

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

**Κιοστεράκης Χαρίδημος
Μεταπτυχιακός Φοιτητής**

**Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης
Επόπτης Μεταπτ. Εργασίας: Καθηγητής, Μ. Κατεβαίνης
Ν. Καλλιμάνης (επιβλέπων)**

Πέμπτη, 27/02/2020, 10:00

Αίθουσα Β106, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

“Αποδοτικές Υλοποιήσεις Ατομικών Στιγμιότυπων Κοινόχρηστης Μνήμης”

ΠΕΡΙΛΗΨΗ

Την τελευταία δεκαετία έχει αυξηθεί αισθητά η χρήση των πολυπύρηνων συστημάτων, οι εφαρμογές των οποίων καλύπτουν ένα ευρύ φάσμα, όπως έξυπνα τηλέφωνα και υπολογιστές έως διακομιστές σε σύγχρονα πληροφοριακά κέντρα. Τα πολυπύρηνια συστήματα παρέχουν καλύτερες επιδόσεις μόνο υπό την προϋπόθεση ότι οι εφαρμογές τους μπορούν να χρησιμοποιήσουν τις δυνατότητες πολλαπλών υπολογιστικών πυρήνων. Ωστόσο αυτό δεν είναι τετριμμένο, μιας που ο παράλληλος προγραμματισμός είναι αρκετά δυσκολότερος από τον σειριακό. Η χρήση αποδοτικών κοινόχρηστων δομών δεδομένων ωστόσο, δίνει τη δυνατότητα πλήρους αξιοποίησης των πόρων ενός πολυπύρηνου συστήματος.

Μια διαδεδομένη κοινόχρηστη δομή δεδομένων είναι τα ατομικά στιγμιότυπα τα οποία και έχουν πληθώρα εφαρμογών στην επιστήμη των υπολογιστών. Τα ατομικά στιγμιότυπα μπορούν να χρησιμοποιηθούν για την επίλυση προβλημάτων, όπου συγκεκριμένες ενέργειες πρέπει να γίνουν όταν η κατάσταση του συστήματος ικανοποιεί κάποιες συνθήκες. Επιπρόσθετα, η χρήση ατομικών στιγμιότυπων είναι ιδιαίτερα διαδεδομένη για την σχεδίαση αλλά και τον έλεγχο ορθότητας κοινόχρηστων δομών δεδομένων, όπως στην κατασκευή concurrent timestamps ή randomized consensus.

Ένα ατομικό στιγμιότυπο αποτελείται από m συνιστώσες, οι οποίες αποθηκεύουν μια τιμή από ένα προκαθορισμένο σύνολο τιμών. Οι διεργασίες μπορούν να τροποποιούν ή/και να διαβάζουν τα αποθηκευμένα δεδομένα του αντικειμένου μέσα από την κλήση των διαδικασιών *UPDATE* και *SCAN*. Η *UPDATE* δίνει τη δυνατότητα στις διεργασίες να τροποποιήσουν την τιμή μιας συνιστώσας θέτοντας σε αυτή μια τιμή από το προκαθορισμένο σύνολο τιμών. Από την άλλη, η *SCAN* επιστρέφει ένα συνεπές αντίγραφο των τιμών όλων των συνιστωσών του στιγμιότυπου. Στην παρούσα εργασία μελετάμε διάφορες κατηγορίες ατομικών στιγμιότυπων. Πρώτα μελετάμε τα single-scanner στιγμιότυπα, σε αυτές τις υλοποιήσεις μόνο μια ενεργή *SCAN* επιτρέπεται να υπάρχει στο σύστημα οποιαδήποτε χρονική στιγμή, ταυτόχρονα το σύστημα όμως υποστηρίζει την συνύπαρξη πολλών *UPDATE*. Μια δεύτερη κατηγορία είναι τα multi-scanner στιγμιότυπα, όπου αυτά υποστηρίζουν ανά πάσα χρονική στιγμή την ύπαρξη στο σύστημα πολλών *UPDATE* αλλά και *SCAN*. Μια ιδιαίτερη περίπτωση, των multi-scanner είναι τα λ -scanner στιγμιότυπα, όπου αυτά υποστηρίζουν λ ενεργές *SCAN* ανά πάσα χρονική στιγμή και οσοδήποτε πολλές *UPDATE*. Όταν το λ είναι ίσο με το συνολικό πλήθος διεργασιών σε ένα σύστημα τότε ένα λ – scanner ατομικό στιγμιότυπο είναι ταυτόσημο με ένα multi-scanner.

Στην παρούσα εργασία, παρουσιάζονται δύο wait-free υλοποιήσεις ατομικών στιγμιότυπων. Η πρώτη αποτελεί υλοποίηση ενός single-scanner στιγμιότυπου ενώ η δεύτερη είναι υλοποίηση ενός λ -scanner στιγμιότυπου. Τόσο η χρονική τους πολυπλοκότητα όσο και η χωρική τους πολυπλοκότητα είναι γραμμική συνάρτηση του λ . Επιπρόσθετα, οι υλοποιήσεις μας έχουν σχετικά χαμηλή χωρική πολυπλοκότητα θυσιάζοντας μέρος της χρονικής πολυπλοκότητας. Η χαμηλή χωρική πολυπλοκότητα που παρέχουν οι υλοποιήσεις μας τις καθιστούν ελκυστικές σε εφαρμογές πραγματικών συστημάτων.

Αρχικά παρουσιάζεται μια απλή υλοποίηση ενός single-scanner στιγμιότυπου, την οποία ονομάζουμε $1 - OPT$. Αυτή η υλοποίηση χρησιμοποιεί $O(m)$ CAS καταχωρητές. Η *UPDATE* έχει χρονική πολυπλοκότητα $O(1)$, ενώ η *SCAN* $O(m)$.

Τέλος παρουσιάζεται μια υλοποίηση ενός λ -scanner στιγμιότυπου, την οποία ονομάζουμε $\lambda - OPT$. Αυτή η υλοποίηση χρησιμοποιεί $O(\lambda m)$ καταχωρητές μη πεπερασμένου μεγέθους. Η χρονική πολυπλοκότητα της *UPDATE* είναι $O(\lambda)$ ενώ εκείνη της *SCAN* είναι μόλις $O(\lambda m)$.

Kiosterakis Charidimos

M.Sc. Thesis

Computer Science Department

University of Crete

Master's Thesis Supervisor: Professor, M. Katevenis

N.Kalimanis (Thesis Co- Advisor)

Thursday, 27/02/2020, 10:00

Room B106, Computer Science Dept., University of Crete

“Efficient Implementations of Concurrent Snapshot Objects”

ABSTRACT

During the last decade, there was a rise on the use of multicore and many-core systems. Their application covers a huge range, from customer products like smartphones and laptops to servers that equip modern data centers. The multicore and many-core systems can provide better performance only if the user can use their potential, although this is not a trivial task since parallel programming is much harder than serial programming. In order to use to the fullest the power of such a system, utilization of efficient implementations of concurrent data structures is needed.

A snapshot object is a concurrent data structure that has numerous applications in concurrent programming. Snapshots can be used to record the state of the system, so they can provide solutions to problems where an action should be taken when the global state of the system satisfies some conditions. Furthermore, snapshots have been widely

used for the design and verification of distributed algorithms such as the construction of concurrent timestamps and randomized consensus.

A snapshot object is a concurrent data structure that consists of m components, each component storing a value from a given set. Processes can read/modify the state of the object by executing the operations *UPDATE* and *SCAN*. An *UPDATE* operation gives processes the ability to change the value of a component of the snapshot, while the *SCAN* operation returns a “consistent” view of all the components of the object. In this thesis, two variants of concurrent snapshot objects are studied. The first one is the single-scanner snapshot object which, can support only one active *SCAN* operation on any given time (whilst supporting many concurrent *UPDATE* operations). The second one is the multi-scanner snapshot object that can support multi concurrent *SCAN* operations at any given time. A variation of a multi-scanner snapshot object is a λ -scanner snapshot object, where the latter can support only λ different *SCAN* operations to be active at any given time.

In this work, two wait-free, linearizable implementations of snapshot objects are presented. The first one is an implementation of a single-scanner snapshot object, and the second one is an implementation of a λ -scanner snapshot object. The operations that are supported by those implementations have a step complexity that doesn't correlate with the number of active processes of the system n , while the space and step complexity of the λ -scanner object is a function of λ . Furthermore, our implementations have been designed to maintain relatively low space complexity even if that means sacrificing some step complexity. The low space complexity that our implementations provide makes them more appealing in real system applications.

We first provide a simple implementation of a single-scanner snapshot object, called $1 - OPT$. This implementation uses $O(m)$ *CAS* registers of unbounded size. The *UPDATE* has a step complexity of $O(1)$, while the *SCAN* has a step complexity of $O(m)$.

Then we present a λ -scanner snapshot object, called $\lambda - OPT$ that is based on $1 - OPT$. When the number of λ is equal to the number of processes in the system, $\lambda - OPT$ implements a multi-scanner object. This implementation uses $O(\lambda m)$ *CAS* registers of unbounded size. The *UPDATE* implementation has a step complexity of $O(\lambda)$, and the *SCAN* has a step complexity of $O(\lambda m)$. This implementation is also wait-free and provides a trade-off between the step/space complexity and the maximum number of *SCAN* operations that can afford to be active on any given point in time.