**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ KΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**



Εργαστήριο Υπολογιστικών Συστημάτων

**Άσκηση 1:** *Επιτήρηση χρήσης πόρων εφαρμογών με Linux Cgroups*

**Ημερομηνία:** *30/03/2016*

**Ομάδα cslabb06**

***Πετρόπουλος-Τράκας Ευθύμης (031 11525)***

***Ροδίτης-Κουτσαντώνης Ορέστης (031 11052)***

**1η Άσκηση**

Ένα κέντρο δεδομένων (data center) προσφέρει υπηρεσίες φιλοξενίας εφαρμογών στην αγορά. Από τη μία, το κέντρο δεδομένων επιθυμεί να εκμεταλλεύεται ολόκληρη την υποδομή του χωρίς αδρανείς πόρους. Από την άλλη, οι πελάτες του απαιτούν οι πόροι που έχουν παραγγείλει να είναι διαθέσιμοι με αξιοπιστία. Εάν υπερφορτωθεί με εφαρμογές το κέντρο δεδομένων τότε δεν θα μπορεί να ανταποκριθεί στις απαιτήσεις όλων των εφαρμογών.

Για την άσκηση αυτή υποθέτουμε ότι η απαίτηση μιας εφαρμογής μετράται σε χιλιοστά της υπολογιστικής ισχύος του ενός επεξεργαστή. Ένας κόμβος με 16 επεξεργαστές μπορεί να καλύψει σύνολο 16000 χιλιοστών και άρα μπορεί να εξυπηρετήσει, για παράδειγμα, 8 εφαρμογές που απαιτούν εγγύηση 2000 χιλιοστών η κάθε μία.

Σκοπός της άσκησης είναι η δημιουργία προγραμμάτων που με τη χρήση του συστήματος cgroup μπορεί να εξασφαλίσει την υπολογιστική ισχύ, ρυθμισμένη σε χιλιοστά του ενός επεξεργαστή όπως παραπάνω.

Ως βοήθημα δίδεται ο cgmond, πρόγραμμα (δαίμων με την έννοια του unix) που αναλαμβάνει να εκκινεί και να παρακολουθεί εφαρμογές, καθώς και να επιβάλλει τα ελάχιστα εξασφαλισμένα χιλιοστά επεξεργαστικής ισχύος όπως δίνονται στην εκκίνηση της εφαρμογής.

Ο cgmond δεν ερμηνεύει τις τιμές για τα ελάχιστα εξασφαλισμένα χιλιοστά κάθε εφαρμογής, ούτε εφαρμόζει ρυθμίσεις cgroup. Αντί αυτού καλεί δύο εξωτερικά εκτελέσιμα προγράμματα, cgmon-policy και cgmon-limit, αντίστοιχα. Κάθε φορά που δημιουργείται ή τερματίζεται μια εφαρμογή, ο cgmond εκτελεί το cgmon-policy δίνοντάς του όλες τις εφαρμογές με τις ρυθμίσεις για τα χιλιοστά τους. Το cgmon-policy παράγει τις ρυθμίσεις για το σύστημα cgroup που πρέπει να εφαρμοστούν. Στη συνέχεια, το cgmon-limit καλείται για να εφαρμόσει τις ρυθμίσεις που παρήγαγε το cgmon-policy.

Σε περίπτωση που ζητηθεί η εκκίνηση μιας εφαρμογής με χιλιοστά περισσότερα από όσα μπορεί να εγγυηθεί το σύστημα τότε προειδοποιεί και απαιτεί επιβεβαίωση.

**Ζητούμενα της Άσκησης**

Ζητείται η συγγραφή κατάλληλων προγραμμάτων cgmon-policy και gmon-limit και η ανεύρεση κατάλληλων εφαρμογών ώστε να ικανοποιηθούν και να επιδειχθούν τα παρακάτω:

1. Να συμφωνούν με τις προδιαγραφές που δίδονται στις αντίστοιχες ενότητες ώστε να μπορούν να χρησιμοποιηθούν από τον cgmond.
2. Θα πρέπει μετά την εφαρμογή από το cgmon-limit των ρυθμίσεων cgroup που παράγει το cgmon-policy να εξασφαλίζονται τα ελάχιστα χιλιοστά χρήσης κάθε εφαρμογής όπως έχουν δηλωθεί κατά την εκκίνησή της.
3. Το cgmon-policy θα πρέπει να βαθμοδοτεί κατάλληλα την είσοδό του ώστε ο cgmond να αποτρέπει την εκκίνηση εφαρμογής εάν πρόκειται να κινδυνεύσουν τα εγγυημένα χιλιοστά οποιασδήποτε εφαρμογής.
4. Εάν υπάρχουν ελαστικές και ανελαστικές εφαρμογές υπό επιτήρηση να εξασφαλίζεται ότι όλοι οι επεξεργαστές του κόμβου χρησιμοποιούνται και κανείς δεν μένει αδρανής.

**Άσκηση 1: Υλοποίηση:**

Ακολουθεί επεξήγηση της λειτουργίας του κάθε εκτελέσιμου από τα ζητούμενα τα οποία υλοποιήσαμε.

1. **cgmon-policy.py:**

Συνοπτικά, το εκτελέσιμο cgmon-policy καλείται να μετατρέψει το μέτρο των ελάχιστων εξασφαλισμένων χιλιοστών χρήσης ενός επεξεργαστή, μέγεθος με το οποίο συναλλάσσεται ο χρήστης, σε cpu.shares, μέγεθος με το οποίο ρυθμίζεται το σύστημα cgroup για το διαμοιρασμό επεξεργαστικής ισχύος σε εφαρμογές. Ως έξοδος επιστρέφεται από αυτό το εκτελέσιμο μια τιμή της μορφής **score:<float>,** η οποία εάν είναι θετική υποδηλώνει ότι τα ελάχιστα όρια χρήσης είναι εξασφαλισμένα για όλες τις εφαρμογές, ενώ αν είναι αρνητική τότε τα όρια δεν μπορούν να τηρηθούν. Ακολουθεί επίσης έξοδος, που αντιστοιχεί σε κάθε application που δόθηκε ως είσοδος, τα cpu-shares που του αντιστοιχούν.

Ο κώδικας του *cgmon-policy.py* δίδεται παρακάτω. Ως “αλγόριθμο” για την εξεύρεση του ζητούμενου score αποφασίσαμε απλώς να αφαιρούμε τα ζητούμενα από κάθε εφαρμογή χιλιοστά, από τα συνολικά 2000 που έχουμε διαθέσιμα. Αν το εν λόγω άθροισμα ξεπερνά το όριό μας το score αποκτά όπως είναι προφανές αρνητική τιμή, οπότε ο cgmon ενημερώνεται ότι τα εγγυημένα χιλιοστά των εφαρμογών δεν μπορού να τηρηθούν. Στην ειδική περίπτωση που το score είναι μηδέν, όταν δηλαδή τα ζητούμενα χιλιοστά είναι ακριβώς όσα υπάρχουν διαθέσιμα, τότε δίνουμε στο score θετική τιμή, αλλά αρκετά μικρή ώστε να μην υπάρχουν άλλα χιλιοστά προς κατανάλση, πράγμα που κάνει την εφαρμογή να εκτελεστεί κανονικά με τα χιλιοστά που έχει ζητήσει.

1. **gmon-limit.py**

Συνοπτικά, το εκτελέσιμο cgmon-limit αναλαμβάνει να εφαρμόσει τις ρυθμίσεις που παρήγαγε το cgmon-policy, καθώς και να συντηρεί την απεικόνιση των εφαρμογών σε ιεραρχίες cgroup κατ'εντολή του cgmond.

Το εκτελέσιμο δέχεται στην είσοδο γραμμές ενός εκ των ακόλουθων τύπων:

Δημιουργία cgroup για μια νέα εφαρμογή:

***create:<monitor>:cpu:<application name>***

Κατάργηση του cgroup μιας εφαρμογής που έχει τερματίσει

***remove:<monitor>:cpu:<application name>***

Εγγραφή μιας διεργασίας στο cgroup μιας εφαρμογής:

***add:<monitor>:cpu:<application name>:<process id>***

Ρύθμιση της τιμής cpu.shares για το cgroup μιας εφαρμογής:

***set\_limit:<monitor>:cpu:<application name>:cpu.shares:<value>***

Το αναγνωριστικό monitor αντιστοιχεί στο cgroup του cgmond η οποία περιέχει όλα τα cgroups των εφαρμογών. Με αυτόν τον τρόπο απομονώνεται ο χώρος ονομάτων των εφαρμογών από το υπόλοιπο σύστημα για την αποφυγή συγκρούσεων (π.χ. κάποιος προσπαθεί να ονομάσει μια εφαρμογή με όνομα cgroup που υπάρχει ήδη στο σύστημα).

Το directory στο οποίο δουλεύαμε ώστε να χειριστούμε τους σχετικούς πόρους ήταν το ***/sys/fs/cgroup/cpu/***. Όταν μας ζητούνταν να δημιουργήσουμε ένα νέο cgroup, δημιουργούσαμε μια διεύθυνση με το αντίστοιχο όνομα στο εν λόγω directory. Όταν μας ζητούνταν να διαγράψουμε ένα cgroup, διαγράφαμε τη διεύθυνση που αντιστοιχούσε σε αυτό. Για την εγγραφή μιας νέας εφαρμογής σε ένα υπάρχον cgroup, βρίσκαμε και προσθέταμε το pid της εν λόγω διεργασίας στο αρχείο task, που βρισκόταν μέσα στη διεύθυνση που αντιστοιχούσε στο cgroup. Και όταν έπρεπε να αλλάξουμε τα *cpu.shares* που αντιστοιχούσαν σε μια εφαρμογή, αλλάζαμε την τιμή που περιείχε το αρχείο με το όνομα *cpu.shares*, το οποίο επίσης βρισκόταν στη διεύθυνση του αντίστοιχου cgroup. Αυτό ήταν ορθό, επειδή στη γενική περίπτωση κάθε application που θέλαμε να έχει κάποια εξασφαλισμένα χιλιοστά, περιεχόταν αυτό και μόνο σε ένα cgroup προορισμένο αποκλειστικά για αυτό. Υπήρχε μία μόνο περίπτωση, όπου διαφορετικά applications περιέχονταν στο ίδιο cgroup, και αυτό αφορούσε τα applications των οποίων τα shares δε θεωρούνταν ανελαστικά. Κρατούσαμε λοιπόν ένα μόνο cgroup για τις ελαστικές εφαρμογές, και το share του συγκεκριμένου cgroup, και άρα όλων των εφαρμογών που περιέχονταν σε αυτό, συλλογικά, άλλαζε ανάλογα με τα shares που περίσσευαν από τα υπόλοιπα cgroup.

Αν και δεν απαιτήθηκε να υλοποιηθεί στην άσκηση, να αναφέρουμε ότι στην περίπτωση που θα θέλαμε να διαγράψουμε ένα application από το cgroup στο οποίο ανήκε, θα έπρεπε να εγγράψουμε το pid του στο αρχείο *tasks* του cgroup πατέρα του τρέχοντος cgroup, ένα επίπεδο “πάνω” δηλαδή. Αυτόματα θα διαγραφόταν από τα tasks του cgroup παιδιού.

1. **cgmon\_ex1.sh**

Αυτό είναι το αρχείο που χρησιμοποιήσαμε για να πειραματιστούμε με διάφορα σενάρια διεργασιών και να επιβεβαιώσουμε ότι το σύστημα μας δουλεύει επαρκώς. Αυτό που κάναμε, είναι χρησιμοποιήσαμε διάφορες διαμορφώσεις με διαφορετικές διεργασίες να τρέζουν στο monitor, τόσο σε πλήθος, όσο και ανάγκες επεργαστικής ισχύος, ώστε να δούμε ότι το σύστημα μας έχει σωστή συπεριφορά σε περιπτώσεις που οι απαιτήσεις ήταν μικρότερες, ίσες ή και μεγαλύτερες από το όριο των χιλιοστών, τόσο όταν όλες οι διεργασίες ήταν ανελαστικές (μεγάλος χρόνος χρήσης υπολογιστικής μονάδας), όταν όλες οι διεργασίες ήταν ελαστικές (μικρός χρόνος χρήσης υπολογιστικής μονάδας - μεγάλος χρόνος αναμονής IO), όσο και σε μίξη αυτών των συνδιασμών.

Ένα δείγμα ενός από τα scripts που χρησιμοποιήσαμε για να τρέξουμε την υλοποίηση μας, μπορείτε να βρείτε παρακάτω.

**ΑΡΧΕΙΑ ΓΙΑ ΤΗΝ ΑΣΚΗΣΗ**

**Κώδικας του αρχείου *gmon-limit.py*:**

*#!/usr/bin/python*

*import sys*

*import subprocess*

*from os import path*

*lines = sys.stdin.readlines()*

*mainPath = '/sys/fs/cgroup/cpu/'*

*for line in lines:*

*line = line.strip('\n')*

*sLine = None*

*if line.startswith('create') or line.startswith('remove'):*

*sLine = line.split(':')[1::2]*

*elif line.startswith('add'):*

*sLine = line.split(':')*

*sLine = [ sLine[1], sLine[3], sLine[4] ]*

*else:*

*sLine = line.split(':')[1::2]*

*if line.startswith('create'):*

*if not path.exists(mainPath):*

*subprocess.call([ 'mkdir', '-p', mainPath + '/'.join(sLine) ])*

*elif line.startswith('remove'):*

*if path.exists(mainPath + '/'.join(sLine)):*

*subprocess.call([ 'rmdir', mainPath + '/'.join(sLine) ])*

*elif line.startswith('add'):*

*if not path.exists(mainPath + '/'.join(sLine[:2])):*

*subprocess.call([ 'mkdir', '-p', mainPath + '/'.join(sLine[:2]) ])*

*with open(mainPath + '/'.join(sLine[:2]) + '/tasks', 'a+') as f:*

*chkPid = str(sLine[-1]) in f.read().split('\n')*

*if not chkPid:*

*f.write(sLine[-1] + '\n')*

*elif line.startswith('set\_limit'):*

*if not path.exists(mainPath + '/'.join(sLine[:2])):*

*subprocess.call([ 'mkdir', '-p', mainPath + '/'.join(sLine[:2]) ])*

*subprocess.call([ 'echo', sLine[-1] ], stdout=open(mainPath + '/'.join(sLine[:2]) + '/cpu.shares', 'w'))*

**Κώδικας του αρχείου *cgmon-policy.py*:**

*#!/usr/bin/python*

*import sys*

*import subprocess*

*# policy:<application name>:cpu:<value>*

*lines = sys.stdin.readlines()*

*policies = {}*

*for line in lines:*

*policy = line.split(':')*

*policies[policy[1]] = int(policy[-1])*

*score = 2000 - sum(policies.values())*

*if score == 0:*

*score = 0.1*

*# set\_limit:<application name>:cpu.shares:<value>*

*sys.stdout.writelines('score:' + str(score) + '\n')*

*for policy in policies:*

*sys.stdout.writelines("set\_limit:" + policy + ":cpu.shares:" +*

*str(policies[policy]) + '\n')*

**Αρχείο για την εκτέλεση προκειμένου να ελεγχεί ο κώδικας *cgmon\_ex1.sh*:**

*#!/bin/bash*

*set -v*

*WD=${PWD}/ex1*

*cgmon daemon stop -w ${WD}*

*sleep 1*

*cgmon daemon start -w ${WD} -p ${WD}/cgmon-policy.py -l ${WD}/gmon-limit.py*

*sleep 1*

*cgmon app list*

*sleep 1*

*cgmon policy create -n anelastic1 -p 100*

*cgmon policy create -n anelastic2 -p 150*

*cgmon policy create -n anelastic3 -p 200*

*cgmon policy create -n elastic -p 50*

*cgmon policy list*

*sleep 1*

*cgmon app spawn -p anelastic1 -e "stress -c 2" -n ANELASTIC1*

*cgmon app spawn -p anelastic2 -e "stress -c 2" -n ANELASTIC2*

*cgmon app spawn -p anelastic3 -e "stress -c 2" -n ANELASTIC3*

*cgmon app spawn -p elastic -e "stress -c 2" -n ELASTIC1*

*cgmon app spawn -p elastic -e "stress -c 2" -n ELASTIC2*

*cgmon app list*

*sleep 15*

*pkill -f stress*

**Σημαντική σημείωση:** *Όλα τα παραπάνω αρχεία μπορείτε να τα βρείτε, σε περίπτωση ανακρίβειας ή άλλης αμφιβολίας, εαν έρθετε σε επικοινωνία μαζί μας:*

* ***timos@email.com***
* ***orestarod@gmail.com***