

Accueil de la communauté du Deep-Learning sur le calculateur Myria du CRIANN

Bilan à 18 mois - mi-juillet 2018

Cette action a été réalisée dans le cadre du projet MNSN (Maison Normande des Sciences du Numérique), cofinancé par la Région Normandie et l'Union européenne.

Lors de l'élaboration de ce projet en 2015, une demande avait été formulée par le LITIS pour accéder à des ressources de calcul intensif régionales pour ce type de travaux. Une réponse a pu être apportée d'une part grâce au renouvellement du calculateur en 2017 (financement Région Normandie, État et Union européenne), et d'autre part grâce à l'embauche de Benoist Gaston en septembre 2016 (financement MNSN).

Sur le territoire normand, les principaux laboratoires d'informatique concernés sont le LITIS et le GREYC. Historiquement, les chercheurs de ces laboratoires utilisaient peu les ressources du CRIANN, dont le mode de fonctionnement était perçu comme trop contraignant (dossier scientifique, batch, pas de droits administrateurs, etc.). Mais les directions de ces laboratoires restent particulièrement conscientes du besoin des équipes en ressources de toujours plus grande capacité, et des implications en termes d'infrastructure que cela engendre.

L'objectif de cette action est donc de :

- Démontrer aux équipes que l'utilisation du plateau technique HPC du mésocentre régional présente certains intérêts : stabilité, puissance, disponibilité, etc.
- Dimensionner, côté CRIANN, les ressources techniques et humaines requises ainsi que les évolutions à mener pour répondre durablement aux besoins des équipes de recherche concernées.

1. Ressources du calculateur Myria mises en jeu

Myria est un calculateur d'une puissance totale de 600 TFlops (403 TFlops Xeon, 170 TFlops GPU et 27 TFlops Xeon Phi KNL) et principalement doté de nœuds de calcul généralistes :

- 366 nœuds bi-sockets Broadwell (28 cœurs à 2,4 GHz, 128 Go de RAM DDR4 à 2400 MHz).

La communauté du Deep-learning utilise le sous-ensemble de nœuds GPU :

- 9 nœuds GPU Kepler K80 (2 cartes par nœud soit 4 unités GPU embarquant chacun 12 Go de mémoire)
- 8 nœuds GPU Pascal P100 (2 cartes par nœud soit 2 unités GPU embarquant chacun 12 Go de mémoire)
- 1 nœud doté de 3 cartes GPU Pascal P100 (3 unités de traitement GPU, 12 Go de mémoire embarquée par unité)

Par ailleurs, des tests ont été effectués par l'équipe du CRIANN sur la partie KNL :

- 10 nœuds de calcul dotés chacun d'un Xeon Phi KNL 7210 (64 cœurs à 1,3 GHz et 96 Go de RAM DDR4 à 2133 MHz par nœud, 16 Go de mémoire rapide MCDRAM embarquée dans le processeur KNL).

L'ensemble des ressources est interconnecté par un réseau Intel Omni-Path à faible latence et haut débit (100 Gbit/s) et accède à un système de stockage rapide d'une capacité de 2,5 Po.

2. Calendrier de la mise en œuvre

- Octobre 2016 : suivi de la formation Deep learning Nvidia par Benoist Gaston
- Rencontre avec les équipes du LITIS (définition des suites logicielles à installer, des cas tests)
- Décembre 2016 – Janvier 2017 : déploiement des outils sur première tranche de Myria et réalisation des premiers tests
 - Janvier 2017 : présentation des premiers résultats dans le cadre d'Equip@meso
- Avril 2017 : demi-journée de formation des utilisateurs du LITIS (9 utilisateurs)
- Été 2017 : tests CRIANN sur architecture Intel KNL (en attente de performances)
- Septembre 2017 : présentation lors des journées mésocentre.
- Octobre 2017 : rencontre avec le GREYC (Caen), suivie par la création d'un projet pour l'équipe « image » de ce laboratoire.
- Juillet 2018 : réunions de bilan avec les utilisateurs.

3. Bilan de l'utilisation (juillet 2018)

Depuis la mise en production du supercalculateur Myria en mars 2017, le CRIANN a accueilli 5 projets dans le domaine du Deep Learning. Ces projets sont principalement portés par les équipes du LITIS et du GREYC et traitent d'activité de recherche sur les méthodes de Deep Learning ou sur l'application de ces méthodes principalement dans le domaine de la santé. Un 6^{ème} projet, porté par le CRIANN, traite de mise en œuvre des framework de Deep Learning sur l'architecture KNL d'Intel. Deux autres projets ont été déposés en 2018 mais n'ont pas encore consommé d'heures de calcul. Le tableau Tab. 1 présente les projets actifs sur Myria en juillet 2018.

Projet	Utilisateurs	Heures ¹	Typologie de travaux
N°1	1	16 000 (2017) <1000 (2018)	1 GPU (P100)
N°2	3 (2 actifs)	< 1000 h (2017) < 2000 h (2018)	2 GPU
N°3	4 (un actif)	2000 h (2017) 9000 h (2018)	1 GPU (P100)
N°4	4 (2 actifs)	30 000 h (2017) 50 000 h (2018)	1 GPU (K80 ou P100) Tests en multi-nœuds et multi GPU
N°5	2	50 h (2018)	multi GPU
N°6 (tests KNL)	3	12 000 h (2017) 23 000 h (2018)	KNL

Tab. 1 : liste des projets Deep Learning hébergés au CRIANN (juillet 2018)

L'utilisation est assez irrégulière, avec des pics d'utilisation intensive et des périodes de moindre activité. La question du partage des ressources sera détaillée dans la suite du document (cf. 5.3).

¹ Comptabilité système (ACCT) : seule la consommation des ressources CPU est comptabilisée (cf. § 6.1 concernant l'évolution à apporter).

Stockage

Pour les projets traités, le quota d'espace de stockage attribué aux utilisateurs a été adapté à leurs besoins, de l'ordre de 100 Go et jusqu'à 1 To.

Les besoins en terme de stockage restent donc - pour le moment - dans les fourchettes habituelles, mais à la différence des projets CFD habituellement traités, les besoins concernent les données d'entrée, et en particulier des jeux de données de test des modèles.

NB : il est souhaitable que les jeux de données standards susceptibles d'être partagés par différentes équipes soient déposés dans des espaces de stockage partagés.

4. Retour d'expérience du CRIANN sur la gestion de la logithèque

4.1. Logithèque et procédures d'installation

La gestion de la logithèque scientifique du CRIANN se base sur les deux critères suivants :

- Minimiser les impacts sur le système
- Permettre l'installation de multiples versions d'un même outil

Pour cela, les logiciels et les bibliothèques informatiques dont ils dépendent sont installés dans des répertoires spécifiques. Un outil, *module*, permet aux utilisateurs de charger à la demande un l'environnement logiciel (outil, version et dépendances) dont ils ont besoin. Cette démarche de cloisonnement permet d'installer, sur un même système, différentes versions d'un même outil tout en évitant les conflits entre elles.

La logithèque de Deep Learning déployée au CRIANN est conforme à cette politique d'installation. Cette logithèque se compose des principaux frameworks couramment utilisés dans la discipline. Le choix de ces outils a été fait en concertation avec les utilisateurs (ou futur utilisateurs) concernés.

La plupart de ces frameworks sont utilisés via python, c'est pourquoi nous avons choisi le schéma d'installation suivant² :

- installation d'une version de python qui est dédiée au Deep Learning
- installation des frameworks et de leur interface python pour la version de python dédiée

L'intérêt de cette démarche est que l'installation des différents frameworks ne nécessite que la construction d'une version de python ; à partir de cette version les frameworks peuvent être installés à partir de packages standards pré-construits par les distributeurs et téléchargés automatiquement sur un dépôt distant par l'outil *pip*.

Pour des raisons de performances, ces frameworks nécessitent l'utilisation de ressources de type GPU. Toutes les installations sont donc réalisées avec prise en charge des ressources gpu.

La plupart des frameworks sont disponibles pour python2 et python3. Nous avons procédé à une installation spécifique pour ces deux versions de python. Le tableau Tab. 2 présente la liste des frameworks installés et indique pour quelles versions de python ils sont disponibles.

² Initialement, sur demande d'un utilisateur, le logiciel Caffe avait été installé en stand-alone ce qui avait nécessité la recompilation complète de l'outil et de la bibliothèque dont il dépend, donc une installation particulièrement lourde.

	python 2.7	python 3.6
Caffe	oui	oui
pyTorch	non	oui
Theano	oui	oui
Tensorflow	oui	oui
opencv	oui	oui

Tab. 2 Frameworks DL installés au CRIANN

4.2. Principaux problèmes rencontrés

Foisonnement des versions

La communauté de Deep Learning est très active et par conséquent les frameworks proposés ont des cycles d'évolution très rapide. A titre d'exemple, un des frameworks les plus populaires, Tensorflow, est passée de la version 1.0 à la version 1.9 de février 2017 à juillet 2018. Ces outils ont des dépendances système, dont les montées de versions de Cuda, dont dépendent d'autres logiciels. Les cycles d'évolution étant rapide, il n'y a quasiment aucune maintenance à moyen terme sur les versions antérieures, en particulier une mise à jour de certaines dépendances d'une version peut entraîner le dysfonctionnement de celle-ci.

Les différents utilisateurs du CRIANN sont parfois contraints d'utiliser des versions de frameworks figées ou, au contraire, demandent à utiliser la dernière version en date. Il est donc nécessaire de gérer plusieurs versions de frameworks au sein d'une même version de python. Bien que l'installation d'un paquet standard soit une tâche simple, l'installation de multiple des versions sur de courtes échéances devient vite laborieuse en particulier dans la gestion des environnements spécifiques.

En conséquence, la gestion de versions cloisonnées nécessite un outil plus souple et/ou plus automatique que celui actuellement utilisé.

Architecture système spécifique

La plupart des packages mis à disposition par les distributeurs de frameworks sont construits pour des environnements système standards et répandus. Bien que compatibles et fonctionnels, ces packages ne sont pas optimaux sur des environnements spécifiques tels qu'une infrastructure HPC, et en particulier de son réseau d'interconnexion rapide. Pour utiliser au mieux ce type de système, il est nécessaire de construire les packages spécifiquement pour l'infrastructure cible.

En particulier, pour tirer parti du réseau OPA du calculateur Myria avec l'outil Tensorflow en multi-nœuds, c'est une version expérimentale du logiciel, bénéficiant de peu de retour d'expérience, qui a dû être utilisée et qui a nécessité un environnement de compilation dédié.

Ce mode de fonctionnement (construire ses propres paquets plutôt qu'utiliser des packages standards) n'est pas recommandé par les développeurs de Tensorflow.

5. Retour d'expérience des utilisateurs, expression de besoins et pistes d'amélioration

Les deux laboratoires fonctionnent en mettant des machines virtuelles à disposition des utilisateurs. Au LITIS, les machines sont attribuées pour une durée de l'ordre du mois, et attachées à une ressource physique GPU.

Parmi les contraintes liées à l'utilisation des ressources mutualisées du CRIANN, les points initialement perçus comme bloquants étaient :

1. L'absence de droits d'administration du système pour les utilisateurs : sur leur ressources internes, les utilisateurs ont pour habitude d'administrer leur système ce qui leur donne une autonomie dans la gestion de leurs applications.

2. Le mode batch : il alourdit le mode de développement incrémental des utilisateurs, est plus difficile à mettre en place qu'un mode interactif, ne permet pas de prévoir précisément la période d'exécution des travaux, et oblige l'utilisateur à estimer la durée de ses travaux.
3. Le partage des ressources de calcul : les utilisateurs craignent que les ressources ne soient qu'occasionnellement disponibles augmentant ainsi les délais de restitution des calculs.

Un bilan est donc établi sur chacun de ces points lors d'une réunion de bilan avec les utilisateurs concernés. Le cas échéant, des pistes d'amélioration sont proposées.

5.1. Absence des droits d'administration

Les demandes de mise à jour (du système ou des logiciels) ont été traitées dans un délai raisonnable. Occasionnellement, certains chercheurs ont installé par eux-mêmes des packages manquants.

Il s'avère que l'utilisation des droits d'administration est plus une habitude des usagers qu'un réel besoin.

5.2. Le mode batch

Le mode batch est bien adapté aux travaux d'apprentissage qui forment la phase la plus coûteuse en temps de calculs des projets de Deep Learning hébergés au CRIANN. De plus, par son aspect automatique, il permet aux utilisateurs d'éviter certaines tâches manuelles laborieuses nécessaires sur leurs ressources internes³.

La limite de durée des calculs (48 heures sur les ressources GPU⁴) n'est pas bloquante pour ces utilisateurs : ils ont le cas échéant mis en place des fichiers de reprise (certains n'avaient pas cette habitude mais l'ont maintenant intégrée).

Besoins spécifiques pour la mise au point des modèles (calibrage)

Le mode batch, tel qu'il est paramétré actuellement au CRIANN, paraît en revanche inadapté aux travaux de calibrage des modèles. Les utilisateurs ont alors besoin de réactivité et d'interactivité : accès rapide à des ressources pour y lancer des calculs assez courts, en suivant l'évolution des résultats.

Plusieurs utilisateurs souhaiteraient effectuer des travaux de calibrage: un délai de deux heures pour l'accès aux ressources comme pour la durée des calculs serait un bon compromis pour ces calculs (par rapport à la durée « incompressible » d'initialisation, sur un cas dimensionnant d'un utilisateur du GREYC).

- Le CRIANN envisage la création d'une file réservée à des travaux courts sur GPU, le cas échéant en dédiant un serveur K80 (soit 4 GPU). Par ailleurs, les 2 serveurs dédiés à la visualisation peuvent aussi être utilisés (s'ils sont disponibles) puisqu'ils sont dotés chacun d'une carte K80.

5.3. Partage des ressources

Globalement, l'utilisation simultanée de plusieurs ressources a permis aux utilisateurs d'obtenir des temps de restitution meilleurs qu'avec leurs ressources internes.

Mais des pics d'usage intensif des ressources GPUs ont été observés et certains utilisateurs se sont découragés avec des temps d'attente trop longs et du coup ont abandonné le CRIANN.

³ Pour l'un des doctorants, qui jonglait entre deux machines virtuelles dans son laboratoire et devait gérer manuellement l'orchestration de ses travaux et les transferts de fichiers, le lancement des travaux sur le calculateur au travers du batch s'avère plus confortable et plus efficace car propice à l'automatisation.

⁴ Cette valeur limite est adaptée par le CRIANN pour une utilisation optimale des ressources en mode partagé. Elle varie (de 3 à 72 heures) suivant les ressources ciblées.

Plusieurs voies sont envisageables. Actuellement, il n'y a pas de limite à l'utilisation de ressources GPU par un même utilisateur. Il n'y a pas non plus de politique de fair share. Tous ces points doivent donc être étudiés par l'équipe du CRIANN pour améliorer la fluidité du trafic ressentie par les utilisateurs.

Un affichage plus direct des ressources disponibles serait également le bienvenu pour les utilisateurs. Au cours de la réunion de bilan, l'idée d'un forum utilisateurs a été proposée comme vecteur de communication et de régulation entre utilisateurs. Mais du point de vue du CRIANN, ce qui fonctionne pour la gestion de ressources à l'échelle d'un laboratoire peut difficilement être transposé à une plus grande échelle : les règles doivent prévaloir, ainsi que la surveillance de la charge au quotidien par les administrateurs. Cependant cette proposition mérite une attention.

- **Mieux partager les ressources, mieux informer les utilisateurs des disponibilités constituent les principaux points à améliorer par le CRIANN.**

5.1. Autres besoins exprimés

Par ailleurs, les utilisateurs du GREYC ont besoin de profiler les consommations GPU / CPU / RAM / et transferts entre mémoires de leurs applications.

- Les outils disponibles ou à installer seront proposés aux utilisateurs.

6. Exploitation du plateau technique HPC : réflexions et évolutions envisagées

Avec un recul d'environ 18 mois dans l'accueil d'utilisateurs de la communauté du Deep Learning, il ressort que, du point de vue du mésocentre, plusieurs points relatifs à l'exploitation du plateau technique HPC sont à considérer.

6.1. Comptabilité des heures consommées

Jusqu'à présent, le CRIANN se base sur la comptabilité du système (ACCT) pour le décompte des heures consommées par les utilisateurs. Cette solution avait été retenue historiquement par le CRIANN pour sa fiabilité, mais elle n'est pas adaptée à l'utilisation des GPU.

L'alternative consiste à utiliser la comptabilité du logiciel de batch (Slurm) et à travailler sur la comptabilité « elapsed » des travaux. Cette solution est actuellement opérationnelle en doublon de la comptabilité acct. Cependant, le décompte des heures peut s'avérer complexe (plusieurs GPU dans un même serveur qui peut être en mode partagé ou non).

NB : une modification a été apportée au dossier de demande d'heures, les utilisateurs de ressources GPU sont invités à renseigner la durée totale de leurs travaux.

6.2. Administration de la logithèque

L'administration de la logithèque est complexe : mises à jour très fréquentes, et versions standard non-adaptées au HPC. À cela s'ajoute le risque de conflit entre des installations de versions différentes.

Les conteneurs, actuellement en test pour les besoins d'une autre communauté, pourraient apporter une solution à ce problème.

6.3. Nécessaire évolutivité des ressources

Les ressources matérielles mises à disposition des utilisateurs de cette communauté peuvent devenir rapidement obsolètes : ce sont les cartes GPU les plus récentes qui intéressent les utilisateurs (P100 plus prisées que K80 sur Myria, et demande pour de nouvelles cartes Volta).

Le mésocentre doit donc être en capacité d'intégrer au fil du temps de nouvelles générations de GPU (ou d'autres ressources qui seraient adaptées à cette activité).

Pour des raisons de compatibilité entre générations de matériels, cette intégration sera possible tant que le calculateur reste relativement récent. De la même façon, les conteneurs, évoqués plus haut comme une solution aux évolutions rapides des logithèques, fonctionnent sous réserve de compatibilité de la couche matérielle et de la couche logicielle basse.

D'autre part, les projets accueillis dans le cadre de cette action effectuent du traitement de données au moyen de techniques de Deep Learning, ce qui n'est qu'une petite partie de l'activité des laboratoires concernés. D'autres besoins existent au sein de ces laboratoires, quelques demandes ont d'ores et déjà été formulées auprès du CRIANN :

- Une demande de ressources HPC pour une utilisation intensive de Matlab sous Windows (environnements imposés) ;
- Un demande de type « big-data » (Hadoop).

Le spectre est donc large et ne peut pas toujours être couvert par l'utilisation du calculateur.

Enfin, les laboratoires concernés ont une activité contractuelle conséquente qui leur permet d'acquérir des ressources. En terme d'infrastructure, vu leur puissance électrique, ces ressources doivent bénéficier d'un hébergement adapté dans un datacentre.

Dans certains cas, ces ressources pourraient être intégrées à un calculateur et mises à disposition d'autres communautés lorsqu'elles ne sont pas utilisées (proposition du GREYC). Une proposition similaire avait été formulée par l'UMR IDEES (géographes). Un mode de fonctionnement doit donc être proposé pour répondre à ces demandes.

Il s'agit au final d'optimiser l'utilisation des ressources ainsi que leurs coûts d'exploitation.

7. Conclusion

En conclusion de ce retour d'expérience, il apparait que le mode de fonctionnement du calculateur est globalement adapté aux besoins de la communauté du Deep-Learning même si des améliorations peuvent être apportées. Les utilisateurs ont pu mesurer les bénéfices du plateau technique HPC en terme de puissance et de stabilité. Le CRIANN doit cependant apporter des améliorations concernant le partage des ressources.

La stratégie d'exploitation du plateau technique HPC pour les besoins de la communauté DL doit être adaptée, du fait de la spécificité des matériels et des logiciels. Pour répondre durablement aux besoins de cette communauté, il paraît indispensable d'être en capacité de suivre le cycle de d'évolution rapide des logiciels, ainsi que d'intégrer régulièrement les dernières technologies matérielles.