

Research Article

A Novel Algorithm of Quantum Random Walk in Server Traffic Control and Task Scheduling

Dong Yumin¹ and Xiao Shufen²

¹ Network Center, Qingdao Technological University, Qingdao 266033, China

² College of Automobile and Transportation, Qingdao Technological University, Qingdao 266033, China

Correspondence should be addressed to Dong Yumin; dym1188@163.com

Received 18 January 2014; Accepted 19 February 2014; Published 7 April 2014

Academic Editor: Feng Gao

Copyright © 2014 D. Yumin and X. Shufen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A quantum random walk optimization model and algorithm in network cluster server traffic control and task scheduling is proposed. In order to solve the problem of server load balancing, we research and discuss the distribution theory of energy field in quantum mechanics and apply it to data clustering. We introduce the method of random walk and illuminate what the quantum random walk is. Here, we mainly research the standard model of one-dimensional quantum random walk. For the data clustering problem of high dimensional space, we can decompose one m -dimensional quantum random walk into m one-dimensional quantum random walk. In the end of the paper, we compare the quantum random walk optimization method with GA (genetic algorithm), ACO (ant colony optimization), and SAA (simulated annealing algorithm). In the same time, we prove its validity and rationality by the experiment of analog and simulation.

1. Introduction

The server cluster technology may be connecting multiple independent servers, and, in the same time, it must provide services as a whole by a cluster. In the server cluster, how to solve the problem of server traffic control and task scheduling is very important.

In order to reduce the access time, optimize the overall performance and achieve parallel program in a high efficiency; the task request must be allocated to each on the server. So, load balancing mechanism is the core of cluster technologies.

In the literature [1], it expands the analogies employed on the development of quantum evolutionary algorithms by putting forward quantum-inspired Hadamard walks, called QHW. In order to solve combinatorial optimization problems, a quantum evolutionary algorithm, abbreviated HQEA, is proposed. From the results of the experiments carried out on the knapsack problem, HQEA performs noticeably better than a conventional genetic algorithm, in terms of convergence speed and accuracy. The literature [2] explores how a spectral technique suggested through coined quantum

walks can be used to differentiate between graphs which are cospectral as for standard matrix representations. This algorithm runs in polynomial time; it can differentiate many graphs for there is no subexponential time algorithm which is proven to be able to differentiate between them.

By the literature [3], they propose a quantum algorithm to evaluate formulas by an extended gate set, including two- and three-bit binary gates. This algorithm is more optimal in read, once formulas for that each gate's inputs are balanced in a certain sense. It describes a very compact triaxial instrument in the literature [4]. The triaxial instrument is based on a rhombic dodecahedral geometry that can accommodate three nonplanar ring light paths with orthogonal sensing axes. Component count can be substantially reduced by a discharge of layout to use a single cathode and two anodes running all three axes in balanced plasma currents. Two Monte Carlo-based approaches to assess parameter uncertainty with complex hydrologic models are considered in the literature [5]. The importance sampling has been carried out in the generalized likelihood uncertainty estimation framework by Beven and Binley. The metropolis algorithm is different from importance sampling which uses a random

walk that adapts to the real probability distribution describing parameter uncertainty.

Because existing search protocols for unstructured peer-to-peer systems to create huge burden on communications or cause long response time and result in unreliable performance. In the literature [6], in order to discover service providers, it reports that an important function of a peer-to-peer system is a distributed message relaying. They present an incentive mechanism which not only relieves the free-riding problem but also accomplishes good system efficiency in message relaying for peer discovery. The passed along message propagation process is promised rewards in the mechanism.

In the literature [7], it analyzes the discrete-time quantum walk by separating the quantum evolution equation into Markovian and its interference terms. Because of this separation, it is possible to show analytically which quadratic increase in the variation of the position of quantum walker with time is a direct aftermath of the coherence of the quantum evolution. As expected, the variation is shown to increase linearly with time, if the evolution is decoherent, as in the classical case. Moreover, it shows that the system has an evolving operator analogous to which of a resonant quantum kicked rotor. At the same time, the rotator can be described by evolution of the quantum walker.

Quantum random walks on a graph, which is analogous to classical stochastic walk, form the basis for many of the recent quantum algorithms that promise to obviously outperform existing classical random walk algorithms. A number of studies have been done on the many applications of quantum random walk to some important computing problems. There are two kinds of quantum random walk algorithms: continuous-time and discrete-time. It is reported that a quantum arithmetic is defined by a sequence of the operations that runs on an actual model of quantum computation in the literature [8]. It proposes quantum circuit designs for both kinds of random walk algorithms which operate on various graphs. It considers two important problems to which random walk arithmetic are applicable: the triangle finding problem and binary tree problem. Because of it a few research works that are related to quantum random walk circuit design on graphs exist; the circuit designs they present here are the first of their kind. At the same time, they also provide an estimate of the quantum cost of the circuits of quantum systems. And it is based on the number of execution cycles and quantum operations.

In the literature [9], the natural random walk causing Brownian motion occurs to be always biased in a very delicate way: emphasizing some possibilities by only approximative maximal uncertainty principle. It introduces a new method of stochastic model, and they use the merely maximizing entropy choice of transition probabilities.

Berry and Wang show numerically that a discrete time quantum random walk of two irrelevant particles is able to differentiate some nonisomorphic powerfully regular graphs from the same household in the literature [10]. They analytically show how it is possible for the walks to differentiate such graphs, while the continuous time quantum walks of two irrelevant particles cannot.

It is reported that the quantum walks are quantum mechanical theory analog of random walks in the literature [11]. By traversing the edges of a graph, a quantum “walker” progresses between initial and final states. They present a hybrid model for general quantum computing in which a quantum walker gets discrete steps of continuous evolution.

Effective server traffic control can extend the “capacity” of the server, and the task scheduling can improve system throughput. In early research methods of it, such as Min-Min algorithm, Max-Min algorithm, genetic algorithm (GA), round robin (RR), simulated annealing algorithm (SAA), dynamic feedback algorithm (DFA), and ant colony optimization algorithm (ACO). These arithmetics have some improvements in different degree on the task scheduling.

But these algorithms have this or that problem, such as local premature problem and divergence problem.

In order to overcome the instability above the algorithms, the quantum random walk algorithm is proposed, and it is proved better than above GA, ACO, and SAA by simulation experiments.

2. Quantum Random Walk

2.1. Random Walk. Random walk is a mathematical method to study the formation of trajectory by a random sequence of continuous; it is not only a means to study mathematics but also a basic tool in the natural sciences. Any stage of the random walk behavior is not limited to previous history of migration; the process is also called Markov process. Random walk can be simply described as follows.

Suppose in a straight line, there is a moving particle; it is at the origin to move left or right one unit of distance; the probability is p and $q = 1 - p$, respectively, each time the particle in accordance with the probability to move a unit distance to the left or right. Here we assume that the probability of the particle is equal to the left or right; that is, $p = q = 1/2$; random variable can be used to represent the probability σ_i ; its value is as follows:

$$\sigma_i = \begin{cases} -1, & p = \frac{1}{2} \\ 1, & q = \frac{1}{2}. \end{cases} \quad (1)$$

If the particles every moment in a straight line position constitute an independent identical distribution of random variables sequence, denoted $\sigma_1, \sigma_2, \dots, \sigma_t, \dots$ is a sequence of independent and identically distributed variables on meet $EX_1 = 0$, $\text{Var}(\sigma_1) = 1$. $S_t = \sum_{i=1}^t \sigma_i$ is its first n terms and $S = (S_t)_{t \in T}$ is called a random walk. After the particles move n steps, the probability of it being found in position m is

$$p_{0,m}^n = \binom{n}{\frac{n+m}{2}} q^{(n+m)/2} p^{(n-m)/2}. \quad (2)$$

Among them, $(n + m)/2$ take only integer, $(n + m)/2 \in 0, 1, \dots, n$; other cases were 0.

Although the classical random walk has a broad application, but compared with the quantum random walk, it feels much ashamed of its inferiority.

Quantum random walk is a quantum computing model proposed in recent years; scholars have also become increasingly interested in research.

2.2. Quantum Random Walk. For discrete quantum random walk, the system added an extra degree of freedom; some literature defines it as chirality that can build an adaptation to global local unitary process. This walk is also called quantum Hadamard walk. The only possible remained unchanged in the global process of unitary transformation is the only mobile operator between adjacent lattice points to the left or the right.

Significantly different between quantum Hadamard random walk and classical quantum is the interference, the diffusion rate of quantum walk square magnitude faster than classical square. Due to the existence of quantum superposition states, in quantum random walk, position of the particle from the probability distribution may be seen; particles may be in several locations simultaneously with different probability. Quantum random walk process is accomplished by a unitary matrix transformation [12].

2.2.1. One-Dimensional Quantum Random Walk. The random walk model sets up corresponding quantum algorithms and does quantum information processing. People commonly used coined quantum walks; it corresponds to Hilbert space which can be expressed as follows:

$$H = H_C \otimes H_V, \tag{3}$$

where $H_V = \text{span}\{|v_i\rangle\}_{i=1}^{|V|}$ is the random walk of grid space; it corresponds to a classic case of d -degree regular graphs $G(V, E)$, $H_C = C^d$, which is a coin flip operator space (coin space). The total unitary evolution matrix U is by the two independent parts, namely, flipping a coin and conditional replacement

$$U = S \cdot (C \otimes I). \tag{4}$$

The first step of the quantum walk is to perform a rotation operation C in coin space, equivalent to the classical random walk in a coin toss, through this operation to get a coin superposition state [13]. Then, the replacement operator S makes the particles by a coin to decide an edge vertex adjacent to move to the next. Starting from the initial state $|\psi(0)\rangle$, repeat the walk after n steps and obtain the probability distribution of each vertex as follows:

$$P(v, n) = {}_v \langle v | N r_c [|\psi(n)\rangle \langle \psi(n)|] | v \rangle_v. \tag{5}$$

Quantum random walks of a variety of ways common are one-dimensional linear walk, ring walk, hypercube walk, and so forth. For one-dimensional linear walk, $H_V = \text{span}\{|x\rangle : x \in Z\}$, $H_C = \text{span}\{|R\rangle, |L\rangle\}$, replacement operator applied to the base is expressed as

$$S|R, x\rangle = |R, x + 1\rangle, S|L, x\rangle = |L, x - 1\rangle. \tag{6}$$

Starting from the initial state $|\Phi_{in}\rangle = |\downarrow\rangle \otimes |0\rangle$, continuous action $U = S \cdot (C \otimes I)$; after each step, the distribution of every point is as follows:

$$\begin{aligned} |\Phi_{in}\rangle &\xrightarrow{U} \frac{1}{\sqrt{2}} (|\uparrow\rangle \otimes |1\rangle - |\downarrow\rangle \otimes |-1\rangle) \\ &\xrightarrow{U} \frac{1}{2} [(|\uparrow\rangle \otimes |2\rangle - (|\uparrow\rangle - |\downarrow\rangle) \otimes |\downarrow\rangle \otimes |-2\rangle) \\ &\xrightarrow{U} \frac{1}{2\sqrt{2}} [(|\uparrow\rangle \otimes |3\rangle + |\downarrow\rangle \otimes |1\rangle + (|\uparrow\rangle - 2|\downarrow\rangle) \\ &\quad \otimes |-1\rangle - |\downarrow\rangle \otimes |-3\rangle)]. \end{aligned} \tag{7}$$

Not only is the distribution situation different with the classical random walk, but one-dimensional linear walk is also higher and faster than the classical random walk in the diffusion velocity.

2.2.2. M-Dimensional Hypercube Quantum Random Walk. For the M -dimensional hypercube quantum random walk, it has 2^m vertices each vertex can be marked by an m binary string,

$H_V = \text{span}\{|\vec{x}\rangle : x \in [0, 2^n]\}$. Each vertex is m degrees; therefore, coin space $H_C = C^c$, using $|d\rangle$ to mark the coin space basis vectors, $d \in [0, n]$; it indicates the direction of the next step; each $|d\rangle$ corresponds to a M -dimensional vector $|\vec{e}_d\rangle$,

$$|\vec{e}_d\rangle = \left| \begin{matrix} 0 \cdots 0 1 0 \cdots 0 \\ (d-1) \quad (n-d) \end{matrix} \right\rangle. \tag{8}$$

$|\vec{e}_d\rangle$, the number of d bit is 1 and that of the other bits is 0. The replacement operator walk on the hypercube is applied to the base vectors and can be expressed as

$$S|d, \vec{x}\rangle = |d, \vec{x} \oplus \vec{e}_d\rangle. \tag{9}$$

It represents two points marked by M quantum bits; only when they have only one bit is not the same when they are connected directly via a side (e.g., 001101 and 011101 are communicating). There are many ways to select hypercube walking coin flip operators; however, the following form is usually taken to maintain certain symmetry:

$$C_{\alpha, \beta} = \begin{pmatrix} \alpha & \beta & \beta & \dots & \beta & \beta \\ \beta & \alpha & \beta & \dots & \beta & \beta \\ & & & \vdots & & \\ \beta & \beta & \beta & \dots & \alpha & \beta \\ \beta & \beta & \beta & \dots & \beta & \alpha \end{pmatrix}. \tag{10}$$

The operator of such forms has the characteristics that in all directions are a permutation invariant; it retains hypercube replacement invariance. Such a form is a commonly used Grover diffusion operator selection

$$G = -I_n + 2|s^c\rangle \langle s^c|. \tag{11}$$

Among them, $|s^c\rangle = 1/\sqrt{m} \sum_{d=1}^m |d\rangle$ are equally weighted superposition states in all directions. Grover operator is one

of the permutation invariant operators, which is the farthest with unit transformation I ; It will effectively mix all of the any given initial state into the superposition of them. The total random walk evolution operator can be expressed as

$$U = S \cdot (G \otimes I). \quad (12)$$

Define the Hamming distance, starting from any point to another point of the minimum number of edges experience (i.e., the required number of steps), with $d_h = 1$. Marking each vertex string the number of "1" is called Hamming weights, for example, Hamming weight 010 to 1. Starting from $00 \cdots 0$, any Hamming weighing the same point total can reach at the same number of steps. When the coins are formed to the symmetry type such as (10), it has the same probability starting from this point to reaching the point with the same weight. This allows us to put all Hamming weight of the same point "accumulation" to a point, thereby reducing the symmetry of the random walk; the walk on the hypercube is becoming walk in a straight line. It is noteworthy that this walk difference on the straight is not unbiased; it differs from the previously discussed one-dimensional linear walk. The number of total vertices after walking on the variable linear is $m + 1$; m is the hypercube dimension.

By putting a hypercube walk into walk on straight line, many problems can be resolved to simplify and get results. Moor and Russell found that when $T = \pi \cdot m/4$, random walk is a balanced distribution. Kempe through the research on the hitting time found hypercube quantum random walk to reach vertical angles of time relative to the classic case which is an exponential acceleration; this shows that the quantum random walk has the potential to make quantum algorithm acceleration.

3. The Model of Server Traffic Control and Task Scheduling

In cluster services, the task scheduling can be described as follows: N tasks need to be allocated to m nodes (these nodes are the servers) with different handling capacity; the goal is finding an optimization schedule to minimize the total completion time. The system model is shown as follows.

We suppose there are m nodes (or servers) and n tasks. Every task should be assigned to only one node. We use $P = \{p_1, p_2, \dots, p_m\}$ denoting the nodes (or servers), in this paper, where p_i denotes one of the nodes (or servers); $L = \{l_1, l_2, \dots, l_m\}$ expresses the current load, where l_i expresses the current load of node p_i . For instance, $l_i = 0$ means that the node (server) p_i has a current load of 0; in other words, the node is idle. Here n tasks are expressed by $X = \{x_1, x_2, \dots, x_n\}$, where x_j is one of the tasks. A $m \times n$ matrix is built between servers and tasks: W_{mn} , where W_{ij} is one of the elements. So, there are two states as follows:

$$W_{ij} = \begin{cases} 1 & \text{Task } x_i \text{ is assigned on node } p_j \\ 0 & \text{Task } x_i \text{ is not assigned on node } p_j, \end{cases} \quad (13)$$

where, $i \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, n\}$.

We use t_{ij} to express the time of processing on one task; in other words, the time of task x_j processed on node p_i . The processing time is denoted as follows:

$$T_{ij} = \begin{cases} t_{ij} & \text{Task } x_i \text{ is processed on node } p_j \\ 0 & \text{Task } x_i \text{ is not processed on node } p_j, \end{cases} \quad (14)$$

where, $i \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, n\}$.

It is not difficult to see that T_{ij} is also an $m \times n$ matrix.

Here, we define the optimal state occurring with these conditions: (a) the total system has a relative short time of processing; (b) the throughput of system is relatively larger in unite time. We can describe this state using the following equations:

$$Y_{\max} = \sum_i^m \omega(x_i, l_i, q_i) \quad (15)$$

$$\omega(x_i, l_i, q_i) = c_1 \left(\left(\sum_{i=1}^m \sum_{j=1}^n t_{ij} + \sum_{i=1}^m q_i t_i \right) - c_2 \sum_{i=1}^m l_i \right)^2,$$

where $X = \{x_1, x_2, \dots, x_n\}$ is the new task, $L = \{l_1, l_2, \dots, l_m\}$ is the current total load at the node, q_i is the length of ready queue at node p_i , t_i is the average processing time at node p_i , c_1, c_2 are constant, $\omega(x_i, l_i, q_i)$ is a function which can show the ability of node processing. The system is on the optimal running state, when the capacity of processing tasks (or task scheduling) reaches the maximal matching at one node.

4. The Method of Task Scheduling Based on Quantum Random Walk (QRW) Clustering Algorithm

In the paper, we mainly research the standard model of one-dimensional quantum random walk. For the data clustering problem of high dimensional space, we can decompose one m -dimensional quantum random walk into m one-dimensional quantum random walk.

4.1. Clustering Algorithm Based on One-Dimensional Quantum Random Walk, Referred to as Quantum Random Walk Clustering Algorithm (QRWA)

Step 1. Assume an unlabeled data set $X = \{X_0^1, X_0^2, \dots, X_0^n\}$, where each data point with m features.

Step 2. Each data point in the data set can be considered as a particle that transfers in the space according to the probability.

Step 3. Establish clustering algorithm based on the one-dimensional quantum random walk.

The clustering algorithm uses a distributed control strategy, that is, each data point of the data set only affected by its neighbor within the neighborhood. The neighbor of data points available k -nearest neighbor method or the method of default scope of R to determine and use $\Gamma_t(i)$ indicates the set of neighbors of a data point X_t^i in t time.

Step 4. Calculate the probability for each data point transfer to all neighbors in the neighborhood $p_t(i, j), j \in \Gamma_t(i)$; the formula is as follows:

$$p_t(i, j) = \begin{cases} \frac{a_t(i, j)}{\sum_{j \in \Gamma_t(i)} a_t(i, j)} & \text{if } j \in \Gamma_t(i) \\ 0 & \text{otherwise} \end{cases}$$

$$a_t(i, j) = \left(\left(\frac{\text{Deg}_t(j)}{\sum_{j \in \Gamma_t(i)} \text{Deg}_t(j)} \right) \times \left(\frac{\text{Deg}_0(j)}{\sum_{j \in \Gamma_t(i)} \text{Deg}_0(j)} \right) \right) \times \left((d(X_t^i, X_t^j)) \times (d(X_0^i, X_0^j)) \right)^{-1}. \quad (16)$$

Among them, $\text{Deg}_t(\cdot)$ and $\text{Deg}_0(\cdot)$, respectively, represent the degree of current and initial data points; similarly, $d(X_t^i, X_t^j)$ and $d(X_0^i, X_0^j)$, respectively, represent the current and initial data point distances between X^i and X^j .

Step 5. Find the maximum transition probability $p_t(i, h)$ and the neighbors of greatest probability of metastasis $X_t^h, h \in \Gamma_t(i), h \neq i$.

4.2. The Server Traffic Control Clustering Method of Quantum Random Walk. As previously mentioned, m -dimensional quantum random walk is decomposed into m one-dimensional quantum random walk. For each dimension, data points X_t^i can only move a step left or right, l_L or l_R . Therefore, the maximum transition probability $p_t(i, h)$ is mapped to the interval $\rho = f(p_t(i, h)) \in [0.5, 1]$, so, the probability of transfer in the opposite direction is $1 - \rho$. When $\rho = 0.5$, and probability of metastasis in both directions is equal, available aforementioned Hadamard transforms H as a coin matrix. However, in normal conditions, $\rho \neq 1 - \rho$, therefore, the coin matrix C is used in the algorithm is

$$C = \begin{pmatrix} \sqrt{\rho} & \sqrt{1-\rho} \\ \sqrt{1-\rho} & -\sqrt{\rho} \end{pmatrix}, \quad (17)$$

$$\rho = f(p_t(i, h)) = f\left(\max_{j \in \Gamma_t(i)} (p_t(i, j))\right).$$

It is easy to verify that the matrix C is unitary matrix meeting the reversibility requirements of quantum mechanics.

Since the quantum random walk clustering algorithm will use r consecutive transformation $U = S \cdot (C \otimes I)$ for initial state, thus, every time it changes left and right transfer step l_L and l_R to the original $1/2$, that is, l_L/r and l_R/r . Then, conditional operator S will press type structure:

$$S = |\uparrow\rangle \langle \uparrow| \otimes \sum_b \left| \frac{b+l_R}{r} \right\rangle \langle b+l|\downarrow\rangle \langle \downarrow| \otimes \sum_b \left| \frac{b-l_L}{r} \right\rangle \langle b|. \quad (18)$$

As is known, in quantum mechanics, each one of superposition states can be seen as a position of particle and indicates the probability of finding the particle at this location. If repeatedly used U transforms for initial state, then the resulting superposition state $|\psi\rangle$ will contain more items; this increases possible appearing position of the particle, and this is not present in the classical random walk. It is these possible positions that increase the searching range of solution space and provide an opportunity for better results. To calculate the probability of multiple locations of particles and their appearance in the corresponding locations, a unitary operation is sufficient because of the quantum parallelism, but in the classical world, you need multiple operations to complete; it also reflects an aspect of quantum computing to accelerate the classical computing.

If the initial state of the particle is $|\psi_0\rangle = |\uparrow\rangle \otimes |0\rangle$, then, after applying $r = 2$ times transform U , get superposition state $|\psi\rangle$ which is

$$|\psi_0\rangle \xrightarrow{U} \sqrt{\rho} |\uparrow\rangle \otimes \left| \frac{l_R}{r} \right\rangle + \sqrt{1-\rho} |\downarrow\rangle \otimes \left| -\frac{l_L}{r} \right\rangle$$

$$\xrightarrow{U} \rho |\uparrow\rangle \otimes |l_R\rangle + \sqrt{\rho(1-\rho)} |\downarrow\rangle \otimes \left| \frac{l_R-l_L}{r} \right\rangle$$

$$+ (1-\rho) |\uparrow\rangle \otimes \left| \frac{l_R-l_L}{r} \right\rangle$$

$$- \sqrt{\rho(1-\rho)} |\downarrow\rangle \otimes |-l_L\rangle = |\psi\rangle. \quad (19)$$

From (19) particles can be found not only with probability ρ^2 and $\rho(1-\rho)$ at the same time appear in the probability of l_R and l_L appearing in another new position $(l_R - l_L)/r$ could be $(1-\rho)$. At this point, if projection measurement of the superposition state $|\psi\rangle$, it will collapse to one of these three positions according to the probability; then, the component of particle in the j dimension is updated with the following formula:

$$X_{t+1}^i(j) = \begin{cases} X_t^i(j) + l_R & \text{After the measurement, if the position} \\ & \text{of the particle at } l_R \\ X_t^i(j) + \frac{(l_R - l_L)}{r} & \text{After the measurement, if the position} \\ & \text{of the particle at } \frac{(l_R - l_L)}{r} \\ (l_R - l_L) - l_L & \text{After the measurement, if the position} \\ & \text{of the particle at } -l_L \end{cases}$$

$$I : l_R = l_L = (X_t^h(j) - X_t^i(j)), \quad j \in \{1, 2, \dots, m\}.$$

$$II : \begin{cases} l_R = \rho \times (X_t^h(j) - X_t^i(j)), & j \in \{1, 2, \dots, m\} \\ l_L = (1-\rho) \times (X_t^h(j) - X_t^i(j)), & j \in \{1, 2, \dots, m\}. \end{cases} \quad (20)$$

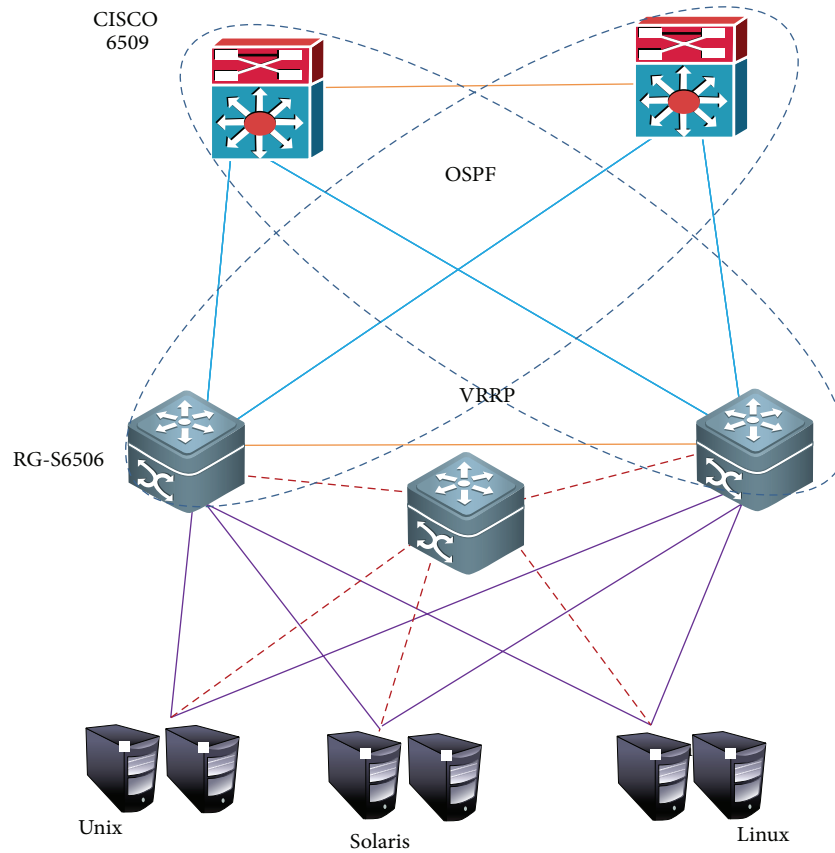


FIGURE 1: The topological structure of the network servers.

As data points are random walk in the space, its position and its nearest neighbor are constantly changing with time. Therefore, in the process of walking, the distance of the data points and the degree of it need to be recalculated. Repeat the entire process above until the sum of moving length of all the particles is less than some preset threshold ϵ . At this time, some separating section of the natural emergence in the space can be observed; each section corresponds a separate cluster.

5. Analog and Simulation Experiments

5.1. *The Experimental Environment.* In order to compare quantum random walk clustering algorithm (QRWA), genetic algorithm (GA), ant colony optimization (ACO), and simulated annealing algorithm (SAA), we select six servers as nodes. In the experiments, we select the number of task from 0 to 2500 (or 3000). We compare the results of these schemes by Matlab. The correlation parameters of selected servers for experiments are in Table 1.

In Table 1, OS represents operation system; NA represents network adapter; MS represents memory size; SM represents specifications and models.

The topological structure of the network servers is as shown in Figure 1.

5.2. *Results.* Figure 2 shows the system flow control rate of QRWA is better than GA, SAA, and ACO. And the more the

TABLE 1: Parameters of selected servers.

SM	Model CPU	MS	NA	OS
SUN SPARC Enterprise T5120	UltraSPARC T2 1.2 G	32.0 G	2 * 1000 M	Solaris
IBM System p5520Q	POWER5+ 2.66 G	32.0 G	4 * 1000 M	Linux
HP rx4640	intel Itanium 2. 1.5 G	32.0 G	2 * 1000 M	Unix

task quantity is, the closer the flow control rate is. The task quantity is from 0 to 2500.

Figure 3 shows the server traffic of GA, SAA, ACO, and QRWA. That is to say, the QRWA is bigger than GA, SAA, and ACO.

The Figure 5 shows that the throughput of QRWA is bigger than ACO, GA, and SAA.

The Figure 6 shows that the throughput of QRWA is bigger than ACO, GA, and SAA.

Figures 4, 5, and 6 show the performances of QRWA are better than GA, SAA, and ACO.

From the results, it is clear that quantum random walk algorithm (QRWA) is better in server traffic control and task scheduling than genetic algorithm (GA), simulated

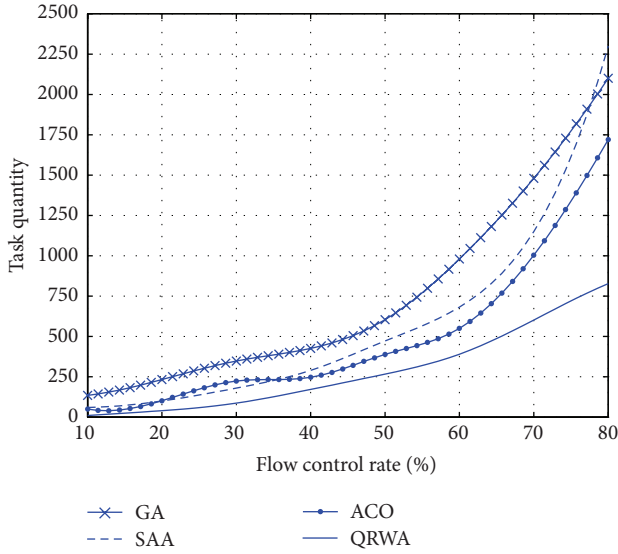


FIGURE 2: The flow control rate of SUN SPARC Enterprise T5120 in GA, SAA, ACO, and QRWA.

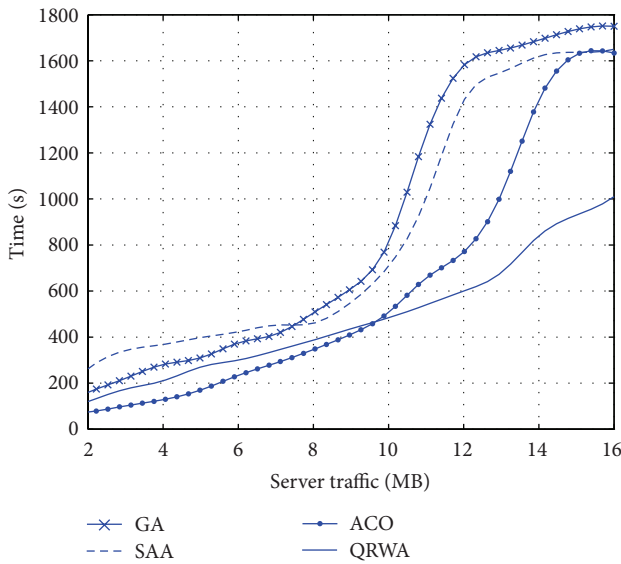


FIGURE 3: The server traffic of SUN SPARC Enterprise T5120 in GA, SAA, ACO, and QRWA.

annealing algorithm (SAA), and ant colony optimization (ACO). QRWA is more effective in task scheduling.

6. Conclusions

The paper gives a quantum random walk model and algorithm on server traffic control and task scheduling. We mainly research the standard model of one-dimensional quantum random walk. For the data clustering problem of high dimensional space, we can decompose one m -dimensional quantum random walk into m one-dimensional quantum random walk.

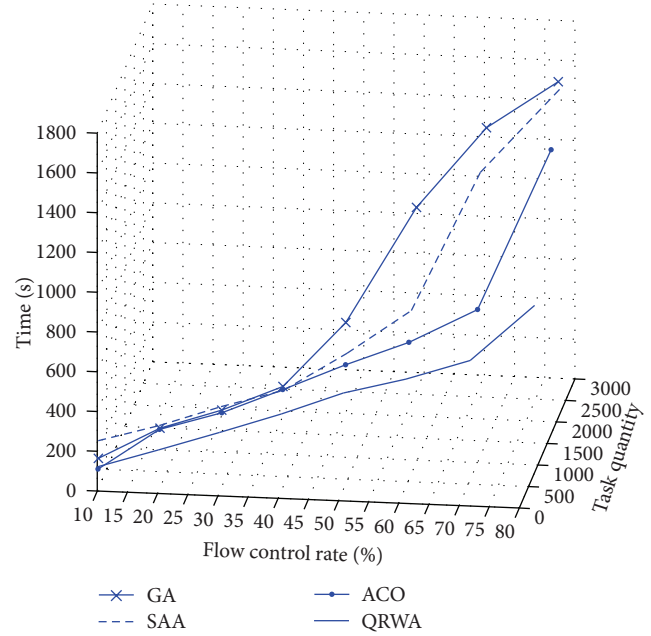


FIGURE 4: In SUN SPARC Enterprise T5120, the relationship of the flow control rate, task quantity and time. And the QRWA is better than GA, SAA, and ACO.

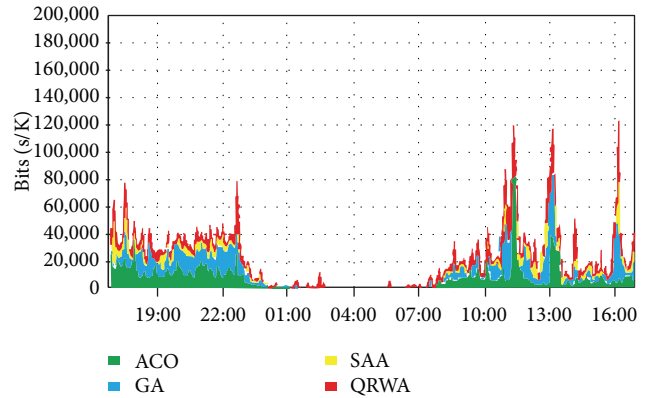


FIGURE 5: The throughput of HP rx4640 in ACO, GA, SAA, and QRWA during one day.

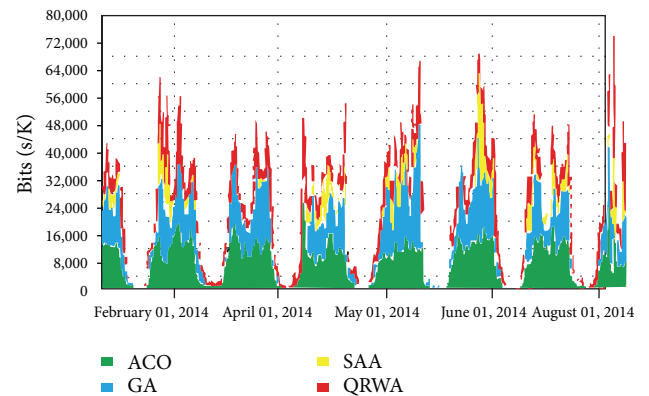


FIGURE 6: The throughput of IBM System p5520Q in ACO, GA, SAA, and QRWA during one week.

The simulation results demonstrate the effectiveness and superiority of QRWA.

The model and algorithm increases the throughput and efficiency of the system, and it had some merits than traditional model and arithmetics.

We will research the two directions in the future. The first one is the effects of noise on the scheme and model; the second one is the method of how to apply in the field of intelligence.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61173056).

References

- [1] S. Koppaka and A. R. Hota, "Superior exploration-exploitation balance with quantum-inspired Hadamard walks," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference (GECCO '10)*, pp. 2093–2094, Companion Publication, July 2010.
- [2] D. Emms, S. Severini, R. C. Wilson, and E. R. Hancock, "Coined quantum walks lift the cospectrality of graphs and trees," *Pattern Recognition*, vol. 42, no. 9, pp. 1988–2002, 2009.
- [3] B. W. Reichardt and R. Špalek, "Span-Program-based quantum algorithm for evaluating formulas," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pp. 103–112, May 2008.
- [4] G. J. Martin, S. C. Gillespie, and C. H. Volk, "Litton 11 cm triaxial zero-lock gyro," in *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS '96)*, pp. 45–55, April 1996.
- [5] G. Kuczera and E. Parent, "Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the Metropolis algorithm," *Journal of Hydrology*, vol. 211, no. 1-4, pp. 69–85, 1998.
- [6] C. Li, B. Yu, and K. Sycara, "An incentive mechanism for message relaying in unstructured peer-to-peer systems," *Electronic Commerce Research and Applications*, vol. 8, no. 6, pp. 315–326, 2009.
- [7] A. Romanelli, A. C. S. Schifino, R. Siri, G. Abal, A. Auyuanet, and R. Donangelo, "Quantum random walk on the line as a Markovian process," *Physica A: Statistical Mechanics and its Applications*, vol. 338, no. 3-4, pp. 395–405, 2004.
- [8] A. Chakrabarti, C. Lin, and N. K. Jha, "Design of quantum circuits for random walk algorithms," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI '12)*, pp. 135–140, 2012.
- [9] J. Duda, "From maximal entropy random walk to quantum thermodynamics," *Journal of Physics: Conference Series*, vol. 361, no. 1, Article ID 012039, 2012.
- [10] K. Rudinger, J. K. Gamble, E. Bach, M. Friesen, R. Joynt, and S. N. Coppersmith, "Comparing algorithms for graph isomorphism using discrete-and continuous-time quantum random walks," *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 7, pp. 1653–1661, 2013.
- [11] M. S. Underwood and D. L. Feder, "Universal quantum computation by discontinuous quantum walk," *Physical Review A: Atomic, Molecular, and Optical Physics*, vol. 82, no. 4, Article ID 042304, 2010.
- [12] L. Jun, *Investigations on Quantum Random Walk Search Algorithm*, 2006.
- [13] N. Shenvi, J. Kempe, and K. B. Whaley, "Quantum random-walk search algorithm," *Physical Review A: Atomic, Molecular, and Optical Physics*, vol. 67, no. 5, Article ID 052307, 2003.