

CDN에서의 서버 상태 및 응답 시간을 고려한 서버 선택 알고리즘

한상희^o 김상하

충남대학교 컴퓨터학과

(shhan^o, shkim)^o@cclab.cnu.ac.kr

A Novel Server Selection Based on Round-Trip Time and Serving Calls in CDN

Sang-Hee Han^o Sang-Ha Kim

Dept. of Computer Science, Chungnam National University

요 약

CDN (Content Delivery Network) [1]은 분배서버와 네트워크의 상태를 고려하여 다양한 서비스의 콘텐츠를 보다 효율적으로 전송할 수 있도록 설계된 네트워크이다. 이러한 CDN을 구성하기 위해서는 네트워크 계획, 콘텐츠 할당, 분배서버 선택 등을 고려해야 한다. 특히, 분배서버 선택을 위한 기존의 연구들은 분배서버 선택 시 모든 분배서버에 대한 측정된 값 (응답시간, 지연)이나 실시간 정보를 기준으로 분배서버를 선택하게 되므로 호 붕괴 확률을 고려하지 않게 되고 제어 프로토콜로 인한 오버헤드가 많아지게 된다. 이를 해결하기 위하여 본 논문에서는 분배서버에서 서비스하고 있는 호의 수와 응답 시간을 고려한 새로운 서버 선택 알고리즘을 제안한다. 제안된 알고리즘에서 CDN의 분배서버들은 호의 수에 따라 논리적 최소 힙(heap) 구조로 구성되며 이를 이용하여 각 호스트는 자신에게 적당한 분배서버들의 서버 집합을 구성함으로써 오버헤드를 줄일 수 있는 메커니즘을 제안한다.

1. 서론

CDN은 다양한 웹 콘텐츠, 그 중에서 대역폭이 큰 콘텐츠들의 전송을 향상시키기 위하여 여러 분배서버들을 적절히 배치하고 이들을 통하여 서비스를 제공할 수 있게 하는 네트워크를 일컫는다. 이러한 CDN을 구성하기 위해서는 기본적으로 각 분배서버들을 네트워크의 어느 부분에 배치할 것인가의 문제, 각 서버에 어떠한 콘텐츠를 할당 할 것인가의 문제, 그리고 마지막으로 사용자들이 어떤 분배서버를 선택하게 할 것인가에 대한 세가지 큰 이슈가 존재하게 된다. 앞에서 살펴본 이슈 중에서 앞의 두 가지 문제는 서비스 제공자 측면에서의 성능 향상에 관한 문제인 반면 마지막 문제는 사용자 측면에서의 효율적인 알고리즘을 통하여 성능 향상을 도모할 수 있는 문제로 인식되고 있다.

이러한 효율적인 분배서버 선택을 위한 다양한 메커니즘이 제안되고 있다. 대표적인 메커니즘으로는 DNS(Domain Name System) 리디렉션[2]이 있다. [2]는 많은 상용 CDN 서비스에서 사용하고 있는 기술로써 사용자의 지역 DNS서버로부터 DNS 서비스 요청을 받은 권위 있는 DNS서버가 가장 가까운 분배 서버의 정보를 알려주게 된다. 그러나, 서버 선택 시에 라우터 홉수나 응답 시간과 같은 단순한 기준치를 이용하므로 지속적으로 서비스 질을 최적화하는 서버 선택을 할 수 없게 되며 호 붕괴 확률을 고려하지 않는 문제점이 생겨 나게 되었다. 이를 해결하기 위하여 실시간 정보를 이용하는 새로운 메커니즘인 [3]이 제안되었다. 이 메커니즘은 요청이 있거나 또는 주기적으로 모든 분배 서버들과 연결된 링크의 상태를 파악하여 최적의 분배서버를 선택하는 방법이다. 그러나 이 메커니즘 또한 정확한 실시간 정보를 얻기 위해서는 비용이 많이 들고, 실시간 정보 수집으로 인해 최대 부하 시에 혼잡을 발생시키는 원인이 될 수 있는 문제점을 남기게 되었다.

즉, 현재의 서버 알고리즘의 대표적인 문제점으로는 하나의 메트릭에 의존한 서버 선택 알고리즘으로 인하여 호의 서비스

확률이 낮아지거나 또는 이러한 서버 선택을 위한 제어 프로토콜로 인한 오버헤드가 급격히 증가한다는 것이다. 이를 해결하기 위하여 본 논문에서는 각 CDN을 구성하는 분배서버들을 현재 서비스 하고 있는 호의 수에 따라 논리적 최소 힙 구조로 구성하며 이를 통하여 적절한 서버 서버 집합을 구성하는 방법을 제안한다. 논리적인 최소 힙 구조를 통하여 각 호스트는 현재 가장 적은 수의 서비스를 제공하고 있는 분배서버를 빠르고 쉽게 찾을 수 있다. 또한 이들에 대한 응답 시간을 측정하여 네트워크 상태의 변화에 동적으로 대처할 수 있다. 두 파라미터의 값을 이용한 기준값 보다 메트릭이 작은 분배서버들과 루트에 해당하는 분배서버가 서버 서버 집합으로 구성되고 호스트는 그 중에서 가장 메트릭이 작은 서버를 선택하게 된다.

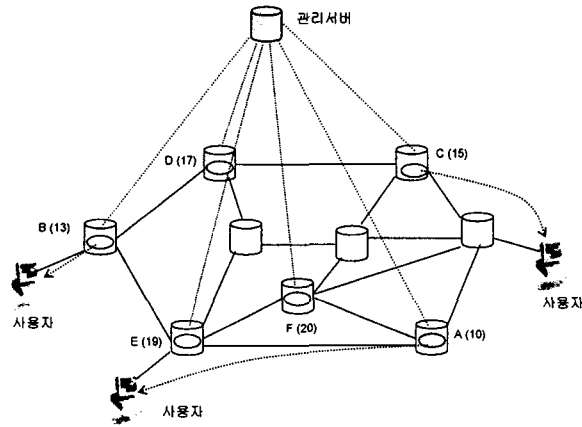
본 논문은 다음과 같이 구성된다. CDN에서의 서버 선택의 중요성과 그 문제점을 설명한 서론에 이어 2장에서는 제안된 메커니즘을 통한 CDN에서의 서버 선택 알고리즘을 제안한다. 마지막으로 3장에서는 결론을 맺는다.

2. 제안 메커니즘

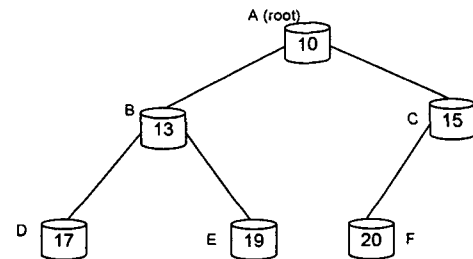
제안된 메커니즘을 통하여 [그림 1]에서처럼 CDN을 지원하기 위해서는 분산 환경을 지원하는 계층적이고 논리적인 구조가 필요하며 모든 분배서버들의 호의 수와 지속적인 상태를 파악하고 있는 관리서버가 새롭게 필요하게 된다. 관리 서버는 각 분배서버들이 현재 서비스를 제공하는 호의 수를 기준값으로 모든 분배서버들을 최소-힙 구조의 논리적인 형태로 유지하여야만 한다. 최소-힙을 이용하게 되면 현재의 서버들중에서 가장 작은 호를 서비스 하고 있는 CDN 분배서버를 가장 효율적으로 찾을 수 있게 된다.

이를 위해서 관리서버는 [그림 2]처럼 주기적으로 각 CDN 분배 서버들의 호의 수를 확인하여 최신의 정보를 가진 최소-힙을 유지하여야 한다. 만약, 특정 분배 서버의 고장으로

인하여 분배 서버에 문제가 발생하면 현재의 호의 수를 최대값으로 설정하여 최소-힙의 마지막 노드로 이동시킴으로써 사용자가 해당 분배 서버를 선택하는 것을 막을 수 있게 한다.



[그림 1] 관리 서버가 있는 CDN 네트워크



[그림 2] 호의 수를 이용한 분배서버들의 최소-힙 예

2.1 서버 선택 알고리즘

사용자가 자신에게 적당한 분배서버 선택 시에 관리서버가 가지고 있는 호의 수를 계산하여 만들어진 최소-힙만을 이용한다면 이는 호의 서비스율을 최대화 함으로써 효율성 측면에서는 장점을 가지게 된다. 하지만, 이러한 방법만을 적용하게 되면 네트워크 상태 변화에 능동적으로 대처할 수가 없게 된다. 따라서, 각 호스트에서 각 분배서버의 호의 수와 함께 그 서버까지의 응답 시간을 고려하는 서버 선택 알고리즘을 제안하였다. 각 호스트는 두 파라미터를 통하여 서버 서버 집합을 구성하게 되고 이 서버 서버 집합 중에서 가장 작은 메트릭을 가진 분배서버를 선택하여 서비스를 받게 된다. 다음은 이러한 알고리즘을 통한 서비스 요청 및 서버 서버 집합을 구성하는 방법, 그리고 서버 서버 집합을 변경하는 절차에 대하여 설명한다.

1) 서비스 요청과 서버 서버 집합 구성

사용자는 자신이 원하는 콘텐츠를 서비스 받고자 하는 경우, 관리서버에게 서비스 요청을 수행하게 된다. 관리서버는 각 요청에 대하여 현재 최소-힙의 루트와 루트의 자식 노드를 서비스를 요청한 호스트에게 전송하게 된다. 서비스를 요청한 호스트는 분배서버 리스트를 받은 다음, 자신의 서버 서버

집합을 만들기 위한 프로시저를 수행하게 된다. 이를 위해서는 각 분배 서버에 대한 성능 파라미터를 정의하여야만 한다. 이러한 성능 파라미터는 식 (1)에 의하여 정의되게 된다.

$$Server[i].metric = (1-w) * \text{분배 서버 } i \text{의 현재 호의 수} + w * \text{분배서버까지의 응답 시간} \quad (1)$$

식 (1)에서 알 수 있듯이 서버 서버 집합을 구성하기 위한 파라미터를 위한 값은 두 개의 파라미터로 구성되어 있다. 하나는 분배서버의 현재 호의 수이고 다른 하나는 분배서버까지의 응답 시간이다. 각 파라미터는 중요성에 따라 다른 가중치가 적용된다. 각 호스트의 서버 서버 집합은 Server[root].metric보다 작은 분배서버들과 루트 자신으로 구성되게 되며 이를 구성하기 위한 알고리즘은 다음과 같다.

호스트는 처음 서비스 요청 시에 루트와 그의 자식노드의 정보를 관리서버에서 요청한다. 이 과정을 통하여 각 호스트는 루트와 루트의 자식노드의 서버 리스트를 구할 수 있게 된다. 이 정보를 받은 후에 호스트에서는 서버 서버 집합을 구성하기 위한 기준으로 정할 server[root].metric과 각 자식노드로 설정되어 있는 분배서버들의 메트릭을 식 (1)을 이용하여 구하게 된다. 만약, 관리서버로부터 전달된 모든 분배서버들에 대한 메트릭 계산이 끝난 뒤 기준값보다 작은 분배서버가 존재한다면 서버 서버 집합은 결정되게 되며 이 과정은 중단된다. 그러나 기준값보다 작은 값을 가진 분배서버가 존재하지 않을 경우에는 자식노드 중에서 작은 메트릭을 가진 노드를 잠재적인 부모로 설정하여 다시 관리서버에게 잠재적인 부모의 자식노드를 요청하게 된다. 관리서버는 서비스 요청에 포함되어 있는 잠재적인 부모의 자식노드를 다시 호스트에게 알려주게 되며 앞의 서버 서버 집합을 구성하기 위한 과정을 계속하여 수행하게 된다. 만약 이 과정이 끝나도록 기준값보다 작은 값을 가지는 분배서버가 없을 경우, 서버 서버 집합은 루트만으로 구성되게 된다. 이러한 서버 서버 집합을 구성하는 과정에서는 다음과 같은 세 가지 경우가 존재하게 된다. 이 예는 [그림 2]를 기준으로 설명하고 있다.

- 1) 자식 분배서버들(B,C)의 메트릭이 모두 루트인 A의 메트릭(기준값)보다 작을 경우에는 A,B,C 모두가 서버 서버 집합에 삽입되게 된다..
- 2) 자식 분배서버들 중 하나만이 루트의 메트릭(기준값) 보다 작을 경우에는 루트의 기준값보다 작은 분배서버와 루트 자신이 서버 서버 집합에 삽입되게 된다.
- 3) 모든 자식 분배서버들(B,C,D,E,F)의 메트릭이 루트의 메트릭(기준값)보다 큰 경우에는 서버 서버 집합은 루트만으로 구성되게 된다.

이러한 메커니즘을 통하여 사용자는 가장 적은 호를 서비스 하고 있는 분배서버들을 빠르게 알 수 있으며 모든 분배서버에 대한 응답 시간 측정이 필요하지 않기 때문에 오버 헤드를 줄일 수 있게 된다. 그러나 서버 상태에 유연하게 대처할 수 있는 반면, 최악의 경우에는 모든 분배서버의 절반에 해당하는 서버에 대하여 메트릭을 측정해야 하는 어려움이 생길 수 있게 된다. 이러한 알고리즘은 [그림 3], [그림 4], [그림 5]와 같이 정의되게 된다.

2) 서버 서버 집합의 변경

호스트는 서비스를 제공 받기 위하여 서비스 요청 과정 중에 구성된 서버 서버 집합에 있는 분배서버에게만 서비스 요청을 할 수 있다. 첫 번째 분배서버에 서비스를 받고 있다가 호스트가 원하는 서비스를 받지 못하게 되는 경우, 서버 서버

집합의 다른 분배서버를 선택하게 되고 해당 서버로부터 서비스를 받게 된다. 만약, 서브 관리 집합에 남아 있는 분배서버가 호스트에게 충분한 서비스를 제공하지 못할 경우, 서비스 요청 과정을 다시 수행함으로써 새로운 서브 서버 집합을 구성 할 수 있다.

```

IF ( request == initial)
    Delivery the information of root, root's children to host.
ELSE
    Delivery the information of potential parent's children to host.
    
```

[그림 3] 관리서버에서의 알고리즘

```

IF (request == initial)
    Request of the information of root, root's children to admin server.
    Server[root].metric is derives from (1).
    compare_metric(root's children, server[root].metric)
ELSE
    compare_metric( potential parent's children,
                    server[root].metric)
    
```

[그림 4] 호스트에서의 알고리즘

```

compare_metric(children, Server[root].metric)
{
    children's metric are derives from (1).
    IF children's metric are less than or equal to
    Server[root].metric
        Sub server set = {root, children}
    ELSE IF one of children's metric is less than or equal
    to Server[root].metric
        Sub server set = {root, child which is its metric is
        less than or equal to Server[root].metric }
    ELSE IF children's metric are greater than
        Server[root].metric
        potential parent is child which is its metric is less
        than other child's.
        Request of the information of potential parent's
        children to admin server.
}
    
```

[그림 5] 서브 서버 집합 구성 알고리즘

3. 결론 및 향후 연구 과제

본 논문에서는 CDN에서의 서버 선택을 위한 알고리즘을 제안하였다. 제안된 알고리즘의 경우, 각 분배서버에서 서비스를 하고 있는 현재 호의 수와 각 분배 서버까지의 응답 시간을 동시에 고려함으로써 호의 서비스율을 높일 수 있을 뿐만 아니라 적은 지연을 통하여 서비스가 가능한 서버를 선택할 수 있도록 한다. 이를 위하여 CDN의 분배서버를 최소-힙 구조로 구성하는 동시에 서브 서버 집합을 구성하기 위한 매트릭 도출을 위한 식을 제안하였다.

본 연구와 관련하여 우선적으로 시뮬레이션을 통하여 성능 향상에 대한 정량적인 비교 분석을 수행할 것이며 이러한 시뮬레이션에는 식 (1)에서의 호의 수와 응답 시간의 연관성 및 가중치(w)의 설정이 전체적인 CDN 성능에 얼마나 영향을 주는 지에 초점이 맞추어 질 것이다.

참고문헌

- [1] K. Fujikawa et al., "A Server Selection Method Based on Communication Delay and Communication Frequency among Users for Networked Virtual environments," Proceedings of 7th Asian Computing Science Conference, pp.125-139, December 2002.
- [2] K.L.Johnson et al., "The Measured Performance of Content Distribution Networks," 5th International Web Caching and Content Delivery Workshop, May 2000.
- [3] S. Makabe et al., "Dynamic mirror server selection method with consideration about fluctuation of network load," IEICE Trans. On Communications, vol.J84-B, no.3, pp.435-442, March 2001.
- [4] N. Kamiyama , "A Server Selection Method in Content Delivery Networks," IEICE Trans. Commun., vol.E86-B, no.6, pp.1796-1804, Jun 2003.
- [5] A. Shaikh et al., "On the Effectiveness of DNS-based Server Selection," Proc. IEEE INFOCOM 2001, April 2001.
- [6] R.L.Carter et al., "Server selection using dynamic path characterization in wide-area networks," IEEE INFOCOM'97,pp.1014-1021,1997.
- [7] B.Zhang et al., "Host Multicast: A Framework for Delivering Multicast To End Users," Proc.of IEEE INFOCOM'02, June 2002.