

이동 컴퓨팅 환경에서 데이터 방송을 위한 클라이언트 요구의 고려

신동천

중앙대학교 정보시스템학과

dcshin@cau.ac.kr

Considering the Needs of Clients for Data Broadcasting in Mobile Computing Environments

Dongcheon Shin

Dept. of Information Systems, Chung-Ang University

요약

이동 컴퓨팅 환경에서 다수의 클라이언트에게 효과적으로 데이터를 방송하는 것은 시스템의 성능에 커다란 영향을 미친다. 특히 변화하는 클라이언트의 요구사항을 고려하여 데이터를 방송하는 것은 시스템의 성능 향상을 기대할 수 있다. 본 논문에서는 방송되는 데이터가 클라이언트에게 미치는 효과를 고려하는 방송 알고리즘을 제안한다. 제안된 알고리즘은 클라이언트로부터 수신하는 비트 벡터를 이용한다.

1. 서론

이동 컴퓨팅 환경에서 방송 기반의 정보 시스템은 통신 기술의 발달과 새로운 응용들의 출현에 힘입어 관심이 고조되고 있다. 기존의 일대일 형식의 데이터 전송과 달리 방송 기법은 다수의 클라이언트에게 효과적으로 데이터를 전송할 수 있는 방법이다[1]. 특히 공통의 데이터에 관심을 갖는 클라이언트들이 많은 경우에 효과적이다.

무선통신은 이동 컴퓨팅 환경의 가장 큰 특징중의 하나이다. 따라서 무선통신망은 기존의 유선통신망과 달리 여러 가지 환경적인 제약이 있다[2,3]. 그 중에서 상대적으로 낮은 대역폭을 갖는다는 점은 데이터 전송에 가장 커다란 장애중의 하나라고 할 수 있다. 뿐만 아니라, 무선과 이동성이라는 제한은 클라이언트와 서버사이의 비대칭성(asymmetry)을 초래한다[4]. 즉, 서버에서 클라이언트로의 데이터 전송이 많고 따라서 서버에서 클라이언트로의 데이터 전송 채널(upstream channel)의 대역폭이 그 반대의 데이터 전송 채널(downstream channel)보다 크게 된다. 또한, 이동성으로 인하여 다수의 클라이언트가 셀 안에 존재하게 된다. 따라서, 기존의 데이터 전송 방식과 다른 새로운 전송 방식의 도입이 필요하며 아울러 낮은 대역폭을 효율적이고도 효과적으로 활용하는 데이터 방송 기법이 필요하게 된다.

서버가 방송할 데이터를 선정하는 방식을 푸쉬(push)방식, 풀(pull)방식, 혼합(hybrid) 방식으로 나눌 수 있다[5,6]. 푸쉬 기반의 방송은 클라이언트의 데이터 요구에 대한 패턴을 정확하게 파악할 수 없으므로 클라이언트의

데이터 요구패턴 변화가 적을수록 효과적일 수 있으나 클라이언트 요구패턴이 동적으로 변화하는 환경에서는 방송 효율성에 한계가 있을 수밖에 없다. 따라서 풀 기반 혹은 혼합 기반의 방송이 대안이 될 수 있다. 즉, 클라이언트의 데이터에 대한 요청 정보를 기반으로 서버가 데이터를 방송하므로 다양한 클라이언트의 데이터 요구사항을 반영할 수 있다는 장점을 상대적으로 갖게 된다. 따라서 이러한 방송 방식에서는 사용자의 요구사항을 반영하는 방법이 시스템의 성능에 크게 영향을 미친다고 할 수 있다.

지금까지 제안된 대부분의 방송 알고리즘은 클라이언트가 필요한 데이터를 단순히 요청 메시지를 통하여 직접 서버에게 요청하면 서버가 이를 반영한 방송 프로그램을 생성하는 형태라고 할 수 있다. [7]에서는 클라이언트의 동적인 요구변화를 반영하기 위하여 비트 벡터(bit vector)를 사용하여 다중 디스크 방송 기법[4,5]을 확장한 방송 알고리즘을 제안하였다.

[7]에서 제안한 방송 알고리즘은 최근에 요청된 데이터와 과거에 요청된 데이터를 차별화 하지 않아 과거에 클라이언트에 필요했던 데이터들이 여전히 방송될 데이터에 포함되는 비효율성을 초래할 수 있게 된다. 뿐만 아니라, 클라이언트의 데이터 요청 메시지에 비트 벡터를 피기백킹(piggybacking)하므로 데이터 요청 메시지를 보내지 않는 클라이언트의 요구변화를 파악하여 방송에 반영할 수 없게 되어 있다.

본 논문에서는 모든 클라이언트의 요구패턴을 반영함은

물론 클라이언트 요구의 최신성에 비중을 두어 방송된 데이터 중에서 방송 효과가 큰 데이터를 지속적으로 방송하기 위한 방안을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 도입하여 활용하고자 하는 다중 디스크 방송 기법의 소개와 함께 비트 벡터를 활용하는 방송 알고리즘을 제안하고 3장에서는 결론을 맺는다.

2. 방송 알고리즘

2.1 다중 디스크 방송 기법

다중 방송 디스크 기법[4,5]은 방송할 데이터를 여러 개의 범주로 나누어 각 범주간에 방송 빈도수를 차별화 하여 주기적으로 데이터를 방송하는 기법이다. 따라서, 방송 디스크 방식에서는 데이터에 대한 클라이언트의 요구 확률에 따라 방송할 데이터에 대한 그룹화가 필요하다. 각 그룹은 별개의 디스크로 간주되며 디스크의 회전 속도 즉, 그룹간의 방송 빈도수를 결정하는 일이 필요하다. 디스크 수와 빈도수의 비율이 성능에 영향을 미칠 수 있으나 이 문제는 최적화 문제로 남겨 두고 있다. 방송 알고리즘을 요약하면 다음과 같다.

1. 방송할 데이터 페이지들을 접근확률이 높은 것에서 낮은 것으로 정렬한다.
2. 접근 확률이 유사한 페이지들끼리 분할한다. N개로 분할한 경우 디스크를 각각 D_1, D_2, \dots, D_n 이라고 한다.
3. 각 디스크 (D_i)의 상대적인 빈도수(f_i)를 결정한다. 빈도수는 정수로 한다.
4. 각 디스크의 상대적인 빈도수에 대한 최소 공배수 T_{max} 를 구한 다음 각 디스크를 $T_i = T_{max}/f_i$ 개의 소단위(chunk)로 나눈다. 소단위 $C_{i,j}$ 는 디스크 D_i 의 j 번째 소단위를 의미한다)
5. 다음과 같은 방식으로 각 디스크의 소단위를 번갈아가며 방송 프로그램을 생성한다.

```

for i:=0 to  $T_{max}$ 
  for j:=1 to N
    소단위  $C_{j,(i \bmod T_j)}$ 를 방송한다
  endfor
endifor
    
```

<그림 1> 다중 디스크 방송 알고리즘

2.2 동기

고정된 방송 프로그램을 방송하는 데이터 방송 환경에서는 클라이언트의 데이터 액세스 패턴에 대한 정보를 나타내는 클라이언트 프로파일의 존재한다고 가정하는 것이 보통이며 클라이언트의 데이터 액세스 패턴을 정확하게 알 수 없으므로 패턴변화에 능동적으로 대처하지 못

하게 되어 궁극적으로 성능 저하를 초래할 수 있다. 이에 대한 해결책 중의 하나는 사용자의 패턴변화를 능동적으로 반영하여 매 방송 주기마다 새로운 방송 프로그램을 생성하여 방송하는 것이다.

[7]에서는 클라이언트의 요구패턴을 반영하기 위하여 비트 벡터를 이용하고 있으나 비트 벡터는 클라이언트의 요청 메시지에 피기배킹되어 서버에 전송되므로 클라이언트로부터 비트 벡터를 수신하지 못하였다는 것은 해당 클라이언트는 방송 프로그램에 만족한다는 것을 의미한다고 볼 수 있다. 그러나 클라이언트가 요구하는 데이터가 캐쉬에 존재하여 방송 내용과 무관하게 클라이언트의 데이터 요구가 만족될 수도 있게 된다. 뿐만 아니라, MFA(most frequently accessed) 값을 단순히 증가시켜 높은 값을 갖는 데이터들을 방송하게 되면 이전에 클라이언트들에게 인기가 높았지만 현재는 인기가 떨어진 데이터들도 당분간은 높은 값을 유지할 수 있어 방송될 가능성이 높아진다. 특히, 새로이 인기가 올라가는 데이터에 대한 방송이 당분간 지연되거나 방송 빈도수가 저하될 가능성이 있다. [보기 1]은 그 예를 보여준다.

[보기 1] 데이터 a, b, c, d에 대한 MFA 값을 각각 10, 9, 8, 1이라고 하자. 한번에 방송될 방송 프로그램의 크기는 3 즉, 3개의 데이터만 방송한다고 하자. 이 경우 d가 새로이 인기가 있어 계속 요청되는 데이터라 할 지라도 (클라이언트의 캐쉬에 있지 않는 한) 여러 번의 방송 주기가 지나 d에 대한 MFA 값이 8보다 크게 되어야 d가 방송될 수 있다. 이는 클라이언트의 응답시간을 크게 하여 성능을 저하시키게 된다. ■

결국, 모든 클라이언트로부터 비트 벡터를 받아 방송에 반영하고 아울러 최근에 요구되는 데이터는 MFA 값을 증가시키고 요구되지 않는 데이터는 MFA 값을 감소시킴으로써 최근에 요구되는 데이터가 방송될 가능성을 높일 수 있다. 특히, 여러 개의 디스크를 사용하는 환경에서는 방송 빈도수를 높여 클라이언트의 대기시간을 줄일 수도 있다.

2.3 방송 알고리즘

각 클라이언트는 서버에서 각 방송 주기 초에 먼저 방송한 방송 스케줄 즉, 데이터 식별자와 방송 주기 버전번호를 수신하여 저장한다. 필요로 하는 데이터가 캐쉬에 있으면 즉시 액세스하고 없는 경우 저장된 방송 스케줄을 검토하여 방송 리스트에 존재하면 방송되기를 기다린다. 방송 리스트에도 없는 경우 서버에게 데이터 요청 메시지와 함께 비트 벡터를 보낸다. 데이터 요청 메시지는 데이터 식별자, 주기 버전번호, 비트 벡터로 구성된다. 필요한 모든 데이터에 대해 캐쉬 히트나 air 히트인 경우 클라이언트는 현 방송 주기 말에 비트 벡터만을 의무적으로 서버에 송신한다.

비트 벡터는 방송된 데이터에 대한 클라이언트의 액세스 정보를 얻기 위해 각 클라이언트가 유지한다. 비트 벡터

의 각 비트는 현재 방송 프로그램의 각 데이터 페이지를 나타내며 새로운 방송 프로그램을 수신하면 클라이언트는 버전 번호를 갱신하고 비트 벡터를 모두 '0'으로 초기화한다. 요구한 데이터가 캐쉬에 없고 방송되는 데이터에 있으면(air hit) 해당 데이터 비트는 '1'로 변경된다. 캐쉬에 있는 경우나 직접 요청하여 방송된 데이터인 경우에는 비트 벡터에 영향을 주지 않는다.

한편, 비트 벡터에 '0'을 갖는 데이터는 클라이언트의 캐쉬에 있거나 불필요한 데이터임을 의미한다. 캐쉬에 있는 데이터는 점차 상대적으로 MFA 값이 낮아지게 되지만 캐쉬에 있으므로(캐쉬 대체 전략이 이상적이라면) 방송여부나 방송 빈도수의 변화가 문제되지 않는다. 따라서, 불필요한 데이터라면 MFA 값을 상대적으로 빠르게 낮추어 줄 필요가 있다. 한편, 비트 벡터에 '1'을 갖는 데이터는 최근에 자주 요구될 가능성이 있는 데이터이므로 방송하거나 혹은 방송 빈도수를 늘려 줄 필요가 있다. 그러므로, 최근에 요구되는 데이터를 처음으로 방송 혹은 자주 방송되는 데이터로 선호하기 위하여 비트 벡터가 '1'인 데이터만 고려하지 않고 '0'인 데이터도 다음과 같이 고려하여 MFA 값을 유지한다.

(R1) 현재 방송 주기동안 수신한 각 비트 벡터에서 각 데이터별로 비트가 '1'인 개수 $\sum_{i=1}^n b_i$ 의 합에서 '0'인 개수의 합을 뺀 값을 해당 데이터의 MFA 값에 더한다.

(R2) 요청된 데이터별로 요청된 횟수의 합을 해당 데이터의 MFA 값에 더한다.

따라서 최근에 air 히트와 air 미스가 많은 데이터일수록 높은 MFA 값을 갖게 되며 그 반대일수록 상대적으로 더욱 낮은 MFA 값을 갖게 되어 불필요한 데이터의 방송 중단이나 방송 빈도수를 빠르게 낮출 수 있게 된다.

서버는 각 데이터의 MFA 값을 갱신한 후 <그림 1>의 알고리즘에서 접근 확률 대신에 MFA 값을 사용하여 방송 프로그램을 생성한다.

3. 결론

본 논문에서는 방송 데이터에 대한 클라이언트의 만족여부를 방송 프로그램 생성에 반영하기 위해 비트 벡터를 이용하여 클라이언트의 변화하는 데이터 요구사항을 반영할 수 있는 방송 방안을 제안하였다. 제안한 방안에서는 기존의 비트 벡터를 사용하는 방안과 달리 모든 클라이언트의 만족도를 반영하고 최근의 요구 추세에 부응하고자 하는 시도를 하였다.

본 연구의 후속 연구로는 방송 프로그램의 효율성을 한층 더 얻기 위한 추가적인 비트 벡터를 전송 비용에 대한 분석이 필요하며 아울러 캐쉬 또한 성능에 영향을 주는 요인이므로 방송 알고리즘에 적합한 캐쉬 관리 방안 [8]에 대한 연구도 뒤따라야 한다.

참고문헌

- [1] D. Aksoy, M. Altinel, R. Bose, U. Cetintemel, M. Franklin, J. Wang, S. Zdonik, "Research in Data Broadcast and Dissemination", AMCP, pp. 194 - 207, 1998.
- [2] R. Alonso and H. F. Korth, "Database System Issues in Nomadic Computing," ACM SIGMOD, pp. 388-392, May 1993.
- [3] D. Barbara, "Mobile Computing and Databases - A Survey", IEEE Transactions on Knowledge Engineering, Vol. 11, No. 1, pp. 108-117, January/February 1999.
- [4] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments", Proc. of ACM SIGMOD, pp. 199 - 210, 1995.
- [5] S. Acharya, M. Franklin, S. Zdonik, "Balancing Push and Pull for Data Broadcast", Proc. of ACM SIGMOD, Tuscon, Arizona, pp. 183-194, May 1997.
- [6] W. Lee, Q. Hu, and D. Lee, "A Study on Channel Allocation for Data Dissemination in Mobile Computing Environments," ACM/Baltzer Mobile Networks and Applications, pp. 117-129, 1999.
- [7] Q. Hu, D. L. Lee, and W. C. Lee, "Dynamic Data Delivery in Wireless Communication Environment," white paper, GTE Lab. Incorporated, 1999.
- [8] G. Cao, "A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments," Proc. 6th Int. Conf. Mobile Computing and Networking, pp. 200-209, 2000.