

그리드 데이터베이스에서 전송비용 최적화를 위한 복제본 갱신 전파 기법

장용일[†], 백성하^{**}, 배해영^{***}

요 약

본 논문에서는 복제본을 갱신할 때 동시성을 유지하기 위해 발생하는 질의 전달비용을 최적화하는 갱신 전파 기법을 제안한다. 그리드 데이터베이스에서는 일반적으로 데이터를 처리 성능과 가용성 향상을 위해 서로 다른 위치에 복제하여 저장한다. 복제본이 수정되는 경우 다른 복제본으로 최신 데이터를 전송하고 반영하는 갱신 전파 과정이 수행된다. 갱신 전파 과정은 질의 전파와 로그 전파 기법으로 구분된다. 그러나 각각의 기법은 갱신 질의의 특성 및 환경에 따라 전송비용이 증가하는 문제를 갖는다. 이러한 문제의 해결을 위해 제안 기법은 기존의 두 가지 갱신 전파 기법의 비용 모델을 정의하였으며, 이를 통해 갱신 질의의 적용 환경에 따라 나타날 수 있는 세 가지 클래스를 얻는다. 또한 각각의 클래스에 따라 산출되는 비용 모델을 기준으로 최적의 갱신 전파 기법을 선택할 수 있는 갱신 전파 기법을 설계한다. 제안 기법은 성능평가를 통해 질의의 특성과 환경이 변화되는 환경에서 최소의 전송비용으로 최적의 성능을 유지함을 보인다.

Replica Update Propagation Method for Cost Optimization of Request Forwarding in the Grid Database

Yong-Il Jang[†], Sung-Ha Baek^{**}, Hae-Young Bae^{***}

ABSTRACT

In this paper, a replica update propagation method for cost optimization of request forwarding in the Grid database is proposed. In the Grid database, the data is replicated for performance and availability. In the case of data update, update information is forwarded to the neighbor nodes to synchronize with the others replicated data. There are two kinds of update propagation method that are the query based scheme and the log based scheme. And, only one of them is commonly used. But, because of dynamically changing environment through property of update query and processing condition, strategies that using one propagation method increases transmission cost in dynamic environment. In the proposed method, the three classes are defined from two cost models of query and log based scheme. And, cost functions and update propagation method is designed to select optimized update propagation scheme from these three classes. This paper shows a proposed method has an optimized performance through minimum transmission cost in dynamic processing environment.

Key words: Grid Database(그리드 데이터베이스), Data Replication(데이터 복제), Replica Synchronization(복제본 동기화), Update Propagation(갱신 전파)

※ 교신저자(Corresponding Author) : 배해영, 주소 : 인천광역시 남구 용현동 253(402-751), 전화 : 032)860-8712, FAX : 032)862-9845, E-mail : hybac@dblab.inha.ac.kr
접수일 : 2006년 8월 8일, 완료일 : 2006년 10월 23일

[†] 준회원, 인하대학교 컴퓨터정보공학과 박사과정
(E-mail : yijang@dblab.inha.ac.kr)

^{**} 인하대학교 컴퓨터정보공학과 석사과정
(E-mail : bshzeratul@dblab.inha.ac.kr)

^{***} 정회원, 인하대학교 컴퓨터정보공학과 교수

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음

1. 서론

그리드 데이터베이스(Grid database)는 그리드 컴퓨팅 환경에서 분산된 데이터의 효율적 처리와 사용을 위한 데이터베이스 관리 시스템이다. 그리드 데이터베이스는 기존 분산 데이터베이스 시스템의 기능을 기본적으로 포함한다. 또한, 그리드 데이터베이스는 대용량 자원을 공유하며 고속 연산을 지원하고, 그리드 데이터베이스를 구성하는 각 노드는 데이터를 처리 성능과 가용성 향상을 위해 서로 다른 위치에 복제하여 저장하며, 데이터 연합(Federation), 데이터 변환, 데이터 배치(Allocation), 데이터 통합 등의 기능을 제공한다[1].

그리드 데이터베이스를 구성하는 각 노드는 수많은 데이터들을 포함하고 있다. 모든 데이터는 어플리케이션에 따라 그 형태와 목적이 다르지만, 가용성과 성능의 향상을 위해 다른 노드에 복제본(Replica)을 구성하게 된다. 하나의 데이터가 여러 복제본을 가지고, 최적의 성능을 낼 수 있는 노드에 각각 배치된다[2].

일반적으로 복제본이 갱신이 되면 다른 노드의 복제본으로 수정된 정보를 보내는 갱신 전파 기법을 통해 최신 정보로 변경하는 동기화 과정이 이루어진다. 갱신 전파 기법은 크게 질의 전파 기법과 로그 전파 기법으로 구분할 수 있다. 질의 전파 기법은 사용자로부터 입력된 갱신 질의를 다른 복제본으로 동일하게 전송한다. 전송된 질의는 모든 노드에서 동일한 처리 과정을 거친다. 로그 전파 기법은 처음 입력된 갱신 질의에 의해 갱신된 결과 데이터를 이용한다. 이때 갱신된 데이터를 기반으로 목적 데이터에 대한 연산 집합인 논리 로그를 생성한다. 생성된 논리 로그는 다른 복제본이 있는 노드로 전송되어 목적 데이터에 대한 갱신 연산을 수행한다[3,4].

질의 전파 기법은 갱신 질의의 크기가 작기 때문에 네트워크를 통한 전송에 부담이 없다. 반면에 갱신 질의에 조건절이 포함되어 다른 노드의 데이터와 조인 연산이 필요한 경우 네트워크 비용이 증가한다. 또한 질의가 복잡할 경우 처리 시간이 증가하여 전체적인 처리 비용이 증가하게 된다. 로그 전파 기법은 한 노드에서만 질의 처리 과정을 수행하므로 질의의 복잡도가 전체적인 처리 성능에 미치는 영향은 아주 작다. 그러나 갱신된 데이터의 크기가 커질 경우 네트워크 비용이 상대적으로 증가하는 문제가 발생한다.

또한 일반적인 응용 환경은 동적으로 변화하기 때문에 갱신 질의의 형태도 계속 바뀐다. 이러한 환경에서 하나의 갱신 전파 기법만을 사용하게 되면 질의의 특성 또는 환경에 따라 전송비용이 증가하게 된다.

제안 기법은 기존의 두 가지 갱신 전파 기법의 복제본 갱신 비용 함수를 정의하고 각각의 비용 특성을 보인다. 이를 기반으로 다양한 응용 환경에서 두 가지 갱신 전파 기법을 병용하여 사용하였을 때 나타날 수 있는 세 가지 분류를 나눈다. 그리고 처음 정의한 비용 함수를 기반으로 각각의 분류를 구할 수 있는 분류 함수를 생성한다. 분류 함수는 주어진 환경이 어떠한 분류에 속하는지 구함으로써 최적의 갱신 전파 기법을 선택할 수 있다. 마지막으로 분류 함수를 적용하여 그리드 데이터베이스에서 최적의 비용 효율을 가지는 갱신 전파 기법을 설계한다.

제안 기법은 성능평가를 통해 본 논문에서 정의한 분류 함수가 올바르게 동작하는 것과 한 가지 갱신 전파 기법만을 사용하였을 경우에 비해 갱신 질의 처리 과정이 항상 최적의 성능을 유지함을 보인다.

본 논문의 구성은 다음과 같다. 2장은 관련 연구로 기존의 그리드 데이터베이스에 대한 연구와 두 가지 복제본 갱신 전파 기법에 대해 설명한다. 3장에서는 본 논문에서 제안하는 복제본 갱신 전파 기법에 대해 설명하고, 4장에서 제안 기법에 대한 성능 평가를 수행한다. 마지막으로 5장에서 결론을 기술한다.

2. 관련 연구

본 장에서는 그리드 데이터베이스의 기능 및 요구 사항들과 이에 따른 기존 연구들을 설명하고, 기존의 복제본 갱신 전파 기법을 설명한다.

2.1 그리드 데이터베이스

기하급수적으로 늘어가는 방대한 데이터들의 효율적 성능향상을 위한 노력은 지속적으로 이루어져 왔다. 이러한 노력의 일환으로 분산 데이터베이스와 데이터베이스 클러스터 그리고 그리드 데이터베이스에 대한 연구가 계속 되었다.

분산 데이터베이스는 컴퓨터 통신망을 이용하여 여러 개의 지역데이터베이스를 논리적으로 연관시킨 통합된 데이터베이스이다. 물리적으로는 분산되고 논리적으로는 집중되어 있는 형태의 구성으로 단

순한 연결이 아닌 각 데이터베이스가 서로 관여를 하는 연결구조이다. 분산 데이터베이스의 장점은 데이터를 분산 배치하므로 장애에 대한 대비에 강하고 다수의 이용자가 대규모의 데이터베이스를 낮은 비용으로 공유할 수 있는 점이다. 분산 데이터베이스는 중앙 집중형 데이터베이스 보다 저비용으로 구성이 가능하며, 확장성 및 가용성에 장점이 있다[5].

분산 데이터베이스가 데이터의 공유, 유통 및 투명성에 초점이 맞추어 있다면, 데이터베이스 클러스터는 고속의 네트워크 연결하여 데이터 처리의 성능, 신뢰성, 가용성을 향상 시키는데 목적이 있다. 데이터베이스 클러스터에서는 작업 부하를 분산시키기 위하여 부하 분산 기법을 사용하여 데이터베이스의 작업량을 균형 있게 분배한다.

그리드 데이터베이스는 그리드 컴퓨팅 환경에서 분산된 데이터의 효율적 처리와 사용을 위한 데이터베이스 관리 시스템이다. 그리드 데이터베이스는 기존 분산 데이터베이스 시스템의 기능을 기본적으로 포함하는 동시에 통합, 자동화 및 가상화 등의 기능을 제공한다. 그리드 데이터베이스의 주요한 기능으로는 대용량 자원에 대한 관리와 고속 연산, 그리고 데이터 복제를 통한 가용성 향상에 있다. 그리드 데이터베이스를 구성하는 각 노드는 데이터를 처리 성능과 가용성 향상을 위해 서로 다른 위치에 복제하여 저장하며, 데이터 연합, 데이터 변환, 데이터 배치, 데이터 통합 등의 기능을 제공한다[6].

2.2 복제본 동기화를 위한 갱신 전파 기법

서비스의 가용성과 확장성을 위해 그리드 데이터베이스에서는 하나의 데이터를 다른 노드에 복제한

다. 복제된 데이터는 복제본(replica)으로 불린다. 복제본이 많을수록 보다 많은 사용자가 동시에 데이터를 읽을 수 있다. 하지만 갱신연산의 경우 복제본이 많을수록 소요되는 시간이 길어진다[7].

일반적으로 복제본 갱신을 위한 두 가지 기법으로서 질의 전파 기법과 로그 전파 기법을 사용한다[3,4].

질의 전파 기법은 사용자의 갱신 질의를 다른 노드로 전송하여 갱신 과정을 수행한다. 질의 전파 기법에서는 동일한 복제본을 가지는 모든 노드로 갱신 질의를 보내어 해당 노드에서 데이터가 직접 갱신된다. 만일 질의의 내용이 다른 데이터와의 비교에 의해 갱신 대상을 결정하는 질의, 즉 조인(join) 질의라면 질의 전파 기법에서는 모든 노드에서 다른 데이터와의 조인 연산이 이루어진다. [그림 1]은 사용자의 갱신 질의가 질의 전파 기법을 통해 처리되는 과정을 나타낸다. 노드 N₂로 입력된 갱신 질의는 우선 동일 복제본을 갖는 다른 노드로 전달(forwarding)된다. 이후 갱신 질의는 각각의 노드에서 조인 과정을 거친 후 갱신 연산이 처리된다.

로그 전파 기법은 갱신된 데이터를 논리 로그의 형태로 다른 노드로 전송한다. 갱신 질의가 입력된 첫 번째 노드에서 갱신 과정을 수행한 후 갱신 데이터에 대한 논리 로그를 생성하고, 이를 다른 노드로 전송한다. 로그 전파 기법에서는 질의 전파 기법에서와 달리 첫 노드에서만 모든 갱신 과정을 처리하고 나머지 노드에서는 논리 로그의 반영 비용만 소요된다. [그림 2]는 갱신 질의가 로그 전파 기법으로 처리되는 과정이다. 처음 사용자의 갱신 질의를 받은 노드 N₂는 우선 갱신 연산을 처리한다. 갱신된 데이터를 기반으로 논리 로그를 생성한 후 다른 노드로 로

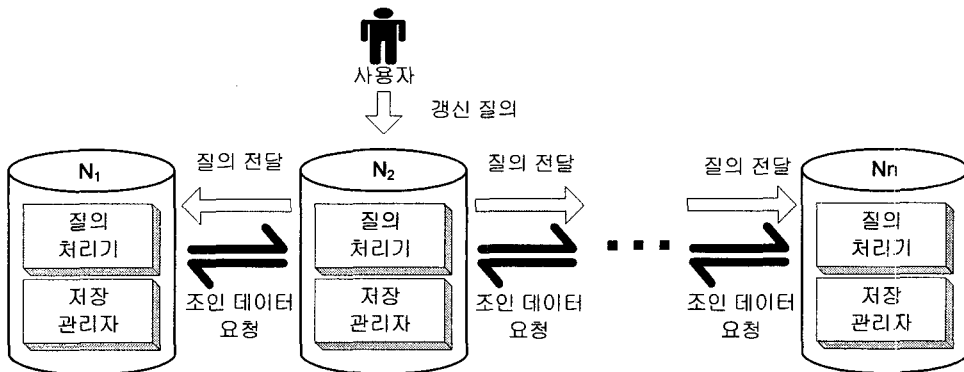


그림 1. 질의 전파 기법

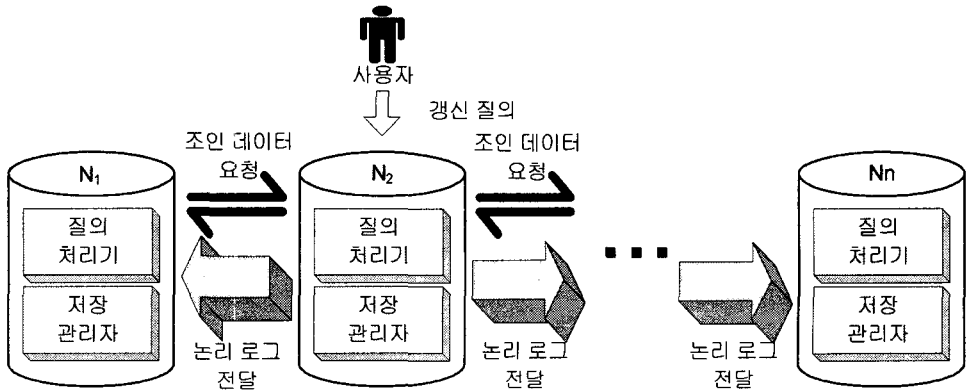


그림 2. 로그 전파 기법

그를 전송한다. 로그를 전송받은 노드는 저장관리자 (storage manager)에서 갱신 연산을 처리한다.

두 가지 기법의 장점과 단점은 다음과 같다. 질의 전파 기법은 조인 데이터와 질의 처리 비용의 크기가 작은 경우 유리하다. 모든 노드에서 갱신 질의가 처리되는 반면에 노드와 노드 간의 네트워크 비용이 적어지기 때문이다. 그러나 질의 처리 비용이 커지고, 조인 데이터가 많아질수록 복제본의 수만큼 갱신 비용이 증가하게 되는 단점이 있다. 로그 전파 기법은 질의 처리 비용과는 상관없이 갱신 데이터의 크기가 작은 경우 유리하다. 첫 노드에서만 질의 처리 과정이 수행되고 나머지 노드에서는 데이터 갱신 과정만 진행되기 때문이다. 하지만 논리 로그의 크기가 커질수록 네트워크 비용이 증가하는 단점이 있다.

임의의 갱신 질의를 크게 질의 처리 부분과 데이터 저장 부분으로 나누어 본다면, 질의 처리 부분에서의 처리 시간이 길어질수록 로그 전파 기법이 유리함을 뜻한다. 반대로 갱신할 데이터의 크기가 커지면서 처리 시간이 길어진다면 질의 전파 기법이 유리하다. 따라서 이러한 두 가지 전파 기법 중 보다 비용이 적은 전파 기법을 선택함으로써 최적의 갱신 처리 비용을 얻을 수 있다.

3. 전송비용 최적화를 위한 복제본 갱신 전파 기법

본 장에서는 복제본을 갱신하기 위한 두 가지 기법의 비용에 따라 전파 방식을 선정하는 분류 함수와 알고리즘을 제안한다.

3.1 전파 방식에 따른 비용 모델

갱신 질의의 응답 시간은 데이터베이스의 구성과 시스템 성능, 네트워크 속도 등에 따라 결과가 크게 달라질 수 있다. 본 장에서는 응답시간을 기준으로 두 가지 전파 기법의 비교를 위해 서로의 비교가 가능한 요소들로 단순화 된 비용 모델을 구성한다. 이를 위해 갱신 질의의 처리 비용에 적용 가능한 인자는 다음과 같다.

- Q : 질의 분석 및 연산 비용 (질의 처리기)
- D : 데이터 반영 비용 (저장 관리자)
- Q_R : $Q+D$ 에서 Q 의 비율
- D_R : $Q+D$ 에서 D 의 비율
- L : 로그의 반영 비용
- J_n : 조인을 위한 접속 노드 수
- J_s : 조인 데이터의 크기
- N : 네트워크 사용 시 단위 비용
- R_n : 복제본의 수

위의 인자를 기반으로 하나의 노드에서 갱신 질의가 처리되는 비용 모델 C_{unit} 은 다음 식 (1)과 같이 나타난다.

$$C_{unit} = Q + D + (J_n \cdot J_s \cdot N) \tag{1}$$

C_{unit} 에서 Q 와 D 가 차지하는 비율을 각각 Q_R 과 D_R 로 정의된다. 따라서 $Q_R + D_R = 1$ 로서 정의된다. 예를 들어 Q_R 이 0.7이면 D_R 은 0.3이 되며, 두 값의 비율은 7:3이 된다. 이때 Q 와 D 는 Q_R 과 D_R 을 통해 다음과 같이 (2), (3)의 식으로 표현 될 수 있다.

$$Q = (C_{unit} - (J_n \cdot J_s \cdot N)) \cdot Q_R \tag{2}$$

$$D = (C_{unit} - (J_n \cdot J_s \cdot N)) \cdot D_R \quad (3)$$

또한 L 은 갱신 데이터를 논리 로그로 변환하여 얻게 되는 값으로서 D 의 크기에 비례하며 다음과 같이 표현된다. 여기서 α 는 논리 로그로 변환 시 추가되는 데이터의 크기를 나타낸다.

$$L = \alpha \cdot D = \alpha \cdot (C_{unit} - (J_n \cdot J_s \cdot N)) \cdot D_R \quad (4)$$

C_{unit} 은 질의 처리기와 저장 관리자에서 연산 처리에 소요되는 시간에 조인 데이터의 전송 시간을 더하여 계산된다. C_{unit} 은 각각의 전파 기법에서 동일하게 사용되는 기본 모델이다. C_{unit} 을 기준으로 구성되는 질의 전파 기법과 로그 전파 기법의 비용 함수가 각각 f_{query} 와 f_{log} 이라하면 다음과 같이 식 (5)와 (6)으로 표현된다.

$$\begin{aligned} f_{query}(D_R) &= C_{query} = Q + D + (J_n \cdot J_s \cdot N) \cdot N \cdot R_n \\ &= C_{unit} \cdot N \cdot R_n \end{aligned} \quad (5)$$

$$\begin{aligned} f_{log}(D_R) &= C_{log} = Q + D + (J_n \cdot J_s \cdot N) + (L \cdot N \cdot R_n) \\ &= C_{unit} + (L \cdot N \cdot R_n) \end{aligned} \quad (6)$$

C_{query} 는 질의 전파 기법에 의한 비용 모델로서, C_{unit} 이 동일 복제본을 갖는 모든 R_n 으로 전송되어 처리됨을 나타낸다. 즉, C_{query} 는 C_{unit} 과 $N \cdot R_n$ 에 비례하여 그 크기가 커지게 됨을 알 수 있다. C_{log} 는 로그 전파 기법에 의한 비용 모델이다. C_{query} 와는 다르게 C_{unit} 은 전체 비용에 큰 영향을 미치지 않는다. 오히려 전송되는 로그의 크기와 복제본의 수인 $L \cdot N \cdot R_n$ 에 비례하여 전체 비용이 증가함을 알 수 있다.

f_{query} 와 f_{log} 는 D_R 의 비율의 변화에 따른 비용을 반환하는 비용함수이다. 이때 f_{query} 는 D_R 의 비율이 바뀌어도 값이 바뀌지 않는다. 이는 D_R 이 바뀌면 Q/R 도 바뀌지만 $Q/R + D_R = 1$ 이라는 관계 때문에 C_{unit} 은 바뀌지 않는다. 그리고 N 과 R_n 이 상수이므로 결과적으로 f_{query} 는 $f_{query}(D_R) = M$ 형태의 상수함수가 된다 (M 은 상수). 또한 f_{log} 는 C_{unit} 이 상수이고 L 은 D 에 비례하고 D 또한 D_R 에 비례한다. 그러므로 f_{log} 는 $f_{log}(D_R) = A \cdot D_R + B$ 의 1차 증가함수 형태가 된다 (A 와 B 는 상수).

두 가지 전파 기법에 대한 각각의 장점과 단점을 통해 질의 처리 비용과 데이터 저장 비용의 비율에

따라 각각의 비용이 다르게 나타날 수 있음을 알 수 있다. 임의의 질의를 기준으로 C_{unit} 의 비용을 동일하게 유지한 상태에서 D_R 을 바꾸면 전체 비용이 아래와 같은 그래프로 나타난다.

[그림 3]의 a)는 C_{query} 의 응답시간 변화를 예측한 그래프이다. a)에서의 응답시간은 항상 일정하게 나타난다. 이는 질의 전파 기법이 D_R 과는 상관없이 항상 일정한 처리 비용을 나타냄을 알 수 있다. b)의 경우 D_R 의 비율이 증가할수록 응답시간이 증가한다. 이는 D_R 의 크기가 커질수록 다른 노드로 전송해야 하는 데이터의 크기가 커지고, 해당 노드에서의 갱신 반영 시간이 길어지기 때문이다. 따라서 C_{log} 에서 D_R 이 커지게 되면 L 도 함께 커지므로 전체 응답시간은 증가하게 된다.

[그림 3]을 바탕으로 두 가지 전파 기법에 대한 3가지 경우를 분류할 수 있다. 그 결과는 [그림 4]과 같다. 우선 [그림 4]의 a)는 C_{query} 의 값이 항상 C_{log} 보다 작은 경우이다. 이러한 상태에서는 질의 전파 기법을 사용하였을 때 최적의 복제본 갱신 성능을 얻을 수 있다. 반대로 b)와 같이 C_{query} 의 값이 항상 C_{log} 보다 큰 경우에는 로그 전파 기법을 사용해야 한다. 그러나 c)처럼 D_R 에 따라 C_{query} 의 값이 큰 영역과 C_{log} 의 값이 큰 영역이 분리될 수 있다. 임의의 교차점을 기준으로 왼쪽과 오른쪽이 각각 로그 전파 기법과 질의 전파 기법을 나누어 사용하였을 때 최적의 전송비용을 얻을 수 있다. 이러한 경우 최적의 갱신 비용을

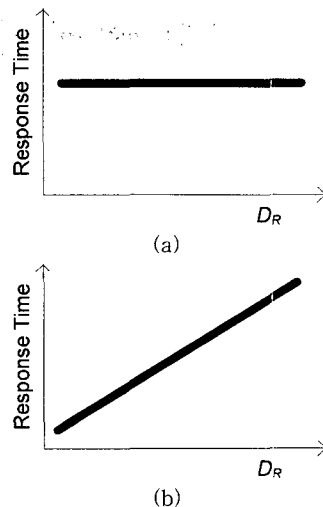


그림 3. D_R 에 따른 처리 비용의 변화: (a) C_{query} 의 처리 비용의 변화, (b) C_{log} 의 처리 비용의 변화

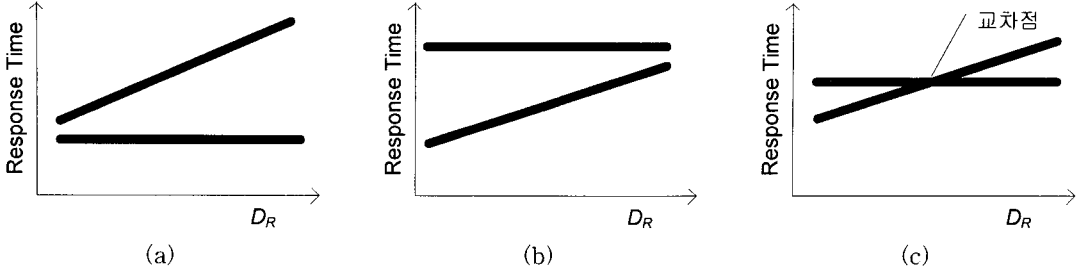


그림 4. 처리 비용에 따른 3가지 분류: (a) C_{query} 의 비용이 낮은 경우, (b) C_{log} 의 처리 비용이 낮은 경우, (c) 두 가지 처리 비용이 교차하는 경우

위해서는 D_R 에 따른 교차점을 찾고, 교차점을 기준으로 전파 기법을 선택하는 알고리즘이 필요하다

3.2 비용 기반 분류 함수와 분류 알고리즘

f_{query} 와 f_{log} 는 각각 상수함수와 1차 증가함수이기 때문에 3가지 경우로 분류 될 수 있다. 따라서 3가지 경우를 판단하기 위한 분류함수를 설명한다.

C_{query} 의 비용이 낮은 경우를 클래스1, C_{log} 의 처리 비용이 낮은 경우를 클래스2, 두 가지 처리 비용이 교차하는 경우를 클래스3이라고 하면, 두 개의 판별 함수로 이 모든 경우를 분류할 수 있다. 먼저 클래스1을 판별하는 함수를 d_{class1} 이라 하고, 클래스2를 판별하는 함수를 d_{class2} 라 한다. 그리고 이 두 판별함수에 모두 속하지 않는 경우는 클래스3으로 분류한다. J_n 과 R_n 은 개수를 의미하므로 자연수이고, 판별함수는 속하거나(참) 안속하거나(거짓)만 판별하므로 두 개의 판별함수는 각각 $d_{class1} : N \times N \rightarrow \{0, 1\}$, $d_{class2} : N \times N \rightarrow \{0, 1\}$ 와 같다.

클래스1은 f_{query} 가 정의역 $[0,1]$ 사이에서 항상 f_{log} 보다 작은 경우이다. f_{query} 는 상수함수이고 f_{log} 함수는 1차 증가함수 이므로 f_{log} 의 최소값이 f_{query} 보다 크면 클래스1의 조건을 만족한다. f_{log} 의 최소값은 $f_{log}(0)$ 이므로, $f_{log}(0) > f_{query}(0)$ 을 만족하는 J_n 과 R_n 이 클래스1에 해당한다. 그리고 클래스2도 비슷하게 접근 할 수 있다. 클래스2는 $[0,1]$ 사이에서 항상 f_{query} 가 f_{log} 보다 큰 경우 이다. 따라서 f_{log} 의 최댓값보다 f_{query} 의 값이 큰 경우이며, $f_{log}(1) < f_{query}(1)$ 을 만족하는 J_n 과 R_n 이 클래스2에 해당한다. 클래스1과 클래스2가 아닌 경우는 클래스3에 해당한다. 클래스3은 [그림 4]의 c)에서 교차점을 계산하여 최적의 전파 기법을 선택한다.

다음은 각각의 클래스를 구별할 수 있는 분류함수

를 나타낸다.

클래스1: $f_{log}(0) > f_{query}(0)$ 을 이용해서 d_{class1} 함수를 구한다. 이때, L 이 0이므로 $f_{query}(0) = C_{unit} \cdot N \cdot R_n$ 이며, $f_{log}(0) = C_{unit}$ 이다.

$$\begin{aligned} f_{log}(0) &> f_{query}(0) \\ \rightarrow C_{unit} &> C_{unit} \cdot N \cdot R_n (\because L = 0) \\ \rightarrow C_{unit} (1 - N \cdot R_n) &> 0 \\ \rightarrow (Q + (J_s \cdot N \cdot J_n)) \cdot (1 - N \cdot R_n) &> 0 \\ (\because D_R = 0 \rightarrow D = 0) \end{aligned} \quad (7)$$

위 식을 만족할 때 모든 D_R 에 대해 $f_{log}(D_R) > f_{query}(D_R)$ 이므로 클래스1의 경우이다. 그러므로 $d_{class1} : N \times N \rightarrow \{0, 1\}$ 는 아래와 같이 정의 된다.

$$d_{class1}(J_n, R_n) = \begin{cases} 1, & \text{if } (Q + (J_s \cdot N \cdot J_n)) \cdot (1 - N \cdot R_n) > 0 \\ 0, & \text{if otherwise} \end{cases} \quad (8)$$

클래스2: $f_{log}(1) < f_{query}(1)$ 을 이용해서 d_{class2} 함수를 구한다. 이때, $f_{log}(1) = C_{unit} + \alpha(C_{unit} - (J_n \cdot N \cdot J_n)) \cdot N \cdot R_n$ 이고, $f_{query}(1) = C_{unit} \cdot N \cdot R_n$ 이다.

$$\begin{aligned} f_{log}(1) &< f_{query}(1) \\ \rightarrow C_{unit} + (L \cdot N \cdot R_n) &< C_{unit} \cdot N \cdot R_n \\ \rightarrow \frac{C_{unit} + (L \cdot N \cdot R_n)}{C_{unit} \cdot N \cdot R_n} &< 0 \\ \rightarrow \frac{1}{N \cdot R_n} + \frac{L}{C_{unit}} &< 0 \\ \rightarrow \frac{1}{N \cdot R_n} + \frac{L}{D + (J_s \cdot N \cdot J_n)} &< 0 \\ (\because Q_R = 0 \rightarrow Q = 0) \end{aligned} \quad (9)$$

위 식을 만족할 때 모든 D_R 에 대해 $f_{query}(D_R) > f_{log}(D_R)$ 이므로 클래스2의 경우이다. 그러므로

$d_{class2} : N \times N \rightarrow \{0, 1\}$ 는 아래와 같이 정의 된다.

$$d_{class2}(J_n, R_n) = \begin{cases} 1, & \text{if } \frac{1}{N \cdot R_n} + \frac{\alpha(Q+D)}{Q+D+(J_n \cdot N \cdot J_s)} < 1 \\ 0, & \text{if otherwise} \end{cases} \quad (10)$$

클래스3 : 클래스3의 경우는 교차점에 해당하는 D_R 을 찾고 이를 기준으로 효율적인 전파 기법을 찾아야 한다. 교차점은 두 비용 함수의 교점이므로, $f_{query}(D_R) = f_{log}(D_R)$ 을 만족하는 해를 찾아야 한다. 따라서 $f_{query}(D_R) = f_{log}(D_R)$ 을 이용해서 교차점에 해당하는 D_R 인 D_{opt} 을 구한다.

$$\begin{aligned} f_{query}(D_R) &= f_{log}(D_R) \\ \rightarrow C_{unit} \cdot N \cdot R_n &= C_{unit} + (L \cdot N \cdot R_n) \\ \rightarrow N \cdot R_n \cdot (C_{unit} - 1) &= L \cdot N \cdot R_n \\ \rightarrow L &= C_{unit} - 1 \\ \rightarrow \alpha \cdot (C_{unit} - (J_n \cdot J_s \cdot N)) \cdot D_R &= C_{unit} - 1 \\ \rightarrow D_R &= \frac{C_{unit} - 1}{C_{unit} - (J_n \cdot J_s \cdot N)} \end{aligned} \quad (11)$$

따라서 C_{unit} 와 $J_n \cdot J_s \cdot N$ 이 주어졌을 때 $D_{opt} = \frac{C_{unit} - 1}{C_{unit} - (J_n \cdot J_s \cdot N)}$ 라 하면 $D_R > D_{opt}$ 인 경우에는 질의 전파 기법을 사용하고, $D_R \leq D_{opt}$ 인 경우에는 로그 전파 기법을 사용하는 것이 효율적이다.

주어진 J_n 과 R_n 에 대해 $d_{class1}(J_n, R_n) = 1$ 이고 $d_{class2}(J_n, R_n) = 0$ 이면 클래스1에 해당한다. 반면에 $d_{class1}(J_n, R_n) = 0$ 이고 $d_{class2}(J_n, R_n) = 1$ 이면 클래스2에 해당하며, 두 함수의 결과값이 동시에 1인 경우는 발생하지 않는다. 둘 다 0인 경우는 클래스3에 해당한다. 클래스1과 클래스2는 모든 D_R 에 상관없이 한쪽의 알고리즘이 성능이 좋지만, 클래스3은 D_R 에 따라 달라진다. 그러므로 클래스3의 경우는 D_{opt} 에 따라 최적의 전파 기법이 달라진다.

이들 분류 함수는 [알고리즘 1]과 같은 순서에 의해 복제본 갱신 시 적용될 수 있다. [알고리즘 1]은 사용자의 갱신 질의가 복제본이 있는 노드로 전송된 후 처리되는 과정을 나타낸다. 처음 질의가 입력되면 해당 노드에서 우선 갱신 질의를 처리한다(01라인). 이는 클래스 분류를 위한 인자값을 얻기 위함이다(02라인). 다음 과정은 클래스1과 클래스2에 대한 판별 과정이다. PropagationMethod는 클래스 종류에 대한 값을 가지는 변수이다. QUERY_PROPAGATION과 LOG_PROPAGATION은 각각 질의 전파 기법과 로그 전파 기법을 뜻한다. 최종 선택되는 전

Input

Query: 갱신 질의

Output

PropagationMethod: 선택된 복제본 갱신 전파 기법의 종류

Begin

```

01 : ProcessUpdateQuery(Query) // 현재 노드에서
    갱신 질의 처리
02 : GetParameterValue() // 분류 함수 적용에 필요
    한 인자 값을 얻어옴
03 : if CalculateClass1() = true
04 : PropagationMethod ← QUERY_PROPAGA-
    TION
05 : else if CalculateClass2() = true
06 : PropagationMethod ← LOG_PROPAGATION
07 : else
08 : Dopt ← CalculateClass3()
09 : if Dr > Dopt
10 : PropagationMethod ← QUERY_PROPAGA-
    TION
11 : else
12 : PropagationMethod ← LOG_PROPAGA-
    TION
13 : endif
14 : endif
15 : SendPropagationMessage(PropagationMethod)
    // 갱신 전파 메시지를 다른 노드로 전송
End
    
```

[알고리즘 1] 갱신 질의 처리 과정

파 기법에 따라서 클래스1인 경우 QUERY_PROPAGATION 값을 PropagationMethod에 대입하며 클래스2인 경우 LOG_PROPAGATION 값을 대입한다. CalculateClass1()은 클래스1인 경우를 판별하고 CalculateClass2()은 클래스2인 경우를 판별한다(03라인~06라인). 클래스3에 해당하는 경우 Dopt 변수에 CalculateClass3()에 의해 계산된 교차점 값을 대입한다. Dr과 Dopt 값을 비교하여 각각의 전파 기법을 선택하게 된다(07라인~14라인). 마지막으로 SendPropagationMessage() 함수를 호출하여 복제본 갱신 전파 과정을 수행한다(15라인).

4. 성능 평가

본 장에서는 제안 기법의 평가를 위한 실험 환경에 대해 설명하고, 조인 데이터의 수, 복제본의 수 등의 변화에 따른 전파 기법들의 성능 비교를 한다.

4.1 실험환경

본 논문에서는 제안 기법의 평가를 위해 C/C++ 언어 기반 시뮬레이션 툴인 CSIM을 사용하였다. CSIM은 분산 환경에서의 컴퓨팅 모델, 알고리즘 평가 등 대부분의 시뮬레이션이 가능한 툴이다. 이미 여러 논문에서 CSIM을 통한 성능평가가 작성된 바 있다[8].

제안 기법의 구현을 위해 20대의 노드를 구성하였으며, 무작위로 질의를 생성하여 모든 노드로 전송하였다. 전파 기법의 비교 평가를 위해 질의 전파 기법, 로그 전파 기법 그리고 본 논문에서 제안하는 비용 별 분류에 따른 전파 최적화 기법을 구현하였다. 전체 구현에 소요된 라인 수는 약 3800라인이며, MS Visual C++ 6.0을 사용하였다. 전체적인 구성은 가능한 실제 환경에 맞게끔 구현하였으며, 평가 환경은 다음과 같이 구성하였다.

실험 환경에서 조인 데이터의 크기, 네트워크 부하에 의한 전송 지연, 노드에서의 작업 부하 등의 요인은 환경에서 제외시켰다. 이는 실험 목적과 상관없는 영향을 끼치는 부분을 제거함으로써 보다 정확한 결과를 얻고자 함이다.

4.2 실험평가

본 실험 평가에서는 3장에서 다룬 D_R 과 조인 데이터의 수 그리고 복제본의 수를 평가 인자로 사용하였다. 이외에 서버의 수, 데이터의 수, 질의 처리 시간 등은 실제 그 값을 바꾸어가며 실험한 결과 변위 값의 증감 외에는 별 다른 변동사항이 없어 본 논문에서는 설명하지 않는다. 모든 실험 결과는 처리 시간(Response Time)과 처리량(Throughput)으로 나타낸다.

표 1. 실험 환경

실험 요소	데이터 범위	실험 요소	데이터 범위
실험 시간	5000 unit time	네트워크 전송 시간	3-4 unit time
노드 수	20 개	질의 입력 간격	3 unit time
클라이언트 수	40 개	질의 처리 시간	10-12 unit time
데이터 수	50 개	데이터 별 복제본 수	2-18 개

[그림 5]는 D_R 의 변화에 따른 처리 성능의 변화를 나타낸다.

[그림 5]는 3장의 [그림 4]와 같이 C_{unit} 의 비용을 동일하게 유지한 상태에서 Q와 D의 값을 변화시키며 측정한 결과이다. [그림 5]의 모든 실험에서 조인 데이터의 수, 네트워크 비용, 복제본의 수 등을 바꾸어가며 실험한 결과 [그림 4]에서 예측한 그래프와 동일하게 나타났다.

모든 실험에서 질의 전파 기법은 일정한 응답 시간을 보였으며, 단지 몇몇 인자로 인한 평균값의 차이만 있다. 로그 전파 기법은 마찬가지로 1차 함수 형태의 기울기를 가진 그래프로 나타났으며, 환경의 변화에 따른 기울기의 변화가 있었다.

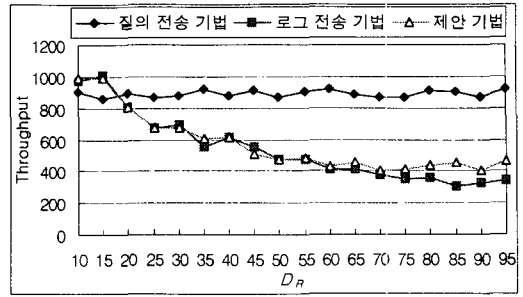
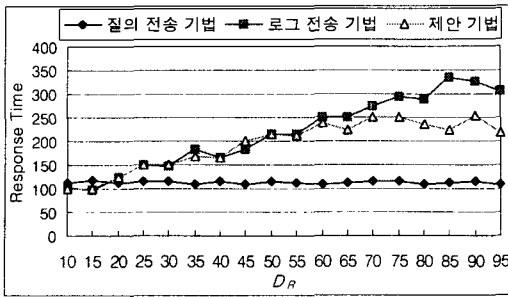
제안 기법은 최적에 가까운 성능을 보이고 있다. 각각의 실험에서 제안 기법은 항상 기존의 두 가지 기법 중 더 낮은 응답 시간을 보이는 것과 비슷한 성능을 나타내고 있다. 이는 3장에서 제안한 클래스에 따른 분류 기법이 실제로 적용이 가능함을 나타낸다. 다만 클래스1에서와 같이 이상적인 그래프와 약간의 차이가 나는 이유는 무작위로 질의를 선정해 보내는 과정에서 발생한 오차인 것으로 추측된다.

[그림 6]와 [그림 7]은 복제본 갱신 성능에 가장 큰 영향을 끼치는 조인 데이터 수와 복제본 수의 변화에 따른 처리 성능의 변화를 나타낸다. 위의 실험 평가 중 클래스3과 동일한 환경에서 실험을 하였으며, 각각의 변위 값이 증가함에 따라 이 두 가지 실험에서도 제안 기법은 항상 최적의 성능을 보이고 있다.

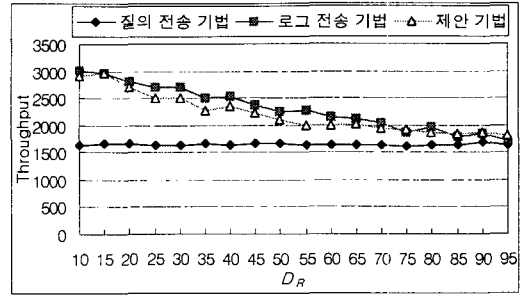
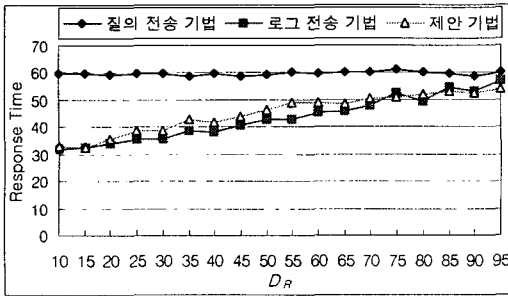
5. 결 론

서비스 가용성과 처리 성능은 이미 데이터베이스 분야에서 가장 중요한 이슈 중 하나이다. 특히 자료를 중복 관리하는 복제본 관리 기술은 처리 비용이 낮고 성능이 높을수록 그 활용도가 높아지기 때문에 최적화를 위한 연구가 지속적으로 이루어지고 있다.

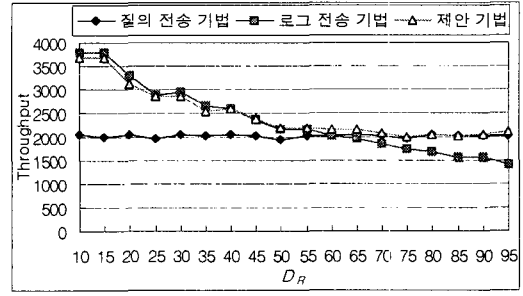
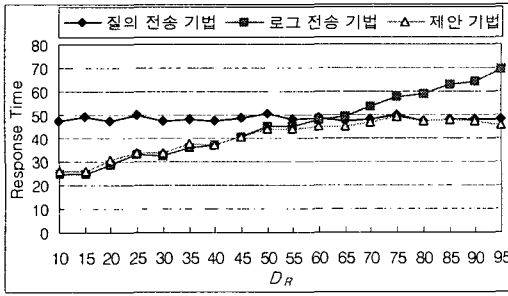
본 논문에서는 전송비용 최적화를 위한 복제본 갱신 전파 기법을 제안하고 이에 대한 성능 평가를 수행하였다. 제안 기법은 일반적으로 복제본 갱신을 위해 사용될 수 있는 두 가지 갱신 메시지의 전파 기법에 대한 장·단점을 통해 두 가지 전파 기법을 동시에 사용할 경우 발생할 수 있는 세 가지 경우에 대한 모델을 정리하였다. 이를 기반으로 세 가지 경우를



(a)

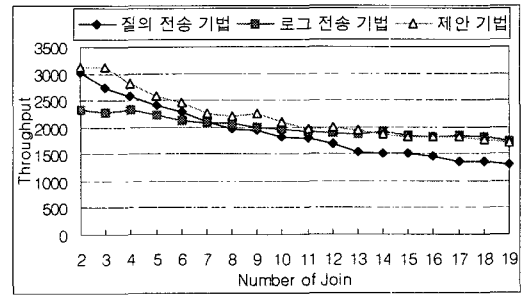
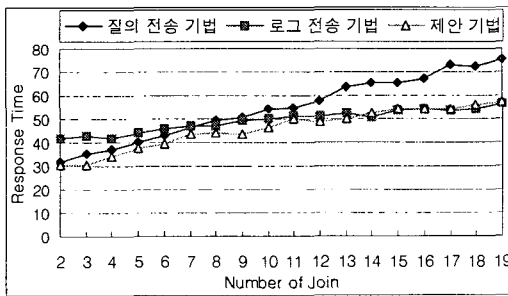


(b)



(c)

그림 5. D_R 의 변화에 따른 처리 성능의 변화: (a) 클래스1에서의 성능 변화, (b) 클래스2에서의 성능 변화, (c) 클래스3에서의 성능 변화



(a)

(b)

그림 6. 조인 데이터 수의 변화에 따른 처리 성능의 변화: (a) 평균 질의 처리 시간, (b) 단위 시간 당 처리량

분류할 수 있는 분류 함수를 설계하였으며, 분류 함수를 통해 최적의 갱신 메시지 전파 방식을 선택할 수 있었다.

성능 평가에서도 제안 기법은 환경의 변화에 상관 없이 항상 최적의 성능이 나타났다. D_R 의 변화에 따른 성능 평가에서 제안 기법은 항상 주어진 환경에

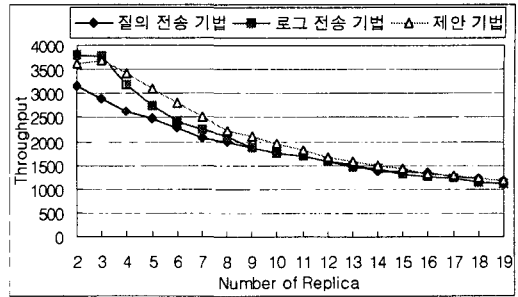
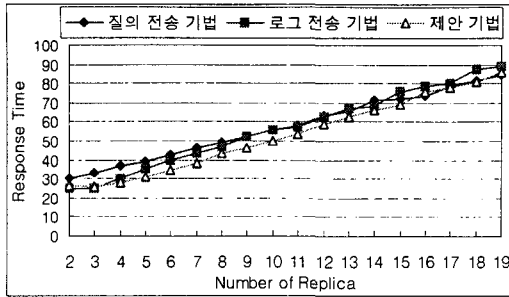


그림 7. 복제본 수의 변화에 따른 처리 성능의 변화: (a) 평균 질의 처리 시간, (b) 단위 시간 당 처리량

최적의 조건을 갖는 복제본 갱신 전파 기법과 비교해 5~10% 이내의 오차에서 거의 비슷한 수준의 성능을 보였다.

제안 기법은 복제본 관리에 드는 비용을 줄이고 동기화 성능을 개선함으로써 데이터베이스 서버의 가용성과 동적으로 변화하는 환경에서의 활용성을 향상시켰다. 또한 본 기법은 복제본을 관리하는 분산 데이터베이스 시스템, 데이터베이스 클러스터 등의 어떠한 시스템에서도 적용이 가능하다.

참고 문헌

[1] 강철, "On-Demand 환경의 데이터베이스 Grid와 고 가용성," 데이터베이스연구회, VOL. 00 NO. 00 pp. 209~224, 2005.

[2] Sidell, J., Aoki, P.M., Sah, A., Staelin, C., Stonebraker, M., Yu, A., "Data replication in mariposa," *In Proceedings of the 12th International Conference on Data Engineering, IEEE Computer Society*, pp. 485~494, 1996.

[3] D. Agrawal, G. Alonso, A. E. Abbadi, and I. Stanoi., "Exploiting atomic broadcast in replicated databases," *In Proceedings of EuroPar (EuroPar'97)*, Passau (Germany), pp. 496~503, 1997.

[4] Kwang Chul Jung, *Performance and Availability Enhancement of Transaction Processing for Main Memory DBMS*, PhD thesis, Department of Computer Science and Information Engineering, INHA Univ., Korea, 2005.

[5] S. Ceri and G. Pelagatti, *Distributed Databases:*

Principles & Systems, McGraw-Hill Company, 1984.

[6] Maria A., Nieto-Santisteban, Jim Gray, Alexander S. Szalay., James Annis, Aniruddha R. Thakar, and William J. O'Mullane, "When database System Meet Grid," *In Proceedings of the 2005 CIDR Conference*, pp. 154~161, 2005.

[7] J. Gray, P. Helland. P. O'Neil, and D. Shasha, "The dangers of replication and a solution," *ACM SIGMOD*, Vol. 25, No. 2, pp. 173~182, 1996.

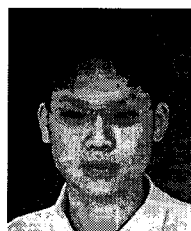
[8] Mesquite Software. Inc., "CSIM19 The Simulation Engine," <http://www.mesquite.com>, 2005.



장용일

1997년 인하대학교 전자계산공학과 (공학사)
 2001년 인하대학교 컴퓨터공학부 (공학석사)
 2003년~현재 인하대학교 컴퓨터 정보공학과 박사과정

관심분야: 웹 & 모바일 GIS, 데이터베이스 클러스터, 위치기반서비스, 그리드 데이터베이스



백성하

2005년 인하대학교 수학과 졸업 (이학사)
 2006년~현재 인하대학교 컴퓨터 정보공학과 석사과정

관심분야: 스트림 데이터베이스, 공간 데이터베이스, 분산 데이터베이스, 하이브리드 데이터베이스 등



배 해 영

- 1974년 인하대학교 응용물리학과(공학사)
- 1978년 연세대학교 대학원 전자계산학과(공학석사)
- 1989년 숭실대학교 대학원 전자계산학과(공학박사)

1992년~1994년 인하대학교 전자계산소 소장
1982년~현재 인하대학교 컴퓨터공학부 교수
1999년~현재 지능형GIS연구센터 센터장
2000년~현재 중국 중경우전대학교 대학원 명예교수
2004년~2006년 인하대학교 정보통신대학원 원장
2006년~현재 인하대학교 일반대학원 원장
관심분야 : 분산 데이터베이스, 공간 데이터베이스, 지리 정보 시스템, 멀티미디어 데이터베이스 등