

[Communiqués de presse](#)

Les primitives de Qiskit Runtime rendent le développement d'algorithmes plus facile que jamais



Par Blake Johnson et Gilad Ben-Shach, le 03 mai 2022 : Environ un an après le lancement de Qiskit Runtime sous forme d'environnement d'exécution conteneurisé pour les programmes quantiques classiques, nous faisons évoluer le concept afin de permettre une expérience de programmation plus simple via les primitives de Qiskit Runtime. En parallèle, nous permettons à quiconque d'accéder à des processeurs Falcon via un modèle de paiement à l'usage sur IBM Cloud.

Lorsque l'informatique quantique arrivera à maturité, les développeurs auront besoin de plus que le meilleur matériel quantique. Ils auront besoin d'un modèle de programmation et d'exécution pour les programmes quantiques. Ce modèle devra être rapide, efficace, facile à utiliser et évolutif. Nous construisons les primitives de Qiskit Runtime pour répondre à ce besoin.

Lorsque nous avons publié en 2021 [Qiskit Runtime](#), disponible en version limitée, notre préoccupation principale était d'améliorer la vitesse et l'efficacité des applications d'informatique quantique. Notre effort initial visait à permettre aux utilisateurs de regrouper l'exécution quantique et classique dans des programmes qui s'exécutent sur une infrastructure de calcul classique colocalisée avec des systèmes IBM Quantum. En changeant le modèle d'exécution d'un service qui exécute des circuits à un service qui exécute des programmes, nous avons éliminé certains des plus importants goulets d'étranglement relatifs aux performances d'une application.

En particulier, nous avons supprimé les temps de communications associés aux aller-retours vers les ordinateurs des utilisateurs. Les programmes exécutés dans Qiskit Runtime ont donc fonctionné plus efficacement, permettant aux utilisateurs de passer moins de temps à attendre dans les files d'attente,

d'exécuter davantage de circuits dans une journée et d'obtenir leurs résultats plus rapidement. Grâce en partie à ces modifications, nous avons constaté une [accélération allant jusqu'à 120X](#) pour un exemple d'algorithme de chimie quantique, et d'autres utilisateurs ont également signalé[1] des [gains de performance](#).

Nous voyons maintenant l'opportunité de tirer parti de cette même infrastructure pour simplifier davantage l'expérience du développeur et, en même temps, établir les bases d'une amélioration des performances. Nous le faisons via les primitives de Qiskit Runtime.

La principale capacité des ordinateurs quantiques qui les distingue des ordinateurs classiques est leur capacité à générer des distributions de probabilité non classiques à leur sortie. Les opérations natives que l'on peut faire avec une distribution de probabilité sont d'échantillonner à partir de celle-ci ou d'estimer des quantités sur celle-ci. Par conséquent, ces opérations d'échantillonnage et d'estimation constituent les éléments fondamentaux du développement d'algorithmes quantiques.

Nos deux premières primitives de Qiskit Runtime exposent directement ces opérations d'échantillonnage et d'estimation comme interfaces principales vers nos systèmes quantiques via respectivement l'échantillonneur et l'estimateur.

L'échantillonneur

L'échantillonneur estime la totalité de la distribution de quasi-probabilité à la sortie d'un circuit quantique en échantillonnant à partir de sa sortie. Ceci est utile pour les algorithmes de recherche tels que la recherche de Grover.

L'estimateur

L'estimateur calcule les valeurs attendues d'observables à la sortie d'un circuit. Ces observables peuvent coder une grande variété de choses, comme la structure électronique d'une molécule, la fonction de coût d'un problème d'optimisation, le noyau d'un problème de machine learning, et bien plus encore.

Lorsque les développeurs utilisent l'une ou l'autre de ces primitives, ils souhaitent exprimer simultanément leur besoin opérationnel ; c'est-à-dire que le développeur peut avoir besoin de connaître une valeur attendue avec une précision cible particulière, ou avec un temps d'exécution maximal. Il s'agit là d'objectifs différents de ceux qui consistent à essayer de contrôler des points de bas niveau tels que le nombre d'exécutions répétées ou la méthode particulière d'atténuation des erreurs utilisée pour obtenir la précision cible.

En élevant ces opérations au niveau de l'interface principale du matériel IBM Quantum, nous posons également les bases permettant aux utilisateurs d'exprimer leurs besoins en termes pertinents pour leur cas d'usage.

Une nouvelle façon d'exécuter vos programmes sur de puissants systèmes quantiques

IBM a ouvert la voie en matière d'informatique quantique basée sur le Cloud depuis 2016, date à laquelle nous avons [mis pour la première fois un processeur quantique à disposition](#) sur la plateforme IBM Quantum, [hébergée par IBM Cloud](#). Nous offrons désormais l'accès à la plus grande flotte de systèmes quantiques haute performance au monde, que nous continuons à mettre à jour avec des systèmes exploratoires et centraux avancés. Actuellement, cependant, l'accès à nos systèmes les plus avancés nécessite un engagement au niveau institutionnel pour l'achat de capacités dédiées via notre offre premium. À l'avenir, nous aimerions mettre l'accès à nos meilleurs systèmes à la portée d'une population plus large de développeurs.

Nous avons mis deux systèmes de 27 qubits à la disposition des utilisateurs avec un plan de paiement à l'usage au tarif de 1,60 \$ par seconde d'exécution.

Notre nouveau plan de paiement à l'usage sur IBM Cloud, lancé en version bêta aujourd'hui, donne accès à deux processeurs Falcon de 27 qubits à toute personne possédant un compte IBM Cloud. Les utilisateurs peuvent développer des programmes quantiques exigeants en utilisant les primitives de Qiskit Runtime, et les exécuter de manière transparente sur nos systèmes de pointe, en ne payant que pour les ressources dont ils ont besoin avec une carte de crédit ou des crédits IBM Cloud (pas d'abonnement requis) au tarif de 1,60 \$ par seconde d'exécution, sans engagement. Un avantage supplémentaire de la disponibilité du service via IBM Cloud est qu'il ouvre la porte aux partenaires de notre écosystème qui conçoivent des services en s'appuyant sur l'API Qiskit Runtime et qui utilisent la gestion des utilisateurs, la facturation et l'infrastructure d'IBM Cloud.

Où tout cela va-t-il mener ?

Le modèle de programmation et d'exécution de la prochaine phase de l'informatique quantique doit être rapide, efficace, facile à utiliser et évolutif.

Les primitives de Qiskit Runtime offrent une expérience qui est déjà plus rapide, plus efficace et mieux adaptée aux besoins des développeurs d'algorithmes que tout autre service quantique comparable. Avec le lancement aujourd'hui de l'accès intégré à IBM Cloud, certains de nos systèmes les plus avancés sont accessibles à un groupe de développeurs plus large qu'il ne l'a jamais été. Mais le dernier avantage important de l'intégration des primitives de Qiskit Runtime dans IBM Cloud est l'évolutivité.

Les gains d'efficacité et de vitesse de Qiskit runtime sont eux-mêmes évolutifs. Au fur et à mesure que nous introduisons de nouvelles optimisations dans la plateforme, les programmes s'exécuteront plus rapidement avec moins d'efforts de la part des développeurs. Au fil du temps, nous introduirons dans le service des outils tels que l'atténuation et la suppression des erreurs, ce qui renforcera encore sa puissance.

Nous prévoyons des gains plus importants à l'avenir avec un [modèle de développement serverless](#) qui permet des configurations flexibles de l'infrastructure quantique-classique, en tirant parti de la grande variété de ressources de calcul classiques disponibles sur le Cloud. Nous travaillons d'arrache-pied à l'élaboration d'outils permettant de mettre en œuvre Quantum Serverless qui n'oblige pas les développeurs à devenir également des experts en infrastructure.

Un avenir est en train de se dessiner où les développeurs identifieront les problèmes nécessitant des solutions quantiques, créeront rapidement et facilement des programmes pour les résoudre et exécuteront ces programmes de manière transparente sans avoir besoin de configurer le matériel. En [suivant notre feuille de route](#) - et avec l'aide de Qiskit Runtime - nous espérons réaliser ce futur quantique « accessible à tous ».

Prêt à faire un pas de plus vers l'informatique quantique « accessible à tous » ? Essayez [Qiskit Runtime](#), dès aujourd'hui.

Contacts presse :

Weber Shandwick pour IBM

IBM

Gaëlle Dussutour

Tél. : + 33 (0) 6 74 98 26 92

dusga@fr.ibm.com

Jennifer Tshidibi / Eric Chauvelot

Tél. : + 33 (0)6 13 94 26 58 / + 33 (0)6 21 64
28 68

ibmfrance@webershandwick.com

[1] Beaulieu, D., Pham, A. [Max-cut Clustering Utilizing Warm-Start QAOA and IBM Runtime](#). arXiv. [Submitted on 30 Aug 2021].
