Hybrid and Exact Approaches for Replica Placement in Content Distribution Networks

Tiago Araújo Neves
Lúcia Drummond
Luiz Satoru
Célio Albuquerque
Eduardo Uchoa

December 14, 2009

Abstract

A content distribution network (CDN) is a system of interconnected distributed servers that cooperate to improve the delivery of content to clients. It maintains replicas of each content in its servers, with the goal of reducing delays, server load and network congestion, therefore improving the quality of the service. However, because of the costs involved in the replication process, it is not reasonable to replicate the contents over the entire set of servers. In this work, exact and heuristic approaches are proposed to solve a dynamic and online problem that appears in CDN management, called the Replica Placement and Request Distribution Problem. This problem consists of deciding which replicas will be placed in each server and which servers will handle each request. The overall objective is to minimize the traffic in the network without violating QoS constraints.

1 Problem Definition and Related Work

The Replica Placement Problem (RPP), which belongs to the NP-complete class [4], consists in finding the best set of servers to place content replicas over the Content Distribution Network (CDN). The problem tackled in this work extends the RPP and is called the Replica Placement and Requests Distribution Problem (RPRDP). This is a dynamic and online problem. In addition to finding the best position for the replicas it is also necessary to distribute the requests through the servers, with the aim at reducing the network load without violating the QoS constraints.

In this work, two QoS constraints are considered: minimal bandwidth and maximum delay.

Multiple servers are allowed to handle the same request at the same time, aiming at improving the quality of service and reducing the time needed to handle a request since limitations of servers and clients are taken into account.

In RPRDP a request may be handled by a server, totally or partially, only if that server has a replica of the content specified in the request. Also, the QoS constraints of the request must be verified. If these constraints are violated, redirecting this request to that server will produce an infeasible solution.

In this work we assume all contents are positioned in their origin servers and then they are replicated over the network. As there may exist costs associated with the transmission of these replicas, the tradeoff between the reduction of load and the cost of transmission must be analyzed beforehand.

The problem is dynamic and online, meaning that costs of communication can change, new contents and requests can come up and the future scenario is not known.

There are a number of recent solutions proposed for CDN management problems [8] [5] [10] [7] [12] [4] [11] [9]. However, the RPRDP remains open and frequently improtant issues like QoS constraints, network capacity and server load are not considered in this problem. This paper proposes and investigates exact and heuristic approaches to address this important problem.

2 Mathematical Formulation

A mathematical formulation considering all the characteristics of the problem is proposed in this paper. Although the use of mathematical formulation is not the most appropriate approach for online problems, it is a key technique for obtaining bounds. It is important to notice that in the offline version of the problem, solved using the formulation, all information about costs and requests is avaiable, allowing a better replica allocation and reduction of operational costs.

The proposed formulation is based on the one presented by [4] and uses concepts of [6].

Let δ be the length of a period in seconds, $origin(i)$ be the server where request i comes from, $ld(i)$ be the local delay of request i, $delay(j1, j2, t)$ be the delay between two servers and $RTT(j1, j2, t)$ be the round trip time between servers $j1$ and $j2$ on period t. The demand of a request in a certain period (D_{it}) is set to the maximum bandwidth that request can bear (BX_i) until the content is delivered.

The formulation uses the following variables: x_{ijt} is the fraction of content asked by request i handled by server j in period t; y_{kjt} assumes 1 if content k is replicated on server$j$ in period t, and 0 otherwise; b_{it} represents the backlog of request i on period t; w_{kjlt} assumes 1 if content k is copied by server j from server l on period t, and 0 otherwise.

The following notations are used. Let R be the set of requests, S be the set of servers, C the set of contents and T the set of time periods. Also let L_k be the size of content k and B_k be the period that content k is submited to the CDN. E_k is the period that content k is removed from the CDN, O_k the origin server of content k, AS_j is the available disk space on server j, MB_j is the maximum bandwidth of server j, D_{it} is the demand of request i on period t, BR_i is the minimal bandwidth of request i, BX_i is the maximum bandwidth request i can bear, G_i is the content asked by request i, c_{ijt} is the cost of handling request i by server j, on period t, given by the equation $c_{ijt} = (RTT(origin(i), j, t) + delay(origin(i), j, t) + ld(i)) \times BR_i$ and p_{it} is the backlog penalty paid for request i on period t.

$$
\begin{aligned}
Min & \sum_{i \in R} \sum_{j \in S} \sum_{t \in T} c_{ijt} x_{ijt} + \sum_{i \in R} \sum_{t \in T} p_{it} b_{it} \\
& + \sum_{k \in C} \sum_{j \in S} \sum_{l \in S} \sum_{t \in T} L_k w_{kjlt} && (1)
\end{aligned}
$$

$$
\begin{aligned}
S.a. & \\
& \sum_{j \in S} L_{G_i} x_{ijt} - b_{i(t-1)} + b_{it} = \delta D_{it} \\
& \forall i \in R, \forall t \in \{B_{G_i}, E_{G_i}\} && (2) \\
& \sum_{i \in R} L_{G_i} x_{ijt} \leq \delta MB_j \ \forall j \in S, \forall t \in T && (3) \\
& \sum_{j \in S} L_{G_i} x_{ijt} \leq \delta BX_i \ \forall i \in R, \forall t \in T && (4) \\
& \sum_{j \in S} \sum_{t \in T} x_{ijt} = 1 \ \forall i \in R && (5) \\
& y_{G_i jt} \geq x_{ijt} \ \forall i \in R, \forall j \in S, \forall t \in T && (6) \\
& \sum_{j \in S} y_{kjt} \geq 1 \ \forall k \in C, \forall t \in [B_k, E_k] && (7) \\
& y_{kjt} = 0 \ \forall k \in C, \forall j \in S, \forall t \notin [B_k, E_k] && (8) \\
& y_{kO_k B_k} = 1 \ \forall k \in C && (9) \\
& y_{kjB_k} = 0 \ \forall k \in C, \forall j \in \{S | j \neq O_k\} && (10) \\
& y_{kj(t+1)} \leq \sum_{l \in S} w_{kjlt} \ \forall k \in C, \forall j \in S, \forall t \in T && (11) \\
& y_{kjt} \geq w_{kljt} \ \forall k \in C, \forall j, l \in S, \forall t \in T && (12) \\
& \sum_{k \in C} L_k \ y_{kjt} \ \leq \ AS_j \ \forall j \in S, \forall t \in T && (13) \\
& x_{ijt} \in [0, 1] \ \forall i \in R, \forall j \in S \forall t \in T && (14) \\
& y_{kjt} \in \{0, 1\} \ \forall j \in S, \forall k \in C \forall t \in T && (15) \\
& b_{it} \geq 0 \ \forall i \in R, \forall t \in T && (16) \\
& w_{kjlt} \in \{0, 1\} \ \forall j, l \in S, \forall k \in C \forall t \in T && (17)
\end{aligned}
$$

The objective function (1) minimizes the operational costs and the backlog. Constraints (2) relate demand and backlog. Servers bandwidth are controlled by constraints (3). Constraints (4) prevent that a request receive more than it can bear. Constraints (5) guarantee that every request is fully handled. Constraints (6) impose that a request must be handled by a server that has a replica of the desired content. Constraints (7) and (8) control the number of replicas of active contents. Constraints (9) and (10) make sure that only origin server has a replica of a content on the submission period. Constraints (11) guarantee that all replications create a new replica. Constraints (12) make sure that a replication occurs from a server that has the content. The servers disk space are controlled by constraints (13). The remaining constraints are the integrality and non-negativity constraints.This formulation, called *FD*, is used to solve several instances of the problem, and its performance is shown in Section 4.

3 A Heuristic for the Problem

A heuristic to the online version of the problem based on CORA [11] is presented in this section.

Observing the formulation in Section 2, it becomes clear that the formulation deals with a mixed integer problem. It is easy to see that the continuous variables are related to the association of requests to servers and the integer ones are associated with the replica placement in servers.

Algorithm 1 Procedure HC

```
solve the positioning problem for period 0
for all periods left do
    Exactly solve the server-request association
    Heuristicaly solve the replication problem for next period
end for
return the solution
```

The *HC* heuristic, shown in Algorithm 1, proceeds as follows. For all periods, the algorithm solves the association between servers and requests using a mathematical formulation extracted from *FD* and shown in (18)-(24). For replica placement of next period, the algorithm uses an average of demands on previous periods, trying to forecast where the higher spots of demand will be, and uses a greedy heuristic for replica positioning.

To solve the association between requests and servers the following notations had their meaning to the current period:

Variables: x_{ij} is the fraction of content asked by request i handled by server j. b_i is the backlog of request i.

Constants: B_i is the previous backlog of request i. Y_{kj} assumes 1 if content k is replicated on server j and 0 otherwise. D_i is the demand of request i in this period. c_{ij} is the cost of handling request i from server j, calculated by the following equation: $c_{ij} = (RTT(origin(i), j) + delay(origin(i), j) + ld(i)) \times BR_i$. p_i is the penalty for backlogging request i.

All other notations that appear on the formulation keep the same meaning presented in Section 2.

$$Min \quad \sum_{i \in R} \sum_{j \in S} c_{ij} x_{ij} + \sum_{i \in R} p_i b_i \tag{18}$$

$$S.a.$$

$$\sum_{j \in S} L_{G(i)} x_{ij} + b_i = \delta D_i + B_i \ \forall i \in R \tag{19}$$

$$\sum_{i \in R} L_{G(i)} x_{ij} \leq \delta M B_j \ \forall j \in S \tag{20}$$

$$\sum_{j \in S} L_{G(i)} x_{ij} \leq \delta B X_i \ \forall i \in R \tag{21}$$

$$x_{ij} \leq Y_{G(i)j} \ \forall i \in R, \forall j \in S \tag{22}$$

$$x_{ij} \in [0,1] \ \forall i \in R, \forall j \in S \tag{23}$$

$$b_i \geq 0 \ \forall i \in R \tag{24}$$

Remark that this formulation for the subproblem is linear, thus, there are no integer variables.

To solve the replica positioning a greedy heuristic is used. The heuristic tries to insert in each server the contents that clients of this server demands more. If there is enough disk space, the replicas are simply placed. Otherwise, the heuristic tries to remove replicas with lesser demand from the server.

4 Results

The algorithms presented on previous sections were implemented in C++ using g++ version 4.3 and executed on a Quad-Core with 2.83 GHz/core, 8 Gigabytes of RAM using Linux (kernel 2.6). To solve both the formulations the solver CPLEX 11.2 [3] was used. To the best of authors knowledge there are no instances for this problem avaliable, so 60 instances were generated using different numbers of servers, contents and requests. These instances were generated using BRITE [1] topology generator, that tries to generate topologies similar to real networks, and are available on the LABIC [2] website.

Table 1 summarizes the results obtained. First column contain the names of the instances, on columns two and three are the number of requests and the number of contents. On columns four and five are the execution times, in seconds, of *FD* and *HC* respectively. On the sixth column are the gaps between these methods. Remark that although the *HC* heuristic does not have any knoledge about future the highest gap was under 21% of the optimal solution. The instances marked with * are instances where the optimal was not proved and the ones marked with ** are instances that the CPLEX processes were aborted for memory reasons. In both cases the result exposed in Table 1 are the best results found.

Table 1: Results

Inst.	# Requests	# Contents	Time FD(s)	Time HC (s)	Gap (%)
1001	613	5	1.21	0.06	1.47
1002	526	6	1.04	0.06	3.81
1003	519	5	1.15	0.09	2.82
1004	632	6	1.28	0.08	2.55
1005	445	5	0.84	0.06	1.32
1006	627	12	6.02	0.26	4.52
1007	675	11	5.00	0.26	4.18
1008	641	13	4.86	0.24	5.16
1009	659	14	4.98	0.26	4.12
1010	649	15	5.07	0.26	6.53
1011	627	12	38.22	0.27	10.46
1012	675	11	9.14	0.28	5.27
1013	641	13	15.14	0.26	9.25
1014	659	14	30.30	0.26	12.29
1015	649	15	32.93	0.26	20.82
2001	715	4	3.47	0.13	2.87
2002	1525	4	8.93	0.31	0.2
2003	1602	5	7.50	0.28	1.61
2004	1224	5	6.76	0.25	2.34
2005	1366	4	6.23	0.26	1.19
2006	1289	15	29.43	0.96	5.99
2007	1356	11	27.58	1.07	3.7
2008	1314	13	31.04	0.93	4.44
2009	1352	14	29.43	1.01	5.06
2010	1367	12	28.57	0.94	3.9
2011	1289	15	666.83	0.95	20.07
2012	1356	11	230.75	0.99	6.14
2013	1314	13	317.04	0.93	10.93
2014	1352	14	520.43	0.95	11.29
2015	1367	12	175.98	0.96	6.06
3001	2190	5	20.77	0.6	0.65
3002	1793	6	14.86	0.42	2.91
3003	1845	6	15.25	0.51	1.9
3004	1896	6	17.51	0.54	2.83
3005	2384	6	19.12	0.65	1.7
3006	2007	12	92.25	2.1	4.6
3007	1963	12	80.79	2.61	4.56
3008	2021	11	73.5	2.16	4.51
3009	1991	11	72.12	2.08	3.71
3010	1998	11	68.56	2.62	4.02
3011	2007	12	433.04	2.16	8.13
3012	1963	12	211.01	1.98	5.93
3013	2021	11	196.27	2.1	6.42
3014	1991	11	157.09	2.06	5.1
3015	1998	11	188.9	1.95	5.62
5001	3360	4	59.98	1.73	0.55
5002	3231	6	63.00	1.4	2.94
5003	3534	6	62.16	1.66	1.77
5004	3646	4	60.19	1.56	6.95
5005	3762	5	58.57	1.9	1.77
5006	3391	11	278.16	6.17	3.75
5007	3329	11	269.80	5.68	4.6
5008	3214	15	357.13	5.69	5.61
5009	3303	15	347.39	6.07	5.72
5010	3295	13	315.46	5.46	4.97
5011*	3391	11	523.88	8.18	5.5
5012*	3329	11	379.95	5.74	5.49
5013**	3214	15	1229.67	5.61	14.89
5014**	3303	15	1239.84	5.83	16.44
5015*	3295	13	666.31	5.43	8.88

5 Conclusion

This paper presents a brief description of Replica Placement and Request Distribution problem, proposes a mathematical formulation to offline version and a heuristic for online version of the problem. We claim that the Replica Placement Probem and the Request Distribution Problem are correlated and therefore should be solved together, instead of solving these two problems in a separated way as seen in the literature [12]. Besides that, minimal bandwidth , maximum delay, better use of network capacity and servers load are considered. To the best of authors knowledge, this is the first work to deal with this problem considering so many realistic details.

The results show that the heuristic can reach good results in much less computational time, observing that the formulation is applied to the offline version of the problem.

A detailed analysis must be done to discover if the GAP, presented by the heuristic, is caused by the difference between the online and offline versions of the problem, or if it is caused by the natural difference between exact and heuristic methods.

References

[1] BRITE. World Wide Web, http://www.cs.bu.edu/brite/. 06/2007.

[2] LABIC. World Wide Web, http://labic.ic.uff.br/. 02/2009.

[3] ILOG S.A., CPLEX 11 user's manual, 2008.

[4] Aioffi, W., Mateus, G., Almeida, J., Loureiro, A. Dynamic content distribution for mobile enterprise networks. *IEEE Journal on Selected Areas in Communications 23*, 10 (2005), 2022–2031.

[5] Almeida, J. M., Eager, D. L., Vernon, M. K., Wright, S. J. Minimizing delivery cost in scalable streamingcontent distribution systems. *IEEE TRANSACTIONS ON MULTIMEDIA 6* (2004), 356–365.

[6] An, C., Fromm, H., Eds. *Supply Chain Management on Demand: Strategies and Technologies, Applications.* Springer, 2005.

[7] Bartolini, N., Presti, F., Petrioli, C. Optimal dynamic replica placement in content delivery networks. In *The 11th IEEE International Conference on Networks 2003. ICON2003* (2003), p. 125–130.

[8] Bektas, T., Oguz, O., Ouveysi, I. Designing cost-effective content distribution networks. *Computers & Operations Research 34* (2007), 2436–2449.

[9] Coppens, J., Wauters, T., Turck, F. D., Dhoedt, B., Demeester, P. Design and performance of a self-organizing adaptive content distribution network. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP* (2006), p. 534–545.

[10] Huang, C., Abdelzaher, T. Towards content distribution networks with latency guarantees. In *Twelfth IEEE International Workshop on Quality of Service, 2004. IWQOS 2004* (2004), p. 181–192.

[11] Tenzakhti, F., Day, K., Ould-Khaoua, M. Replication algorithms for the world-wide web. *Journal of Systems Architecture 50* (2004), 591–605.

[12] Zhou, X., Xu, C.-Z. Efficient algorithms of video replication and placement on a cluster of streaming servers. *Journal of Network and Computer Applications 30* (2007), 515–540.