> mais aussi en science par exemple pour identifier des cellules cancéreuses sur des images biomédicales, trouver des événements clés en physique des hautes énergies, repérer des événements biomédicaux rares…

Afin que les scientifiques puissent extraire de la richesse des volumes énormes de données mesurées, modélisées ou simulées, l'évolutivité est la clé. Cette évolutivité consiste à répartir les calculs sur un grand nombre de processeurs (ou nœuds de calcul), et donc d'effectuer les calculs plus rapidement ou bien de traiter des volumes de données plus importants. Les programmes doivent alors être capables de tirer parti d'autant de processeurs que possible.

MONTE-CARLO, PATRIE DU HASARD

Par exemple, dans la communauté HPC, les scientifiques ont tendance à se concentrer sur la modélisation et la simulation avec des techniques telles que Monte-Carlo afin de produire des représentations précises de ce qui se passe dans la nature. Rappelons que les méthodes Monte-Carlo sont une famille de procédés algorithmiques visant à calculer une valeur numérique approchée en utilisant des techniques probabilistes. L'idée essentielle est d'utiliser le hasard pour résoudre des problèmes qui seraient déterministes en principe. Ils sont souvent utilisés dans des problèmes physiques ou mathématiques et sont plus utiles lorsqu'il est difficile voire impossible d'utiliser d'autres approches.

Les méthodes de Monte-Carlo sont intéressantes, car elles peuvent être parallélisées, et sont donc théoriquement rapides (chaque élément fourni par le processus probabiliste étant indépendant, les calculs peuvent se faire sur autant de processeurs que disponibles). Cependant, elles ne convergent que si la taille de l'échantillonnage est suffisamment grande. Chaque processeur doit donc traiter séquentiellement plusieurs éléments, augmentant d'autant le temps de calcul total.

En pratique, les physiciens des hautes énergies et les astrophysiciens examinent des images à identifier grâce à des techniques d'apprentissage profond (le *deep learning*), c'est-à-dire en les comparant à celles d'une base de données importante. Pour ce faire, les images de référence ont été étiquetées avec des informations les décrivant, puis livrées à l'ordinateur pour qu'il apprenne à « voir » ce qu'elles représentent. Ainsi éduquée, la machine peut reconnaître ce que montrent des images inconnues (on parle d'inférence).

Toujours en astrophysique, le programme RobERt (Robotic Exoplanet Recognition) traque la composition de l'atmosphère des exoplanètes. Mis au point à l'University College de Londres, il fonctionne grâce à une phase d'apprentissage à partir de larges bases de données non structurées recueillies par les instruments de mesure (*voir la figure page précédente*).

Dans d'autres domaines, l'apprentissage consiste à examiner des séries, faire du traitement du signal, des analyses harmoniques, de la modélisation et de la prédiction avec des systèmes non linéaires…

De même, l'IA tente de générer des représentations précises de ce qui se passe dans la nature à partir de données non structurées recueillies par des capteurs, des caméras, des microscopes électroniques, des séquenceurs… La nature inconnue de ces données non structurées conduit les *data scientists* à passer une grande partie de leur temps à extraire et nettoyer les informations utiles pour entraîner le système IA.

Les programmes écrits en langage productif, tel Julia, parviennent rarement à tirer parti du supercalculateur

La convergence HPC-IA concerne aussi bien les algorithmes qui doivent s'adapter à la taille des données et à celle de l'ordinateur cible, que la façon dont les calculs sont répartis sur plusieurs processeurs, le matériel, le logiciel, le prétraitement des données. Les langages de programmation à haute productivité sont également un moteur de rapprochement. De tels langages, comme Python et Julia, permettent de développer des programmes rapidement, car ils sont plus intuitifs dans leur écriture. L'objectif est que les utilisateurs puissent implémenter leurs modèles de calcul, indépendamment du volume de données disponible et de la plateforme de calcul cible.

Soulignons que les seules mesures de performance qui comptent pour la phase d'apprentissage du *deep learning* sont le temps nécessaire pour obtenir le modèle, et la précision de ce dernier. On sait depuis des décennies que cette étape d'apprentissage se comporte de façon quasi linéaire en fonction du nombre de nœuds de calcul, c'est-à-dire que le temps de

calcul diminue proportionnellement au nombre de nœuds de calcul.

L'évolutivité des applications d'apprentissage pour le *deep learning* sur des architectures distribuées est bien plus limitée. L'une des raisons est l'indispensable communication des résultats intermédiaires entre les différents processeurs de l'architecture distribuée, ce qui est très coûteux en temps. Ainsi, on a utilisé des mesures de performance liées au matériel, à savoir les sous-systèmes mémoire de stockage des données au plus proche du processeur, les caches qui accélèrent cet accès aux données, et les calculs flottants (portant sur des nombres décimaux, par opposition aux calculs sur des entiers) pour identifier les solutions matérielles plus susceptibles de supporter des performances élevées, et d'atteindre rapidement l'objectif, en l'occurrence fournir le modèle.

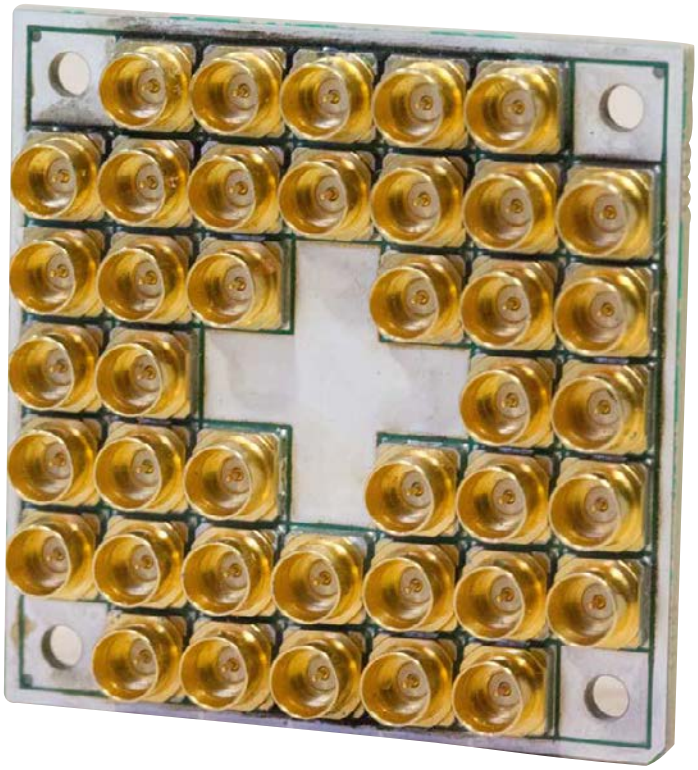
TOUJOURS PLUS DE FLOPS

Le fossé qui nous sépare de la convergence HPC-IA idéale ou au moins attendue par les communautés doit être rapidement comblé. La collaboration avec le monde de la recherche qui a permis d'atteindre une performance de 15 pétaflops avec 9 600 nœuds équipés de processeurs Intel® Xeon® Phi™ sur le supercalculateur Cori du National Energy Research Scientific Computing Center (le NERSC), un organisme dédié au HPC, à Berkeley, aux États-Unis, est un exemple de progrès matériel.

En effet, auparavant, on ne savait pas faire travailler ensemble et efficacement autant de nœuds de calcul : au-delà d'un nombre seuil de nœuds, le temps de calcul total ne diminuait plus. Les supercalculateurs actuels ont quelques dizaines de milliers de processeurs, et ceux qui permettront d'atteindre la prochaine barrière de l'exaflops devront sans doute faire travailler ensemble plus de 100 000 processeurs, soit un ordre de grandeur de plus.

Les principaux efforts concernant les logiciels pour l'IA consistent à aider les *data scientists* à utiliser des outils logiciels familiers fonctionnant aussi bien sur des petits que des très gros systèmes, et sur le matériel actuel aussi bien que sur le matériel du futur.

Un autre défi logiciel important dans la convergence du HPC et de l'IA est l'unification des modèles de programmation. Les programmeurs HPC peuvent être des gourous de la programmation parallèle, c'est-à-dire de la répartition des calculs effectués en même temps par plusieurs processeurs, mais l'IA est principalement programmée avec des outils de type MATLAB, à savoir des outils de représentation dans des langages de haut niveau et un environnement interactif qui permet d'exécuter du calcul intensif plus rapidement qu'avec les langages de programmation traditionnels tels que C, C ++ et Fortran.



© INTEL

Cette puce quantique contient 17 qubits supraconducteurs. Elle a été mise à disposition de l'université technologique de Delft, aux Pays-Bas.

La communauté IA essaie de résoudre le difficile problème d'obtenir des performances élevées sur des architectures évolutives en taille, sans avoir besoin de former les *data scientists* à la programmation parallèle de bas niveau, c'est-à-dire proche du matériel et exploitant explicitement ses caractéristiques : en d'autres termes, les *data scientists* doivent s'abstraire des machines.

Dans cette optique, en collaboration avec des partenaires universitaires, Intel a multiplié par 10 les performances d'Apache Spark (un moteur rapide et général pour le traitement de données à grande échelle), grâce notamment au développement de bibliothèques mathématiques optimisées. De plus, Intel, en collaboration avec Julia Computing et le MIT, a réussi à accélérer de façon significative les programmes écrits en Julia aussi bien au niveau d'un nœud de calcul (monoprocesseur ou biprocesseur) qu'au niveau d'un *cluster* (multiprocesseur), permettant ainsi l'évolutivité tant recherchée par les utilisateurs.

Des outils open source (ParallelAccelerator ou HPAT) transforment des programmes écrits en langages productifs (Julia et Python) en codes qui seront exécutables sur des supercalculateurs, et qui fourniront de très hautes performances. De la sorte, on pallie un inconvénient majeur : le plus souvent, les programmes écrits en langage productif ne parviennent pas à tirer parti du supercalculateur car ils ne prennent pas en compte ses caractéristiques.

D'autres développements matériels sont en cours. Intel a annoncé en octobre 2017 la mise au point de la famille de processeurs Intel® >

> Nervana™ Neural Network Processors (NNP), dont l'architecture est inspirée des réseaux de neurones du cerveau. Une collaboration avec Facebook aidera à développer des méthodes et globalement l'écosystème IA lié à ces dispositifs. De nouveaux jeux d'instruction pour des processeurs à venir ont été annoncés à l'été 2017, ils accéléreront certains calculs du *deep learning*. Côté logiciel, le développement d'outils de productivité donnera accès plus facilement aux puissances de calcul disponibles.

QUANTIQUE ET NEUROMORPHIQUE

L'IA et le HPDA mis en œuvre sur l'infrastructure HPC conduiront à des avancées importantes dans de nombreux domaines grâce à l'extraction d'informations dans de gros volumes de données, par exemple pour le déploiement de la conduite automobile autonome. Plus en amont, d'autres segments de l'informatique émergent, pour l'instant encore cantonnés au stade de la recherche, et qui, associés à l'analyse de données, sont théoriquement prometteurs.

À ce titre, deux secteurs se distinguent : l'ordinateur quantique et les puces neuromorphiques pourraient accélérer une partie des calculs, valider des décisions prises, supprimer des choix possibles… En effet, si les supercalculateurs savent calculer rapidement certains algorithmes, l'ordinateur quantique serait idéal pour les calculs combinatoires. De son côté, la puce neuromorphique peut apprendre seule. En repensant les algorithmes pour tirer avantage de chacune de ces techniques, on réduirait notablement les temps de calcul.

Un ordinateur quantique exploite le phénomène physique de superposition d'états. Alors qu'un bit classique est soit dans l'état 0 soit dans l'état 1, l'équivalent quantique, un qubit, est dans une combinaison des deux états. Exploiter cette propriété accélérerait de façon exponentielle les calculs en parallèle. De la sorte, les ordinateurs quantiques pourraient s'attaquer à des problèmes inaccessibles aux ordinateurs classiques : simuler la nature pour faire progresser la recherche en chimie, en science des matériaux et en modélisation moléculaire ; créer un supraconducteur à température ambiante ; découvrir de nouveaux médicaments.

En novembre 2017, à l'occasion du colloque QuTech sur l'architecture des ordinateurs quantiques qui s'est tenu à Delft, aux Pays-Bas, Intel a annoncé la mise à disposition d'une puce de test supraconductrice de 17 qubits à son partenaire académique de recherche sur le sujet, l'université de technologie de Delft. Cet événement illustre les progrès dans la recherche et le développement d'un système informatique quantique fonctionnel. De fait, une puce de 49 qubits est prévue d'ici la fin de l'année 2018.

Quant aux puces neuromorphiques, Intel a mis au point une puce autodidacte unique en son genre. Baptisée Loihi, elle imite le mécanisme du cerveau en apprenant à fonctionner en analysant la réaction de son environnement. L'informatique neuromorphique s'inspire de l'architecture du cerveau et de ses calculs associés, notamment du fait que les réseaux neuronaux relaient les informations, modulent le poids des interconnexions, et stockent ces changements localement aux interconnexions. Cette puce extrêmement économe en énergie, qui utilise les données pour simultanément apprendre et décider, devient plus intelligente avec le temps et n'a pas besoin d'être entraînée de façon traditionnelle.

Les puces neuromorphiques utilisent les données pour apprendre et décider en même temps

Les avantages des puces d'autoapprentissage sont vertigineux. Par exemple, avec les informations de rythme cardiaque d'un individu dans diverses conditions (après le jogging, après un repas ou avant d'aller au lit), un système neuromorphique en déduirait ce qu'est un rythme cardiaque « normal ». Le système serait alors apte à surveiller en continu des données cardiaques et surtout à identifier un signal « anormal» pouvant trahir une pathologie.

Ce type de logique pourrait également être appliqué à d'autres domaines, comme la cybersécurité où une anomalie ou une différence dans les flux de données pourrait identifier une violation ou un piratage puisque le système a appris le «normal» dans divers contextes.

L'IA n'en est sans doute qu'à ses balbutiements. D'autres architectures et méthodes continueront d'émerger et amélioreront encore les performances. Ainsi, les modèles d'apprentissage automatique ont récemment fait d'énormes progrès, mais ces systèmes ne sont le plus souvent pas capables de faire des généralisations. C'est un des défis à relever. ■

BIBLIOGRAPHIE

M. ANDERSON *ET AL.*,
Bridging the Gap Between HPC and Big Data Frameworks,
***Proceedings of the VLDB Endowment*, vol. 10,**
pp. 901-912, 2017.

J. PARK *ET AL.*,
Faster CNNS with direct sparse convolutions and guided pruning, *ICLR 2017 Conference paper*, 2017.

T. KURTH *ET AL.*,
Deep Learning at 15PF : Supervised and Semi-Supervised Classification for Scioentific Data
Cornell University Library, 2017.