URL　https://journal.ntt.co.jp/article/40573
DOI　https://doi.org/10.60249/26095004

光量子コンピュータを支えるソフトウェア技術

光量子コンピュータを実用的な計算基盤として活用するには，アプリケーションと光量子ハードウェアをつなぐソフトウェア技術が不可欠です．本稿では，光量子コンピュータ用ソフトウェアの全体像を示すとともに，量子の専門知識がなくても問題記述やパラメータ調整を行えるSDK（Software Development Kit）と，量子・古典計算資源を連携させ，量子誤り抑制によって効率的かつ高信頼な実行を支えるシステムソフトウェアの研究開発を紹介します．

キーワード：#光量子コンピュータ，#光量子SDK，#量子誤り抑制

八木（やぎ）　哲志（さとし）
宮原（みやはら）　和大（かずひろ）
笠江（かさえ）　優美子（ゆみこ）

NTTコンピュータ&データサイエンス研究所

光量子コンピュータ用ソフトウェアの全体像

光量子コンピュータを実際の問題解決に利用するためには，光量子ハードウェアの性能向上だけでなく，アプリケーションと光量子ハードウェアをつなぐソフトウェア技術が不可欠です．光量子コンピュータ用ソフトウェアは，ユーザが解きたい問題を量子計算として記述し，対象となる光量子ハードウェアで実行可能な形へ変換したうえで，計算資源を管理し，得られた結果をユーザへ返す役割を担います．

図1に，光量子コンピュータ用ソフトウェアの代表的な構成を示します．図中の箱は各階層の機能を，箱の間の矢印に付した項目は機能間のインタフェースとして受け渡される主なデータや表現を示しています．

最上位のアプリケーションでは，組合せ最適化，機械学習，物理・化学シミュレーションなど，光量子コンピュータを用いて解決させたい問題を定義します．次に，その問題を量子計算として処理するための量子アルゴリズムを選択・設計します．量子アルゴリズムでは，量子計算と古典計算の役割分担や，計算を反復する方法などを定めます．

ユーザはSDK（Software Development Kit）が提供する量子プログラミングライブラリを用いて，量子アルゴリズムを量子回路として記述します．量子回路は，量子状態に対してどのような演算を，どの順序で適用するかを表すものです．連続量光量子計算では，量子ビットの代わりに光の量子モードを扱い，位相回転，変位，圧搾，2モード演算，測定などを組み合わせて回路を構成します．SDKは，量子回路を記述するためのライブラリに加え，シミュレータや実機への接続機能，サンプルプログラム，ドキュメントなどを提供し，量子プログラムの開発と実行を支援します．図1では，その中でも処理の主経路となる回路記述と実行支援の機能を示しています．

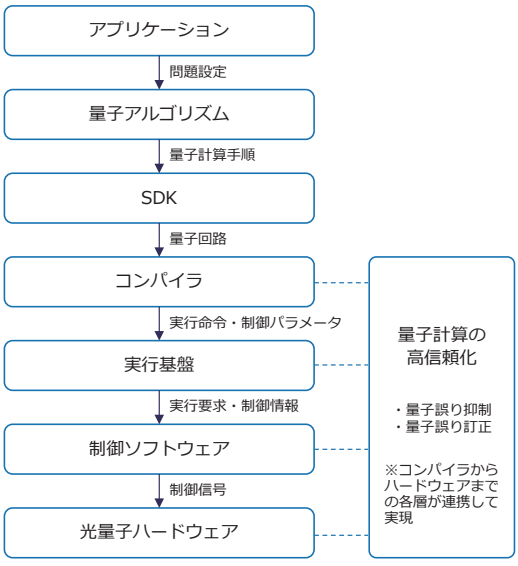

図1　光量子コンピュータ用ソフトウェア全体像

作成された量子回路は，コンパイラによって対象ハードウェアで実行可能なかたちへ変換されます．コンパイラは，量子回路を対象ハードウェアで実行可能な演算や表現へ変換し，必要に応じて回路の構成や実行方法を最適化します．内部では複数段階の中間表現を用いる場合がありますが，通常，ユーザがそれらを直接意識する必要はありません．

コンパイラから出力された実行命令や制御パラメータは，ランタイムやジョブ管理からなる実行基盤へ渡されます．実行基盤は，ジョブ受付，実行順序の決定，量子・古典計算資源の割当て，実行状態の監視などを担います．複数のユーザが光量子コンピュータを共用する場合には，計算資源を安全かつ公平に管理し，装置を効率的に利用することも重要です．

NTTが研究開発を進める光量子コンピュータでは，光の連続的な物理量を利用する連続量方式と，量子もつれ状態に対する測定によって演算を実現する測定型量子計算を用います．時間領域多重[*1]された多数の光パルスから二次元クラスタ状態[*2]を生成し，それぞれの光パルスを適切な測定基底[*3]で測定することで計算を進めます．

このため，測定型の連続量光量子コンピュータ向けコンパイラには，連続量量子回路をクラスタ状態上の測定手順へ変換し，各光パルスの測定基底や測定結果に応じた後続の測定条件などを，実行命令や制御パラメータとして生成する機能が求められます．これらは，実行基盤を介して制御ソフトウェアへ渡され，装置を動作させるための制御情報へ変換されます．装置の動作には，各光パルスに対する測定基底の高速な切替え，測定データの収集，測定結果に基づくフィードフォワード[*4]など，量子状態の時間スケールに合わせた高速制御が必要です．

また，量子コンピュータでは，方式によらず，外部環境の影響や装置の不完全性により，計算結果に誤りが含まれます．このため，量子計算の信頼性を高める量子誤り抑制と量子誤り訂正も，ソフトウェアスタックを横断する重要な技術です．

量子誤り抑制を実現するには，コンパイラや実行基盤を連携させ，回路変換，追加実行，結果の補正を行う必要があります．

一方，量子誤り訂正では，1つの論理的な量子情報を複数の物理的な量子状態に符号化し，計算途中で誤りを検出・訂正しながら処理を進めます．これを実現するには，コンパイラ，実行基盤，制御ソフトウェア，光量子ハードウェアを一体として設計する必要があります．光量子方式では，光損失，有限圧搾，測定誤差などへの対策に加え，将来的にはボソニック符号[*5]や連接符号[*6]を用いた誤り耐性型量子計算への発展が求められます．

NTTコンピュータ＆データサイエンス研究所では，量子アルゴリズムから実行基盤，高信頼化まで，ソフトウェアスタックの各層にわたる研究開発を進めています．本稿では，これらの取り組みの中から，ユーザによる問題記述と実行を支援するSDK，および量子プログラムの効率的かつ高信頼な実行を支えるシステムソフトウェアを紹介します．

ユーザによる問題記述と実行を支援するSDK

光量子方式を含めた量子コンピュータの実用化，特に実際の社会的問題を解く，もしくはビジネスに応用するといった用途を考えると解決すべき課題は大きく2つです．1つは量子コンピュータへ解きたい問題を入力するために「量子アルゴリズムに関する知識」が必要であること．もう1つは量子コンピュータの利点である古典コンピュータを上回

＊1 時間領域多重：時間的に連続して生成される光パルスを，それぞれ独立した量子モードとして利用する方式．装置を大規模化することなく，多数の量子モードを扱える利点があります．
＊2 クラスタ状態：多数の量子モードを量子もつれによって結び付けた状態．測定型量子計算では，各量子モードを順次測定することで量子演算を実行します．
＊3 測定基底：量子状態をどの物理量・方向に沿って測定するかを定める条件．測定型量子計算では，測定基底の選び方によって実行する量子演算が決まります．
＊4 フィードフォワード：先に得られた測定結果に応じて，後続の測定条件や制御内容を変更する処理．
＊5 ボソニック符号：光や超伝導共振器などの連続的な量子状態を利用して，論理量子ビットを符号化する量子誤り訂正符号．
＊6 連接符号：複数の量子誤り訂正符号を階層的に組み合わせることで，論理量子情報の信頼性を高める方式．

る性能を引き出すために「量子コンピュータ・アルゴリズムの性能に関する知識」が必要であることです．いずれの知識も量子に関する高度な専門知識であり修得にも時間を要するため，多くの人が多くの場面で気軽に量子コンピュータの恩恵を受けるための大きな課題です．

NTTコンピュータ＆データサイエンス研究所ではこれらの課題を解決するため，量子に関する専門知識無しで扱える量子コンピュータ向けSDKの開発を進めています．これは前述した量子アルゴリズムを量子回路として記述するSDKを補完する，量子アルゴリズム領域に特化したSDKであり，量子を用いた機械学習や最適化問題といった領域をターゲットにしています．ここでは組合せ最適化問題を解くことができる量子アルゴリズムの1つQAOA（Quantum Approximate Optimization Algorithm）[1]を例にこのSDKが解決する課題について具体的に説明します．QAOAは量子ビット向けのアルゴリズムですが，GKP符号[2]といった符号化手法により量子モードによる量子回路上でも量子ビット向けアルゴリズムを実行可能です．QAOAはある程度ノイズを含む量子コンピュータ上でも効果的なアルゴリズムとされており，早期の実用化が期待できます．

QAOAが解く「組合せ最適化問題」とは，多数の答えの組合せの中から与えられた制約条件を満たすもののうち，もっとも良い答えを得る問題です．例えば，巡回セールスマン問題[3]は，複数の都市を巡る移動ルートを「答えの組合せ」とし，すべての都市を1回ずつ訪問し出発地に戻るという「与えられた制約条件」の下で，移動距離や時間などのコストがもっとも小さいルートを「もっとも良い答え」として求める問題であり，物流の最適化などに応用されます．このように組合せ最適化問題は社会の多くの場面で現れる応用範囲の広い概念です．

QAOAを用いてこの組合せ最適化問題を解くには，量子アルゴリズムに合わせた問題入力と，性能を引き出すためのパラメータ調整が必要です．QAOAに問題入力をするには，各候補解を量子状態に対応付け，その評価値がエネルギーとして表されるハミルトニアンを構築する必要があります．答えの内容を量子ビットのそれぞれにどのように対応させるかを考え，制約条件を満たす良い答えの量子エネルギー状態がもっとも低くなるよう，実際の量子回路に効率的に組み込み可能な範囲のハミルトニアンを考える必要があります．また，ハミルトニアンは問題ごとに設計し直す必要があります．

また，QAOAの性能を引き出すためには量子エネルギー状態の変化をコントロールするパラメータを適切に設定する必要があります．解きたい問題が大きくなるほど，このパラメータの数は多くなり，問題によって内容も異なります．小規模な問題であればすべてのパラメータの組み合わせを試すことも可能ですが，実用的には長時間かかってしまい現実的ではありません．したがって，問題ごとに適したパラメータの傾向を踏まえた効率的なパラメータ調整が求められます．

NTT開発のSDKではこれらの課題を解決し，専門知識がなくても量子コンピュータを使いこなせる機能を提供します．QAOA向けにはハミルトニアン入力の方法はいくつかありますが，従来手法の1つであるQUBO（Quadratic Unconstrained Binary Optimization）*7多項式とNTT開発SDKによる入力を比べてみます．ここではシンプルな例として整数a, b, cに対する$(a+b+2c=7)\wedge(a=0\vee b=0\vee c=0)$という制約条件を記述した例で考えます．従来手法による入力（**図2**）は比較的煩雑であり，各量子ビットに対する式を考えなければなりません．NTT開発SDKによる入力は**図3**で，制約条件の論理式をほぼそのまま直感的に記述され，量子ビットの詳細を考えずとも扱うことができます．パラメータ調整についてもSDKには自動調整機能を実装してあり，NTT独自の知見を組み込んだ効率的なパラメータ調整を提供します．

このようにNTT開発のSDKでは量子コンピュータの専門知識がなくても迅速に導入できるものを提供予

＊7 QUBO：二値変数からなる二次式の最小化として組合せ最適化問題を表現する形式．

手作業による多項式記述

```
def generate_constraint(self):
    x = self.variable_generator()
    a = x[0] + 2*x[1] + 4*x[2]
    b = x[3] + 2*x[4] + 4*x[5]
    c = x[6] + 2*x[7] + 4*x[8]
    sum_const += (a + b + 2 * c - 7) ** 2
    a_is_nonzero = x[9]
    b_is_nonzero = x[10]
    c_is_nonzero = x[11]
    zero_const = 0
    zero_const += ¥
        (a - a_is_nonzero - x[12] - 2*x[13] - 3*x[14]) ** 2 + ¥
        ((1 - a_is_nonzero) * x[12]) + ¥
        ((1 - a_is_nonzero) * x[13]) + ¥
        ((1 - a_is_nonzero) * x[14])
    zero_const += ¥
        (b - b_is_nonzero - x[15] - 2*x[16] - 3*x[17]) ** 2 + ¥
        ((1 - b_is_nonzero) * x[15]) + ¥
        ((1 - b_is_nonzero) * x[16]) + ¥
        ((1 - b_is_nonzero) * x[17])
    zero_const += ¥
        (c - c_is_nonzero - x[18] - 2*x[19] - 3*x[20]) ** 2 + ¥
        ((1 - c_is_nonzero) * x[18]) + ¥
        ((1 - c_is_nonzero) * x[19]) + ¥
        ((1 - c_is_nonzero) * x[20])
    or_zero_const = ¥
        a_is_zero * b_is_zero + b_is_zero * c_is_zero + c_is_zero * a_is_zero - ¥
        (a_is_zero + b_is_zero + c_is_zero) - x[21] * (a_is_zero + b_is_zero + c_is_zero - 2)
    return sum_const + zero_const + or_zero_const
```

図 2　従来方式（多項式表現）のプログラム化イメージ

```
def generate_constraint(self, a, b, c):
    a = self.cxt.integer("a")
    b = self.cxt.integer("b")
    c = self.cxt.integer("c")
    return (
        (a + b + 2 * c == 7).l_and(
            (a == 0).l_or(b == 0).l_or(c == 0)
        ))
```

数理モデルをそのまま記述

図 3　NTT 独自技術のプログラム化イメージ

定です．また，段階的に量子コンピュータの細部に対し指定できる機能も併せて提供し，量子コンピュータに興味を持っていただいた方からより深い研究・ビジネス用途まで幅広くご利用いただける SDK として，量子コンピュータの実用化に向け懸け橋となるものを実現していきます．

効率的かつ高信頼な実行を支えるシステムソフトウェア

ここでは，量子プログラムの効率的な実行を支える実行基盤と，高信頼化のためのノイズ対策を紹介します．

量子コンピュータを複数のユーザが共用する計算基盤として利用する場合，実行基盤にはジョブの受付，実行順序の決定，量子・古典計算資源の割当て，実行状態の監視などの機能が求められます．特に，量子計算では，古典コンピュータ上での前処理，量子コンピュータでの実行，測定結果の収集，古典コンピュータ上での後処理が連続して行われます．そのため，量子コンピュータ単体の実験装置として扱うのではなく，古典コンピュータや制御ソフトウェアと連携する計算サービスとして扱うためのシステムアーキテクチャが重要になります．

ま た，VQE（Variational Quantum Eigensolver: 変分量子固有値ソルバ）や QAOA に代表される量子・古典ハイブリッド計算では，量子コンピュータで得られた結果を古典コンピュータで評価し，その結果を基に次に実行する量子計算の条件を更新する，という反復的な処理が行われます．このような計算では，量子コンピュータの性能だけでなく，古典コンピュータとの連携や実行制御，結果評価，再実行の効率化がシステム全体の性能を左右します．

さらに現在の量子コンピュータでは，実行結果に含まれるノイズへの対策が不可欠です．前述のように，量子誤り抑制は追加の実行や古典コンピュータによる後処理を用いてノイズの影響を低減する技術です．しかし，実際に誤り抑制を利用するには，単に後処理アルゴリズムを用意するだけでは不十分です．誤り抑制手法に応じた追加実行の生成，複数回の実行管理，測定結果の収集，補正結果の評価などを，実行基盤と連携して行う必要があります．

NTT コンピュータ＆データサイエンス研究所では，光量子コンピュータを多くのユーザが利用できる計算基盤へ発展させるため，量子・古典コンピュータの連携，複数ユー

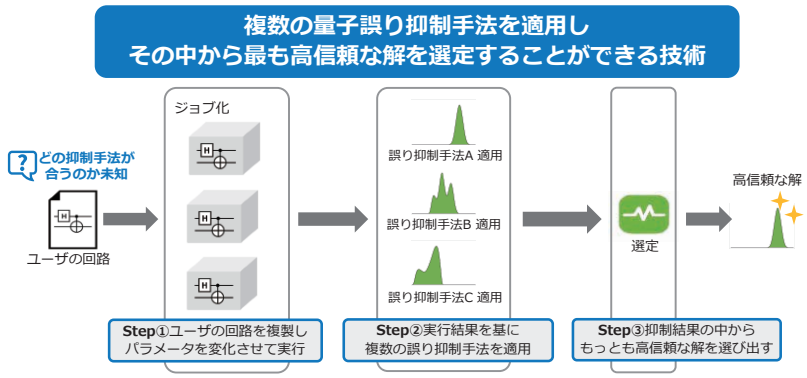

図 4　データ駆動型量子誤り抑制手法

ザによる安全かつ効率的な共用，ノイズ対策などを支えるソフトウェア・システムアーキテクチャの研究開発を進めています．

本稿では，これらの研究開発の中でもノイズ対策に関する成果の 1 つとして，量子計算の高信頼化に向けた量子誤り抑制のアプローチを紹介します．量子誤り抑制にはさまざまな手法が考案されていますが，ユーザが実行したい量子プログラムに対して，どの誤り抑制手法が有効かを事前に判断することは容易ではありません．そこで本アプローチでは，**図 4** に示すように，ソフトウェア工学分野の複数の独立した実装を比較して信頼性を高める N-version programming という考え方を取り入れ，量子コンピュータから得られる確率分布を対象に，複数の誤り抑制結果の一貫性を評価します．これにより，実行結果そのものの特徴を利用して，信頼性が高いと考えられる結果を選定するデータ駆動型の手法を提案しています．量子ハードウェアを大きく変更するのではなく，得られた実行結果を古典コンピュータ上で解析・評価する後処理的な手法であるため，早期の量子コンピュータ利用に向けた実用的なノイズ対策として期待されます[4]．

本研究開発では，光量子コンピュータを実用的な計算サービスとして発展させるため，古典コンピュータと連携するソフトウェア基盤や，出力結果の信頼性を高める技術の検討を進めています．今後も，光量子コンピュータの早期活用と，多くのユーザが安心して利用できる計算基盤の実現をめざし，ソフトウェア・システムアーキテクチャとノイズ対策の両面から研究開発を進めていきます．

今後の展望

本稿では，専門知識を持たないユーザの利用を支援する SDK と，量子・古典計算資源の連携や高信頼化を担うシステムソフトウェアを紹介しました．

NTT コンピュータ＆データサイエンス研究所では，今後も対象とする問題領域や量子アルゴリズムを拡充するとともに，光量子ハードウェアとの連携を一層強化し，実機を用いた評価やユースケースの検証を進めます．さらに，SDK から実行基盤，高信頼化技術までを一体として研究開発することで，多くのユーザが光量子コンピュータを容易かつ安心して利用できる計算基盤の実現をめざします．これらの取り組みを通じて，光量子コンピュータの早期実用化と，新たな社会価値・事業価値の創出に貢献していきます．

■参考文献

(1) E. Farhi, J. Goldstone, and S. Gutmann：“A Quantum Approximate Optimization Algorithm,” arXiv:1411.4028, 2014.
(2) D. Gottesman, A. Kitaev, and J. Preskill：“Encoding a Qubit in an Oscillator,” Physical Review A, Vol. 64, 012310, 2001.
(3) G. Dantzig, R. Fulkerson, and S. Johnson：“Solution of a Large-Scale Traveling-Salesman Problem,” Journal of the Operations Research Society of America, Vol. 2, No. 4, pp.393-410, 1954.
(4) R. Shimazu, S. Endo, S. Hakkaku, and S. Saito：“Data-driven adaptive quantum error mitigation for probability distribution,” arXiv:2511.13231, 2025.

（左から）　八木　哲志／ 宮原　和大／ 笠江　優美子

光量子コンピュータの実用化には，ユーザによる問題記述と実行を支援する SDK と，効率的かつ高信頼な実行を支えるシステムソフトウェアの両方が不可欠です．本稿を通じて，それぞれの役割と一体的な研究開発の重要性を感じていただければ幸いです．

◆問い合わせ先
NTT コンピュータ&データサイエンス研究所