

모바일 Broadcasting 환경에서 User Preference 기반 캐시 관리 기법

최영환, 황인준
고려대학교 전기전자공학과
e-mail : {work48, ehwang04}@korea.ac.kr

A User Preference-based Cache Management Scheme In a Mobile Broadcasting Environment

Young-Hwan Choi, Een-Jun Hwang
Dept. of Electrical Engineering, Korea University

요 약

최근 모바일 Broadcasting 환경에서의 캐시 관리 기법에 관한 다양한 연구가 활발히 진행되고 있으며 가장 많이 쓰이는 기법으로는 무효화 보고(Invalidation Report) 기법을 들 수 있다. 하지만 무선 기기들의 대용량 지원 및 많은 사용자의 무선 이용으로, 사용자 요청에 대한 지연이나 Multi-Cell 환경과 대용량 갱신 등에 대한 대처 능력의 부족 등이 문제점으로 부각되고 있다. 본 연구는 이런 무효화 보고의 대처 능력을 보완할 뿐 아니라, User Preference 를 추가하여 사용자의 QoS 를 만족시키는 새로운 시스템을 제안한다. 본 연구는 서버 측에서의 일방적인 브로드캐스팅에 의한 데이터 전송이 아닌, 사용자로부터의 요청에 따른 캐시 데이터 관리 기법을 제안한다. 연구의 주된 효과는 사용자로 하여금 선택적 청취(Selective Listening)을 하게 함으로써 서버와의 교류를 적게 하고, 자주 사용하는 많은 양의 데이터를 한번에 가져와 빠른 시간 내에 데이터를 사용할 수 있게 한다. 또한, 자신이 필요한 데이터에 한에서만 자료 갱신(Update) 여부를 확인하여, 짧은 시간 안에 동적으로 자신의 정보를 확인 할 수 있다.

1. 서론 및 기존 연구

User Preference 란 사용자측이 중심이 되어, 사용자가 원하는 것을 요청하고 받아 낼 수 있는 사용자 편의 환경이다. 무선 환경에서의 User Preference 는 사용자가 중심이 되어, 이동 기기(Mobile Device)와 서버와의 통신을 제어하고, 사용자 요청에 의하여 움직이는 사용자측 모델이다. 하지만 현재 모바일 환경에는 대역폭 한계, 비대칭 통신 환경과 같은 제약이 있어, 사용자들의 요구를 충족 시켜 주는데 많은 어려움을 수반하고 있다. 무선 환경에서의 User Preference 의 필요성에 대한 예로써 증권 시장을 들 수 있다. 증권 시장의 특징은 빠르고 정확한 정보를 사용자에게 최단 시간에 도착 해야 한다는 것이다. 이동 기기를 사용하는 사용자가 자신이 원하는 주식에 대한 정보를 기존 데이터 갱신 주기보다 더 빠르고 정확하게 얻기 위하여, 다양한 데이터 갱신 기법들이 제안 되었다.

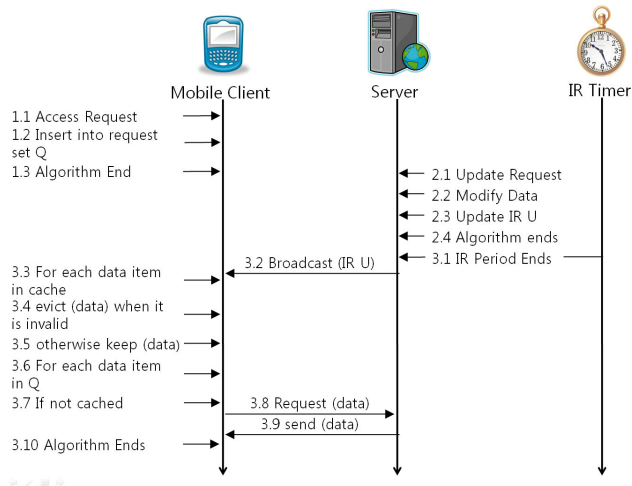
데이터 갱신 기법은 크게 서버 중심과 사용자 중심 갱신 기법으로 나눌 수 있다. 서버 중심의 대표적인 기법으로는 무효화 기법 (Invalidation Report, IR)이 있다. IR 기법 외에도, 캐시를 관리하기 위한 다른 기법들에 대한 연구가 활발하게 진행되어 왔다. 예를 들면, Call-Back Scheme[6], Lease Scheme[6][9][10], Poll-Each-Read(PER)[6][8]등이 있다. IR 기법은 서버에서

Invalid 한 데이터에 대해 보고를 보내는 형식이므로, 만약 많은 양의 갱신이 서버에서 이루어 졌다면 사용자가 자신의 데이터가 Invalid 한지 확인 하기 까지는 시간이 걸리므로, 사용자가 새로운 정보를 받았을 때, 이미 데이터는 과거의 데이터가 되어 있을 가능성이 있기 때문이다. 이런 문제들로 인해 무선 환경에서의 통신 방법의 개선이 필요하다. 기존에 연구 되어 왔던 IR 기반의 Timestamp 기법[4][6]은 서버는 주기적으로 Timestamp 를 포함한 bit-stream 형태의 IR 을 사용자에게 보내고, 사용자는 자신의 캐시에 저장 되어 있는 데이터의 Timestamp 와 비교하여, 만약 기간이 만료된 데이터일 경우 캐시를 갱신 하는 기법이다. 서버는 사용자에게 IR, 즉 갱신된 데이터의 ID 들과 데이터들의 Timestamp 를 그룹화 시켜 이를 사용자에게 Broadcast 하게 되고, 사용자는 IR 에 포함된 데이터 ID 들 중에서 자신의 캐시가 포함하고 있는 데이터 ID 를 찾은 후, Timestamp 를 비교하게 된다. 만약 데이터가 갱신 되었다면, 서버에 다시 자료를 요청하여, 자신의 캐시에 저장된 데이터를 갱신 하는 방식을 사용하고 있다. (그림 1)은 앞에서 설명한 Timestamp IR 방식을 보여주고 있다. Timestamp 기법의 문제점은 많은 양의 갱신이 서버 측에서 이루어 지게 될 경우, IR 의 길이가 길어 지게 되고, 사용자는 자신의 캐시에 저장 되어 있는 데이터의 갱신 여부를 확

인 하기까지 지연이 생긴다는 것이다. 또한, 사용자가 Off-line 상태에서 On-line 상태가 되었을 때, 사용자 기기의 캐시에 저장 되어 있는 데이터가 Valid 한지 확인을 하기 위해서, 서버에서 IR 을 보낼 때까지 기다려야 하는 문제점이 있다.

Call-Back Scheme[6]은 서버가 사용자에게 데이터의 갱신을 통보 해야 하는 모든 책임을 가지기 때문에, 갱신이 이루어지면 바로 사용자에게 Invalidation Message 를 보내어 사용자가 갱신 요청을 진행 하는 방식이다. 하지만 이 방법의 문제점은 서버가 모든 사용자가 캐시에 어떤 내용을 저장 하고 있는지 알고 있어야 한다는 것이다. Lease Scheme[6][9][10]은 만약 사용자가 Off-Line 상태에서 On-line 상태가 되었을 때, 서버 측에 자신의 캐시에 있는 데이터에 대한 Lease 정보를 보내게 되고, 만약 Lease 가 만료 됐다면, 서버는 갱신된 데이터와 새로운 Lease 정보를 보내게 된다. 그리고 On-Line 에 있었던 사용자에게 대해서는 갱신이 일어나면 사용자에게 데이터를 보내는 Call-Back Scheme[6] 과 같은 구조를 가지고 있다. 결국 Lease Scheme[6][9][10]또한 Call-Back Scheme[6]이 가지는 단점을 동일 하게 가지게 된다. 본 논문에서 제안하는 기법에도 사용되는 Poll-Each-Read(PER)[6][8] 기법은 사용자가 서버에게 자신이 가지고 있는 데이터에 대한 갱신 여부를 물어보고 이에 대한 Acknowledgement (Ack)를 통해 그 여부를 확인하여 자신의 캐시 갱신을 결정하게 된다. 사용자 측에서 확인을 한다는 측면이 사용자측 모델을 제안하는 본 연구와 비슷하여 이 기법을 사용하였다.

이 논문은 새로운 User Preference 에 의한 무선 환경의 통신 방법과 캐시 관리 기법을 제안한다. 기존의 연구들은 무효화 보고(Invalidation Report) 를 기반으로 보다 빠르고 정확한 정보를 제공하는 데 노력을 기울였다. 본 연구는 서버 측의 일방적인 보고를 받는 형태인 무효화 보고를 사용하는 시스템이 아닌 사용자의 요구에 따라 데이터를 저장하고 일관성을 유지하게 관리함으로써, 사용자에게 최신의 정보를 빠르게 제공하는 사용자 중심의 시스템 개발을 목표로 두고 있다.



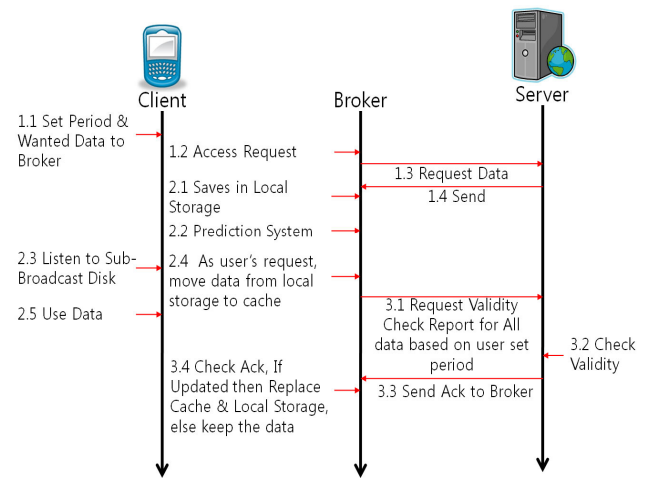
(그림 1) Time Stamp IR

본 논문의 구성은 다음과 같다. 2 장에서는 본 논문에서 제안하는 모델의 상세 내용을 서술하고 있다. 3 장에서 제안된 모델의 예상되는 결과를 보여주고, 마지막으로 4 장에서는 본 연구에 대한 결론을 맺는다.

2. User Preference 캐시 관리 기법

본 장에서는 논문에서 제안하는 User Preference 캐시 관리 기법에 대해 살펴본다. 우선 2.1 장에서는 User Preference 의 시나리오를 제시하고, 2.2 장에서는 제안된 시스템의 구조에 대해 설명한다.

2.1 시나리오



(그림 2) User Preference 기반 캐시 관리 기법

(그림 2)는 제안하는 전체 시스템의 시나리오를 보여주고 있다. 1.1 은 사용자가 새로운 갱신을 확인하고 싶은 주기를 설정하고, 사용할 데이터들을 선정 하게 된다. 1.2 에서 데이터 Broker 가 서버에게 접근 요청을 하게 되고, 접근이 허가가 되면 서버에게 사용자가 필요로 하는 데이터를 요청하게 되고, 서버는 사용자가 원하는 데이터를 전송한다.

2.1 에서 데이터 Broker 는 이를 받아 로컬 저장소에 저장을 하게 된다. 2.2 에서는 저장을 한 후 Prediction System 을 실행하게 된다. 2.4 에서는 사용자가 필요하다고 요청하는 자료나 다음에 사용할 데이터를 캐시로 데이터를 옮기게 되고, 2.5 에서 사용자는 이를 사용하게 된다.

3.1 에서는 사용자가 설정해 놓은 주기가 되면, 서버에게 자신이 받아 놓은 데이터에 한해서, Validity Check Report 를 보내게 된다. 3.2 에서는 서버는 받은 Report 에 대해서 Ack 를 작성하여, Validity Ack 를 보내게 된다. Broker 는 3.4 에서 Ack 를 분석하여 만약 갱신이 일어난 데이터에 대해서는 서버에 다시 요청을 하여 정보를 다시 받게 된다.

실제 사용 예와 비교를 해보면, 사용자가 주식 항목 1, 2, 3, ..., 20 에 대한 정보를 요청하고, 주기는 3

초로 설정한다고 가정하자. 데이터 Broker 는 요청을 받고 서버에 주식 항목 1 부터 20 까지 자료를 요청하게 된다. 그리고 서버로부터 자료가 수신되는 대로, 데이터를 로컬 저장소에 저장하고, 받은 데이터를 기반으로 Prediction System 을 이용 하여, Data scheduling 을 하게 된다. Prediction System 은 사용자가 데이터를 이용하는 패턴을 분석 하여, 사용자가 현재 사용하고 있는 데이터 다음에 사용할 가능성이 있는 데이터들을 캐시에 옮겨 놓는 역할을 담당한다.

2.2 System Architecture

시스템 설명으로 들어가기 앞서, 디자인 하고자 하는 시스템은 PER scheme 을 사용 하고, 서버는 사용자의 요청을 받고, 필요로 하는 데이터를 사용자에게 보내어 주는 시스템으로 가정 한다.

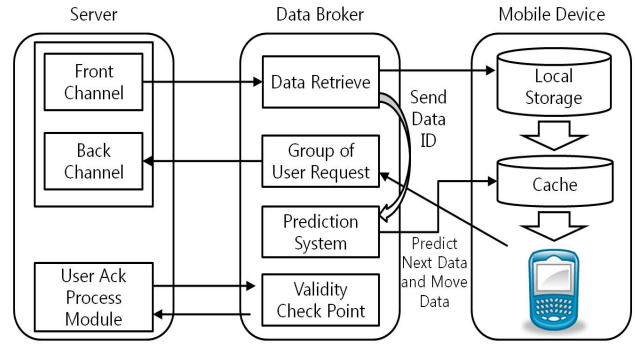
데이터 Broker 의 개념은 사용자의 자원에서 서버와 1:1 로 통신하며, 중간에서 기존보다 원활한 데이터 교류를 충족 시켜 주는 개념을 의미한다. 데이터 Broker 가 기본적으로 하는 일로는 Prediction System, Validation Check Report 와 이에 대한 Ack 를 수신하여 분석 하는 역할을 한다.

기존의 IR 기반의 캐시 관리 방법은 서버에서 보내어 지는 모든 데이터에 대한 IR 을 분석하여, Local Cache 의 유효성을 확인 한 뒤, Invalid 한 데이터에 대해 갱신을 하는 시스템을 가지고 있다. (그림 3)에서 보는 것과 같이 Prediction System 을 두어, 사용자에게 꼭 필요한 데이터들만 확인하는 방법이다. 사용자의 Mobile Device 에서 작동하는 데이터 Broker 시스템은 초기 상태에서 서버로의 접근을 허용 받게 되면, 사용자가 요청한 많은 데이터들을 Back-channel 을 통해 서버에 요청 하게 된다. 서버는 데이터를 Front-channel 을 이용하여 데이터 Broker 에게 전달 하게 되고, 데이터 Broker 는 데이터들을 디스크화 시켜 로컬 저장소에 저장하게 된다. Prediction System 은 데이터 Broker 가 데이터를 서버로부터 전송 받을 때 부여한 데이터 ID 를 이용하여, 시스템에게 사용자가 다음에 사용할 것 같은 데이터를 ID 에 의해 물어보게 된다. 예를 들어, 사용자가 ID: 2 번 다음에 ID: 10 번을 자주 사용한다면, Prediction System 은 사용자의 패턴을 분석 하여, 현재 캐시에 존재하는 데이터의 ID 가 2 번이라면 2 번을 캐시에서 Drop 하고 ID 10 번의 데이터를 캐시에 옮겨 놓는다.

하지만 로컬 저장소의 크기가 이동 기기라는 것을 가정 했을 때 크지 않을 것이므로, Dynamic 하게 삭제를 해야 한다. 사용자가 데이터를 이용하는 빈도수에 따라 만약 많이 사용하지 않는 데이터가 존재한다면, 이를 로컬 저장소에서 삭제를 해주어야 한다.

또한 시스템은 저장된 데이터들의 ID 를 그룹화 시켜, 서버에게 갱신을 물어보는 Report 형식의 Validation Report 를 보내게 된다. 서버는 이에 대한 대담으로 데이터 ID 에 따른 단순 bit-stream 형태의 Ack 를 보내주게 된다. 데이터 Broker 시스템은 Ack 를 분석하여, 갱신된 데이터를 서버에 요청하게 되고, 서버서부터 전송 받은 데이터로 로컬 데이터의 Invalid 한

데이터를 갱신 한다.



(그림 3) 데이터 Broker

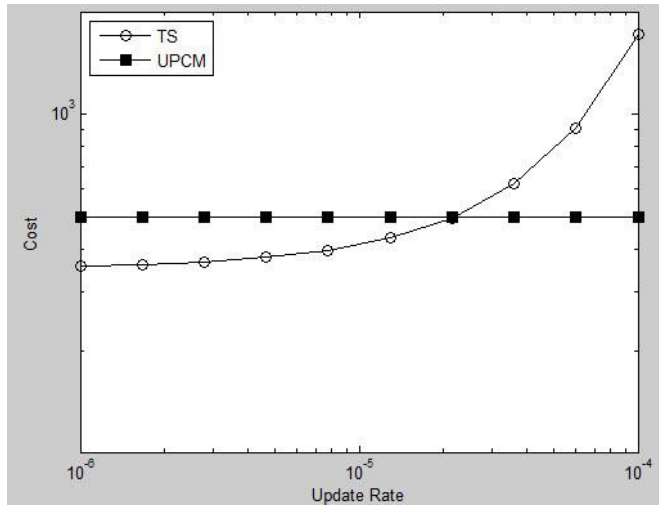
데이터 Broker 의 중요한 역할은 사용자가 자신이 원하는 정보 갱신 주기와 필요한 데이터를 선정 할 수 있다. 만약 사용자가 필요한 데이터가 10 초 마다 갱신이 이루어지기 원한다면 주기를 설정 하여, 시스템은 10 초 마다 서버에게 Validation Report 를 보내주고, 갱신이 이루어진 데이터가 존재 한다면, Broker 시스템은 갱신을 진행 하게 된다. 갱신은 사용자가 현재 사용하지 않고 있고, 로컬 저장소에 존재하는 데이터에 대해서 Validation Check Report 를 보내고, 서버의 Ack 를 분석하여 갱신이 이루어진다. 기존의 IR 방식의 일방적인 Report 전송 방식이 아닌, 사용자의 필요에 따른 방식이라고 할 수 있다.

또한, 데이터 Broker 시스템은 초기 설정에서 사용자가 직접 필요하거나, 필요할 수 있는 데이터를 선정하여 List 를 만들게 되고, Request Report 를 만들어 초기에 많은 데이터를 한꺼번에 다운로드 하게 된다. 이 과정에서 사용자의 자원에 영향을 미칠 수 있으나, 다운로드 후의 사용에서는 빠른 속도로 사용 할 수 있게 되므로 Trade-Off 라고 할 수 있다.

데이터 Broker 시스템의 장점으로 보면 사용자는 IR 을 받는 시간을 기다리지 않아도 되는 것이고, 항상 새로운 정보를 사용 할 수 있는 것이다. 그리고 서버는 IR 을 송출 하는 시간을 줄일 수 있으며, 사용자 요청에 응답을 하는 단순한 서버의 역할을 하게 된다. 마지막으로 서버로부터의 다운로드가 아닌 로컬 저장소상에서의 정보 이용이므로 사용자의 지연 시간이 줄어 들게 된다.

제안하는 시스템의 가장 큰 단점은 데이터 Broker 가 사용자 측에서 실행이 되어야 함으로, 사용자의 자원에 영향을 끼칠 수 있다. 이에 대한 해결방법으로는 사용자의 기기의 성능에 따라, 주기에 제한을 두어 사용자의 기기에 최소한의 Overload 를 고려한 시스템의 설계가 필요하다. 데이터 Broker 실행 시, 대용량의 데이터를 원하게 된다면, 데이터 갱신 시에 기기의 성능 저하가 우려 되므로, 기기의 성능에 따라 Dynamic 하게 초기 설정 데이터의 양을 제한하는 시스템의 설계 또한 필요 하다.

3. Preliminary Result



(그림 4) Preliminary Result

(그림 4)는 예상되는 결과를 나타내고 있다. (그림 4)는 데이터의 갱신 비율에 따른 사용자측의 캐시 갱신 Cost 를 나타내고 있다. 기존 연구[6]의 결과를 참조 하여 예상 되는 결과를 만들어 보았다. 기존 연구에서는 PCS 환경을 기준으로 Performance Evaluation 을 실험 하였다. UPCM 은 User Preference 캐시 관리 기법을 나타내고, IR 기반의 Timestamp 를 사용한 기존의 시스템을 나타낸다.

기존 연구[6]에서 보면 IR 기반의 환경에서는 데이터에 대한 갱신 비율이 높아 짐에 따라, Cost 가 상승 하는 결과를 보여줬다. 하지만 Poll-Each-Read(PER)를 사용한 시스템은 데이터의 갱신 비율이 상승 함에 따라, Cost 가 낮아지는 것을 [6]에서 확인 할 수 있었다. IR 기반 Scheme 의 갱신 비율이 높아 짐에 따라 Cost 가 상승 하는 이유는, 많은 데이터 갱신이 이루어 지는 경우 IR 의 크기가 커지므로 Cost 가 상승 하게 된다. 하지만 PER Scheme 은 자신이 필요한 데이터들에 한에서만 사용자가 갱신을 함으로, 갱신 비율과는 상관 없이 오히려 Cost 가 하향하는 결과를 보여줄 것으로 예상된다. 따라서, 본 논문에서 제안 하는 시스템 또한 PER 을 사용 하므로 데이터의 갱신 비율에 대한 Cost 가 줄어 들 것으로 예상된다.

4. 결론

본 논문은 User Preference 를 사용한 캐시 관리 기법을 제안 하였다. 제안한 시스템을 사용 한다면, 많은 사용자들의 데이터 일관성을 보장 할 수 있을 것이다. 서버 측의 이점은 IR 의 구성이 필요 하지 않고, 사용자의 이점은 새로운 정보를 원하는 시간에 받아 볼 수 있게 된다는 것이다. 서버는 데이터 Broker 측에서 보내어 지는 Validity Check Report 로 인해, 서버는 Ack 만을 보내게 됨으로 부담이 적어 지게 된다. 사용자는 많은 양의 정보를 한번에 가져와, Local 에서의 데이터 전송으로 많은 양을 빠르게 확인 할 수 있다는 장점이 있다.

향후 과제로는 지금 현재 시스템이 가지고 있는 문

제인 다수의 사용자들의 Validity Check Report 의 효과적인 처리 방법의 개발이 필요하다. 다수의 사용자가 많은 양의 데이터를 포함하는 Validity Check Report 를 보내게 되면, 서버는 Report 를 처리 하기 위하여, Interruption 이 걸리게 되고, 결국엔 Delay 로 이어 지게 된다. 다른 향후 과제로는 사용자측의 자원 제어를 위하여 Dynamic 하게 사용자의 자원을 관리 하는 관리 시스템의 개발이 필요 하다. 사용자측의 자원은 한정되어 있기 때문에, 데이터 Broker 의 자원을 항상 유지 할 수 없다. 그러므로 데이터 Broker 의 최소의 자원을 유지 시켜 줄 수 있는 시스템의 개발이 필요하다.

참고문헌

- [1] S.Acharya, R.Alonso, M.Franklin, S.Zdonik, "Broadcast Disks: Data Mangement for Asymmetric Communication Environments", Proc. ACM SIGMOD Conf., San Jose, CA, May, 1995
- [2] S.Acharya, M.Franklin, S.Zdonik, "Prefetching from a Broadcast Disk", 12th International Conference on Data Engineering, New Orleans, LA, February, 1996
- [3] S.Acharya, M.Franklin, S.Zdonik, "Balancing Push and Pull for Data Broadcast", Proc. ACM SIGMOD Conf., Tuscon, Arizona, May, 1997
- [4] D.Barbara and T.Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," VLDB Journal, Vol. 4, No. 4, pp. 567-602, 1995
- [5] D.Barbara, "Mobile Computing and Databases-A Survey", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, January/February, 1999
- [6] H.Chen, Y.Xiao, S. Vrbsky, "Invalidation Report Scalability of Cache Access Mechanisms in Future Multiple-Cell Wireless Internet", Proc. Of IEEE GLOBECOM, 2007
- [7] K.Wu, P.Yu, M.Chen, "Energy-Efficient Caching for Wireless Mobile Computing", IEEE Proc. Of the Twelfth International Conf. on Data Engineering, 1996
- [8] Y.-B.Lin, W.-R. Lai and J.-J. Chen, "Effects of Cache Mechanism on Wireless Data Access," IEEE Trans. On Wireless Communications, Vol. 2, No. 6, pp. 1240-1246, 2003.
- [9] V. Duvvuri, P.Shenoy, R.Tewari, "Adaptive Leases: A Strong Consistency Mecahnism for the World Wide Web," IEEE Trans. Knowledge and Data Engineering, Vol. 15, No. 4, July/August, 2003
- [10] J.Yin, L.Alvisi, M.Dahlin, and C.Lin, "Volume Leases for Consistency in Large-Scale Systems," IEEE Trans. On Knowledge and Data Engineering, Vol. 11, No. 4, pp.563-576, 1999