Quantum Computing: A Distributed Approach

Report Submitted for Partial Fulfillment of the Requirements for
Degree of Bachelors of Technology

By

Devesh Tiwari
(Y3109)

To

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur India 208016

April 2007

Certificate

This is to certify that this report entitled "**Quantum Computing: A Distributed Approach**", contains the work carried out by ***Devesh Tiwari***, under our supervision and the same has not been submitted elsewhere towards achievement of a degree. Prof. Harish C Karnick from CSE Department has been the administrative co-guide for this project without whose help this report could not have been completed.

Advisor

Prof. Debabrata Goswami
Department of Chemistry
IIT Kanpur

April 2007

Abstract

Quantum Computation literature, though bursting with interesting paradigms and problems, is still in its nascent phase and a careful focus is necessary on scaling the problem size so that an efficient physical realization is possible. In this work, we explore potential improvement in the run-time of quantum algorithms. Our proposed scheme of distributed quantum computing environment overcomes the restriction of small computing space by virtually providing a larger qubit space in a distributed network. We theoretically discuss the performance issues of this proposed computing network, which promises better performance with little or no implemental quantum overhead.

We focus on the Grover's search algorithm. Grover's Algorithm finds a unique element in a stock of N-elements in $\sqrt{N}$ queries through quantum search. A single-query solution can also be designed, but with an overhead of $N \log_2 N$ steps to prepare and post process the query, which is worse than the classical $\frac{N}{2}$ queries. We show, here, that by distributing the computing load on a set of quantum computers, we achieve better information theoretic bounds and relaxed space scaling. Such a distributed approach is not handicapped by implementation issues, yet it promises a scalable quantum performance.

Acknowledgement

At the very outset, I convey my heart-felt gratitude to my advisor Prof. Debabrata Goswami who has been mentoring me throughout the project. I am especially thankful for the fruitful discussions we had, which led to novel ideas and helped me learn to address various topics in different manner appropriately and appreciate the motivation behind and extend them further. He has been a great mentor in every possible way. Apart from the technical discussions, I would be always thankful to him for sharing his personal experiences with me. The lessons I learnt from those experiences will help me excel in my academic and non-academic career.

I also wish to extend my thanks to Prof. Harish C Karnick for becoming my Departmental Guide and also for co-operating with our schedule while fixing evaluation plans.

I am indebted to Prof. Sanjeev Saxena, Prof. Somenath Biswas and Prof. Ajai K Jain for their valuable suggestions and criticism which helped in great way to make progress throughout.

I am also very thankful to my QUACK Laboratory Members (all students and staff esp. Ram Kumar) for discussions and help provided at different occasions. Quantum Computing Class Discussions also deserve a notable mention for providing insight of the subject.

Lastly, I owe many thanks to my parents and siblings who have always encouraged me in all of my educational activities.

Funds for this research were available from the MCIT (Govt. of India), the Swarnajayanti Program of the DST (Govt. of India) and the International SRF Program of the Wellcome Trust Foundation, UK.

Contents

Figures

I. Introduction

Grover's Search Algorithm shows that a quantum mechanical system needs $O\left(\sqrt{N}\right)$ steps in order to identify a unique candidate satisfying a particular condition out of an unsorted dataset of N candidates [1, 2]. This quadratic improvement is less optimal than the possible exponential improvement in quantum computing [3, 4] as is seen, for example, in Shore's Factorization Algorithm [5]. However, this has great significance as the search problem is a universal necessity in quantum computing. Grover's subsequent work [6] concludes that one can overcome $O\left(\sqrt{N}\right)$ bottleneck by making more elaborate queries, however, it increases overhead in preparing and post-processing of queries by $O\left(N\log_2 N\right)$ steps resulting in a worse case than the classical situation. In this work, we present a novel approach of a distributed quantum computing model, which promises to perform a better quantum search by providing a lower theoretical bound on the resource requirements of Grover's Algorithm. This study is motivated primarily by the fact that at present, achieving a large qubit space is difficult for a quantum computer, which is one of the major bottlenecks for the effective implementation of the proposed algorithms. In the future, such a network of quantum computers could virtually produce the required qubit space for effective implementation of various algorithms [7].

II. Search Problem

Suppose we are given a set of many items out of which there is only one item which satisfies a given condition. Our aim is to find out that eligible item. We can phrase the same problem in a different way also. Let us say we have a set of balls out of which only one ball is marked, while the others are not marked. Our goal is to locate that marked ball.

Mathematically, let there be a given database of N elements, say $\left(X_1, X_2, \ldots, X_N\right)$, where only one element satisfies a given condition and that very element is X_k. Our objective is to identify that very element X_k by asking minimum possible number of queries. Queries are presented to the Oracle (black box) which can give an answer to any asked questions but it outputs only as high/low (Yes/No). For example, if we ask whether X_i satisfies the condition, it would set the output signal high only if $X_i = X_k$, otherwise the output signal is low. Since the black box can answer any question, it implies that a classical search can return the element X_k in $O\left(\log N\right)$ queries at best. We refer such an element that satisfies the given conditions as a qualified element. The aim is to get the high signal (Yes answer) in as minimum number of queries as possible. Classically, the optimal approach is to ask questions that would eliminate half of the elements under consideration with each question. This approach results in approximately $\log_2 N$ queries to reach the answer [1,2]. In Grover's single query approach [6], he considered a quantum system composed of multiple subsystems where each subsystem has N-dimensional states and each basic state of a subsystem corresponds to an element in the database. The purpose of the single query approach was to amplify the probability of the qualified element in each subsystem by performing unitary operations on the subsystems. However, the required number of subsystem for such a single query case is so high that it makes the pre-processing and post processing of the queries of $O\left(N \log N\right)$ and hence it is even worse than the classical case [8].

III. Previous Work

We have first built up on Varun Garg’s BTP work on quantum networking where he extended Grover’s many subsystem and single query framework [6]. His work repeats Grover’s step for ‘n’ times and achieves comparable theoretical performance using probability theory [9]. It promises classical scale improvement. However, all this seems to be fruitful where there would not be any restriction on qubit space and its scalable implementation. Varun’s work is also limited by the fact that it depends on communication among quantum subsystems. As we move away from such an ideal situation to a more practical one, this might cause serious problems. So we have focused on developing easily scalable quantum system with better run time performance. We have also focused on avoiding communication problem so that physical implementations of such systems are not handicapped by communication problem [10].

IV. Distributed Approach

(a) Design of Proposed Model

Our distributed quantum computing model consists of many quantum computers on which quantum operations can be done parallelly. We should emphasize here that each quantum computer is a different computing entity though quantum operations can be done parallelly on all systems. Thus this model is similar to Grover’s subsystem model. Schematic of such proposed network is shown in Figure1.

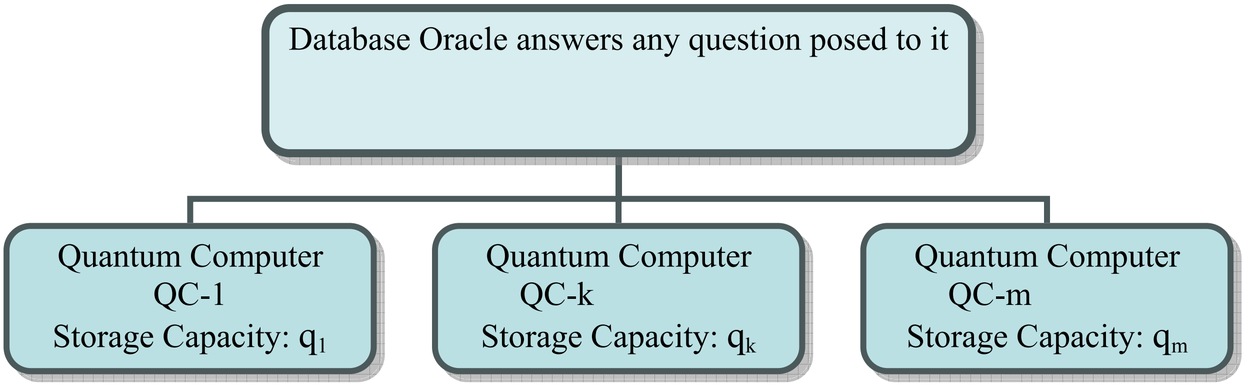
Database Oracle answers any question posed to it
Quantum Computer QC-1 Storage Capacity: q_1
Quantum Computer QC-k Storage Capacity: q_k
Quantum Computer QC-m Storage Capacity: q_m

Figure 1: Distributed Quantum Computing Model

Let us say that there are a total of m quantum computers, where the k^{th} computer has the storage space of $q_{k.}$ Therefore total quantum computing space is sum of q_1, q_2, q_3 … q_m. There is no restriction on storage capacity of any particular quantum computer and thus our proposed model has the virtue of scalability.

At this point, it should also be noted that storage capacity at different computing nodes does affect run time of quantum algorithm. This is discussed later.

Another key feature of the proposed model is that it does not require any communication between different quantum computers for executing the quantum search algorithm, neither in the form of data sharing nor in the form of information exchange.

(b) Quantum Search Algorithm

In this section, we will first discuss the steps necessary for running the quantum search algorithm.

First we divide the given database among ‘m’ quantum computers according to maximum capacity of each quantum computer. Note that if there is only one marked item in the entire database then only quantum computer out of ‘m’ quantum machines will give an answer. We first discuss the case when only one item is marked.

Load distribution can be explained with the help of an example. Let us say that the original search problem has 28 elements and only one item (say the 19th item) is marked. We have four quantum computers such that the first, second and the fourth quantum computers have 3 qubits each while the third one has only 2 qubits. Thus, the first and the second quantum computer can store 8 items and hence the first 16 items would be stored in first and second quantum computer. The 17th, 18th, 19th and 20th elements would be stored on third quantum computer and rest of the 8 elements on the fourth quantum computer. It is worth noting that if we want to solve this problem by using single quantum computer, we would need a five qubit size quantum computer. However, here all we need is a quantum computer of size three qubits as compared to the otherwise required five-qubit single computer. Given the fact that building large-scale quantum computers is a major bottleneck, such efforts would be appreciated where we can solve bigger problems circumventing the use of large quantum computers. It should be noted that such benefit is highly amplified in case of very large database search problems.

We use two quantum operators named as Walsh-Hadamard and Selective phase rotation operators which are also used by Grover in his algorithms[1,6] It has been proved that both operators are unitary and physical realization of them is possible[1].

(c) Quantum Steps

We perform quantum operations on all quantum computers in parallel. First of all, we initialize quantum registers in all quantum computers by putting them uniquely in the first state (1, 0, 0…0). Then, we place the register in an equal superposition of all states (${}^{1}\!/\!{\sqrt{N}}$, ${}^{1}\!/\!{\sqrt{N}}$, ${}^{1}\!/\!{\sqrt{N}}$,….., ${}^{1}\!/\!{\sqrt{N}}$) by applying the Walsh-Hadamard operator. Here '*N*' also refers to the total number of elements that a particular quantum machine can hold, which can be different for different machines. For an arbitrary machine we have assumed it to be equal to data-size *N*. The Walsh-Hadamard operator puts the state vector in an equal superposition of each state.

We repeat $O\left(\sqrt{N}\right)$ times the following two steps (step 1 and 2) which are similar to Grover's work [1]. The precise number of iterations is important, and is discussed later:

1. Let the system be in any state S. If C(S) = 1, rotate phase by π radians, and if C(S) = 0 leave the system unaltered. Value of C(S) tells whether the state is marked or not. If state 'S' is marked, value of C(S) comes out to be one, otherwise zero.

2. Apply the diffusion transform D which is defined by the matrix D as: $D_{ij} = \frac{2}{N}$ if $i \neq j$; $D_{ii} = -1 + \frac{2}{N}$; where *D* can be physically implemented as a product of three local unitary matrices [2].

3. Measure the quantum register. The measurement will yield n bit label of the marked state $C(S_M) = 1$ with a probability of at least half. S_M is the marked state.

Let us look at the second step more closely. The diffusion matrix is defined as follows: $D_{ij} = \frac{2}{N}$ if *i*≠*j*; $D_{ii} = -1 + \frac{2}{N}$. *D* can be broken into $D = \text{-}I + 2P$ where *I* is the identity matrix and *P* is a projection matrix with $P_{ij} = \frac{1}{N}$ for all *i*, *j*. It is clear that when we apply *P* on any vector, it produces a new vector whose individual components are equal to the average of all the components of the original vector. When we apply *D* on a vector *v* we get:

$$Dv = (\text{-}I + 2P)v = \text{-}v + 2Pv$$

From the above discussion, each component in *Pv* is A, where A is the average of all the components of the vector *v*. Hence, i^{th} component of the vector *Dv* is given by (-*vi* + 2A) which can be expressed as (A + (A – *vi*)) and this is indeed the inversion about average.

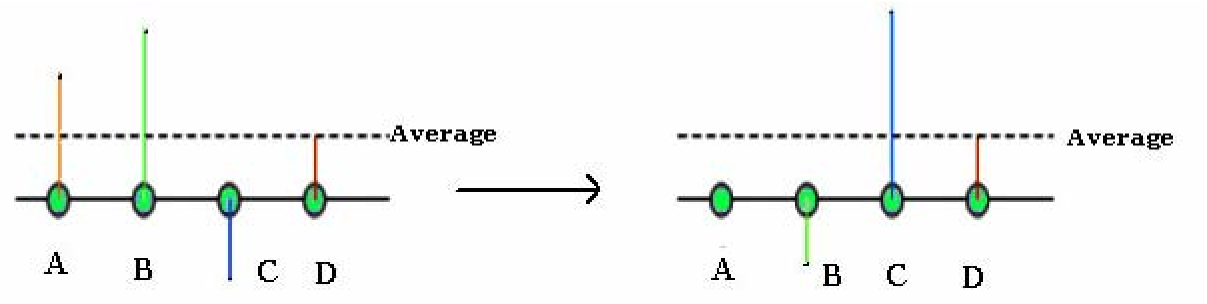

Figure 2: Inversion about average operation

We have seen that the diffusion transform, *D*, can be interpreted as an inversion about average operation. If we let α_i to be the amplitude in the i^{th} state, then the average is given by $\frac{1}{N}\sum_{i=1}^{N}\alpha_i$. As a result of the operation *D*, the amplitude in each state increases, so that after the operation it is as much as below α_i as it was above α_i before the operation and vice versa (Figure 2). Every iteration in this loop increases the amplitude in the desired state by $O\left(1/\sqrt{N}\right)$ and as such the number of steps have to repeated $O\left(\sqrt{N}\right)$ times. In a later section we will show that we can do even better than this. As proved by Grover in [1] such quantum operations do amplify the amplitude of marked item and are physically realizable.

An illustration is shown in Figure 3 for a quantum computer with 3 qubit space where the 4th element is marked.

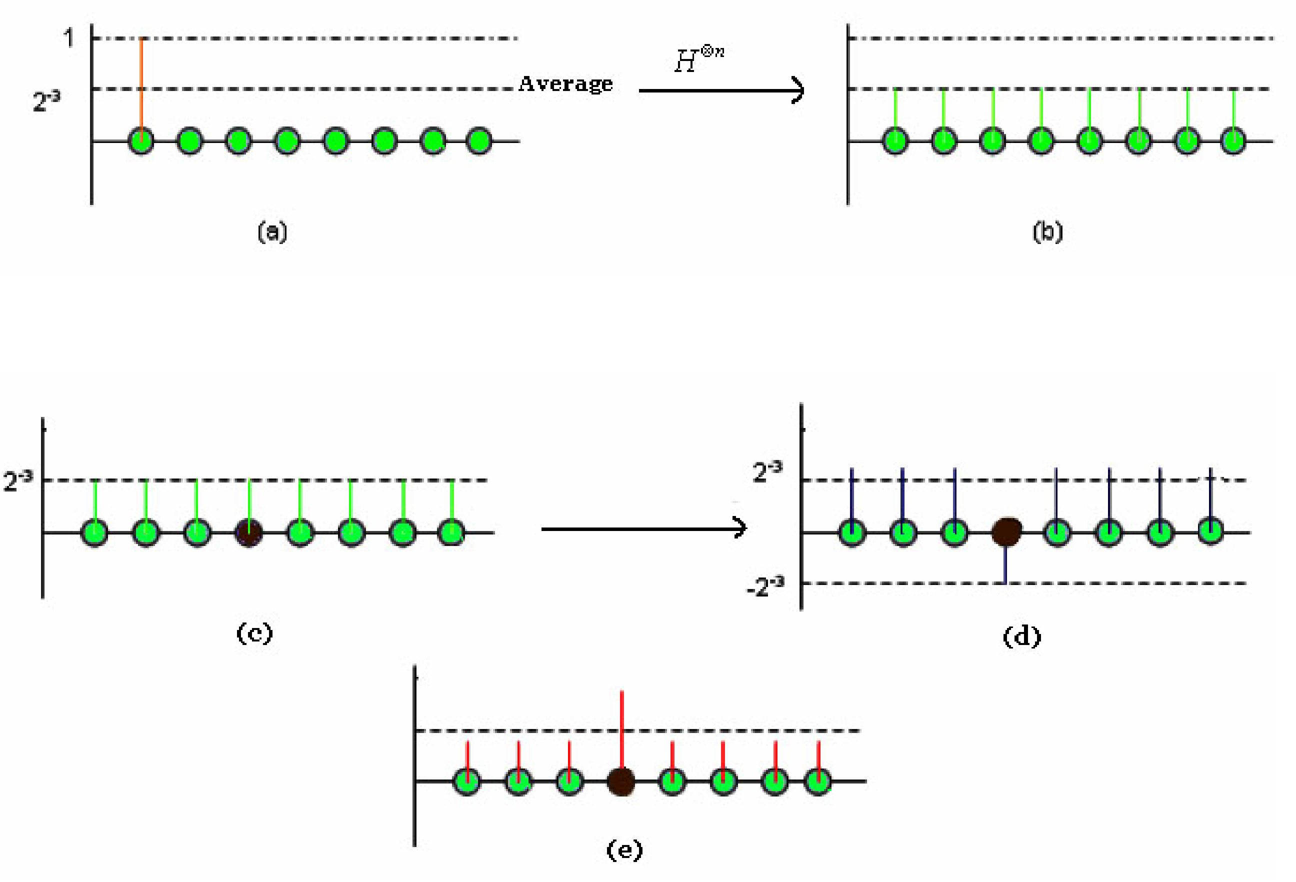

Figure 3: Illustration of Quantum Steps

(d) Measurement

As the storage space (N, number of elements a particular quantum computer can hold) is different for each quantum computer, we have to wisely decide when to make measurement for each quantum computer. Measurement is made for each quantum computer after $O\left(\sqrt{N}\right)$ steps where N refers to number of elements which a particular node can store. As N is different for each machine, there would be measurement at different times, though it is not necessary to make the measurement for all the machines. We discuss two cases: in the first case, there is only one marked item, and in the second case, number of marked items is more than one.

Case 1: One Marked Item

When only one item is marked, the operation on all computers is stopped as soon as *any* quantum machine answers positively (i.e. marked item resides on that particular quantum computer). Those machines which do not have the marked element will not reply positively after the $O\left(\sqrt{N}\right)$ steps have been performed on them. It is very important to note this point since it is critical in the discussion of lower bounds in our distributed quantum search algorithm.

Case 2: Multiple Marked Items

In case of multiple marked items, let us say that the total number of marked items is 't'. In this case, we cannot stop measuring until all the 't' items have been found. However, this 't' cannot be greater than one-fourth of the total data base size as discussed by Grover[6]. In case, we do not know the number of marked items in advance, we encounter the worst case scenario and need to perform $O\left(\sqrt{N}\right)$ operations for each quantum computer in parallel. We will prove in next section that we can, in fact, do better than this and reduce the number of required queries irrespective of the fact whether we know the number 't' in advance or not.

V. Discussion

In this section we discuss the lower bounds of the proposed distributed algorithm.

Case I. Unique Marked Item

It has been shown [11,12] that any quantum computer that can hold N elements, can be used to detect the marked element in approximately $\frac{\pi}{4}\sqrt{N}$ queries, provided N >>1. Remarkably, only $\frac{\pi}{8}\sqrt{N}$ queries are sufficient to identify the marked item with at least 50% probability of success as also shown by Grover in his first paper [1]. This has been experimentally realized for case N=4 [12]. Such experimental results further boost up the novelty and significance of our distributed approach. Since approximately $\frac{\pi}{8}\sqrt{N}$ are required to identify the marked element on any given quantum computer, there are two extreme cases for our distributed approach:

1. Best case: When the marked item is stored at a quantum computer, which has the smallest storage capacity (say L elements), we merely need approximately $\frac{\pi}{8}\sqrt{L}$ queries. This is huge improvement for all practical cases (where L elements are, for example, as small as $2^3 = 8$) compared to Grover's algorithm where the approximate number of queries is $\sqrt{Z}$ (Z is size of total database). For real life search problems, Z can be million times larger than L or perhaps even more.

As Grover's subsystem extension case is worse than Grover's original case, we skip the performance comparison with that.

2. Worst case: When the marked item lies within a quantum computer, which has the largest storage capacity (say M elements), we merely need about $\frac{\pi}{8}\sqrt{M}$ queries. Again this comes with big performance enhancement over Grover's original case.

3. Average case: On an average, one would need approximately $\frac{\pi}{8}\sqrt{K}$ queries, where K is average number of elements that a quantum computer holds in a given set of quantum computers.

Clearly, our approach promises huge theoretical performance improvement over Grover's original and extended work.

Case II. Multiple Marked Items

It can be shown that, if there are 't' marked elements in the database, it takes $O\left(\sqrt{N/t}\right)$ time to identify them [12], irrespective of the fact whether we know '*t*' in advance or not. A very good implication of this result is that our distributed approach will again perform better than Grover's Algorithm. In worst possible case, we will have only one marked item (out of total '*t*' marked items) at the quantum computer which has largest storage capacity (M elements). In this case, $\sqrt{M}$ queries are required to identify all marked elements. In the best possible case, all marked elements are residing on the smallest storage quantum computer. In practice, however, problems with such skewed distribution seem to be rare, and hence claim very high theoretical performance of lesser relevance. But, we have shown already that even the worst possible case promises very good theoretical performance improvement and in the light of virtue of scalability of our design, we claim it to be better than previous attempts towards the same direction.

VI. Conclusion and Future Work

In this work, a quantum distributed computing model has been proposed, which promises much better theoretical performance for quantum search algorithm. Every crucial step of such a computation has been explained in detail in this report and performance analysis has been discussed.

In addition to better theoretical performance, our model dilutes the concern of implementation difficulties. This proposed distributed model facilitates easy physical implementation of quantum computing. We observe a major leap in performance despite of the fact the computing is still performed on a collection of small qubit size computers. This shows that such computation model virtually promises the performance of what usually comes by working on very large single computing space. Remarkably, this model does not run into communication overhead and data sharing problems and hence it can function as an ideal model with little effect of moving away from ideality. It has no constraint on the problem size or resource requirements hence should be preferred as elegant quantum computing model.

One important point to be concerned with is the case of Grover's Algorithm restriction where the number of marked items cannot be more than one-fourth of the whole database. In some special cases, this restriction can particularly give rise to a faulty load distribution and hence should be solved in further attempts.

Future work towards this direction may also include attempting to run Shore's algorithm on this model efficiently. GHZ states can be used to communicate in a distributed model. [13]

References

[1] L. K. Grover, “Proceedings of the Twenty-Eighth Annual Symposium on the Theory of Computing, 1996”, Philadelphia, Pennsylvania (ACM Press, New York, 1996), pp. 212–218.

[2] L. K. Grover, “Quantum Mechanics Helps in Searching for a Needle in a Haystack”, Phys. Rev. Letter Vol 79, pp. 325-328 (1997).

[3] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM J. Comput. 26, 1510 – 1524 (1997).

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computing and Quantum Information* (Cambridge University Press, Cambridge, U.K., 2000)

[5] P. W. Shor, *Proceedings of the Symposium on the Foundations of Computer Science*, 1994, Los Alamos, California (IEEE Computer Society Press, New York, 1994), pp. 124–134.

[6] L. K. Grover, Phys. Rev. Letter Vol 79, pp 4709-4712 (1997).

[7] Norbert Schuch and Jens Siewert,“ Programmable Networks for Quantum Algorithms” ,Phys. Rev. Letter Vol 91, pp 027902 (2003).

[8] D. E. Knuth, *Fundamentals of Algorithms: The Art of Computer Programming* (Addison-Wesley, Reading, MA, 1973)

[9] W. Feller, *An Introduction to Probability Theory & Its Applications* (John Wiley, New York, 1971), Vols. I and II.

[10] Adam Miranowicz, Kiyoshi Tamaki, *Math. Sciences (Suri-Kagaku)* Vol **473**, pages 28 (2002).

[11] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, e-print quant-ph/9605034. *Proceedings of 4th Workshop on Physics and Computation (PhysComp '96)*, Boston, MA, pp. 36 – 43 (1996).

[12] K.-A. Brickman, P. C. Haljan, P. J. Lee, M. Acton, L. Deslauriers, and C. Monroe, " *Implementation of Grover's quantum search algorithm in a scalable system"* Phys. Rev. Letter A Vol 72, pages 050306 (2005)

[13] Anocha Yimsiriwattana and Samuel J. Lomonaco Jr., "*Generalized GHZ States and Distributed Quantum Computing*" e-print quant-ph/0402148 (2004).