한국시뮬레이션학회 '99춘계학술대회 논문집 1999.4.17 동의대학교

WAN 환경에서 멀티미디어 서비스를 위한 효율적인 자원 배치

조성봉, 이효일, 김종헌
연세대학교 전산학과 대학원
강원도 원주시 흥업면 매지리 234
전화: +371-760-2289
email : jhkim@dragon.yonsei.ac.kr, chobong@magics.yonsei.ac.kr

요 약 문

초고속 정보통신망의 구축과 개인용 컴퓨터의 보급이 늘어나면서 인터넷을 비롯한 다양한 형태의 정보 서비스들이 크게 증가하고 있다. 그러한 서비스를 위한 정보들은 대부분 대규모 데이터 전송을 요구하는 음성이나 동영상과 같은 멀티미디어 정보들로 이루어지며, 그들은 전자도서관이나 거대한 정보 저장소에 디지털로 인코딩 되어진 형태로 저장된다. 그런데 사용자들이 원하는 정보가 하나의 서버에만 저장되는 중앙집중적인 환경에서는 그 서버가 모든 서비스 요구들을 처리해야 하기 때문에 병목 현상이 발생할 수 있고, 네트워크의 과부하로 인하여 양질의 서비스를 제공하기 어려워진다. 본 연구는 멀티미디어 정보 중에서도 가장 많은 전송량이 요구되는 VOD 서비스를 효율적으로 지원할 수 있는 환경 구축 방안을 찾는 것을 목표로 한다. 이를 위하여 비디오 정보들을 다수의 서버들에 분산 저장하고, 서버의 위치와 정보의 배치 방법에 따른 서비스 시간과 throughput을 시뮬레이션으로 분석하고, 각각의 경우에 대한 비용을 추정하여 가격대성능비 측면에서 가장 효율적인 방법을 제안하였다.

1. 서론

초고속 정보통신망을 비롯한 정보 인프라의 구축이 확대되면서 다양한 종류의 정보 서비스들이 활성화되고 있다. 그들 중에서도 멀티미디어-온-디멘드(Multimedia-On-Demand) 시스템은 정보 제공자(information provider)가 통신망에 접속된 사용자들이 원하는 멀티미디어 정보를 원하는 시간에 실시간으로 전송해주어야 한다. 그러나 사용자 요구들이 급속히 증가하거나 특정 정보에 집중되는 경우에는 서버 혹은 통신망에 병목 현상이 발생하여 요구들에 대한 서비스 시간이 길어지고 단위 시간당 제공할 수 있는 서비스 수도 감소하게 된다. 이러한 문제들은 모든 정보들을 하나의 서버에 모두 저장하는 일반적인 중앙집중식 환경에서 더욱 심각해진다. 따라서 정보들을 여러 개의 서버들에 분산 저장하는 방법들이 연구되었다 [1,2,3,4]. Bisdikian and Patel [2]은 트리 구조(tree

structure)의 계층적 분산 네트워크에서 각 노드에 접속된 서버들에 빈번히 액세스되는 정보일수록 사용자측과 가까운 하위층 노드들에 저장하고, 상위층 노드들에는 하위층에 저장된 정보들을 모두 포함한 슈퍼세트 개념의 분산 저장 방식을 제안하였다. 그러나 이 방법은 대규모의 저장 용량을 필요로 하는 단점을 가지고 있다. 또한 캐싱(caching) 방식을 이용하는 시스템 구조에 대한 연구들도 진행되어 왔다 [3,4,5]. 이 방법들은 캐쉬 관리를 위한 오버헤드가 부담이 되고 있다.

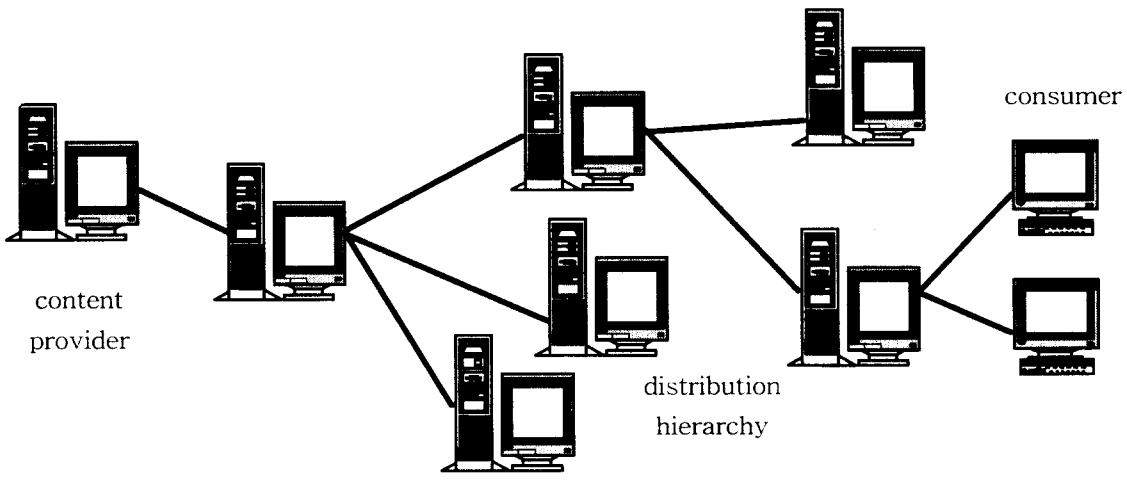
본 연구에서는 앞에서 살펴보았던 유사 연구들에서도 고려한 계층적 분산 네트워크(hierarchical distributed network)에서 저장 용량에 따른 비용과 서비스의 질을 모두 고려한 효율적인 자원 배치 방법을 제안하는 것을 목표로 한다. 이를 위하여 멀티미디어 정보들을 인기도(액세스 빈도)에 따라 중복 혹은 분산 저장하는 방법을 다양하게 사용하여 각각에 대한 가격대성능비를 비교하고자 한다.

2. 시스템 구성

2.1 계층적 분산 네트워크 모델

분산 멀티미디어-온-디멘드 시스템을 구성하는 기본적인 접근 방식으로는 서버의 비용-최적화 접근 방식과 네트워크의 비용-최적화 접근 방식이 있다. 최저의 서버 비용을 갖는 방법은 모든 가능한 비디오 정보들을 하나의 중앙 서버(central server)에 위치시키는 것이다. 그러나 이러한 구조는 디스크 대역폭의 한계로 인한 극심한 병목 현상을 야기하기 때문에 다수의 사본들(copies)을 배치해야 하는 문제가 발생한다. 최저의 네트워크 비용을 갖는 방법은 모든 비디오 정보들을 사용자의 종단 노드에 저장하는 것이다. 하지만 이러한 구조는 저장 공간의 제약으로 인하여 실용적이지 못하다.[1]

<그림 1>의 트리 구조는 다양한 사용자들을 중간 노드들을 통하여 중앙 서버에 연결시켜준다. 만약 이러한 중간 노드들이 정보를 직접 저장하거나 캐싱할 수 있는 능력을 가진다면, 이러한 구조를 계층적 분산 네트워크 구조라고 할 수 있다.[1,2,3,4]



<그림 1> 계층적 비디오 서버의 구조

2.2 시뮬레이션 모델

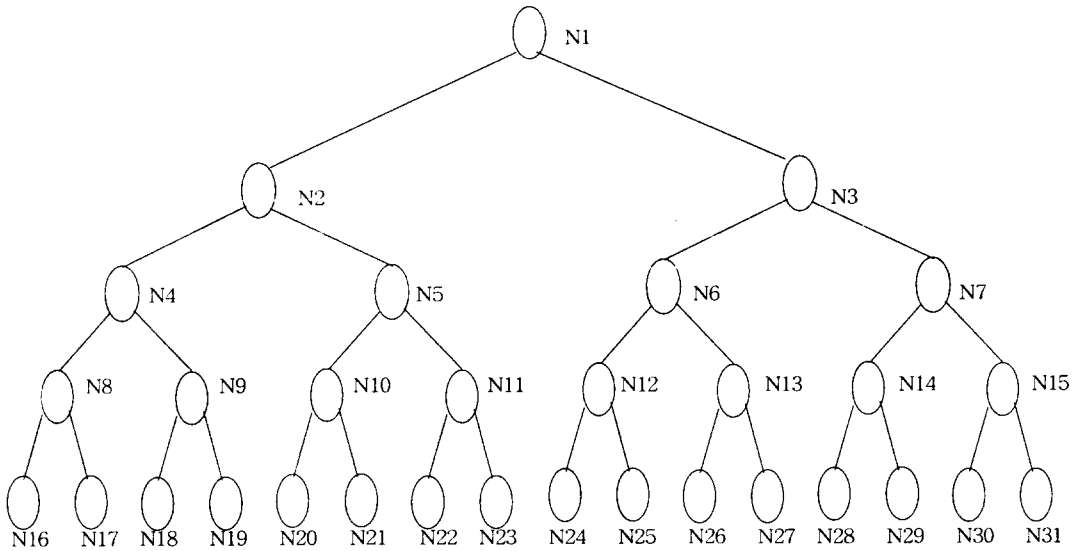
시뮬레이션에서 사용한 네트워크 토포롤지는 2진 트리(binary tree) 구조로서, 트리의 각 노드들은 스위칭 노드들이다. 각 스위칭 노드(N_i)에는 비디오 서버(S_i)가 위치하고, 각 서버는 자신의 자 노드(children nodes)로부터 들어오는 요구에 대하여 정보들을 서비스한다. 사용자 터미널들은 트리의 최하위 노드들에만 연결되어 있고, 트리의 중간 노드들은 사용자들과 직접 연결되지 않는 것으로 가정하였다. 트리는 4-레벨로 구성되어 있으며, 각 레벨의 노드들을 연결하는 네트워크 용량(속도)은 다를 수 있지만, 같은 레벨의 링크들은 같은 용량을 가진다. 각 서버에서 요구에 따른 멀티미디어 정보들을 분배하는 용량에는 제한이 없는 것으로 하였다. 즉, 어느 한 서버에 저장된 정보에 대한 액세스 요구들이 동시에 도착하여도 지연없이 즉시 응답할 수 있는 것으로 가정하였다.[1.5]

그리고 각 서버는 캐싱 기능은 가지지 않은 것으로 하였다.

시뮬레이터는 시뮬레이션 전용언어인 AweSim을 사용하여 이산사건 시뮬레이터(discrete-event simulator)로 구현하였다.[6] 시뮬레이터에서 각 노드는 ATM 스위치로 가정하였으며, 스위치간 링크의 전송 속도는 변화시킬 수 있도록 하였다. 이 연구에서는 자원(정보 및 저장장치)의 배치 방법에 대한 가격대성능비를 분석하는 것이므로 저장장치에서의 액세스 시간은 시뮬레이션 시간에 반영하지 않았다.

2.3 사용자 요구의 분포

최하단 노드들에 접속된 사용자들의 요구 패턴(원하는 정보, 요구간 시간간격 등)은 동일한 것으로 가정하였다. 그리고 각 정보에 대한 요구 패턴은 그 정보의 인기도에 의해 결정되는 것으로 하였는데, 이러한 액세스 빈도는 일반적으로 널리 사용되는 Zipf의 법칙(Zipf's Law)을 따른다고 가정하였



<그림 2> 트리 토폴로지

다.[1,3,7] 이 법칙에 따르면, i 번째 정보를 M_i라고 할 때 각 정보가 요구될 확률 π_i는 식(1)과 같다.

$$\pi_i = \Pr(M_i)$$

$$\text{그리고 } \sum_{i=1}^{M} = 1, \qquad (1)$$

$$(\text{단, } 1 > \pi_1 \geq \pi_2 \geq \ldots\ldots \geq \pi_M > 0)$$

즉, 번호 i가 낮을수록 요구 빈도가 더 높다.

2.4 멀티미디어 정보의 저장 위치

기본적으로 본 연구에서 사용한 자원 배치 및 정보 저장 방법은 [2]에 근거를 두고 있다. 즉, 루트(root) 노드에는 모든 정보들이 저장되어 있지만, 빈번하게 요구되는 정보들은 사용자들에게 가능한 가까이 위치하도록 중복 저장한다. 그러나 사용자에게 가까이 위치시킬수록 더욱 많은 양의 사본들이 필요하게 된다. 만약 한 개의 비디오 정보가 레벨 k에 놓여지게 한다면, r-ary 트리 구조에서 r^k 개의 사본의 영화들이 시스템에 존재해야 한다. 본 연구에서는 다음과 같은 다섯 가지의 자원 배치 방법들을 고려하였다. 단, 여기서 정보의 종류는 16개(M=16)인 것으로 한다.

(1) DB0 : 모든 정보들을 루트 노드에 저장하는 방법 (중앙집중식).

(2) DB1 : [1]에서 사용한 방법으로서, 가장 빈번히 요구되는 M_1은 최하위(레벨 4)의 노드들에 위치시키고, M_2는 레벨 3에, M_3와 M_4는 레벨 2에, M_5 M_6, M_7 및 M_8은 레벨 1에, 그리고 $M_9 \sim M_{16}$ 은 루트에 위치시키는 비대칭적 방법.

(3) DB2 : 모든 정보들을 최하단 노드들에 하나씩 분산 저장하는 방법으로서, 사용자 요구량이 대칭을 유지하게 된다. 즉, 그림 2에서 N16에 M_1, N31에 M_2, N17에 M_3, N30에 M_4, ... N23에 M_{15}, N24에 M_{16}을 각각 저장한다.

(4) DB3 : 모든 노드들에 하나씩의 정보를 배치하는 대칭적 방법으로서, 모든 종단 노드들에 M_1의 사본들을 위치시키고, $M_2 \sim M_{16}$은 중간 노드들에 분산하였다. 이 때 분산시키는 알고리즘은 식(2)에 따른다.

$$\text{MAX}(\pi_i \mid i_{\min} \leq i \leq i_{\max}), \qquad (2)$$

M_2는 루트 노드에 배치하고, 나머지 정보들은 요구의 대칭성을 유지하기 위하여 DB2에서와 같은 순서로 각 레벨의 노드들에 배치한다 (즉, M_3는 N2, M_4는 N3, M_5는 N4, M_6는 N7, M_7는 N5, M_8는 N6, ... M_{15}는 N11, M_{16}는 N12에 배치).

(5) DB4 : 모든 노드들에 두 개씩의 정보들을 배치하는 대칭적 시스템으로서, 최하단의 모든 노드들에 M_1과 M_2를 배치하고 레벨 3의 모든 노드들에 M_3와 M_4를 배치하며, M_5, M_6은 대칭성을 유지하기 위하여 레벨 1에 배치하고, 나머지 정보들은 DB3와 같은 방법으로 위치시키는 방법.

3. 작업부하

사용자들이 요구하는 정보의 번호(ID)는

Zipf의 법칙에 따른 확률에 근거하여 결정하였다.[9] Zipf는 영어에서 단어의 사용 빈도를 표현하는 것으로서, 식 (3)과 같다.

$$f_i \approx i \times f_i, \qquad (3)$$

여기서 f_i는 i 번째로 빈번히 사용된 단어를 나타낸다. Zipfian 확률 분포는 n 개의 항목들로 이루어진 배열이 된다: $a_1, a_2, \ldots, a_n$. 단, 여기서 a의 순서는 액세스 빈도를 나타낸다. a_k가 액세스될 확률 P_k는 다음과 같아진다.

$P_k = P_1/k$, $\sum_{k=1}^{n} P_k = 1$이므로,

$$P_1 \sum_{k=1}^{n} 1/k = 1, \quad H_n = \sum_{k=1}^{n} 1/k$$

$$P_1 H_n = 1, \; P_1 = 1/H_n$$

결과적으로, Zipfian 확률 분포는

$P_k = 1/(k \times H_n)$ for k = 1, 2, ..., n

여기서 $H_n = \sum_{k=1}^{n} 1/k$

$$= 0.77216 + \ln(n) + 1/2n - 1/12n^2 + 1/120n^4 - 1/252n^6 \cdots$$

$$H_n = \sum_{k=1}^{n} 1/k, \quad P_k = 1/(k \times H_n)$$

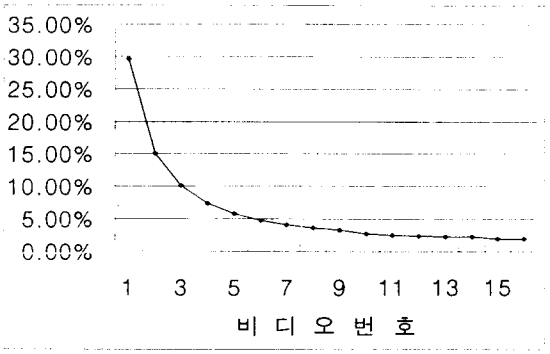
$S_J = \sum_{k=1}^{J} P_k$, 그리고 $S_0 = 0, S_n = 1$

If S_{k-1}<=rand()< S_k then return k (4)

일반적으로 광대역망(WAN) 환경에서의 작업 부하 특성은 지수분포를 따르는 것으로 알려져 있다.[7,8] 따라서 본 연구에서도 멀티미디어 정보에 대한 요구간 간격을 지수분포로 가정하였고, mean value는 100 μsec를 기준으로 하여 변경할 수 있도록 하였다.

요구하는 정보의 크기는 NASA의 Web 트레이스[8]에 근거한 정보 크기인 439,151 Bytes로 하였으며, 모든 정보들의 크기를 정보의 종류에 관계없이 동일한 것으로 가정하였다.

이에 근거하여 사용자들이 요구하는 비디오 정보의 분포를 분석해본 결과, <그림 3>에서 보는 바와 같이 M_1, M_2, M_3 및 M_4에 대한 요구들이 전체의 62.17%를 차지하는 것으로 나타났다. 따라서 가장 빈번히 요구되는 M_1, M_2, M_3, 및 M_4를 사용자 가까이 위치시킬수록 서비스 시간을 효과적으로 줄일 수 있을 것이며, 서버에 필요한 저장공간의 크기도 결정될 것이다. 다시 말하면, 요구빈도가 낮은 정보들은 여러 서버들에 효과적으로 분산시키고, 빈도가 높은 정보들은 중복 저장하는 것이 성능 향상에 도움이 될 것이다.

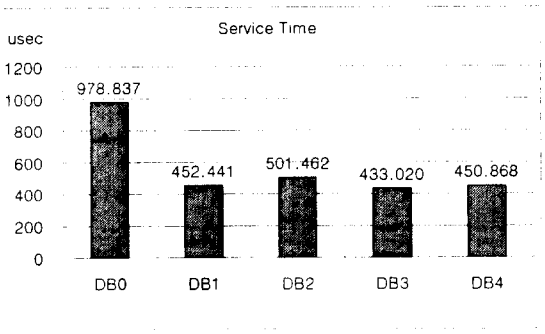


<그림 3 > Zipf's Law에 의한 요구분포
(전체요구수 : 32016, M =16)

4. 시뮬레이션 결과

4.1 평균 서비스 시간

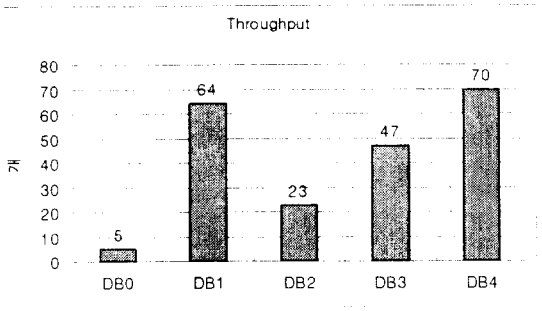
사용자가 정보를 요구한 시간부터 응답을 받은 순간까지의 시간을 서비스 시간이라고 하며, 이 시간이 사실상 서비스의 질을 나타낸다. 네트워크상에서 처리된 모든 서비스들의 평균 응답 시간인 평균 서비스 시간을 각 배치 방법들에 대하여 분석한 결과는 <그림 3>과 같다. 결과에 따르면, 모든 정보들을 루트 노드에 배치한 DB0의 경우에 평균 서비스 시간이 978.837μs로 가장 길고, 모든 서버에 하나의 비디오 정보씩 배치한 DB3가 433.020μs로 가장 짧은 서비스 시간을 보였다. 그러나 다른 방법들도 큰 차이는 없는 것으로 나타났다.



<그림 4 > 각 시뮬레이션 모델에서의 평균 서비스 시간(s)

4.2 서비스된 요구의 수

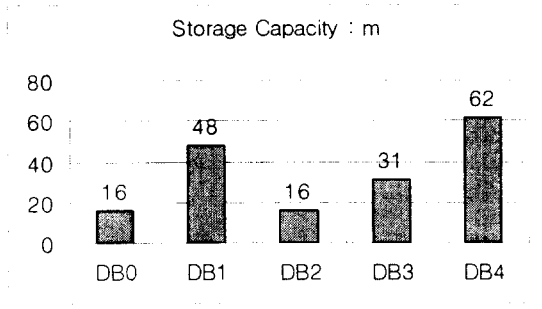
서비스가 완료된 요구들의 수, 즉 시스템의 throughput은 <그림 4>에서 보는 바와 같이, 모든 서버들에 두 개씩의 정보들을 배치하는 DB4가 70개로 가장 많았다. 그리고 가장 적게 처리한 방법은 모든 비디오 정보들을 루트 노드에 위치한 DB0이었다. 평균 서비스 시간에서는 크게 차이를 보이지 않았던 DB2와 DB3는 DB4와 많은 차이를 보였다



<그림 5> 각 시뮬레이션 모델에서 서비스된 개수(Throughput : t)

4.3 서버의 저장 공간

각 배치 방법에서 필요한 서버의 저장 용량(storage capacity) m은 <그림 5>와 같다. 즉, DB0과 DB2의 경우는 m=16이며, DB4가 m=62로서, 가장 큰 저장 용량을 필요로 한다.

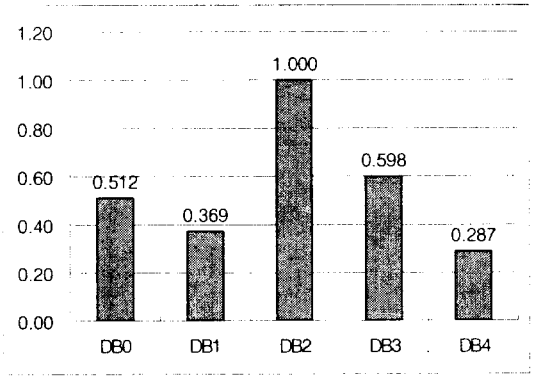


<그림 6> 각 시뮬레이션 모델에서 필요한 서버의 저장공간(Storage Space: m)

4.4 가격대성능비의 분석

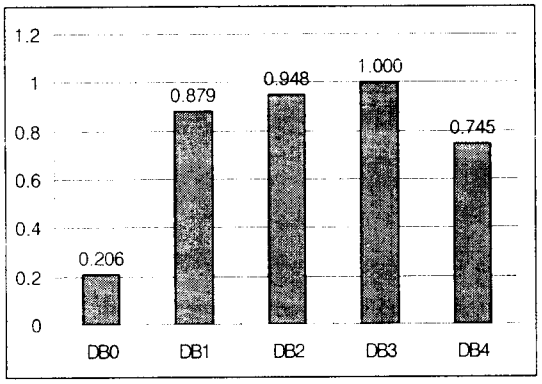
본 연구의 목적은 사용자들의 요구를 신속히 서비스해주는 것과 시스템 구성 비용을 최소화시키는 것이다. 즉, 그 두 가지를 적절히 조정(tradeoff)할 수 있는 최적의 배치 방법을 찾는 것이다. 따라서 지금까지 분석한 결과들을 바탕으로 하여 가격대성능비가 가장 높은 배치 방식을 계산한 결과를 보면 다음과 같다.

먼저 서비스의 질을 나타내는 평균 서비스 시간과 비용을 나타내는 저장 용량간의 비율을 모든 배치 방법들에 대하여 비교한 결과는 <그림 7>와 같다. 이 비교에서는 서비스 시간과 저장 용량의 중요도를 1 : 1로 정하였다. 결과에 따르면, 정보들이 최하단 노드들에만 분산 저장한 DB2가 가장 높은 값을 보였다.



<그림 7> 각 시뮬레이션 모델에서의 평균 서비스 시간과 저장공간의 비율

DB2의 throughput은 <그림 4>에서 본 바와 같이 가장 낮았으나, 비용이 현저히 낮기 때문에 그와 같은 결과를 가져온 것이다. 그러나 만약 비용보다는 성능을 중요시하는 경우라면 비교 결과는 달라질 것이다.



<그림 8> 각 시뮬레이션 모델에서 throughput과 저장용량의 비율

시스템 성능을 나타내는 throughput과 저장 용량간의 비율을 분석한 결과는 <그림 8>과 같다. 이 경우에는 액세스 빈도가 가장 높은 정보들을 최하단에 저장하고 나머지 정보들을 중간 노드들에 대칭적으로 분산 저장한 DB3 방법이 상대적으로 가장 비율이 높은 것으로 나타났다. 그러나 DB1과 DB2도 비교적 높은 비율을 보여주었다.

5. 결론

본 연구에서는 2진 트리 구조를 가진 네트워크에서 사용자들이 요구하는 정보들을 다수의 서버들에 적절히 분산 배치함으로써 성능대가격비를 최적화할 수 있는 방법을 분석하였다. 그 비교를 위하여 다른 연구들에서 사용한 배치 방법인 DB1 외에도 다양한 방법들을 고려하였다. 시뮬레이션 결과에 따르면, 서비스 질을 나타내는 평균 서비스 시간과 저장 용량의 비율에서는 DB1 보다 DB2가 더 좋은 비율을 보여 주었고, 시스템 성능을 나타내는 throughput과 저장용량의 비율에서는 DB3가 DB1보다 더 좋은 비율을 나타냈다. 이것은 멀티미디어 정보를 제공하는 측면에서 평균 서비스 시간을 서버의 비용보다 중요시할 때나 throughput을 저장 용량보다 중요시 하는 경우에 선택할 수 있는 시스템의 모형을 제공하는 것이다. 이 것은 네트워크망의 링크를 추가하여 서비스를 개선하는 방법[4]과 비교하여 볼 때, 추가적인 비용부담이 없기 때문에 기존 네트워크의 망의 형태에서 멀티미디어 서비스를 위한 개선 방안이 될 수 있다.

참고문헌

[1] Carsten Griwodz, Michael Bar, Lars C. Wolf, "Long-term Movie Popularity Models in Video-on-Demand Systems or The Life of an-Demand Movie", In Proceeding of the Conf on Multimedia 97, pp 349.

[2] C. Bisdikian and B. V. Patel. Cost-Based Program Allocation for Distributed Multimedia-on-Demand Systems. IEEE Multimedia, pp.62-72, Fall 1996.

[3] C. C. Bisdikian and B. V. Patel, "Issues on Movie Allocation in Distributed Video-on-Demand System s", Proc. IEEE Int'l Conf. on Communications, IEEE Communica -tions Society, New York, 1995, pp.250-255.

[4] Cyrus Shahabi, Mohammad H. Alshayyeji, and Shimeng Wang, A Redundant Hierarchical Structure for a Distributed Continuous Media Server, Proceeding of the European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS), 1997

[5] D.J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L, A. Rowe. Multimedia Storage Servers: A Tutorial. IEEE Computer, May 1995.-357.

[6] Gerard DAMM, Gerard BABONNEAU, Andre-Luc BEYLOT and Monique BECKER, Performance Evaluation of a Multimedia Server for ATM Networks, Journal of Parallel and Distributed Computing, pp.41-50, 1994.

[7] James E. Pitkow, Summary of WWW characterizations, Chair, HTTP-NG WCG, Xerox Palo Alto research Center. http://decweb.ethz.ch/WWW7/1877/com1877.htm, 1997.

[8] M. F. Arlitt and C.L. Williamson. Web server workload characterization : The search for invariants. In Proceeding of the Sigmetrics Conference on Measurement and Modeling of Computer Systems, pp 126-137, May 1996.

[9] Zipf's Law, http://www.elisdsu.edu/courses/fall96/cs660/notes/splayPref/splayperf.html