

QUANTUM SEARCH: REDUCES THE TIME AND COST FOR SEARCHING FOR OBJECTS IN MULTIPLE-SERVER PEER-TO-PEER NETWORKS

Phaneendra H.D. , Dr. M.S. Shivakumar

Assistant Professor, Department of Computer Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India

Principal, The National Institute of Engineering, Mysore, Karnataka, India

Abstract: In number of Internet applications we need to search for objects to down load them. This includes peer-to-peer (P2P) file sharing, grid computing and content distribution networks. Here the single object will be searched for in multiple servers. There are many searching algorithms existing today for this purpose and uses the concept of classical physics and classical algorithms. The principles of quantum mechanics can be used to build and analyze a quantum computer and its algorithms. Quantum searching is one such algorithm. In this paper we are proposing a search method based on quantum physics and quantum algorithms.

Keywords: Quantum mechanics, Quantum algorithm, qubits, Quantum search, linear search, object.

1. INTRODUCTION

1.1 Quantum computation

Quantum Computation is the field of study, which focused on developing computer technology based on the principles of quantum theory. The aim of this paragraph is to make computer scientists to go through the barriers that

separate quantum computing from conventional computing. We have introduced the basic principles of quantum computing. It is important for the computer science community to understand these new developments since they may radically change the way we think about computation, programming, and complexity [1]. The basic variable used in quantum computing is a qubit, represented as a vector in a two dimensional complex Hilbert space where $|0\rangle$ and $|1\rangle$ form a basis in the space. The difference between qubits and bits is that a qubit can be in a state other than $|0\rangle$ or $|1\rangle$ whereas a bit has only one state, either 0 or 1. It is also possible to form linear combination of states, often called superposition. The state of a qubit can be described by

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (1)$$

The numbers α and β are complex numbers. The special states $|0\rangle$ and $|1\rangle$ are known as computational basis states. We can examine a bit to determine whether it is in the state 0 or 1 but we cannot directly examine a qubit to determine its quantum state, that is values of α and β . When we measure a qubit we get either the result 0, with probability $|\alpha|^2$ or the result 1, with probability $|\beta|^2$, where $|\alpha|^2 + |\beta|^2 = 1$, since the probabilities must sum to one. Consider the case of two qubits. In two classical bits there would be four possible states, 00, 01, 10 and 11. Correspondingly, a two qubit system has four computational basis states denoted $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. A pair of qubits can also exist in a superposition of these four states, which is given as

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \quad (2)$$

The logic that can be implemented with qubits [6].

1.2 Quantum algorithms

Quantum algorithms are based on the principles of quantum mechanics. They are different from classical computing in two specific features: superposition and entanglement. Superposition can transfer the complexity of the problem from a large number of sequential steps to a large number of coherently superposed quantum states. Entanglement is used to create complicated correlation's that permit the desired interference.

A typical quantum algorithm starts with a highly superposed state, builds up entanglement, and then eliminates the undesired components providing compact results. In classical systems, the time taken to do certain computations can be decreased by using parallel processors. To achieve an

exponential decrease in time, it requires an exponential increase in the number of processors, and hence an exponential increase in the amount of physical space. However, in quantum systems the amount of parallelism increases exponentially with the size of the system. Thus, an exponential increase in parallelism requires only a linear increase in the amount of physical space. This property is called quantum parallelism [6][2][5].

For example Traveling salesman problem can be solved with $O(\sqrt{N})$ operations using quantum algorithm, which requires $O(N)$ operations in classical algorithm.

The quantum search algorithm is general in the sense that it can be applied far beyond the route finding example just described to speed up many (though not all) classical algorithms that use search heuristics. Thus given a search space of size N , and no prior knowledge about the structure of information in it, if we want to find an element in search space satisfying a known property, then this problem requires approximately N operations, but the quantum search algorithm allows it to be solved using approximately \sqrt{N} operations [5][6].

2. PEER-TO-PEER NETWORKS

Efficiently looking for a single object in multiple servers is fundamental to many emerging applications on the Internet. For example, in peer-to-peer (P2P) file sharing a single file is searched for in multiple “servent” nodes that act as servers. A servent is a node in a P2P network having both server and client capabilities. An efficient search for the object returns with the location of the searched object quickly and with a low cost. Cost can be measured as the total server utilization per search [4].

One of the methods for an efficient search is to maintain a centralized directory of all objects. A centralized directory is the approach Napster [4] used for P2P file sharing in the Internet. However, for reasons of robustness distributed solutions are also typically sought.

Another and simple approach is a fully distributed search based on a broadcast search. Here the search query is broadcasted to all servers. Gnutella uses this broadcast approach in an overlay network on the Internet. The time to find an object is small, however the cost is significant – all Gnutella servents are queried independently of the likelihood of (the servent) having the searched object. It has been shown that broadcast based search does not scale well for systems with many servers [8]. As the number of servers, N , in a system increases linearly, the load on each server increases exponentially. Currently, at least 25% of the traffic in the Internet is P2P file

query related [4]. Thus, to reduce load on P2P nodes and reduce traffic in the Internet new ideas in searching in P2P networks are needed [4].

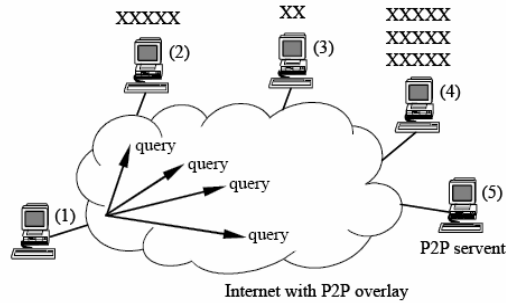


Figure 1. P2P network with stored objects

Here X means shared object. Node (1) is for sending queries. Node (4) has the most objects stored. Nodes (2) and (3) are also stored with some objects. Nodes (1) and (5) are free riders

Characterization of P2P networks has shown that connectivity of servents to other servents follows a power law where very few servents are high degree and the majority of servents are low degree [9]. File sharing also follows a power-law where few servents contain the majority of files shared (and many servents may share no files at all and are so called "free riders") [4]. These Characterizations clearly show that connectivity and file distribution between servents are not uniformly distributed. Hence exploiting these characteristics can result in greater search efficiency.

3. THE QUANTUM SEARCH METHOD

In this section we show that how quantum search is possible looking for a single object. We develop a quantum search method for finding objects. The quantum search method exploits superposition and quantum parallelism and a non-ordered distribution of objects in nodes to achieve an efficient search in terms of search time and cost. Figure. 1 shows a P2P network with multiple servents that stores the objects.

The object required can be obtained in three different methods. They are

1. Centralized directory of all objects
2. Centralized directory of all objects along with server names.
3. The directory of objects in various servers.

In the first method the server contains a directory of all objects. This directory is searched for required object, if found can be downloaded. The query for this purpose directly searches this directory. The object names

stored in this directory could be an unordered list. The quantum search works onto an unordered list efficiently, by searching for the object in the order of \sqrt{N} , where N is the number of objects in the server.

Here linear search on the directory is possible, because the directory with object names is an unordered list (other searching methods requires object names to be stored in some order, for example binary search requires list must be sorted in ascending order). The linear search takes $O(N)$ steps to search for an object at the worst case and $O(N/2)$ in an average . Therefore quantum search is better than the linear search, and object can be added the server and object names can be added to the directory without sorting.

The objects can be searched with only \sqrt{N} steps rather than $O(N/2)$ steps with a single query.

This method suffers with scalability, where it requires more space to store objects and the number of entries to the directory increases with more number of objects.

The second method, again the server contains a directory of all objects, but the difference is each entry in the directory contains two columns with entries (object name, server name). The first query is made to the server with object name required, if found, will give the server name, where exactly the object is available. Now another query can be sent directly to the server, where the object is present, quantum search for it, and can be downloaded.

The required object can be searched with two queries with \sqrt{N} steps in each query, rather than $O(N)$ steps.

This method also overcomes the problem of scalability, where objects can be added to the servers in the second level, along with an entry to the directory in the server which maintains the centralized directory of all objects. The cost here is it requires two queries to search for an object.

In the last method the directory of objects along with objects are maintained in various servers. The query with desired object is broadcasted to all servers. Quantum search of each directory in each server is carried out independently and simultaneously. The server which contains the desired object responds after quantum searching its directory.

The required object can be searched with only one query with $O(\sqrt{N})$ steps, rather than $O(N)$ steps.

This method further overcomes the problem of scalability, where more servers can be added to the network. When an object is added to the server, the entry for this object should be made only in the directory of the respective server

4. ALGORITHMS FOR THESE METHODS

The algorithms for searching for an object using the above three methods are as follows.

Method 1:

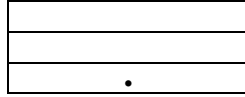


Figure 2. Directory with object names

Step 1: Send a query to the server, which maintains the centralized directory of all objects (Figure 2).

Step 2: Quantum search the directory

Step 3: If the object is found in the server then download the object.

Method 2:

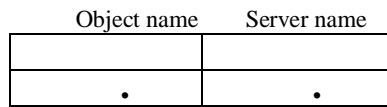


Figure 3. Directory with object and server names

Step 1: Send a query to the server, in the first level, which maintains the centralized directory of all objects (Figure 3).

Step 2: Quantum search the directory

Step 3: If object is found, obtain the server name where it is available

Step 4: Send query again to the server name obtained in step 3

Step 4: If the object is found in the server then download the object.

Method 3:

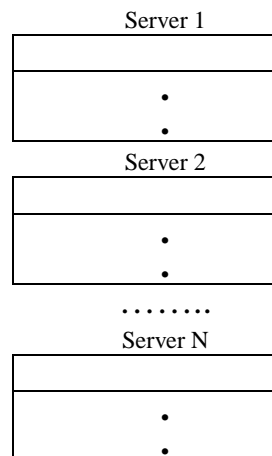


Figure 4. Directory with object names in each of the N servers

Step 1: Broadcast a query to all N servers.

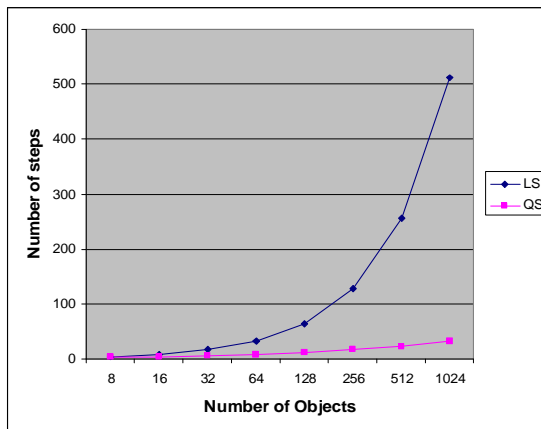
Step 2: Quantum search the directory in each server independently and simultaneously (Figure 4).

Step 3: If the object is found, then download the object from the respective server.

5 PERFORMANCE EVALUATION OF QUANTUM SEARCH

Table 1. Comparison between Linear and Quantum Search

Number of Objects(N)	Linear Search (Unordered list)	Quantum Search (Unordered list)
8	4	3
16	8	4
32	16	6
64	32	8
128	64	11
256	128	16
512	256	22
1024	512	32



Graph 1

Here the table 1 gives the comparison between use of linear search and quantum search for searching the objects. The graph shows how quantum

search out performs the linear search, searching in an unordered list of objects.

Method 1 requires single query with \sqrt{N} steps, to find an object.

Method 2 requires two queries with \sqrt{N} steps for each query, to find an object

Method 3 requires a broadcasted query with \sqrt{N} steps to find an object

6. CONCLUSION

In this paper, we studied and analyzed quantum search algorithm based on quantum mechanics, by applying to multiple-server peer-to-peer networks searching for an object. This algorithm works on unsorted list of objects, and provides a quadratic speed-up and the desired object is located with $O(\sqrt{N})$ steps with the best case. Theoretically it can be concluded that quantum search algorithm provide fast results by taking the help of quantum mechanics concepts like quantum parallelism and superposition. The consensus is that looking at the research in quantum computation and quantum information, quantum search could supplement the classical search problems.

REFERENCES

1. Apoorva Patel, Quantum Database Search can do without Sorting, quant-ph/0012149.
2. Graciela perera, Ken Christensen and Allen Roginsky, Targeted search: Reducing the Time and Cost for Searching for Objects in Multiple-Server Networks, IEEE transaction, 2005, pp 143-149.
3. Grover L.K, (1996), A Fast Quantum mechanical Algorithm for Database Search, In proceedings of the 28th Annual ACM Symposium on the Theory of Computing, pp.212-219, quant-ph/9605043.
4. Nielsen M and Chaung, I (2000), Quantum Computation and Quantum Information, Cambridge University press, Cambridge, United Kingdom.
5. S. Saroiu, P. Gummadi, and S. Gribble, A measurement Study of Peer-to-Peer File Sharing Systems, Proceedings of SPIE, pp. 156-170, 2001.
6. R. Schollmeier and G. Schollmeier, Why Peer-to-Peer (P2P) Does Scale: An Analysis of P2P Traffic Patterns, Proceedings of the 2nd International Conference on Peer-to-Peer Computing, pp. 112-119, 2002.
7. B. Yang, P. Cao, and H. Molina, Efficient Search in Peer-to-Peer Networks, Proceedings of the International Conference on Distributed Computing Systems, pp. 5-14, 2002.
8. Terry Rudolph and Dr. Lov Grover, (2002), Quantum Searching a classical database, quant-ph/0206066.