MÁY TÍNH LƯỢNG TỬ: GIỚI THIỆU CHUNG, CÁC VẤN ĐỀ KHÓ KHĂN KHI XÂY DỰNG HIỆN NAY, KẾT QUẢ HIỆN TẠI VÀ DỰ ĐOÁN

Trịnh Viết Cường[1], Lê Văn Vinh[2]

TÓM TẮT

An toàn bảo mật của các hệ thống hiện nay được xây dựng dựa trên các bài toán khó, ví dụ như bài toán phân tích ra thừa số nguyên tố hay bài toán logarit rời rạc. Các bài toán này hiện chưa có thuật toán hiệu quả cài đặt trên máy tính thông thường (máy tính nhị phân) để giải quyết. Gần đây các nhà nghiên cứu đã phát triển được các giải thuật lượng tử hiệu quả để giải các bài toán khó trên. Tuy nhiên, để cài đặt, chạy các thuật toán lượng tử thì yêu cầu phải xây dựng được máy tính lượng tử với cấu hình tương ứng với yêu cầu của thuật toán.

Trong bài báo này chúng tôi trước tiên trình bày giới thiệu chung về các thành phần và cách hoạt động của một máy tính lượng tử, sau đó chúng tôi trình bày những vấn đề kỹ thuật khó khăn hiện nay trong việc xây dựng máy tính lượng tử. Phần cuối của bài báo chúng tôi trình bày những kết quả mới nhất hiện nay trong việc xây dựng máy tính lượng tử, đồng thời trình bày những dự đoán của giới nghiên cứu về lĩnh vực này.

Từ khóa: *Máy tính lượng tử, Qbits, thuật toán Shor, thuật toán Grover.*

1. ĐẶT VẤN ĐỀ

Các hệ thống an toàn bảo mật thông tin hiện nay cơ bản dựa trên sự kết hợp của hai phương pháp chính. Phương pháp thứ nhất và cũng là đơn giản nhất là phương pháp điều khiển truy nhập (Access control). Với phương pháp này người dùng phải có cặp tên người dùng, mật khẩu và quyền hợp lệ thì mới được phép truy nhập. Tuy nhiên, phương pháp này không đảm bảo an toàn tuyệt đối trước kẻ tấn công do nhiều tình huống phát sinh trong thực tế, ví dụ như lộ tên người dùng, mật khẩu, kẻ tấn công là người trong nội bộ, hay hệ thống có lỗ hổng bảo mật,… Phương pháp thứ hai để khắc phục nhược điểm trên là phương pháp dựa trên xác thực và mã hóa. Trong đó với phương pháp xác thực (Authentication), người truy cập phải chứng minh được rằng mình có quyền truy cập thông qua một giao thức xác thực (ví dụ như sơ đồ xưng danh [1]). Với phương pháp mã hóa thì dữ liệu trong hệ thống sẽ được mã hóa, để dù cho kẻ tấn công có xâm nhập được vào trong hệ thống cũng không lấy được thông tin. Ngoài ra, việc truyền dữ liệu trên mạng hiện nay để đảm bảo tính an toàn, yêu cầu tiên quyết là dữ liệu phải được mã hóa trước khi truyền đi (ví dụ như giao thức Transport Layer Socket - TLS). Như vậy phương pháp xác thực và mã hóa có tầm quan trọng rất lớn đối với việc đảm bảo an toàn cho các hệ thống hiện nay. Các phương pháp này về bản chất là dựa trên các bài toán khó như bài toán phân tích ra thừa số nguyên tố hay bài

[1] *Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Hồng Đức; Email: trinhvietcuong@hdu.edu.vn*
[2] *Học viên Cao học Lớp K13 chuyên ngành Khoa học máy tính, Trường Đại học Hồng Đức*

toán logarit rời rạc. Bên cạnh đó, các hệ chữ ký điện tử đang được dùng phổ biến trong thực tế hiện nay như hệ chữ ký RSA, hệ chữ ký DSA hay ECDSA cũng dựa trên các bài toán khó này. Tuy nhiên, hiện nay đã tồn tại các thuật toán lượng tử để giải các bài toán này một cách hiệu quả, ví dụ như thuật toán Shor [2][3][4] giải bài toán phân tích ra thừa số nguyên tố, thuật toán Shor [2][3][4] và Grover [5] để giải bài toán logarit rời rạc. Vấn đề hiện nay là để cài đặt chạy được các thuật toán này ta phải xây dựng được một máy tính lượng tử đủ mạnh (có đủ số Qbits nhất định). Cụ thể để cài đặt được thuật toán Shor hay thuật toán Grover ta cần một máy tính lượng tử có khoảng 2000 Qbits.

Ngoài ra, do ưu điểm của các thuật toán lượng tử là vấn đề cải thiện tốc độ so với các thuật toán truyền thống, do đó một số lĩnh vực đang gặp phải khó khăn trong vấn đề tốc độ như trí tuệ nhân tạo,… các nhà nghiên cứu cũng đang trong quá trình xây dựng các thuật toán lượng tử cho các lĩnh vực này.

Như vậy, với tầm quan trọng của vấn đề an toàn bảo mật thông tin cũng như trí tuệ nhân tạo, việc nghiên cứu xây dựng các thuật toán lượng tử cũng như máy tính lượng tử là cần thiết đối với mọi công ty và quốc gia. Ở Việt Nam hiện nay, Ban cơ yếu Chính phủ đang thực hiện rất nhiều đề tài nghiên cứu về lĩnh vực này.

Đóng góp của bài báo: Máy tính lượng tử là hướng nghiên cứu được các nhà nghiên cứu và các quốc gia quan tâm, tuy nhiên do tính chất đặc thù về an toàn bảo mật thông tin, tài liệu về lĩnh vực này chưa được các quốc gia cho phép chia sẻ rộng rãi như trong các hướng nghiên cứu khác. Trong bài báo này đóng góp của chúng tôi là dựa trên các nguồn tài liệu hiện có [2][3][4][5][6] trình bày lại một cách có hệ thống kiến trúc (các thành phần) và cơ chế hoạt động của một máy tính lượng tử nói chung, các khó khăn về mặt kỹ thuật hiện nay khi xây dựng máy tính lượng tử, đồng thời đưa ra dự báo chung của các nhà nghiên cứu về hướng phát triển trong tương lai của lĩnh vực này.

2. KIẾN TRÚC VÀ CƠ CHẾ HOẠT ĐỘNG CỦA MỘT MÁY TÍNH LƯỢNG TỬ

Cũng như máy tính thông thường, máy tính lượng tử (Quantumn Computer - QC) cũng được xây dựng nên từ sự kết hợp phần cứng (hardware) và phần mềm (software).

Với hardware máy tính lượng tử bao gồm các thành phần sau:

Host Processor: thành phần đầu tiên là host processor, bản chất là một máy tính thông thường chạy các phần mềm hỗ trợ cho máy tính lượng tử như môi trường, thư viện hỗ trợ và ngôn ngữ lập trình cho máy tính lượng tử, để từ đó các lập trình viên có thể lập trình các thuật toán lượng tử. Trình biên dịch (compiler) sau đó biên dịch chương trình ra các chỉ thị lệnh, các chỉ thị lệnh này được gửi đến thành phần Control processor để thực hiện trên QC. QC sau khi thực hiện xong gửi kết quả trả về cho host processor, và host processor hiển thị kết quả lại cho người dùng. Như vậy người dùng hoàn toàn trong suốt với QC và chỉ tương tác trực tiếp với máy tính thông thường.

Host processor, cũng đóng vai trò kết nối mạng, lưu trữ dữ liệu (QC chỉ đóng vai trò tính toán, còn lưu trữ dữ liệu được lưu trên máy tính thông thường là host processor), chạy các phần mềm hỗ trợ khác như mô phỏng (simulation), kiểm tra kết quả (verification) hay phát hiện lỗi (debugging).

Control processor: thành phần này tiếp nhận kết quả biên dịch từ compiler, từ đó xác định và ra lệnh thực hiện một chuỗi các phép biến đổi, đo (operations, measurements) trên Qbits dựa trên thuật toán lượng tử đã được compiler biên dịch. Các chỉ thị lệnh này sẽ được gửi đến thành phần Control and Measurement Plane. Sau khi nhận được kết quả trả về từ Control and Measurement Plane nó chuyển tiếp kết quả đến cho thành phần Host processor để hiển thị cho người dùng.

Control and Measurement Plane: thành phần này đóng vai trò chuyển đổi các chỉ thị lệnh của control processor từ tín hiệu số (digital signal) sang tín hiệu tương tự (analog control signal) để từ đó thành phần tiếp theo Quantum Data Plane thực hiện các phép biến đổi, đo (operations, measurements) trên Qbits; Đồng thời cũng chuyển đổi kết quả đầu ra nhận được từ Quantum Data Plane từ tín hiệu tương tự sang tín hiệu số để trả về cho control processor.

Quantum Data Plane: đây là thành phần đóng vai trò là trái tim của QC, là nơi chứa các Qbits và các cấu trúc để các Qbits này làm việc với nhau. Nó cũng bao gồm các cơ chế để có thể thực hiện các phép biến đổi và đo trên các Qbits. Thành phần này nhận lệnh từ Control and Measurement Plane để thực hiện các phép biến đổi, đo, sau đó gửi trả kết quả về cho Control and Measurement Plane.

Qbit: Qbit được xây dựng dựa trên Cbit, trong đó Cbit có hai trạng thái là $|0\rangle$ và $|1\rangle$. Biểu diễn cụ thể dưới dạng ma trận:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \text{ và } |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

Với Qbit thì tại một thời điểm mỗi Qbit có thể tồn tại ở một trong hai trạng thái là trạng thái tổng quát hoặc trạng thái cụ thể. Trạng thái cụ thể của Qbit chính là một trong hai trạng thái $|0\rangle$ và $|1\rangle$ tương tự như Cbit. Trạng thái tổng quát của Qbit có dạng:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

Trong đó $|\alpha_0|^2 + |\alpha_1|^2 = 1$, α_0 và α_1 là các số phức. Khi thực hiện phép đo trên Qbit ở trạng thái tổng quát sẽ chuyển Qbit ở trạng thái tổng quát về một trong hai trạng thái cụ thể là $|0\rangle$ (với xác xuất $|\alpha_0|^2$) hoặc $|1\rangle$ (với xác xuất $|\alpha_1|^2$). Lưu ý rằng sau khi đã thực hiện phép đo thì ta không thể biến đổi ngược một Qbit từ trạng thái cụ thể về trạng thái tổng quát ban đầu trước khi đo. Ta chỉ có thể xây dựng ngược lại các Qbit ở trạng thái tổng quát với các giá trị ngẫu nhiên α_0 và α_1 khác. Do vậy khi xây dựng thuật toán lượng tử ta cần rất cân nhắc khi quyết định thực hiện phép đo đối với những Qbits nào.

Các Qbit có thể được biểu diễn (tồn tại) riêng hoặc kết hợp nhiều Qbits với nhau.

Ví dụ với 2 Qbits (tương tự cho nhiều Qbits cùng lúc) được biểu diễn (tồn tại) cùng nhau:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

Trong đó $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$ và α_{00}, α_{01} α_{10}, α_{11} là các số phức. $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ là 2 Cbits được biểu diễn cùng nhau.

Các phép biến đổi trên Qbits (biến đổi trên một hay nhiều Qbits cùng lúc) là các phép biến đổi sao cho thay đổi các giá trị α (theo mục đích của thuật toán) nhưng vẫn đảm bảo tổng bình phương bằng 1. Bản chất của các thuật toán lượng tử là việc đi biến đổi các giá trị α đến một mức độ nào đó, sau đó thực hiện phép đo trên các Qbits để thu về kết quả mong muốn.

Với software của máy lượng tử bao gồm các thành phần sau:

Hệ điều hành thông thường để chạy trên host processor.

Ngôn ngữ lập trình, trình biên dịch, thư viện hỗ trợ máy tính lượng tử. Một vài ngôn ngữ lập trình cho máy tính lượng tử có thể kể đến như Q#, Quipper, Quafl và Liquid. Cũng như máy tính thông thường, ngôn ngữ lập trình cho máy tính lượng tử có hai mức độ, bậc cao thân thiện với người dùng (ví dụ như C, java,... trên máy tính thông thường), và bậc thấp gần với hardware (ví dụ như assembly trên máy tính thông thường).

Phần mềm chạy mô phỏng (simulation), dùng để mô phỏng tính đúng đắn của hardware, thuật toán,... trước khi chương trình được chạy cụ thể trực tiếp trên QC. Simulation được dùng để dự đoán các lỗi có thể phát sinh, đặc biệt do các phép biến đổi trên QC luôn kèm theo lỗi phát sinh do tác động của môi trường.

Phần mềm kiểm tra kết quả (verification) và phát hiện lỗi (debugging), dùng để kiểm tra kết quả chạy của QC có chính xác không, nếu có lỗi thì debugging kiểm tra xem lỗi xảy ra ở đâu. Đây là các phần mềm thiết yếu cho bất kỳ một loại máy tính và ngôn ngữ lập trình nào.

Như vậy hoạt động chung của một QC như sau:

Xác định vấn đề cần giải quyết. Ví dụ bài toán phân tích ra thừa số nguyên tố;

Chọn thuật toán lượng tử để giải bài toán. Ví dụ thuật toán Shor;

Chọn ngôn ngữ lập trình lượng tử (ví dụ Q#), lập trình viết thuật toán trên host processor (máy tính thông thường), biên dịch chương trình và nhận lại kết quả mong muốn;

Ngoài ra lập trình viên có thể dùng thêm các phần mềm hỗ trợ khác như simulation, verification, debugging để đạt được hiệu quả, độ chính xác như mong muốn.

3. VẤN ĐỀ KHÓ KHĂN HIỆN NAY KHI XÂY DỰNG MÁY TÍNH LƯỢNG TỬ

Xây dựng được máy tính lượng tử như trên hiện nay đang gặp những khó khăn sau:

Xây dựng thành phần Quantum Data Plane với số Qbits lớn đang còn gặp khó khăn. Do có sự ảnh hưởng của môi trường nên trong quá trình biến đổi (operation) các Qbits sẽ phát sinh lỗi, cuối cùng sẽ ảnh hưởng đến kết quả đầu ra là không chính xác. Số lượng Qbits càng lớn, các phép biến đổi càng nhiều thì tỉ lệ lỗi càng cao. Cụ thể với hai công nghệ tốt nhất hiện nay là *Trapped Ion* và *Superconducting* người ta đã tạo ra được QC với số bits là 10 đến 50 Qbits. Tuy nhiên tỉ lệ lỗi phát sinh cho mỗi phép biến đổi (trên 1 Qbit hay 2 Qbits) là vài phần trăm (tương đương 10^{-2}), tức là cứ khoảng một trăm phép biến đổi thì có thể cho kết quả sai vài lần. Như vậy nếu giải thuật bao gồm 100 phép biến đổi thì kết quả cuối cùng có khả năng cao là sai. Hiện nay một số công ty (như IBM) tuyên bố đã tạo ra được QC với khoảng hơn 100 Qbits, tuy nhiên họ vẫn chưa công bố cụ thể tỷ lệ lỗi cho mỗi phép biến đổi là bao nhiêu.

Ngoài ra với hai công nghệ Trapped Io và Superconducting kể trên người ta hoàn toàn có khả năng trong tương lai gần tạo ra được mỗi IC (intergrated circuit) với hàng trăm Qbits, tuy nhiên với càng nhiều Qbits thì sự tương tác với môi trường càng cao và tỉ lệ lỗi sẽ càng cao. Khó khăn hiện nay là làm sao tạo ra được nhiều Qbits nhưng tỉ lệ lỗi phát sinh ở mỗi phép biến đổi phải thấp hoặc ít nhất là giữ nguyên được ở mức tương đương 10^{-2} ở trên. Để tăng lên số lượng hàng ngàn Qbits, ta cần phải thiết kế được các cấu trúc Qbits hoàn toàn

mới khác cho hai công nghệ Trapped Ion và Superconducting, hoặc kết hợp lai tạo giữa hai công nghệ này với các công nghệ khác hiện có. Hướng này vẫn đang là thách thức được các nhà nghiên cứu tìm cách giải quyết. Một hướng khác là xây dựng cấu trúc theo kiểu modul hóa, tức là kết hợp nhiều IC lại với nhau, mỗi IC chứa một số lượng nhỏ các Qbits được xây dựng bằng các công nghệ hiện có. Tuy nhiên để các IC này kết nối được với nhau thì cần phải có sự kết nối lượng tử (quantumn interconnection), sự kết nối lượng tử này do tác động với môi trường nên lại phát sinh lỗi. Hướng nghiên cứu để giảm lỗi cho quantumn interconnection cũng đang là hướng nghiên cứu được quan tâm hiện nay.

Để giải quyết vấn đề lỗi phát sinh, giống như máy tính truyền thống, ta dùng kỹ thuật mã sửa sai (quantumn error corection - QEC). Với kỹ thuật QEC mới nhất hiện nay áp dụng vào QC có tỉ lệ lỗi sai ở mỗi bước biến đổi là tương đương 10^{-3} (tốt hơn các QC với 10 đến 50 Qbits hiện nay khoảng 10 lần) thì có thể giảm được tỉ lệ lỗi sai ở mỗi bước biến đổi xuống còn tương đương 10^{-10}. Tuy nhiên, nhược điểm là cần 16 Qbits phụ trợ, tức là ta phải có 17 Qbits vật lý để có thể có được 1 Qbit thực tế có thể dùng được (logical Qbit). Như vậy với QC ở trên cho dù tỉ lệ lỗi sai có được giảm đi 10 lần thì thực tế nếu áp dụng QEC ta chỉ thu được QC với khoảng 1 Qbit. Lưu ý rằng nếu tỉ lệ lỗi sai ở mỗi bước biến đổi xuống còn tương đương 10^{-4} thì khi áp dụng QEC như ở trên thì có thể giảm được tỉ lệ lỗi sai ở mỗi bước biến đổi xuống còn tương đương 10^{-18}. Như vậy, mấu chốt để có thể xây dựng được QC hiện nay là tạo ra được các công nghệ mới tốt hơn Trapped Ion và Superconducting để có thể giảm được sự ảnh hưởng của môi trường để tỉ lệ lỗi sai ở mỗi bước biến đổi là bé nhất. Song song với đó là cải tiến các thuật toán QEC để giảm số lượng các Qbits phụ trợ cần thiết.

Khó khăn trong xây dựng phần mềm mô phỏng giải thuật Simulation. Để QC có thể dùng được trong thực tế, simulation là công cụ không thể thiếu. Các phép biến đổi hiện nay ở mỗi Qbit đều là các phép tích matrix và vector, như vậy simulation cho một phép biến đổi trên mỗi Qbit dùng máy tính thông thường là một phép tích matrix và vector. Tuy nhiên, với *N* Qbits kết hợp thì số lượng phép biến đổi sẽ là hàm mũ 2^N, như vậy dù cho có dùng máy tính thông thường với tốc độ mạnh nhất cũng không thể simulation được. Hiện nay, người ta mới chỉ đang dùng giải pháp là mô phỏng từng phần trong giải thuật.

Vấn đề phát hiện lỗi (debugging) cũng là một thách thức lớn. Để phát hiện lỗi như trong máy tính thông thường, ta phải kiểm tra kết quả của từng bước trung gian, tuy nhiên đây là điều không thể đối với QC, vì khi thực hiện phép đo trên Qbits thì ta sẽ mất toàn bộ trạng thái tổng quát của Qbits.

Vấn đề tiếp theo là tìm ra giải thuật để có thể nạp dữ liệu một cách hiệu quả vào QC. Với một thuật toán cụ thể ta cần dữ liệu đầu vào (có thể rất lớn), dữ liệu đầu vào này được lưu ở host processor (máy tính thông thường), để chuyển đổi dữ liệu này vào các Qbits, tức là xây dựng được trạng thái của các Qbits tương ứng với dữ liệu đầu vào này đang còn là vấn đề. Hiện chưa có thuật toán hiệu quả nào để giải quyết vấn đề này, thời gian để xây dựng được trạng thái của các Qbits tương ứng với dữ liệu đầu vào đang còn khá lớn làm giảm đi ưu thế về tốc độ của QC.

Với các vấn đề khác như ngôn ngữ lập trình cho QC, compiler, verification,... không có các trở ngại nào quá lớn tuy nhiên cũng rất cần được nghiên cứu cải thiện.

4. KẾT QUẢ HIỆN NAY VÀ DỰ ĐOÁN

Hiện nay đang phân ra làm ba loại QC

QC không dựa trên phép biến đổi trên gate (phép biến đổi Qbit dựa trên cổng transitor) gọi là Analog QC, loại này ít được quan tâm vì không có khả năng áp dụng QEC để sửa lỗi. Toàn bộ máy tính thông thường ngày nay đều là máy tính dựa trên phép biến đổi trên gate, lý do là chỉ có loại này mới có khả năng kháng lỗi. Với QC ta vẫn quan tâm đến Analog QC là do việc sản xuất QC có khả năng kháng lỗi đang còn gặp rất nhiều khó khăn nên ta vẫn cần loại Analog QC với mục đích thử nghiệm.

QC dựa trên phép biến đổi trên gate nhưng không áp dụng QEC, gọi là *noisy intermediate-scale quantum - NISQ* QC. Như vậy với NISQ QC, số phép biến đổi của mỗi thuật toán và số Qbits phải bé để sao cho tỉ lệ lỗi là có thể chấp nhận được. Người ta quan tâm đến loại này vì khó khăn trong việc xây dựng QC kháng lỗi.

QC dựa trên phép biến đổi trên gate và áp dụng QEC. Đây là loại máy tính mong muốn, là đích đến cuối cùng để dùng được trong thực tế.

Công nghệ hiện nay đã đạt được (với hai công nghệ Trapped Ion và Superconducting):

Analog QC với khoảng gần 1000 Qbits.

NISQ QC với từ 10 đến 50 Qbits và với tỉ lệ lỗi ở mỗi phép biến đổi là tương đương 10^{-2}. Một vài công ty công bố họ đã sản xuất được QC với 100 Qbits, tuy nhiên họ vẫn chưa công bố tỉ lệ lỗi là bao nhiêu. Như vậy khi áp dụng QEC với công nghệ hiện tại, thực tế ta mới chỉ sản xuất được QC với khoảng 1 đến 10 Qbits thực tế (logical Qbit), tức là chỉ mới giải quyết được bài toán phân tích ra thừa số nguyên tố với giá trị $N = p*q$ rất bé.

Dự đoán

Người ta dự đoán rằng với sự phát triển của công nghệ Trapped Ion và Superconducting hiện có, đầu những năm 2025 sẽ có thể tạo ra được NISQ QC với khoảng vài trăm Qbits và có tỉ lệ lỗi ở mỗi phép biến đổi là khoảng $0.5 * 10^{-2}$. Tức là nếu mọi thứ thuận lợi đầu những năm 2025 sẽ có QC kháng lỗi với khoảng vài chục Qbits. Xa hơn thì chưa thể dự đoán được. Lưu ý rằng để chạy được thuật toán Shor hay Grover yêu cầu phải có QC kháng lỗi cỡ 2000 Qbits.

Những công nghệ cần đột phá để có thể xây dựng được QC như mong muốn:

Một loại công nghệ hoàn toàn mới (phát triển từ Trapped Ion hay Superconducting, hay kết hợp với các công nghệ khác) để có thể sản xuất được nhiều Qbits hơn và tỉ lệ lỗi ở mỗi phép biến đổi trên Qbits phải nhỏ hơn10^{-3}. Có thể là xây dựng IC chứa được nhiều Qbits hơn, hoặc công nghệ cho phép kết hợp hiệu quả các IC lại với nhau.

Một công nghệ mới cho phép vừa biến đổi, vừa đo (tức là đo mà không làm mất trạng thái tổng quát) để giải quyết vấn đề debugging.

Phương pháp mới hiệu quả để giải quyết vấn đề simulation và nạp dữ liệu đầu vào.

Cải tiến các giải thuật lượng tử hiện có như Shor, Grover sao cho khi cài đặt cần ít Qbits hơn, cải tiến QEC sao cho để tạo ra một logical Qbits cần ít physical Qbits hơn, cải tiến các vấn đề về Compiler, Ngôn ngữ lập trình, Verification,...

Các yếu tố ảnh hưởng chính

Trong tương lai gần có sản xuất được thành công một QC nào đó có khả năng thương mại hóa hay không? tức là có sản xuất được một QC nào thực sự có ý nghĩa, giải quyết được

một bài toán cụ thể nào đó tốt hơn so với máy tính thông thường hay không? điều này rất quan trọng vì theo định luật Moor đây là yếu tố then chốt để có tài chính tái đầu tư tiếp cho việc nghiên cứu. Ở đây người ta chỉ hi vọng đến việc sản xuất được NISQ QC có ý nghĩa, chưa phải là QC có khả năng kháng lỗi. Tức là song song với việc cải tiến công nghệ sản xuất, các nhà phát triển thuật toán cũng cần phát triển các giải thuật lượng tử giải quyết các bài toán có ý nghĩa trong thực tế, đồng thời các bước trong thuật toán cũng phải đủ đơn giản. Hiện nay, các nhà nghiên cứu đang đặt sự quan tâm vào các giải thuật lượng tử cho lĩnh vực trí tuệ nhân tạo.

Các chính phủ có còn tiếp tục tài trợ tiền cho nghiên cứu QC nữa hay không?

Theo dự đoán trong báo cáo của Bộ quốc phòng Mỹ [6] ít nhất cho tới 10 năm tới ta chắc chắn chưa thể tạo ra được một QC có khả năng kháng lỗi như mong muốn.

5. KẾT LUẬN

Máy tính lượng tử nếu tồn tại ở mức dùng được sẽ có sức ảnh hưởng rất lớn trong thực tế, đặc biệt trong lĩnh vực an toàn bảo mật thông tin. Rất nhiều vấn đề phức tạp không thể giải quyết được với máy tính thông thường hiện nay đã có thể giải quyết được với máy tính lượng tử. Gần đây các lĩnh vực có độ phức tạp của thuật toán lớn như trí tuệ nhân tạo cũng được các nhà nghiên cứu quan tâm phát triển các thuật toán lượng tử. Trong bài báo này chúng tôi trình bày kiến trúc chung và cách hoạt động của một máy tính lượng tử, các khó khăn về mặt kỹ thuật hiện nay trong xây dựng máy tính lượng tử. Chúng tôi cũng trình bày những kết quả mới nhất hiện nay của các nhà nghiên cứu khi xây dựng máy tính lượng tử, cũng như những dự đoán của các nhà nghiên cứu về lĩnh vực này.

TÀI LIỆU THAM KHẢO

[1] IgnacioVelásquez and AngélicaCaro and AlfonsoRodríguez (2018), Authentication schemes and methods: A systematic literature review, *Information and Software Technology*, Vol.94, 30-37

[2] Edward Gerjuoy (2005), Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers, *American Journal of Physics*, 73, 521-540.

[3] P. S. Bourdon and H. T. Williams (2007), *Sharp probability estimates for shor's order finding algorithm*, http://arxiv.org/abs/quant-ph/0607148.

[4] N. David Mermin (2019), *Quantum computer science: An introduction*, Cambridge University Press.

[5] L.K. Grover (1996), *A fast quantum mechanical algorithm for database search*. Proceedings, 28th Annual ACM Symposium on the Theory of Computing.

[6] The US national academies (2018), *Quantum computing: Progress and prospects*, the national academies press.

GENERAL INTRODUCTION OF QUANTUMN COMPUTER

Trinh Viet Cuong, Le Van Vinh

ABSTRACT

Cryptography is based on hard problems such as prime factorization problem or discrete logarithm problem. Currently, there doesn't exist any efficient algorithm (implemented on binary computer) to solve these hard problems, but we do have quantumn efficient algorithms (implemented on quantumn computer) to solve several aforementioned hard problems. For examples, we have Shor algorithm to efficently solve the prime factorization problem and Grover algorithm to efficently solve the discrete logarithm problem. However, to implement Shor algorithm or Grover algorithm we need to have a quantumn computer with around 2000 Qbits. Unfortunatelly, we currently can only build a quantumn computer with around 50 Qbits.

In this paper, we first present the architecture of a quantumn computer and how a quantumn computer works. We then discuss some remaining technical problems and the latest researching results of building a practical quantumn computer. Finally, we present the predictions of researchers on this direction of research.

Keywords: *Quantumn computer, Qbits, Shor algorithm, Grover algorithm.*

** Ngày nộp bài:3/6/2022; Ngày gửi phản biện: 3/6/2022; Ngày duyệt đăng: 27/10/2022*