

## 선형 구조상에서의 복제 웹 서버 배치 알고리즘

김선호<sup>o</sup>, 고훈, 신용태

동덕여자대학교 정보학부, 대진대학교 컴퓨터공학과, 숭실대학교 컴퓨터학과  
shkim98@dongduk.ac.kr, shkim98@dongduk.ac.kr, shin@comp.ssu.ac.kr

### Replica Placement Algorithm on the Linear Topology

Seonho Kim<sup>o</sup>, Hoon Ko, Yongtae Shin

Dept. of Computing and Engineering, Daejin University<sup>o</sup>  
Dept. of Information and Science, Dongduk Women's University  
Dept. of Computing, Soongsil University

### 요 약

최근 웹 서비스 사용자와 웹상에 대용량 콘텐츠의 급증으로 웹 서버의 부하가 가중되고 서비스의 질이 떨어지는 문제가 발생하였다. 이러한 문제의 해결로 콘텐츠를 다수의 지역 서버에 복제하고 복제 서버로 하여금 클라이언트의 요청을 처리하도록 하여 웹 서버의 부하를 줄이고 트래픽을 분산하는 콘텐츠 분산 네트워크 기술이 사용되고 있다. 이러한 다수의 복제 서버 사용하는 환경에서는 복제 서버의 효율을 극대화 할 수 있는 배치 전략이 필요하다. 본 논문에서는 복제 서버의 효율을 위하여 선형 구조상에서 지연과 트래픽에 기반 한 각 노드의 비용이 임계값을 초과하지 않도록 복제 서버를 배치할 수 있는 방안을 제시하여  $O(MN)$ 의 시간복잡도로 문제를 해결하였다.

### 1. 서론

인터넷 기술의 발달로 웹을 통한 대용량 멀티미디어 서비스가 급증하고 이에 따라 서버와 네트워크의 부담이 가중되고 있다. 웹은 클라이언트-서버 모델을 기본으로 하기 때문에 웹 서버로의 부하 집중은 트래픽의 병목을 발생시키고 사용자의 대기 시간을 증가시키고 있다.

이러한 문제의 해결을 위해 사용자에게 의해 자주 요청되는 데이터를 사용자와 가까운 곳에 보관하여 서비스하는 캐싱(Caching), 특정 사이트에 가중되는 부하를 분산시키기 위하여 사이트의 일부 또는 전부를 복사하여 다른 사이트에 저장하여 운영하는 미러링(Mirroring) 기술이 사용되었으며, 최근에는 이러한 기술을 한 단계 발전시켜 원본 서버의 데이터를 사용자와 가까운 위치에 설치된 복제 웹 서버로 복사하여 사용자로 하여금 인터넷 미들 마일을 거치지 않고 바로 가까운 복제 서버로부터 서비스 받도록 하는 CDN(Content Delivery Network)과 서버의 효율적인 자원 분산을 위해 사용자간의 협력으로 자원을 공유하여 사용하고자 하는 P2P(Peer To Peer) 기술이 대두되었다[1].

CDN은 웹 서버의 부하를 분산하여 네트워크 대역폭의 낭비를 줄이고 응답시간을 단축하기 위하여 콘텐츠를 원본 서버로부터 인터넷 곳곳에 흩어진 복제 서버에 복사해 놓고 가까이에 있는 복제 서버로 하여금 클라이언트의 요청에 응답하게 하는 서비스이다.

그러나 이러한 기술들의 성능을 향상시키기 위하여 중

요한 이슈가 될 수 있는 복제 서버의 배치 전략에 관한 연구는 활발하지 않은 상황이다. 적절한 수의 복제 서버의 할당과 배치는 시스템의 전체적인 성능에 크게 영향을 미칠 수 있으며 사용자가 웹 서버에 접근하는데 드는 비용과 지연 시간을 줄일 수 있다. 기존에 캐쉬와 프락시 서버의 위치 선정에 관한 연구들은[3-8] 최적의 프락시 서버 수를 알아내는 Greedy 알고리즘과 선택된 프락시 서버를 적절한 곳으로 배치하기 위한 NP-hard 문제로 접근하여 해결하고 있으나 복잡한 계산 시간을 필요로 하며 복제 웹 서버와는 성격이 다르기 때문에 그대로 적용하기는 어려운 상황이다.

이에 본 연구에서는 인터넷상의 노드들이 선형 구조를 이루고 있다고 가정하고 적절한 복제 서버의 위치와 개수를 찾기 위한 알고리즘을 제안한다.

### 2. 관련연구

복제 웹 서버의 적절한 수와 위치 선정 문제는 기존의 프락시 서버의 배치 문제와 연관지어 생각해 볼 수 있는데, 대부분의 기존 프락시들은 지연을 줄이기보다는 네트워크 혼잡을 줄이는 것에 초점을 두고 보통 적관적으로 중요한 지점이라고 생각할 수 있는 LAN의 라우터나 인터넷의 게이트웨이 또는 기업이나 조직의 전략적 위치에 배치되었다[2].

그러나 분산 웹 서버 환경에서 효율적으로 사용자의 접속 지연을 줄이고 네트워크 대역폭 사용을 줄이기 위해서는 얼마나 많은 수의 복제 서버를 네트워크의 어디

에 위치시킬 것인가에 관한 연구가 필요하다.

[3]에서는  $n$ 개의 노드에서 CDN 서버가 될 수 있는  $k$ 개의 노드를 선택하는데 있어서 특정 노드  $j$ 가 CDN 서버에 할당되는 경우  $d_j$ 를 노드  $j$ 에 대한 수요,  $C_{ij}$ 를 노드  $i$ 와  $j$ 사이의 거리라고 할 때, 비용  $d_j C_{ij}$ 를 최소화 하는  $k$ 를 선택함으로써 최적의 위치를 알아내는 연구를 하였다. AS(Autonomous System) 레벨의 인터넷 토폴로지를 모델링하여 시뮬레이션 한 결과 거리와 요청 부하를 기반으로 Greedy 알고리즘에 의한 배치가 가장 좋은 성능을 나타내었다

[4, 5]에서는 웹 프락시의 배치를 위한 연구가 시도되었는데 노드들이 선형 토폴로지를 이룰 때와 트리 토폴로지를 이룰 때 시스템 자원과 트래픽 패딩을 고려하여 지연을 최소화 할 수 있는 알고리즘을 제안하였다. 여기에서는 프락시의 개수가  $M$ 개로 지정되어 있음을 가정하고 다이나믹 프로그래밍 알고리즘을 이용하여  $M$ 개 프락시의 배치 문제를 해결하였으나 트리 토폴로지의 경우  $N$ 개의 노드에  $M$ 개의 프락시 서버를 배치하는데  $O(N^3M^2)$ 의 시간복잡도를 필요로 한다.

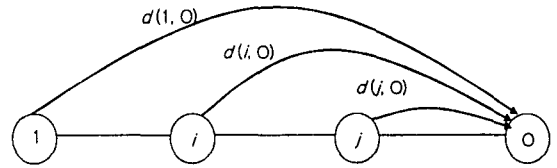
[6, 7]에서는 클라이언트와 복제 웹 서버의 거리를 최소화 할 수 있는 복제 서버의 배치 문제를  $k$ -median problem 으로 접근하여 해결하였으며 BGP 라우팅 테이블을 이용하여 실제 인터넷 토폴로지를 모델링하여 시뮬레이션 하였다. [7]은 보다 정확한 결과를 위하여 AS와 라우터 레벨의 인터넷 토폴로지 상에서 모델링하여 degree가 큰 노드에 복제 서버를 배치하는 것이 Greedy에 비해서 성능이 더 좋음을 보였다. 그러나 모든 노드가 비슷하게 degree가 적은 경우는 적용하기에 적합하지 않다.

[8]에서는 선형, 링 토폴로지에서의 캐쉬 프락시 배치 알고리즘을 제안하였다. 각 알고리즘의 시간 복잡도는  $O(NM^3)$ ,  $O(NM^4)$ 으로 나타났다.

### 3. 복제 서버 배치 알고리즘

제안하는 알고리즘은 원본 웹 서버로부터 모든 노드들이 선형구조를 이루고 있음을 가정하여 트래픽 양과 지연시간에 따른 각 노드의 비용이 정해진 임계값을 초과하지 않도록 복제 서버를 배치함으로써 복제 서버의 적절한 위치와 개수를 도출한다.

<그림 1>에서 0은 원본 웹 서버를 의미하며  $i, j$  노드는 복제 웹 서버가 될 수 있는 잠재 노드들이며 1번 노드가 원본 웹 서버에서 가장 먼 노드이다.  $N$ 개의 노드에서 임의의 노드  $i$ 에서는 자신과 하위 노드간의 거리 차이에 의해 생기는 지연 시간과 하위 노드로부터 발생되어 지나가는 트래픽이 발생하게 된다. 그러므로 각 노드의 비용은 해당 노드를 지나가는 누적된 트래픽 양과 지연시간으로 결정한다.



<그림 1> 선형구조에서의 노드 비용

그러므로 각 노드의 비용은 수식 (1)과 같이 해당 노드를 지나가는 트래픽의 양( $w$ )과 하위 노드와의 거리 차이에서 생기는 링크의 시간 지연( $d$ )으로 나타낼 수 있다.

$$Cost(i) = \sum_{i=1}^{i-1} (w(i-1) * d(i-1, i)) \quad (1)$$

$i$ 의 비용이 서비스 임계값( $\mathcal{L}$ ) 보다 크면 이전 노드가 복제 서버가 된다. 그러면 복제 서버의 아래 노드들은 모두 복제 서버로부터 서비스를 받음으로 현재 노드의 비용 산정에는 고려할 필요가 없으므로 현재 노드의 비용은 0 이되며 다시 이후 노드들의 비용을 계산하면서 임계값을 초과하는 노드를 찾아 복제 서버를 배치하는 것이다. 임계값은 지연과 대역폭의 곱으로 정할 수 있는데 지연값은 이전의 지연 평균과 현재의 지연을 고려하며 대역폭은 현재 사용가능한 대역폭을 의미한다.

임계값을 설정함으로써 모든 노드가 최대 대역폭을 넘지 않으며 지연값에 대해서는 지수함수를 사용함으로써 일정 수준의 지연을 보장할 수 있다. 문제해결을 위한 알고리즘은 <그림 2>와 같으며  $O(MN-I)$ 의 시간복잡도로 문제를 해결하였다.

<b>Algorithm</b>
$\mathcal{L}$ : 서비스 임계값 $w(i)$ : node $i$ 에서 발생하는 트래픽 양 $d(i-1, i)$ : node $i$ 와 $i-1$ 사이의 시간 지연 $k$ : 복제 서버의 수 $r_k$ : 복제 서버가 되는 노드
$k, start, i = 1$
for $n = 1$ to $N$ if $n = start$ $Cost(n) = 0$ ; else $Cost(n) = \sum_{i=1}^{n-1} (w(i-1) * d(i-1, i))$ ; if $Cost(n) > \mathcal{L}$ { $r_k = n-1$ ; $start = n$ ;

```

    k++; }
n++;
    
```

<그림 2> 서버배치 알고리즘

4. 수 예제를 통한 분석

<그림 3>과 같은 네트워크 상태에서 제안한 방법에 의해 복제 서버 배치 문제를 해결한다. 1노드는 원본 서버에서 가장 먼 노드를 의미하며 7노드 다음이 원본 서버인 경우의 예이다. 접근 비용은 각 노드들이 원본 서버를 포함한 각 복제 서버까지 접근하는데 드는 비용을 의미하며 각 링크 상의 지연값은 실제 ping 테스트로 100여개 대학 사이트를 상대로 측정된 평균 지연값 9.02을  $\frac{1}{\lambda}$ 로 하는 지수함수 분포에 의해 임의로 생성한다. 지연값을 생성하는 식은 다음 수식 (2)와 같으며 임계값  $\mathcal{L}$ 은 40이 된다.

$$d = -\frac{\ln R}{\lambda}, \quad 0 < R < 1 \quad (2)$$

노드 <i>i</i>	1	2	3	4	5	6	7
<i>i</i> 의 트래픽 양	3	2	4	2	5	4	3
<i>i</i> 와 이전 노드 사이에서의 지연	0	5	3	7	6	2	3

<그림 3> 각 노드의 트래픽 양과 지연

첫 번째 노드부터 원본 서버까지 가는 동안 각 노드의 비용을 구하여 복제 서버의 위치를 선정한다.

$$\begin{aligned}
 \text{Cost}(1) &= 0 \\
 \text{Cost}(2) &= w(1)d(1,2) = 3*5 = 15 \\
 \text{Cost}(3) &= w(1)d(1,3) + w(2)d(2,3) = 3*8 + 2*3 = 30 \\
 \text{Cost}(4) &= w(1)d(1,4) + w(2)d(2,4) + w(3)d(3,4) \\
 &= 3*15 + 2*10 + 4*7 = 93
 \end{aligned}$$

노드 4의 비용이 임계값을 초과하므로 노드 3이 첫 번째 복제 서버가 되어야 한다. 노드 3이 복제 서버가 되면 노드 1, 2는 노드 3으로부터 서비스를 받으므로 노드 4부터는 이전의 비용을 생각할 필요가 없이 노드 4의 비용은 0이 되고 이제부터 노드 4가 서비스 받을 복제 서버를 선택하면 된다.

$$\begin{aligned}
 \text{Cost}(4) &= 0 \\
 \text{Cost}(5) &= w(4)d(4,5) = 2*6 = 12 \\
 \text{Cost}(6) &= w(4)d(4,6) + w(5)d(5,6) = 2*8 + 5*2 = 26 \\
 \text{Cost}(7) &= w(4)d(4,7) + w(5)d(5,7) + w(6)d(6,7) \\
 &= 2*11 + 5*6 + 4*3 = 64
 \end{aligned}$$

노드 7의 비용이 임계값을 초과하므로 노드 6이 두 번째 복제 서버가 되어야 한다. 그러므로 3번 6번 노드가 복제 서버가 되고 복제 서버의 수는 2개가 된다.

5. 결론

본 연구에서는 선형 구조상에서 적절한 복제 서버의 배치에 관한 문제를 다루었다. 문제의 해결 방법은 복제 서버가 서비스하는 비용이 지연과 트래픽에 기반하여 일정 비용을 넘지 않도록 하여  $O(MN-I)$ 의 시간복잡도로 복제 서버를 배치하였다.

6. 참고문헌

- [1] G. Peng, "CDN: Content Distribution Network", in *Stony Brook University Tech. Reports*, TR-125, 2003.
- [2] Masaaki Nabeshima, The Japan CacheProject: An Experiment on Domain Cache, In *Proceedings of the Sixth International WWW Conference*, Santa Clara, CA, 1997
- [3] Lili Qiu, Venkata N. Padmanabhan, and Geoffrey M. Voelker. On the Placement of Web Server Replicas. In *Proceedings of INFOCOM 2001*, Anchorage, AK, USA, April 2001.
- [4] B. Li, X. Dong, J. Golin, and K. Sohaby, On the Optimal Placement of Web Proxies in the Internet: Linear Topology, in *proceedings of 8th IFIP Conference on High Performance Networking*, HPN'98.
- [5] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohaby, "On the optimal placement of Web proxies in the Internet," in *Proceedings of IEEE INFOCOM'99*, Mar. 1999.
- [6] Sugih Jamin, Cheng Jin, Yixin Jin and Danny Raz and Yuval Shavitt and Lixia Zhang", On the Placement of Internet Instrumentation, in *IEEE Infocom 2000*
- [7] Pavlin Radoslavov, Ramesh Govindan, Deborah Estrin, *Topology-Informed Internet Replica Placement*, *Proceedings of WCW'01*.
- [8] P. Krishnan, Danny Raz, Yuval Shavitt, *The CacheLocationProblem*, *IEEE/ ACM Transactions on Networking*, 8(2):568-582, Oct. 2000.