

Άσκηση 2 (προθεσμία 8/1/2020)

Γράψτε δύο ζευγάρια από ισοδύναμα προγράμματα (δηλαδή τα δύο προγράμματα στο κάθε ζευγάρι πρέπει να κάνουν ακριβώς το ίδιο πράγμα, σε κάθε τους βήμα) στη C++ και στη Java για να επιδείξετε τα πλεονεκτήματα και μειονεκτήματα της αυτόματης ανακύκλωσης μνήμης (garbage collection) σε σχέση με τον πλήρη έλεγχο μνήμης από τον προγραμματιστή. Το πρόγραμμα A_{Java} πρέπει να είναι ακριβώς ισοδύναμο με το πρόγραμμα A_{C++} , με το πρώτο να είναι σημαντικά γρηγορότερο από το δεύτερο. Αντίστοιχα το πρόγραμμα B_{Java} πρέπει να είναι ακριβώς ισοδύναμο με το πρόγραμμα B_{C++} , αλλά πιο αργό.

Στη συνέχεια (αφού πετύχετε το παραπάνω με default ρυθμίσεις), πειραματιστείτε με τις ρυθμίσεις του GC για να δείτε αν και πώς αλλάζει η επίδοσή του στα προγράμματά σας. Διαλέξτε τουλάχιστον 2 από τις παρακάτω παραμέτρους/επιλογές:

- το σχετικό μέγεθος της παλιάς και νέας γενιάς
- το πότε ένα αντικείμενο πηγαίνει στην παλιά γενιά (tenuring)
- το αν ο GC είναι παράλληλος ή όχι, καθώς και το πόσα threads έχει
- το αν ο GC (παλιάς γενιάς) είναι mark-sweep ή compacting
- το αν ο GC (παλιάς γενιάς) είναι concurrent.

(Συμβουλευτείτε το επίσημο “Garbage Collection Tuning Guide” ή άλλο αντίστοιχο documentation. Χρειάζεστε πιθανότατα server VM και ίσως `-XX:+UnlockExperimentalVMOptions` για κάποιες επιλογές.)

Οδηγία 1: Ο ευκολότερος τρόπος να κάνετε έναν garbage collector αργό είναι να δημιουργήσετε ζωντανά δεδομένα συνολικού μεγέθους συγκρίσιμου με τη μνήμη RAM του μηχανήματος και μετά να αναγκάσετε επανειλημμένα να γίνει συλλογή των νεκρών δεδομένων. Αντίστοιχα, ο ευκολότερος τρόπος να κάνετε έναν μηχανισμό πλήρους ελέγχου μνήμης να είναι αργός είναι να δημιουργήσετε πολλά δεδομένα με κατάλληλη οργάνωση μεταξύ τους και να σκοτώσετε αρκετά ώστε τα εναπομείναντα να είναι διασπαρμένα χωρίς καθόλου καλή εντοπιότητα (locality).

Οδηγία 2: Η σύγκριση πρέπει να είναι όσο το δυνατόν πιο δίκαιη. Π.χ. η Java πιθανότατα δεσμεύει περισσότερο χώρο ανά αντικείμενο. Τα αντικείμενα της C++ δεν θα έχουν δείκτες σε v-table αν η κλάση τους δεν περιέχει virtual μεθόδους. Προσπαθήστε να κάνετε τα αντικείμενα όσο πιο ισοδύναμα γίνεται από απόψεως απαιτήσεων μνήμης. Η Java αργεί στο ξεκίνημα ενός προγράμματος καθώς και τις πρώτες φορές που εκτελείται μια μέθοδος (λόγω δυναμικής μεταγλώττισης). Ξεκινήστε τις μετρήσεις σας αφού το πρόγραμμα έχει δουλέψει για λίγο, ή διαλέξτε παραμέτρους έτσι ώστε το πρόγραμμα να δουλεύει για αρκετή ώρα οπότε το αρχικό κόστος δεν έχει μεγάλη σημασία. Χρησιμοποιείστε την παράμετρο `-Xmx<size>` για να ορίσετε το μέγιστο μέγεθος μνήμης στο JVM. Εξετάστε αν ο garbage collector τρέχει σε ξεχωριστό thread, παράλληλα με το κυρίως πρόγραμμα.

Παραδώστε κώδικα, σύντομη εξήγηση και μετρήσεις, μαζί με πλήρεις πληροφορίες για το υλικό και λογισμικό που χρησιμοποιήσατε.