

Μοντέλα επικοινωνίας μεταξύ νημάτων και αρχέγονες δομές συγχρονισμού: Μια μελέτη περίπτωσης (Working title)

Νεσλεχανίδης Οδυσσέας 15537

June 6, 2019

Θα παρουσιαστούν κοινές αρχέγονες δομές συγχρονισμού (Semaphore, Spinlock, Barrier), και οι εφαρμογές τους στη λύση κλασικών προβλημάτων συγχρονισμού. Θα μελετηθούν πολυνηματικές υλοποιήσεις τους σε κώδικα C. Ακόμη, θα αναλυθούν οι παράμετροι που αφορούν την υλοποίηση μοντέλων επικοινωνίας μεταξύ νημάτων. Τέλος, θα γραφεί πρόγραμμα σε C με χρήση της βιβλιοθήκης pthreads, ως εφαρμογή των παραπάνω στο πρόβλημα του υπολογισμού της μεταφοράς θερμότητας.

Part I

Θεωρία παράλληλου προγραμματισμού συστημάτων

Εισαγωγή

Τα προγραμματιστικά εργαλεία που μας επιτρέπουν να εκμεταλλευόμαστε τις δυνατότητες παράλληλης επεξεργασίας των σύγχρονων πολυεπεξεργαστών και συστοιχιών (clusters) επεξεργαστών σχεδιάζονται βάσει ενός συστήματος μοντέλων παράλληλου προγραμματισμού. Ιστορικά, αυτά τα μοντέλα ανταποκρίνονταν πιο άμεσα στη δομή του παράλληλου υλικού. Σήμερα, αποτελούν κυρίως βολικές αφαιρέσεις για την κατηγοριοποίηση των αλγορίθμων που χρησιμοποιούνται στον παράλληλο προγραμματισμό και για τους όρους σύνθεσης και χρήσης τους εντός των προγραμμάτων.[1][2]

Την ίδια στιγμή, η επιλογή ενός μοντέλου έχει θεμελιώδη ρόλο στις αποφάσεις που λαμβάνονται κατά το σχεδιασμό εξειδικευμένων παράλληλων υπολογιστικών συστημάτων μεγάλης κλίμακας, στον κλάδο της υπολογιστικής υψηλών αποδόσεων (high performance computing).[3]

Τα μοντέλα παράλληλου προγραμματισμού χωρίζονται σε άξονες βάσει του τρόπου (εάν υφίσταται) που επικοινωνούν μεταξύ τους οι παράλληλες διεργασίες και βάσει του είδους προβλημάτων που επιδιώκεται να επιλυθούν με παράλληλη επεξεργασία.[1]

Η παρούσα εργασία κινείται στον πρώτο άξονα, παρουσιάζοντας τα βασικά μοντέλα επικοινωνίας μεταξύ διεργασιών και ειδικότερα μελετώντας τα κλασικά προγραμματιστικά εργαλεία για την επίλυση προβλημάτων συγχρονισμού που ενυπάρχουν στον προγραμματισμό κοινόχρηστης μνήμης, τις λεγόμενες αρχέγονες δομές συγχρονισμού (synchronization primitives).

Επικοινωνία μέσω κοινόχρηστης μνήμης (shared memory)

Στην παράλληλη ή ταυτόχρονη (concurrent) υπολογιστική, ένας πολύ φυσικός τρόπος για την επίτευξη επικοινωνίας μεταξύ διεργασιών, καθώς και για την αποφυγή σπατάλης χώρου από πολλαπλά όμοια αντίγραφα δεδομένων, είναι η χρήση ενός κοινού χώρου μνήμης από πολλές διεργασίες. Στο παρελθόν, ήταν διαδεδομένα τα υπολογιστικά συστήματα πολυεπεξεργαστών που διέθεταν αληθινά κοινόχρηστη μνήμη, με την οποία επικοινωνούσαν μέσω κοινού διαύλου. Σήμερα, η πραγματικότητα είναι διαφορετική.[2]

Με την αύξηση του πλήθους και της ταχύτητας των κεντρικών μονάδων επεξεργασίας (CPU) των πολυεπεξεργαστών, η ανάγκη διαχείρισης του προβλήματος συμφόρησης (bottleneck) von Neumann επέβαλε αλλαγή στο μοντέλο επικοινωνίας των κεντρικών μονάδων επεξεργασίας με τη μνήμη. Οι σημερινοί πολυεπεξεργαστές παρακάμπτουν σε ένα βαθμό το παραπάνω πρόβλημα έχοντας σχεδιαστεί σε πρότυπο μη-ομοιογενούς πρόσβασης μνήμης (Non-Uniform Memory Access: NUMA).[4]

Στις αρχιτεκτονικές αυτού του προτύπου, όπως μαρτυράει το όνομα του, χαρακτηριστική είναι η ανομοιογένεια στην ταχύτητα πρόσβασης (latency) ενός επεξεργαστή στα διάφορα στοιχεία μνήμης, ανάλογα με την απόσταση του στοιχείου μνήμης από τον εν λόγω επεξεργαστή.[5] Οι καθιερωμένες διεπαφές προγραμματισμού εφαρμογών (APIs) κοινόχρηστης μνήμης (pthread, OpenMP) είτε δεν παρέχουν εύχρηστα μέσα για τη διαχείριση, μέσω κώδικα, της παραπάνω ιδιότητας των σύγχρονων πολυεπεξεργαστών[6], είτε οι λύσεις που παρέχουν, παρεχόμενες ως επεκτάσεις, καταργούν τη φορητότητα του κώδικα[7]. Η διαχείριση της μνήμης συνήθως αφήνεται, κατά συνέπεια, στο λειτουργικό σύστημα.

Τα αναφερθέντα χαρακτηριστικά των εργαλείων προγραμματισμού που στηρίζονται στην αφαίρεση μιας εικονικά ενιαίας κοινόχρηστης μνήμης συντελούν στο αποτέλεσμα η χρήση τους να περιορίζεται σε υπολογιστικά συστήματα μικρής κλίμακας. Διαδεδομένη, ωστόσο, είναι η χρήση τους σε συνδυασμό με εργαλεία

προγραμματισμού κατανεμημένης μνήμης, για την υλοποίηση παράλληλων προγραμμάτων που εκτελούν επί μέρους εργασίες σε μικρά σύνολα επεξεργαστών, εντός υπολογιστικών συστημάτων μεγαλύτερης κλίμακας. Σε αυτού του είδους τις υβριδικές προσεγγίσεις, η επικοινωνία των επί μέρους εργασιών μεταξύ τους, αλλά και ευρύτερα με το σύστημα, υλοποιείται σε όρους προγραμματισμού κατανεμημένης μνήμης (message passing).[8]

Επικοινωνία μέσω μηνυμάτων (message passing)

Part II

Κλασικά προβλήματα συγχρονισμού

Λεπτομέρειες υλοποίησης

Εισαγωγή στη βιβλιοθήκη pthreads, παρουσίαση των μεθόδων που πρόκειται να χρησιμοποιηθούν.

1 Dining philosophers

Παρουσίαση του προβλήματος (εισαγωγή όρων deadlock, resource starvation), λύση με μέθοδο arbitrator για επίδειξη barriers. Επιλογή δυαδικού σηματοφορέα αντί mutex, για συνοχή.

2 Readers-writers

Παρουσίαση και λύση του προβλήματος, με προτίμηση στην απλούστερη εκδοχή όπου δίνεται αμοιβαίο αποκλεισμό με χρήση spinlock. Εάν αυτό αποδειχτεί δύσκολο, λύση της απλούστερης εκδοχής (producer-consumer equivalent) με χρήση monitor. Στην τελευταία περίπτωση, θα προηγηθεί κάλυψη της εν λόγω δομής στη θεωρία.

Part III

Μεταφορά θερμότητας σε δισδιάστατο χώρο: ένα πρόβλημα παράλληλου προγραμματισμού

Παρουσίαση του προβλήματος παράλληλου υπολογισμού της μεταφοράς θερμότητας σε δισδιάστατο χώρο. Μελέτη της παράλληλης λύσης με προγραμματισμό κατανεμημένης μνήμης (MPI), λύση με προγραμματισμό κοινόχρηστης μνήμης (pthreads).

Τι προκύπτει από τη σύγκριση των δυο λύσεων ως προς την απόδοση και την απλότητα υλοποίησης; Ποια γενικά κριτήρια για την επιλογή ανάμεσα στους δύο τύπους παράλληλου προγραμματισμού μπορούν να εξαχθούν από τη μελέτη των λύσεων του συγκεκριμένου προβλήματος;

Η εσωτερική δομή της ενότητας αυτής θα οριστεί αργότερα.

References

- [1] https://en.wikipedia.org/wiki/Parallel_programming_model
- [2] <https://stackoverflow.com/questions/36642382/main-difference-between-shared-memory-and-distributed-memory#answer-36659895>
- [3] www.shodor.org/media/content/petascale/materials/UPModules/beginnersGuideHPC/moduleDocumentation
- [4] https://en.wikipedia.org/wiki/Von_Neumann_architecture#Mitigations
- [5] https://en.wikipedia.org/wiki/Non-uniform_memory_access
- [6] <https://stackoverflow.com/questions/11959906/openmp-and-numa-relation#answer-11975593>
- [7] http://man7.org/linux/man-pages/man3/pthread_setaffinity_np.3.html
- [8] <https://software.intel.com/en-us/articles/hybrid-parallelism-parallel-distributed-memory-and-shared-memory-computing>