

그룹화 기반의 퍼지 참조 기법을 이용한 분산 모바일 스트리밍 서비스

정택원*, 이종득**

요약

분산 모바일 시스템에서 혼잡제어와 끊김 문제 해결은 주요한 관심사로 떠오르고 있다. 본 논문에서는 분산 모바일 트랜잭션을 지원하는 시스템의 성능 개선을 위해 그룹화 기반의 퍼지 참조 스트리밍 기법을 이용한다. 이 기법은 모바일 클라이언트에서 제기된 트랜잭션 요청문제를 GS인터페이스를 이용하여 해결하도록 한다. 그리고 끊김 문제를 해결하기 위해 트랜잭션 읽기/갱신기법을 이용하여 스트리밍 서비스를 효율적으로 수행하도록 한다. 실험 평가 결과 제안된 기법이 다른 기법보다 효율적임을 보인다.

Distributed Mobile Streaming Service using Grouping-based Fuzzy Reference Scheme

Taeg Won Jeong*, Chong Deuk Lee**

Abstract

In distributed mobile systems, the congestion control and disconnection problems are current major issues. This paper proposes a grouping-based fuzzy reference streaming method to improve the performance of systems supporting distributed mobile transactions. The proposed method resolves transaction requests issued by mobile clients using the GS interface. In the paper disconnection problems are resolved efficiently using transaction read and update for improved streaming service. Experimental results show that the proposed method outperforms the other existing methods significantly.

Keywords : congestion control, disconnection problem, mobile transaction, fuzzy reference streaming

1. 서론

스트리밍 트랜잭션은 트랜잭션 데이터를 모바일을 통해 이동단말기로 전송해주며 요구되었던 시간을 최소한으로 줄여주기 위한 기법으로서 최근 들어 컴퓨터 처리기술과 모바일 서비스 기법 성능 향상으로 인하여 인터넷 방송 서비스, 근거리통신망 서비스, 원격교육 서비스 등의 여러 분야에서 널리 적용되고 있다[1][2][3][4][5].

스트리밍(Streaming)이란 모바일 상에서 트랜잭션 데이터를 실시간으로 스트림하기 위한 기법이며, 모바일을 통해서 처리하기 위한 업무정보, 세일즈맨의 명세서 요구정보 및 클라이언트 정보, 여행자의 예약 및 취소 그리고 멀티미디어 트랜잭션 콘텐츠를 모바일 상에서 서비스하기 위한 솔루션이다. 그러나 모바일 환경에서 낮은 대역폭, 빈번한 끊김 현상, 혼잡제어 그리고 사용자의 이동성 등은 모바일 트랜잭션 스트리밍의 성능과 신뢰성을 떨어뜨리고 있다[6][7].

이러한 문제를 해결하기 위하여 [8]에서는 록(lock) 관리 기법을 제안했으며, 이 기법은 언록(unlock) 기법을 이용하여 임의의 복제 사이트에서 트랜잭션을 처리하고 있다. 캥거루(kangaroo) 모바일 트랜잭션 기법[9]은 모바일 클라이언트 로밍 시에 발생하는 트랜잭션을 처리하기 위한

※ 제일저자(First Author) : 정택원

접수일:2008년 09월 26일, 완료일:2008년 12월 07일

* 전북대학교 응용시스템공학부

cdlee1008@chonbuk.ac.kr

** 전북대학교 응용시스템공학부(교신저자)

기법으로서 이 기법은 각기 다른 서브 트랜잭션을 서로 다른 기지국에 할당하여 트랜잭션 정보를 현재의 사이트에 자동으로 할당하여 서비스하는 기법이다. 그리고 TCOP(time-out-based Commit Protocol)[10]은 타임아웃(time-out)을 이용하여 커밋(commit) 프로토콜을 수행하고 있으며, 커밋 프로토콜에 따라 트랜잭션들이 스트리밍을 수행하게 된다. 현재 개발 중인 대표적인 스트리밍 프레임워크로는 ISO/IEC JTC1 SC29/WG11 MPEG 그룹에서 정의한 DSM-CC(Digital Storage Media -Command and Control)와 분산 객체 관리 구조에 대한 표준을 정의하고 있는 OMG(Object Management Group)의 CORBA AV 스트리밍 서비스 기법 등이 있다[5]. 그러나 이들 기법들은 트랜잭션 모드들에 읽기와 쓰기 권한이 모두 부여될 수 있어 끊임 문제를 찾기가 어려운 문제점이 있다. 따라서 본 논문에서 제안되는 그룹화 기반의 퍼지 참조 스트리밍은 스트림 데이터의 전송 경로에서 트랜잭션 응용들을 효율적으로 제어하고 서비스하기 위해 읽기/갱신 퍼지 참조를 기반으로 한 서비스 미들웨어 플랫폼으로 제공되며, 이를 통해서 송신측과 수신측 사이의 사용자 위주의 스트림 데이터 서비스가 가능하도록 제공한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해서 기술하며, 3장에서는 제안된 기법의 트랜잭션 그룹화 모델을 제안한다. 그리고 4장에서는 제안된 기법의 성능평가를 기술하고 끝으로 결론에 대해서 살펴본다.

2. 기존 연구

모바일 트랜잭션 서비스 프로토콜들은 모바일 노드에서 트랜잭션 서비스를 수행하고 있지만 대부분 프로토콜들은 혼잡제어와 끊임 문제 등에 대해서는 많은 문제점을 가지고 있다. 실제로 분산 모바일 환경에서 혼잡제어와 끊임 등과 같은 문제는 전송지연 및 데이터 손실 등의 QoS에 성능에 영향을 미치게 되며[11], 이러한 문제를 개선하기 위하여 퍼지 정보를 결합한 제어방법이 제안되고 있다[12]. 퍼지 참조 제어기법은 스트리밍이 수행되는 트랜잭션을 퍼지 개념 구조로 구조화하는 단계로서 트랜잭션을 퍼지

관계로 사상하여 그룹화를 수행하게 된다. 분산 모바일 환경에서 트랜잭션 처리와 관련된 스트리밍은 로그 축소 스트리밍, 경험적 복제 기반 스트리밍, 빈도 기반 복제 스트리밍으로 구분된다. 로그 축소 스트리밍(LRS : Log Reduction Streaming)[5,13]은 트랜잭션 데이터 대신에 레코드 S를 이용하여 로그 레코드 수를 줄여 스트리밍을 수행하는 기법이다. 이 기법은 두 개의 트랜잭션 t_1 과 t_2 즉 $D(t_1)$ -DMN과 $D(t_2)$ -DMN가 같고 $D(t_1)$ 과 $D(t_2)$ 가 서로 다를 때 레코드들을 하나의 레코드로 결합하여 로그 레코드 수를 축소하게 된다. 여기서 DMN은 트랜잭션 스트리밍이 수행되는 모바일 노드이다. 로그 축소에 의한 스트리밍은 읽기 트랜잭션 $|Q(S)|$ 와 갱신 트랜잭션 $|W(S)|$ 를 이용하여 스트리밍 서비스를 수행하게 된다. 이때 읽기 트랜잭션은 $|Q(S)|$ 는 $D(t)$ -DMC=S에 의해 수행되고 로그 레코드에 대한 갱신 $|W(S)|$ 는 $S=D(t) \cap DMC$ 에 의해 수행된다. 경험적 복제 기반 스트리밍(HRS : Heuristic Replica based Streaming)[9]은 스트리밍을 결정하기 위하여 집합-확장(Set-Expansion)알고리즘을 이용한다. 제안된 알고리즘에 경험적 데이터 집합 Sheu를 적용하여 각 단계를 탐색하면서 데이터 객체 S 중에서 $cost_a(SheuUS) < Cost_a(Sheu)$ 를 만족하는 최소 집합을 탐색하여 스트리밍을 수행하게 된다. 이후 집합 SheuUS를 만족하는 집합으로 확장을 수행하게 된다. 확장을 수행하는 과정에서 한 개 이상의 같은 크기를 가진 데이터 집합이 있으면 그들 중 하나는 임의로 선택되게 되며, 선택된 후에는 데이터 집합 S를 가진 레코드들은 스트리밍에서 삭제된다. 그리고 알고리즘에서 데이터 집합 S가 $|S-Sheu|$ 를 만족하는 스트리밍을 찾게 되면 알고리즘은 $cost_a(SheuUS)$ 또는 $cost_a(S)$ 를 수행하게 되며 empty가 될 때까지 반복 수행되게 된다. 그러나 이 알고리즘은 스트리밍을 수행할 최적의 데이터 집합을 찾기가 어렵다는 문제점이 제기되고 있다. 그리고 빈도기반 복제 스트리밍(FRS : Frequency based Replica Streaming)[14]는 접근 빈도를 분석하여 데이터 객체가 기지국 노드 PMC에 할당 될 것인지 아니면 PMC에 분할 될 것인지를 결정하여 스트리밍을 수행하는 기법이다. 이 기법은 트랜잭션 데이터 d_i 가 PMC에 복제될 경우 $update_aP(d_i)$ 의 갱신비용에 의해 수

행되며, 스트리밍 접근을 위한 갱신비용으로 $update_aP(di) = \sum |W(S)|$ 를 사용한다. 여기서 $di \in S$ 이고 $S \subseteq DBC$ 이다. di 가 스트리밍을 수행하는 접근비용은 $cost_aP(di) = \{\omega * update_aP(di)\} - \{read_aP(di)\}$ 이다. 이때 접근비용 $cost_aP(di)$ 가 di 에 접근될 때 비용함수와 가중치 ω 가 적용되며, $cost_aP(di) < 0$ 일 경우 di 를 PMC에 복제하게 된다. 그러나 이 기법은 이웃한 트랜잭션 객체가 많을 때 가중치가 적용되어 빈도에 따른 복제는 용이하나 읽기와 갱신기능이 순차적으로 부여될 수 있어 끊임 문제 해결이 어려운 문제점이 제기되고 있다.

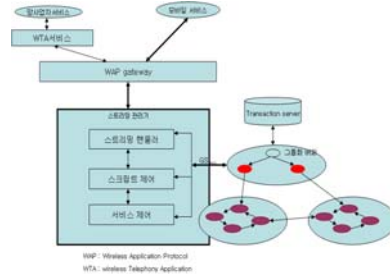
3. 그룹화 기반의 스트리밍 관리기

트랜잭션 서비스를 위한 그룹화 기반의 스트리밍은 그룹화 버퍼에 스트리밍을 수행할 트랜잭션 데이터 객체의 퍼지 참조에 의해 무선망에서 모바일 서비스를 수행하게 된다.

그룹화 버퍼는 모바일 프로토콜이 트랜잭션들 간의 통신 및 프로세스 간(IPC) 통신이 수행되도록 구성하며, 헤더(header)와 트레일러(trailer), 그룹화 스트리밍을 위한 읽기/갱신 트랜잭션 GS 인터페이스를 통하여 스트리밍 관리를 수행한다. 이때 그룹화 버퍼에서 트랜잭션 스트리밍을 요구하면 스트리밍 관리기는 GS인터페이스와의 상호작용에 의해 WAP 프로토콜을 통하여 모바일 서비스를 수행하게 된다. 이후 모바일 서비스가 요구되면 스트리밍 관리기는 스트림 서비스를 수행할 것인지, 갱신 및 삭제 기능을 수행할 것인지를 결정하게 되며, 모바일 서비스 상에서 트랜잭션 접근을 위한 스트리밍을 다음과 같이 가정한다.

- i) 모바일 서비스 노드(SN : Service Node)는 트랜잭션 스트리밍 서비스가 가능하며, $TDSN = \{td1, td2, \dots, tdn\}$ 은 모바일 클라이언트 MC(Mobile Client)에서 스트리밍 서비스가 가능한 n 개의 트랜잭션 데이터 (TD : Transaction Data) 집합이다.
- ii) 모든 모바일 노드들은 서로 독립이며, 독립된 모바일 상호 간은 스트리밍 관리기와 상호작용을 한다.
- iii) PMC는 개인 모바일 노드이고, SMC는 개

인 모바일 노드PMC에서 스트리밍이 수행되는 모바일 클라이언트이다. 이와 같은 가정에 의해 트랜잭션 서비스 수행을 위한 시스템 구성도는 (그림 1)과 같다.



(그림 1) 제안된 시스템 구성도

3.1 그룹화 버퍼

버퍼에서의 스트림 트랜잭션은 트랜잭션 큐로 구성되며, 큐에는 스트리밍을 수행할 트랜잭션들로 구성된다. 각 트랜잭션은 트랜잭션을 호출하여 읽기/갱신을 수행하게 되며, 각 트랜잭션 서비스블록은 서비스 데이터 블록을 가리키게 된다. 버퍼에서 스트림 트랜잭션은 트랜잭션 블록과 서비스 트랜잭션 데이터 블록에 의해서 수행되며, 서비스 트랜잭션 데이터 블록에서는 멀티캐스트가 수행되도록 스트리밍을 수행할 트랜잭션들을 여러 개의 포트에 구성하며, 트랜잭션들의 복사본을 여러 개를 만드는 대신에 트랜잭션들을 직접 할당하도록 퍼지 참조 값[15]을 이용한다. 퍼지 참조 값을 이용하여 스트림을 해제하면 트랜잭션 블록과 연관된 스트림이 해제되는 것이 아니라 스트림을 효율적으로 수행하기 위해 데이터 블록의 참조 값 필드를 감소시키게 된다. 이때 스트림 트랜잭션은 버퍼가 실행중일 때 동적으로 로드될 수 있으며 또한 언로드될 수가 있다. 버퍼에서 스트리밍 관리기와 스트리밍을 위한 트랜잭션 데이터 블록 구조와 그룹화 버퍼 블록 구조는 <표 1>과 같다.

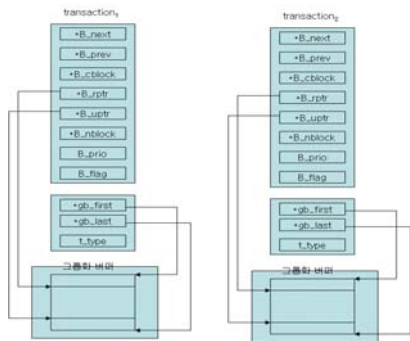
<표 1> 트랜잭션 데이터 블록 구조와 그룹화 버퍼 블록 구조
-트랜잭션 데이터 블록 구조-

구조	기능
*b_next	큐에 있는 다음 트랜잭션을 가리키는 포인터
*b_prev	큐에 있는 이전 트랜잭션을 가리키는 포인터
*b_cblock	현재 버퍼 블록을 가리키는 포인터
*b_rptr	읽을 트랜잭션을 가리키는 포인터
*b_uptr	갱신할 트랜잭션을 가리키는 포인터
*b_nblock	다음 버퍼 블록을 가리키는 포인터
b_prio	트랜잭션 스트림을 위한 우선순위
b_flag	스트림 헤드가 참조되는 퍼지 참조 플래그

-그룹화 버퍼 블록구조-

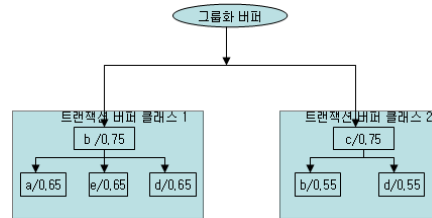
*gb_base	그룹화 버퍼의 첫 번째를 가리키는 포인터
*gb_last	그룹화 버퍼의 마지막을 가리키는 포인터
t_type	그룹화 버퍼에서의 서비스를 수행할 트랜잭션 타입

그룹화 버퍼 구조는 트랜잭션 데이터 블록구조와 그룹화 버퍼 블록구조로 구성되며, 스트리밍을 위한 각 트랜잭션은 하나 이상의 트랜잭션 블록으로 구성된다. 스트림 버퍼 관리를 위한 트랜잭션 데이터 블록구조는 *b_next, *b_prev, *b_cblock, *b_rptr, *b_uptr, *b_nblock, b_prio, b_flag로 구성되며, 그룹화 버퍼 블록구조는 *gb_base, *gb_last, t_type으로 구성된다. 그룹화 버퍼구조에서 스트리밍이 수행되기 위한 트랜잭션 버퍼 관리 구조는 (그림 2)와 같으며,



(그림 2) 트랜잭션 버퍼 관리 구조

트랜잭션 스트리밍이 수행될 때 그룹화 버퍼는 버퍼 클래스의 트랜잭션들을 퍼지 값으로 인코딩하게 된다. 인코딩 된 트랜잭션 td는 $\mu(tdi)$ 로 표현되며 $0 \leq \mu \leq 1$ 사이의 퍼지 속성 값을 가지게 된다[14]. 예를 들어 (그림 3)과 같은 그룹화 버퍼 구조의 트랜잭션 버퍼 클래스를 가정하자.



(그림 3) 트랜잭션 버퍼 클래스

(그림 3)의 트랜잭션 버퍼 클래스는 두 개의 트랜잭션 버퍼 클래스로 구성되어 있으며, 각각의 버퍼 클래스를 퍼지 속성 값으로 표현하면 다음과 같다.

$$F_{\text{그룹화버퍼}} = \{0.75/b, 0.75/c, 0.65/a, 0.65/e, 0.65/d, 0.55/b, 0.55/d\}$$

$$F_{\text{트랜잭션버퍼클래스1}} = \{0.75/b, 0.65/a, 0.65/e, 0.65/d\}$$

$$F_{\text{트랜잭션버퍼클래스2}} = \{0.75/c, 0.55/b, 0.55/d\}$$

이와 같은 퍼지 표현은 퍼지 참조 값에 따라 트랜잭션의 읽기/갱신 기능을 통하여 스트림에 로드가 발생될 때 로드 상황을 파악하여 스트리밍이 정상적인 상태로 수행되도록 하는 기능이다.

3.2 GS 인터페이스

모바일 환경에서 서비스 노드들은 원하는 원치 않는 시스템과의 연결이 끊어질 수 있다. 끊김 현상과 재연결 프로토콜들은 네트워크 단계에서 수행되며 이것은 응용도메인과는 독립적으로 수행되게 된다. GS인터페이스는 스트리밍 관리기와 그룹화 버퍼사이의 상호작용을 통하여 끊김 현상을 최소화하는 기능으로서 스트리밍을 요구하면 스크립트 제어모듈과 서비스 제어모듈과의 상호작용을 통해서 스트림을 생성할 것인지, 유지 및 갱신 기능을 수행할 것인지를 결정하게 된다.

3.3 스트리밍 핸들러

끊김 현상은 모바일 서비스에 영향을 주게 되며, 트랜잭션 스트리밍에도 영향을 주게 된다. 스트리밍 핸들러는 끊김 현상과 혼잡제어를 최소화하기 위한 기능으로서 읽기 트랜잭션과 갱신트랜잭션을 반복하여 스트리밍을 제어하게 된다. 읽기 트랜잭션은 그룹화 버퍼의 트랜잭션 버퍼 클래스에서 참조된 퍼지 값을 제어하게 되며, 이때 트랜잭션 버퍼 클래스에서 기준 값 이상의 관련도에 따라 트랜잭션이 갱신되게 된다. 갱신된 트랜잭션 클래스를 위해 본 논문에서는 α -cut[5]을 이용하여 갱신 클래스를 구성하며, 읽기 트랜잭션과 갱신 트랜잭션은 다음과 같이 정의된다.

(정의1) $R(TD) = \sum |F_{\text{그룹화버퍼}} TD_{SN}|$ 이다.

여기서 $read_GSR_U(TD)$ 는 트랜잭션 버퍼 관리구조에서 임의의 트랜잭션 데이터(td)를 접근하기 위한 읽기 트랜잭션 수이다.

(정의2) $U(TD) = F_{td}^{filtering} \{ \sum |F_{\text{그룹화버퍼}} TD_{SN}| \geq \alpha\text{-cut} \}$ 이다.

예를 들어 (그림 3)에서 $R(TD)$ 는 $F_{\text{그룹화버퍼}} = \{0.75/b, 0.75/c, 0.65/a, 0.65/e, 0.65/d, 0.55/b, 0.55/d\}$ 이고 $R(TD)$ 에 대해서 $F_{td}^{0.7}$ 즉 $\alpha=0.7$ 을 수행하면 $U(TD)$ 는 $\{0.75/a, 0.75/c\}$ 가 되게 된다.

그리고 그룹화 버퍼에서 두 트랜잭션들 사이의 퍼지 관련성이 있을 때는 최대 퍼지 관련성 F_{maxRel} 을 분석하며, 최대 관련성은 다음과 같이 정의된다.

(정의3) : $F_{\text{maxRel}} =$

$$\max_{A, B} \frac{1}{1 + |\mu(td1) - \mu(td2)|}$$

$F_{\alpha\text{-class}}(\{ \text{트랜잭션버퍼클래스1}[td] \}, \{ \text{트랜잭션버퍼클래스2}[td] \})$

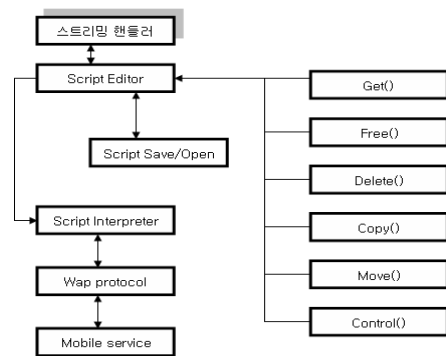
여기서 $f1$ 은 트랜잭션버퍼클래스1에서의 트랜잭션 데이터 td가 발생될 퍼지 값이며, $f2$ 는 트랜잭션버퍼클래스2에서의 트랜잭션데이터 td가 발생될 퍼지 값이다.

이후 스트리밍 제어기는 그룹화 버퍼와의 상호작용을 통해서 스트리밍을 로컬하게 서브할 것인지 아니면 읽기와 갱신 트랜잭션을 다시 수행할 것인지를 결정하게 된다. 만일 읽기 트랜잭션을 수행하는 트랜잭션 td가 그룹화 구조의 트랜잭션 버퍼 클래스를 참조하는 트랜잭션과 $D(t)$

$\cap F_{\text{트랜잭션버퍼클래스}} \neq \emptyset$ 이면 스트림의 생성, 유지 및 갱신기능은 중단된다. 그리고 $D(t) \cap F_{\text{트랜잭션버퍼클래스}} \neq \emptyset$ 이 아니면 $F_{\text{트랜잭션버퍼클래스}}$ 를 갱신된 퍼지 값에 따라 재배치되게 된다. 이와 같은 기능은 읽기와 갱신에 의한 연산을 동적으로 변화시켜 모바일 서비스를 혼잡상태와 비혼잡 상태로 구분하여 스트리밍을 안정적으로 유지시키기 위한 기능이다. 이러한 기능은 패킷손실을 줄여 끊김 현상이 감소되게 된다. 그러나 읽기/갱신과정이 $D(t) \cap F_{\text{트랜잭션버퍼클래스}} = \emptyset$ 을 반복 수행하면 패킷 손실률에 따른 혼잡상황 발생으로 인해 데이터 전송률은 감소되게 된다.

3.4 스크립트 제어

스크립트 제어모듈은 서비스를 수행하게 될 트랜잭션 데이터 스크립트들을 제어하는 모듈로서 스크립트 리스트와 스트리밍을 수행할 트랜잭션들을 b_flag 키 값에 따라 편집 저장하거나 불러오는 기능을 수행하게 된다. 즉 스트리밍 핸들러가 스트리밍 서비스를 요청하면 스크립트 에디터는 키 값에 따른 스크립트들을 분석하여 음성호, SMS, MMS 등 해당 트랜잭션들을 스크립트 함수를 수행한 후 WAP 프로토콜을 통해 모바일 단말기로 스크립트 명령어를 전달하게 된다. 이때 스크립트 제어는 스크립트 에디터를 통하여 스크립트 함수들에 의해 제어되며, 스크립트 제어 모듈은 (그림 4)와 같다.



(그림 4) 스크립트 제어 모듈

(그림 4)의 스크립트 모듈은 스트리밍 핸들러와 인터랙션을 수행하여 스크립트들을 에디팅을 수행하며, 에디팅을 수행하는 함수는 $Get()$,

Free(), Delete(), Copy(), Move()와 Control()함수로 구성되어 있다. 이와 같은 함수들은 스트리밍 핸들러의 트랜잭션들을 제어하여 트랜잭션 서비스를 효율적으로 수행하기 위해 사용된다.

3.5 스트리밍 서비스 제어

서비스 제어모듈은 스크립트 제어모듈과의 상호작용을 통하여 스크립트 해석기에 의해 트랜잭션 데이터가 스트리밍 될 수 있도록 제어하는 기능을 수행한다. 스트리밍 서비스 제어가 수행되면 모바일 단말기로부터 서비스되는 트랜잭션 데이터는 스트리밍 핸들러에서 그룹화 버퍼로 제어가 전송되고 전송된 트랜잭션 데이터는 차례대로 스크립트가 해석되어 스트리밍 수행 여부를 결정하게 된다.

이때 스트리밍 서비스 요청이 성공적으로 수행되면 그룹화 버퍼의 트랜잭션 버퍼 클래스에서 TranList를 탐색하게 되며, 서비스가 수행될 TranList를 탐색하는 과정은 다음과 같다.

TranList 탐색

$TD_{SN}=\{td_1, td_2, \dots, td_n\}$: 서비스될 스트리밍 트랜잭션 데이터, $td \rightarrow 0$ 으로 초기화

$F_{td}^{filtering}$: 퍼지참조를 위한 퍼지 필터링

$\sum |F_{그룹화버퍼}^{TD_{SN}}|$: 전체 그룹화 버퍼 탐색이 수행될 트랜잭션 데이터

TranListIndex : 0으로 초기화

if (TranListindex > $F_{td}^{filtering} \{ \sum |F_{그룹화버퍼}^{TD_{SN}}| \geq \alpha-cut \}$)

$S_{TranListIndex} = GetTransactionDataSet(TranList[TranListIndex]);$

$S_{new} = S_{td} \cup S_{TranListIndex};$

(GetTransactionDataSet(TranList[TranListIndex]), TranListIndex+1); //서비스를 수행할 트랜잭션 데이터 객체들을 재귀적으로 호출

}

else if(TranListindex < $F_{td}^{filtering} \{ \sum |F_{그룹화버퍼}^{TD_{SN}}| \geq \alpha-cut \}$) // 스트리밍 서비스 제어

{ F_{maxRel} ; // 트랜잭션 관련성 체크

$D(t) \cap F_{트랜잭션버퍼클래스} \neq \emptyset$; // TranList 탐색 중단

}

예를 들어 그룹화 버퍼에서 탐색과정을 수행한 트랜잭션 데이터 td와 트랜잭션 버퍼 클래스

와의 퍼지 표현이 <표 2>와 같다고 하자.

<표 2> 트랜잭션 데이터 td와 트랜잭션 버퍼클래스의 퍼지표현

	td ₁	td ₂	td ₃	td ₄	td ₅	td ₆	td ₇	td ₈	td ₉
트랜잭션 버퍼클래스1	1/1	0.7/1	0.7/1	0.7/1	0.7/1	0.58/1	0	0	0.58/1
트랜잭션 버퍼클래스2	1/1	0.7/1	0.4/1	0.58/1	0.58/1	0	0.68/1	0.58/1	0
트랜잭션 버퍼클래스3	1/1	0.3/1	0.5/1	0.7/1	0	0	0	0.82/1	0

이때 <표 2>의 트랜잭션 버퍼클래스1, 트랜잭션 버퍼클래스2, 트랜잭션 버퍼클래스3에 대해서 $\alpha-cut$ 을 0.6이상으로 하여 TranList 탐색과정을 수행하면 $F_{td}^{0.6} \{ \sum |F_{그룹화버퍼}^{TD_{SN}}| \geq 0.6-cut \}$ 에 의해 트랜잭션 버퍼클래스1의 서비스 제어는 {td₁, td₂, td₃, td₄, td₅}, 트랜잭션 버퍼클래스2의 서비스 제어는 {td₁, td₂, td₇}이고, 트랜잭션 버퍼클래스3의 서비스 제어는 {td₁, td₄, td₈}이 된다.

4. 성능평가

이 장에서는 시스템 성능과 효율성을 알아보기 위하여 본 논문에서 제안된 기법과 로그축소 스트리밍(LRS), 경험적 복제 기반 스트리밍(HRS), 빈도기반 복제 스트리밍기법(FRS)과 비교평가를 수행한다. 제안된 시스템 평가를 위해 무선 모델, PDA, USB 케이블, 노트북 등과 네트워킹을 수행하여 모바일 스트리밍 서비스를 수행하게 되며 접속 인터페이스 구조는 (그림 5)와 같다.



(그림 5) 접속 인터페이스

포트 인터페이스를 위해 PC의 serial port와 무선 데이터 포트 간의 접속은 RS-232 또는 I/F 포트를 이용하며 보율은 4800, 9600, 19200, 38400, 57600, 115200bps 등을 유지하도록 한다.

그리고 parity bit는 no parity bit, stop bit는 one stop bit와 two stop bit를 유지하며, flow control은 H/W flow control을 이용하여 스트리밍을 제어하게 된다. 이와 같은 시스템 환경에 의해 설정된 트랜잭션 데이터를 크기와 퍼지 참조값에 따라 평가를 수행한다. 트랜잭션 데이터와 퍼지 참조값에 따른 평가는 그룹화 버퍼 즉 트랜잭션 버퍼 클래스에 퍼지 참조가 적용되지 않았을 때와 트랜잭션 버퍼 클래스에 퍼지 참조가 적용되었을 때로 구분하여 혼잡 오류율과 서비스율 그리고 평균응답률에 대해서 평가를 수행한다. 끊김 현상과 혼잡 오류율을 평가하기 위해 본 논문에서는 다중 갱신 트랜잭션이 수행된 트랜잭션 버퍼 클래스를 생성하고 생성된 트랜잭션 버퍼 클래스에 대해서 각각의 기법을 적용한다. 또한 실험을 위해서 트랜잭션 버퍼 클래스에서 트랜잭션 데이터 수를 50,000개로 하고 퍼지 참조는 하나의 트랜잭션 버퍼 클래스에서 평균 갱신 트랜잭션 데이터 수와 같도록 세트시킨다. 그룹화 버퍼에 접근되는 트랜잭션 데이터 객체들은 트랜잭션 버퍼 클래스 그룹으로 구성하며, $M_i(i=1,\dots,n)$ 개의 트랜잭션 버퍼 클래스 그룹들이 식별될 때까지 그룹 탐색을 수행한다. 트랜잭션 버퍼 클래스의 트랜잭션 데이터들은 트랜잭션 버퍼 클래스의 트랜잭션 데이터 객체들과 중복될 수 있지만 두 개의 트랜잭션 버퍼 클래스 그룹들은 같을 수 없다. 트랜잭션들의 스트리밍 생성을 위하여 트랜잭션에 의해 접근 될 퍼지 참조 값 μ 를 먼저 생성한다. 이후 평가를 위해 트랜잭션 버퍼 클래스 그룹을 M 개로 나누고, M 개의 트랜잭션 버퍼 클래스 그룹들을 스트리밍을 수행하기 위한 트랜잭션 데이터 그룹의 클러스터로 분리한다. 이렇게 분리된 각각의 트랜잭션 버퍼 클래스에 대해서 α -cut을 적용하여 혼잡 제어율과 서비스율을 평가하고 끊김 현상을 알아보기 위해 평균 응답률을 수행한다. 스트리밍을 수행할 때 혼잡 오류율은 스트리밍의 성능평가에 중요한 영향을 미치게 되며, 혼잡 오류율이 작을수록 전송효율이 높게 된다.

혼잡 오류율은 스트리밍이 수행되는 트랜잭션들에 대한 품질척도로서 오류율 ψ 는 $N \times M$ 트랜잭션 버퍼 클래스에서 스트리밍이 수행되는 트랜잭션 데이터 객체 TDSN와 스트리밍이 수행되지 않는 TDSN'에 의해 수행되며, 혼잡 오류

율 ψ 은 다음과 같다.

$$\psi = \text{SQRT}(1/N \times M \sum_{x=0}^N \sum_{y=0}^M [TDSN - TDSN']^2)$$

이다.

이때 혼잡 오류에 의한 제어 품질은 서비스 요구자와 스트리밍 서비스 사이의 대역폭과 손실률에 의해 결정되며, 대역폭이 크고 손실률이 적으면 스트리밍 서비스율은 높아지게 된다.

서비스율이란 미리 정의된 트랜잭션 버퍼 클래스 그룹 내에서 관련된 트랜잭션들을 스트리밍 하기 위한 척도로서 서비스율 λ 는 다음과 같다.

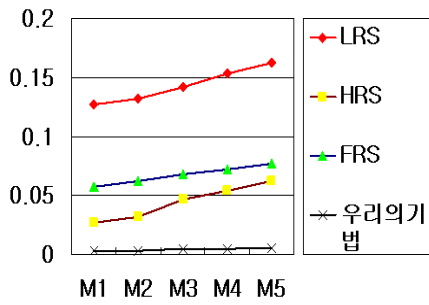
$$\lambda = 1 / (\sum_{i=1}^n [td_i \times (1 - \prod_{j=1}^i (1 - \mu_{i,j}))])$$

여기서 μ 는 트랜잭션 버퍼 클래스 그룹 내에서 관련된 트랜잭션을 참조할 퍼지 참조값이다. 그리고 평균 응답률이란 질의를 수행한 후의 스트리밍의 반응 정도를 측정하는 끊김 반응속도로서, 끊김 현상을 측정하기 위한 평균응답률은 다음과 같다.

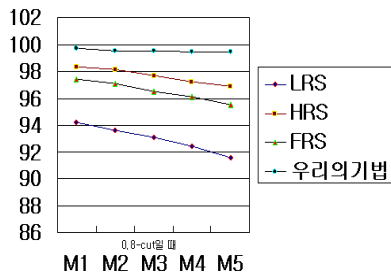
$$\text{평균 응답률} = \text{TDtime} (1 + 1/N) + \frac{td}{\text{트랜잭션버퍼클래스}} \times \text{TDsearchtime}$$

여기서 TDtime은 α -cut μ 를 만족하는 td_i 에 반응하는 응답시간이며, TDsearchtime은 해당 td 를 탐색하는데 걸리는 시간이다.

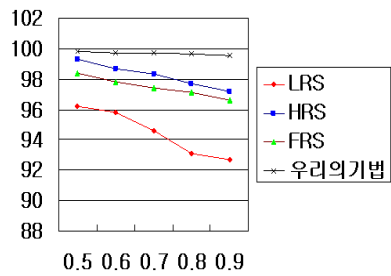
다음은 트랜잭션 데이터 객체수와 퍼지 참조에 의한 실험 평가로서 평가 트랜잭션 데이터 객체 수 50,000개를 10,000개 단위로 하여 M1, M2, M3, M4, M5의 5개의 트랜잭션 버퍼 클래스 그룹으로 설정하여 혼잡 오류율을 평가하였다. 그리고 퍼지 참조 값 0.7-cut를 적용하여 트랜잭션 클래스 그룹에 대한 평균 서비스율을 평가하였으며 M 개의 트랜잭션 버퍼 클래스에 0.5-cut, 0.6-cut, 0.7-cut, 0.8-cut, 0.9-cut로 하여 평균응답률을 평가하였다.



(그림 6) 혼합 오류율



(그림 7) 서비스율



(그림 8) 평균응답률

(그림 6)에서 보듯이 M1, M2, M3, M4, M5에서 M1, M2일 때는 혼합 오류율의 변화가 크지 않지만 버퍼 클래스가 증가될 때 제안된 기법의 오류 제어율이 효율적임을 알 수 있다. (그림 7)에서는 각각의 퍼지 참조값에 대해서 성능을 평가해야 하지만 본 논문에서는 0.7-cut 퍼지 참조값을 적용하여 서비스율을 평가하였으며 그 결과 제안된 기법의 서비스율이 향상됨을 알 수 있다. 그리고 (그림 8)에서는 M1, M2, M3, M4, M5의 5개의 트랜잭션 버퍼 클래스 그룹에 대해서 0.5-cut, 0.6-cut, 0.7-cut, 0.8-cut, 0.9-cut을 적용하여 평균응답률을 평가하였으며, 제안된 기

법의 성능이 향상됨을 알 수 있다.

5. 결론

분산 모바일 환경에서 혼잡제어 그리고 끊임 문제 등은 통신비용을 줄일 수 있는 중요한 기법이 되고 있다. 이러한 기법 개선을 위해 빈도, 로그축소, 클러스터링과 같은 여러 기법들이 제안되었으나 트랜잭션 스트리밍 서비스에 많은 문제점이 제기되고 있다. 본 논문에서는 트랜잭션 스트리밍 서비스에 있어서 발생하는 혼잡제어와 끊임 문제를 개선하기 위하여 그룹화 기반의 퍼지 참조 기법을 제안하였다. 제안된 기법은 그룹화 버퍼를 통하여 스트리밍 관리기와 상호작용을 수행하도록 하였으며, 상호작용을 통해서 모바일 스트리밍 서비스가 수행되도록 하였다. 그리고 모바일 트랜잭션 스트리밍을 위해 스트리밍 핸들러에 의해서 스트리밍을 수행할 트랜잭션들에 대해서 퍼지참조를 수행하도록 하였으며 이렇게 퍼지 참조된 트랜잭션들에 대해서 스크립트를 수행하여 서비스를 제어하도록 하였다. 그 결과 제안된 기법의 성능이 향상됨을 알 수 있다.

참 고 문 헌

- [1] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu, " A Frame work for On-Demand Classification of Evolving Data Streams," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 5, 2006, pp. 577-589.
- [2] Jun Yan, Benyu Zhang, Ning Liu, Shuicheng Yan, Qiansheng Cheng, Weiguo Fan, Qiang Yang, Wensi Xi, and Zheng Chen, " Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 3, 2006, pp. 320-333.
- [3] Thanana M. Ghanem, Moustafa A. Hammad, Mohamed F. Mokbel, Walid G. Aref, Ahmed K. Elmagarmid, " Incremental Evaluation of Sliding-window Queries over Data Streams," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.

19, NO. 1, 2007, pp. 57-72.

[4] Yufei Tao, Xiang Lian, Dimitris Papadias, and Mario s Hadjieleftheriou, " Random Sampling for Continuous Streams with Arbitrary Updates," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 19, NO. 1, 2007, pp. 96-110.

[5] ISO/IEC JTC1/SC29/WG11, Digital storage Media Command and Control, ISO/IEC 13816-6 N1 100, 1995.

[6] V. Kumar, M. Dunham, and A. Seydim, " TCOP-A Time Out-Based Mobile Transaction Commitment Protocol," IEEE Trans. Computers, vol.5, no. 10, 2002.

[7] A. P. Sistla, O. Wolfson, and Y. Huang, " Minimization of Communication Cost through Caching in Mobile Environments," IEEE Trans. Computers, vol. 51, no. 10, 2002.

[8] Like Gao, Xiaoyang Sean Wang, " Continuous Similarity-Based Queries on Streaming Time Series," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 10, 2005, pp. 1320-1332.

[9] M. Dunham, A. Helal, and S. Balakrishnan, " A Mobile Transaction Model That Captures both the Data Movement Behavior," ACM/Baltzer J. Special Topics in Mobile Networks and Applications, vol.2, 1997.

[10] Manghui Tu, Peng Li, Liangliang Xiao, I-Ling Yen, and Farokh B. Bastani, " Replica Placement Algorithms for Mobile Transactions Systems," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 7, 2006, pp. 954-970.

[11] T. Hara , " Replica Allocation Methods in Ad Hoc Networks with Data Updates," Mobile networks and Applications, vol. 8, 2003, pp. 343-354.

[12] I.x. wANG, A Course in Fuzzy System and Control, Eng-lewood Cliffs, NJ: Prentice-Hall, 1997.

[13] J. Jing, O. Bukhres, and A. Elmagarmid, " Distributed Lock management for Mobile Transactions," Proceedings of the 15th Int'l conf. Distributed Computing Systems, 1995.

[14] Bi-Ru Dai, Jen-Wei Huang, Mi-Yen Y도, and Minh-Syan Chen, " Adaptive Clustering for Multiple Evolving Streams," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 9, 2006, pp. 1166-1180.

[15] Jung-Hsien Chiang, Shihong Yue, and Zong-Xian

Yin, "A new fuzzy cover approach to clustering," IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 12, NO. 2, 2004, PP. pp. 199-208.

정택원



1979년 : 서울대학교 공과대학 전기공학과 졸업(공학사)
 1981년 : 서울대학교 대학원 전기공학과 졸업(공학석사)
 1991년 : University of Florida, Dept of EE 졸업(Ph.D)

1981-1983 : (주)선경종합건설
 1983-1998 : 한국전자통신연구원(ETRI)
 2008년~현재 : 전북대학교 공과대학 응용시스템공학부 전자통신전공 교수
 관심분야 : 무선통신, 이동통신 등

이종득



1983년 : 전북대학교 전산통계학과 졸업(이학사)
 1989년 : 전북대학교 대학원 전산통계학과 졸업(이학석사)
 1998년 : 전북대학교 대학원 전산통계학과 졸업(이학박사)

1992년~2002년 : 서남대학교 컴퓨터정보통신학과 교수
 2008년~현재 : 전북대학교 공과대학 응용시스템공학부 전자통신전공 교수
 관심분야 : 멀티미디어 통신, 무선통신, 모바일 성능평가 등