Journal of Digital Contents Society Vol. 10 No. 3 Sept. 2009(pp. 395-400)

지리공간 웹 기반 서비스에서 응답시간을 향상시키기 위한 효율적인 캐싱 기법

문진용*

요 약

인터넷 서비스가 광범위하게 제공되면서 혁신적으로 웹 GIS가 발달하게 되었다. 그러나 현실적으로 인터넷 지리 정보 시스템에서는 대용량의 공간 데이터의 전송에 따른 사용자 응답 시간 지연, 네트워크의 트래픽, 그리고 서버 과부하 문제가 발생한다. 본 논문에서는 이와 같은 문제를 해결하기 위하여 클라이언트 측의 캐시 알고리즘을 제안하고자 한다. 제안된 알고리즘은 단위 시간과 공간 근접성의 개념을 이용하여 기존의 연구를 개선한다. 그리고 효율성의 향상 정도를 측정하기 위해 성능 평가를 실시하고 결과를 분석한다. 성능 평가 결과, 공간 데이터의 질의 시에 기존의 알고리즘에 비하여 적중률에 있어서 보다 좋은 성능을 나타내었다.

An Efficient Caching Scheme to Enhance the Response Time of Geospatial Web Services

Jin-Yong Moon

Abstract

The wide spread of the Internet service has brought in web GIS development. But, there are problems of the user access latencies, the network traffic, and the server overload in Internet Geographic Information System because spatial data are transferred in large volumes. In this paper, I propose a cache algorithm on client side to solve the above problems. The proposed algorithm demonstrates the performance improvement over known studies by utilizing unit time and spatial proximity. In addition, this paper conducts a performance evaluation to measure the improvement in algorithm efficiency and analyzes the results of the performance evaluation. When spatial data queries are conducted, according to our performance evaluation, hit rate has been improved over the existing algorithms.

Keywords : Cache, Geospatial, Performance Evaluation

1. 서 론

지리 정보 시스템(Geographic Information System; 이하 GIS) 분야에서는 인터넷 기술을 접목하여 인터넷을 통해 지리 정보 서비스를 제공하기 위한 인터넷 GIS에 관한 연구가 활발히 진행되고 있다 [9, 11]. 인터넷 GIS는 웹을 기반으로 하고 있기 때문에 웹 브라우저만 설치되어 있으면 플랫폼 독립적으로 일반 사용자들은 공간과 시간의 제약을 받지 않고 손쉽게 사용할 수 있는 장점을 가지고 있다. 하지만, 현재와 같이 폭발적으로 증가하는 네트워크의 트래픽 상황에서 상대적으로 용량이 큰 공간 데이터의 특성 때문에 인터넷 GIS 사용자들에게 응답 시간의 지연, 네트워크의 과도한 트래픽, 그리고 서버의 과부하 문제가 발생하게 된다.

본 논문에서는 인터넷 GIS의 문제점 중에서 위와 같은 문제를 해결하기 위해 적중률을 높일 수 있는 캐시 알고리즘을 제시하고자 한다. 사용자 응답 시간의 지연은 인터넷 GIS 클라이언트에서의 캐시의 적중률을 높임으로써 서버와 클라이언트간의 통신 트래픽 문제를 효과적으로 개선할

※ 제일저자(First Author) : 문진용
접수일:2009년 07월 04일, 완료일:2009년 9월 15일
* 극동정보대학 방송영상미디어과
jmoon@kdc.ac.kr

수 있다 [4, 7, 10].

제시하는 알고리즘은 기존의 방식에 비해 공간 근접성과 단위 시간의 개념을 이용한다는 점에서 차별화 될 수 있다. GIS 분야에서 이미 잘 알려진 특성으로 접근되는 객체들 간의 지역성이 있다는 점이다. 인터넷 GIS를 위한 캐시를 설계할 때, 공간 근접성에 대한 추가 정보를 이용하면 기존의 방식에 비해 캐시 적중률을 높일 수 있다. 캐시 적중률의 향상은 결과적으로 클라이언트와 서버 간의 접속 회수 및 통신량을 줄임으로써 사용자 응답 시간 지연을 효과적으로 개선할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 인터넷 GIS에서 사용되고 있는 공간 데이터의 특성과 기존의 대체 알고리즘에 대한 연구를 살펴보고, 3장에서 연구 환경 및 제안된 알고리즘을 상세하고, 4장에서는 기존의 알고리즘과의 성능 평가를 통해 기존의 알고리즘에 대한 성능 개선 여부를 검증하여 본다. 마지막으로 5장에서는 결론 및 향후 연구 과제에 대하여 언급한다.

2. 관련 연구

2.1. 공간 데이터의 특성

인터넷 GIS에서 사용되는 공간 데이터는 공간상의 지리 표현 객체로 복합적으로 표현되는 가변 길이 데이터이다 [3, 7, 9]. 이 공간 데이터는 기하 데이터와 위상 데이터로 분류된다. 기하 데이터는 공간 데이터를 표현하는 것으로 라스터나 벡터 포맷을 갖는다. 반면 위상 데이터는 이러한 기하 데이터 사이의 공간 관계를 표현하는 것으로 인접, 포함 등의 공간 관련 정보를 갖는다. 이와 같은 공간 데이터들은 그 형태와 무관하게 대부분이 비정형이며 그 크기가 방대하다는 특성을 가진다. 또한, 인터넷 GIS는 공간 데이터 외에 이에 관련된 특징을 포함하는 속성 데이터를 관리해 주어야 한다.

2.2. 대체 알고리즘

현재 인터넷 캐시를 위해 연구된 기존의 알고리즘들은 다음과 같이 크게 3가지로 분류할 수 있다. 본 논문에서는 제안된 알고리즘과의 성능 평가를 4장에서 하여본다.

첫째, LRU(Least recently used), LFU(Least frequently used)와 같은 기존의 대체 알고리즘들을 가변크기 객체라는 특수성을 갖는 인터넷 분야에 응용・적용하는 것이다 [1, 3].

LRU는 크기가 서로 다른 객체를 처리하기 위해 기존의 LRU 기법의 확장이라 할 수 있다. 이 기법은 새롭게 들어오는 인터넷 객체의 여유 공간을 마련하기 위하여 가장 최근에 사용되지 않은 객체부터 삭제하는 방식이다. 이렇게 하면, 삭제되는 객체는 0개에서부터 수많은 객체들이 캐시에서 삭제될 수 있다 [4, 6].

LRU-K는 각 객체의 마지막 K번의 참조 시간을 기억한다. 그리고 잦은 사용 빈도를 가진 것과 아닌 것을 구분함으로써 LRU보다 나은 성능을 제공한다. 즉, 각 객체들의 참조되는 시간 간격을 따로 저장해서 이를 다시 알고리즘에 반영하고, 이렇게 자치적으로 파라미터를 바꿈으로써 능동적으로 동작이 가능하고, 한 객체의 현재의 참조 빈도와 과거의 참조 빈도를 구분함으로써 변화하는 참조의 형태에 적응할 수 있다 [4, 10].

둘째, 키에 기반을 둔 대체 알고리즘들이다. SIZE는 인기가 있는 작은 크기의 객체가 큰 객체의 삽입으로 제거되는 상황을 막기 위해 새로 삽입되는 객체의 여유 공간이 생길 때까지 크기가 가장 큰 객체부터 차례로 제거하는 전략이다 [1, 12].

셋째, 전송비용, 객체의 유효기간 등과 같은 비용에 기반을 둔 알고리즘이다. GD(Greedy Dual)-Size 알고리즘은 전송비용/객체크기가 가장 낮은 객체를 캐시 공간에서 삭제한다 [3].

3. 알고리즘의 설계

3.1. 연구 환경 및 가정

본 논문에서는 인터넷 기반의 클라이언트/서버 모델의 인터넷 GIS를 연구의 대상으로 한다. 서버에서 관리하는 공간 데이터는 벡터 데이터로써, 클라이언트로 전송되는 데이터 역시 벡터 데이터이다. 서버의 공간 데이터 관리를 위한 공간 인덱스는 고정 그리드 파일이 사용되고 있다고 가정함으로 각 그리드 셀 내의 공간 데이터는 하나의 페이지에 저장되며, 여러 셀에 겹치는 공간 데이터는 객체 자체가 쪼개어져 저장되어 중복 참조

되는 공간 데이터는 존재하지 않는다.

클라이언트는 공간 처리 및 분석 등의 복잡한 연산보다는 단순히 지도 보기를 주된 작업으로 하기 때문에, 대부분의 사용자 응답 시간 지연은 지도 출력의 지연으로 발생한다. 클라이언트는 사용자가 화면으로 출력하려는 영역과 겹치거나 포함하는 셀 데이터를 서버에 요청한다. 각각의 셀에 포함된 데이터는 서버에서 하나의 페이지에 할당되어 저장되어 있으므로, 셀 단위의 데이터 전송은 페이지 단위의 전송을 의미한다.

또한, 클라이언트 캐시의 도입으로 발생되는 공간 데이터 중복으로 인한 일관성 유지 문제와 객체 식별자 유지 문제는 다루지 않는다. 일관성 유지 문제는 기존 연구의 인터넷 캐싱 일관성 유지 기법을 적용하여 해결하는 것으로 가정한다. 또한, 객체 식별자 유지는 특정 영역의 공간 데이터를 화면에 출력하는 것을 기본 기능으로 하는 인터넷 GIS에서는 중요한 문제가 아니므로 이 논문에서는 언급하지 않는다.

3.2. 제안된 캐시 알고리즘

본 논문의 알고리즘은 (알고리즘 1)과 같이 기본적으로 LFU의 개념을 사용한다. 하지만, 단위 시간의 개념을 사용하여 우선 LRU-K처럼 잦은 사용 빈도를 가진 페이지와 그렇지 않은 페이지를 구분하기 위해 각 캐시의 참조 회수를 단위 시간 내에 행해진 것과 단위 시간 후에 행해진 것으로 구분해서 저장한 후, 대치될 캐시를 찾을 때 먼저 단위 시간 내에 행해진 참조 회수를 가지고 비교한다.

그리고 이 값이 같을 경우 단위 시간 후에 행해진 참조 회수를 가지고 다시 비교한다. 또한, 이를 위해 사용한 단위 시간을 LFU의 치명적인 단점인 영역 질의에서 캐시의 개수가 많아질 때 오히려 성능이 더 떨어지는 것을 보완하기 위해 사용한다. 즉, 단위 시간 내에 행해진 참조 회수를 가지고 비교하기 전에, 캐시에 있는 페이지 중 단위 시간 동안 참조되지 않은 것이 있는지를 검사한다. 만일 있다면, 이를 대치 대상으로 결정한다. 이렇게 함으로써 변화하는 참조의 형태에 적응할 수 있게 된다.

이렇게 해서도 대체 대상을 찾지 못하면, 가장 최근에 사용된 페이지를 선택한다. 왜냐하면, 인터넷 GIS에서는 영역 질의의 경우에 가장 최근에 사용된 페이지가 계속해서 참조되는 것이 아니라 과거에 참조된 페이지가 주기적으로 참조되는 패턴을 가지기 때문이다.

```
if(p is already in the cache)then
  if(t-time(p) > Unit_time)then
    increment Uncorrelated_Reference_Counter(p) by 1;
  else
    increment Correlated_Reference_Counter(p) by 1;
  time(p)=t;
//end_if
else
  min=MAX_NUMBER;
  for(all pages q in the cache)
    if(t-time(q) > Unit_time)then
      victim=q;
      min=MIN_NUMBER;
    //end_if
    if(Correlated_Reference_Counter(q) < min)then
      victim=q;
      min=Correlated_Reference_Counter(q);
    //end_if
    else if((Correlated_Reference_Counter(q) = min)and
(Uncorrelated_Reference_Counter(victim)
 <  Uncorrelated_Reference_Counter(q))or(time(victim)  <
time(q))then
      victim=q;
  //end_for
  fetch p into victim;
  if(t-time(p) > Unit_time)then
    increment Uncorrelated_Reference_Counter(p) by 1;
  else
    increment Correlated_Reference_Counter(p) by 1;
  time(p)=t;
//end_else
```

(알고리즘 1) 제안된 캐시 알고리즘

본 논문에서 제안한 알고리즘은 다음과 같다.

첫째 단계로 우선 검색해야 하는 공간 데이터를 가진 페이지 p가 캐시에 있는지를 검사한다. 그렇다면 이 참조가 단위 시간 내에 행해진 것인지 아닌지를 검사해서 그에 맞게 참조 계수를 증가시킨다.

둘째 단계로 캐시에 없다면 우선 캐시에 있는 페이지 중 단위 시간 동안 참조되지 않은 페이지를 찾는다. 만일 있다면 이를 교체시킨다. 하지만 없다면 단위 시간 내의 참조 횟수가 가장 적은 캐시나 단위 시간 후의 참조 횟수가 가장 적은 캐

시, 그리고 참조된 시간이 가장 오래된 캐시를 찾아서 이 캐시에 필요한 페이지를 대체한다.

셋째 단계로 교체가 끝나면 이 페이지로의 참조가 단위 시간 내에 행해진 것인지 아닌지를 검사해서 그에 맞게 참조 계수를 증가시킨다. 마지막 단계로 대체된 캐시의 시간을 참조될 때의 시간으로 지정한다.

4. 성능 평가

4.1. 실험 환경

본 논문에서는 성능 평가를 하기 위해서 HP xw6000 Workstation에서 운영체제로는 Fedora Core 6을 사용하였고, 컴파일러는 gcc 4.4.0을 사용하였다. 성능 평가용 공간 데이터는 국립지리원에서 DXF(Data eXchange Format) 포맷으로 경기도 전체를 1376개의 도엽으로 제작한 지적도를 사용하였다.

4.2. 성능 평가 결과

본 논문에서 비교 대상으로 LRU와 LRU-K 알고리즘을 선택한 이유는 LRU-K가 LRU를 개선한 방법이고, LRU와 LRU-K의 구현이 다른 알고리즘들에 비해 쉽기 때문이다. LRU-K는 K가 2인 경우를 예를 들었다. LRU-3도 실험해 보았으나 LRU-2와 별로 성능 차이를 보이지 않았다.

전체 페이지의 수는 가로×세로, 32×32, 1024로 가정하고, 각 페이지의 크기는 345×345로 가정한다. 이는 우리나라의 남한 지형만한 크기로써 앞에서 언급한 바와 같이 공간 데이터는 실제로 가변 길이 데이터이지만, 여기서는 실험의 편의를 위해 고정 그리드 인덱스를 사용하는 페이지로 제한한다.

본 논문의 알고리즘의 단위 시간은 캐시의 수나 다른 파라미터에 영향을 받지 않고, 단지 반경 r에 해당하는 페이지의 수에만 영향을 받음을 알 수 있었다. 따라서 단위 시간은 반경 r에 해당하는 페이지 수×1.5로 하였다.

(1) 영역 질의의 경우

영역 질의의 경우, 인터넷 GIS에서의 공간 근접성의 특성을 보임으로 중심점에서 먼 객체일수록 그 참조 확률이 떨어진다. 즉, 식 1이 성립하게 된다.

식 1

$$\text{참조 확률} = \frac{\alpha}{\text{중심점과 공간 객체 사이의 거리}}$$

여기서, α는 임의의 상수이다.

따라서 각 공간 객체에 할당할 수 있는 가중치는 식 2와 같다.

식 2

$$\text{가중치} = \beta(r - \text{중심점과 공간 객체 사이의 거리})$$

여기서, β는 임의의 상수이다.

이 가중치를 각 객체에 할당할 경우, 더 높을 성능을 보일 수 있다. 여기에서의 거리는 유클리드 거리이며 아주 정확한 가중치가 필요한 것이 아니라, 근사한 가중치만이 필요하고, 또한 수행 속도를 빠르게 하기 위해 중심점이 (X, Y)이고, 객체의 첫 번째 데이터의 좌표가 (x, y)라면, 거리 = |X - x|+|Y - y|가 된다. 따라서 이를 각각의 알고리즘에 모두 반영하면 보다 나은 성능을 보일 수 있다.

영역 질의시, 가장 최소한으로 필요한 캐시의 크기는 반경 r을 포함하는 객체의 수이다. 만약 이 최소한의 캐시의 크기보다 작게 할당을 하게 되면, 실험 결과는 많게는 15%까지 떨어지는 결과를 보였다. 따라서 최소한의 캐시의 크기는 식 3과 같다.

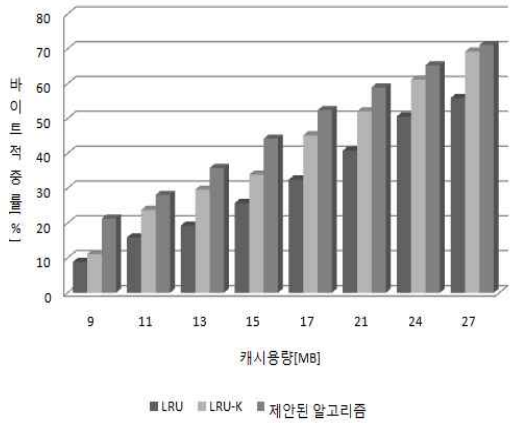
식 3

$$\text{캐쉬의 수} = \left\lceil \frac{\text{반경 } r}{\text{한 블럭의 가로의 크기}} \right\rceil + \delta$$

여기서 δ는 임의의 상수이다.

하지만, 본 논문에서는 같은 크기의 캐시에 대해 여러 반경에서 실험을 했으므로, 위의 수식을 적용하지는 않았다. (그림 1)은 반경 r을 변경시켜

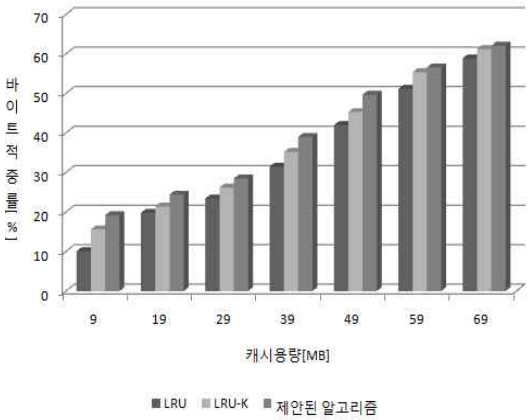
가면서 측정한 영역 질의시의 적중률의 평균치를 나타낸다. 본 논문의 알고리즘은 캐시의 수가 적을 때는 다른 알고리즘에 비해 성능이 월등히 우수했지만, 캐시의 수가 많아지면 LRU-K와 비슷해졌다.



(그림 1) 영역 질의시의 성능 평가 결과

(2) 점 질의의 경우

점 질의의 경우에는 임의의 좌표의 데이터를 검색하는 질의가 연속적으로 들어오는 경우를 가정하여 실험하였다. (그림 2)는 실험 결과를 나타내고, 각 알고리즘마다 그 차이가 영역 질의시보다 거의 없었다. 이것은 점 질의의 경우, 참조의 형태가 난수적인 형태라서 어떤 알고리즘이든지 최적의 상황이 될 수 없기 때문이다.



(그림 2) 점 질의시의 성능 평가 결과

5. 결론

본 논문에서는 인터넷 GIS를 위한 캐시 알고리즘을 제안하였다. 성능 평가 결과 제안된 알고리즘은 LRU나 LRU-K에 비해 보다 나은 성능을 발휘하였다. 본 논문의 알고리즘은 클라이언트 측 캐시의 크기나 다른 파라미터에 따라 그 성능이 크게 차이가 나지 않았다. LRU는 다른 알고리즘에 비교하여 그 성능 차이가 너무 심해서 인터넷 GIS와 같은 대용량의 공간 데이터를 처리해야 하는 상황에서는 적합하지 않음을 알 수 있었다.

향후 연구 과제로는 본 논문에서는 두 가지 형태의 질의만으로 성능 평가를 실시하였는데, 인터넷 GIS에서는 이외에도 많은 형태의 질의가 있으므로 향후 모든 질의에 적용할 수 있고, 자치적으로 동작할 수 있는 능동적인 캐시 알고리즘의 연구가 필요하리라 생각된다. 또한, 인터넷 GIS에서의 참조의 특성인 공간 근접성을 이용하여 단위 시간을 반경 r에 해당하는 페이지의 수×1.5로 일정하게 정의하였는데, 이 파라미터를 조정하여 능동적으로 동작할 수 있는 연구가 진행되어야 하겠다.

참 고 문 헌

[1] H. J. Bekker, I. Melve, and T. Verschuren, "Survey on Caching Requirements and Specifications for Pro totype," DESIRE European Project Deliverable D4.1.

[2] N. Basher, A. Mahanti, C. Williamson, and M. Arlin, "A Comparative Analysis of Web and Peer-to-Peer Traffic," Proc. of the 17th Int. WWW Conf., 2008.

[3] L. Cherkasoba and G. Ciardo, "Role of Aging, Frequency, and Size in Web Cache Replacement Policies," Proc. of the 6th Int. Symp. on Computers and Comm unications, pp. 3-15, 2001.

[4] K. Cheng and Y. Kambayashi, "Enhanced Proxy Caching with Content Management," Knowledge and Information Systems, 2002.

[5] J. Erman, A. Gerber, M. T. Hajiaghayi, D. Pei, and O. Spatscheck, " Network-Aware Forward Caching," Proc. of the 18th Int. WWW Conf., 2009.

[6] L. Rizzo and L. Vicisano, "Replacement Policies for a Proxy Cache," Technical Report RN/05/13, UCL-CS.

[7] Surveys: Internet Traffic Touched by Youtube, Light Reading, 2009.
[8] P. Trianfillou and I. Aekaterinidis, “ProxyTeller: a Proxy Placement Tool for Content Delivery under Performance Constraints,” Proc. of the 4th Int. Conf. on Web Information System Engineering, 2003.
[9] C. Torniai, S. Battle, and S. Cayer, "Discovering and Browsing Geotagged Pictures on the World Wide Web," Int. Workshop on Distributed Agent-Based Retrieval Tools, 2008.
[10] A. E. M. Taha and A. E. Kamal, “Optimal and Near Optimal Web Proxy Placement Algorithms for Net works with Planar Graph Topologies,” Proc. of the 23rd Int. Conf. on Distributed Computing Systems, 2003.
[11] Y. Teranishi, J. Kamahara, and S. Shimojo, “MapWi ki: A Ubiquitous Collaboration Environment on Sha red Maps,” Int. Symp. on Applications and the Inter net Workshops, pp. 146-149, 2006.
[12] S. Varadarajan, R. Harinath, J. Srivastava, and Z. L. Zhang, “Coverage-Aware Proxy Placement for Dynamic Content Management over the Internet,” Int. Conf. on Distributed Computing Systems Work shops, 2003.

문 진 용

1998년 : 건국대학교 대학원 (공학 석사)
2001년 : 수원대학교 대학원 (이학 박사)

2001년～현 재 : 극동정보대학 방송영상미디어과 부교수
관심분야 : 인터넷, 데이터베이스, 멀티미디어 등