Research on anycast cross service based on Chaos-Particle Swarm Optimization Algorithm

Zhang Yuanyuan[1,a,] Zhang Zhen[2],

[1]CCIDIT, Beijing 100048, China;

[a]yuanyuanzhang@163.com

Keywords: anycast; cross service; PSCO; multi-object optimization

Abstract. Anycast is a new mode of communication to balance load and improve network quality. Due to the presence of cross service issues, it makes a great decline in service efficiency. In order to solve the problem above, this paper proposed a particle swarm optimization algorithm (CPSO) based on chaos from a global point of view of the selected route optimal adjustment path, and the simulation verified the feasibility and efficiency of the algorithm.

Introduction

In order to improve network quality of service and balance the network load, anycast has been proposed as a new model of communication network service[1]. In the traditional Internet, different communication packets share network resources equally and the network provides "best-effort" service. But with the development of real-time multimedia services such as video on demand, online meetings and network television, requirements on the network quality of service (QoS) have become more sophisticated[21]. Anycast is an important application to increase the transference efficiency of video streaming. As multicast can solve the problem of video stream broadcast, anycast can effectively solve the problem of video streaming on demand. The current research on anycast QoS issues will lead to cross service in whatever way, especially for a long time anycast stream, the problem will decrease the efficiency, even lower than the unicast stream. Therefore, we should not only design the QoS anycast routing algorithm but also necessary provide an optimization from a global perspective on the route [3].

For anycast service will take some time and the state of the network is constantly changing[4], the server of optimal anycast initialization is not necessarily optimal in the service process, this will lead to cross service issues shown in Figure 1.At the time t_1 , the user C_1 sends a request service, the server S_1 accepts the request and establish a connection①to communicate; At the same time, the user C_4 sends a request services, server S_2 accepts the request and establish a connection②to communicate. At another moment t_2 arrives, the user C_2 requests service at this time, cause the border router ER_1 is occupied, server S_2 response and accept the request to establish a connection ③;At the same time the user C_3 requests a service, the border router ER_2 is occupied at this time, the server S_1 to respond and accept the request connection④. This results in cross visit between different regions of the server and client point. The mainly reason of cross service issues falls into two categories: server capacity constraints and limited network bandwidth, which greatly reduces the efficiency of the service.

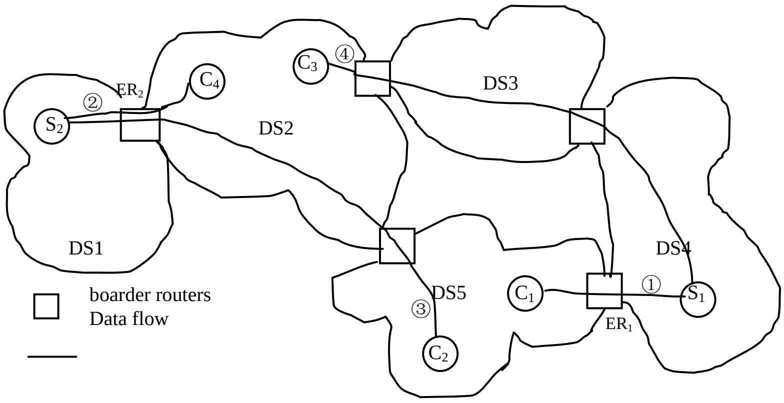


Fig.1 the cross service problem

Unlike general QoS Anycast Routing Algorithm for the purpose of selection a link to meet the user's parameters, the solution for the problem of cross service anycast QoS routing is to adjust and optimize the router from the global point of view, that is, to meet the various constraints context, the client's service requests redirected to the nearest anycast server to realize the real sense of the anycast service.

Description of the problem and mathematical model

QoS path adjustment must meet the following requirements[5]: 1) The adjusted path still needs to support QoS path; 2) Reasonable allocation of network resources. Specifically compress the selected anycast stream transmission path, reduce network resource consumption, while achieve a balance between network traffic and server load balancing. The network is defined as a directed weighted graph $G=(V,E)$ [6], where V is a finite set of vertices in the network, E is the set of limited network links, where the set of vertices V includes two types of nodes: client node M and server nodes N. For any application for a request for anycast $n_i,(n_i \in N)$ of the client m_i, there are $m_i^{n_i}$ kind of feasible paths; the required solution of the problem here is that from these options, select one router path, in order to satisfy the constraints in the network in the case of the overall optimal performance.

Making decision variable $X=[x_i]_m$, the server set that the user x_i selected. $RD=[rd_k]_l$ is the set of link delay, rd_k is a certain link K ‘s delay; $LD=[ld_k]_l$ is the set of that link length, ld_k that the length of link k ;server load here is measured by its processing the number of packets during unit time, $SP=[sp_j]_n$ is the set of network server load, sp_j stands for server j the current load condition; $CB=[cb_i]_m$ means that the set of user bandwidth, cb_i is the user i occupied bandwidth, while total bandwidth of current network is SB; $R=[r_{i,j,k}]_{m*n*l}$ represents the set of link from the client i to the server j, $R=1$ shows including this paragraph links, while $R=0$ does not include this section describes the link. In this paper, four network performance parameters are selected as the optimization objectives:

1) the minimum total delay between clients and servers, which can be adjusted the path between the clients and servers to ensure that they are not too far away as possible;

2) anycast server load balance, this parameter ensure the average distribution of the user's request to each anycast server in the case of maximum utilization of each server;

3) load balance of network resources, this parameter avoid the link congestion as a result of repeated use of the network link;

4) the number of adjusted paths, this parameter ensure to do little re-adjustment of the overall network path to complete optimization.

The anycast routing optimization mathematical description of multi-objective optimization problem as follows:

$$\min f_1(x), \min f_2(x), \min f_3(x), \min f_4(x) .$$

$f_1(x) \ = \sum_{i=0}^{m-1}\sum_{k=0}^{k-1} l_1(RD, R, x_i, k)$; the total delay from source to destination;

$f_2(x) = \sum^{m-1}\sum^{n-1} l_2(SP, x_i, j)$; The total load of all servers

$f_3(x) = \sum_{i=1}^{m-1}\sum_{j=1}^{n=1}[\frac{CB}{cb_i}P - \frac{\sum_{i=1}^{m-1}\sum_{j=1}^{n=1}\frac{CB}{cb_i}P}{LD}]^2 Px_i$;the variance of the network link resource utilization;

$f_4(x) = [\sum_{i=0}^{m-1} l_4(xi)]$; The adjusted number of routing.

Constraint conditions:

$CB \le SB$, $x = 0,1,...N-1$. That the bandwidth consumed by users is less than the bandwidth of each link provided;

$f_2(x) < SP$; The current server which can bear the load current must be less than the servers can provide the total load.

This is actually a multiple constraint optimization problem, which is proved to be NP-complete problem. This means that the calculation of the polynomial can not be found on the level of the optimal solution. To solve this problem now almost take the heuristic algorithms, however, these algorithms have their own shortcomings, such as computing overhead is too large, can not be used in the scale network, and only to find the local optimum . With the research on chaos, it is recognized that chaos with the ergodicity characteristics is a strong global and local search optimization algorithm. In this paper, chaos-particle swarm optimization algorithm is based on this.

Realization optimization algorithm based on chaotic particle swarm

The basic idea: Particle swarm optimization (PSO) is an intelligent optimization algorithm of simulation foraging behavior of birds[7], the particles individuality organic combined with sociality produce group intelligence, guide optimization search and have the characters of constructing model in simple way, fast convergence and so on. But there are also shortcomings, such as easy to fall into local optimal, slow convergence evolution in later period and poor robustness. Chaos is a nonlinear phenomenon in nature; chaotic variable actually appears inherent regularity which seems in a messy process of change. This article will introduce the idea of chaotic into particle swarm optimization algorithm, the basic idea is to give the optimal particles groups of particles chaotic optimization firstly, and then replace the result of random chaos optimization for a particle randomly. This approach improves the ability of the traditional particle swarm optimization out of local optimization and improves the convergence speed and accuracy.

The realization step:

Suppose an n-dimensional target in the search space, there are a group of N particles, where particle i represents a n-dimensional vector $x_i = (x_{i1}, x_{i2},..x_{in}), i = 1,2,..N$, component x_{ij} in $[a_i, b_j]$

range values, that $a_i \le x_{ij} \le b_j, i = 1,2,...N, j = 1,2,..,n$ .calculate each degree function of each particle x_i to measure the size of the applicable scale value x_i merits. Define $v_i = (v_{i1}, v_{i2},..v_{in}), i = 1,2,..N$. Particle i remembers so far to search for the local optimal $P_i = (p_{i1}, p_{i2},..p_{in}), i = 1,2,..., N$. The whole particle swarm's best position searched so far $P_g = (p_{g1}, p_{g2},..p_{gn}), n = 1,2,..N$. update operation of particle swarm is:

$$v_{i+1} = v_i + c_1 r_1(p_i - x_i) + c_2 r_2(p_g - x_i) \qquad ①$$

$$xi_{+1} = x_i + v \qquad ②$$

$i = 1,2,..N$; Learning factors c_1 and c_2 are non-negative constant; r_1 and r_2 is random number between [0, 1]. Iteration termination condition is generally chosen according to the maximum number of iterations according to the specific issues or the best searched position to meet the predetermined minimum threshold so far. In order to prevent the iterative process into local optimum, based on the ergodicity of chaos movement of the current particles swarm so far, the best position produces the chaotic sequences, which replace one of the current particles swarm randomly. Generally random state of motion determined by certainty equations is called chaotic, chaotic state of the variables showed a chaotic variable. Here we use the Logistic chaotic system into the particle swarm algorithm:

$$z_{n+1} = \mu z_n (1 - z_n) \quad n = 1,2,...N \qquad ③$$

Where μ is the control parameter, the equation can be seen as a dynamic system. value of μ determined by any initial value $z_0 \in [0,1]$, can be determined by iteration of a time series of $z_1, z_2, z_3,..$ Concrete steps are as follows:

1) determine the initial parameters: population size N , learning factors c_1, c_2 ,the number of the evolution, the number of chaotic optimization. Generation size produced randomly and operate according to formula ① and ②;

2) The fitness function determined: $F(x) = w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x).w_1, w_2, w_3, w_4$ are the objective function weights, which are determined according to different application.

3) Chaotic optimization for the optimal position $P_g = (p_{g1}, p_{g2},..p_{gn}), n = 1,2,..N..$ The p_{gi} is mapped to the Equation 3 domain [0,1]: $\frac{p_{gi} - a}{b_i - a_i},(i = 1,2,..n)$, then iterative the chaotic variables $z_i^{(m)}, m = 1,2,...$ sequence generated by Equal③, then generated by chaotic variables $z_i^{(m)}, m = 1,2,...$ sequence through the inverse mapping $p_{gi}^{(m)} = a_i + (b_i - a_i) z_i^{(m)}$ $m = 1,2,..$ returns to the original solution space, the result is $p_g^{(m)} = (p_{g1}^{(m)}, p_{g2}^{(m)},...p_{gn}^{(m)}), m = 1,2,...$ Chaos in the original variables through the solution space for each feasible solution $p_g^{(m)}$.applied to calculate the function value, retaining the best possible performance solution p^*;

4) Selected from the current population p^* instead of using a particle randomly;

5) when reach the maximum algebra or satisfactory solution, the optimization process is complete, or return ③.

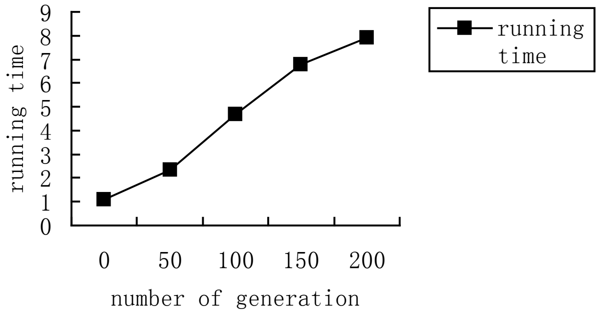
Algorithm analysis and simulation

Convergence Analysis: In coding, the path sequence with an integer that makes operations on the original sequence to simplify the path for the integer operations. According to the reference, in the integer domain CPSO has the better performance in the optimal convergence and convergence speed when direct evolutionary search algorithm works. According to the reference [9], the proposed CPSO algorithm can reach global optimum converges and has good stability.

Effectiveness Analysis: The proposed chaotic particle swarm algorithm and the traditional PSO algorithm are compared by the simulation experiment. The simulation network generated randomly with method proposed by Waxman[9,10], the source node and anycast server nodes are generated randomly. Server resources are in [1, 10] and the chaotic particle swarm algorithm is realized in C language for implementation.

First, the chaotic particle swarm algorithm is executed in different size of the network, the results shows in Figure 1.The chaotic PSO algorithm is still a very good performance with the network size increases, with the increase in the number of nodes, its running time in the network slowly increases and there is no exponential growth, which shows that the algorithm has higher efficiency to meet the real time requirements in the current network.

For ease of comparison the traditional PSO algorithm, in some target constraints the two algorithms take the same bandwidth and delay constraints, the same number of initial generations.



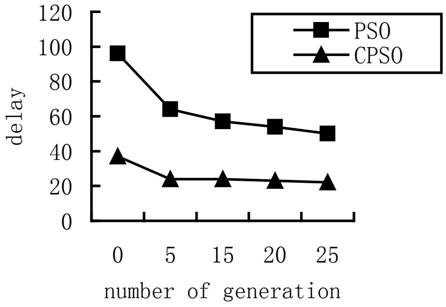


Fig.1 running time of CPSO

Fig.2 the degree of convergence in PSO and CPSO

Here it is mainly researched on the convergence ability and convergence speed. Figure 2 shows the CPSO and the traditional PSO convergence condition in a network with 100 nodes. Both algorithms have very good convergence in a feasible route. But in the search efficiency, due to the introduction of chaotic mapping ideas, the CPSO can reach the global optimum; therefore, PSO algorithm is better than chaos PSO algorithm in the convergence speed.

Summary

The article proposed based on chaotic particle swarm optimization for anycast cross service issues, the optimized path has been adjusted, viewed as a multi-objective routing optimization problem, the PSO algorithm based on chaotic preserve both the diversity of groups and can speed up the search of efficiency, avoid falling into local optimum and achieve the intended purpose.

References

[1] He Hong, Qian Feng, Du Wenli. A chaotic immune algorithm with fuzzy adaptive parameters [J]. Asia-Pacific Journal of Chemical Engineering, 2008, 3(6):695-705.

[2] Lovbjerg M, Rasmussen T K, Krink T. Hybrid particle swarm optimizer with breeding and subpopulations[C]// Proceedings of Genetic and Evolutionary Computation Conf. San Francisco: Morgan Kaufmann Publishers Inc, 2001: 469-476.

[3] B Liu,L Wang,Y-H Jin,F Tang,D X Huang. Improved particle swarm optimization combined with chaos [J].Chaos Solitons &Fractals (S0960-0779), 2005,25(21): 1261-1271.

[4] UN J,FENG B,XU WB. Particle Swarm Optimization with Particles Having Quantum Behavior [A].Proceedings of 2004 Congress on Evolutionary Computation[C].2004.325-331.

[5] Shan L,Qiang H, L.j,et al. Chaotic Optimization Algorithm Based on Tent Map[J].Control and Decision,2005,20(2):179-182.

[6] Ma, Zhen-Yuan. A scalable framework for global application anycast[C]. 2009 2nd International Conference on Information and Computing Science, ICIC2009.

[7] Seunghoon Lee,Jaechul Suh,Kwanho Song.IPv6 Anycast-based DNS Service Model[C]. Communications,2007.APC2007.Asia-Pacific Conference 2007:247-250.

[8] MA Zhenyuan, ZHOU Jie, ZHANG Ling.Tunnel based approach to scalable global IP anycast [J]. Journal of South China University of Technology (Natural Science), 2008, 36(4):75-79.

[9] Jetty Kleijn, Maciej Koutny. A Petri net model for membrane systems with dynamic stucuture[J].Natural Computing.2009, 16(10):781-796.

[10]Jun zhi,Jianyong Liu,Kai chen.Coures Optimization on based on Improved Immune Genetic Algorithm[J].Computational Sciences and Optimization2009, 660-662.