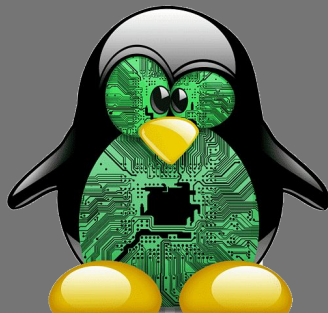

cgroups



Thierry Vaira
LaSalle Avignon BTS SN IR

v0.1 28/05/2020

Présentation

cgroups (*control groups*) est une fonctionnalité du noyau Linux pour limiter, compter et isoler l'utilisation des ressources (processeur, mémoire, utilisation disque, etc.).

Un **cgroup** est un groupe de contrôle de processus.

Un groupe de contrôle est une suite de processus qui sont liés par le(s) même(s) critère(s).

Ces groupes peuvent être organisés hiérarchiquement, de façon que chaque groupe hérite des limites de son groupe parent.

Contrôleurs

Le noyau Linux fournit l'accès à plusieurs contrôleurs (sous-systèmes) à travers l'interface **cgroup** :

- **cpuset** : allocation de ressources CPU et mémoire vive
- **cpuacct** : permet de comptabiliser la consommation de cycle CPU
- **memory** : contrôle de la mémoire vive et du cache d'un groupe
- **devices** : autorise ou refuse l'accès à un périphérique
- **net_cls** : gère l'accès au réseau
- **blkio** : gère l'accès aux périphériques de type *block* (disque durs...)

Objectif

Cgroups fournit :

- Limitation des ressources : des groupes peuvent être mis en place afin de ne pas dépasser une limite de mémoire
- Priorisation : certains groupes peuvent obtenir une plus grande part de ressources processeur ou de bande passante d'entrée-sortie
- Comptabilité : permet de mesurer la quantité de ressources consommées par certains systèmes en vue de leur facturation par exemple
- Isolation : séparation par espace de nommage pour les groupes, afin qu'ils ne puissent pas voir les processus des autres, leurs connexions réseaux ou leurs fichiers.
- Contrôle : figer les groupes ou créer un point de sauvegarde et redémarrer

Test : installation

Il faut installer les paquets suivants :

- cgroup-tools - contrôle et surveillance des groupes de contrôle (outils)
- libcgroup1 - control and monitor control groups (library)

```
$ sudo apt install cgroup-tools
```

Test : liste des contrôleurs

```
$ cat /proc/cgroups
```

#subsys_name	hierarchy	num_cgroups	enabled
cpuset	8	1	1
cpu	5	74	1
cpuacct	5	74	1
blkio	4	74	1
memory	6	112	1
devices	2	74	1
freezer	9	1	1
net_cls	7	1	1
perf_event	10	1	1
net_prio	7	1	1
...			

Test : création des politiques (à la volée)

Création d'une politique par défaut sans limite

\$ sudo cgcreate -g cpu:/cpudefault

Création d'une politique avec limite

\$ sudo cgcreate -g cpu:/cpulimited

Limite avec un ratio de consommation CPU de 2:1

(valeur totale CPU = 1024 donc $1024/2 = 512$)

\$ sudo cgset -r cpu.shares=512 cpulimited

Test : outil de test

L'outil **stress** permet de créer une charge de calcul sur un système. L'option **-c** (**--cpu**) lance N travail(s) basé(s) sur **sqrt()**.

```
$ sudo apt install stress
```

La commande **cgexec** permet de lancer un processus en le plaçant dans un groupe (**-g**). Le processus va hériter des limitations de ce groupe.

Exemple :

```
$ sudo cgexec -g cpu:cpudefault stress -c 1 --timeout 200s &
```

Test n°1

On lance 4 fois : (avec la politique par défaut sans limite)

```
$ sudo cgexec -g cpu:cpudefault stress -c 1 --timeout 200s &
```

Ou :

```
$ for i in $(seq 4); do (sudo cgexec -g cpu:cpudefault stress -c 1 --timeout 200s &) ; done
```

```
$ top
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
26164	root	20	0	8240	92	0	R	25,2	0,0	0:04.21	stress
26167	root	20	0	8240	96	0	R	25,2	0,0	0:02.68	stress
26173	root	20	0	8240	92	0	R	25,2	0,0	0:01.95	stress
26170	root	20	0	8240	96	0	R	24,8	0,0	0:02.21	stress

On obtient une répartition équitable du CPU (autour de 25%).

```
$ sudo killall stress
```

Test n°2

On lance 1 fois : (avec la politique par défaut sans limite)

```
$ sudo cgexec -g cpu:cpudefault stress -c 1 --timeout 200s &
```

On lance 3 fois : (politique avec limite ratio 2:1)

```
$ for i in $(seq 3); do (sudo cgexec -g cpu:cpulimited stress -c 1 --timeout 200s &) ; done
```

```
$ top
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
26222	root	20	0	8240	96	0	R	66,3	0,0	0:13.17	stress
26231	root	20	0	8240	96	0	R	11,2	0,0	0:01.44	stress
26234	root	20	0	8240	96	0	R	11,2	0,0	0:01.42	stress
26235	root	20	0	8240	96	0	R	10,9	0,0	0:01.42	stress

Le ratio 2:1 sur 66% donne 33% réparti sur 3 processus soit 11% chacun.

```
$ sudo killall stress
```

Configuration

La configuration de **Cgroups** se fait dans un fichier **/etc/cgconfig.conf**. Par exemple :

```
group limited {  
    perm {  
        admin {  
            uid = nom_utilisateur; # l'administrateur du groupe  
        }  
        task {  
            uid = nom_utilisateur; # utilisateur capable de lancer des processus avec ce groupe  
        }  
    }  
    cpu {  
        cpu.shares = 50; # partage du CPU  
    }  
    memory {  
        memory.limit_in_bytes = 2147483648; # limitation 2GiB  
    }  
}
```
