



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ

## *Εργαστήριο Λειτουργικών Συστημάτων*

*Αναφορά για την 1<sup>η</sup> Άσκηση:*

### *Επιτήρηση χρήσης πόρων εφαρμογών με Linux Cgroups*

Ον/μο :Τελάλη Ειρήνη  
Α.Μ.: :03113009

Ον/μο :Μπερέτσος Θεόδωρος  
Α.Μ.: :03111612

Ομάδα: b27  
Ημερομηνία εξέτασης: 16/3/2017  
Ημερομηνία παράδοσης: 22/3/2017

## 1 Σκοπός

Αντικείμενο αυτής της άσκησης είναι η χρησιμοποίηση των linux cgroups ώστε να καθορίζουμε με τον τρόπο που θέλουμε τα resources (cpu, memory, network bandwidth κλπ) ώστε να χρησιμοποιηθούν κατάλληλα από κάποια ομάδα διεργασιών. Για παράδειγμα μπορεί να θέλουμε να εξασφαλίσουμε ότι μια συγκεκριμένη ομάδα διεργασιών θα έχει πάντα x μονάδες (χιλιοστά) από τον χρόνο της CPU.

## 2 Πηγαίος κώδικας (source code)

### mod\_policy\_cpumin.py

```
#!/usr/bin/env python

import sys
import os

# reading input line by line
# splitting and storing applications' names and cpu usage
# computing and printing cpu's workload
# printing applications' names and cpu usage
def policy():
    Available = 2000
    Percentage = 1
    app_name_l = []
    for line in sys.stdin:
        name = line.rstrip()
        app_name = name.split(":")[1]
        value = int(name.split(":")[3])
        Available -= value
        Percentage = Available/2000.0
        app_name_l.append((app_name,value))
    print ('score:'+str(Percentage))
    for (app_name, value) in app_name_l:
        print ('set_limit:'+app_name+':cpu.shares:'+str(value))

if __name__ == '__main__':
    policy()
```

### mod\_limit\_cpu.py

```
#!/usr/bin/env python

import sys
import os

# fun that creates the path for a cgroup of an application
# if the path already exists, the fun returns does nothing
def create_proc(proc_command):
    monitor = proc_command[1]
    task = proc_command[3]
    path = "/sys/fs/cgroup/cpu/"+monitor+"/"+task
    if os.path.exists(path):
        return
    else:
        os.makedirs(path)
```

```

# fun to delete the cgroup of an application
# if the path does not exist, the fun returns does nothing
def remove_proc(proc_command):
    monitor = proc_command[1]
    task = proc_command[3]
    path = "/sys/fs/cgroup/cpu/"+monitor+"/"+task
    if not os.path.exists(path):
        return
    else:
        os.removedirs(path)

# fun that adjusts the cpu.shares value for the cgroup of an application
# if the path does not exist, the fun returns does nothing
def set_limit_proc(proc_command):
    monitor = proc_command[1]
    task = proc_command[3]
    cpu_shares = proc_command[5]
    path = "/sys/fs/cgroup/cpu/"+monitor+"/"+task+"/cpu.shares"
    if os.path.exists(path):
        f = open(path, "w")
        f.write(cpu_shares)
        f.close
    else:
        return

# fun that adds a process ID into the cgroup's tasks of an application
# if the path does not exist, the fun returns does nothing
def add_proc(proc_command):
    monitor = proc_command[1]
    task = proc_command[3]
    pid = proc_command[4]
    path = "/sys/fs/cgroup/cpu/"+monitor+"/"+task+"/tasks"
    if os.path.exists(path):
        f = open(path, "a")
        f.write(pid)
        f.close
    else:
        return

# main fun for reading line by line input and calling the selected
# procedure
def limit():
    diction = {"create":create_proc, "remove":remove_proc,
    "set_limit":set_limit_proc, "add":add_proc}

    for line in sys.stdin:

        proc_command = line.rstrip().split(":")
        print proc_command[0]
        proc = diction[proc_command[0]]
        #we assume that the input is always correct
        proc(proc_command)

if __name__ == '__main__':
    limit()

```

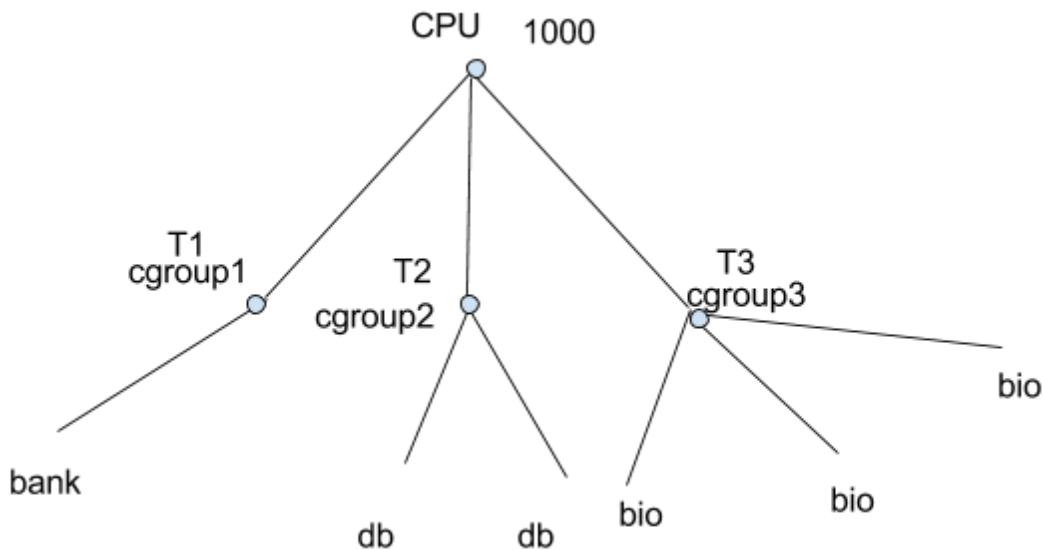
### 3 Περιγραφή υλοποίησης των ζητούμενων

Από τον κώδικά των παραπάνω δύο ζητούμενων εκτελέσιμων ικανοποιούνται τα πρώτα τρία ζητούμενα της εκφώνησης της άσκησης. Συγκεκριμένα, τα ελάχιστα χιλιοστά εξασφαλίζονται με την τύπωσή τους από τη limit ενώ το policy εγγυάται τα χιλιοστά των εφαρμογών μέσω του score.

Ζητούμενο 4. Εάν υπάρχουν ελαστικές και ανελαστικές εφαρμογές υπό επιτήρηση να εξασφαλίζεται ότι όλοι οι επεξεργαστές του κόμβου χρησιμοποιούνται και κανείς δεν μένει αδρανής.

Απάντηση:

Ο daemon εξασφαλίζει το ότι δε θα μένουν ανεκμετάλλευτοι οι επεξεργαστικοί πόροι εφαρμόζοντας την εξής ανάθεση πόρων :



Κάθε cgroup χρησιμοποιεί ποσοστό cpu:

$$\Pi\% = \frac{t_i}{\sum t} \times 1000$$

Με τον τρόπο αυτό η cpu είναι πλήρως απασχολημένη. Ταυτόχρονα η policy εξασφαλίζει τα ελάχιστα χιλιοστά μίας εφαρμογής.

Επίσης γίνεται έλεγχος της λίστας των εφαρμογών από το daemon σε περίπτωση που πρέπει να γίνει force της εφαρμογής που πάει να μπει στη cpu, αν πρόκειται για μία εφαρμογή με υψηλή προτεραιότητα, θα πάρει πόρους από εφαρμογές με χαμηλή προτεραιότητα οι οποίες προσωρινά τερματίζονται.

#### 4 Συνοπτική παρουσίαση εκτέλεσης του script cgmon\_demo.sh και παραδείγματος εξόδου

```
root@snf-742918:~/cgmon# ./cgmon_demo.sh

WD=${PWD}/demo

cgmon daemon stop -w ${WD}
Server stopping...      OK

sleep 1

cgmon daemon start -w ${WD} -p ${WD}/mod_policy_cpumin.py -l ${WD}/mod_limit_cpu.py
Server starting...      OK

sleep 1

cgmon app list
+-----+
| App    |
+-----+
| No apps |
+-----+

sleep 1

# assuming total 2000 millicpus == 2 cpus
cgmon policy create -n platinum -p 1000
cgmon policy create -n silver -p 500
cgmon policy create -n elastic -p 50
cgmon policy list
+-----+-----+
| Name          | cpu |
+-----+-----+
| default_min100| 100 |
| default_min1000| 1000|
| default_min500 | 500 |
| elastic        | 50  |
| platinum       | 1000|
| silver         | 500 |
+-----+-----+

sleep 1

cgmon app spawn -p platinum -e "stress -c 2" -n BANKDB
cgmon app spawn -p silver -e "stress -c 2" -n WEBDB
cgmon app spawn -p elastic -e "stress -c 2" -n VIDEOENC
cgmon app spawn -p elastic -e "stress -c 2" -n SPAMBOT

cgmon app list
+-----+
| App    |
+-----+
| WEBDB  |
| BANKDB |
| VIDEOENC |
| SPAMBOT |
+-----+

sleep 5

# This should fail: not enough cpu
cgmon app spawn -p platinum -e "stress -c 2" -n MEDICALDB
ERROR::Server returned: 'Resources 'CPU' returned negative scores'

sleep 1

# forcing it violates policies
cgmon app spawn -p platinum -e "stress -c 2" -n MEDICALDB -f

sleep 10
pkill -f stress
```