



# ΠΑΡΑΛΛΗΛΟΣ ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

ΚΑΘΗΓΗΤΗΣ: ΜΑΡΓΑΡΙΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Αρχοντής-Εμμανουήλ Κωστής | ics21044

## ΕΡΓΑΣΙΑ 6 (2)

Ανάλυση Χρόνου Εκτέλεσης Sieve of Eratosthenes

Για τη διεξαγωγή των πειραμάτων, εκτελέσαμε κάθε πρόγραμμα 10 φορές στο ίδιο σύστημα. Τόσο η ακολουθιακή όσο και οι εκδοχές των παράλληλων εκτελέσεων εκτελέστηκαν υπό όσο το δυνατόν πιο πανομοιότυπες συνθήκες. Στην συνέχεια καταγράφηκαν οι χρόνοι εκτέλεσης για κάθε υλοποίηση και υπολογίστηκαν οι αντίστοιχοι μέσοι χρόνοι (σε ms).

**Μέγεθος Προβλήματος:** 1.000.000.000

**Υλοποιήσεις:**

1. Απλή Ακολουθιακή Εκτέλεση
2. Παράλληλη Εκτέλεση με Στατική Κατανομή
3. Παράλληλη Εκτέλεση με Κυκλική Κατανομή
4. Παράλληλη Εκτέλεση με Master Worker

A/A	Ακολουθιακά (1)	Στατική Κατανομή (2)	Κυκλική Κατανομή (3)	Master Worker (4)
1	18979	23831	13048	17783
2	19381	25710	14372	17548
3	21072	24719	14521	16809
4	19217	26522	13371	18669
5	18380	23991	13227	17748
6	19066	22737	13308	17387
7	19234	23881	13726	17557
8	18045	23684	12972	16598
9	18598	23226	12837	18046
10	19301	24640	12916	18400
M.O.	19127,3	24294,1	13429,8	17654,5

Η Ακολουθιακή Υλοποίηση αναμένεται να έχει τον μεγαλύτερο χρόνο εκτέλεσης, καθώς εκτελείται σε ένα μόνο νήμα χωρίς να αξιοποιεί πολλαπλούς επεξεργαστές και οι υπολογισμοί εκτελούνται ο ένας μετά τον άλλο, γεγονός που αυξάνει την ταχύτητα εκτέλεσης και περιορίζει την αποδοτικότητα. Με βάση τα πειράματα παρατηρούμε ότι ο μέσος χρόνος εκτέλεσης για την ακολουθιακή υλοποίηση είναι περίπου 19127 ms κάτι που δεδομένης της έλλειψης παραλληλισμού, αυτός ο μεγάλος χρόνος εκτέλεσης αναμενόμενος.

Στην *Παράλληλη Υλοποίηση με Στατική Κατανομή* οι εργασίες κατανέμονται ισομερώς μεταξύ των διαθέσιμων νημάτων και κάθε νήμα αναλαμβάνει έναν προκαθορισμένο αριθμό εργασιών. Αυτή η μέθοδος θα πρέπει θεωρητικά να μειώσει το χρόνο εκτέλεσης εφόσον οι εργασίες είναι ομοιόμορφες ως προς το μέγεθος τους και το χρόνο εκτέλεσης. Ωστόσο, εάν υπάρχει σημαντική διαφορά μεταξύ των εργασιών, κάποια νήματα ενδέχεται να ολοκληρώσουν την δουλειά τους νωρίτερα και να παραμείνουν αδρανή (idle), ενώ άλλα νήματα συνεχίζουν να “εργάζονται”, κάτι που μπορεί να οδηγήσει σε μεγαλύτερους χρόνους εκτέλεσης. Σύμφωνα με τα πειράματά μας, ο μέσος χρόνος εκτέλεσης της συγκεκριμένης υλοποίησης είναι περίπου 24294 ms. Παρατηρούμε ότι ο χρόνος αυτός είναι αρκετά μεγαλύτερος από την αντίστοιχη ακολουθιακή υλοποίηση, κάτι που θα περιμέναμε να μην ισχύει αφού ο παραλληλισμός θα έπρεπε να μειώσει τον χρόνο εκτέλεσης. Αυτό μπορεί υποδηλώνει κάποια ανισορροπία ανάμεσα στον φόρτο εργασίας των νημάτων.

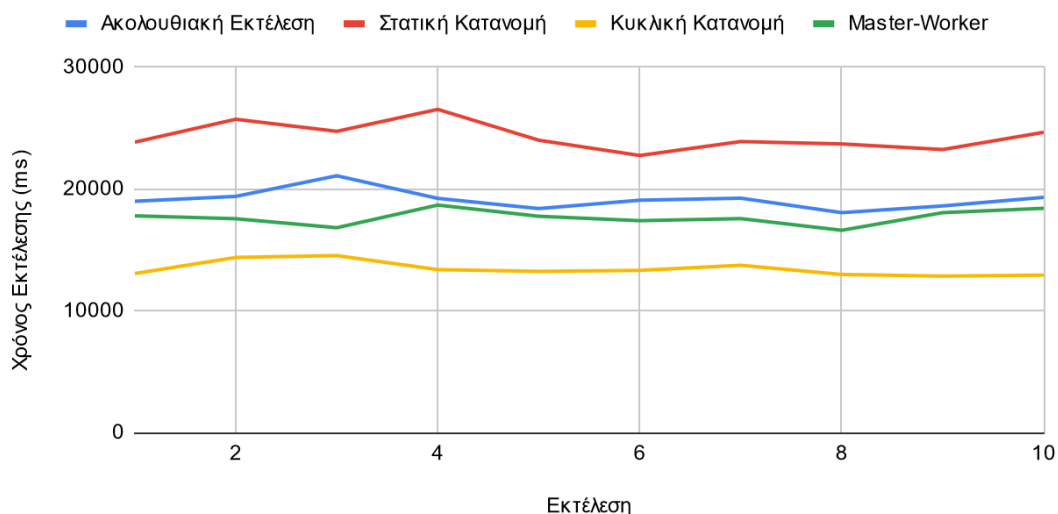
Χρησιμοποιώντας την μέθοδο της *Κυκλικής Κατανομής*, ο φόρτος εργασίας μοιράζεται στα νήματα με round robin λογική. Στην πράξη σπάμε και πάλι τον υπολογισμό σε τμήματα, και το Νήμα 1 παίρνει το 1ο τμήμα, το Νήμα 2 το 2ο κ.ο.κ. Μόλις φτάσουμε στο τελευταίο νήμα “πηγαίνουμε” ξανά στο 1ο Νήμα. Η μέθοδος αυτή εξισορροπεί τον φόρτο εργασίας των νημάτων με πιο ομοιόμορφο τρόπο, μειώνοντας τον χρόνο που κάποια νήματα είναι idle. Στα πειράματά μας οι μετρήσεις δείχνουν σημαντική βελτίωση στους χρόνους εκτέλεσης σε σύγκριση τόσο με την παράλληλη υλοποίηση με στατική κατανομή όσο και με την ακολουθιακή. Ο μέσος χρόνος εκτέλεσης για τη μέθοδο κυκλικής κατανομής είναι περίπου 13429ms κάτι που όπως είπαμε δείχνει σημαντική βελτίωση σε σχέση τόσο με την ακολουθιακή υλοποίηση όσο και με την παράλληλη υλοποίηση με στατική κατανομή.

Η τελευταία υλοποίηση είναι η *Παράλληλη Υλοποίηση με Master-Worker*. Σε αυτή την υλοποίηση το Νήμα “Αφέντης” (Master) κατανέμει δυναμικά τον φόρτο εργασίας σε ένα σύνολο Νημάτων που ονομάζονται “Εργάτες” (Worker). Οι εργάτες με την σειρά τους ζητούν νέες εργασίες καθώς ολοκληρώνουν τις τρέχουσες, εξασφαλίζοντας ότι εργάζονται συνεχώς χωρίς να παραμένουν αδρανείς. Αυτή η δυναμική κατανομή του φόρτου εργασίας μπορεί να προσαρμοστεί στο μέγεθος των εργασιών και στους διαφορετικούς χρόνους εκτέλεσης, παρέχοντας μια πιο αποτελεσματική εξισορρόπηση φορτου. Κάτι που αξίζει να σημειωθεί εδώ είναι πως όλη αυτή η διαδικασία μπορεί να εισάγει λίγο overhead λόγω της επικοινωνίας μεταξύ των εργατών και του master. Στα πειράματά μας, ο μέσος χρόνος εκτέλεσης της μεθόδου master-worker είναι περίπου 17654ms, όπου αν και είναι καλύτερος χρόνος από τη ακολουθιακή υλοποίηση, παρατηρούμε πως δεν είναι τόσο αποδοτικός όσο η μέθοδος κυκλικής κατανομής. Αυτό μπορεί να οφείλεται στην επιβάρυνση που εισάγεται, όταν χρησιμοποιούμε το μοντέλο master-worker, λόγω της δυναμικής ανάθεσης εργασιών και την επικοινωνία του νήματος Master με τα νήματα Workers. Παρά την επιβάρυνση αυτή, ο μέσος χρόνος εκτέλεσης υπερτερεί τον αντίστοιχο της ακολουθιακής, κάτι το οποίο περιμέναμε.

Παρακάτω ακολουθούν κάποια γραφήματα που συγκρίνουν τους χρόνους εκτέλεσης:

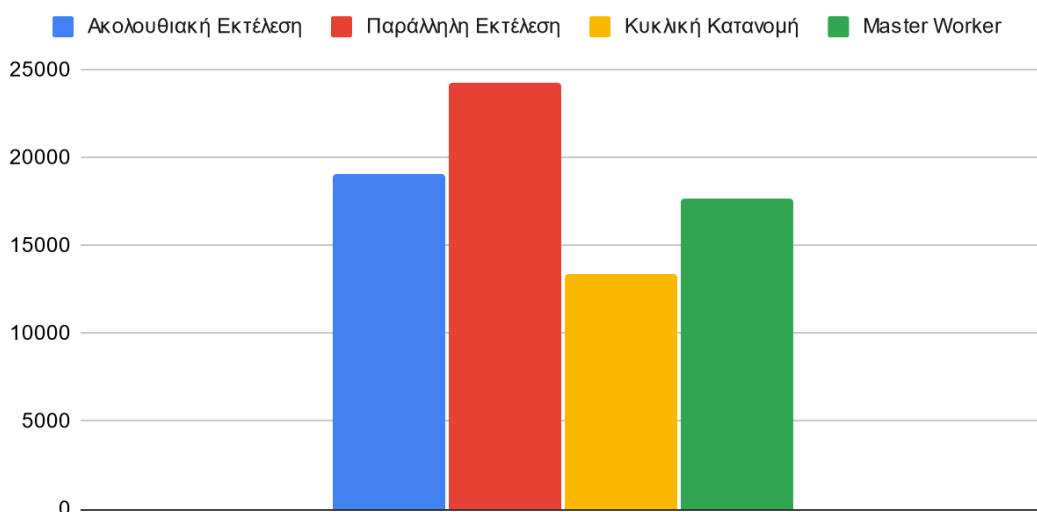
## Χρόνος Εκτέλεσης(ms)

Sieve Static Distribution



## Μέσος Χρόνος Εκτέλεσης

Sieve Static Distribution



Από τα πειράματά μας μπορούμε να βγάλουμε διάφορα συμπεράσματα σχετικά με την απόδοση των υλοποιήσεών μας. Η Ακολουθιακή Υλοποίηση, η οποία χρησιμοποιείται ως βάση των πειραμάτων μας, παρουσίασε τους μεγαλύτερους χρόνους εκτέλεσης, κάτι που περιμέναμε λόγω της έλλειψης παραλληλισμού. Η παράλληλη υλοποίηση με στατική κατανομή δείχνει να μην είναι πιο αποδοτική πιθανότατα λόγω νημάτων που καταλήγουν να είναι “αδρανή” κάτι που εισάγει overhead ενώ η υλοποίηση με master-worker παρέχει καλύτερη απόδοση μέσω της δυναμικής κατανομής εργασιών, εξασφαλίζοντας τη βέλτιστη αξιοποίηση του επεξεργαστή και τους συντομότερους χρόνους εκτέλεσης. Τέλος, την καλύτερη απόδοση από όλες τις υλοποιήσεις την έχει υλοποίηση με κυκλική κατανομή, μειώνοντας τον χρόνο αδράνειας και βελτιώνοντας σημαντικά τους χρόνους εκτέλεσης σε σχέση με τις υπόλοιπες μεθόδους.