

Mobile Computing EELRU¹:

Ο αλγόριθμος που είχαμε να υλοποιήσουμε λέγεται Early Eviction LRU (EELRU) που αποτελεί μια adaptive επέκταση της LRU, συγκεκριμένα στην περίπτωση που η LRU κάνει fault σε packets που έδωσε πρόσφατα, η EELRU το αναγνωρίζει και αντί να διώξει το LRU packet κάνει "early eviction", δηλαδή διώχνει ένα packet που κάναμε πιο πρόσφατα reference από το LRU με την ελπίδα ότι κρατώντας τα πιο παλιά packets θα γίνουν reference ξανά σύντομα. Ο αλγόριθμος έχει δυο versions με τη δεύτερη να αποτελεί την γενίκευση της πρώτης, η οποία είναι:

```
if total · (M - e)/(l - e) ≤ early
    or (r(l) is in memory
    and the fault is on a less recently accessed page)
then evict the least recently accessed page
else evict page r(e)
```

Συγκεκριμένα το M είναι το μέγεθος της cache. e είναι το early eviction point και l είναι το late eviction point. total είναι τα references που έγιναν στην περιοχή [e, l] και early στην περιοχή [e, M]

Στην γενική version κρατάμε πολλά e και l και αποφασίζουμε τη θα διώξουμε με βάση το benefit.

```
let benefit be the maximum of the values
    totali · (M - ei)/(li - ei) - earlyi
and j be the index for which this value occurs
```

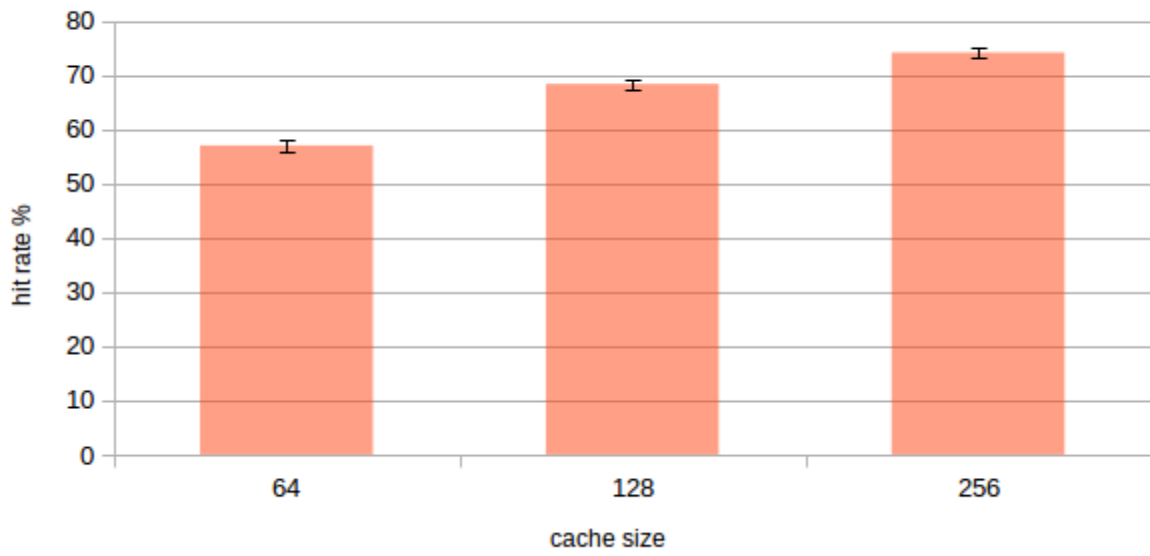
```
if benefit ≤ 0
    or a page r(n), n > lj is in memory
    or (r(lj) is in memory
    and the fault is on a less recently accessed page)
then evict the least recently accessed page
else evict page r(ej)
```

Τρέξαμε διάφορα πειράματα αλλάζοντας τον αριθμό packets [512, 768, 1024] και μεγέθη cache [64, 256, 512]. Επίσης αλλάξαμε τις παραμέτρους e και l του αλγορίθμου (στον πρώτο τα σημεία και στον γενικό τα regions και τα μεγέθη τους).

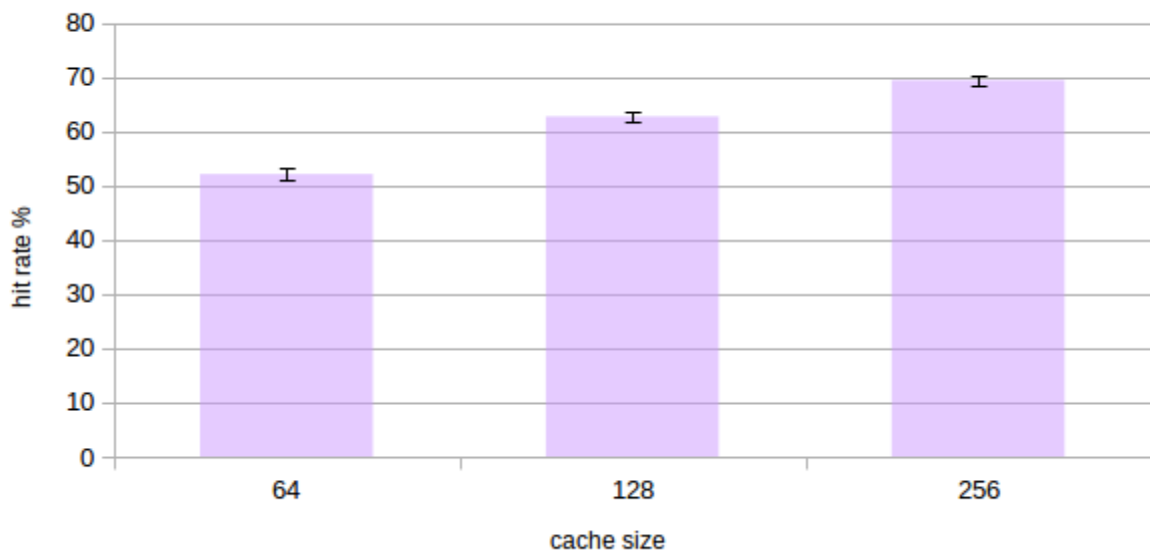
	512 packets	768 packets	1024 packets
64	56.98	52.08	47.94
128	68.32	62.72	58.16
256	74.14	69.4	66.8

Παράδοξα, παρατηρήσαμε ότι ανεξάρτητα του αλγορίθμου που χρησιμοποιήσαμε (LRU, EELRU, General EELRU) τα hit rates παρέμειναν τα ίδια (το οποίο επιβεβαιώνει και το paper παράγραφος 5.1.2 σελίδα 109) και πήραμε της ακολουθες παραστάσεις.

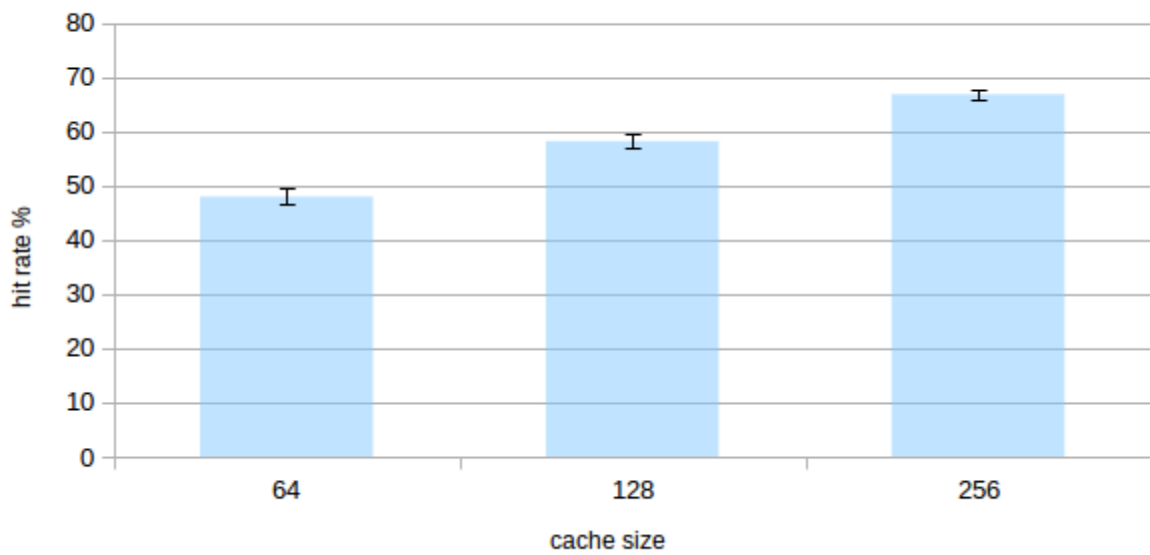
512 packets

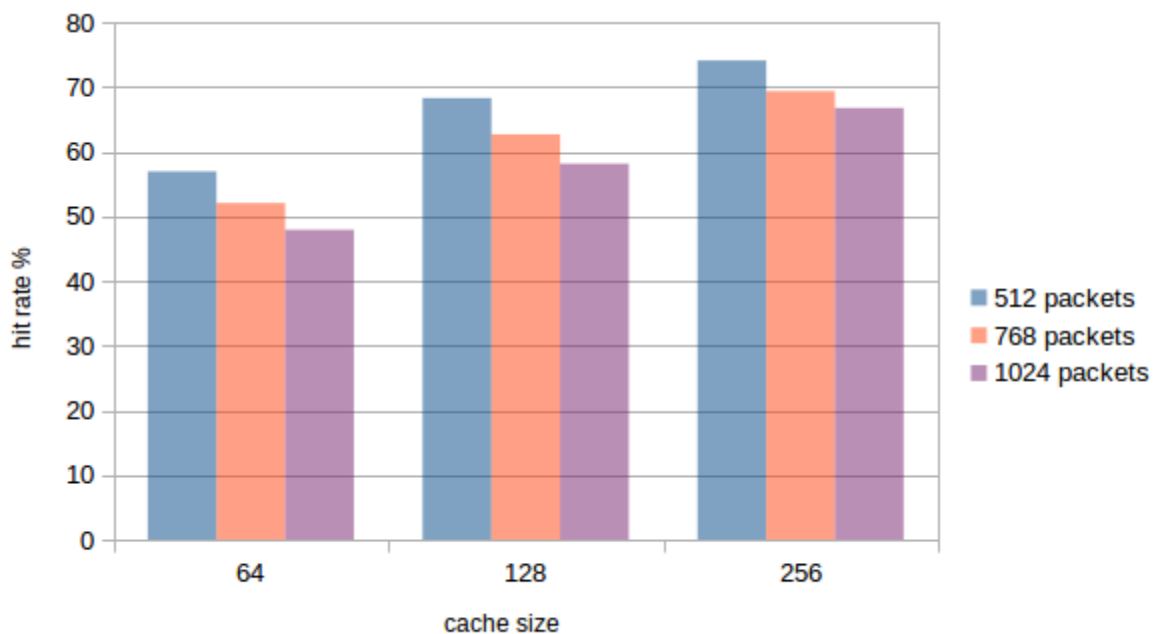


768 packets



1024 packets





Παρατηρούμε όσο αυξάνονται τα packets το hit rate πέφτει και όσο αυξάνεται η cache το hit rate αυξάνεται.

Επίσης στον φάκελο βρίσκεται το txt αρχείο με τα αποτελέσματα των πειραμάτων στην μορφή python dictionaries με το format

LRU

```
{packet number: {cache size: [mean, stdev], cache size: [mean, stdev], cache size: [mean, stdev]}, ...}}
```

EELRU

```
{ packet number {
    cache size : {(e, l): [mean, stdev], ...}...}
```

General EELRU

```
{ packet number {
    cache size : {(eMin, lMax, number of Regions): [mean, stdev], ...}...}
```

¹: The EELRU daptive replacement algorithm 2003