

# 웹로그를 이용한 부분 멀티미디어 서버의 동적 선택 알고리즘

이경희<sup>\*</sup>, 한정혜

충북대학교 전자계산학과, 행정자치부 국가전문행정연수원

khlee@trut.chungbuk.ac.kr, hanjh@nipa.go.kr

## Dynamic Selection Algorithms for Replicated Multimedia Servers by Analyzing their Web Logs

Kyung Hee Lee<sup>\*</sup>, Jeong-Hye Han

Dept. of Computer Science Chungbuk Nat'l Univ.,

National Institute of Professional Administration

### 요 약

인터넷 망을 통한 멀티미디어 콘텐츠 서비스는 다른 종류의 서비스와 달리 제 시간에 연속적으로 재생되어야 의미를 갖는 데이터들로 이루어져있으며, 이러한 속성을 얼마나 충족시키느냐에 따라 QoS가 결정된다. 좋은 서비스를 제공하기 위하여 원래서버의 부분서버를 여러 개 두어 서비스 요청을 분산시키는 방법을 많이 사용하고 있다. 본 연구에서는 클라이언트의 요청에 능동적으로 그리고 효과적으로 서비스하도록 웹로그의 문서전송 서비스상의 분포에 따른 사전정보를 가지고 각 부분서버의 부하량을 체크하고, 이후에 발생하는 클라이언트의 요청을 분산시킬 수 있는 동적 알고리즘을 제안한다. 본 동적선택 알고리즘은 QoS가 중요한 대량의 멀티미디어 콘텐츠를 전송함에 있어서 HTTP 반응시간과 문서크기의 변동에 따른 근접척도 공정능력지수를 이용하여 클라이언트 요청을 확률분산시키는 것이다.

### 1. 서 론

초고속 인터넷 망이 보편화되면서 다양한 멀티미디어 콘텐츠 제공 서비스가 생겨나고 있다. 멀티미디어 시스템에서는 서비스의 질(Quality of Service: QoS)을 보장하면서 동시에 얼마나 많은 사용자를 서비스할 수 있는지가 시스템의 성능을 나타내는 가장 중요한 척도이다. 멀티미디어 데이터들은 텍스트나 이미지 데이터와는 그 특성이 다른 오디오나 비디오 데이터를 다루는데 있다. 오디오와 비디오 데이터들은 제 시간에 연속적으로 재생되어야 의미를 갖는 데이터들로서, 이런 특성을 갖는 데이터를 CM(Continuous Media) 개체라고 한다[권진백00].

멀티미디어 서비스의 상업화에 따라 서비스 질이 더욱 중요해졌으며, 좋은 서비스를 필요로 함에 따라 여러 연구들이 진행되고 있는데 크게 분류하여 서버에 여러 개의 버퍼를 두어 데이터 스트림을 버퍼링하는 방법[Deit83, PH96], 하나의 서버에서 모든 클라이언트의 요청을 처리하는 것이 아니라 클라이언트를 가까운 부분서버(Duplicated Server)로 분산시켜 서비스를 제공하도록 하는 방법 등이 있다. 이미 여러 웹 사이트(web site)에서 성능향상과 신뢰도 향상을 위하여 웹 문서를 분산시키고, 부분서버를 통해 서비스하는 방법을 사용하고 있다. 예를 들어 Netscape browser 설치파일을 다운로드 받을 수 있는 Netscape 미러링(mirroring) 사이트는 101개가 존재하는 경우가 해당된다[SBSV98].

그런데 이러한 미러링 사이트들은 클라이언트가 직접 부분서버를 선택하는 방식으로서, 서버의 성능이나 서버의 현재 접속 부하

량 등의 사전정보 없이 서비스를 일단 요청해보고, 서비스가 만족스럽지 않을 경우 사용자가 직접 다른 서버를 선택하여 서비스를 이용하는 방식이다[SBSV98]. 그러나 멀티미디어 서비스의 경우는 서비스가 연속적으로 재생되어야 하기 때문에 클라이언트가 수동으로 서비스를 요청할 서버를 선택하는 것은 바람직하지 못하다. 좋은 서비스를 제공할 서버를 선택할 수 있는 능력이 사용자에게 있거나 혹은 서버간에 트래픽을 조정할 중재자가 있어야 한다. 트래픽 감시를 통해서 자동으로 클라이언트의 요청을 서비스할 서버를 선택해주는 중재자를 말하는 것이다.

따라서 본 연구에서는 멀티미디어 서버가 질 높은 서비스를 제공할 수 있도록 부분서버를 두어 운영됨과 동시에 클라이언트의 요청에 능동적으로 서비스할 수 있도록 하는 동적 알고리즘을 제안한다. 클라이언트의 요청에 능동적으로 서비스할 수 있기 위한 전제 조건은 웹로그를 통한 트래픽 사전정보를 통해 각 부분서버의 부하량을 체크하고, 이후에 발생하는 클라이언트의 요청을 분산시킬 수 있는 능력을 갖는 것이다. 본 논문의 구성은 2절에서 부분서버를 이용한 트래픽 신뢰도 향상에 대한 관련 선행연구와 알고리즘을 요약하고, 3절에서는 웹로그의 문서크기 분산을 이용한 동적선택 공정능력지수(Process Capability Index: PCI) 알고리즘을 제안한다. 마지막 4절에는 결론과 향후 연구를 제시하였다.

### 2. 관련연구

사용자가 접속했을 때 효과적으로 부분서버 하나를 선택하기

위한 기준으로 근접척도(approximity metric)라는 것이 있는데, 이러한 근접척도는 지리적 위치, 네트워크 연결상태, 트래픽 부하 등 다양하게 설정될 수 있다. 먼저 [CC95]은 지리적 거리, 연결 홉(hop)의 개수, 난수발생, 핑 요청의 라운드 트립시간(round trip measurement)의 평균값을 제안하였다. 그러나 [SBSV98]은 홉의 수나 핑 요청의 라운드 트립 시간이 웹 문서의 크기가 아주 작은 경우에는 적당하지만, 근접한 부분서버를 결정하는 일반적으로 좋은 척도가 아님을 보였다. 즉, 요청 후 완전한 문서가 오기까지의 HTTP 요청 응답시간(response time)보다 요청 후 첫 바이트가 오기까지의 HTTP 요청 반응시간(latency time)의 사용이 더 효과적임을 제안했으며, <표1>과 같이 두 척도간의 상관관계도 살펴보았다. 직관적으로 HTTP 반응시간과 HTTP 응답시간의 상관관계는 매우 높을 것이고, HTTP 반응시간과 문서크기의 상관관계는 거의 없을 것이다.

<표 1> 근접척도간의 상관관계

알고리즘 척도	HTTP 응답시간	HTTP 반응시간
핑	0.51	0.76
홉의 개수	0.16	-
HTTP 응답시간	-	0.73

HTTP 반응시간을 이용한 부분서버 동적선택 알고리즘을 정리하면 다음과 같다[SBSV98][VBSS99].

- 병렬(parallel): 모든 부분서버에 매번 요청을 보내 가장 빠른 HTTP 반응시간을 보이는 부분서버 선택. 가장 효과적이거나 동시 사용자에게 같은 부분서버를 선택하게할 위험.
- 확률(probabilistic): 가장 빠른 HTTP 반응시간을 갖는 부분서버가 선택될 확률을 크게 배정. 부분서버의 정보를 갱신할 추가요청이 필요없으나, 최선이 아닌 부분서버에 보내지는 요청회수를 통제할 수 없음.

$$\hat{P}(S_i) = \frac{k}{\bar{t}_i}, \quad \text{단, } \bar{t}_i \text{는 평균반응시간}$$

$$k \text{는 } \sum_{i=1}^n P(S_i) - 1 \text{ 을 만족하는 상수}$$

- 재시도(refresh): 제한된 시간에 HTTP 반응시간내 반응이 오면 그 부분서버를 선택하고, 초과하면 새로운 HTTP 반응시간 값을 얻는다. 추가 요청을 생성하는 단점이 있으며, 리플래시 요청 회수는 HTTP 반응시간값으로 조정할 수 있음.
- 확장된 재시도(extended refresh): 여러 부분서버의 HTTP 반응시간 평균은 유사하나 서비스 질을 의미하는 분산은 크게 다르므로, S-percentile기법과 EWMA, EWMV 모형을 이용하여 HTTP 반응시간값을 추정하는 재시도 알고리즘의 확장[VBSS99].

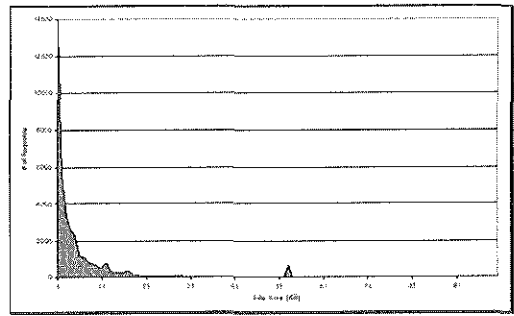
위의 동적선택 알고리즘 외에도 동일확률배분의 고정 알고리즘과 홉의 개수를 이용한 정적 알고리즘은 [CC95]를 참고하면 된다.

이러한 정적·동적선택 알고리즘들은 기존의 서버운영 로그결과 분석이 선행되어야 하므로, 서비스를 처음 시작하는 경우에는 적용하기 적당치 않은 것도 있다. 이런 경우 문서의 인기도인 이용 접속 확률분포에 의하여 부분서버를 선택하거나[Azer95], 문서의 멀티미디어 개체 크기와 문서의 이용접속 확률행렬을 고려하여 부분에서 다운로드 시간을 최소화 하기 위한 비용모형(cost model)

을 제시하였다[LA00]. 즉, 멀티미디어 콘텐츠의 크기가 크거나 스트리밍 데이터인 경우는 HTTP 반응시간같은 근접척도를 이용한 이러한 알고리즘 보다는 서비스될 멀티미디어 콘텐츠 문서의 크기와 해당 부분서버가 서비스를 제공하고 있는 밴드위스가 더 유용하다. 따라서 다음 절에서는 문서의 양에 따른 멀티미디어 서버의 QoS를 높이기 위하여, 웹로그 자료에 근거한 공정능력지수를 근접척도로 하는 PCI 알고리즘을 제안하겠다.

### 3. PCI(Process Capability Index) 알고리즘

사용자의 서비스 이용반응에 의한 웹로그 분포에 의하면 문서용량에 따른 접속이용 확률분포는 [그림 1]과 같이 지수분포를 따르고 있다[SBSV98, Azer95]. 즉 문서의 양이 적은 경우의 이용 확률은 매우 크고 문서의 양이 커질수록 이용확률은 지수적으로 감소하는 것이다.



[그림 1] 문서 크기와 이용회수 간의 분포[SBSV98]  
x축: 문서크기, y축: 이용회수

이러한 콘텐츠들을 똑같이 갖고 있는 각 부분 서버들에게 특정 문서의 요청이 이루어졌을 때, 2절의 알고리즘에 의해서 해당 부분서버가 선택될 것이다. 그러나 서비스하는 문서크기가 큰 경우에 이를 고려하지 않고 해당 부분서버의 HTTP 반응시간은 짧았던 자료에 근거해서 선택이 이루어진다면, 동시접속자수와 같은 네트워크 부하 외에도 문서크기에 따른 서버의 CPU와 메모리의 사용 증가요인이 고려되어 허용 밴드위스를 초과하여 현재 서비스받고 있는 사용자에게도 장애가 발생할 수 있는 것이다.

실제로 작은 문서의 경우는 HTTP 반응시간이 밴드위스보다 우세하지만, 큰 멀티미디어 문서의 경우는 밴드위스가 HTTP 반응시간보다 더 좋다. 이렇게 멀티미디어 서버의 QoS는 HTTP 반응시간보다는 문서의 크기와 동시접속자 수에 크게 의존되므로, 부분서버를 선택하기 위한 알고리즘에 이 두가지 근접척도를 이용하고자 한다.

전송하고자 하는 멀티미디어 문서의 크기 변인 B에 대한 척도는 다음과 같은 공정능력지수  $C_p$ 를 이용하여, 다양한 접속자에 따라 전송되고 있는 멀티미디어 문서크기의 분산변동을 고려할 수 있도록 한다[Han98].

$$C_p = \frac{USL - LSL}{6\sigma}$$

단, USL, LSL: 최대최소허용규격,  $B \sim$  지수분포( $\lambda$ )

이때 멀티미디어 서버의 경우 특성변수는 지수확률분포를 갖는

문서의 크기가 될 것이고, 최대최소허용규격은 현재 전송문서의 최대최소값으로  $USL$ 은 웹프로그램부터  $\lambda$ 를 추정해서 얻을 수 있다.

$$\begin{aligned} LSL &= 0 \\ USL &= a, \end{aligned}$$

단,  $a \leq \text{bandwidth}$  을 만족하는 상수

위의 공정능력지수  $C_p$ 의 추정량은 스텔링의 공식에 의하여  $n$  ( $\geq 30$ )이 충분히 클 때 불편추정량으로 사용할 수 있다.

$$\hat{C}_p = \frac{\hat{\sigma}}{6S}$$

단,  $S$ :  $B$ 의 표본 표준편차

이  $C_p$ 의 추정량값이 크면 해당 부분서버의 웹로그 값으로부터 얻을 수 있는 접속자간 분산  $\sigma$ 값이 작아 문서크기들의 변동이 작으므로 해당 부분이 선택될 확률을 HTTP 반응시간을 이용하고, 반대로 추정된  $C_p$ 값이 작으면 해당 부분서버의 접속자간 분산  $\sigma$ 값이 커서 문서크기들의 변동이 심하므로 QoS를 좋게 하기 위하여 해당 부분서버가 선택될 확률을 작게 하는 것이다. 즉,  $C_p$ 이 PCI를 이용한 알고리즘을 정리하면 다음과 같다.

- 부분서버  $i$ 의 웹로그를 통하여 단위기간(예를 들어 하루) 당 서비스한 문서의 표본 표준편차  $S_{bi}$ 를 계산한다.

- $\hat{C}_{pi} = \frac{\hat{\sigma}}{6S_{bi}}$  를 계산한다.

- 부분서버  $i$ 가 선택될 다음 확률을 계산한다.

$$\hat{C}_{pi} \geq 1 \text{ 이면 } P(S_i) = \frac{k}{l_i},$$

$$\hat{C}_{pi} < 1 \text{ 이면 } P(S_i) = k \frac{C_{pi}}{l_i},$$

단,  $k$ 는  $\sum_{i=1}^{n_1} P(S_i) + \sum_{i=1}^{n_2} P(S_i) = 1$  을 만족하는 상수.

$n_1 + n_2 = n$  은 전체 부분 서버 개수,  $l_i = \frac{1}{m} \sum_{j=1}^m l_{ij}$ .

- 부분서버  $i$ 에 현재 서비스 받고 있는 동시 접속자 수가 허용치 미만이면, 부분서버 선택확률 추정값에 의해 서버를 선택한다.

- 부분서버  $i$ 에 현재 서비스 받고 있는 동시 접속자 수가 허용치 이상이면, 다른 부분서버를 선택한다.

위의 공정능력지수  $C_p$  추정량의 분포는 정규분포 가정하에서조차 매우 복잡하므로 [Han98], 지수분포와 같은 분포에 관계없는 추정량으로 붓스트랩 알고리즘을 적용할 수도 있을 것이다 [HCL00]. 또한 [HCL00]에서의 다양한 붓스트랩 퍼센타일 기법을 이용하여 지수분포의 모수를 추정할 수도 있다.

#### 4. 결 론

부분서버를 이용하여 멀티미디어 콘텐츠 서비스를 제공하는 경우에 사용하는 다양한 알고리즘을 살펴보았다. 멀티미디어 문서의 이용분포를 살펴보면 크기가 작은 경우가 빈번히 사용되는 지수분포를 갖는 것에 의해, 작은 문서의 경우는 HTTP 반응시간을 이용

한 부분서버를 동적으로 선택하는 알고리즘이 제안되었다.

그러나 큰 멀티미디어 문서의 경우는 밴드위스가 HTTP 반응시간보다 더 중요하므로, 문서의 크기변동을 고려한 근접척도를 이용한 부분서버 동적선택 알고리즘을 제안하였다. HTTP 반응시간이 아닌 문서크기와 밴드위스에 의한 QoS를 높일수 있는 공정능력지수  $PCI$   $C_p$ 를 이용한 알고리즘을 제안하였다. 즉 부분에서 서비스하는 문서크기의 변동의 폭이 큰 부분서버에는 작은 선택확률을 부여하고, 문서의 변동폭이 작은 경우 HTTP 반응시간을 이용한 확률을 부여하는 것이다.

향후 연구로는 이러한 모수 추정량이 편의되어 있으므로, 불편추정량과 붓스트랩 기법을 이용한 추정량을 제안함으로써 분포에 관계없는 근접척도로서 그 응용범위를 확대하고자 한다.

#### 참고문헌

[권진백00] 권진백, 염현영, 이경오, 실시간 멀티미디어 시스템에서의 캐싱을 위한 동적 버퍼 할당 기법, 정보과학회논문지 시스템 및 이론 27(4), pages 420-429, 2000년 4월

[Azer95] Azer Bestavros, Demand-based Document Dissemination to Reduce Traffic and Balance Load, Proceedings of SPDP, Oct 1995

[CC95] M.E. Crovella, R.L. Carter, Dynamic server Selection in the Internet, proceedings of the Third IEEE workshop on the Architecture and Implementation of High Performance Communication Subsystems, June 1995

[Deit83] Harvey M. Deital, An Introduction to Operating Systems, Addison Wesley, 1983

[Han98] Han J.H., Various Process Indices based on Bootstrap Theory and Applications, Chungbuk National University, Doctorial Thesis, 1998

[HCL00] Han J.H., Cho J.J., Leem C.S. Bootstrap Confidence Limits for Wright's  $C_s$ , Communications in Statistics: Theory and Methods, 2000., Volume 29, Number 3., pp. 485-505

[LL00] Thanasis Loukopoulous and Ishfaq Ahmad, Replicating the Contents of a WWW Multimedia Repository to Minimize Download Time, Proceedings of the 14th International Parallel and Distributed Processing Symposium, 2000

[PH96] David A. Patterson and John L. Hennessy, Computer Architecture a Quantitative Approach, Morgan Kaufmann Publisher, 1996

[SBSV98] Mehmet Sayal, Yuri Breitbart, Scheuermann, Radek Vingralek, Selection Algorithms for Replicated Web servers, Proceedings of the Workshop on Internet Server Performance, 1998

[VBSS99] Radek Vingralek, Yuri Breitbart, Mehmet Sayal, Peter Scheuermann, Web++: A System for Fast and Reliable Web Service, Proceedings of the USENIX Annual Technical Conference, 1999