

# 맵 서버 컴포넌트를 위한 공유 캐싱의 설계 및 구현

박경미<sup>0</sup>, 안경환<sup>\*\*</sup>, 홍봉희<sup>\*\*</sup>

<sup>0</sup>부산대학교 GIS학과, <sup>\*\*</sup>컴퓨터공학과

{kmpark, khan, bhong}@hyowon.cc.pusan.ac.kr

## Design and Implementation of Shared Caching for MapServer Component

Kyung-Mi Park<sup>0</sup>, Kyoung-Hwan An<sup>\*\*</sup>, Bong-Hee Hong<sup>\*\*</sup>

<sup>0</sup>Dept. of GIS, <sup>\*\*</sup>Dept. of Computer Engineering, Pusan National University

### 요 약

이 논문은 OGC에서 제안하는 3계층 구조의 웹 매핑 테스트베드(Web Mapping Testbed) 환경에서 각 계층간의 통신량과 데이터 변환량으로 인한 속도 저하 문제를 해결하고자 한다. 이를 위해, 다수의 클라이언트가 일정 시간 내에 공통으로 접근하는 지도 영역에 대한 미들웨어 컴포넌트들의 중복된 작업 수를 줄이기 위해 맵 서버 컴포넌트들 간에 공유 캐시를 사용하는 구조를 설계 및 구현한다.

이 논문에서 제시하는 미들웨어의 캐시 구조는 전체 데이터 세트의 영역을 일정 간격의 그리드 셀로 나누어 관리함으로써 캐시 단위의 관리 비용을 줄이고 캐시를 효율적으로 사용한다. 또한, 기존의 교체 전략 기법을 적용하여, 다수의 클라이언트의 관심 대상인 영역을 중심으로 캐시를 유지하기 위해 접근 빈도수와 최근 사용 시간을 고려한 교체 전략을 사용한다.

### 1. 서론

현재 인터넷 GIS 분야는, ISO나 OGC 등의 국제 표준기구에서 제시하고 있는 상호 운용성을 지원하기 위한 3계층 구조의 인터넷 GIS로 나아가고 있다. 특히 OGC의 웹 매핑 테스트베드는 OpenGIS의 표준 인터페이스를 통해서 다양한 데이터 소스에 접근하는 구조의 인터넷 GIS를 위해 제안되었다[1]. 그러나, 웹 매핑 테스트베드는 표준 인터페이스를 통한 데이터 접근만을 목적으로 하기 때문에 각 계층간의 통신량과 데이터 변환량으로 인한 사용자 응답 시간의 지연 문제는 고려하지 않고 있다.

이러한 3계층 구조의 인터넷 GIS 환경에서의 속도 저하 문제점은 미들웨어 계층에서의 캐싱을 통해 해결할 수 있다. 이 논문에서는 다수의 클라이언트가 일정 시간 내에 일부 지역을 공통으로 접근하는 경우 미들웨어 컴포넌트들의 중복된 데이터 통신과 변환량을 줄이기 위해 맵 서버 컴포넌트들 간에 캐시를 공유하는 구조를 설계하고 구현한다.

이 논문에서 제시하는 미들웨어의 캐시 구조는 전체 데이터 세트의 영역을 일정 간격의 그리드 셀로 나누어 하나의 셀을 캐시 단위로 이용하며, 각 셀에 대한 인덱스를 유지하여 캐시를 효율적으로 관리한다. 또한, 여러 클라이언트의 질의를 고려하여 접근 빈도가 높은 영역의 공간 데이터를 중심으로 캐시에 유지하기 위한 교체 전략 기법을 사용한다. 또한, 최근 사용 시간을 추가 정보로 이용하여 캐시의 효율성을 높인다.

이 논문의 구성은 다음과 같다. 2장에서는 기존의 캐싱에 관한 연

구를 살펴 보고, 3장에서는 이 논문의 대상 환경과 문제점을 살펴 본다. 4장에서는 이 논문에서 제시하는 공유 캐시를 사용하는 미들웨어 구조 및 전체 시나리오와 캐시 교체 전략에 대해 설명한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해 기술한다.

### 2. 관련연구

기존의 공간 데이터 캐싱에 관한 연구로는 클라이언트/서버 환경에서 클라이언트의 캐시 적중률을 높이기 위해 사용자 접근 유형을 고려한 공간 근접성 기반의 알고리즘[2][4]과 서버에서의 디스크 I/O를 줄이기 위해 질의간 위치 관련성에 따라 질의 스케줄링을 하는 방법[5]이 있다. 그러나, 이러한 기법들은 3계층 구조의 인터넷 GIS 환경에서 여러 클라이언트를 고려해야 하는 미들웨어 단계의 캐싱에는 적용하기 어렵다. 왜냐하면, 공간 관련성에 따른 캐시 전략은 하나의 클라이언트의 경우 공간 근접성의 기준을 현재 질의로 할 수 있으나, 여러 클라이언트의 경우 공간관련성의 기준을 결정할 수 없기 때문이다. 따라서, 여러 클라이언트들을 고려한 캐시 전략이 필요하다.

기존의 미들웨어 캐싱에 관한 연구[3]는 단일 시스템 내에서 관리하고 처리하는 구조로써, 컴포넌트 기반의 미들웨어에 적용될 수 없다. 왜냐하면, 웹 매핑 테스트베드 환경에서와 같이 컴포넌트 기반의 미들웨어는 클라이언트의 요구에 따라 생성된 하나의 프로세스로써 질의 수행이 끝나면 해제되기 때문에 하나의 컴포넌트 내에서의 캐싱은 의미가 없다. 따라서, 미들웨어 단계에서 컴포넌트들이 공유해서 사용할 수 있는 캐시 구조의 설계가 필요하다.

### 3. 대상 환경 및 문제 정의

이 논문의 대상 환경은 그림 1)의 a)와 같이 인터넷 기반의 클라이언트-미들웨어-데이터 소스 구조를 연구 대상으로 한다.

- 클라이언트 : 데이터 소스의 종류에 관계없이 미들웨어에서 제공하는 표준 인터페이스를 통해 공간 데이터를 접근한다. 미들웨어에 HTTP와 같은 인터넷 통신 프로토콜을 이용하여 원하는 영역의 맵을 요청하고 XML 데이터를 전송받는다. XML 데이터를 해석하여 기존의 웹 브라우저로 지도를 출력한다.
- 미들웨어 : 웹 서버, 맵 서버, 데이터 제공자로 구성되며 맵 서버나 데이터 제공자는 표준 인터페이스를 제공하기 위해 데이터를 변환한다. 웹 서버는 요청된 질의에 대하여 맵 서버에서 만들어진 XML 데이터를 클라이언트에 전송한다. 맵 서버는 해당 데이터 소스를 위한 데이터 제공자를 이용하여 데이터 소스로부터 영역 질의의 결과를 전송 받아 XML 데이터로 변환한다.
- 데이터 소스 : 기존의 GIS 서버로서 고유의 통신 프로토콜을 가지기 때문에 서로 다른 인터페이스로 지리 정보를 제공한다. GIS 서버는 공간 데이터에 대한 영역 질의의 결과를 효율적으로 생성하기 위해 공간 데이터의 인덱스를 구성할 수 있으며, 요청된 질의 영역에 대해 벡터 형태의 결과를 미들웨어로 전송한다.

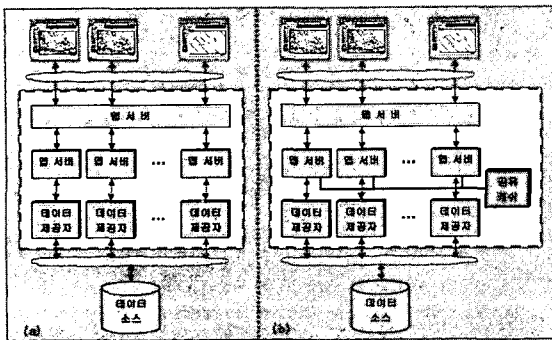


그림 1 3계층 구조의 인터넷 GIS 환경

이러한 환경에서 어떤 기간 동안에 질의 집중 지역이 발생할 경우, 즉 다수의 클라이언트들의 일정 영역에 대한 맵 요청이 많은 경우에 각 클라이언트에 대한 맵 서버들은 각각의 데이터 제공자를 통해 데이터 소스로부터 질의 결과를 받아 XML 데이터를 생성하게 되므로, 데이터 소스의 중복적인 질의 처리와 데이터 제공자들간의 중복된 데이터 변환이 일어나게 된다.

일정 시간 동안의 질의  $Q_1, \dots, Q_n$ 에 대해, 그림 1)의 (a)의 구조에서는  $n$ 개의 데이터 제공자가 생성되며, 수식(1)과 같이 질의 결과 크기의 합에 해당하는 데이터 통신량 및 변환량이 발생한다.

$$\sum_{i=1}^n (R(Q_i)) \quad (\text{단, } R(Q_i) \text{는 질의 } Q_i \text{에 대한 결과 데이터 길이}) \quad \text{--- (1)}$$

그림 1)의 (b)와 같이 맵 서버 컴포넌트 간에 공유 캐시를 사용하는 경우, 생성되는 데이터 제공자는  $n$ 개 이하이며 질의 집중 지역에 대한 영역 질의가 많은 경우 그 수는 현저하게 떨어진다.

$$\sum_{i=1}^n (SC(Q_i) - TC(T(Q_i))) \quad \text{--- (2)}$$

(단,  $SC(Q_i)$ 는 질의  $Q_i$ 에 대한 해당 셀의 집합,

$TC(T(Q_i))$ 는 질의가 들어오는 시간  $T(Q_i)$ 에 캐시된 셀의 집합)

데이터 통신량 및 변환량은 평균 질의 영역의 크기와 셀의 크기에

따라 달라진다. 셀의 크기가 평균 질의 영역의 크기보다 어느 정도 크다고 가정하면, 각 질의 영역에 해당하는 셀의 수는 4 이하라고 할 수 있다. 일정 기간 동안 어떤 지역에 대해 집중될 경우 수식(2)에서 해당 셀과 캐시된 셀의 차, 즉 캐시되지 않은 셀의 수는 시간에 따라 줄어들게 된다.

### 4. 캐시 구조

이 논문에서 제시하는 미들웨어의 캐시 구조는 그림 2)와 같이 공유 캐시, 캐시 관리자, 맵 서버로 구성된다.

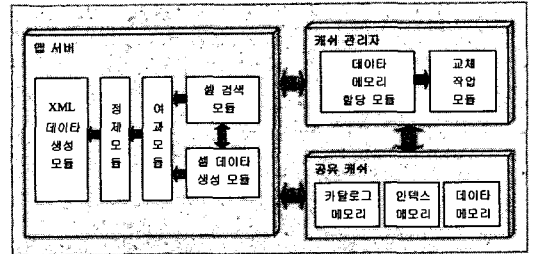


그림 2 미들웨어 캐시 구조

#### 4.1 공유 캐시의 구조

공유 캐시는 그림 3)과 같이 카탈로그 메모리, 인덱스 메모리, 데이터 메모리로 구성되며, 메모리 맵 파일과 같은 공유 메모리로 구현한다

- 카탈로그 메모리 : 데이터 소스의 메타 정보와 캐시 관리를 위한 셀 분할 정보로 구성된다. 전체 데이터 세트의 영역 정보로 전체 영역의 시작과 끝 좌표(StartX, StartY, EndX, EndY), 각 레이어별로 레이어 이름, 레이어를 구성하는 칼럼에 대한 정보 등의 레이어의 메타 정보가 저장된다. 셀 분할 정보로 전체 영역에 대한 X축/Y축 셀 분할 수(XRes, YRes), 분할된 셀의 X축/Y축 길이(CellSizeX, CellSizeY)를 유지한다.

- 인덱스 메모리 : 각 레이어별로 분할된 셀 데이터를 관리하기 위한 인덱스로 구성된다. 각 레이어에 대해 XRes x YRes의 2차원 배열 형태로 셀에 대한 인덱스가 저장된다. 각 셀별로 NOTCACHED, RESERVED, ALLOCATED, CACHED의 캐시 플래그를 두고 셀의 캐시 상태를 유지하며, 캐시된 셀에 대해서는 사용자수와 접근 빈도수를 유지하여 교체 작업을 위한 정보를 제공한다.

- 데이터 메모리 : 각 캐시된 셀의 실제 데이터가 저장되는 메모리로서 맵 서버의 셀 데이터 생성 모듈에서 생성된 셀 데이터에 대해 캐시 관리자에 의해 그 길이만큼의 공간이 할당된다. 데이터 메모리의 여유 공간이 없을 경우 캐시 관리자에 의해 교체 작업이 수행되어진다.

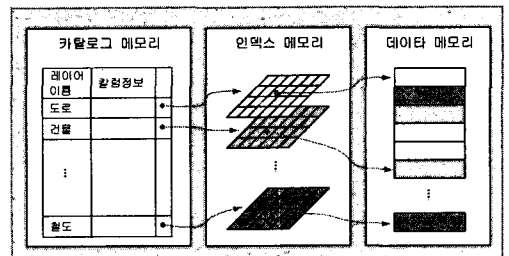
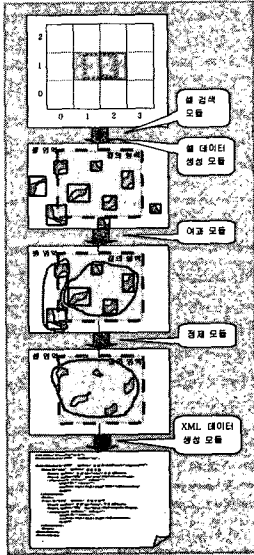


그림 3 공유 캐시의 구조

### 4.2 맵 서버의 구조

맵 서버는 클라이언트가 요구하는 맵 영역에 대해 셀 검색 모듈, 셀 데이터 생성 모듈, 여과 모듈, 정제 모듈, XML 데이터 생성 모듈을 통해 결과 데이터를 생성한다.

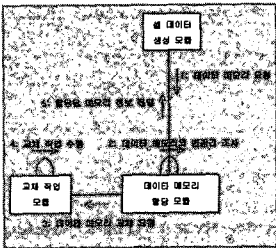


- 셀 검색 모듈 : 질의 영역에 해당하는 셀을 조사하여 캐쉬된 셀들과 캐쉬되어야 할 셀들을 검색한다.
- 셀 데이터 생성 모듈 : 캐쉬되어야 할 셀에 대해 데이터 제공자를 통해 셀 데이터를 생성하고 캐쉬 관리자에 의해 할당 받은 데이터 메모리의 공간에 저장한다.
- 여과 모듈 : 질의 영역에 해당하는 각 셀 데이터에서 객체들의 MBR을 비교하여 후보 객체를 추출한다.
- 정제 모듈 : 여과 모듈에서 만들어진 중간 결과로부터 질의 영역 내에 완전히 포함되는 객체를 추출한다.
- XML 데이터 생성 모듈 : 클라이언트가 요청한 맵을 위한 XML 데이터로 만든다.

그림 4 맵 서버의 구조

### 4.3 캐쉬 관리자의 구조

캐쉬 관리자는 데이터 메모리 할당 모듈과 교체 작업 모듈로 구성된다.



- 데이터 메모리 할당 모듈 : 맵 서버의 셀 데이터 생성 모듈에서 요구하는 데이터 메모리의 공간을 할당하고 해당 셀의 인덱스에 할당 정보를 저장한다.
- 교체 작업 모듈 : 데이터 메모리 할당 모듈에서 데이터 메모리의 공간이 부족한 경우 공간 확보를 위해 데이터 메모리의 교체 작업을 수행한다.

교체 작업을 위해 각 캐쉬된 셀의 접근 빈도수, 캐쉬에 로딩된 시간, 최근 사용 시간 정보 등의 교체 작업 정보를 이용하여 교체 값을 계산하고 교체 값이 가장 큰 순서로 제거한다.

#### 4.3.1 교체 전략

이 논문에서 사용하는 교체 전략은 여러 클라이언트들이 공통적으로 접근하는 영역의 데이터를 캐쉬에 유지하기 위해 접근 빈도수를 고려한다. 또한, 일정 시간 내에 집중률이 높은 데이터를 캐쉬에 유지하기 위해 최근 사용 시간을 추가 정보로 이용하여 교체 값을 결정할 필요가 있다.

$$C_x RP = \alpha \frac{1}{F_x} + (1-\alpha) \frac{T_{Current} - T_{Rx}}{T_{Current} - T_{Sx}} \quad \text{--- (3)}$$

교체 함수  $C_x RP$ 는 하나의 셀 데이터  $C_x$ 가 캐쉬에 남아있는 시간, 즉 셀이 캐쉬된 시점( $T_{Sx}$ )과 현재의 시점( $T_{Current}$ )의 차이에 대한 마지막 사용된 시간( $T_{Rx}$ ) 후부터 현재까지의 시간의 비율을 사용한다. 이것은, 셀의 접근 빈도수( $F_x$ )가 높지만 현재의 시점에서 오랫동안 사용되지 않으면서 캐쉬에 상주할 수 있는 가능성을 배제하고 최근에 사용이 빈번한 셀 데이터를 캐쉬함으로써 캐쉬의 효율성을 높인다.  $\alpha$

값은 접근 빈도수와 최근 사용 시간을 얼마나 고려할 것인가에 따른 가중치로 이용될 수 있으며, 실험을 통해 최적의 값이 결정될 수 있다.

### 4.4 질의 결과 생성 시나리오

앞에서 설명한 구조에서 클라이언트의 맵 요청에 대해 질의 결과를 생성하는 시나리오는 그림 5와 같다. 여기서, 캐쉬된 셀이 없는 경우, 즉 캐쉬된 셀들만으로 질의 수행이 가능할 때 중간 과정인 셀 데이터 생성 모듈과 캐쉬 관리자 모듈들의 작업은 필요 없게 된다.

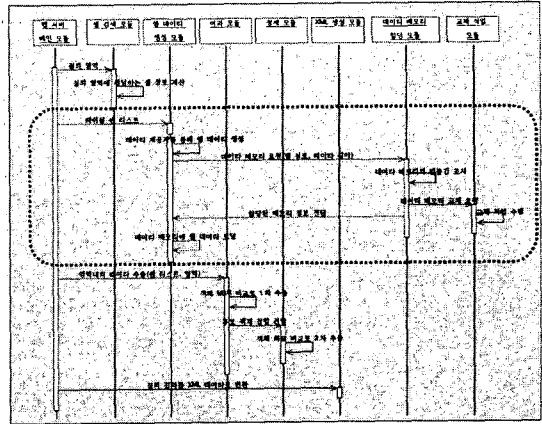


그림 5 질의 결과 생성 시나리오

## 5. 결론 및 향후 연구

이 논문은 3계층 구조의 인터넷 GIS 환경에서 미들웨어 컴포넌트 간에 공유 캐쉬를 사용하여 데이터 통신량과 변환량을 줄이기 위한 캐쉬 구조를 설계하였다. 전체 데이터 세트의 영역을 일정한 간격의 그리드 셀로 나눔으로써 캐쉬를 효율적으로 관리한다. 또한 접근 빈도수와 최근 사용 시간을 고려한 교체 전략을 사용하여 최근에 사용 빈도가 높은 영역의 공간 데이터를 캐쉬에 유지하도록 한다.

향후 이 논문에서 설계한 구조를 이용한 실험 평가가 요구된다. 이 구조에서 효율성을 높일 수 있는 캐쉬의 일계 크기와 캐쉬 셀의 적정 크기를 결정하기 위해 그 변화에 따른 실험 결과의 검토가 필요하다. 또한, 교체 함수의 가중치 변화에 따른 캐쉬 적중률의 비교 실험을 하고자 한다.

## 6. 참고 문헌

- [1] Open GIS Consortium, The Web Mapping Testbed Public Page, 2000 <http://www.opengis.org/>
- [2] S.Dar, M.J.Franklin, B.T.Jonsson, D.Srivastava, and M.Tan, "Semantic Data Caching and Replacement", In Proceedings of the 22nd VLDB Conference, 1996
- [3] L.M.Haas, D.Kossmann and I.Ursu, "Loading a Cache with Query Results", In Proceedings of the 25th VLDB Conference, 1999
- [4] 조대수, 안경환, 홍봉희, "인터넷 지리정보서비스의 성능 개선을 위한 클라이언트 캐쉬 알고리즘", '99 한국 데이터베이스 학술대회 논문집, 제 15권 1호, 99. 2.
- [5] 박영민, "디클러스터된 공간 데이터베이스에서 다중 질의의 병렬 처리", 부산대학교 대학원 컴퓨터공학과 석사 학위 논문, 2000