

고가용성의 DSMS 환경에서 보조 서버 규모 최소화를 위한 선택적 데이터 동기화 기법

지민섭*, 신승선*, 배해영*

*인하대학교 컴퓨터공학과

e-mail:{msji, hermit}@dmlab.inha.ac.kr, hybae@inha.ac.kr

A Selective Data Synchronization Algorithm for Minimizing Backup Server Scale in DSMS Environment of High-Availability

Min-Sub Ji*, Sung-Sun Sin*, Hae-Yeong Bae*

*Dept of Computer Science, Inha University

요 약

최근 DSMS의 연구가 활발히 진행되고 있고, 그에 따른 고가용성 서비스를 제공하기 위한 환경으로 확장되고 있다. DSMS와 같은 실시간 데이터베이스 응용 시스템 분야나 임무 결정적인 응용 시스템을 다루는 산업 분야에서 가용성을 유지 하지 못하는 것은 치명적인 결과를 초래 할 수 있다. 이런 DSMS 환경에서 고가용성을 유지하는 기법으로는 Active/Active 알고리즘과 Active/Standby 기법이 있다. 하지만 이 기법들의 고가용성을 유지하기 위해 주 서버와 동일한 규모의 보조 서버를 유지해야 하는 단점이 있다. 본 논문에서는 질의를 두 가지 타입으로 나누어 누적 데이터가 필요한 질의들만 실시간 동기화를 통해 보조 서버에 유지하고 나머지 질의들에 대해서는 메타 데이터들만 보조 서버에 유지해 보조 서버의 규모를 줄일 수 있었다.

1. 서론¹⁾

최근 센서장비 및 RFID, 통신기술의 발달에 의해 수많은 정보가 빠르게 유입되는 데이터 스트림(Data Stream)을 처리하기 위한 데이터 스트림 관리시스템(Data Stream Management System)에 대한 연구가 활발히 진행되어왔다[1,2]. DSMS는 데이터 스트림을 입력받아 등록된 연속질의를 통해 질의를 처리하여 그 결과를 사용자 또는 다른 시스템이나 응용프로그램으로 제공하는 시스템으로 대표적인 연구로는 Aurora[4], STREAM[5], NiagaraCQ[6], TelegraphCQ[7]등이 있다.

이런 DSMS 시스템 중 국내 지능형국토정보기술혁신 사업과제의 일환으로 개발된 GeoSensor Data Stream Management System은 유비쿼터스 공간 컴퓨팅 환경에서 GeoSensor는 RFID, WSN, Web CAM, Digital Camera, CCTV, 텔레메틱스 단말 등에서 발생하는 다양한 데이터와 함께 직간접적으로 지리적 정보를 포함하는 데이터 스트림을 발생하는 센서들로, 공간적 특성에 기반을 둔 서비스를 제공한다[8].

이러한 DSMS는 최근 인터넷 환경에서 급속히 증대되는 24시간 무정지 서비스 요구되고 있고, 상업적으로 사용

되는 시스템에서 서비스 중단은 심각한 손실을 초래할 수 있기 때문에 최근 시스템들은 고가용성(high availability)의 서비스를 제공한다 고가용성이란 하드웨어, 소프트웨어 혹은 네트워크 등에 문제가 발생하여도 서비스를 지속적으로 제공할 수 있도록 해주는 기술을 말한다[3].

DSMS와 같은 실시간 데이터베이스 응용 시스템 분야나 임무 결정적(mission critical)인 응용 시스템을 다루는 산업 분야에서는 데이터의 성능이나 일관성보다 시스템의 가용성을 더욱 중요시 하는 경우가 많다[10]. 이와 같은 고가용성을 얻기 위하여 오류-극복(fail-over) 구조의 기본적인 방법인 이중화(replication) 기법이 지속적으로 연구되어 왔다. 그중에 대표적인 고가용성을 위한 이중화 알고리즘에는 Active/Active와 Active/Standby 알고리즘이 등이 있다.

먼저 Active/Active 알고리즘은 주 서버의 역할을 대체하는 시간이 매우 짧지만 여러 대의 기계를 동시에 가동해야 하므로 시스템 운영비용이 많이 들며, 또한 장시간 가동으로 인해 서버들의 고장 발생 가능성이 커지게 된다 반면에 특정시간에 주서버만 가동되는 Active/Standby 방법의 경우는 Active/Active 방법에 비해서 작업전이 시간은 길지만 그 대신 비용이 적게 들고 시스템이 하드웨어적인 고장 발생 가능성이 상대적으로 낮다.

Active/Active 와 Active/Standby 알고리즘 모두 가용

1) 본 연구는 건설교통부 첨단도시기술개발사업 - 지능형 국토정보기술혁신 사업과제의 연구비 지원(07국토정보C05)에 의해 수행되었습니다.

도를 높이기 위해 주(primary)서버와 그 역할을 대신 수행할 수 있는 보조(backup)서버를 두고 있다[9] 이때 주 서버의 모든 역할을 대신 수행 가능한 서버로 이용하기 위해선 주 서버의 모든 데이터가 동기화 되어야 하기 때문에 여분서버의 규모가 주 서버의 규모와 동일해야 한다. 이로 인해 한 대 규모의 고가용성 서버를 유지하기 위해서 2배의 하드웨어 비용이 소비되는 단점이 존재한다.

이러한 이중화 기법의 알고리즘들은 고가용성의 서비스를 제공하기 위해 항상 보조 서버는 주 서버의 역할을 전부 대신 할 수 있는 상태를 유지한다. 그렇기 위해서는 기본적으로 주 서버의 모든 데이터를 보조 서버도 똑같이 가지고 있어야 한다. DSMS 시스템은 실시간으로 입력되는 데이터를 처리하고, 결과를 실시간 서비스로 제공하기 때문에 모든 데이터를 디스크에 저장할 수 없다. 그렇게 때문에 메인 메모리에서 서비스 제공에 필요한 대부분의 데이터를 저장 및 사용하고 있다. 또한 이 데이터들은 한번만 수행되어 저장되는 데이터들이 아니라 등록 후 연속적으로 수행된다. 때문에 보조 서버에서 주 서버의 데이터를 중복적으로 저장하고, 그것을 연속적으로 수행해야 하기 때문에 보조 서버는 주 서버와 같은 규모로 만들어야 한다. 하지만 실제로 평상시에 사용하지 않는 보조 서버를 값 비싼 주 서버의 하드웨어 성능과 똑같이 구성한다면 기업 입장에서는 서버 한 대 규모의 서비스를 제공하기 위해 두 배의 하드웨어 비용이 지불된다.

본 논문에서는 공간 정보를 처리하는 DSMS인 GeoSensor Data Stream Management system 환경에서 고가용성을 유지하기 위해 기존의 이중화 기법의 단점인 보조 서버의 규모를 최소화 하는 동시에 고가용성 유지할 수 있는 선택적 데이터 동기화 알고리즘을 제안한다. 선택적 데이터 동기화 알고리즘은 보조 서버가 주 서버의 모든 데이터를 유지하고 있는 방식이 아닌 중요도가 높고 누적 데이터가 중요한 연산에 대한 데이터만을 유지하고 나머지 중요도가 낮거나 누적 데이터가 필요하지 않은 데이터들에 대해서는 메타 데이터들만을 유지한다. 이때 보조 서버는 주 서버의 데이터를 선택적으로 중복했기 때문에 보조 서버의 규모를 작게 유지할 수 있다. 그리고 실시간 데이터의 동기화 데이터가 줄어들어 따라 그에 따른 동기화 통신비용도 줄어드는 이점도 얻을 수 있다.

2. 관련연구

2.1 Active/Active 기법

기업의 운영자 입장에서 봤을 때, 대기 서버는 보통 때 아무 일도 안 하니 투자한 비용이 아깝다는 생각을 하게 된다. 어떠한 방식으로든지 대기 서버도 보통 때 다른 업무를 서비스하기를 원한다.

그래서 나온 방식이 Active/Active 알고리즘이다. Active/Active 알고리즘은 대기 서버 없이 두 개의 서버 모두 서비스를 제공하는 서버로 이용한다. 서버의 장애가 발생했을 때 두 개의 서버는 서로의 서비스를 대신해 수행할 수 있도록 설계되어 있다. 그 방법은 주 서버의 모든

데이터를 보조 서버가 가지고 있고, 보조 서버의 모든 데이터 또한 주 서버가 실시간으로 동기화되어 한쪽의 서버 장애시 살아있는 서버에서 모든 서비스를 제공하는 상호 대기 서버의 방식이다.

하지만 두 서버는 상대 서버의 장애 발생시 자신이 상대 서버의 모든 서비스를 제공해야 하기 때문에 하드웨어의 성능을 100%로 활용할 수 없고, 두 서버가 동시에 서비스를 제공해도 한 대의 서버 규모로 운용될 수밖에 없다. 여기에 여러 대의 기계를 동시에 가동해야 하므로 시스템 운영비용이 많이 들며, 또한 장시간 가동으로 인해 서버들의 고장 발생 가능성이 커지게 된다.

2.2 Active/Standby 기법

Active/Standby 알고리즘은 고가용성을 위한 이중화 기법의 가장 기본적인 기법이다. 구성은 간단하다. 주 서버가 서비스를 제공하고, 나머지 보조 서버는 주 서버의 장애 발생시에만 작동 하도록 되어 있다. Active/Standby 알고리즘 또한 장애 발생시 보조 서버가 데이터의 손실 없이 서비스를 제공하기 위해서는 계속적인 주 서버의 모든 데이터의 동기화가 필요하다.

Active/Active 알고리즘과 다르게 한 개의 서버만 운용되고 있지만, 나머지 보조 서버가 대기 하면서 주 서버의 모든 서비스를 제공할 수 있는 상태를 유지하고 있기 때문에 주 서버의 하드웨어 사용률을 100%로 사용할 수 있다. 이는 Active/Active 알고리즘의 두 대의 서버 모두 운용될 때의 성능과 비슷하다. 하지만 한 대의 서버만 운용되기 때문에 운영비용이 상대적으로 적게 들며, 서버 고장 발생가능성이 상대적으로 적다.

하지만 Active/Standby 알고리즘도 마찬가지로 주 서버의 장애를 모두 보완할 수 있는 주 서버와 동일한 규모의 보조 서버가 필요하다는 단점이 계속 존재한다.

3. DSMS 환경에서 선택적 데이터 동기화 기법

이중화 기법의 Active/Active 알고리즘과 Active/Standby 알고리즘 모두 주 서버와 보조 서버 간 데이터 동기화를 위해 많은 비용이 사용되고 있고, 그에 따라 보조 서버의 규모도 주 서버의 규모와 동일하게 구성된다. 이러한 단점을 보완하기 위해 선택적 데이터 동기화 기법을 제안한다.

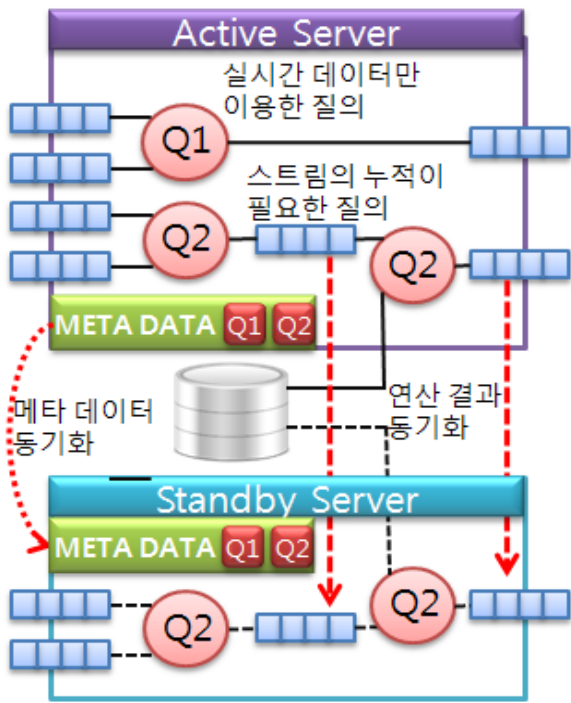
제안하는 선택적 데이터 동기화 기법은 서버 간 동기화에 필요한 데이터를 최소화해 동기화 비용 및 보조 서버의 규모를 줄이는 방법과 제안 기법을 바탕으로 한 장애 복구 기법을 설명한다.

3.1 선택적 데이터 동기화 기법

DSMS가 제공하는 서비스는 Input Stream에서 들어오는 실시간 데이터와 이 기종 데이터들을 사용해 질의를 등록해 해당 질의를 수행하는 방식으로 이루어져 있다. 이러한 특징 때문에 질의 종류를 크게 두 가지의 형태로 나눌 수 있다. 첫 번째로 집계 연산과 같이 실시간 데이터 이외에 이전에 저장된 데이터들을 사용하는 스트림의 누적 데이터 이용 질의가 있다. 예를 들어, 최근 1시간동안

의 인천 지역의 평균 온도를 구하는 질의가 그 예이다. 이 질의를 처리하기 위해서는 온도 데이터 스트림의 1시간 동안의 누적 데이터가 필요하다. 두 번째로는 One Time Query나 일반적인 프로잭션 연산과 같은 실시간 데이터들만 필요한 스트림의 실시간 데이터 이용 질의가 있다. 예를 들어, 지금 현재 인천 지역의 온도를 구하는 질의와 같이 온도 데이터 스트림의 누적 데이터가 필요 없이 실시간으로 들어오는 현재의 값만을 이용해 질의를 수행할 수 있다.

선택적 데이터 동기화 기법은 위의 두가지 질의의 특성에 따라 실시간 데이터와 데이터의 누적이 필요한 질의들에 대해서는 실시간 동기화를 통해 보조 서버로 저장 유지하고 나머지 연산들, 즉 실시간 데이터들만 요구하는 질의들에 대해서는 메타 데이터만 보조 서버와 동기화가 이루어진다.



----- 질의의 내부 연산 결과 복사
 질의의 Meta데이터 복사

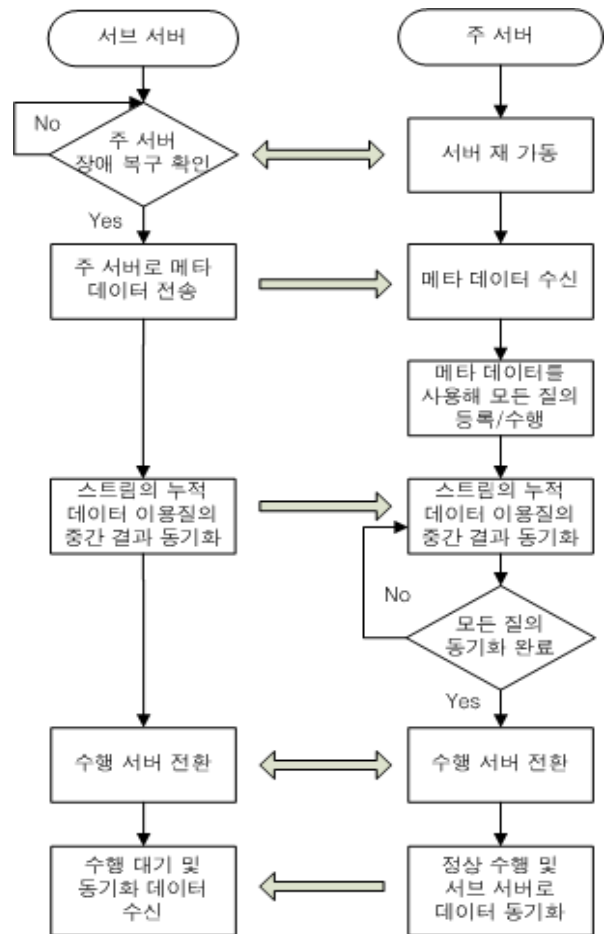
(그림 1) Active/Standby 서버 간 데이터 동기화

이러한 방법으로 보조 서버에서 적은 데이터와 연산만을 유지한다. 이때 주 서버의 장애 발생 시 주 서버의 역할을 보조 서버가 담당하게 된다. 클라이언트들은 서버의 장애를 느끼지 못하고 보조 서버를 이용해 서비스를 제공받는다. 이때 스트림의 누적 데이터 이용 질의는 그대로 보조 서버에 유지하고 있기 때문에 주 서버의 역할을 문제없이 수행할 수 있다. 하지만 스트림의 실시간 데이터 이용 질의의 경우에는 질의의 특성상 클라이언트의 요청이 들어온 후 질의가 수행되어도 서비스 제공에는 문제가 없다. 그렇기 때문에 스트림의 실시간 데이터 이용 질의는 메타 데이터만 유지하고 클라이언트의 요청이후 보조서버

에 등록된 후 서비스를 제공한다. 클라이언트의 요청이 끝나면 다시 해당 질의를 보조 서버에서 해제 후 메타 데이터만 유지한다.

3.2 고장복구 기법(가제)

선택적 데이터 동기화 기법은 주 서버의 장애 발생 시 보조 서버가 주 서버의 역할을 대신한다. 하지만 보조 서버의 규모는 주 서버보다 작기 때문에 장기적으로 주 서버의 역할을 대신 수행하기는 힘들다. 그래서 주 서버의 장애 복구 시 빠른 보조 서버에서 주 서버로의 전환이 필요하다. 하지만 서버의 규모 때문에 주 서버와 보조 서버는 내부에서 운용되고 있는 방식이 다르기 때문에 데이터들을 그대로 복사할 수 없다.



(그림 2) 장애 복구 후 수행 서버 전환

보조 보조는 주 서버의 장애복구를 인지하고 메타 데이터에 있는 모든 질의들에 대한 정보를 가장 먼저 동기화 한다. 그러면 주 서버는 메타 데이터를 통해 스트림의 누적 데이터 이용 질의와 스트림의 실시간 데이터 이용 질의 상관없이 모두 등록 시킨다. 그 다음 스트림의 누적 데이터 이용 질의들은 현재 실시간으로 들어오는 스트림으로 서비스를 제공할 수 없다. 필요한 누적 데이터가 들어오기 전까지는 정상 수행이 할 수 없기 때문에 빠른 복구를 위해 보조 서버에서 수행되고 있는 중간 스트림 결

과를 실시간으로 동기화 한다. 스트림의 실시간 데이터 이용 질의의 경우에는 실시간 데이터들만으로 서비스를 제공할 수 없기 때문에 다른 동기화 작업이 필요하다.

모든 스트림의 누적 데이터 이용 질의에 대한 동기화가 이루어지면 주 서버의 역할을 제대로 수행할 수 있는 상태가 된다. 그러면 바로 보조 서버에서 주 서버로 수행되는 서버를 전환해 서비스를 제공한다. 그리고 다시 정상적으로 스트림의 누적 데이터 이용 질의들은 실시간 데이터 동기화를 하고 스트림의 실시간 데이터 이용 질의들은 메타 데이터만 동기화하는 상태를 유지해 다시 발생할 수 있는 장애에 대비한다.

4. 성능평가

4.1 실험환경

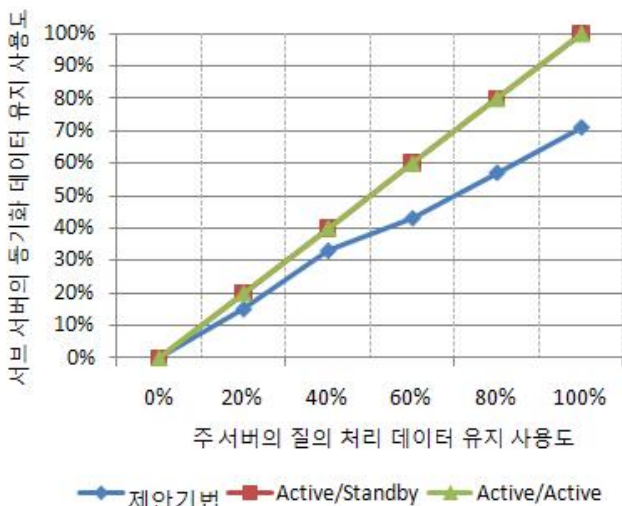
본 실험은 Fedora Core 8.0을 기반으로 수행했으며, 각 노드를 이루는 시스템 환경은 Intel(R) Pentium(R) 4 CPU 3.00GHz이고, 메인 메모리는 4GB인 장비 2대를 가지고 실험한다. 발생하는 데이터 셋은 센서 시뮬레이터(Sensor Simulator)으로 대체하였다. 센서 시뮬레이터는 제안 기법의 성능을 평가하기 위하여 개발된 어플리케이션으로 시스템 관리 툴의 일종이다. GIS 데이터는 TIGER/Line 2007 데이터로, Oracle에 구축하여 본 실험에 사용한다.

실험에 사용된 질의는 본문에서 제시한 두 가지 질의 타입의 비율을 스트림의 누적 데이터 이용 질의를 70%, 트림의 실시간 데이터 이용 질의를 30% 비율로 등록하였다.

4.2 성능평가

4.2.1 보조 서버의 데이터 유지량 측정

본 제안 기법은 주 서버와 보조 서버 간 동기화에 필요한 데이터를 줄임으로써 보조 서버의 규모를 최소화 했다.



(그림 3) 동기화 데이터 유지 사용도

Active/Active 기법과 Active/Standby 기법은 주 서버 데이터 유지 사용도와 동일하기 보조 서버도 유지 비용이 들었다. 하지만 제안 기법은 스트림의 누적 데이터 이용 질의의 등록 비율인 70%정도의 유지 비용으로 고가용성

을 유지할 수 있었다.

5. 결론

본 논문은 DSMS환경에서 고가용성을 유지하기 위한 방법중 이중화기법의 단점인 보조 서버 구축 비용증가의 단점을 보완한 선택적 데이터 동기화 기법을 제안 하였다.

제안 기법은 DSMS에 사용되는 질의를 두 가지 타입으로 나누고 누적 데이터가 필요 없는 질의에 대해서 서버 간 메타 데이터만 유지하는 방법으로 보조 서버의 크기를 줄일 수 있었다. 이에 따라 서버 간 데이터 동기화 비용도 감소하는 효과도 얻을 수 있었다. 그리고 주 서버의 장애 복구시 메타 데이터들로 주 서버에 질의를 등록한 후 보조 서버에서 수행되고 있는 질의들의 중간 결과 값을 동기화함에 따라 보조 서버에서 주 서버로 데이터 손실 없이 전환할 수 있었다.

향후 DSMS의 고가용성뿐만 아니라 고성능성도 만족시키면서 보조 서버의 규모를 줄일 수 있는 연구가 필요하다.

참고문헌

- [1] 이충호, 안경환, 이문수, 김주완, "u-GIS 공간정보 기술 동향," 전자통신동향분석, ETRI, 2007.
- [2] Brian B., Shivnath B., Mayur D., Rajeev M., and Jennifer W., "Models and Issues in Data Stream Systems," PODS, 2002.
- [3] Gregory F. Pfister, *In search of Clusters*, Prentice Hall PTR, 1998.
- [4] D.J.Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul and S. Zdonik, Aurora: A new model and architecture for data stream management" VLDB J. Vol 12 No. 2, pp. 120-139, 2003.
- [5] A. Arasu and et. al., "STREAM: The Stanford Data Stream Management System" <http://dbpubs.stanford.edu/pub/2004-20>, 2004.
- [6] J. Chen, D.J. DeWitt, F. Tian and Y. Wang, "NiagaraCQ: a scalable continuous query system for internet databases" Proc. of the ACM SIGMOD International Conference on Management of Data, pp. 379-390, 2000.
- [7] F. Reiss and J.M. Hellerstein, "Data triage: an adaptive architecture for load shedding in TelegraphCQ" Proc. of the International Conference on Data Engineering, pp. 155-156, 2005.
- [8] 정원일, 신승선, 백성하, 이연, 이동욱, 김경배, 이충호, 김주완, 배해영, "u-GIS 컴퓨팅을 위한 GeoSensor 데이터 스트림 처리 시스템," 한국공간정보시스템학회 논문지, 제 11권 제1호, 2009, pp. 9-16.
- [9] R. Buyya. High Performance Cluster Computing Volume Architectures and Systems, p.849, Prentice-Hall, 1999.
- [10] M. Wiesmann, F. Pedone, A. Shiper, B. Kemme, G. Alonso, "Understanding Replication in Databases and Distributed Systems", Proc. of the 20th International conference on Distributed Computing Systems, 2000.