# Module 3 : Les clients pour l'infonuagique

3.1 La compagnie Netflix ne possède pas d'infrastructure informatique et effectue toutes ses opérations informatiques avec les services Amazon EC2. Une stratégie qu'elle utilise est de démarrer une instance, de mesurer sa performance et de la conserver si elle est au-dessus d'un certain seuil, ou au contraire de la terminer si la performance n'est pas acceptable. Combien d'instances un fournisseur de services infonuagiques devrait-il ordonnancer sur un même nœud physique : 4, 8, 10...? Est-ce que ce nombre doit plutôt dépendre de la mémoire ou du nombre de coeurs du nœud, du comportement des l'instances? Quelle est la conséquence de mettre trop ou trop peu d'instances sur un même nœud?

Avec trop d'instances, une ou des ressources (e.g., CPU, disque, réseau) seront saturées et deviendront un goulôt d'étranglement. En conséquence, tout s'en trouvera considérablement ralenti inutilement, et les clients seront mécontents. Avec trop peu d'instances, aucune des ressources ne sera pleinement utilisée et il y aura de la capacité excédentaire inutilisée et donc gaspillée. L'idéal est de grouper sur un même nœud des instances complémentaires qui n'utilisent pas toutes les mêmes ressources en même temps. Ce peut être plusieurs instances avec un taux d'utilisation faible et des périodes de pointes différentes, une instance gourmande en CPU avec une autre limitée par les entrées-sorties… Il est possible en première approximation de prendre un chiffre comme 6 ou 8 pour le nombre d'instances, en fonction de la charge moyenne d'une instance, et de la mémoire et du nombre de coeurs disponibles sur un nœud. Sur un service comme Amazon, les clients choisissent un type d'instance, ce qui donne une bonne idée quant à la charge attendue par l'instance et peut aider à estimer le nombre d'instance optimal pour un noeud. Les métriques collectées sur chaque nœud permettent de savoir si un nœud est sous-utilisé et candidat à recevoir une instance supplémentaire.

3.2 Peut-on rouler Kubernetes par-dessus OpenStack ? OpenStack par-dessus OpenStack ? Qu'en est-il de OpenStack par-dessus Kubernetes et Kubernetes par-dessus Kubernetes? Expliquez pour chaque cas comment cela peut ou non se faire, et commentez sur l'opportunité et l'efficacité d'une telle configuration.

Kubernetes par-dessus OpenStack est possiblement la configuration la plus courante. Le fournisseur infonuagique offre des machines virtuelles via OpenStack et le client utilise cela pour un déploiement Kubernetes. Il n'y a qu'un seul niveau de virtualisation, ce qui est supporté par matériel et la performance est adéquate. L'inverse est aussi fréquent. OpenStack est assez compliqué à déployer et une configuration au-dessus de Kubernetes permet de simplifier grandement son déploiement. Un fournisseur infonuagique pourrait facilement déployer OpenStack par-dessus Kubernetes pour offrir des machines virtuelles à

ses clients. Encore là, il n'y a qu'un seul niveau de virtualisation et la performance est bonne. Le client peut encore ici rouler Kubernetes par-dessus OpenStack (qui est par-dessus Kubernetes) sans problème et avec une bonne performance. Rouler Kubernetes par-dessus Kubernetes est moins fréquent mais est possible avec une bonne performance. La difficulté est que certaines opérations pour configurer les conteneurs de Kubernetes requièrent les privilèges d'administrateur. Les conteneurs au premier niveau devront donc être des conteneurs avec des privilèges spéciaux. Ceci se fait parfois pour une infrastructure de test et d'intégration. OpenStack par-dessus OpenStack, avec la virtualisation à chaque niveau, serait possible mais peu performant. Par contre, il est possible d'utiliser OpenStack en mode « bare metal » pour orchestrer le déploiement de OpenStack par-dessus des nœuds physiques commandés par des modules de gestion de nœud (e.g., IPMI ou AMT). Tout comme OpenStack par-dessus Kubernetes, ceci est un moyen d'automatiser le déploiement de OpenStack.

3.3 Une faille de sécurité dans les processeurs Intel récents (Meltdown) permet à une application de lire le contenu de la mémoire du système d'exploitation qui est lié par sa table de page, mais normalement inaccessible sauf lors des appels système. Il est important dans un système infonuagique que les instances soient isolées ne puissent sortir de leur coquille. Est-ce que cette vulnérabilité pose un problème entre deux applications dans une instance ou sur l'hôte? Entre une application et le système d'exploitation, dans une instance ou sur l'hôte? Entre une instance et l'hôte sur OpenStack/KVM? Entre une instance et l'hôte sur Kubernetes? Entre l'hôte et une instance? Expliquez.

Meltdown permet à une application de lire le contenu de la mémoire du système d'exploitation, ce qui inclut des informations sur les autres applications dans le même système. Ceci pose un problème sérieux entre deux applications ou entre une application et le système d'exploitation, autant sur une instance que sur l'hôte. Avec Kubernetes, tous les conteneurs sont sur le même système que l'hôte et le problème demeure présent entre une instance et l'hôte. Ce n'est pas le cas avec OpenStack/KVM car l'instance ne pointe pas vers la mémoire de l'hôte avec ses tables de pages. Le cas entre l'hôte et une instance est complètement différent. Le système d'exploitation de l'hôte gère entièrement la mémoire des instances et pouvait déjà l'accéder sans problème. La vulnérabilité n'y change rien.

3.4 Une machine virtuelle dans KVM contient 4GiO de mémoire organisée en pages de 4KiO. On désire migrer cette machine virtuelle vers une autre machine physique localisée sur le même réseau local et connectée avec un réseau de 100Mb/s. Il est donc possible de transmettre 2000 pages/seconde. Par contre, on veut minimiser le temps où la machine virtuelle est interrompue pour la migration et il faut donc copier toutes les pages sans suspendre la machine virtuelle en notant les pages qui sont modifiées après avoir été copiées. L'exécution de la

machine virtuelle modifie ses pages en mémoire au rythme de 100 pages différentes par seconde. On veut que le temps de suspension, pour copier les dernières pages modifiées, soit inférieur à 1 seconde. Combien de temps durera la migration au total? Combien d'itérations de copies seront requises? Quel sera le temps d'interruption?

La machine virtuelle contient 1Mi pages = 1048576, ce qui demande 1048576 pages / 2000 pages/s = 524.288s. Pendant ce temps, 52428 pages pourraient être modifiées, ce qui demande 26.21s secondes au deuxième tour. Pendant ce temps, 2621 pages seraient modifiées, demandant 1.31 secondes pour le transfert du troisième tour. Rendu là, on suspend la machine virtuelle et il y aura 131 pages à transférer en .06 secondes pour finaliser le transfert avant de reprendre l'exécution de la machine virtuelle sur la nouvelle machine physique. Il faudra donc 4 tours de copies, un temps total de 551.86s mais une interruption de seulement .06s.