



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
Η/Υ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

1^Η ΑΣΚΗΣΗ

Επιτήρηση χρήσης πόρων εφαρμογών με Linux Cgroups

Ομάδα Α20:
Καραμπάση Αικατερίνη, 3112517
Μανδηλαράς Νικηφόρος, 3112012

ΑΘΗΝΑ, 2016

Στην εργασία αυτή μας ζητήθηκε να φτιάξουμε δύο αρχεία τα οποία θα τα καλούσε ο δαίμονας προκειμένου να διαχειριστεί τα cgroups του συστήματος.

Πρακτικά κάθε εφαρμογή χρειάζεται πόρους να τους αξιοποιεί όταν τους χρειαστεί αλλά και το data center θέλει να εκμεταλλεύεται όλη την υποδομή του. Από την άλλη δεν μπορεί να υπερφορτωθεί με εφαρμογές γιατί τότε δε θα μπορεί να ανταποκριθεί σωστά.

Μία εφαρμογή μπορεί να είναι είτε ελαστική είτε ανελαστική. Όταν αναφερόμαστε σε ελαστικές εφαρμογές αυτές είναι που όταν υπάρχουν παραπάνω διαθέσιμοι πόροι, τους οποίους αυτή μπορεί να εκμεταλλευτεί, τότε έτσι θα γίνει. Ο καταμερισμός των παραπανίσιων πόρων θα γίνει ποσοστιαία αναλογικά, σύμφωνα με τα χιλιοστά που έχει ζητήσει κάθε τέτοια εφαρμογή. Αντίθετα, για τις ανελαστικές, τα χιλιοστά τους θα παραμείνουν τα ίδια, είτε υπάρχουν παραπάνω διαθέσιμοι πόροι, είτε όχι, θα αξιοποιήσουν δηλαδή τα ελάχιστα χιλιοστά τους. Οι τύποι σύμφωνα με τους οποίους υπολογίζονται τα χιλιοστά της κάθε εφαρμογής διακρίνονται ως εξής:

$$\text{Ανελαστικές} = \frac{\text{cpu}}{\text{given_cpu_shares}} \cdot 2000$$

$$\text{Ελαστικές} = \frac{\text{cpu}}{\text{sum}} \cdot \frac{(\text{given_cpu_shares} - \text{sum2})}{\text{given_cpu_shares}} \cdot 2000, \text{ όπου το sum αναφέρεται στο}$$

σύνολο που είναι δεσμευμένο από τις ελαστικές και στο sum2 από τις ανελαστικές. Συνολικά sum+sum2 πρέπει να καταναλώνουν μικρότερο ή ίσο χώρο από το συνολικό που διαθέτει το data center.

Για να διαχειριστεί αυτούς τους διαχωρισμούς το σύστημα χρειάζεται εργαλεία, τα οποία θα διαβάζουν τι θέλει κάθε εφαρμογή να δεσμεύσει, αν υπάρχει αυτός ο χώρος κι αν υπάρχει να τον «κρατήσει» για τη δεδομένη εφαρμογή. Τα εργαλεία αυτά είναι το cgmon-policy και cgmon-limit. Ο φάκελος που λάβαμε και περιείχε το δαίμονα, περιείχε επίσης κι ένα directory demo – το οποίο ουσιαστικά δημιουργούνταν μετά από την εκτέλεση του ./cgmon_demo.sh - το οποίο αξιοποιεί τα προγράμματα που εμείς του δώσαμε και βλέπουμε ότι λαμβάνουμε τελικά σωστό αποτέλεσμα. Ωστόσο, για να εξασφαλίσουμε τη γενικότερη σωστή λειτουργία των προγραμμάτων μας θα τρέξουμε και χειροκίνητα εντολές και θα δούμε πώς το σύστημά μας ανταποκρίνεται σε αυτές.

cgmon_limit:

```
1.#!/mport fileinput
2.
3. import os
4.
5. for line in fileinput.input():
6.     line=line.rstrip('\n')
7.     b=line.split(':')
8.     filename = "/sys/fs/cgroup/cpu/monitor/"
9.     filename=filename+b[3]
10.    if b[0]=="create" :
11.        if os.path.exists(filename) :
12.            pass
13.        else:
14.            os.makedirs(filename)
15.    elif b[0]=="remove":
16.        os.removedirs(filename)
17.    elif b[0]=="add":
18.        filename=filename+'/tasks'
19.        a=int(b[4])
20.        os.system("echo %d >> %s" % (a,filename))
21.    elif b[0]=="set_limit":
22.        filename=filename+'/cpu.shares'
23.        a=int(b[5])
24.        os.system("echo %d > %s" % (a,filename))
25.    else:
26.        print("Wrong Input")
```

Η εργασία που αναλαμβάνει το `cgmon_limit` είναι να υλοποιήσει τις ρυθμίσεις που επέβαλε το `cgmon_policy`. Δηλαδή αν θέλουμε να δημιουργήσουμε μία νέα εφαρμογή ή αντίστοιχα να διαγράψουμε, ή ακόμα αν θέλουμε να δημιουργήσουμε ένα καινούριο `cgroup` τότε αυτό το script θα διαχειριστεί τα δεδομένα που θα του δώσουμε ως είσοδο στο `stdin` και θα τα επεξεργαστεί αναλόγως. Συγκεκριμένα μπορούμε να δούμε πως στη γραμμή 12 προβλέπουμε το ενδεχόμενο να υπάρχει ήδη η εφαρμογή που ζητάμε να δημιουργήσουμε κι επομένως δεν κάνουμε καμία ενέργεια. Σε διαφορετική περίπτωση (γρ. 14) δημιουργούμε το αντίστοιχο `directory`. Όταν θελήσουμε να διαγράψουμε κάποια εφαρμογή εργαζόμαστε αναλόγως. Όταν ζητήσουμε να προσθέσουμε μία εφαρμογή στο υπάρχων `cgroup` τότε πρέπει να δημιουργήσουμε για αυτήν ένα `directory` ονόματι `tasks` και στη συνέχεια θα μας ενημερώσει σχετικά με τη δημιουργία της καινούριας εφαρμογής.

Αντίστοιχα θα πράξει όταν θα θέλει να μεταβάλει το cpu limit της εφαρμογής. Σε κάθε άλλη περίπτωση τυπώνουμε μήνυμα λανθασμένης εισόδου.

cgmon_policy:

```
1. #!/usr/bin/python
2.
3. import fileinput
4. suma=0
5. limit=2000
6. lol=[]
7. for line in fileinput.input():
8.     b=line.split(':')
9.     suma+=int(b[3])
10.    lol.append(["set_limit",b[1],"cpu.shares",b[3]])
11.    print("score:%f" % float(limit-suma))
12. for x in lol:
13.     a=int(x[3])
14.     x[3]=str(a*limit/suma)
15.     print(":".join(x))
```

Η εργασία που ανατίθεται στο cgmon_policy είναι να του δώσουμε ως είσοδο το όνομα της διεργασίας που θέλουμε να δημιουργήσουμε αλλά και το μέγεθος της cpu που θέλουμε να δεσμεύσουμε (γρ. 10). Ως έξοδο το πρόγραμμα αυτό θα μας εμφανίζει στην πρώτη γραμμή έναν αριθμό κινητής υποδιαστολής ο οποίος αν είναι θετικός τότε έχουμε εξασφαλίσει τα ελάχιστα χιλιοστά που απαιτήσε κάθε εφαρμογή ενώ αν είναι αρνητικός τότε κάποια εφαρμογή ζήτησε παραπάνω από όσα είχαμε διαθέσιμα να δώσουμε και επομένως απορρίφθηκε από την είσοδό της στη χρήση χιλιοστών του data center. Οι επόμενες γραμμές θα μας εμφανίσουν τα χιλιοστά που εξασφαλίσαμε σε κάθε εφαρμογή.

Τα προγράμματα αυτά θα καλούνται κάθε φορά μέσω του δαίμονα, και μέσω αυτού θα εμφανίζονται και τα αποτελέσματα που επιστρέφονται.

Τα αρχεία μας ονοματίζονται όπως μας ζητείται από το δαίμονα προκειμένου να τα εντοπίσει και να τα εκτελέσει. Εναλλακτικά μπορούμε να τα ονομάσουμε κατά πώς επιθυμούμε και φυσικά να κάνουμε τις κατάλληλες αλλαγές στο δαίμονα. Επίσης τα αρχεία αυτά χρειάζεται να τοποθετηθούν μέσα στο φάκελο demo που θα δημιουργηθεί με την εκτέλεση του ./cgmon_demo.sh. Σε περίπτωση που

επιθυμούμε να τα αποθηκεύσουμε σε κάποιο διαφορετικό working directory θα πρέπει να δώσουμε τα αντίστοιχα ορίσματα στο δαίμονα.

Όπως αναλύσαμε και πιο πάνω, τα ελάχιστα χιλιοστά που θα ζητήσει κάθε εφαρμογή θα εξασφαλιστούν σε περίπτωση που αυτά υπάρχουν ($\text{score} > 0.00$) ενώ σε αντίθετη περίπτωση θα υπάρχει κατάλληλο μήνυμα. Αρνητική τιμή στο score δίνεται μόνο εφ' όσον έχει «ερωτηθεί» το σύστημα εάν έχει τη δυνατότητα να μας παρέχει τα αιτούμενα χιλιοστά και στη συνέχεια δίδεται η αντίστοιχη τιμή στη μεταβλητή αυτή.

Τέλος, στην περίπτωση που θα έχουμε ελαστικές και ανελαστικές εφαρμογές, θα χειριστούμε σύμφωνα με τα κλάσματα που αναφέραμε παραπάνω, τους πόρους μας ώστε να μη μένουν αδρανείς, σε κάθε στιγμή. Τα παραπάνω χιλιοστά από αυτά που έχει ζητήσει η κάθε εφαρμογή μοιράζονται στις ελαστικές εφαρμογές ποσοστιαία αναλογικά με το ποσό που έχουν απαιτήσει κατά τη διαδικασία της δημιουργίας τους.