Petite introduction aux concepts de **Cgroups** & **Namespaces** sous Linux

c’est quoi ? ça sert à quoi ?

Laurent FACQ - Lundi 21 Novembre 2022 – ANF PAAS - Mathrice - CIRM

Plan

- **Cgroups**
  - Terminologie / Vue d’ensemble
  - Quelques caractéristiques
- **Namespaces**
  - Terminologie / Vue d’ensemble
  - Quelques caractéristiques

Les Cgroups

Terminologie / vue d'ensemble

***C**groups = **Contrôle** Groups*

- Objectifs : re**Grouper** des processus pour les **contrôler**
- Quelles formes de contrôles ?
  - **Comptabiliser** l'usage de ressources (*accounting*)
  - **Limiter/Garantir** l'usage des ressources
- Contrôle de quelles ressources ?
  - Des **sous-systèmes** (*subsystems* = *resource controllers* = *controllers*)
    - CPU/Exécution, Mémoire, Réseau, Accès aux périphériques/Débits,...
- Caractéristiques :
  - Système **hiérarchique** (cgroups dans cgroups)
    - Héritage/Réduction des droits,
    - Agrégation hiérarchique pour Accounting/Limites
  - S'appuie sur des pseudos fichiers et répertoires /proc… /sys...

Cgroups - versions V1 → V2

- Version 1 (Linux Kernel 2.6.24)
  - Premier développement, très complet
  - Étendu au fil de l'eau pour supporter plus de ressources et de fonctionnalités
    - ***=> manque de cohérence***
- Version 2 (Linux Kernel 4.5.0) (expérimental dès 3.10.0)
  - Réécriture plus cohérente, mais incomplète vs V1

- Version 1 & 2 cohabitent (compatibilité) au sein d'un même système Linux
  - … mais pas au sein d'un même cgroup
  - Pour un certain sous système (ex : mémoire) et un cgroup donné, on doit choisir entre la v1 ou la V2

Les sous systèmes 1/3
(contrôleurs de ressources)

- Cpu / Execution
  - **cpu** : garantir une quotité cpu et limiter la consommation max cpu
  - **cpuacct** : comptabiliser la consommation cpu
  - **cpuset** : « binding » (attacher) des processus à des cœurs
  - **freezer** : suspendre/relâcher l'ensemble des processus d'un cgroup checkpointing/restart (migrations)
- Mémoire
  - **memory** : comptabiliser, limiter l'utilisation de la mémoire et du swap
  - **hugetlb** : comptabiliser, limiter l'utilisation de la mémoire sous forme de *huge pages*

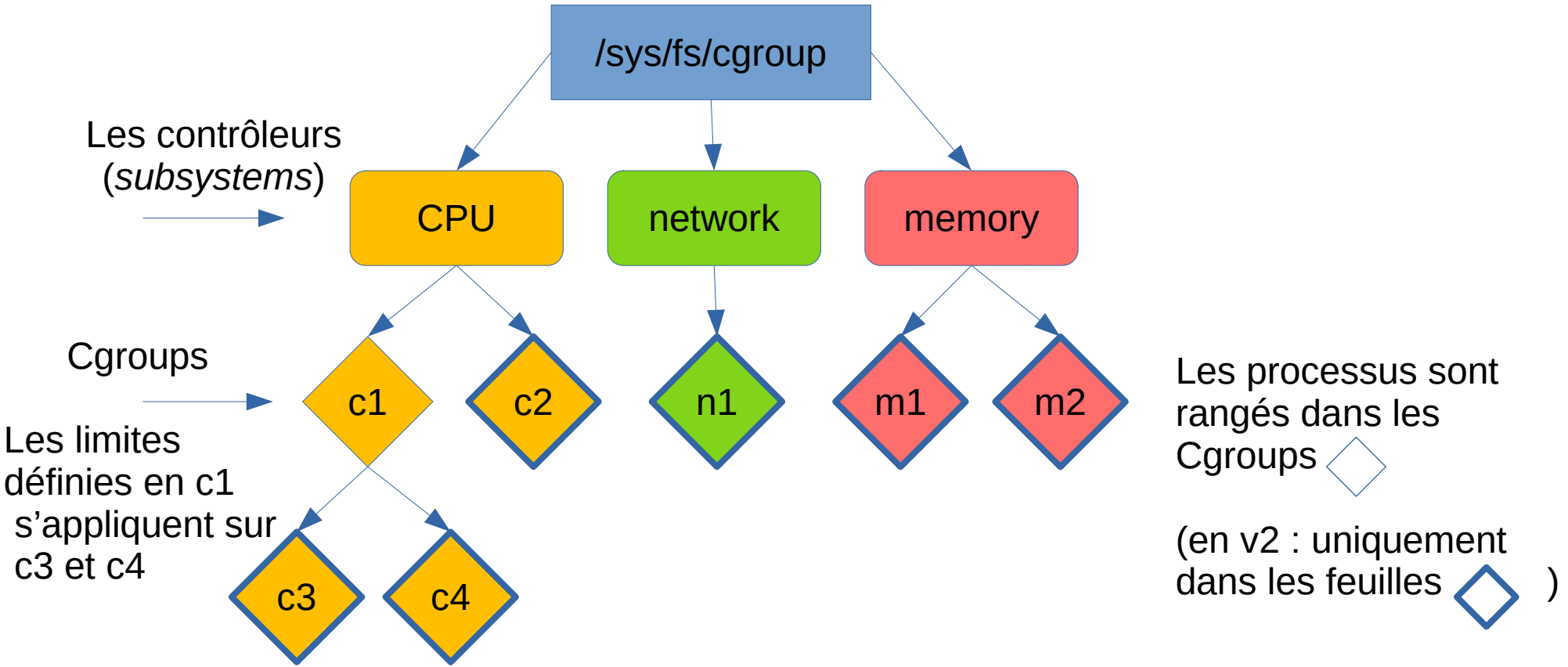
Les sous systèmes 2/3
(contrôleurs de ressources)

- Devices
  - **blkio (io** en v2**)** : contrôle de l'accès et de la bande passante d'accès aux périphériques blocks (disques)
  - **devices** : contrôle de l'accès aux périphériques
- Réseau
  - **net_cls** : tagguer (***n°classid* unique**) les paquets sortant du cgroup (iptables, TC/traffic controller)
  - **net_prio** : affecter un niveau de priorité (**générique** entre 0 et 7) aux paquets sortants du cgroup

Les sous systèmes 3/3
(contrôleurs de ressources)

- Divers
  - **perf_events** : gestion des compteurs hardwares
    associés au processeurs/processus du cgroup
  - **pids** : limitation du nombre de processus par cgroup
  - **rdma** : gestion des possibilités RDMA
    (Remote Direct Memory Access)
    échanges mémoire directs entre ordinateurs (InfiniBand)

Cgroups hierarchiques



Les Namespaces

Terminologie / vue d’ensemble 1/2

Name**s**paces

- Objectifs :
  - **Isoler** des processus en les mettant dans des espaces de nommage disjoints
  - … pour leur faire croire qu’ils sont tout seuls ou privilégiés
    - Ex :« init » veut avoir le PID 1
    - Ex : les processus privilégiés aiment avoir des Uid/Gid à 0
  - Besoin : Sécurité / Multi-tenant

Terminologie / vue d’ensemble 2/2

Namespaces

- Isolation par type de ressources
  - Chaque type de ressources à ses propres namespaces de confinement/isolation
    - **Cgroups**, Montages FS, Réseau/Nommage, Numérotation (ID), Horloges, ...
- Caractéristiques :
  - Système en partie **hiérarchique**
    (namespaces user et pid imbriqués)
  - S’appuie sur des pseudos fichiers et répertoires

Les **N**ame**s**paces 1/2

- cgroup :
  - arborescences cgroup propre à ce namespace
- ipc :
  - Inter Process Communication propres au namespace (socket, sémaphore, mémoire partagée, file de message...)
- mnt :
  - **montages** de filesystems propres au namespace
- net :
  - pile v4/v6, routage, filtrage, interfaces, **veth (virtual ethernet devices)** propres au namespace => possibilité de faire des bridges entre namespace

Les **N**ame**s**paces 2/2

- pid : process Id (*hierarchique*)
  - **numérotation** des processus propre au namespace
    (en correspondance avec le namespace parent)
- time :
  - **horloges** propres au namespace (monotonie, boot-time)
- user : user/group Id (*hierarchique*)
  - redéfinition des **uid/gid** par correspondances avec le namespace parent.
    - Ex : Un processus peut avoir des UID/GID à 0 dans son namespace
      est en réalité à 65535 dans le système réel
  - **racine « */* », keyrings** et **capabilities**
- uts : *unix time sharing*
  - **hostname** et **domainname** propre au namespace

Commandes liées aux namespaces

- **unshare** : démarrage d'un processus dans de nouveaux namespaces
- **nsenter** : démarre un processus dans d'autres namespace
- **lsns** : lister les namespaces et processus

Conclusion

- **Namespaces + Cgroups**
  - **Isoler** les ressources
  - **Comptabiliser** l'usage des ressources
  - **Limiter/Garantir** l'usage des ressources

=> … tout ce qu'il faut pour
conteneuriser des **processus**
dans un contexte **multi-tenant**