

# 블루투스를 이용한 데이터 일치성 모델의 성능 분석\*

한국회<sup>o</sup> 김찬우 김재훈 고영배  
아주대학교 정보통신전문대학원  
{justlike<sup>o</sup>, chanux, jaikim, youngko}@ajou.ac.kr

## Performance Analysis of Data Consistency Model Using Bluetooth

Kook-Hee Han<sup>o</sup> Chan-Woo Kim Jai-Hoon Kim Young-Bae Ko  
Graduate School of Information and Communication, Ajou University

### 요 약

인터넷의 보급과 활용이 증대되고 있고, 이동 통신에 대한 관심과 함께 분산 환경을 통한 컴퓨팅의 성능 향상이 중요시 되고 있다. 분산 환경에서 데이터 사용의 성능을 향상 시키기 위하여 데이터 중복 기법을 사용하는데, 서로 다른 복사본 사이의 데이터 일치성 유지가 필요하다. 본 논문에서는 데이터 일치성의 대표적인 모델을 Eager, Lazy, Periodic 세 가지로 구분하고, 각 모델의 특징을 알아보고 비용을 분석하여 사용자가 적절한 모델을 사용할 수 있도록 하였다. 또한 무선 컴퓨팅 환경에서의 데이터 일치성에 대한 테스트 베드를 블루투스를 이용하여 구축한 후 실제 성능을 측정하여 비용분석 모델의 유용성을 검증하였다.

## 1. 서 론

분산 환경을 통한 성능향상을 위한 방법으로 데이터 중복 기법이 사용된다. 데이터 중복은 데이터를 다른 노드에 복제 시켜 부하를 분산시키고, 결함 허용하며, 향상된 응답 시간을 제공 하는 등 많은 장점을 갖지만 데이터의 일치성이 필수 조건으로 충족되어야 한다. 데이터 일치성이란 데이터 사용에 있어서의 투명성을 유지 하는 것으로, 물리적으로 중복성을 갖으며 여러 곳에 위치하는 데이터를 사용자가 하나의 데이터를 이용하는 것처럼 처리 하는 것을 말한다. 데이터 일치성 모델에는 대표적으로 Eager, Lazy, Periodic 의 방법이 있다. 이것은 하나의 중복 데이터가 변경된 시점부터 다른 노드의 복사본으로 업데이트 되는 시점에 따라 결정된다.

주어진 특정 환경에서 데이터 일치성을 수행하기 위한 일치성 유지 모델을 선택함에 있어서, 환경에 대한 이해와 모델 선택에 영향을 주는 요소에 대한 정확한 분석이 필요하다. 본 논문에서는 특정 환경에서 데이터 일치성에 영향을 주는 변수에 따라 각 모델별 성능 차이를 비교하고, 블루투스를 이용하여 테스트베드를 구축한 후 실제 성능을 측정하여 비용분석 모델의 유용성을 검증한다.

## 2. 관련 연구

### 2.1 데이터 일치성

분산 환경에서 성능 향상을 위해서 데이터 중복의 방법이 많이 사용된다. 이것은 데이터의 복사본을 여러 노드에 두어서 사용하는 방법으로, 하나의 데이터 서버 노드에 집중되는 부하를 분산시킬 수 있으며, 데이터에 오류가 발생 하였을 때 다른 노드의 복사본을 통해 복구가 가능하고, 데이터를 사용하는 노드에서 가장 가까운 데이터 복사본에 접근 하게 되므로 응답시간이 향상된다. 그러나 각 복사본들이 별개로 존재하게 되므로 복사본 간에 데이터의 일치성이 이루어지지 않는다면, 데이터에 대한 신뢰성을 잃게 되며, 데이터 중복을 통해서 얻을 수 있었던 결함 허용도 불가능 하게 된다. 그러므로, 데이터 중복의 방법에는 필수적으로 데이터 일치성 작업이 필요하다. 데이터 일치성 작업을 통해 중복 데이터의 내용을 일치시켜 데이터의 사용자는 물리적으로는 여러 곳에 존재 하는 데이터이지만 하나의 데이터처럼 사용한다.

## 3. 데이터 일치성 모델의 비용 분석

### 3.1 데이터 일치성 모델

데이터 일치성 모델이란 한곳의 데이터 복사본의 업데이트

로 인해서 각 복사본으로 업데이트 된 내용에 대한 일치성 작업이 수행되는 시기와 방법에 따라 분류되는 방법으로, 업데이트 된 데이터가 다른 복사본으로 전파 되는 시점에 따라 일반적으로 Eager, Lazy, Periodic 모델이 있다. 다음은 각 모델의 특징과 응용의 예이다.

#### • Eager

한 복사본의 데이터가 수정 될 때마다 즉시 각 복사본으로 수정 된 내용이 전달되어 복사본의 데이터를 갱신 시키는 기법이다. 이 방법은 데이터가 수정 될 때마다 일치성 작업이 수행되기 때문에 항상 최신의 정확한 데이터로의 접근이 가능하다는 장점을 갖는다. 그러나 데이터의 수정 때 마다 일치성 작업을 수행하면 불필요한 오버헤드가 생기고, 복사본의 데이터 사용 요청률이 적을수록 이 오버헤드는 커지게 된다.

이 모델의 응용으로 주식 정보시스템과 종합 예매 시스템이 있는데, 주식정보시스템은 시간에 대한 변화가 빠르고, 이 변화에 데이터 업데이트 시간이 최소한이어야 하며, 데이터에 대한 민감도가 크다. 종합 예매 시스템은 모든 극장에 대한 예매와 연극 등 다양한 조건의 검색에 대한 요청이 다발적으로 발생하므로 예매 정보 변화에 대해서 데이터 일치성에 대한 신뢰도가 중요하다.

#### • Lazy

이 방식은 복사본의 데이터가 업데이트 되는 것에는 상관 없이, 복사본에 대한 사용자의 데이터 요청에 영향을 받는다. 한 복사본은 사용자의 요청에 대하여 다른 복사본과 데이터 일치성 작업을 수행한다. 이때 해당 복사본은 자신의 데이터 버전과 최신의 데이터 버전을 확인하여 데이터 일치성 수행 여부를 선택한다. 그러므로 불필요한 일치성 작업에 대한 오버헤드가 발생하지 않는 장점이 있지만, 해당 복사본이 최신의 데이터를 갖고 있지 않는 경우에 복사본을 업데이트 하게 되어서, 복사본의 일치성 작업을 하는 동안의 대기 지연 시간이 생기는 단점이 있다. 또한 데이터의 요청에 대한 일치성 작업을 하기 때문에 데이터의 변화에 대한 정보를 알 수 없다.

이 모델의 예로는 개인 휴대용 단말기에서의 개인 정보 데이터(PIM) 동기화와 웹을 이용한 택배 물품 조회 서비스가 있는데, 개인 정보 데이터의 동기화는 PDA 등의 이동형 기기와 컴퓨터간의 개인 데이터를 동기화 하는 것으로 일치성 작업에 의한 발생 빈도와 업데이트 빈도가 적다. 또한 복사본이 업데이트 될 때 일치성 작업을 요청할 비율이 낮다. 웹을 이용한 택배 물품 위치 조회 서비스는 배송중인 물건의 위치 조회할 수 있는 서비스로 물건의 위치 변화는 지속적이며, 이 정보에 대한 사용자의 요청은 매우 드물게 발생한다.

#### • Periodic

이 방식은 복사본의 데이터가 업데이트 되는 것에 대하여, 일정한 업데이트의 시간, 횟수를 주기로 복사본에 업데이트

\* 본 논문은 정보통신부에서 지원하는 정보통신 기초기술연구 지원사업으로 수행하였다.

된 데이터가 전파 되는 것이다. 이 방법은 설정된 업데이트 시간이나 횟수 등을 주기로 데이터 일치성 작업을 수행하게 된다. 이 방식은 주기에 따른 차이를 보이지만 데이터 일치성 수행을 적게 하여, 효율성을 증가 시킨다. 하지만, 사용자의 요청에 대한 최신의 데이터를 제공하지 않는 경우가 생기므로, 데이터에 대한 신뢰성이 없다는 단점이 있다.

이 모델의 예로는 기상변화 측정 시스템이 있는데, 인공위성을 통한 구름과 폭풍, 기압과 같은 기상 변화를 분석하는 경우 분석에 필요한 정보는 변화가 매우 빈번하며, 모든 변화에 대한 데이터 보다는 주기적인 정보의 변화가 중요하게 사용 되므로, 이 모든 변화의 기록 보다는 주기적인 변화에 대한 기록 및 분석이 필요하다.

3.2 데이터 일치성 모델 선택에 영향을 주는 요소

데이터 특성 및 활용에 따라 적절한 일치성 모델을 선택하는 것은 시스템 성능에 큰 차이를 준다. 그러므로 데이터의 특성과 그에 따른 영향을 주는 변수의 변화에 대한 정확한 분석으로 최적의 모델을 선택 하도록 한다. 각 변수는 모델에 따라 영향력이 다르고, 각 모델별로 영향을 주는 변수는 다르다. 그러므로 모든 상황에 대한 최적의 모델은 선택 할 수 없지만, 각 변수가 모델에 주는 영향에 대한 정확한 이해가 필요하다.

<표 1> 일치성 모델에 영향을 주는 요소

요 소	영 향
업데이트의 빈도수	업데이트의 빈도에 대하여 일치성을 수행하는 횟수가 증가하고, 이것에 의하여 일치성 비용의 차이를 준다.
데이터 크기	Data의 크기에 따라서 일정한 비율로 일치성 비용이 증가한다. 그러나 Lazy의 경우 최신 데이터를 확인하는 단위 패킷의 크기와 데이터의 차이에 따라 영향이 다르다.
데이터 전송시간	Eager나 Periodic은 정보 요청에 대해서 항상 같은 응답시간을 보이지만, Lazy의 경우 데이터 요청 시 복사본의 데이터 버전제에 따라 요청시간 차이를 보인다.
사용자요청 빈도수	사용자의 데이터 요청에 대해서 데이터 사용률이 변하며, 이것은 데이터 일치성의 빈도수에 영향을 준다.
데이터의 정확성대 대한 민감도	데이터의 요청에 대한 응답 데이터가 최신이 아닐 경우의 영향을 미치는 정도를 말한다.

그러므로 <표 1>에 제시된 각 요소를 참고로 각 일치성 모델에 대한 비용식을 구할 수 있다.

3.3 모델별 데이터 일치성 비용

각 모델에 대한 비용 계산은 서버-클라이언트간에 전송된 데이터량에 대한 패킷당 전송 비용으로 산출 할 수 있으며, 모델에 대한 비용 계산식은 <표 2>와 같다.

<표 2> 모델별 일치성 비용

모델	비용 종류	비용 분석
Eager	통신 비용	$D_d \times C_p$
	일치성 작업 대기시간 비용	$(1-\lambda_m)2C_s + \lambda_m(D_c + D_d)C_s$
Lazy	통신 비용	$(1-\lambda_m)2C_s + \lambda_m(1+D_d)C_s$
	일치성 작업 대기시간 비용	$(1-\lambda_m)2C_s + \lambda_m(1+D_d)C_s$
Periodic	통신 비용	$(D_d \times C_p) / T$
	부정확에 대한 비용	$S(T-1)$

$D_d$ : 데이터 패킷수  
 $C_p$ : 패킷당 비용,  $C_s$ : 패킷당 전송 시간 비용  
 $\lambda_m$ : 복사본의 데이터 변화율  
 $T$ : Periodic모델의 설정 주기(T 업데이트 마다 일치성 작업 수행)  
 $S$ : Periodic모델 데이터가 최신이 아닌 경우 민감도

Eager의 경우 비용은 항상 업데이트에 대해서 전송이 이루어지므로, 전송된 데이터 패킷에 대한 비용이다. Lazy는 사용자의 데이터 요구에 대한 버전 확인을 수행하며 이때 버전이 같을 경우 버전확인 요청 패킷과 버전일치 확인 패킷의 합

데이터량에 대한 비용이며, 버전이 다를 경우 버전확인 요청 패킷과 데이터 패킷의 합이 전송된 데이터량에 대한 비용이 된다. 여기에 버전이 다를 경우 소요되는 사용자의 일치성 작업에 대한 대기시간에 대한 비용이 추가된다. Periodic은 업데이트에 대한 일치성 작업의 빈도가 비용을 결정하게 되는데, 모델의 비용은 데이터의 업데이트 확률 당 Periodic모델에 설정된 일치성 작업 확률에 대한 업데이트 된 데이터 패킷 비용이며, 여기에 데이터 사용상의 민감도 비용을 추가 하게 된다.

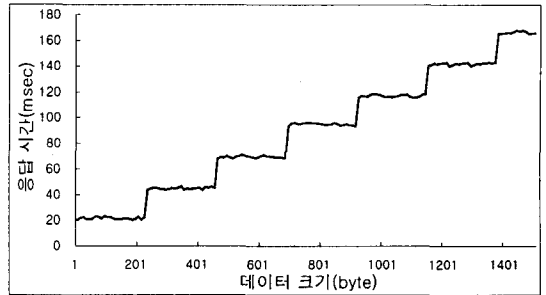
4. 구 현

각 모델에 대한 성능 테스트를 위해서 블루투스를 이용하여 다음과 같은 환경에서 일치성 테스트 프로그램을 만들었다.

- 통신기기: 블루투스 (CSR사의 MicroSira)
- 운영체제: Windows2000
- 컴파일러: MS Visual C++ 6.0

테스트 환경의 비교는 각 복사본에서의 데이터 일치성 작업에 소요되는 비용으로 하였다. 해당 복사본이 데이터를 수정하는 것에 대한 것은 고려하지 않았는데, 이것은 모든 모델이 데이터에 대한 업데이트를 수행할 확률은 같으며, 데이터를 수정 할 경우 데이터 일치성 작업이 먼저 수행 되어야 하므로, 이 데이터 일치성 작업에 대한 비용만을 고려 하도록 하였다. 고정 요소는 기본 최소 패킷의 크기가 256byte이고, 500초 동안의 패킷의 전송량을 비용으로 계산 하였다. 복사본의 데이터의 수정은 3초 주기를 기본으로 하는 시스템이며, 데이터의 요청 간격은 요청률( $\lambda$ )에 대한 지수 분포로 가정 하였다.

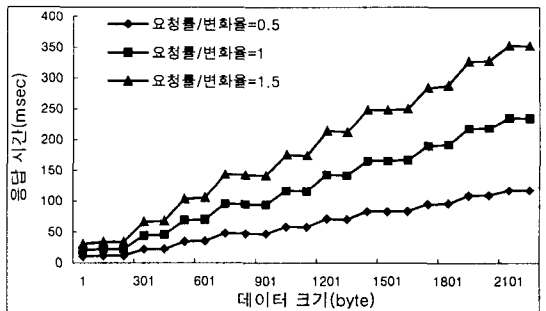
4.1 데이터의 크기에 따른 응답시간.



<그림 2> 데이터크기에 따른 응답시간

<그림 2>는 Lazy모델에서의 데이터 크기에 따른 응답시간의 변화를 나타내는 그래프이다. Lazy의 경우 버전 체크 후 물리엔 데이터 일치성 작업을 하기위해 최신의 복사본으로부터 데이터를 업데이트 하는 시간이 필요하다. 이 업데이트 시간이 사용자측에 대한 응답시간이 되며, 환경의 특성상 데이터의 최소 단위는 256byte이므로, 그래프는 256byte 단위로 응답시간이 증가 하는 계단형 그래프를 보인다.

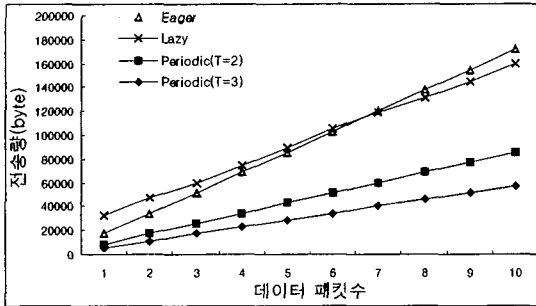
4.2  $\lambda_r / \lambda_m$ 에 변화에 따른 데이터의 전송량



<그림 3> 요청률/변화율의 변화에 따른 데이터의 전송량

<그림 3>은 데이터 변화율을  $\lambda_m$ 라 하고, 데이터 요청률을  $\lambda_r$ 라 할 때, 데이터 크기에 따른 응답 시간을 데이터 변화율에 대한 데이터 요청률( $\lambda_r / \lambda_m$ )의 변화에 따라 보여준다. Periodic 모델에서의 비용 요소(Eager의 경우  $\lambda_r / \lambda_m$ 의 값은 항상 1이며 이때 전송량은 430byte)로 Periodic의 요청률이 변화율보다 작으면, 최신의 복사본으로부터 데이터 일치성 작업하는 데이터의 전송량은 적어진다. 그러나 이때 데이터에 대한 신뢰성 문제가 발생하게 되며, 요청률이 변화율보다 크다면, 데이터 신뢰성 문제는 향상되지만, Eager 모델 보다 성능은 감소한다.

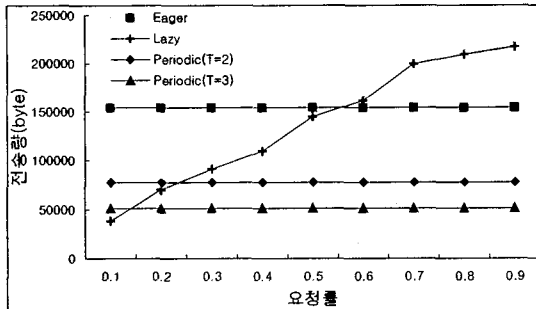
4.3 데이터의 크기에 따른 모델별 전송량을 비교.



<그림 4> 데이터 크기에 따른 모델별 전송량

<그림 4>는 요청( $\lambda_r$ )이 0.5이고, 서버에서 데이터를 66번 업데이트 할 때(약 500sec)의 데이터 크기에 따른 모델별 평균 데이터 전송량을 나타낸다. Eager 방식은 데이터량이 증가할수록 전송량이 일정한 비율로 증가하며, Periodic 방식은 주기(T)가 길수록 효율적이지만 신뢰성 비용이 추가 되어야 한다. Lazy 방식은 Eager 방식에 비해 좀더 완만한 기울기로 증가하므로, 데이터 패킷의 크기가 커질수록 Eager에 비해서 효율적이다. 이것은 Lazy 방식의 특성상 복사본의 데이터 버전이 같은 경우 버전확인과 버전확인 메시지 패킷만으로 일치성 작업이 완료 되기 때문이다. 그러나 데이터의 크기가 단위 패킷의 크기 정도로 가장 작을 경우는 Eager보다 비용의 차이가 크다.

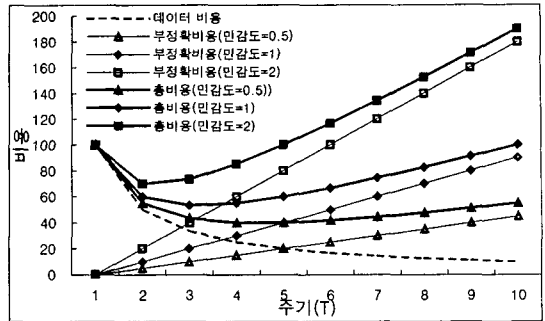
4.4 데이터 요청률에 대한 모델별 데이터 전송량



<그림 5> 요청률에 대한 모델별 데이터 전송량

<그림 5>는 데이터 크기/패킷 크기가 9인 환경에서 서버가 데이터를 66번 업데이트 할 때의 사용자의 요청률( $\lambda$ )에 따른 모델별 평균 전송량을 나타낸다. Eager나 Periodic의 경우 데이터 요청 확률과 무관하게 데이터의 업데이트와 주기( $T=1/\lambda$ ) 따라 전송 비용이 결정 된다. 그러므로, 데이터 요청 확률과는 무관하게 Eager와 Periodic은 항상 일정한 일치성 비용이 소모 된다. 그러나 Lazy의 경우 요청 확률에 따라 비용이 증가하며, 확률이 높아 질수록 기울기가 좀더 완만해 지는데, 이것은 요청 확률이 높아질수록 요청시 버전이 최신 버전일 확률이 증가 하기 때문이다.

4.5 민감도에 대한 가상의 비용 측정



<그림 6> 민감도에 대한 비용 측정

<그림 6>은 Periodic에서의 데이터 비용과 부정확한 데이터로 인한 가상 손실 비용의 합을 측정한 그래프이다. 주기의 데이터 비용을 100으로 환산하고, 이때 데이터 비용과 주기 1에서의 부정확한 데이터로 인한 민감도 비율이 0.5, 1, 2일 때의 부정확 비용 그래프와 부정확 비용과 데이터 비용을 합한 총 비용 그래프이며, 주기가 1인 경우의 Periodic의 비용 그래프는 Eager 그래프와 일치 한다. 총 비용 값의 그래프를 보면, 민감도의 값의 크기에 따라 최소 비용이 되는 주기는 각각 다르다는 것을 알 수 있다. 그러므로, Periodic 모델에서 부정확한 데이터로 인한 비용의 크기에 따라 설정될 최소 비용 주기는 다르다는 것을 알 수 있다.

5. 결론

데이터 일치성에 대한 모델 Eager, Lazy, Periodic에 대한 특징과 분산 시스템 환경에서의 데이터 일치성을 수행 할 때의 해당 시스템에서 영향을 주는 요소들을 제시 하였고, 실제 응용에 프로그램의 구현을 통해서 데이터 일치성 수행에 영향을 주는 요소와 그에 따른 성능차이를 비교 하였다. 해당 시스템에 맞는 데이터 일치성 모델을 적용시키기 위해서는 각 모델의 특성과 해당 시스템에 대한 정확한 분석이 필요하다. 그렇지 못하다면 Eager 모델에 의한 불필요한 오버헤드와 네트워크 부하, Lazy 모델에서의 데이터 일치성 작업의 지연시간으로 인한 손실, Periodic 방식에 의한 부적절한 데이터의 접근으로 문제점을 야기 시킬 수 있다. 그러므로 적절한 응용 모델의 적용으로, 시스템 전체의 성능 향상과 적절한 데이터를 사용자에게 접근 가능하게 하는 효율적인 데이터 중복 기법의 분산 시스템을 가능하게 할 데이터 일치성 모델의 선택은 중요하다 하고 할 수 있다.

참고 문헌

- [1] Atul Adya and Barbara Liskov, "Lazy Consistency Using Loosely Synchronized Clocks," In Proceedings of the ACM Symposium on Principles of Distributed Computing, August 1997.
- [2] D. Barbara and H. GarciaMolina, "Replicated Data Management in Mobile Environments: Anything new under the sun?," In Proceedings of the IFIP Conference on Applications in Parallel and Distributed Computing, April 1994.
- [3] Venkata Duvvuri, Prashant Shenoy and Renu Tewari, "Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web," Master's Thesis, Univ. of Massachusetts, June 1999.
- [4] Yuri Breitbart and Henr F. Korth "Replication and Consistency: Being Lazy Helps Sometimes," PODS 1997: p173-184.
- [5] Bluetooth Lab., <http://www.Bluetoothlab.com/>
- [6] CSR, <http://www.csr.com/>