Algorithms for Content Distribution in Networks

Dr Deepak Garg, Senior Member, IEEE
Computer Science and Engineering Department
Thapar University, Patiala

Abstract— **In this paper an algorithm is presented which helps us to optimize the performance of content distribution servers in a network. If it is following the pay-as-you-use model then this algorithm will result in significant cost reduction. At different times the demand of different kind of content varies and based on that number of servers who are serving that demand will vary.**

Keywords- Content Distribution; Efficient Algorithms; Capacitated.

I. Introduction

The problem of content distribution in Networks is defined in the context of demand and supply paradigm. There may be downloadable software with a considerable size. At peak loads 10,000 persons may be downloading that software at the same time. However, downloads will come to an average number of 15, 00 downloads at any given point of time. Whenever the new versions of that software are launched, peak is again reached. There can be a number of such software downloads that you want to make available through your website. The demands will also be coming from different regions and different IP addresses. There will be certain pattern that can be found out by the analysis. These demand number will also keep on changing based on the time of the day in that region. The distance of the servers from that client who is requesting the software is crucial in determining the time taken to download that software. Overall performance of the site in terms of average time being taken to download one software is crucial to the image and working of the company. This is a dynamic problem, where new servers need to be made active once the active server reaches a threshold value. Similarly once there is a drop in demand from a particular region, than the server servicing that region must be relieved from service to save money. The input in this kind of problem is given in terms of a matrix containing the cost of opening of each server location and a set of locations generating the demand.[1-4]

The demands from the clients will come one by one in the form of the http request and must be handled by our algorithm. These client demands must be assigned to some server based on the location of that client and the nearest server from that position. However, it is not possible to open a new or passive demand location for a few demands. In that case these demands will be transferred to the nearest active server. When the new demands crosses a particular threshold then only new service points can be opened.

In Incremental Content Distribution Algorithm when an existing server serves number of clients less than a defined value then that server is stopped and existing demand services are transferred to another content distribution servers.

II. Content Distribution System Algorithm for Dynamic and Incremental Content

Content Distribution Server problem can be characterized as:

- A universe, *U*, from which a set *C* of client input positions is selected,
- A distance metric, $d: U \times U \rightarrow R^{+,}$ defined over the universe R^{+},
- An integer, $p \geq 1$, denoting the number of servers to be located, and
- An optimization function *g* that takes as input a set of client positions and *p* server positions and returns a function of their distances as measured by the metric *d*.

III. Approaches for Handling Location Models

Formulating an appropriate model is only one step in analysing a location problem. Another much challenging task is identifying the optimal solution. Attempting a solution with well-known branch and bound optimization methods often consume unacceptable computational resources. The reason behind is that even the most basic location models are NP Hard [6]. As a result, the location analyst must have to devise other methods to identify optimal solution or at least near optimal ones. Some of the most common approaches used by location analysts are discussed below.

A. Greedy heuristic

A sequential approach that begins by evaluating each site individually and selecting the one facility site that yields the greatest impact on the objective. That facility site is then fixed open. The location of the next facility is then identified by enumerating all remaining possible locations and choosing the site that provides the greatest improvement in the objective. Each subsequent facility is located in an identical manner. The method stops when the required number of facilities has been sited.

B. Improvement heuristic

While greedy heuristics are effective at identifying a feasible solution with modest computational effort, they can't be relied upon to produce consistently good solutions. One of the earliest improvements heuristic is neighborhood search algorithm. In 1968, the most widely known improvement method was introduced [7]. The basic idea is to move a facility from the location it occupies in the current solution to an unused site. Each unused location is tried in turn and when a move produces a better objective function value, then that relocation is accepted and we have a new (improved) solution. The search procedure is repeated on the new solution and stops when no better solution can be found via this method. A variable neighborhood search algorithm was presented for solving the p-median problem [8]. The algorithm performs an intensive local search on the current solution until it settles in local optima. It then repeats the process by randomly selecting a solution from a neighborhood at a distance k from the current best solution. The process continues, increments k, until some predefined maximum value of k is attained.[9-11]

C. Lagrangian relaxation

When using any heuristic we are trading on savings in solution time against the quality of the solution while the heuristic often find good solutions to a variety o location problems, it is difficult to evaluate the trade off since we have no way of knowing how far from optimality those solutions are. Without having the optimal value of the objective function available for comparison, we can sometimes approximate the difference between a heuristic's solution and the optimal solution by finding bounds. One of the primary attraction of the technique known as Lagrangian relaxation is that it provides upper and lower bound on the value of objective function [12]. This is done by eliminating i.e relaxing one or more of the constraint of the original model and adding these constraints multiplied by an associated Lagrange multiplier to the objective function. The role of these multipliers is to derive the Lagrangian problem towards a solution that satisfies the relaxed constraints. The primary challenge in applying such technique is in selecting which constraint to relax. Ideally the relaxed problem ought to be solvable by inspection or by a simple sorting the objective function coefficients.

Algorithm

Input : Cost of starting a new server and a queue of client requests {cr1,cr2,…,crn}

A set S is maintained which consists of currently active servers to process that demand. The set will look like {φ , s1,s2,…sn}. Sets of Client requests which are being handled by individual servers will also be maintained.

Step 1: When a new client request for a download to start, either it can be assigned to the existing active server or a new server should be started.

Step 2. Let si is the server nearest to the client request and number of client requests assigned to si< threshold value then the download request is assigned to that server

Step 3: If si is the server nearest to the client request and number of client requests assigned to si=threshold value then the new server needs to be assigned. New server will be started in the cloud of regions being represented by the incoming request for download.

Step 4: Another thread will keep a check on the number of requests that are being services by a server because some of them will be completed at any given time.

Step 5: Once the number of requests are below a threshold then it will check for the possibility of transferring the existing download request of one server to another server keeping in mind that the total load does no increases beyond the limits after such transfer.

The Total cost will be calculated as

(Number of active servers) * cost of starting a server + Sum of all client requests for download taken over all distances of existing servers

The evaluation of this algorithm tells that it will give results close to the optimum values that can be reached.

IV. Results and Discussion

By the nature of the problem it can be seen that essentially it an NP-complete problem in which the number of options can be exponential. Because a particular request canbe assigned to any of the active set of n servers giving rise to n different options, similarly if there are m different requests that will come to exponential possibilities. By the above algorithm, the solution will be close to the actual solution and limited to 1.5 times the value of the exact solution.

V. Conclusion

Content management algorithms initially were only dealing with static data. Now improved algorithms can also handle dynamic nature of the data, locations, demands and resources. The resultant saving is significant in terms of time and cost. There is also a possibility of applying the concepts of linear programming to further improve the approximation ratio.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A General Approach to Online Network Optimization Problems. ACM Transactions on Algorithms, 2(4):640 to 660, 2006.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local Search Heuristics for k-Median and Facility Location Problems. SIAM J. on Computing, 33(3):544 to 562, 2004.

[3] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis. Cambridge University Press, 1998.

[4] M. Charikar and R. Panigrahy. Clustering to Minimize the Sum of Cluster Diameters. In Proc. of the 33rd ACM Symposium on Theory of Computing (STOC '01), pages 1 to 10, 2001.

[5] M. Chrobak and C. Kenyon-Mathieu. Online Algorithms Column 10: Competitiveness via Doubling. SIGACT News, 37(4):115 to 126, 2006.

[6] J. Fakcharoenphol, S. Rao, and K. Talwar. Approximating Metric Spaces by Tree Metrics. Encyclopedia of Algorithms, pages 51 to 53, 2008.

[7] D. Fotakis and C. Tzamos. Winner-Imposing Strategyproof Mechanisms for Multiple Fa-cility Location Games. In Proc. of the 6th Workshop on Internet and Network Economics (WINE '10), volume 6484 of LNCS, pages 234 to 245, 2010.

[8] G. Frahling and C. Sohler. Coresets in Dynamic Geometric Data Streams. In Proc. of the 37th ACM Symposium on Theory of Computing (STOC '05), pages 209 to 217, 2005.

[9] P. Lu, X. Sun, Y.Wang, and Z.A. Zhu. Asymptotically Optimal Strategy-Proof Mechanisms for Two-Facility Games. In Proc. of the 11th ACM Conference on Electronic Commerce (EC '10), pages 315 to 324, 2010.

[10] R.R. Mettu and C.G. Plaxton. The Online Median Problem. SIAM J. on Computing,32(3):816 to 832, 2003.

[11] A. Meyerson. The Parking Permit Problem. In Proc. of the 46th IEEE Symposium on Foundations of Computer Science (FOCS '05), pages 274-284, 2005.

[12] R.L. Francis (Editors) P.B. Mirchandani. Discrete Location Theory. Willey, 1990.

[13] B. Krishnamurthy, C. Wills, Y.Zhang On the use and performance of content distribution networks IMW'01 poceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement,2001

[14] G.Pallis, A.Vakali Insight and perspectives for content delievery networks , Communications of the ACM, 2006

AUTHORS PROFILE

Dr Deepak Garg, Senior Member of ACM and is Chair, ACM North India SIGACT. He is Senior Member IEEE and Secretary IEEE Delhi Section Computer Society. He has more than 50 publications to his credit. His research interests are Algorithm design and complexity issues. He is having 15 years of research experience