Compte-rendu et suivi de la mise en place de l'infrastructure

Pour la mise en place de l'infrastructure serveur, nous avons décidé de partir sur un serveur Kubernetes. Étant donné les caractéristiques du projet, nous avons besoin d'une carte graphique pour faire tourner nos modèles de langage dans de meilleures conditions.

Pour l'infrastructure, idéalement, il faudrait deux environnements : un de production et un de test. Ces deux environnements seront séparés par des namespaces différents dans le cluster. Il nous suffira donc d'un master et d'un worker. Un deuxième worker aurait été idéal pour une meilleure disponibilité de notre application, mais étant limités en ressources, nous nous contenterons de cela.

Nous avons décidé d'acheter un ordinateur. En fonction du budget, nous avons pu trouver pour 80 euros un ordinateur équipé d'un processeur Intel Core i7 4930k, 16 Go de RAM et une GTX 770. À cela, nous avons ajouté un HDD de 1 To et remplacé la carte graphique par une GTX 970.

Après l'achat de l'ordinateur, il a fallu installer un hyperviseur pour gérer les VM. Le choix s'est porté sur Proxmox, car il est open source et que nous avons déjà travaillé avec. L'installation ne fut pas de tout repos. Un problème est survenu lors de l'installation : nous nous sommes retrouvés face à un écran noir, sans plus d'informations pour déboguer.
Après investigation, il s'est avéré que le problème était lié au pilote de la carte graphique utilisée pendant l'installation. La solution trouvée consistait à ajouter l'instruction « nomodeset » lors du lancement de l'installation. Cette instruction permet d'utiliser un pilote par défaut au lieu d'en chercher un nouveau.
Une fois cela fait, nous avons tenté d'installer un pilote approprié pour la carte, mais cela s'est avéré très compliqué. Les pilotes disponibles dans les packages ne détectaient pas la carte, et l'installation du pilote fourni par Nvidia était impossible en raison d'une incompatibilité avec rivafb, le pilote utilisé par la machine. Le seul moyen de désactiver ce pilote était de recompiler le noyau, ce qui était impossible sans perdre les ajouts de Proxmox.
Si l'installation du pilote était recommandée, c'était avant tout pour virtualiser la carte graphique et attribuer de la VRAM à chaque VM. Dans notre cas, cela n'est pas spécialement utile, car seule une VM a besoin de VRAM.

La solution a donc été de configurer le **Passthrough** de la carte graphique, c'est-à-dire de donner l'usage entier de la carte à une seule VM.
Après cela, nous avons créé deux VM Debian : une pour le worker et une pour le master. Sur le worker, nous avons installé les pilotes Nvidia et configuré CUDA pour l'utilisation du GPU dans nos fonctionnalités d'IA.
Sur le master, nous sommes en train d'installer Kubernetes. Pour l'instant, le cluster

n’est pas opérationnel en raison de plusieurs problèmes de **CrashLoopBackOff** sur les conteneurs qui sont censés faire tourner le cluster.