

Παράλληλος και Κατανεμημένος Υπολογισμός

Εργαστήριο 7

1. Μελετήστε τους κώδικες Code/WordCound (κλασσικό παράδειγμα για MapReduce)

Φορτώστε το αρχείο bible.txt από το σύνδεσμο

<https://github.com/pfalcon/canterbury-corpus/blob/master/large/>

και τετραπλασιάστε (τουλάχιστο) το μέγεθός του. Εκτελέστε πειράματα για να μετρήσετε τον χρόνο.

Αν ο χρόνος είναι πολύ μικρός (ανάλογα με τον υπολογιστή σας) πολλαπλασιάστε πάλι το μέγεθος του αρχείου.

Προσπαθήστε να το παραλληλοποιήσετε τον κώδικα. Η εύκολη λύση (στη Java τουλάχιστο) είναι να βρείτε μια ταυτόχρονη δομή. Μια πιο σύνθετη λύση είναι να δοκιμάσετε με τοπικές δομές δεδομένων (σε κάθε νήμα) και αναγωγή στο τέλος.

Στους κώδικες δίνεται ένας κώδικας με Java Streams. Προσπαθείστε να παραλληλοποιήσετε και αυτόν. Συγκρίνετε τις επιδόσεις (τους χρόνους εκτέλεσης) κάθε υλοποίησης.

Οι σύνδεσμοι

<https://github.com/juditacs/wordcount>

<https://github.com/caiogranero/word-count-in-hadoop>

<https://github.com/austinjoyal/spark-word-count>

δίνουν πληροφορίες για μεγαλύτερα benchmarks και υλοποιήσεις σε Hadoop, Spark.

2. Μελετήστε τους κώδικες Code/LoadBalancing και Code/SieveOfEratosthenes

Για να καταλάβετε τη λειτουργία του Κόσκινου του Ερατοσθένη δείτε το σύνδεσμο

https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

Γράψτε παράλληλο κώδικα για το πρόγραμμα SieveOfEratosthenes με τις εξής κατανομές φόρτου:

2.1. Στατική κατανομή: όπως έχουμε υλοποιήσει μέχρι τώρα τη κατανομή φόρτου.

2.2. Απλή κυκλική κατανομή: όπως στο κώδικα LoadBalancing/Cyclic.java

2.3. Δυναμική κατανομή: όπως στο κώδικα LoadBalancing/MasterWorkerLock.java

Συγκρίνετε τις επιδόσεις τους για όσο μπορείτε μεγάλο πλήθος πρώτων αριθμών, για παράδειγμα 1000000000.

Η Java έχει υλοποιήσει ειδικό πλαίσιο 'αυτόματης' δυναμικής κατανομής, τους *Executors*, με πολλές επιπλέον λειτουργίες

<https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html>

<https://www.geeksforgeeks.org/what-is-java-executor-framework/>

Οι πρώτοι αριθμοί και ο υπολογισμός μεγάλων πρώτων αριθμών είναι ιδιαίτερα χρήσιμοι στη κρυπτογραφία, στα κρυπτονομίσματα, το blockchain και αλλού. Παρακάτω δίνονται ορισμένα ενδεικτικά links για όσους ενδιαφέρονται.

https://en.wikipedia.org/wiki/Prime_number

https://en.wikipedia.org/wiki/Formula_for_primes

https://en.wikipedia.org/wiki/Prime_number_theorem

