THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Implementing PSO Algorithm for Efficient Request Routing in Content Delivery Network

Athira K. B.
Master of Technology in Computer Science and Engineering
Department of Computer Science And Engineering, Palakkad, India
Scaria Alex
Assistant Professor, Department of Computer Science and Engineering, Palakkad, India

Abstract:
Content Delivery Networks are the only practical solution to reduce the overwhelming condition of the server's load due to the excessive traffic due to client's request. This paper focus to provide Request Routing so that the selection of server for serving the request becomes more easy and efficient without a wastage in the resources consumed. The request redirector performs the request routing based on certain policies. In order to provide an efficient request routing in content delivery network, there are certain parameters introduced like memory utilization, bandwidth utilization, cpu utilization and connection availability. These parameters are considered each time a request arrives. PSO algorithm is used to evaluate these searching parameters in the searching space. It finds the value of parameters for each server and the generated value is compared with each other. The server with less optimized value will be selected for processing the request. Thus request routing will be done based on this result. By considering all the parameters in the searching space with pso algorithm ,proposed request routing mechanism becomes more efficient.

Keywords*: CDN, Request Routing, PSO*

1. Introduction

A Content Delivery Network is a part of internet, which operates on a network of servers around the world and rents capacity of these servers to customers who wants their websites to work faster by distributing contents from locations close to the user. When the user navigates to the URL the browser is redirected to one of the copies of this website within the network of servers. In addition to using the network's own server ,it can deliver contents from other Content Delivery Network in the form of peer-to-peer networking. The CDN operator gets paid by the content provider for improving the performance of their website and also for producing a user friendly atmosphere to the clients.

A CDN mainly consist of a main server called a back- end server which contains the original data which needs to be replicated to the network of servers. Through CDN the back-end server is more benefited because of the reduction in traffic entering into it. The second part of CDN is a redirector or a router, used for the purpose of distributing the requests from the clients to the particular server having the content. Another important part is the servers which are deployed in different locations contains the replicated content from the main server.

Akamai, LimeLight, MaxCDN, CloudFlare, CatchFly, CloudLayer , CloudCache, TinyCDN etc are the popular CDN operators provides services such as improved website performance for highly interactive contents, on demand global capacity to meet peak traffic, improved protection and protection and uptime against load based online attacks, etc to the popular website providers. Basically a CDN helps in loading the static content of your website and now even streaming multimedia contents in a fast and efficient manner. Thus the pressure exerted on the hosting server can be fully avoided. Researchers are still going in the field of publishing a streaming content to the end- users efficiently

Operations of CDN(Fig.1) explains the step by step processing of a request through a CDN. There will be an original server and the website which makes use of the services of CDN will be distributed to the replica server through the Distribution System. At the same time it will update the Request Routing System in order to make the search of the content in the replica server more easy. When the client requests for a content the Request Routing System forwards the request to the replica server and the request will be processed there. Finally the requested content will be delivered to the client through the request routing system. In case of an unavailability of content in the replica server, the request routing system forwards the request to the main server and thereby update the replica server through the above explained process.

There are several approaches in practice for selecting the replica server. The server with better response time , based on the closest distance from the client etc. Request-Routing techniques are used as a vehicle to extend the reach and scale of Content Delivery Networks. Some of the request routing mechanisms are in practice is given below:

DNS based Request-Routing techniques are common due to the ubiquity of the DNS system. In DNS based Request-Routing techniques, a specialized DNS server is inserted in the DNS resolution process. The server is capable of returning a different set of contents based on user defined policies, metrics, or a combination of both.
Single Reply is an approach in which the DNS server is authoritative for the entire DNS domain or a sub domain. The DNS server returns the IP address of the best surrogate server. The IP address of the surrogate could also be a virtual IP(VIP) address of the best set of surrogates for requesting DNS server.
Multiple Reply is an approach in which multiple Request-Routing DNS servers can be involved in a single DNS resolution. The rationale of utilizing multiple Request-Routing DNS servers in a single DNS resolution is to allow one to distribute more complex decisions from a single server to multiple, more specialized, Request-Routing DNS servers.
NS Redirection is another approach in which a DNS server can use NS records to redirect the authority of the next level domain to another Request-Routing DNS server. The technique allows multiple DNS server to be involved in the name resolution process. One drawback is that the number of DNS servers are limited than the number of NS requests.
Another approach called CNAME Redirection in which Request Routing DNS returns a CNAME record to redirect into an entirely different domain. The disadvantage of this approach is the overhead in the resolution of the new domain. The request routing DNS server is independent of the format of the domain name is the disadvantage.

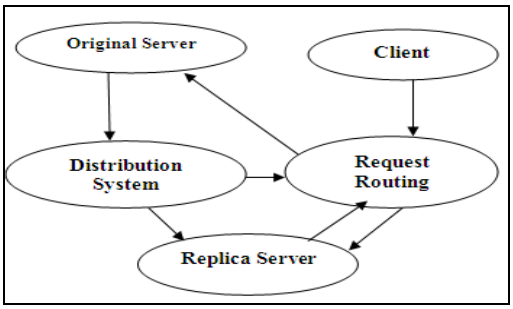

Figure 1: CDN Operation

Another important mechanism is the Transport layer request routing. In this request routing system the request from the client in the form of packets are verified first and extracts the information like client's IP address, port information and layer 4 protocol. Combining these information and the user defined matrices are used to determine the appropriate surrogate server. It is possible to combine both DNS based techniques and Transport layer mechanisms.
Application Request Routing mechanism is more advanced than the Transport Layer Request Routing. It examines the requests in the form of packets from the clients more deeply than the Transport layer and extracts the necessary information along with the finely grained objects for the requested content. Thus the selection of the server and processing made easy.
Header Inspection is another mechanism included in the Application Layer. It extracts the header information which provide hints in the initial portion of the session about how the client request must be directed.

2. PSO Algorithm

Particle Swarm Optimization is an approach to problems whose solutions can be represented as a point in an n-dimensional solution space[5]. A number of particles are randomly set into motion through this space. At each iteration, they observe whether it is optimal of themselves and their neighbours and emulate successful neighbours (those whose current position represents a better solution to the problem than theirs) by moving towards them. Various schemes for grouping particles into competing, semi-independent flocks can be used, or all the particles can belong to a single global flock. This extremely simple approach has been surprisingly effective across a variety of problem domains.
PSO was developed by James Kennedy and Russell Eberhart in 1995 after being inspired by the study of bird flocking behaviour by biologist Frank Heppner[5]. It is related to evolution-inspired problem solving techniques such as genetic algorithms. As stated before, PSO simulates the behaviours of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. The effective strategy is to follow the bird which is nearest to the food.
There are three variants of PSO algorithms are in practice. It includes Discrete PSO, MINLP PSO and Hybrid PSO. Discrete PSO can handle discrete binary variables, MINLP PSO can handle both discrete binary and continuous variables and Hybrid PSO utilizes basic mechanism of PSO and the natural selection mechanism, which is usually utilized by EC methods such as GAs
PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.
PSO is initialized with a group of random particles and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (optimal) it has achieved so far. This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbours, the best value is a local best and is called lbest.

After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).
v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) ----------(a)
present[]=present[]+v[]----------(b)
The pseudo code of the procedure is as follows
For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value
as (PBest) in the history.
set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle
Calculate particle velocity according equation (a)
Update particle position according equation (b)
End
While maximum iterations or minimum error criteria is not attained.
The end of the process the particle with minimum fitness value will be selected as the final solution.
Most of evolutionary techniques have the following procedure:

- Random generation of an initial population
- Reckoning of a fitness value for each subject. It will depend on the distance to the optimum.
- Reproduction of the population based on fitness values.
- If requirements are met, then stop. Otherwise go back to 2.

From the procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques. However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm
Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only gBest (or lBest) gives out the information to others. It is a one -way information sharing mechanism. The evolution only looks for the best solution.
The applications of PSO are in the fields like Telecommunications, data mining, power systems signal processing, Function optimization, artificial neural network training, fuzzy system control where Genetic Algorithm can be applied.

3. PSO Algorithm in CDN Request Routing

PSO is a searching algorithm which is used to find the solution within the search space with different searching criteria called dimensions of the searching space. The basic concept of PSO lies in accelerating each particle toward its pbest and the gbest locations, with a random weighted acceleration at each time step There are four parameters which are used as four dimensions of the search space to select the particular server for responding to the request from the client. The parameters include bandwidth utilization ,connection utilization ,memory utilization and cpu utilization.
When a client requests for a content to the request redirector, it will select an appropriate replica server which satisfies the above mentioned four parameters. PSO algorithm generate particles ,each request from the client can be considered as a particle and it can be represented as P=(p1,p2,p3.....pn) and within the searching space each particle can be represented as Pi=(p_{c1},p_{c2},p_{c3},p_{c4}) where c1(criteria 1) is the bandwidth utilization, c2(criteria 2) is the connection availability, c3(criteria 3) is the memory utilization and c4(criteria 4) is the cpu utilization. During the generation of each particle the request redirector evaluates the parameters based on the equations below and select the replica server. There will be n replica servers within a network. Hence to evaluate the parameters it is necessary to calculate its values for each replica server.
The PSO algorithm first assumes each replica server as the best solution. During a search an optimum value will be evaluated using the optimum function until there is no server left to calculate. The replica server with the smallest optimum value will be selected using the aggregate function MIN. Thus the selected server can deliver the content to the requested client.

3.1. Determining The Parameter Values

In this paper, there are four parameters specified and which are all used in the searching space. Each request from the client will be utilizing a specific parameter value. According to the type of the request the parameter values varies. Thus during each iteration for searching, the values of the parameter should be separately determined. There will be a particular amount of bandwidth assigned to each network, and each request demands a specific bandwidth. Thus to calculate the bandwidth utilization for the request, subtract the required bandwidth value from the total bandwidth. The formula to calculate the bandwidth is given below.

BWU = TBW- RBW
Where BWU is the Bandwidth Utilization, TBW is the Total Bandwidth and RBW is the required Bandwidth.
The Connection Availability is the load of each server experiencing during the processing of a request and it can be calculated using the intermediate redirector which counts the total number of connections and number of requests served . Hence it is possible to obtain the value of Connection Availability by subtracting the number of processed requests from the total number of requests. The formula for calculating Connection Availability is given below.
CA=TC-PR
Where CA is the Connection Availability, TC is the Total Connections and PR is the Processed Requests.
Memory Utilization is the memory available in the servers to accept a new request. It can be calculated by subtracting the required memory from the total memory available.
ie, MU=TM-MR
where MU is the Memory Utilization ,TM is the Total Memory and MR is the Memory Required.
Similarly the fourth parameter CPU Utilization is the usage of CPU for the request processing.
It can be calculated as,
CPU-U=1-CPU-U
Where CPU-U is the CPU Utilization.

3.2 Implementing The Parameters Within The Search Space
The parameters listed above are implemented in the searching space for making a decision to select an appropriate server which is efficient. There will be four parameters which are considered as four dimensions of the searching space. The searching space can be represented as SS=(d1,d2,d3,d4) where d1,d2,d3,d4 are the parameters listed. The parameter values for each replica server is calculated separately. It can be represented as $R_n = (r_nd1,r_nd2,r_nd3,r_nd4)$ where r_nd1,r_nd2,r_nd3,r_nd4 are the parameters calculated for each replica server. Then the requests from the clients are considered as the particles in the searching space. When a particle is generated the request redirector search within the searching space using these four dimensions. The particles can be represented as $P_n=(P_nd1,P_nd2,P_nd3,P_nd4)$ where P_nd1,P_nd2,P_nd3,P_nd4 are the required parameter value for each particle
The generated results are used to calculate the optimized value using the optimization function. Using the optimized value the server with less optimized value will be generated using the aggregate function called MIN. The resulting server with less value is selected for forwarding the content to the requested client.

3.3. Generating Optimized Value
The optimized value can be generated by the following formula:
Optimization Function ,

$$\sum_{n=1}^{N} ((rnd1 - pnd1) + (rnd2 - pnd2) + (rnd3 - pnd3) + (rnd4 - pnd4))/N$$

The optimization function produce the average of the optimized value. This value is generated for each replica server and the minimum of optimized value among all replica server is calculated and the resulting server is selected.
ie, MIN(F(Opt.Value).

3.4. PSO Algorithm
- Step 1:Calculate the four parameter values for each replica server
- Step 2:Calculate the required parameter values for each particle generated.
- Step 3:Using the parameter values generate the optimized value by optimization function.
- Step 4:Find the replica server with the minimum optimized value
- Step 5:Select the replica server with less optimized value

The above algorithm can be used to implement the selection process. .

4. Experimental Results
We implemented the proposed PSO algorithm by using Java 2 Standard Edition (Jdk 7) in a personal computer running Microsoft Windows 7 with the specifications such as processor above 1.5 GHz, hard disk of 40 GB and RAM of 512 MB. NetBeans IDE 7.4 provides a platform framework for running the application. We experimented the algorithm using two servers excluding the main or backend server and five clients. While a request comes from the client after checking the algorithm, the request redirector selects the best server .

5. Conclusion
The PSO algorithm implemented in Content Delivery Network for request routing can enhance the routing efficiency by selecting the best server. The ultimate problem experienced in a network is the bandwidth control, through implementing the algorithm routing will be much more efficient.

6. References

1. Sabato Manfredi, Francesco Oliviero, and Simon Pietro Romano,"A Distributed Control Law for Load Balancing in Content Delivery Networks", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 21, NO. 1, FEBRUARY 2013.
2. Jian-Bo Chen, and Chu-Chuan Chen, "Using Particle Swarm Optimization Algorithm in Multimedia CDN Content Placement," International Journal on Parallel Architectures, Algorithms and Programming,2012
3. HongboJiang, Huazhong University of Science and Technology**, "**Design, Implementation, and Performance of a Load Balancer for SIP Server Clusters,**",** IEEE/ACM TRANSACTIONS ON NETWORKING,, vol. 20**,** Aug. 2012
4. Akamai, http://www.akamai.com
5. http://www.cs.ru.ac.za/courses/Honours/AI/PSONotes.doc