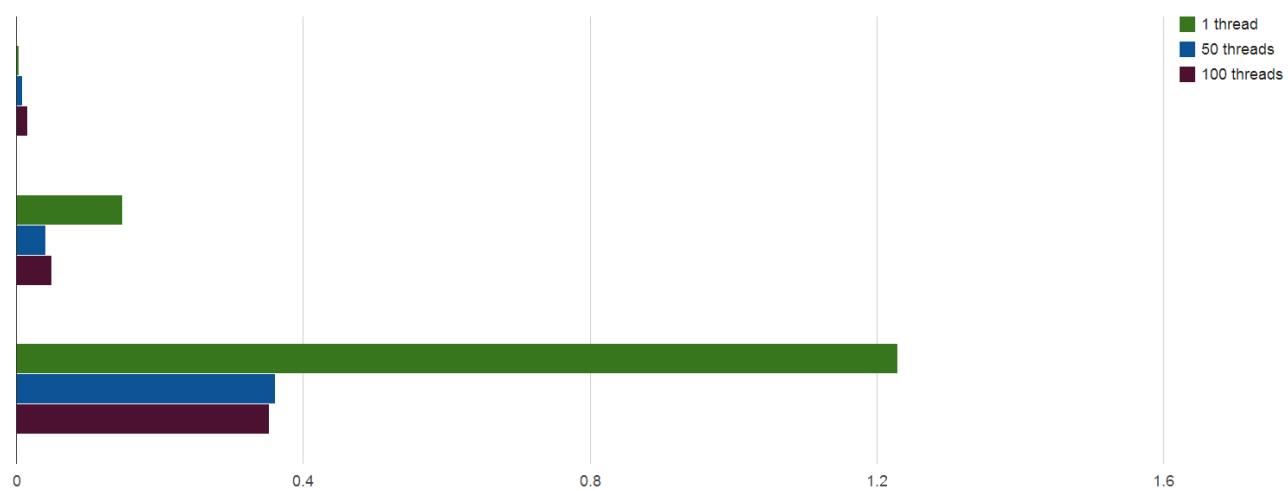
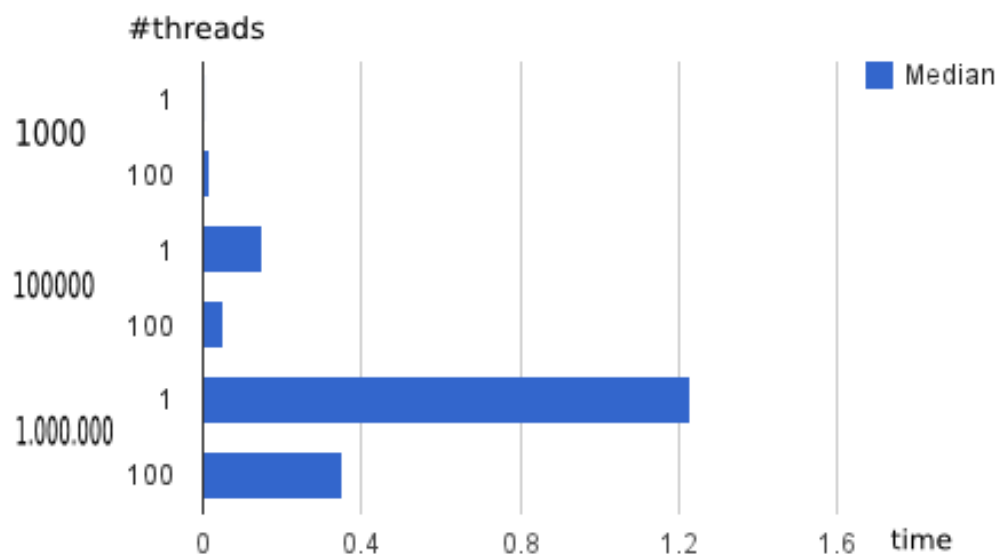
N:1.000.000 #10

#thread: 1	#thread: 50	#thread 100
1,274530	0.386364	0,333025
1,185644	0,391828	0,340859
1,200193	0,333712	0,372401
1,120712	0,360999	0,369894
1,048226	0,371232	0,354196
1,299043	0,346126	0,350759
1,345863	0,359781	0,365924
1,315845	0,341124	0,366870
1,269480	0,338484	0,334625
1,237172	0,380251	0,344710
Μέσος Όρος: 1.2296708	Μέσος Όρος: 0.3609901	Μέσος Όρος: 0.3533263

Οπότε:

N	M	Μέσος Όρος
1.000	1	0.0031624
1.000	50	0.0079173
1.000	100	0.0157188
100.000	1	0.1481383
100.000	50	0.0405842
100.000	100	0.0497169
1.000.000	1	1.2296708
1.000.000	50	0.3609901
1.000.000	100	0.3533263

Και το γράφημα μας:



Παρατηρήσεις, Επεξηγήσεις αποτελεσμάτων και Συμπεράσματα:

Από τις μετρήσεις, αντιλαμβανόμαστε την χρησιμότητα των threads για μεγάλο N, δηλαδή μεγάλο string. Ενώ από την άλλη, παρατηρούμε ότι για μικρά string, η χρήση πολλών threads μπορεί να καθυστερήσει ακόμα περισσότερο τον χρόνο εκτέλεσης της διαδικασίας. Αυτό συμβαίνει για τον εξής λόγο:

Σε ένα μικρό string, η ύπαρξη πολλών threads σπαταλάει πολύ χρόνο αφού το κόστος του συγχρονισμού τους δεν είναι αμελητέο. Συνεπώς, σε αυτή τη περίπτωση μπορεί η ύπαρξη ενός μόνο thread να είναι πολύ πιο αποδοτική από την χρήση πολλών νημάτων για παράλληλη επεξεργασία.

Όμως, η αύξηση των γραμμάτων του string, βλέπουμε ότι η χρήση πολλών νημάτων μας μειώνει πολύ τον χρόνο εκτέλεσης για την εύρεση του secret string. Αυτό συμβαίνει όταν το κόστος συγχρονισμού και δημιουργίας των νημάτων, καθώς και τα context switches στον επεξεργαστή είναι πολύ μικρότερα σε σχέση με τον χρόνο υπολογισμού του string.

Συνεπώς, καταλαβαίνουμε ότι στην άσκηση μας συμφέρει η δημιουργία πολλών threads για μεγάλα strings (πχ 100.000, 1.000.000 χαρακτήρες) ενώ λίγα για μικρό αριθμό γραμμάτων (π.χ. 10, 100, 1000 χαρακτήρες).

*Οι μετρήσεις έγιναν στον υπολογιστή μου:

Architecture:	i686
CPU op-mode(s):	32-bit, 64-bit
Byte Order:	Little Endian
CPU(s):	4
On-line CPU(s) list:	0-3
Thread(s) per core:	2
Core(s) per socket:	2
Socket(s):	1
Vendor ID:	GenuineIntel
CPU family:	6
Model:	37
Stepping:	5
CPU MHz:	1199.000
BogoMIPS:	5054.71
Virtualization:	VT-x
L1d cache:	32K
L1i cache:	32K

L2 cache:	256K
L3 cache:	3072K