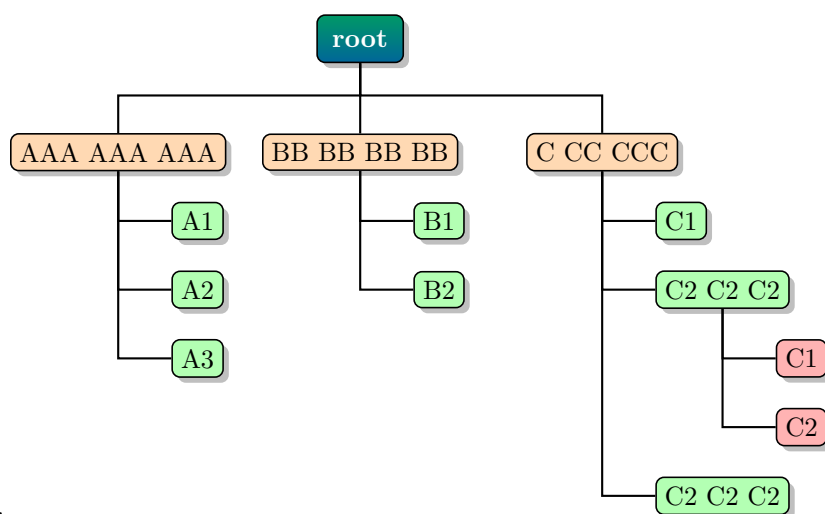


Résumé Journalier

Joffrey Hérard

6 avril 2017

1 WBS



Ebauche standard de base a modifier .

2 Détails sur chaque test

2.1 Lien avec le Hardware

Au delà de la manière de gérer de chaque technologies de conteneurisation ou d'hypervision , il y a les limites imposée elle même par les hyperviseur/conteneurs.

TABLE 1 – My caption

Systeme	Ressources	Windows Server 2012 r2	VMWare
Hote	Processeurs logiques	320	2 CPU physiques (480 CPU logiques),
	Memoire physiques	4 To	4 To (12 To sur systèmes certifiés)
	Processeurs virtuel par hote	2048	to do
Invite	Socket virtuel par machine virtuelle	64	to do
	Memoire par machine virtuelle	1 To	Illimité
	Machine virtuelle active par hote	1024	1024 VM, 4096 vCPU
Cluster	Nombre de noeud maximum	64	8
	Nombre max de machine virtuelle	1000	to do

2.2 Impact d'un Hyperviseur

Sachant que le monde des Hyperviseurs est assez vaste, chacun à une politique différentes ne seras-ce que par son choix sur le type de son hyperviseur (1 et 2). Par conséquent il y a forcément des Hyperviseurs qui vont avoir une politique différente sur l'accès concurrent à des ressources. Les hyperviseurs ont besoin d'isoler les interruptions et les accès à la mémoire. C'est très coûteux en termes de performances. Les surcoûts en termes de performances pour virtualiser un système comportent trois aspects principaux : la virtualisation du processeur, de la mémoire et des entrées/sorties.

Processeur L'utilisation d'un hyperviseur au-dessous du système d'exploitation diffère du schéma habituel où le système d'exploitation est l'entité la plus privilégiée dans le système. De nombreuses architectures de processeur ne fournissent que deux niveaux de privilèges. Dans une virtualisation efficace. L'utilisation du processeur a des implications critiques sur les performances des autres caractéristiques du système. Dans l'évaluation des performances d'une machine virtuelle, les processus sont gérés par la machine virtuelle à la place du système d'exploitation sous-jacent. Les threads émulés des environnements multi-thread, en dehors des capacités du système d'exploitation d'origine, sont gérés dans l'espace utilisateur à la place de l'espace noyau, permettant le travail avec des environnements sans support natif des threads. De bonnes performances sur un microprocesseur multi-cœur sont obtenues grâce à l'implémentation de threads natifs pouvant attribuer automatiquement le travail à plusieurs processeurs, ils permettent un démarrage plus rapide de processus sur certaines machines virtuelles.

2.3 Impact d'un conteneur

A la base, le concept de conteneurisation permet aux instances virtuelles de partager un système d'exploitation hôte unique, avec ses fichiers binaires, bibliothèques ou pilotes. Cette approche réduit le gaspillage des ressources car chaque conteneur ne renferme que l'application et les fichiers binaires ou bibliothèques associés. On utilise donc le même système d'exploitation (OS) hôte pour plusieurs conteneurs, au lieu d'installer un OS (et d'en acheter la licence) pour chaque VM invitée. Le conteneur de chaque application étant libéré de la charge d'un OS, il est nettement plus petit, plus facile à migrer ou à télécharger, plus rapide à sauvegarder ou à restaurer. Enfin, il exige moins de mémoire. La conteneurisation permet au serveur d'héberger potentiellement beaucoup plus de conteneurs que s'il s'agissait de machines virtuelles. La différence en termes d'occupation peut être considérable, car un serveur donné accueillera de 10 à 100 fois plus d'instances de conteneur que d'instances d'application sur VM.

Références

- [1] Performance des hyperviseurs. https://fr.wikipedia.org/wiki/Performances_des_hyperviseurs.
- [2] An Analysis of Performance Interference Effects in Virtual Environments <http://ieeexplore.ieee.org/document/4211036?arnumber=4211036>
- [3] The impact of Docker containers on the performance of genomic pipelines <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4586803/>