

# 비공유 공간 데이터베이스 클러스터에서 효율적인 병렬 공간 조인 기법

김종현\*, 김명근\*, 김재홍\*\*, 배해영\*  
\*인하대학교 전자계산공학과  
\*\*영동대학교 컴퓨터공학전공  
e-mail : kjh1592@dblab.inha.ac.kr

## Efficient Parallel Spatial Join Method In Shared-Nothing Spatial Database Cluster

Jong-Hyun Kim\*, Myung-Keun Kim\*, Jae-Hong Kim\*\*, , Hae-Young Bae\*  
\*Dept. of Computer Science & Engineering, Inha University  
\*\* Dept. of Computer Engineering, Youngdong University

### 요 약

최근 인터넷 환경에서 지리 정보 서비스를 제공 받으려는 사용자들의 지속적인 증가로 인해 저비용의 여러 개의 단일 노드를 고속의 네트워크로 연결하여 고성능을 제공하는 클러스터 기반의 공간 데이터베이스에 대한 연구가 활발하게 진행되고 있다. 이러한 공간 데이터베이스 클러스터에서 사용자가 요구한 공간 질의를 빠르게 처리하기 위해서는 고비용의 공간 조인 연산을 효율적으로 처리하기 위한 기법이 요구된다.

본 논문에서는 비공유 공간 데이터베이스 클러스터 환경하에서 공간 조인 연산 수행 시 효율적인 병렬 처리를 위한 영역 분할 기법 및 병렬 공간 조인 기법을 제안한다. 기존의 병렬 공간 데이터베이스 시스템에서의 분할 기반 병렬 공간 조인 기법들은 병렬로 수행할 작업 분배 및 할당과 분할 경계선 상에 존재하는 공간 객체들에 대한 중복 조인 연산 및 중복 결과 제거 등의 추가적인 연산을 해야 한다는 문제점들이 있다.

제안된 기법은 공간 릴레이션들을 일정 영역들로 분할하여 비공유 공간 데이터베이스 클러스터의 각 노드에서 중복없이 저장, 관리하도록 하며 분할 영역의 경계선 상에 위치하는 공간 데이터에 대해서만 중복 저장을 허용하여 병렬 공간 조인 연산 시 누락되는 공간 데이터가 없도록 한다. 본 기법은 공간 조인 연산 시 병렬 처리를 위한 별도의 작업 할당 과정을 거치지 않고 각 노드에서 병렬적으로 공간 조인 연산을 수행하며, 분할 경계선 상에 존재하는 공간 객체들은 여과 과정을 거쳐 한번만 공간 조인이 수행되므로 중복 결과들을 제거하기 위한 별도의 연산이 필요없는 특징을 갖는다.

### 1. 서론

최근 고성능 프로세서와 초고속 네트워크 등의 하드웨어 기술이 발전하게 됨에 따라 슈퍼컴퓨터와 같은 고가의 대형 컴퓨터를 사용하는 대신에 여러 개의 프로세싱 노드들을 클러스터링 기술을 사용하여 고속의 네트워크로 묶는 클러스터 시스템이 많이 활용되고 있다. 특히 인터넷 및 무선 통신 환경의 급속한 발전과 더불어 웹 및 무선 단말기를 이용하여 지리 정보 서비스를 제공받으려는 사용자들의 지속적인 증가로 인해 기존의 단일 대용량 공간 데이터베이스 서버로는 사용자의 질의 요구를 감당하기 어려워짐에 따라 저비용의 여러 개의 단일 노드를 고속의 네트워크로 연결하여 고성능을 제공하는 클러스터 기반의 공간 데이터베이스에 대한 연구가 활발하게 진행되고 있다.

이러한 공간 데이터베이스 클러스터에서 주로 사용하는 공간 연산은 영역 질의와 공간 조인으로서 연산의 복잡도가 매우 큰 연산자이다. 특히 공간 조인은 단일 주사(single scan) 방식의 질의인 점 질의나 영역 질의와는 달리 다중 주사(multiple scan) 방식의 질의이

\* 본 연구는 정보통신부의 대학 S/W 연구센터 지원사업의 연구결과임.

기 때문에 객체의 수가 증가함에 따라 연산 시간이 급격히 증가한다.[1] 그러므로 사용자가 요구한 공간 질의의 빠른 처리를 위해서는 공간 조인의 효율적인 처리 방안이 필요하다.

본 논문에서는 이러한 고비용의 공간 조인 연산을 비공유 공간 데이터베이스 클러스터 환경하에서 빠르고 효율적으로 처리하기 위한 병렬 공간 조인 기법을 제안한다.

기존의 병렬 공간 데이터베이스 시스템에서 영역 분할 기반의 병렬 공간 조인 기법은 병렬 수행을 하기 위한 영역 분할 및 작업 할당 등의 작업 생성 단계를 필요로 하며, 분할의 경계선 상에 존재하는 중복 객체들에 대해 해당 병렬 작업에서 중복 공간 조인 연산을 수행한 후 최종 결과에서 중복 결과들을 제거하는 등 추가적인 수행 비용을 필요로 한다.

제안된 기법은 공간 릴레이션들을 비공유 공간 데이터베이스 클러스터 노드 개수만큼 일정한 영역들로 분할하여 해당 영역의 공간 데이터들을 관리하는 각 노드에 저장하게 되며, 공간 릴레이션의 분할 영역 경계선에 위치하는 공간 데이터에 대해서만 해당 영역을 관리하는 노드들에 중복 저장한다. 사용자로부터

요구된 공간 조인 질의 수행 시 각 노드마다 자신이 관리하는 영역 내의 공간 릴레이션들에 대해서만 병렬적으로 공간 조인을 수행하여 별도의 작업 생성 단계가 필요하지 않으며, 분할 영역 경계선 상에 위치하여 중복 저장된 공간 데이터에 대해서는 여과 단계에서 두 공간 데이터의 MBR 중점의 위치에 따라 여과를 수행하여 고비용이 요구되는 정제 연산을 여러 노드에서 중복 수행하는 것을 방지한다. 또한, 병렬 공간 조인 결과에 대해 별도의 중복 결과 제거 작업이 필요없다.

본 논문의 내용 구성은 다음과 같다. 2 장에서 관련 연구 내용으로 기존의 병렬 공간 조인 기법에 대해 설명한다. 3 장에서는 본 기법의 바탕이 되는 공간 데이터베이스 클러스터의 환경에 대해 설명하고, 공간 분할 기법 및 병렬 공간 조인 기법에 대해 설명한다. 마지막으로 4 장에서 성능평가를 하며, 5 장에서 결론 및 향후 연구로 마치도록 하겠다.

## 2. 관련 연구

### 2.1 병렬 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법

공간 조인은 두 공간 데이터 집합에 대해 포함(within), 교차(cross), 겹침(overlap)등의 특정 공간 조건을 만족하는 객체 쌍의 집합을 찾아내는 연산이며 두 데이터 집합에 대한 다중 주사를 필요로 하므로 객체의 수가 증가함에 따라 연산 시간이 급격히 증가한다는 특징을 갖는다. 따라서 빠른 질의 처리를 요구하는 응용에서 공간 조인의 효율적인 처리 방안이 필요하며 이를 위해, 병렬 공간 데이터베이스 시스템에서 다중 프로세서를 이용한 병렬 공간 조인에 대한 많은 연구가 진행되어 왔다. 관련연구[2]에서는 단일 할당 공간 색인 방식인  $R^*$ -tree 상에서 공유 가상 메모리(shared-virtual memory)를 가지는 병렬 시스템에서의 공간 조인을 위한 방법들을 제시하고 있다. 이 연구에서는 정적 태스크(task) 할당, 정적 라운드 로빈 태스크 할당, 동적 태스크 할당으로 나누어 처리하는 방법을 소개하고 있다. 이 때 서로 동일  $R^*$ -tree 노드를 다른 프로세서에 할당하고 있는데 이는 디스크 병목 현상을 일으키는 요인이 된다. 또한 중복 정제 수행 및 중복된 결과를 제거해야 하는 문제가 있다..

관련연구 [3]에서는 단일 할당 공간 색인을 이용한 것으로서 비겹침-정규 데이터 공간 분할 방식을 사용하는 PMR Quad tree 에서 병렬 조인을 위한 방법에 대해서 다루고 있다. 이 연구에서는 SIMD(Single Instruction Multiple Data) array processor 환경에서 병렬 공간 조인 방법을 제시하고 있다. 그러나 SIMD 는 특수 목적 응용에 주로 사용되는 시스템이며 또한 이 방법은 노드의 병합과정에서 병합된 사분면들의 데이터를 주기억장치 내에 적재하여 처리하므로 극단적인 경우에는 많은 병합과정이 인해 여러 번의 페이지 교체(replacement)가 일어나 전체적인 수행속도가 느려지며, 중복 정제 및 중복 결과에 대한 문제점을 갖고 있다. 한편, 관련 연구 [9]에서는 먼저 여과 연산을 병렬적으로 수행하여 후보 객체 쌍들로부터 중복을 제

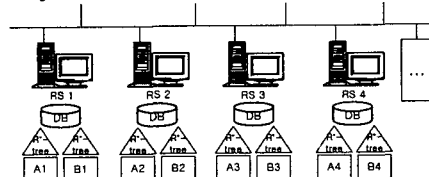
거한 후 다시 병렬적으로 정제 연산을 수행하여 중복 정제 연산을 제거했지만 여과와 정제 연산을 분리하여 각각에 대해 태스크 생성 및 할당을 해야하는 단점이 있다.

이러한 기존의 단일 병렬 공간 데이터베이스 시스템에서는 고비용의 공간 조인 연산 수행 시 프로세서간의 병렬 작업 분배의 문제, 디스크 병목 현상의 문제 및 중복 결과들을 제거하기 위한 연산 비용 등의 문제를 유발시키며, 이러한 문제들은 본 논문에서 제안한 비공유 공간 데이터베이스 클러스터 환경에서의 병렬 공간 조인 기법으로 해결될 수 있다.

### 3. 비공유 공간 데이터베이스 클러스터에서 효율적인 병렬 공간 조인 기법

#### 3.1 비공유 공간 데이터베이스 클러스터 환경

제안된 비공유 공간 데이터베이스 클러스터의 환경은 [그림 3-1]과 같다.



[그림 3-1] 비공유 공간 데이터베이스 클러스터

위의 [그림 3-1]과 같이 비공유 공간 데이터베이스 클러스터는 Shared-Nothing 기반의 여러 개의 복제 서버(Replication Server)들로 구성되어 있으며, 각 복제 서버는 복제 서버의 개수만큼 분할된 영역들로 구성되는 공간 릴레이션의 특정 영역의 공간 데이터들을 관리, 저장하게 된다. 한 복제 서버에 들어있는 공간 릴레이션들의 데이터들은 서버가 관리하고 있는 특정 영역안에 포함되는 객체들이며, 영역에 객체가 포함되는지의 여부는 해당 객체의 MBR 중점들을 가지고 판단하게 된다. 또한 복제 서버에서 영역별로 관리되는 분할된 모든 공간 릴레이션들은 효율적인 공간 질의 처리를 위하여  $R^*$ -tree 공간 색인이 구축되어 있다고 가정한다.[1]

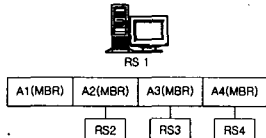
각각의 복제 서버는 고가용성을 위해 자신의 공간 데이터를 이중화하여 백업하는 별도의 백업 서버가 있을 수 있으나 본 논문에서는 비공유 공간 데이터베이스 클러스터의 고성능성에 초점을 맞추었으므로 고가용성에 대한 자세한 언급은 생략하도록 하겠다.

#### 3.2 공간 분할 기법

제안된 비공유 공간 데이터베이스 클러스터 환경 하에서 공간 릴레이션들은 복제 서버의 수만큼 일정한 영역들로 분할되어 각 노드에 저장된다. 이런 분할 영역들은 공간 객체들의 분포나 공간 객체에 대한 접근 유형 등을 분석하여 나눌 수도 있으나, 본 논문에서는 편의상 동일한 크기의 영역들로 분할된다고 가정한다.

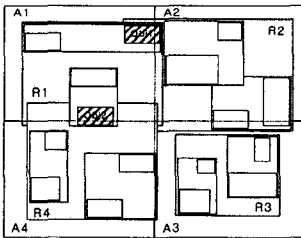
이와 같은 공간 분할을 기반으로 하는 병렬 공간 조인 시에는 분할 경계 영역상에 존재하는 객체들에 대해 각 서버간의 불필요한 중복 공간 조인 연산이

발생하게 되며, 중복 공간 조인 결과들을 제거하기 위한 추가적인 연산 비용 등의 문제가 발생하게 된다. 이러한 문제들을 해결하기 위해 본 논문에서는 특정 객체의 MBR 이 서버간 분할 경계 영역상에 존재할 경우, 객체의 MBR 과 겹쳐지는 분할 영역들을 관리하는 복제 서버들에 중복해서 저장하도록 하며, 공간 질의 처리 시 여과 단계에서 객체의 MBR 의 중심점이 속해 있는 복제 서버에서만 해당 객체에 대한 공간 연산을 수행하도록 한다. 이때 중복 저장되는 객체는 각각의 복제 서버의 R\*-tree 공간 색인에도 중복되어 삽입된다.



[그림 3-2] 분할 영역에 대한 메타 정보

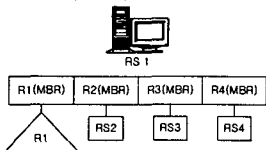
각 복제 서버들은 위의 [그림 3-2]와 같이 복제 서버들의 분할 공간 영역에 대한 메타 정보를 가지고 있으므로 새로운 공간 객체가 해당 공간 릴레이션에 삽입될 경우, 메타 정보의 분할 공간 영역과 삽입될 공간 객체의 MBR 을 비교하여 중복 저장될 복제 서버를 간단하게 알아낼 수 있다.



[그림 3-3] 공간 릴레이션 A 에 대한 분할 영역

위의 [그림 3-3]은 공간 릴레이션 A 에 대해 4 대의 복제 서버가 공간 릴레이션 A 를 4 개의 영역(A1, A2, A3, A4)로 분할하여 각각 저장, 관리하는 것을 나타낸 것이다. 각 복제 서버는 분할된 영역에 대해 독립적으로 R\*-tree 공간 인덱스를 구축해 놓는다.

[그림 3-3]에서 Obj1 과 Obj2 의 MBR 은 각각 (A1, A2)와 (A1, A4) 분할 영역의 경계선 상에 있으므로 각각의 복제 서버에 중복되어 저장되며, 각 복제 서버의 R\*-tree 공간 인덱스에도 추가된다.



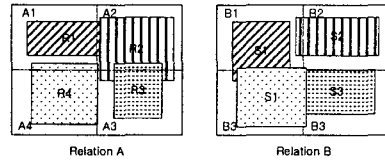
[그림 3-4] R\*-tree 공간 인덱스 메타 정보

각 복제 서버는 자기 자신의 R\*-tree 루트 노드의 MBR 영역과 다른 복제 서버들의 루트 노드 MBR 영역들을 메타 정보로 갖게 되며, 점 질의나 영역 질의 등의 단순 공간 질의 처리 시에는 이러한 메타 정보를 사용하여 공간 질의 처리에 관련된 복제 서버들을 찾아낸다.

### 3.3 병렬 공간 조인 기법

공간 조인의 병렬 수행 작업은 별도의 작업 할당의 과정 없이 각 복제 서버에서 병렬적으로 자신이 소유하고 있는 분할된 공간 테이블들에 대해 공간 조인을 수행한 후 해당 결과를 질의 요청을 받은 서버로 전송하면 된다.

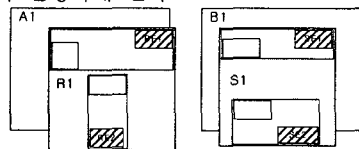
각 복제 서버에서 병렬적으로 수행되는 공간 조인은 R\*-tree 를 이용한 여과 단계와 여과된 공간 객체들에 대한 실제 조인 가능 여부를 판단하는 정제 단계로 진행된다.



[그림 3-5] 공간 조인 대상 릴레이션 A, B

위의 [그림 3-5]과 같이 각 복제 서버는 자신에게 할당된 분할 영역에 포함되는 공간 릴레이션들만 관리하며, 각각의 공간 릴레이션들에 대해 독립적으로 R\*-tree 공간 색인이 구축되어 있다. 만일 클라이언트 사용자로부터 공간 릴레이션 A 와 공간 릴레이션 B 사이에 공간 조인 연산 수행 요청이 들어오면 모든 복제 서버가 병렬적으로 공간 조인 연산에 참여하게 되며, 다른 복제 서버와 통신할 필요없이 자신이 가지고 있는 분할된 릴레이션들에 대해서만 공간 조인 연산을 수행하게 된다.

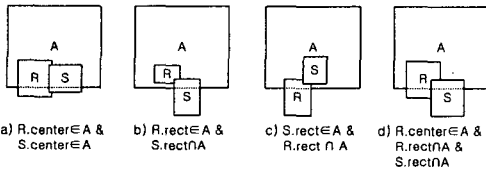
R\*-tree 를 이용한 여과 단계는 기존의 [1]에서 사용한 알고리즘을 사용하며 각 복제 서버에서 중복되어 저장, 관리되는 분할 영역 경계상의 객체들에 대한 중복 정제 연산을 제거하기 위해 여과 수행 시 두 객체의 MBR 중점의 위치 및 영역 포함 관계에 따라 해당 복제 서버에서 정제 연산을 수행할 결과 객체로 출력할 것인지 결정하게 된다.



[그림 3-6] 중복 저장 객체에 대한 공간 조인 예

위의 [그림 3-6]과 같이 R\*-tree 를 이용한 분할 릴레이션 A1 과 B1 의 MBR 여과 연산의 결과로 (RE1, SE1)과 (RE2, SE2)를 얻을 수 있으나 이들 중 (RE2, SE2)는 해당 MBR 들의 중심점이 A1 분할 영역에 속하지 않으므로 여과 결과에서 제외되며 고비용의 정제 과정을 거치지 않게 된다. 이 두 객체에 대한 공간 조인 연산은 A4 분할 영역을 가지고 있는 서버에서 수행될 것이다.

분할 영역의 경계선 상에서 중복 저장되는 객체들에 대한 공간 조인은 해당 객체의 위치에 따라 아래의 [그림 3-7]과 같은 경우로 나누어 볼 수 있다.



[그림 3-7] 중복 객체의 정제 대상이 되는 경우 그 외의 분할 영역 경계선 상의 중복 객체들에 대한 공간 조인은 [그림 3-7]의 A 영역 이외의 분할 영역을 가지는 복제 서버에서 수행하게 된다. 이러한 중복 객체를 고려한 여과 단계는 기존의 MBR 만을 사용한 여과 단계보다 복잡해지나 복제 서버의 분할 영역에 속하는 대부분의 객체가 [그림 3-7]의 a)와 같은 경우에 해당되며 나머지의 경우도 간단한 부동 소수점 연산만을 필요로 한다. 따라서, 여과 연산에 비해 훨씬 비용이 많이 드는 중복 객체에 대한 정제 연산을 줄임으로써 전반적인 공간 연산 비용을 줄일 수 있게 된다. 이러한 여과 단계의 알고리즘은 아래와 같다.

여과 단계를 거친 후보 쌍(Candidate pairs)들은 각 노드에서 병렬적인 정제 과정을 통해 조인 결과를 얻게 된다. 본 기법에서는 정제 단계 시 CPU 연산 비용을 줄이기 위해 보편적으로 사용되는 plane-sweep 알고리즘[1]을 사용한다.

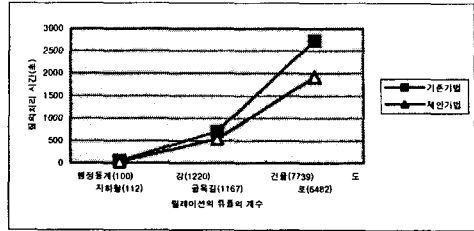
이러한 여과 단계와 정제 단계를 거쳐 얻어진 최종

```
SpatialJoin(R1, S1 : R NODE);
FOR (all ES1 ∈ S1) DO
  FOR (all ER1 ∈ R1 with ER1.rect ∩ ES1.rect ≠ ∅) DO
    IF (R1 is a leaf page) THEN ("S1 is also a leaf page")
      IF (ER1.center ∈ A1 and ES1.center ∈ A1) THEN
        Output(ER1, ES1)
      ELSE IF (ER1.rect ∈ A1 and ES1.rect ∈ A1) THEN
        Output(ER1, ES1)
      ELSE IF (ES1.rect ∈ A1 and ER1.rect ∈ A1) THEN
        Output(ER1, ES1)
      ELSE IF (ER1.center ∈ A1 and ER1.rect ∈ A1 and
        ES1.rect ∈ A1) THEN
        Output(ER1, ES1)
      END
    ELSE
      ReadPage(ER1.ref); ReadPage(ES1.ref);
      SpatialJoin(ER1.ref, ES1.ref)
    END
  END
END
END; END SpatialJoin.
```

결과 레코드들은 공간 조인 질의를 요청받은 복제 서버로 전송되며, 결과 레코드들을 받는 즉시 최종 클라이언트 사용자에게 전송하게 된다. 이러한 결과 레코드들은 여과 단계에서 이미 중복 공간 조인 객체들이 제거되었으므로 중복 결과 여부를 판별하기 위한 추가적인 연산없이, 즉시 사용자에게 전송하면 되므로 빠른 사용자 응답시간을 만족시킨다.

4. 성능 평가

본 장에서는 비공유 데이터베이스 클러스터에서 본 논문에서 제안한 분할 기법을 사용한 병렬 공간 조인 기법과 일반적인 분할 기반의 병렬 조인 기법의 성능을 질의 처리 속도를 기준으로 비교한다. 본 평가는 인하대학교 데이터베이스 연구실에서 개발한 비공유 공간 데이터베이스 클러스터를 위한 프로토타입 시스템 GMS CLUS 에서 구현되었다.



위의 결과를 보면 제안 기법이 기존의 병렬 공간 조인 기법보다 더 나은 성능을 보인다는 것을 알 수 있으며, 이는 제안 기법이 작업 할당, 중복 연산 및 결과 제거 등의 추가적인 연산을 수행하지 않기 때문이다.

5. 결론 및 향후 연구

본 논문은 비공유 공간 데이터베이스 클러스터 환경에서 영역 분할 기반의 병렬 공간 조인 기법을 제안하였다. 제안 기법은 병렬 수행을 위한 작업 생성 및 할당 단계가 필요 없으며, 중복 정제 연산 및 중복 결과 제거 연산을 제거함으로써 기존의 병렬 공간 데이터베이스 시스템의 병렬 공간 조인 기법보다 향상된 성능을 보인다.

향후 연구 과제로는 클러스터 노드간에 중복 저장된 릴레이션의 복사본(Replica)이나 공간 데이터의 영역 분포를 고려한 영역 분할 등 다양한 요소들을 이용한 최적의 병렬 공간 조인 계획 생성에 관한 연구가 필요하다.

6. 참고 문헌

[1] T. Brinkhoff, H. P. Kriegel, R. Schneider, B. Seeger, "Efficient Processing of Spatial Joins Using R-Trees", Proc. of Int. Conf. on Management of Data, ACM SIGMOD, pp237-246, 1993.  
 [2] T. Brinkhoff, H. P. Kriegel, R. Schneider, B. Seeger, "Parallel Processing of Spatial Joins Using R-Trees", Proc. of Int. Conf. on Management of Data, ACM SIGMOD, pp237-246, 1993.  
 [3] E. G. Hoel, H. Samet, "Data-Parallel Spatial Join Algorithms," Proc. of Int. Conf. on Parallel Processing, pp.227-234, 1994.  
 [4] J. D. Kim, B. H. Hong, "Parallel Spatial Join Algorithms Using Grid Files," Proc. of Int. Symp. On DANTE'99, pp.127-135, 1999.  
 [5] M. L. Lo, C. V. Ravishankar, "Spatial Joins Using Seeded Trees," Proc. of Int. Conf. on Management of Data, ACM SIGMOD, pp209-220, 1994.  
 [6] J. M. Patel, D. J. Dewitt, "Partition based spatial merge join," Proc. of Int. Conf. on Management of Data, ACM SIGMOD, pp.259-270, 1996.  
 [7] T. Brinkhoff, H. P. Kriegel, R. Schneider, B. Seeger, "Multi-Step Processing of Spatial Joins" Proc. of Int. Conf. on Management of Data, ACM SIGMOD, pp197-208, 1994.  
 [8] Preparata. F. P. and Shamos, M. I., "Computational Geometry," Springer, 1985  
 [9] 김진덕, 서영덕, 홍봉희, "고정 그리드를 이용한 병렬 공간 조인의 태스크 할당에 관한 연구", 한국정보처리학회 논문지, 제 8 권, 제 4 호, pp. 347-360. 2001