

การศึกษาและเปรียบเทียบแพลตฟอร์มการจำลองประมวลผลควอนตัม

A STUDY AND COMPARISON OF QUANTUM COMPUTING SIMULATION PLATFORMS

เชียร สืบวงษ์¹

ดร.ชัยพร เชมะภาคะพันธ์²

บทคัดย่อ

ปัจจุบันมีความก้าวหน้าทางเทคโนโลยีมากขึ้นหนึ่งในนั้นคือเทคโนโลยีควอนตัมคอมพิวเตอร์ซึ่งอาจเป็นหนึ่งในเทคโนโลยีสำคัญในอนาคต ทำให้มีการพัฒนาแพลตฟอร์มสำหรับการประมวลผลควอนตัมหลายแพลตฟอร์ม ทำให้การเลือกใช้งานและศึกษาในแต่ละซอฟต์แวร์ควอนตัมแพลตฟอร์มค่อนข้างยุ่งยาก ทำให้ผู้ใช้งานจำเป็นต้องใช้เวลากับการศึกษาและเลือกใช้แพลตฟอร์มที่เหมาะสมและถูกต้อง

การศึกษานี้ได้คัดเลือกซอฟต์แวร์ควอนตัมแพลตฟอร์มที่น่าสนใจมาทั้งหมด 4 แพลตฟอร์มได้แก่ Forest (pyQuil), Qiskit, ProjectQ และ Quantum Developer Kit (Q#) โดยจะทำการรวบรวมข้อมูลต่างๆ ภาพรวมระบบ การติดตั้ง, การใช้งาน ภาษาและรูปแบบการเขียนโปรแกรม การทดสอบฟังก์ชันและอัลกอริทึมที่สามารถใช้งานอะไรได้บ้าง และนำมาเปรียบเทียบภาพรวมการเพื่อคัดเลือก 2 แพลตฟอร์มที่เหมาะสมกับการใช้งาน โดยขั้นตอนสุดท้ายจะนำ 2 แพลตฟอร์มที่เลือกเอาไว้มาเปรียบเทียบประสิทธิภาพการทำงานแบบจำลองโดยใช้ Shor's algorithm

การทดสอบการใช้งานและฟังก์ชันต่างๆของทั้ง 4 แพลตฟอร์มจะเห็นได้ว่ามีเพียง 2 แพลตฟอร์มที่สามารถใช้งานฟังก์ชันได้อย่างครบถ้วนและสามารถเชื่อมต่อกับควอนตัมคอมพิวเตอร์จริงได้อันได้แก่ ProjectQ และ Qiskit จึงเป็นตัวเลือกในการนำมาจำลองการทำงานเพื่อวัดประสิทธิภาพประมวลผลโดยใช้ขั้นตอนวิธี Shor's algorithm จะเห็นได้ว่าเวลาการทำงานของ QisKit ใช้เวลาที่เสถียรและน้อยกว่าหากเทียบกับ ProjectQ ที่มีเวลาการทำงานที่นานขึ้นเรื่อยๆ ถ้าผลลัพธ์มีจำนวนมากขึ้น เป็นการสรุปได้ว่าแพลตฟอร์ม QisKit มีฟังก์ชันและคุณสมบัติโดยรวมค่อนข้างครบถ้วนมากกว่า รวมทั้งสามารถจำลองได้อย่างมีประสิทธิภาพเร็วกว่าและมีความถูกต้องมากกว่าเมื่อเทียบกับแพลตฟอร์มอื่น ๆ

¹นักศึกษาหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาสารนิพนธ์หลัก

คำสำคัญ: ซอฟต์แวร์ควอนตัมแพลตฟอร์ม, ฟังก์ชัน, อัลกอริทึม, pyQuil, QisKit , ProjectQ, Q#,
Shor's algorithm

ABSTRACT

Nowadays, with more technological advances, quantum computing technology is one of the key technologies in the future. There are several simulation platforms for quantum computing. This makes the selection and study of each quantum platform quite difficult. Thus, users who need to use the platform to study and research have to take times and select the platform which is appropriate and accurate.

The study selected four interesting quantum software platforms: Forest (pyQuil), Qiskit, ProjectQ and Quantum Developer Kit (Q#). Use programming languages and formats. Testing what functions and algorithms can be used and comparing the overall picture to select 2 platforms that are suitable for use The final step was to compare the performance of the two selected platforms using Shor's algorithm.

Testing the functionality and functions of the four platforms shows that only two are fully functional and can be connected to a real quantum computer: ProjectQ. And Qiskit is the optional simulation to measure performance using Shor's algorithm. QisKit's runtime is less stable compared to ProjectQ's higher uptime compared to ProjectQ. Keep getting longer If the result is more In conclusion, the QisKit platform has more complete functionality and overall features. Plus, it can be simulated more efficiently, faster, and more accurately compared to other platforms.

Keywords: Quantum Platform Software, Function, Algorithms, pyQuil, QisKit , ProjectQ, Q#,

Shor's algorithm

บทนำ

ในยุคสมัยที่มีการพัฒนาเทคโนโลยีอย่างรวดเร็วในปัจจุบันและพัฒนาซอฟต์แวร์แข่งขันอย่างแพร่หลายเป็นจำนวนมาก การพัฒนาซอฟต์แวร์ไม่ว่าจะเป็นเพื่อใช้งานในภายในองค์กรแต่ละ

องค์กรที่มีธุรกิจที่แตกต่างกันนั้นทำให้มีการแข่งขันการพัฒนาที่ค่อนข้างสูงเพื่อให้ได้ซอฟต์แวร์ที่มีประสิทธิภาพและทำประโยชน์สูงสุดให้แก่องค์กรในแต่ละธุรกิจ ตัวนักพัฒนาซอฟต์แวร์เองก็ต้องพัฒนาความสามารถในการเขียนโปรแกรมแต่ละภาษาเพื่อเท่าทันเทคโนโลยีใหม่ๆที่มีการปรับเปลี่ยนอยู่ตลอดเวลา

เมื่อแต่ละองค์กรต้องการประสิทธิภาพและความเร็วในการทำงานของซอฟต์แวร์ เพื่อแก้ไขปัญหาเฉพาะเจาะจงและมีความซับซ้อนของแต่ละองค์กร ตัวนักพัฒนาซอฟต์แวร์เองก็ต้องมองหาเทคโนโลยีที่ทันสมัยและมีประสิทธิภาพ คู่ควรกับเวลาที่ทำการศึกษา และนำความรู้มาประยุกต์ใช้ได้อย่างรวดเร็ว เทคโนโลยีควอนตัมจึงเป็นอีกทางเลือกขององค์กรที่ต้องการพัฒนาธุรกิจและเป็นหลักสำคัญที่นักพัฒนาซอฟต์แวร์ควรให้ความสำคัญแก่เทคโนโลยีสมัยใหม่ซึ่งนักพัฒนาซอฟต์แวร์ซอฟต์แวร์จะได้พัฒนาซอฟต์แวร์โดยใช้เทคโนโลยีนี้อย่างแน่นอนในอนาคต เทคโนโลยีควอนตัมจะมีบทบาทความสามารถในการแก้ไขปัญหาได้รวดเร็วกว่าคอมพิวเตอร์ทั่วไป ซึ่งถือเป็นวิวัฒนาการคอมพิวเตอร์แบบก้าวกระโดด และในอนาคตมี Quantum Software Platform ให้เลือกใช้งานเป็นจำนวนมากอีกทั้งแต่ละ Quantum Software Platform ยังมี Framework หลายตัวในหนึ่งค่ายอีกด้วย

ดังนั้นในการเลือกใช้งาน Quantum Software Platform เป็นสิ่งสำคัญมากในการเลือกใช้งานพัฒนาซอฟต์แวร์ว่ามีค่าใช้จ่ายอะไร การติดตั้ง Framework มี Algorithm อะไรที่รองรับแต่ละ Framework บ้าง ภาษาที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ความรู้เดิมในการพัฒนาได้หรือไม่ เพื่อจะลดระยะเวลาในการเลือกใช้งาน รวมถึงเวลาในการศึกษาของตัวนักพัฒนาซอฟต์แวร์ด้วย

วัตถุประสงค์การวิจัย

เพื่อเปรียบเทียบการใช้งานและภาพรวมของซอฟต์แวร์ควอนตัมแพลตฟอร์มทั้ง 4 แพลตฟอร์ม ได้แก่ Forest (pyQuil), Qiskit, ProjectQ และ Quantum Developer Kit (Q#) เพื่อหาแพลตฟอร์มที่เหมาะสมสำหรับการศึกษาและพัฒนา

ขอบเขตการวิจัย

1. ทดสอบการใช้งานและภาพรวมของซอฟต์แวร์ควอนตัมแพลตฟอร์มทั้ง 4 แพลตฟอร์ม ได้แก่ Forest (pyQuil), Qiskit, ProjectQ และ Quantum Developer Kit (Q#)

2. ทดสอบประสิทธิภาพการทำงานแบบจำลองโดยใช้ Shor's Algorithm โดยเลือก 2 แพลตฟอร์มที่ดีที่สุดจาก 4 แพลตฟอร์มที่กล่าวมา

แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1. เทคโนโลยีควอนตัมคอมพิวเตอร์ (Quantum Computer)

เมื่อเทคโนโลยีการพัฒนา CPU เกือบจะเดินมาถึงทางตัน ทั้งขนาดทรานซิสเตอร์ที่ปัจจุบันอยู่ที่ 7 นาโนเมตร และการที่จะพัฒนาให้เล็กกว่านี้เป็นเรื่องที่แทบจะเป็นไปไม่ได้ยาก วิทยาศาสตร์แขนงใหม่จึงเป็นหนทางในการพัฒนาคอมพิวเตอร์ใน Generation ใหม่ ซึ่งหนึ่งในนั้นคือ Quantum Computer เทคโนโลยี Quantum เกี่ยวข้องกับ อนุภาคขนาดเล็กมากๆ ที่มีคุณสมบัติสองอย่างคือ Superposition (สภาวะทับซ้อน) และ Entanglement (การพัวพันเชิงควอนตัม)

2. อัลกอริทึมของชอร์ (Shor's algorithm)

ตั้งชื่อตามนักคณิตศาสตร์ชื่อปีเตอร์ ชอร์ที่คิดขึ้นในปีค.ศ.1994 โดยขั้นตอนวิธีนี้เป็นขั้นตอนวิธีควอนตัม (ขั้นตอนวิธีที่ทำงานบนควอนตัมคอมพิวเตอร์) ที่ใช้ในการแยกตัวประกอบของจำนวนเต็ม ซึ่งโดยทั่วไปแล้วใช้ในการแก้ปัญหา: ให้จำนวนเต็ม N แล้วให้หาตัวประกอบเฉพาะของ N ในควอนตัมคอมพิวเตอร์นั้น การแยกตัวประกอบด้วยขั้นตอนวิธีของชอร์จะใช้เวลาในการทำงานไม่เกินฟังก์ชันพหุนาม (Polynomial) ของขนาดข้อมูล โดยจะใช้เวลาเป็น $O((\log N)^3)$ ซึ่งแสดงให้เห็นว่าเป็นการแก้ปัญหาการแยกตัวประกอบของจำนวนเต็มที่มีประสิทธิภาพในควอนตัมคอมพิวเตอร์ วิธีนี้จัดเป็นวิธีที่เร็วกว่าหลาย ๆ วิธีที่มีประสิทธิภาพที่รู้จักกันทั่ว ๆ ไปที่มักจะใช้เวลาเป็นฟังก์ชันเลขชี้กำลัง (Exponential) ขนาดของข้อมูล

งานวิจัยที่เกี่ยวข้อง

Ryan LaRose กล่าวไว้ว่าคอมพิวเตอร์ควอนตัมพร้อมใช้งานบนระบบคลาวด์ แต่การขยายตัวของแพลตฟอร์มซอฟต์แวร์ควอนตัมเมื่อเร็ว ๆ นี้อาจทำให้ผู้ตัดสินใจว่าจะใช้อะไรอย่างสิ้นหลาม ในบทความนี้ เรานำเสนอภาพปัจจุบันของภูมิทัศน์คอมพิวเตอร์ควอนตัมที่พัฒนาอย่างรวดเร็วโดยเปรียบเทียบ ที่ทำให้นักวิจัยสามารถ ใช้อุปกรณ์

ควอนตัมจริงและจำลอง การวิเคราะห์ของเราครอบคลุมข้อกำหนดและการติดตั้ง
ไวยากรณ์ภาษาผ่านโปรแกรมตัวอย่าง การสนับสนุนไลบรารี และความสามารถจำลอง
ควอนตัมสำหรับแต่ละแพลตฟอร์ม สำหรับแพลตฟอร์มที่รองรับคอมพิวเตอร์ควอนตัม เรา
จะเปรียบเทียบฮาร์ดแวร์ ภาษาแอสเซมบลีควอนตัม และคอมไพเลอร์ควอนตัม เราสรุป
โดยครอบคลุมคุณสมบัติของแต่ละรายการและกล่าวถึงแพ็คเกจซอฟต์แวร์คอมพิวเตอร์
ควอนตัมอื่นๆ โดยสังเขป [1]

Damian S. Steiger, Thomas Häner และ Matthias Troyer กล่าวไว้ว่า ProjectQ
ซึ่งเป็นความพยายามของซอฟต์แวร์โอเพ่นซอร์สสำหรับการคำนวณควอนตัม รุ่นแรกมี
เฟรมเวิร์กของคอมไพเลอร์ที่สามารถกำหนดเป้าหมายฮาร์ดแวร์ประเภทต่างๆ ได้ เครื่อง
จำลองประสิทธิภาพสูงพร้อมความสามารถในการจำลอง และปลั๊กอินคอมไพเลอร์สำหรับ
การวาดวงจรและการประเมินทรัพยากร เราแนะนำภาษาเฉพาะโดเมนที่ฝังด้วย Python
นำเสนอคุณสมบัติและจัดเตรียมตัวอย่างการใช้งานสำหรับอัลกอริทึมควอนตัม กรอบงาน
อนุญาตให้ทดสอบอัลกอริทึมควอนตัมผ่านการจำลองและเปิดใช้งานบนฮาร์ดแวร์
ควอนตัมจริงโดยใช้แบ็คเอนด์ที่เชื่อมต่อกับบริการคลาวด์ IBM Quantum Experience ด้วย
กลไกการขยาย ผู้ใช้สามารถจัดเตรียมแบ็คเอนด์ให้กับฮาร์ดแวร์ควอนตัมเพิ่มเติม และ
นักวิทยาศาสตร์ที่ทำงานเกี่ยวกับการคอมไพล์ควอนตัมสามารถจัดหาปลั๊กอินสำหรับการ
คอมไพล์ การเพิ่มประสิทธิภาพ การสังเคราะห์เกต และกลยุทธ์การจัดวาง [2]

Thomas Häner, Damian S Steiger, Krysta Svore and Matthias Troyer ได้กล่าวไว้
ว่าระเบียบวิธีซอฟต์แวร์สำหรับการรวบรวมโปรแกรมควอนตัมคอมพิวเตอร์ควอนตัม
สำคัญว่าจะเปลี่ยนแนวคิดเรื่องการคำนวณของเราโดยเสนอกระบวนทัศน์ใหม่ทั้งหมด
เพื่อให้ได้การคำนวณควอนตัมที่ปรับขนาดได้ การเพิ่มประสิทธิภาพคอมไพเลอร์และ
ขั้นตอนการออกแบบซอฟต์แวร์ที่เกี่ยวข้องจะมีความจำเป็น เรานำเสนอสถาปัตยกรรม
ซอฟต์แวร์สำหรับการคอมไพล์โปรแกรมควอนตัมจากโปรแกรมภาษาระดับสูงไปจนถึง
คำแนะนำเฉพาะฮาร์ดแวร์ เราอธิบายเลเยอร์ที่จำเป็นของนามธรรมและความแตกต่างและ
ความคล้ายคลึงกันกับเลเยอร์คลาสสิกของโฟลว์การออกแบบโดยใช้คอมพิวเตอร์ช่วย
สำหรับแต่ละเลเยอร์ของสแต็ก เราจะพูดถึงวิธีการพื้นฐานสำหรับการคอมไพล์และการ
ปรับให้เหมาะสม ระเบียบวิธีซอฟต์แวร์ของเราอำนวยความสะดวกในการสร้างสรรค์
นวัตกรรมที่รวดเร็วยิ่งขึ้นในหมู่นักออกแบบอัลกอริทึมควอนตัม วิศวกรฮาร์ดแวร์

ควอนตัม และนักทดลอง ช่วยให้สามารถคอมไพล์อัลกอริธึมควอนตัมที่ซับซ้อนที่ปรับขนาดได้ และสามารถกำหนดเป้าหมายไปยังการใช้งานฮาร์ดแวร์ควอนตัมเฉพาะใดๆ [3]

ระเบียบวิธี

การออกแบบที่ใช้ในการทดลองครั้งนี้ประกอบด้วย เก็บรวบรวมข้อมูล Quantum Software Platforms ทั้ง 4 ตัว, แสดงการติดตั้งโปรแกรมและสิ่งที่ Platforms ทั้ง 4 ตัวต้องการ, ทดสอบ Algorithm ที่รองรับแต่ละ Platforms, แสดงภาพรวมทั้ง 4 Platforms, นำ 2 Platforms ที่ดีที่สุดมาวัดประสิทธิภาพการทำงานแบบจำลอง โดยการใช้ Shor's Algorithm

ผลการทดลอง

ผลการทดสอบการเชื่อมต่อ

ในการทดลองเป็นการทดลองเพื่อทดสอบว่าแพลตฟอร์มที่ 4 แพลตฟอร์ม มีการเชื่อมต่อไปยังคอมพิวเตอร์ตั้งและควอนตัมคอมพิวเตอร์จริงแบบใด และตรวจสอบว่าแพลตฟอร์มใดไม่มีควอนตัมคอมพิวเตอร์หรือเสียค่าใช้จ่ายอะไรในการใช้งาน

ผลการทดสอบการติดตั้ง ไวยากรณ์ และภาษาที่รองรับ

รายละเอียด	ซอฟต์แวร์ควอนตัมแพลตฟอร์ม			
Institution	Rigetti	IBM	ETH Zurich	Microsoft
Version	Version 1.1.0	Version 0.36.1	Version 0.7.3	Version 0.24.200024
Open Source	True	True	True	True
OS	Mac, Windows, Linux	Mac, Windows, Linux	Mac, Windows, Linux	Mac, Windows, Linux
Requirements	Python 3, Anaconda	Python 3.7+, Qiskit , Jupyter, Anaconda 3	use pip version v0.1.0+, Python 3.7+	Visual Studio Code
Classical Host Language	Python	Python, JavaScript, Swift	Python	C#
Quantum Program Language	pyQuil	Qiskit	ProjectQ	Q#
Quantum Language	Quil	QASM	-	-
Quantum Hardware		IBMQX4 (5 qubits)	can connect to IBM QASM	-
Simulator	20 qubits locally, 26 qubits with most API keys to QVM, 30+ w/ private access	25 qubits locally, 30 through cloud	28 qubits locally	30 qubits locally, 40 through Azure cloud

ภาพที่ 1: ภาพรวมผลทดลองรายละเอียดภาพรวม 4 แพลตฟอร์ม

ผลการทดสอบฟังก์ชันและควอนตัมอัลกอริทึมที่ใช้งานได้

ในการทดลองนี้เป็นการทดลองเพื่อทดสอบว่าแพลตฟอร์มที่ 4 แพลตฟอร์มได้แก่ Forest (pyQuil), Qiskit, ProjectQ และ Quantum Developer Kit (Q#) ใช้งานฟังก์ชันและควอนตัมอัลกอริทึมอะไรได้บ้างโดยมีแต่ตัวฟังก์ชันหลักที่ไม่ได้ลงส่วนเสริมใดๆเข้าไป

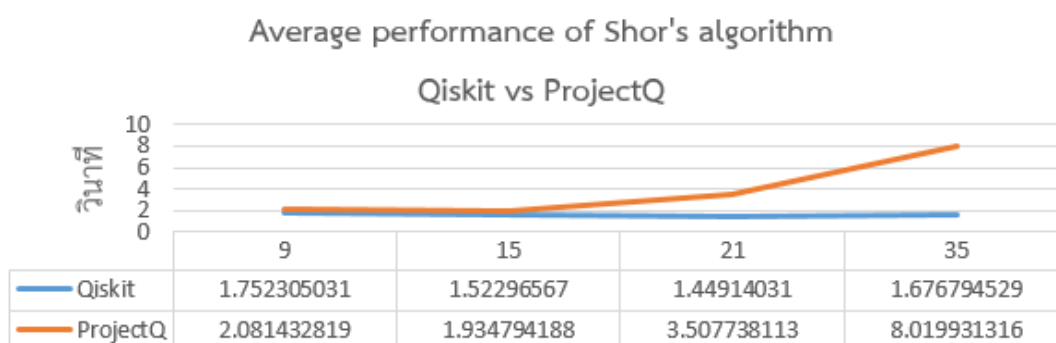
ผลการทดสอบเปรียบเทียบเพื่อคัดเลือก 2 แพลตฟอร์ม

องค์ประกอบ	Qiskit	Qiskit	ProjectQ	Q#
ซอฟต์แวร์ฟรี	✓	✓	✓	✓
มีควอนตัมคอมพิวเตอร์จริง	✓	✓		
สามารถใช้งานควอนตัมคอมพิวเตอร์จริง		✓	✓	
สามารถใช้งานบนคอมพิวเตอร์	✓	✓		✓
การใช้งานคลาวด์คอมพิวเตอร์	✓	✓		
มีเอกสารใช้ในการศึกษาที่ละเอียด		✓	✓	✓
ซอฟต์แวร์มีข้อผิดพลาดน้อย		✓	✓	
สามารถใช้อัลกอริทึมที่สำคัญได้ครบ		✓	✓	
ภาษาที่ใช้ในการพัฒนาเข้าใจง่าย	✓	✓	✓	
การแสดงผลการใช้งานง่าย	✓	✓	✓	

ภาพที่ 2: ผลการทดสอบเปรียบเทียบเพื่อคัดเลือก 2 แพลตฟอร์ม

ผลการทดสอบการเปรียบเทียบ 2 แพลตฟอร์มแบบวัดประสิทธิภาพการทำงานแบบจำลองโดยใช้

Shor's algorithm



ภาพที่ 3: ภาพค่าเฉลี่ยการทดสอบเปรียบเทียบประสิทธิภาพการทำงานแบบจำลอง 2 แพลตฟอร์ม

สรุปผลการวิจัย

การใช้งานฟังก์ชันและอัลกอริทึมของแต่ละแพลตฟอร์มใช้งานได้ดีแต่จะโดดเด่นที่ 2 แพลตฟอร์มคือ Qiskit และ ProjectQ ที่มีอัลกอริทึมเด่นๆและมีชื่อเสียงที่ดีกว่าแพลตฟอร์ม Forest (pyQuil) และ Quantum Developer Kit (Q#) โดยเปรียบเทียบจากการใช้งาน ทดสอบฟังก์ชันและอัลกอริทึมแล้วจะเห็นได้ว่า Qiskit และ ProjectQ จะมีความครบเครื่องใช้งานมาก จึงเป็น 2 แพลตฟอร์มที่ถูกคัดเลือกมาเปรียบเทียบประสิทธิภาพแบบจำลองโดยใช้ Shor's algorithm จะการทดสอบทำให้เห็นว่า Qiskit เป็นแพลตฟอร์มที่ดีที่สุดในเรื่องการใช้งาน ฟังก์ชันอัลกอริทึม เอกสารที่ใช้สำหรับศึกษาสามารถตอบโจทย์ได้ดีกว่า 3 แพลตฟอร์ม

ข้อเสนอแนะ

ข้อเสนอแนะในการนำผลการวิจัยไปใช้

หากของการนำผลการทดลองหรือการวัดประสิทธิภาพในวิจัยนี้ไปใช้งานควรใช้งานให้ใกล้เคียงกับเวลาปัจจุบันของวิจัยนี้ให้มากที่สุดเพราะเทคโนโลยีมีการพัฒนาที่รวดเร็วหากนำไปใช้ในอนาคตอาจจะทำให้เกิดข้อแตกต่างของผลลัพธ์การทำงานได้ แต่หากนำไปใช้เพื่อศึกษาความรู้พื้นฐานแพลตฟอร์มควอนตัมคอมพิวเตอร์หรือฟังก์ชันการทำงานต่างๆ ก็สามารถนำไปใช้งานได้ตลอดเวลา

ข้อเสนอแนะในการวิจัยครั้งต่อไป

ในการทำวิจัยครั้งต่อไปหากมีผู้สนใจในหัวข้อที่ใกล้เคียงกันสามารถนำประเด็นการทดสอบและวิจัยของบทความนี้ไปเพิ่มเติม โดยปัญหาหลักๆในปัจจุบันจะเป็นในเรื่องของทรัพยากร และข้อจำกัดในการใช้งานแต่ละแพลตฟอร์มควอนตัมคอมพิวเตอร์ หากในอนาคตมีการพัฒนาของเทคโนโลยีและทางเลือกแพลตฟอร์มควอนตัมคอมพิวเตอร์ที่มากขึ้นควรนำมาเปรียบเทียบกับแพลตฟอร์มควอนตัมคอมพิวเตอร์ตัวที่มีอยู่ก่อนแล้วเพื่อให้ได้แพลตฟอร์มที่ดีและตอบโจทย์กับการใช้งานของผู้ที่สนใจ

บรรณานุกรม

Damian S. Steiger, Thomas Häner, and Matthias Troyer. (2016). ProjectQ: An Open Source Software Framework for Quantum Computing. [2]

Ryan LaRose. (2019) Overview and Comparison of Gate Level Quantum Software Platforms. [1]

Thomas Häner, Damian S Steiger, Krysta Svore and Matthias Troyer. (2018) A software methodology for compiling quantum programs. [3]