

The Proof of Quantum Search Algorithm Optimization and the Simulation of the Algorithm Implementation

Hong-Tao Zhang^{1,2}, Yong-Tao Dai^{1,2}, Ling-Ying Tu^{1,3,*}, Jun Shu^{1,4}, Hong-Mei Xiong^{1,2}, Yi-Fan Hu^{1,2}

¹Nanoelectron Technology and Micro-system Laboratory, Hubei University of Technology, Wuhan, China

²Department of Communication, School of electrical and Electronic Engineering, Hubei University of Technology, Wuhan, China

³Department of Electronic Information Engineering, School of electrical and Electronic Engineering, Hubei University of Technology, Wuhan, China

⁴Department of Automation, School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan, China

Email address:

tuly.hbgd@163.com (Ling-Ying Tu)

To cite this article:

Hong-Tao Zhang, Yong-Tao Dai, Ling-Ying Tu, Jun Shu, Hong-Mei Xiong, Yi-Fan Hu. The Proof of Quantum Search Algorithm Optimization and the Simulation of the Algorithm Implementation. *Communications*. Vol. 3, No. 2, 2015, pp. 42-48.

doi: 10.11648/j.com.20150302.13

Abstract: This paper illustrated the optimality of Grover quantum search algorithm, and simulated the number of iterations and the specific implementation steps of quantum search algorithm with QCL in Linux operating systems, then validated the time complexity of Grover's quantum searching algorithm is $O(\sqrt{N})$ while the algorithm's time complexity on classical computers is $O(N)$. The Grover quantum search algorithm were shown to improve the performance significantly in the random database search problem. This conclusion was drawn from the simulation in the Linux operating systems adopted to quantum computation language. The present study highlighted the importance of high efficiency of the Grover quantum search algorithm to achieve the best possible performance in such cases. The existence of such efficiency is an outstanding advantage of quantum search algorithm over the conventional approach.

Keywords: Grover Quantum Search Algorithm, Optimality, Time Complexity, Linux, QCL

1. Introduction

In 1996, Grover proposed the groundbreaking quantum search algorithm[1] and put forward that the application of quantum computation can search problems for unordered database which carried on two times of acceleration effect in 1997, and it realized the fact that the classical algorithm searching number reduced from N to \sqrt{N} . The intention of the algorithm[2] is to perform a unitary transform through the initial amplitude superposition, increasing the target of quantum probability amplitude, while reducing other non-target of quantum probability amplitude. The bigger the final goal of quantum probability amplitude is, the greater the searching (measuring) probability of the correct target is. In recent years, in order to improve the success rate of Grover quantum search algorithm, scientists continued to change the algorithm. In 2007, Li Sun et al[3] verified on Experiment 3 qubits Grover quantum search in quantum

computation simulation program and achieved the nuclear magnetic resonance and the multiple quantum operator algebraic theory as the foundation, the multi qubit Grover quantum search algorithm of the NMR pulse sequence design method. In the same year, Hongyuan Ma et al[4,5] achieved the Grover quantum search algorithm in a thermal cavity. In 2009, Yudong Zhang et al [6,7] also proposed to improve the algorithm based on expanding the search space. At the same time, in order to compromise between the success rate and the number of iterations, a new parameter I was added to the algorithm which made it adjustable. The simulation experiment of the inverse problem showed that the method was more successful than the traditional Grover algorithm in the same number of iterations. And if the number of iterations was not limited, the probability of success can be higher. In 2013, Guangfu Han et al [8,9] modified the

corresponding relationship between the number of iteration step and the phase rotation angle of the Grover improved algorithm ; then simulated numerically these four kinds of improved algorithm which based on $\pi/2$ phase rotation, phase rotation, fixed phase rotation and accurate Grover algorithm and also gave a detailed comparative analysis for the simulation results. In 2015, Avatar Tulsi[10] offered implementation advantages for clause satisfaction problems.

Therefore, after years of improvement, Grover quantum search algorithm is gradually perfect, which has formed a relatively complete system of search algorithm, it can adapt to various different search demand. Evidence has been accumulating to establish generic superiority of quantum search algorithm.

In the present paper we focus our attention on a different aspect of quantum search algorithm, the simulation of the algorithm. In particular, the aim of this work is to simulate the number of iterations and the specific implementation steps of quantum search algorithm with QCL in Linux operating systems. It is necessary for us to do further research on quantum search algorithm, it may be used to speed up the solution to problems in the complexity class NP (such as Hamiltonian cycle problem), to do quantum counting, and to accelerate the Quantum search of an unstructured database and other classic problems, which is widely used and effective. It can effectively crack the DES^[11,12] cipher system, which has accelerated to search for potential use of key password system. This feature of efficiency is an outstanding advantage of quantum search algorithm.

We organize this paper as follows. In Sec. 2, we first explain the basic ideas and the theory of Grover quantum search algorithm. We then analyze the algorithm's flow chart and the specific implementation steps. Lastly, we prove that the algorithm we have demonstrated is optimal and illustrate the number of iterations of Grover quantum search algorithm. In Sec. 3, the simulation of the Grover quantum search algorithm is carried by QCL in Linux operating systems. we then present results of simulations and analysis it. Section 4 is devoted to conclusion.

2. Grover Quantum Search Algorithm

2.1. Oracle Searching Based on the Idea of Black Box

Oracle represents for the meaning of prophecy, God, philosophers, saints in ancient Greek, while it commonly known as "black box oracle". "Black box" is a program that can perform some computation tasks in computer science, while in quantum computation, it refers to a series of unitary transform that can accomplish some calculation tasks. When it analyzed the complexity of the problem, the costs of computing resources of "black box"

are not included.

Suppose we wish to search through a search space of N elements. For convenience we assume $N=2^n$, so the index can be stored in n bits, and that the search problem has exactly M solutions, with $1 \leq M \leq N$. A particular instance of the search problem can conveniently be represented by a function $f(x)$, which takes as input an integer x , in the range 0 to $N-1$. By definition

$$f(x) = \begin{cases} 1, & \text{if } x \text{ is a solution to the search problem} \\ 0, & \text{if } x \text{ is not a solution to the search problem} \end{cases} \quad (1)$$

Suppose there is a black box, a bit of which is used to identify the search of solution. In fact, the black box can be regarded as a definition for the unitary operator function

$$|x\rangle|q\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle \quad (2)$$

Where \oplus denotes addition modulo 2, $|x\rangle$ is the index register, and the oracle qubit $|q\rangle$ is a single qubit which is flipped if $f(x)=1$, and is unchanged otherwise. We can check whether x is a solution to our search problem by preparing $|x\rangle|0\rangle$, applying the oracle, and checking to see if the oracle qubit has been flipped to $|1\rangle$.

Usually, in the quantum search algorithm it is useful to apply the oracle with the oracle qubit initially in the state $(|0\rangle - |1\rangle)/\sqrt{2}$. Then the action of oracle is thus: $|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle$.

We say that the oracle marks the solutions to the search problem, by shifting the phase of the solution. For an N item search problem with M solutions, it turns out that we only need apply the search oracle $O(\sqrt{N/M})$ times in order to obtain a solution, on a quantum computer. There is a distinction between knowing the solution to a search problem, and being able to recognize the solution. Oracle can identify solutions, but if you want to find the solution requires the application of a certain algorithm once or repeatedly calls oracle.

2.2. The Theory of Grover Quantum Search Algorithm

The basic idea of Grover quantum search algorithm^[13,14,15] is repeatedly apply Grover quantum iteration process to enlarge the search for target items of probability amplitude, while suppressing the non target item probability amplitude. Finally, search for the target item with a high probability (close to 1) by measuring the quantum state.

Design input for n qubits $|0\rangle$ and Oracle work space. Schematically, the search algorithm operates as it shown in Figure 1.

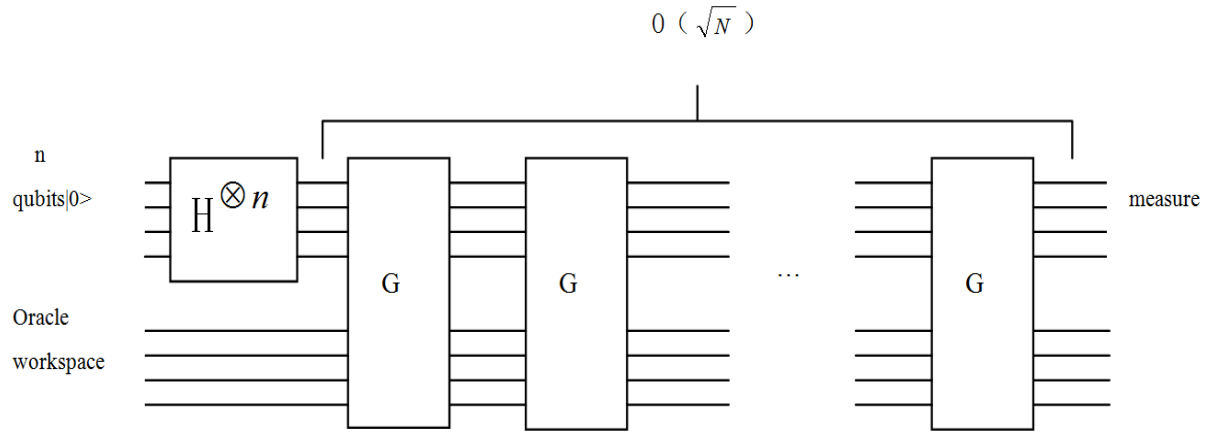


Figure 1. Schematic circuit for the quantum search algorithm.

From the Figure 1, we need to execute $O(\sqrt{N})$ search, and each search is called a Grover iteration. The algorithm begins with the computer in the state $|0\rangle^{\otimes n}$, the Hadamard transform is used to put the computer in the equal superposition states,

$$\begin{aligned} |\psi\rangle &= \frac{1}{N^{1/2}} (|0\rangle + |1\rangle + \dots + |\tau\rangle + \dots + |N-1\rangle) \\ &= \frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle \end{aligned} \quad (3)$$

Where $|\tau\rangle$ is the marked state we're looking for.

And then through the $O(\sqrt{N})$ Grover iteration to complete the search process. The Grover iteration, whose quantum circuit is illustrated in Figure 2, may be broken up into four steps:

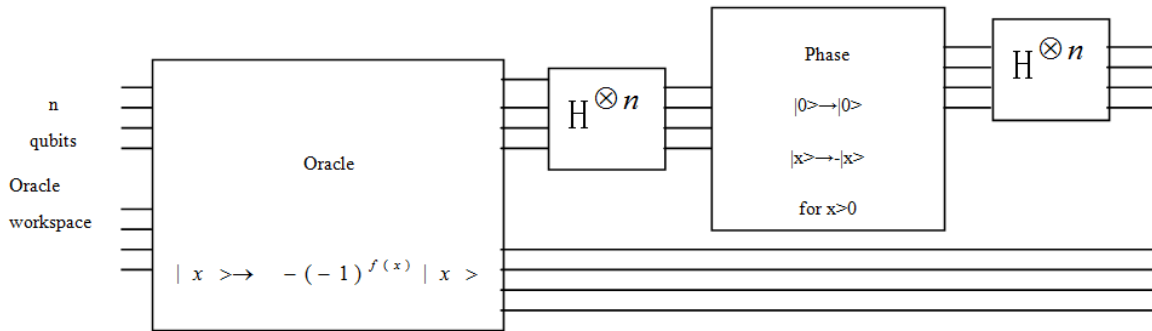


Figure 2. Circuit for the Grover iteration.

(1) Apply the Oracle O to verify whether the search element is the need to search in the solution actual problem solving.

(2) Apply the Hadamard transform $H^{\otimes n}$. That is, do the following operations on each bit:

$$H | 0 \rangle = \frac{1}{\sqrt{2}} (| 0 \rangle + | 1 \rangle) \quad (4)$$

$$H | 1 \rangle = \frac{1}{\sqrt{2}} (| 0 \rangle - | 1 \rangle) \quad (5)$$

(3) Perform a conditional phase shift on the computer, with every computational basis state except $|0\rangle$ receiving a phase shift of -1,

$$|x\rangle \rightarrow -(-1)^{\delta_{x0}} |x\rangle \quad (6)$$

(4) Apply the Hadamard transform $H^{\otimes n}$.

In the above steps, steps 2 and 4, the Hadamard transforms, require $n = \log(N)$ operations each, step 3, the conditional phase shift, may be implemented using $O(n)$ gates. The cost of the oracle call depends upon the specific application; for now, we merely need to note that the Grover iteration requires only a single oracle call.

2.3. Grover Quantum Search Algorithm'S Flow Chart and the Specific Implementation Steps

Step1: Preparation of superposition state.

$$|0^n, 0\rangle \xrightarrow{\text{Perform QFT: Fourier transform}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle \quad (7)$$

Step2: Compute parallelly.

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle \xrightarrow{F: |x, b\rangle \rightarrow |x, b \oplus f(x)\rangle} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, f(x)\rangle \quad (8)$$

Step3: Z operation transform.

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, f(x)\rangle \xrightarrow{\text{Z operation}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x, f(x)\rangle \quad (9)$$

Step4: Perform D transform on the first component of $|x, f(x)\rangle$ ---- augment the probability of solution b correspond to the state $|b, 1\rangle$.where D is described by the matrix

$$d_{ij} = \begin{cases} 2^{1-n} - 1, & \text{if } i = j \\ 2^{1-n}, & \text{if } i \neq j \end{cases} \quad (10)$$

Namely: the variation amplitude

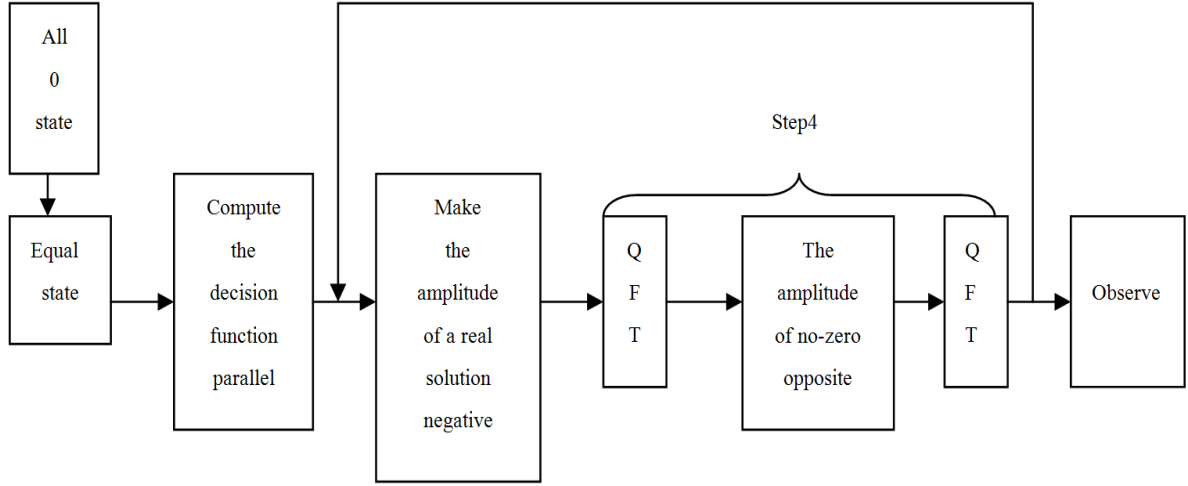


Figure 3. Grover quantum search algorithm's flow chart.

2.4. Optimality of the Grover Quantum Search Algorithm

In the case that there is only one solution for the search problem, we have shown that a quantum computer can search N items, consulting the search oracle only $O(\sqrt{N})$ times. We now prove that no quantum algorithm can perform this task using fewer than $\Omega(\sqrt{N})$ accesses to the search oracle, and thus the Grover quantum search algorithm we have demonstrated is optimal^[16, 17].

Suppose the algorithm starts in a state $|\psi\rangle$ and applies the oracle O_x exactly k times, with unitary operations U_1, U_2, \dots, U_k interleaved between the oracle operations. Define

$$|\psi_k^x\rangle = U_k O_x U_{k-1} O_x \dots U_1 O_x |\psi\rangle, \quad (13)$$

$$|\psi_k\rangle = U_k U_{k-1} \dots U_1 |\psi\rangle \quad (14)$$

$$(a_0, a_1, \dots, a_{2^n-1}) \rightarrow (a'_0, a'_1, \dots, a'_{2^n-1}) \quad (11)$$

$$a'_i = \sum_{j=0}^{2^n-1} d_{ij} a_j = d_{ii} a_i + \sum_{j:j \neq i} d_{ij} a_j = (2^{1-n} - 1) a_i + 2^{1-n} \sum_{j:j \neq i} a_j = 2^{1-n} \sum_{j=0}^{2^n-1} a_j - a_i \quad (12)$$

Among them, a_i is the amplitude of the $|i\rangle$.

Step5: Repeat the implementation of step3 and step4 step3 and step4 for k times, and measure the state, we will get the result $|x, 1\rangle$, and output x as a real solution.

That is, $|\psi_k\rangle$ is the state that results when the sequence of unitary operations U_α is carried out, without the oracle operations. In order to simplify formulas, we use the notation ψ_k^x for $|\psi_k^x\rangle$ and ψ_k for $|\psi_k\rangle$ as a convenience.

$$D_k = \sum_x \|\psi_k^x - \psi_k\|^2 \quad (15)$$

Where D_k is the estimator, it is a measure of the deviation after k steps caused by the oracle, from the evolution that would otherwise have ensued. If this quantity is small, then all the states $|\psi_k^x\rangle$ are roughly the same, and it is impossible to correctly identify x with high probability. This demonstrates two things: (1) a bound on D_k that shows it can grow no faster than $O(k^2)$; (2) a proof that D_k must be $\Omega(N)$ if it is to be possible to distinguish alternatives. Combining these two results gives the desired lower bound.

Firstly, we give an inductive proof that $D_k \leq 4k^2$.

(1) When $k=0$, $D_k=0$ is clearly established.

(2) Assume that k is established, so for $k+1$, there is

$$D_{k+1} = \sum_x \|U_{k+1} O_x \psi_k^x - U_{k+1} \psi_k\|^2 = \sum_x \|O_x \psi_k^x - \psi_k\|^2 \quad (16)$$

Applying the unitary transform to the second equals on the right hand side: it is only change the abnormal vector direction, do not change the abnormal mold of vector.

Furthermore, we add

$$O_x \psi_k \text{ to } \|O_x \psi_k^x - \psi_k\|,$$

thus

$$D_{k+1} = \sum_x \|O_x (\psi_k^x - \psi_k) + (O_x - I) \psi_k\|^2 \quad (17)$$

Applying $\|a+b\|^2 \leq \|a\|^2 + 2\|a\| \cdot \|b\| + \|b\|^2$ with $a = O_x (\psi_k^x - \psi_k)$ and $b = (O_x - I) \psi_k = -2|x\rangle\langle\psi_k| + |x\rangle\langle x|$, gives

$$D_{k+1} \leq \sum_x (\|\psi_k^x - \psi_k\|^2 + 4\|\psi_k^x - \psi_k\| \cdot |\langle x|\psi_k\rangle| + 4|\langle x|\psi_k\rangle|^2) \quad (18)$$

Applying the Cauchy-Schwarz inequality to the second term on the right hand side, and noting that $\sum_x |\langle x|\psi_k\rangle|^2 = 1$, gives

$$D_{k+1} \leq D_k + 4(\sum_x \|\psi_k^x - \psi_k\|^2)^{\frac{1}{2}} (\sum_x |\langle x|\psi_k\rangle|^2)^{\frac{1}{2}} + 4 \leq D_k + 4\sqrt{D_k} + 4 \quad (19)$$

By the inductive hypothesis $D_k \leq 4k^2$, we obtain

$$D_{k+1} \leq 4k^2 + 8k + 4 = 4(k+1)^2 \quad (20)$$

In summary, the establishment of

$$D_k \leq 4k^2 \quad (21)$$

Secondly, to complete the proof we need to show that $D_k \geq \Omega(N)$. Defining $E_k = \sum_x \|\psi_k^x - x\|^2$, and we suppose

$$|\langle x|\psi_k^x\rangle|^2 \geq \frac{1}{2} \text{ for all } x, \text{ so without loss of generality we}$$

may assume that $\langle x|\psi_k^x\rangle = \langle\psi_k^x|x\rangle$, and therefore

$$E_k = \sum_x \|\psi_k^x - x\|^2 = \sum_x (2 - 2|\langle x|\psi_k^x\rangle|) \leq \sum_x (2 - \sqrt{2}) = (2 - \sqrt{2})N \quad (22)$$

Defining

$$F_x = \sum_x \|x - \psi_k\|^2, \text{ and } F_x \geq 2N - 2\sqrt{N} \quad (23)$$

We have

$$D_k = \sum_x \|(\psi_k^x - x) + (x - \psi_k)\|^2 \geq E_k + F_k - 2\sum_x \|\psi_k^x - x\| \cdot \|x - \psi_k\| \quad (24)$$

Applying the Cauchy-Schwarz inequality gives

$$\sum_x \|\psi_k^x - x\| \cdot \|x - \psi_k\| \leq \sqrt{E_k F_k}, \text{ so we have}$$

$$D_k \geq E_k + F_k - 2\sqrt{E_k F_k} = (\sqrt{F_k} - \sqrt{E_k})^2 \quad (25)$$

Combining (22) with (23) gives $D_k \geq cN$ for sufficiently large N , where c is any constant less than $(\sqrt{2} - \sqrt{2 - \sqrt{2}})^2 \approx 0.42$. Since $D_k \leq 4k^2$ this implies that

$$k \geq \frac{\sqrt{cN}}{2}.$$

Therefore, to find a solution to the search problem with at least one-half probability, we must call the oracle $\Omega(\sqrt{N})$ times.

2.5. The Number of Iterations of Grover Quantum Search Algorithm

In order to obtain a solution of search problems, how many times must the Grover iteration be repeated to rotate $|\psi\rangle$ near $|\beta\rangle$? The initial states of the system can be described as

$$|\psi\rangle = \sqrt{(N-M)/N} |\alpha\rangle + \sqrt{M/N} |\beta\rangle \quad (26)$$

Where $|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_x |x\rangle$, $|\beta\rangle = \frac{1}{\sqrt{M}} \sum_x' |x\rangle$, N is the total number of state, M is the total number of labeled state. And \sum_x' indicates a sum over all x which are solutions to the search problem, \sum_x indicates a sum over all x which are not solutions to the search problem.

So rotating through $\arccos\sqrt{M/N}$ radians takes the system to $|\beta\rangle$. Then repeating the Grover iteration.

$$R = CI\left(\frac{\arccos\sqrt{M/N}}{\theta}\right) \quad (27)$$

$CI(x)$ is represented as integer which is closest to the real numbers.

To achieve this, note from the above formula that $R \leq \lceil \pi/2\theta \rceil$, so a lower bound on θ will give an upper bound on R . Assuming for the moment that $M \leq N/2$, we have $\theta/2 \geq \sin(\theta/2) = \sqrt{M/N}$, from which we can derive the number of iterations required, where by convention we

round halves down, $R \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil$. That is, $R = O(\sqrt{N/M})$

Grover iterations must be performed in order to obtain a solution to the search problem with high probability. The search times of the classical algorithm is $O(N)$, which is two times of acceleration of the classical search algorithm.

3. The Simulation of the Grover Quantum Search Algorithm

3.1. The Basic Concept of QCL

QCL^[18,19] was put forward by B.Omer in 1998, which is a structured imperative quantum programming language. The grammar of it is similar to C/Pascal, using quantum process and function to define the quantum operation, introducing the quantum register, defining quantum rich data types, considering quantum control structure based on conditions transform. The QCL statement has three categories, simple statements, control flow statements, and interactive command; there existed several types which include scalar, Zhang Liang, quantum, Boolean, string and so on, while the type of scalar include integer, real type, complex type, Boolean and string type; Zhang Liang types include three quantum types, such as a vector, matrix and n order Zhang Liang; Quantum types has the general register (qreg), quantum constant (quconst), the destination register (quvoid), erase the register (quscratch) of several, and designed the institutions of transparent removal which make few qubits can be fully utilized.

3.2. QCL Setup in Linux Operating Systems

QCL is only available in Linux operating systems, and ubuntu is a desktop application based on Linux operating system, so we choose to install QCL in ubuntu. Here are installation steps and the corresponding solutions to problems:

- (1) Download the QCL program, the current version is here. (<http://tph.tuwien.ac.at/~oemer/tgz/qcl-0.6.4.tgz>)
- (2) Download and install the bison and flex tools.
- (3) Install libplot2c2 and libplot-dev (apt-get install libplot2c2 libplot-dev). This is a key step, otherwise you couldn't find the include<plot.h> file, the errors will occur, thus unable to install qcl successfully.
- (4) Download the latest readline (readline-5.2.tar.gz), then decompress and install.
- ①tar xvfz readline-5.2.tar.gz; ②./configure; ③make; ④make install.
- (5) Entering the folder after the decompression, make and then to generate direct current folder QCL, then QCL has been installed successfully.

If the error occurs: /usr/bin/ld:cannot find -lncurses

The solution: apt-get install libncurses-dev

- (6) The experimental steps: ①Type 'qcl' to start the environment; ②Type 'include "grover.qcl";' to import the

grover.qcl file; ③Use 'grover(N)' to run the Grover Search for the number N.

3.3. Experimental Results and Analysis

When N=2048, the search results of Grover quantum search algorithm as shown in figure 4:

```
root@ubuntu:/home/qcl/qcl-0.6.4# ./qcl
QCL Quantum Computation Language (32 qubits, seed 1437117806)
[0/32] 1 |0>
qcl> include "grover.qcl"
qcl> grover(2048)
: 12 qubits, using 26 iterations
: measured 2048
[0/32] 1 |0>
qcl> exit
root@ubuntu:/home/qcl/qcl-0.6.4#
```

Figure 4. The search results of Grover quantum search algorithm.

Analysis: Floor (x) is the function of "rounding down", or "rounded down", the largest integer that is not larger than x. Ceil (x) function: the smallest integer greater than or equal to the specified expression returns. When N is equal to 2048, this is equivalent to an English dictionary pages, Grover quantum search algorithm is very effective. The number of bits required for $n = \text{floor}(\log(2048, 2)) + 1 = 12$, the number of iterations of $m = \text{ceil}(\pi/8 * \sqrt{2^{12}}) = 26$, is consistent with the number of

iterations of $R \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil$ previously mentioned, which

verified the time complexity of Grover's quantum searching algorithm is $O(\sqrt{N})$, but the algorithm's time complexity on classical computers is $O(N)$. Owing to the qubits entangled, the effect of quantum interference will make the previous results to the next quantum operation. After the operation of this kind of interference is repeated \sqrt{N} times, the probability of obtaining the correct answer for 1/2. And if you have to repeat the operation a few times, you can find the required solution close to 1.

4. Conclusion

This paper introduces the method of Grover quantum search algorithm in detail, for a database, which contains $N = 2^n$ element, and the search problem has exactly M solutions, with $1 \leq M \leq N$. Firstly, prepare for the uniform superposition state, and then repeat $G = H^{\otimes n} UH^{\otimes n} O$, where O is a Oracle search. The role of O is, if x is a solution to the search problem, then $|x\rangle \rightarrow -|x\rangle$, otherwise unchanged; secondly, the function of U is changed from $|0\rangle$ to $-|0\rangle$, and keep all the other computing base invariant. Finally, measuring the final state, a solution with a high probability of the problem is obtained, and the Grover quantum search algorithm has been proved to be the optimal search algorithm. Then simulate the number of iterations of quantum search algorithm with QCL in Linux operating systems, and validate the time complexity of Grover's quantum

searching algorithm is $O(\sqrt{N})$, but the algorithm's time complexity on classical computers is $O(N)$.

Acknowledgement

This project was supported by “the quantum computation and quantum communication research team program” of School of electrical and electronic engineering of Hubei University of Technology and “thousand new energy vehicle program of ten cities” of Wuhan Science and Technology Bureau (NO.2013011801010600).

References

- [1] L. Grover. In Proc. 28th Annual ACM Symposium on the Theory of Computation, pages 212–219, ACM Press, New York, 1996.
- [2] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett, 79(2):325, 1997. arXiv e-print quant-ph/9706033.
- [3] Sun Li, Lu Chun-hong. NMR Simulation of 3-qubit Grover Quantum Search Algorithm, Master's degree, Jiangnan University, 2007.
- [4] MA Hong-yuan, WANG Hong-fu, ZHANG Shou. Implementation of the Grover Quantum Search Algorithm in Thermal Cavity[J], Journal of Yanbian University, 2008, 34(1):27-30.
- [5] Ye A L, Guo G C. Scheme for Implementing Quantum Dense Coding in Cavity QED[J]. Phys Rev A, 2005, 71:034304.
- [6] ZHANG Yu-dong, WEI Geng, WU Le-nan. An Improved Grover Quantum Searching Algorithm[J], Signal processing, 2009, Vol25, No. 2:256-259.
- [7] V. Protopopescu, J. Barhen. Solving a class of continuous global optimization problems using quantum algorithms[J]. Physics Letters A, 2002, 296(2002):9-14.
- [8] HAN Guang-pu. The Improvement of Quantum Grover Algorithm and Its Application to Image Retrieval, Master's degree, Nanjing University of Posts and Telecommunications, 2013.
- [9] Christoph Durr, Peter Hoyer. A quantum algorithm for finding the minimum. Quantum Physics, 1999, 1.
- [10] Avatar Tuli. Quantum search algorithm tailored to clause satisfaction problems[J]. Quantum Physics, 2015.
- [11] SU Xiao-qin, GUO Guang-can. Quantum communication and quantum computation[J]. Chinese Journal Of Quantum Electronics 2004, 6:31-34.
- [12] Kwiat P G, Mitchell J R, Schwindt P D D, et al. Grover's search algorithm: an optical approach[J]. J. Mod. Optic, 2007, 47(2-3):257-266.
- [13] Grover L K. Quantum computers can search rapidly by using almost any transformation[J]. Physical Review Letters, 1998, 80(19): 4329-4332.
- [14] Grover L K. Fixed-point quantum search[J]. Phys. Rev. Lett, 2005, 95(150501)1-4.
- [15] Grover L K.: Rapid sampling through quantum computing. In: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC 2000), pp. 618–626 (2000).
- [16] Boyer M, Brassard G, Hoyer P, Tapp A. Tight bounds on quantum searching. In: Proceedings of the Workshop on Physics of Computation: PhysComp'96. IEEE Computer Society Press, 1996.
- [17] Nielson MA, Chuang IL. Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2000.
- [18] B. ömer. A procedural formalism for quantum computing. Master's thesis, Department of Theoretical Physics, Technical University of Vienna, 1998.
- [19] B. ömer. Structured Quantum Programming, PhD thesis, Technical University of Vienna, 2003.