



ΠΑΡΑΛΛΗΛΟΣ ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΟΣ ΥΠΟΛΟΓΙΣΜΟΣ

ΚΑΘΗΓΗΤΗΣ: ΜΑΡΓΑΡΙΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Αρχοντής-Εμμανουήλ Κωστής | ics21044

ΕΡΓΑΣΙΑ 6 (3)

Ανάλυση Χρόνου Εκτέλεσης NumInt

Για τη διεξαγωγή των πειραμάτων, εκτελέσαμε κάθε πρόγραμμα 10 φορές στο ίδιο σύστημα και κάτω από το όσο πιο δυνατό πανομοιότυπες συνθήκες. Στην συνέχεια καταγράφηκαν οι χρόνοι εκτέλεσης για κάθε υλοποίηση και υπολογίστηκαν οι αντίστοιχοι μέσοι χρόνοι (σε seconds).

Μέγεθος Προβλήματος: 100.000.000

Υλοποιήσεις:

1. Απλή Ακολουθιακή Εκτέλεση
2. Παράλληλη Εκτέλεση με LIMIT = 1.000
3. Παράλληλη Εκτέλεση με LIMIT = 50.000
4. Παράλληλη Εκτέλεση με LIMIT = 100.000
5. Παράλληλη Εκτέλεση με LIMIT = 250.000
6. Παράλληλη Εκτέλεση με LIMIT = 500.000
7. Παράλληλη Εκτέλεση με LIMIT = 1.000.000

A/A	1	2	3	4	5	6	7
1	0,316	Timeout (> 100 second)	0,490	0,218	0,117	0,085	0,079
2	0,315	Timeout (> 100 second)	0,539	0,203	0,154	0,097	0,120
3	0,313	Timeout (> 100 second)	0,432	0,204	0,143	0,127	0,085
4	0,314	Timeout (> 100 second)	0,445	0,211	0,131	0,093	0,086
5	0,327	Timeout (> 100 second)	0,425	0,213	0,133	0,132	0,096
6	0,314	Timeout (> 100 second)	0,438	0,228	0,134	0,151	0,110
7	0,316	Timeout (> 100 second)	0,434	0,214	0,133	0,187	0,086
8	0,324	Timeout (> 100 second)	0,419	0,210	0,145	0,103	0,072
9	0,317	Timeout (> 100 second)	0,531	0,213	0,130	0,140	0,125
10	0,317	Timeout (> 100 second)	0,429	0,208	0,124	0,147	0,160
M.O.	0,3173	-	0,458	0,212	0,134	0,126	0,102

Η Ακολουθιακή Υλοποίηση χρησιμεύει ως βάση για τη σύγκριση των επιδόσεων. Οι χρόνοι εκτέλεσης για αυτή την υλοποίηση κυμάνθηκαν σταθερά από 0,313 έως 0,327 δευτερόλεπτα και στις δέκα εκτελέσεις με μέσο χρόνο εκτέλεσης τα 0,32 δευτερόλεπτα.

Η παράλληλη εκτέλεση με $LIMIT=1.000$ παίρνει σε όλες τις εκτελέσεις πάνω από 100 δευτερόλεπτα, οπότε δεν αφήσαμε καμία εκτέλεση να ολοκληρωθεί (αφού η ακολουθιακή εκτέλεση ήταν κατα πολύ πιο γρήγορη). Αυτός ο μεγάλος χρόνος μπορεί υποδηλώνει ότι η επιβάρυνση (overhead) που σχετίζεται με τη δημιουργία και τη διαχείριση νημάτων ήταν δυσανάλογη με το μέγεθος των υπο-εργασιών που επεξεργάζονταν κάθε νήμα. Στο επόμενο πείραμα αυξήσαμε την τιμή του $LIMIT$ σε 50.000, κάτι που έδειξε ακόμη μικρότερους χρόνους εκτέλεσης (μεταξύ 0,419 και 0,539). Η μεταβλητότητα που υπάρχει ανάμεσα στους χρόνους εκτέλεσης θα μπορούσε να υποδηλώνει ότι ένα $LIMIT = 50.000$ δεν είναι το βέλτιστο καθώς και πάλι ο χρόνος εκτέλεσης της παράλληλης υλοποίησης είναι μεγαλύτερος του χρόνου της ακολουθιακής.

Η τιμή $LIMIT = 100.000$ οδήγησε σε μια πιο σταθερή απόδοση, με χρόνους εκτέλεσης που κυμαίνονταν από 0,203 έως 0,228 δευτερόλεπτα. Αυτή η τιμή έδειξε μια αξιοσημείωτη βελτίωση σε σχέση τόσο με την ακολουθιακή όσο και με την παράλληλη εκτέλεση $LIMIT = 50.000$. Αυτή η τιμή φαίνεται να επιτυγχάνει μια αποτελεσματικότερη ισορροπία μεταξύ της απόδοσης της παράλληλης εκτέλεσης και του κόστους της διαχείρισης πολλαπλών νημάτων, παρουσιάζοντας χαμηλότερο μέσο χρόνο εκτέλεσης. Στο επόμενο πείραμα αυξήσαμε την τιμή του $LIMIT$ στα 250.000. Η αύξηση αυτή έδειξε σημαντική βελτίωση στην απόδοση συγκριτικά με την ακολουθιακή υλοποίηση και τις άλλες υλοποιήσεις με χαμηλότερες τιμές $LIMIT$. Η μείωση αυτή του χρόνου εκτέλεσης μπορεί να υποδηλώνει ότι το $LIMIT = 250.000$ είναι και αυτό ιδιαίτερα αποτελεσματικό για το συγκεκριμένο μέγεθος προβλήματος, και επιτρέπει στην παράλληλη εκτέλεση να χειρίζεται πιο αποτελεσματικά τον φόρτο εργασίας, ελαχιστοποιώντας έτσι το overhead και αυξάνοντας τη συνολική απόδοση.

Συνεχίζοντας αυξήσαμε ξανά την τιμή του $LIMIT$ στα 500.000, με τους χρόνους εκτέλεσης για αυτή την τιμή να κυμαίνονται μεταξύ 0,085 και 0,187 δευτερόλεπτα. Η τιμή αυτή έδειξε χαμηλότερο μέσο χρόνο εκτέλεσης από ότι προηγούμενη η τιμή (250.000), αλλά καθώς το εύρος τιμών 0,085 - 0,187 παρουσιάζει μεγαλύτερη μεταβλητότητα αυτό θα μπορούσε να υποδηλώνει ότι ενώ οι υψηλότερες τιμές $LIMIT$ μπορεί να οδηγήσουν σε μεγαλύτερη απόδοση, μπορεί επίσης να εισάγουν extra overhead λόγω του αυξημένου μεγέθους αναδρομής και υπό-διεργασιών.

Στο τελευταίο πείραμα θέσαμε $LIMIT = 1.000.000$ και παρατηρήσαμε ότι οι χρόνοι εκτέλεσης κυμαίνονται μεταξύ 0,072 και 0,160. Η τιμή αυτή του $LIMIT$ είχε τους χαμηλότερους χρόνους εκτέλεσης στις περισσότερες εκτελέσεις (αν και είναι πολύ κοντά με τιμές του προηγούμενου πειράματος), καθώς και τον χαμηλότερο μέσο χρόνο εκτέλεσης από όλες τις υλοποιήσεις, αλλά όπως και με το προηγούμενο πείραμα οι χρόνοι εκτέλεσης παρουσίασαν μεγαλύτερη μεταβλητότητα, κάτι που υποδηλώνει ότι το extra overhead που εισάγεται από τη διαχείριση μεγαλύτερων υπό-εργασιών μπορεί να αντισταθμίσει τα οφέλη της παραλληλοποίησης σε ορισμένα προβλήματα-σενάρια.

Συμπεράσματα:

Η ανάλυση των επιδόσεων αποκαλύπτει ότι ενώ η παράλληλη εκτέλεση βελτιώνει σημαντικά τις επιδόσεις σε σύγκριση με την ακολουθιακή, η σωστή επιλογή του $LIMIT$ είναι κρίσιμη για την πραγματική βελτιστοποίηση των αποτελεσμάτων. Το ιδανικό $LIMIT$ για το δεδομένο μέγεθος προβλήματος (100.000.000) φαίνεται να είναι το 250.000, καθώς σύμφωνα με τα πειράματα αυτή η τιμή παρέχει μια καλή ισορροπία μεταξύ του overhead που εισάγεται και της επίτευξης χαμηλότερου χρόνου εκτέλεσης καθώς και παρουσιάζει χαμηλή μεταβλητότητα στους χρόνους εκτέλεσης.

Υψηλότερες τιμές $LIMIT$, όπως 500.000 και 1.000.000, προσφέρουν πρόσθετες βελτιώσεις στην απόδοση, αλλά εισάγουν ένα βαθμό μεταβλητότητας που μπορεί να μειώσει τη

συνολική απόδοση. Επομένως, ενώ αυτές οι υψηλότερες τιμές μπορεί να είναι επωφελείς, είναι λιγότερο συνεπείς στους χρόνους εκτέλεσης και μπορεί να μην παρέχουν πάντα καλύτερη απόδοση.

Στην πράξη, για την επιλογή ενός “καλού” LIMIT πρέπει να λάβουμε υπόψην τόσο το μέγεθος του προβλήματος όσο και το overhead που εισάγεται από τη διαχείριση των νημάτων. Για μεγάλα μεγέθη προβλήματος, ένα LIMIT των 250.000 προσφέρει μια ισχυρή και “συνεπή” βελτίωση σε σχέση με την ακολουθιακή εκτέλεση, καθιστώντας αυτή την τιμή μια καλή επιλογή για τη βελτιστοποίηση της απόδοσης. Ακολουθούν κάποια γραφήματα που συγκρίνουν τους χρόνους εκτέλεσης

