

무선 근거리 통신망 환경에서의 데이터베이스 미들웨어

박제호

단국대학교 컴퓨터과학과

e-mail:dk_jhpark@dankook.ac.kr

A Database Middleware for Wireless LAN

Je-ho Park

Dept of Compute Science, Dankook University

요 약

무선 근거리 통신망의 보급이 확대에 의해, 기존의 유선 통신망을 이용한 데이터베이스 접근을 중심으로 고려한 데이터베이스 시스템의 구조는 새로운 무선 접근 경로를 이용하는 사용자들을 효율적으로 지원해야 한다. 무선 통신망의 성능이 비교적 느린 전송 속도와 무선 통신망을 이용하는 사용자의 이동성은 데이터베이스 시스템의 환경에 변화를 필요로 하게 한다. 막대한 데이터의 양과 많은 수의 사용자를 동시 지원을 위한 과거 클라이언트-서버 두 계층 구조는 데이터의 지원이 서버에 국한되어 성능의 한계를 가지고 있다. 이 논문은 클라이언트-서버 데이터베이스 시스템의 구조를 사용자들의 데이터 사용 행태를 분석하여 데이터 서비스를 서버와 클라이언트에 분산시켜 시스템 효율을 증가시킬 수 있는 새로운 데이터베이스 시스템에 대해 논의한다. 실제 활용성의 검증을 위해 시뮬레이션 기법을 사용하여 효율을 예측하였다.

1. 서론

무선 근거리 통신망은 유선 설치가 어려운 환경 까지도 네트워크를 확장시킬 수 있는 이동성, 유연성, 휴대성, 간편성 등의 이점으로 응용분야가 확산되고 있다. 이러한 무선 근거리 통신망을 이용한 데이터 접근 경로는 유선 기반 시스템과 연동되면서 혼합형 시스템 환경을 만들어 내고 있다. 일반적으로 대다수 산업적인 생산 시스템에 사용되는 데이터베이스는 그 양이 방대할 뿐만 아니라 동시에 데이터를 사용하는 사용자가 많다는 특징을 보이고 있다. 따라서 시스템을 구성하는 네트워크가 유무선 시스템으로 혼합적인 형태일 경우 데이터베이스 시스템은 다른 통신 접근 경로를 이용하는 사용자들을 모두 고려하여 효율적인 구조를 갖추어야 한다.

네트워크 환경에서 대용량 데이터를 접근하는 많은 수의 동시 사용자를 지원하는 문제를 해결하기 위해 새로운 방법에 대한 연구를 한 결과 데이터를 관리하는 서버와 고성능의 워크스테이션을 이용하는

클라이언트-서버 데이터베이스 시스템은 원활한 데이터의 운영과 개선된 데이터 서비스라는 우수함을 보였다[1]. 하지만, 시스템에 동시 접근하는 사용자의 수가 일정 한계를 넘을 경우 시스템의 성능이 현격히 저하되는 현상 즉 범위성(scalability) 문제를 해결하고자 하는 개선책에 대한 연구가 진행되어 왔다. 그 해결 방법으로 클라이언트 사이트에 과거에 접근한 데이터를 임시 기간 동안 저장하는 캐싱(caching)이 유효한 것으로 알려져 있다[2]. 또한 고속 네트워크의 실용적인 보급이 확대되면서 네트워크를 통해 연결된 사용자가 저장하고 있는 데이터를 이용하는 방법도 제안되었다[3]. 대용량 데이터에 접근하는 다수의 사용자가 데이터를 공유하면서 협업을 기반으로 특정 목적을 달성하는 작업환경에서의 데이터베이스 시스템은 또 다른 응용환경의 특성을 보여준다. 이러한 환경에서는 각 사용자는 논리적 측면에서 데이터베이스의 특정 부분(working area)을 많이 접근하는 양상을 보인다[4]. 또한 사

용자 간에는 이러한 특정 부분을 공유하는 현상도 볼 수 있다. 각 특정 부분 데이터베이스에 접근하는 사용자들은 다른 특정 부분에 접근하는 빈도수가 적은 것으로 알려져있다[?]. 상위 환경과 유사한 데이터 접근 형태를 보이는 환경에서 클라이언트-서버 데이터베이스를 적용하였을 때, 클라이언트 캐싱은 접근된 데이터를 서버가 아닌 자체 캐싱 저장소에서 찾을 수 있는 확률이 높기 때문에 캐싱을 이용하여 성능을 개선할 수 있다[2,4,5,6].

이 논문에서는 공유되는 데이터를 사용하는 클라이언트들을 하나의 논리적 클러스터로 구성하여, 기존의 클라이언트-서버가 가지는 범위성의 한계를 해결하고자 한다.

2. 디렉토리 미들웨어 구조

클라이언트-서버 데이터베이스를 접근하는 클라이언트의 데이터 요구는 서버로 전달되어, 서버에 캐싱 기능이 있을 경우, 요구된 데이터가 캐시에 저장되었는가 하는 여부에 따라, 캐시에 요구된 데이터가 부재하는 경우에는 데이터를 저장하고 있는 서버로 데이터 요구 메시지가 전달된다. 이 때 서비스 되는 데이터는 캐시에 먼저 저장된다. 반대로 캐시에서 요구된 데이터가 발견된 경우 데이터 객체가 클라이언트로 전송된다. 여러 클라이언트 중 한 클라이언트가 데이터 갱신을 할 때는 다른 모든 클라이언트로부터 갱신에 대한 동의를 구하기 위해 자료소환(callback) 방법을 사용한다. 이와 같이 클라이언트-서버 데이터베이스 환경에서는 서버가 데이터를 저장하고 관리하는 총괄적 역할을 수행하고, 아울러 데이터의 읽기/쓰기 작업간의 조절을 담당하고 있다. 따라서 클라이언트-서버 시스템은 범위성 문제를 피할 수가 없게 된다.

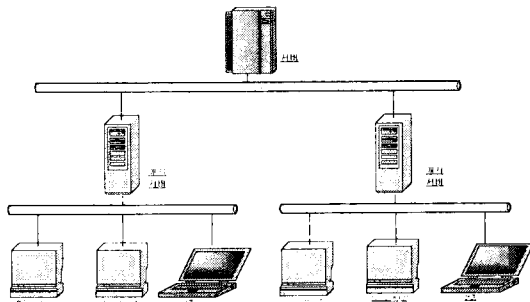


그림 1. 캐시서버를 이용한 구조

이러한 범위성 문제를 해결하기 위해서는 서버 작

업량을 감소시켜 시스템 성능을 높여야 한다. 이를 가능하게 하기 위하여 서버와 클라이언트 계층 사이에 새로이 캐싱 서버 계층을 사용할 수 있다. 전체 클라이언트군은 다수의 네트워크 고정 클러스터로 분할하여 한 클러스터 당 캐싱 서버(CSL)를 장착하여, 데이터 요구는 먼저 캐싱 서버에서 서비스 만족 여부를 검사하고, 클러스터 내부에서 만족되지 못한 데이터 요구만이 서버 계층에 의해 충족된다. 결과적으로 클라이언트 간에 데이터의 사용 행태가 유사하다면 데이터 서비스의 많은 양을 서버가 아닌 클러스터 범위 안에서 해결할 수 있게 되어, 전체 시스템의 효율을 높일 수가 있다. 이러한 구조는 그림 1에 예시되어 있다.

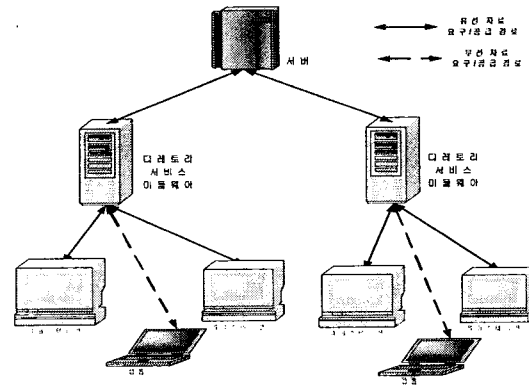


그림 2. 논리적 연결에 의한 구성

이 같은 구조에서 자료소환 과정은 두 개의 계층에서 두 단계로 수행된다. 데이터 갱신이 필요할 경우 먼저 갱신허가 신청을 해당 CSL에게 전송하면 CSL은 서버에게 신청을 전달하고, 서버는 해당 데이터를 소환한다. 서버는 데이터 소환을 중간 계층인 CSL들에게 전달되어 소속 클라이언트로부터 해당 데이터를 모두 소환한 다음 서버에 자료소환에 대한 동의를 전송한다. 모든 CSL로부터 자료 갱신에 대한 동의를 모두 받았을 때 서버는 자료 갱신 요구를 보낸 CSL에게 갱신 허가를 전송한다. 이 때 해당 CSL은 자료 갱신을 요청한 클라이언트에게 허가사항을 통보한다. 자연스럽게 이러한 구조는 삼중 구조 형태를 가지게 되며, 서버에 집중되는 작업량의 분산 효과를 창출시켜 전체 시스템의 효율을 높게 된다. 이러한 구조를 위해 네트워크 분할을 이용할 경우, 클라이언트의 소속 클러스터 변경이 필요한 할 경우 많은 수작업이 필요하게 되어 또 다른 문제에 직면한다. 이에 대한 해결책은 클러스터를

논리적인 연결을 이용하여 구성하는 것이다. 새로운 논리적 연결 기반 네트워크 구성은 그림 2 와 같이 보인다.

논리적 클러스터를 구성할 때 소속 클라이언트의 선택은 클라이언트의 데이터 이용 패턴의 분석에 기초를 둔다. 클러스터링 알고리즘은 클라이언트의 과거 데이터 접근 정보를 토대로 접근 행태 유사성 기반하여 클러스터에 대한 소속을 결정할 수 있다. 이 논문에서는 제안된 구조가 범위성 문제를 해결함을 검증하기 위해 수행한 실험 결과를 제시한다.

3. 구현 사례

CSIM을 이용한 시뮬레이션은 세 가지 형태로 구성되었다. 각 시뮬레이션은 클러스터의 수를 증가시켜 한 개(SCCA-1), 다섯 개(SCCA-5), 열 개(SCCA-10)의 클러스터로 구성되는 되는 시스템을 두 계층(FCA) 환경과 성능을 비교하였다. 수집한 실험 수치로는 클라이언트 수를 증가시키면서, 서버 수행률을 측정하였다.

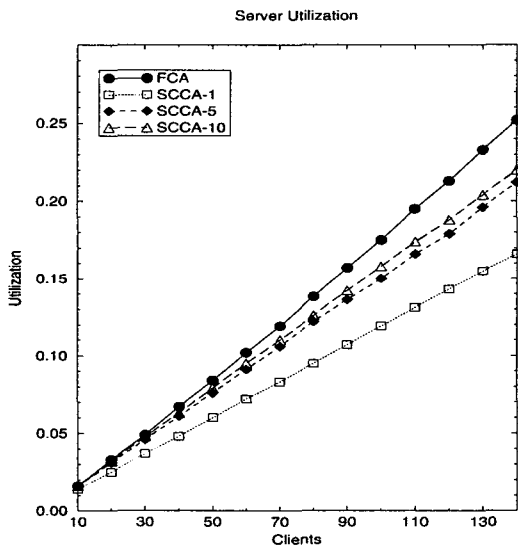


그림 3. 사용자 증가 시 서버 수행률

그림 3은 사용자를 증가시키면서 발생하는 서버 수행률을 보여주고 있다. 결과에서 볼 수 있듯이 두 계층 데이터베이스를 적용할 때 클라이언트의 수에 상관없이 서버의 사용률이 제일 많은 것으로 나타난다. 클러스터를 하나로 구성하였을 때, 클러스터의 내부에서 만족 가능한 데이터요구의 양이 최대가 되어 서버의 수행률은 최저를 나타내고 있다. 클라이언트의 수가 140 일 때, 클러스터 하나를 사용하는 것이 다섯 개를 사용할 경우보다 약 10 %가 넘는

서버의 수행률이 감소되는 것으로 시뮬레이션 결과에서 알 수 있다. 이는 클러스터 내부 만족도가 여러 개의 클러스터보다는 한 개의 클러스터로 구성될 때 이론적으로 높기 때문이다.

4. 결론

클라이언트-서버 데이터베이스는 범위성은 사용자의 자료 접근 방법이 유/무선으로 혼용되는 환경에서 이 문제는 더욱 심각해진다. 이에 대한 대안으로 사용자의 데이터 사용 행태 분석에 의거하여 계층을 확장하고 중간 계층에 서버가 담당해야 하는 데이터 요구의 일부를 처리하는 계층을 가진 새로운 데이터베이스의 구조를 제안하였다. 시뮬레이션을 통해서도 제안된 시스템은 클라이언트-서버 시스템보다 효율이 월등히 개선되는 것을 알 수 있다.

참고문헌

- [1] L. Liu, C. Pu, and W. Tang. "WebCQ: Detecting and Delivering Information Changes on the Web." In Proceedings of International Conference on Information and Knowledge Management, Washington, DC, 2000.
- [2] M. Franklin, M. Carey and M. Livny. "Transactional Client Server 캐시 Consistency: Alternatives and Performance." ACM Transactions on Database Systems, 22(3):315-363, 1997.
- [3] M. Dahlin, C. Mather, R. Wang, T. Anderson, and D. Patterson. "A Quantitative Analysis of 캐시 Policies for Scalable Network File Systems." In Proceedings of the Sigmetrics Conference on Measurement and Modeling of Computer Systems, pages 150-160, Nashville, Tennessee, May 1994.
- [4] M.A. Ranft and S. Rehm and K.R. Dittrich. "How to Share Work on Shared Objects in Design Databases.", In 1990 Proceedings of Sixth International Conference on Data Engineering, Los Angeles, CA, February, 1989.
- [5] S. Helal, J. Hammer, J. Zhang and A. Khushraj. "A Three-tier Architecture for Ubiquitous Data Access." In Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications, Beirut, Lebanon, June 25-29, 2001.