Επίδοση Υπολογιστικών Συστημάτων
Θέμα 2 - Group5

8ο Εξάμηνο, Ακαδημαϊκή περίοδος 2021 – 2022

Ονοματεπώνυμο	Αριθμός Μητρώου
Μπούφαλης Οδυσσεύς Δημήτριος	el18118
Αναστάσιος Παπαζαφειρόπουλος	el18079

Στην παρούσα εργασία θα ασχοληθούμε με την προσομοίωση υπολογιστικών συστημάτων, τις μεθόδους που χρησιμοποιούμε για να εξάγουμε τις εκτιμήσεις μας για τους δείκτες επίδοσης και την τελική σύγκριση 3 διαφορετικών πολιτικών χρονοδρομολόγησης. Το σύστημα με το οποίο θα ασχοληθούμε απαρτίζεται από δύο σταθμούς, την CPU και έναν δίσκο. Οι εργασίες που έχουμε είναι μιας κατηγορίας και οι αφίξεις τους γίνονται σύμφωνα με την κατανομή Poisson με ρυθμούς 0.9 εργασίες ανά sec. Οι εργασίες φτάνουν στην CPU η οποία έχει μια ουρά αναμονής. Η πολιτική για την ουρά της CPU είναι αυτή η οποία θα μελετηθεί. Στον δίσκο υπάρχει επίσης ουρά αναμονής που ακολουθεί πολιτική FIFO. Όπως είναι αναμενόμενο οι εργασίες της CPU ανά τακτά χρονικά διαστήματα μεταβαίνουν στον δίσκο και στην συνέχεια επιστρέφουν πάλι πίσω στην CPU για να συνεχιστεί η εξυπηρέτησή τους. Ο χρόνος εξυπηρέτησης ανά επίσκεψη είναι εκθετικά κατανεμημένος για όλους τους σταθμούς (CPU και δίσκο). Επιπλέον, μας δίνονται οι εξής πληροφορίες που προέρχονται από μετρήσεις στο σύστημα:

Μέσος χρόνος εξυπηρέτησης στην CPU	54
Μέσος χρόνος εξυπηρέτησης στον δίσκο	35
Μέσος αριθμός επισκέψεων στον δίσκο	18
Μέσος χρόνος μετάδοσης αποτελέσματος (εξερχόμενη σύνδεση)	234

Στόχος της προσομοίωσης είναι να συγκρίνουμε τις πολιτικές FIFO, LWTF και LRUF για την CPU με βάση τον μέσο χρόνο απόκρισης του συστήματος. Το διάστημα εμπιστοσύνης θα υπολογίζεται κάθε 20 αναγεννητικούς κύκλους και το πρόγραμμα θα τερματίζει όταν το διάστημα εμπιστοσύνης για τον μέσο χρόνο απόκρισης έχει μήκος μικρότερο από το 10% της μέσης τιμής ή όταν εκτελεστούν 1000 αναγεννητικοί κύκλοι. Για την δημιουργία των αφίξεων καθώς και των εκθετικά κατανεμημένων χρόνων εξυπηρέτησης φτιάξαμε μια συνάρτηση η οποία υλοποιεί την γεννήτρια για την κατανομή Poisson και την εκθετικής κατανομή (έχουν την ίδια γεννήτρια). Η γεννήτρια προκύπτει με την μέθοδο της αντιστροφής χρησιμοποιώντας μια ομοιόμορφα κατανεμημένη στο [0,1] τυχαία μεταβλητή U. Η γεννήτρια είναι η $-\frac{1}{\lambda} lnU$ όπου λ είναι οι αφίξεις ανά sec ή ισοδύναμα $-\mu lnU$ για την περίπτωση της εκθετικής κατανομής όπου μας δίνεται ως μ ο μέσος χρόνος εξυπηρέτησης για CPU και δίσκο. Για να καθορίσουμε την αλληλουχία των γεγονότων είναι απαραίτητο να διατηρούμε στο πρόγραμμά μας τον χρόνο που θα συμβεί η επόμενη άφιξη στο σύστημα καθώς και τους χρόνους ολοκλήρωσης της τρέχουσας επίσκεψης σε CPU και δίσκο αντίστοιχα. Επίσης, γνωρίζοντας ότι ο μέσος αριθμός επισκέψεων στον δίσκο είναι 18 μπορούμε να βρούμε την πιθανότητα μια εργασία να μεταβεί από την

CPU στο δίσκο ή να αποχωρήσει ολοκληρωτικά από το σύστημα. Συνεπώς, μια εργασία μεταβαίνει με πιθανότητα 18/19 στον δίσκο ενώ με πιθανότητα 1/19 αποχωρεί από το σύστημα (19 = μέσος αριθμός επισκέψεων στην CPU = 18 +1). Με βάση την ενότητα 7.3.4.3 του βιβλίου για την αναγγενητική μέθοδο, για την εκτίμηση ενός ζητούμενου δείκτη r μπορούμε να χρησιμοποιήσουμε την σχέση $R = \frac{\bar{y}}{\bar{c}}$ όπου $\bar{y}\ και\ \bar{c}$ είναι οι δειγματικές μέσες τιμές των $\{y_i\}, \{c_i\}$. Συνεπώς, για τον μέσο χρόνο απόκρισης θα διατηρούμε για κάθε κύκλο ως y_i το άθροισμα των συνολικών χρόνων απόκρισης, δηλαδή το άθροισμα των χρόνων που κάθε εργασία χρειάστηκε για να εξυπηρετηθεί σε αυτόν τον κύκλο και σε αυτό θα προσθέτουμε και τα 234 msec της εξερχόμενης σύνδεσης. Αντίστοιχα, το c_i θα είναι ο αριθμός των εργασιών που ολοκληρώθηκαν σε αυτόν τον κύκλο. Με την ίδια λογική για να εκτιμήσουμε τον βαθμό χρησιμοποίησης των πόρων του συστήματος θα ορίσουμε ως y_i τον χρόνο που κάθε πόρος ήταν μη-απασχολημένος για κάθε κύκλο και ως c_i την χρονική διάρκεια του κύκλο. Έτσι ο ζητούμενος βαθμός χρησιμοποίησης θα προκύπτει ως $1 - \frac{\bar{y}}{\bar{c}}$. Σχετικά με την συνθήκη τερματισμού εφόσον θέλουμε βαθμό εμπιστοσύνης 95% βρίσκουμε την τιμή z από τον πίνακα της κατανομής Student και η τιμή αυτή προκύπτει ίση με 1.96. Στη συνέχεια υπολογίζουμε το s^2 με βάση τις σχέσεις του βιβλίου. Για την ουρά αναμονής της CPU διακρίνουμε τρείς περιπτώσεις:

- FIFO: Εδώ χρησιμοποιήσαμε deque έτσι ώστε όταν έχουμε εισαγωγή στην ουρά αναμονής να κάνουμε απλά append στο τέλος της ενώ όταν αφαιρούμε κάτι από την ουρά επειδή έχουμε πολιτική FIFO να κάνουμε popleft.
- LWTF: Εδώ ορίσαμε 1 λεξικό το οποίο έχει ως μοναδικά keys τον αύξοντα αριθμό job_id της κάθε εργασίας. Το λεξικό είναι cpu_delay. Επίσης στην cpu_queue βάζουμε 3πλετες που περιέχουν το job_id καθώς και την πιο πρόσφατη στιγμή όπου η εργασία αυτή μπήκε σε αναμονή στην ουρά της CPU. Στο cpu_delay κρατάμε σαν value τον συνολικό αθροιστικό χρόνο που έχει βρεθεί σε αναμονή μια εργασία στην CPU. Όταν θέλουμε να κάνουμε pop από την ουρά διαλέγουμε την εργασία με τον μεγαλύτερο συνολικό χρόνο αναμονής. Τον χρόνο που ήταν σε αναμονή μια εργασία τον υπολογίζουμε εύκολα κοιτώντας το id της εργασίας που κάνουμε pop κάθε φορά από την ουρά και την χρονική στιγμή που αυτό συνέβη και αφαιρώντας στην συνέχεια την στιγμή που η εργασία εισήλθε στην ουρά. Τέλος, προσθέτουμε αυτήν την τιμή στην τρέχουσα τιμή του cpu_delay[job_id].
- LRUF: Εδώ ορίσαμε 2 λεξικά τα οποία έχουν ως μοναδικά keys τον αύξοντα αριθμό job_id της κάθε εργασίας. Τα λεξικά είναι τα cpu_last_time και cpu_delay. Στο cpu_last_time κρατάμε σαν value για το κάθε job_id την πιο πρόσφατη στιγμή όπου η εργασία αυτή μπήκε για εξυπηρέτηση στην CPU. Στο cpu_delay κρατάμε σαν value τον πιο πρόσφατο χρόνο εξυπηρέτησης για κάθε εργασία στην CPU και όταν θέλουμε να διαλέξουμε εργασία από την ουρά διαλέγουμε εκείνη με το μικρότερο cpu_delay. Γνωρίζοντας την στιγμή που μια εργασία μπήκε για εξυπηρέτηση στη cpu, όταν αυτή η εργασία φύγει από την cpu μπορούμε να αφαιρέσουμε τις 2 χρονικές αυτές στιγμές και να ανανεώσουμε την τιμή cpu_delay[job_id].

Σε περίπτωση ισοπαλιών, διαλέγουμε την εργασία που θα βγάλουμε από την ουρά με βάση τα arrival times των εργασιών. Για τον λόγο αυτό διατηρούμε ένα λεξικό arrival_times που για κάθε εργασία κρατάει την στιγμή που έφτασε στο σύστημά μας.

Εκτελέσαμε την προσομοίωση αρκετές φορές για να δούμε που συγκλίνουν τα αποτελέσματα για κάθε μέθοδο. Παρατηρούμε πως ο μέσος χρόνος απόκρισης είναι λίγο μικρότερος όταν χρησιμοποιούμε στην CPU την πολιτική LWTF και συγκεκριμένα κυμαίνεται ανάμεσα στα 13 και 15 sec. Ο βαθμός χρησιμοποίησης της CPU κυμαίνεται ανάμεσα στο 93 με 94% ενώ του δίσκου στα 55 με 56%. Ενδεικτικά, παρουσιάζουμε τα αποτελέσματα μιας προσομοίωσης με 1000 αναγεννητικούς κύκλους:

```
Regen cycles = 1000
Average response time  13748.357
cpu utilization 0.94
Disk utilization 0.564
```

Παρατηρούμε πως ο μέσος χρόνος απόκρισης για την πολιτική FIFO και για την LRUF είναι πολύ κοντά μεταξύ τους και συγκεκριμένα κυμαίνεται ανάμεσα στα 14 και 16.5 sec. Ίσως για την πολιτική LRUF μπορούμε να πούμε πως έχουμε λίγο καλύτερα αποτελέσματα. Ο βαθμός χρησιμοποίησης της CPU κυμαίνεται ανάμεσα στο 93 με 95% ενώ του δίσκου στα 55 με 56%. Ενδεικτικά, παρουσιάζουμε τα αποτελέσματα μιας προσομοίωσης με 1000 αναγεννητικούς κύκλους για LRUF και FIFO αντίστοιχα:

```
Regen cycles = 1000
Average response time  14021.61
cpu utilization 0.945
Disk utilization 0.566
```

```
Regen cycles = 1000
Average respone time  14929.19
cpu utilization 0.938
Disk utilization 0.56
```

Παρατηρούμε πως η χρησιμοποίηση των πόρων του συστήματος δεν παρουσιάζει κάποια ιδιαίτερη μεταβολή. Όσον αφορά τον μέσο χρόνο απόκρισης είναι λίγο βελτιωμένος όταν χρησιμοποιούμε την τεχνική LWTF. Διαισθητικά τα αποτελέσματα αυτά είναι ίσως αναμενόμενα αφού η μέθοδος LWTF θέτει σε προτεραιότητα εργασίες οι οποίες έχουν περιμένει για να πάρουν την CPU για μεγαλύτερο διάστημα ενώ η πολιτική FIFO δεν ακολουθεί κάποιο τέτοιο «έξυπνο» κριτήριο. Βέβαια, και οι 3 τεχνικές είναι πολύ κοντά όσον αφορά τον χρόνο απόκρισης κάτι που ίσως εξηγείται αν σκεφτούμε ότι δεν αλλάζουμε και την πολιτική εξυπηρέτησης στον δίσκο.