韓國컴퓨터情報學會 論文誌
第16卷 第3號, 2011. 3.

2011-16-3-1-10

캐쉬메카니즘을 이용한 시맨틱 스키마 데이터 처리

김 병 곤*, 오 성 균**

Semantic schema data processing using cache mechanism

Byung-Gon Kim*, Sung-Kyun Oh**

요 약

네크워크상의 분산되어 있는 정보를 접근하는 온톨로지와 같은 시맨틱 웹 정보 시스템에서는 효율적인 질의 처리를 위하여 질의 응답 시간을 줄여주는 향상된 캐쉬 메카니즘을 필요로 한다. 특히, P2P 네트워크 시스템은 웹 환경의 기본적인 하부 구조를 이루고 있으며, 질의가 발생하면, 소스 피어(Peer)로의 데이터 전송량을 줄이는 문제가 효율적인 질의 처리의 중요한 부분이다. 전통적인 데이터베이스 캐쉬 메카니즘으로부터 현재의 웹 환경에 적합한 질의 메카니즘들이 연구되어 왔으며, 질의 처리 결과를 캐쉬하는 것은 입력 질의 요구를 빠른 시간에 바로 사용자에게 전달할 수 있다. 웹 환경에서는 시맨틱 캐싱 방법이 연구되어 왔으며, 이는 캐쉬를 의미적인 영역들로 이루어진 공간으로 관리하는 개념이며, 논리적인 캐싱 단위가 질의와 질의 결과이므로 웹 환경에서 적합한 개념이다. 본 연구에서는 온톨로지와 같은 시맨틱 웹 정보가 클러스터 단위로 여러 피어에 분산되어 있는 경우에 캐쉬 메카니즘을 이용하여 효율적인 질의 처리가 이루어지도록 하는 방법을 제시한다. 특히, 캐쉬를 유지하고 처리하는 방법으로 스키마를 이용한 캐쉬 데이터 필터링 방법과 온톨로지와 질의 결과의 유사도를 측정하여 캐쉬 대체 영역 선택에 사용하는 방법을 제시한다.

▸ Keyword : 온톨로지, 캐쉬메카니즘, 클러스터, 질의필터링, 캐쉬대체

Abstract

In semantic web information system like ontology that access distributed information from network, efficient query processing requires an advanced caching mechanism to reduce the query response time. P2P network system have become an important infra structure in web environment. In P2P network system, when the query is initiated, reducing the demand of data transformation to source peer is important aspect of efficient query processing. Caching of query and query result takes a particular advantage by adding or removing a query term. Many of the answers may

• 제1저자, 교신저자 : 김병곤
• 투고일 : 2011. 01. 08, 심사일 : 2011. 02. 10, 게재확정일 : 2011. 02. 13.
* 부천대학 e-비즈니스과(Dept. of e-business, Bucheon University)
* 서일대학 소프트웨어과(Dept. of software, Seoil University)
※ 본 논문은 2009년도 서일대학 학술연구비에 의해 연구되었음.

already be cached and can be delivered to the user right away. In web environment, semantic caching method has been proposed which manages the cache as a collection of semantic regions. In this paper, we propose the semantic caching technique in cluster environment of peers. Especially, using schema data filtering technique and schema similarity cache replacement method, we enhanced the query processing efficiency.

▸ Keyword : Ontology, Cache mechanism, Query Filtering, Cache Replacement

Ⅰ. 서 론

온톨로지와 같은 시맨틱 웹 정보가 각 피어마다 구축되어 있는 분산 환경에서 P2P방식으로 시스템을 운영할 때 시스템의 처리 성능은 피어들이 지니는 저장 자원의 효율적인 이용과 많은 연관성을 지닌다. 웹 문서를 저장하는 방식에 따라 효율이 결정되기도 한다[1]. 그러나, P2P 방식의 처리 환경에서 질의가 발생한 피어의 요구에 따라 다른 피어로부터 데이터가 전송되면 소스 피어의 메모리에 데이터를 캐싱할 수 있고 이는 추후에 유사한 질의가 발생하였을 때 네트워크 전송량을 최소화하는데 이용할 수 있다. 이러한 데이터 캐싱의 개념은 네트워크상의 데이터 전송량을 최소화 하면서 사용자에게는 빠른 응답시간을 제공할 수 있으므로 시맨틱 웹 환경과 같은 웹 환경에서의 데이터 검색에서 중요한 개념으로 사용될 수 있다.

데이터 처리를 위한 캐싱 기법은 운영체제와 데이터베이스, 정보 검색 시스템에 이르기까지 여러 환경에서 좀더 효율적인 시스템을 구축하기 위하여 사용되어 왔다[2,3,4]. 웹 환경에서의 캐싱은 운영체제나 데이터베이스 시스템에서 사용되는 페이지단위의 캐싱이나 튜플 단위의 캐싱과 같은 일반적 캐싱 방법을 그대로 적용하기는 적합하지 않다. 웹 환경에 더욱 적합한 개념의 캐싱은 미래에 다시 발생할 가능성이 있는 사용자의 질의와 질의 결과를 캐싱하는 방법이다. 이러한 캐싱 기법은 데이터베이스의 뷰의 개념과 유사하며 이를 좀더 인터넷 환경에 적합하도록 운영될 수 있도록 하는 것이며 갈수록 복잡해지는 인터넷 환경에서 필수적인 기술이라 할 수 있다.

본 연구에서는 시맨틱 웹이 각 피어에 구축된 환경에서의 P2P방식으로 데이터를 효율적으로 처리하기 위한 캐싱 기법을 연구하였다. 네트워크상의 피어들이 서로간의 유사성을 가지고 클러스터를 형성하고 있는 환경을 가정하여, 이러한 환경에서 같은 클러스터로부터 가져온 질의의 결과와 다른 클러스터에서 가져온 결과를 효율적으로 캐싱하고, 캐싱된 데이터와 데이터에 대한 스키마를 이용하여 질의를 처리하는 가장 효율적이고 적합한 방법을 찾도록 하였다.

2장에서는 전통적인 캐싱 기법에 관한 관련연구들을 살펴보고 이를 시맨틱 웹 환경에 적용하는 시맨틱 캐싱 방법을 소개하였으며, 3장에서는 이를 바탕으로 실질적인 클러스터 환경에서의 시맨틱 캐싱 기법을 소개하였다. 특히, 스키마 정보를 이용한 질의 필터링 기법과 문서 유사도에 의한 캐쉬 대체 방법을 제안하였다. 4장에서는 결론과 추후 과제를 언급하였다.

Ⅱ. 관련 연구

1. 캐싱기술분석

전통적인 클라이언트-서버시스템의 전송 단위는 페이지 혹은 레코드 단위이다. 이와 같이 페이지 단위의 캐싱을 하는 페이지 캐싱 방법은 운영체제나 데이터베이스 시스템에서 많이 사용되는 방법이다. 이는 클라이언트의 질의가 지역적으로 수행될 때 페이지 단위로 처리가 가능하다는 가정을 지닌다. 따라서, 요구된 정보가 클라이언트 캐쉬에 존재하지 않으면 모든 페이지에 대한 요구가 서버로 전송된다. 페이지 캐싱 방법은 클라이언트가 키워드를 기본으로 검색을 수행하고, 클라이언트는 서버에 존재하는 데이터 구조에 대한 정보를 지니지 않기 때문에 웹 환경의 검색 환경에서는 적합하지 않다.

튜플 캐싱에서는 각 레코드 단위로 정보를 유지하므로 페이지 캐싱에 비해서 더 나은 융통성을 제공할 수 있다. 튜플 캐싱은 웹상의 문서를 URL을 사용하여 접근할 때 사용할 수 있는 방법이다. 또한, 프록시 서버에서 웹 페이지들을 캐싱하는 경우에도 사용할 수 있다. 프록시 서버에서 최근에 사용된 웹 페이지들을 유지하고 URL을 사용하여 클라이언트가 요청한 내용을 재사용 가능하도록 하는 것이다.

튜플 단위의 캐싱의 경우에도 질의상 사용자의 요구사항에 요구 페이지의 URL이 포함되어 있지 않고 검색 폼 형태로 되어 있으며 해답의 URL은 미리 동적으로 변환되는 경우에는 효과가 없다. 이 경우에는 프록시 캐쉬가 필요한 정보를 얻기가 힘들다. 클라이언트 역시 지역 캐쉬가 질의에 대한 완벽한 대답을 제공했는지를 알아낼 수 없다. 결론적으로, 클라이언트는 질의를 수행하는 동안 캐쉬에 저장된 튜플들을 무

시하게 되며, 질의는 서버로 보내지고 모든 관련 튜플들이 반환되는 결과를 초래한다.

데이터를 질의할 때 페이지캐싱과 튜플 캐쉬의 이러한 단점들을 보완하기 위하여 시맨틱 캐쉬를 이용하는 방법이 여러 연구에 의하여 진행되어 왔다[5,6,7]. 이러한 접근 방법은 클라이언트 캐쉬를 의미적인 영역의 그룹으로 유지하도록 한다. 의미적 영역은 말 그대로 의미적으로 연관된 데이터들을 함께 그룹으로 묶는 개념이다. 데이터를 접근하거나 캐쉬를 교환하는 단위로 시맨틱 영역을 사용한다. 시맨틱 캐쉬를 사용하는 경우에 질의가 발생하면 질의는 로컬 영역의 캐쉬로부터 데이터를 가져오는 부분과 지역 캐쉬에 존재하지 않는 부족한 부분을 서버로부터 가져오기 위한 질의로 나누어져 처리된다. 캐싱의 단위를 물리적 단위가 아닌 실질적인 질의 결과를 사용하는 이러한 시맨틱 캐싱의 기본적인 개념이 웹 환경에서 질의 결과를 캐싱하는데 보다 적합한 특성을 지닌다.

이러한, 웹 캐쉬는 웹 브라우저의 지연시간을 줄이고 네트워크상의 대역폭 소비를 줄이고, 웹서버의 로드시간을 줄이기 위해서 많이 사용되는 기술이다. 클라이언트 노드들간의 데이터의 상호 공유를 원활히 하는 것을 목표로 한다. 현재 모든 노드의 웹 브라우저들은 브라우저가 수행한 웹데이터에 대한 지역캐쉬를 유지한다. 각 노드들은 각각 웹 브라우징과 웹 캐싱을 모두 수행한다. [5]에서는 클라이언트-서버 데이터베이스 시스템의 클라이언트쪽의 캐싱과 대체 방법을 제안하였고 기존의 페이지 캐싱과 튜플 캐싱과 비교하여 시맨틱 캐싱의 성능을 보여주었다. 논문에서 제안된 내용들은 P2P 환경에서 적용가능하다고 언급되었지만 구체적인 방법을 제시하지는 않았다. [6]연구에서의 시맨틱 캐싱에 대한 기본적인 아이디어는 질의의 결과와 더불어 질의 자체를 기억해놓는 것이다. 시맨틱 캐싱은 동적으로 변화하는 튜플들의 집합을 정의하여 사용하였다. 이와같이 일반적인 인터넷 환경에서의 웹 캐싱에 대한 연구는 많이 연구되었고, 이러한 연구들을 바탕으로 P2P 환경에서의 노드들간의 효율적인 질의 처리를 위한 웹 캐싱 방법에 대한 연구가 필요로 하는 시점이다. 초기의 웹 캐싱 연구들은 전통적인 전용 하드웨어를 사용하는 중앙 집중적인 환경에서의 웹 캐싱 방식을 많이 연구하였으나, 이는 유지 비용이 비싸다는 단점과 웹 환경이 갈수록 복잡해지고 이에 따라 적응력이 떨어질 수 있다. 반면, P2P 시스템을 이용한 웹 캐싱 방법은 피어 자신이 가지고 있는 하드웨어를 통하여 캐싱을 수행하므로 별도의 하드웨어 없이도 원하는 효율을 얻을 수 있다[8].

P2P 환경에서의 시맨틱 캐싱에 대한 연구를 살펴보면, [9]의 연구에서는 각 피어마다 스키마를 기반으로 정보가 구축되어 있는 환경에서 슈퍼피어를 중심으로 클러스터가 형성되어 있는 경우의 시맨틱 캐싱에 대한 연구를 진행하였다. 캐쉬를 구현하기 위하여 저장된 질의와 그 결과들을 뷰의 개념으로 처리하였으며, 질의 재작성시에 가장 큰 장애인 질의의 포함 문제를 해결하기 위하여 입력된 질의를 캐쉬 뷰와 내용들을 가지고 처리하기 위하여 뷰를 사용하여 질의를 처리하는 알고리즘중 하나인 MiniCon 알고리즘을 사용하였다. 그러나, 이 연구는 네트워크 환경이 슈퍼피어를 중심으로 클러스터를 구축한 상태에서 연구되었으므로, 슈퍼피어가 존재하지 않는 환경에서의 시맨틱캐싱의 연구가 필요하다. 현재 P2P 환경의 시맨틱웹에 대한 연구에 있어서 더욱 다양한 환경에서의 연구가 진행되어야 하나, 현재는 기존의 캐쉬 기법의 한계를 벗어나지 못하고 있다. 또한 캐쉬 대체 방법에 있어서도 FIFO, LRU 방법과 같은 일반적인 메카니즘을 따르고 있다.

본 연구는 우선 웹 환경에서의 캐쉬 기법으로 적합한 시맨틱 캐쉬 기법에 대한 기존의 연구들을 파악하였다. 위의 연구들에서 제시하지 못한 두가지 관점에서의 캐쉬 메카니즘을 제시한다. 한가지는 질의 필터링이며, 다른 하나는 캐쉬 대체방법이다. 질의 필터링은 일반적인 데이터 필터링의 관점이 아닌 온톨로지가 지니는 스키마 정보를 이용한 필터링을 의미하는 것이며, 캐쉬 대체 방법은 각 온톨로지가 지니는 스키마와 프로퍼티들의 유사도를 이용한 대체 방법을 제시한다.

III. 본 론

1. 클러스터환경의 정의

P2P 방식의 데이터 처리 방식이란 각 피어마다 각자 데이터베이스나 온톨로지와 같은 정보가 구축되어 있고 각 피어들은 서로 네트워크를 통하여 연결되어 있을 때, 사용자들의 질의를 응답하기 위하여 별도의 중앙서버와 같은 중계 시스템없이 피어간의 자율적인 데이터 통신을 통하여 정보를 획득하는 환경을 의미한다.

이러한 환경은 갈수록 복잡해지고 동적으로 변해가는 웹환경에 적응력을 높일 수 있고, 별도의 중앙시스템을 필요로 하지 않으므로 유지비용을 줄일 수 있다. 하지만, 피어들간의 과도한 메시지 전송으로 시스템의 효율이 오히려 떨어질 수 있기 때문에 질의가 발생하였을 때 모든 피어에 질의를 전송하는 것은 효율적이지 않다. 그러므로, 온톨로지가 연결되어 있는 P2P 네트워크에서의 효율적인 질의 전송을 위하여 클러스터의 개념이 필요하다. 각 클러스터에서는 인덱스를 사용하여 자기 클러스터에 소속되어 있는 피어들을 관리하고, 클러

스터에서 얻을 수 없는 정보는 클러스터간의 정보 교환을 통하여 해결하는 방법이 필요하다. 본 연구에서 캐싱메카니즘을 구현하는 환경은 다음과 같은 클러스터환경의 질의 처리를 가정한다.

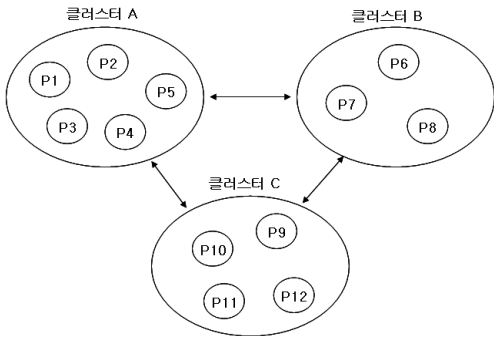

그림 1. 클러스터 구성 개념
Fig. 1. Cluster concept

그림 1은 클러스터의 개념을 보여준다. 클러스터에 존재하는 각각의 피어는 클러스터에 존재하는 다른 피어들에 대한 메타데이터 정보를 지니도록 설계한다.

새로운 피어가 클러스터에 등록될 때 해당 피어는 클러스터의 다른 피어들에게 자신의 메타데이터 정보를 제공한다. 이를 클러스터인덱스라 하며, 클러스터인덱스는 항상 최신의 정보를 지녀야 하며 이를 위하여 인덱스의 업데이트는 변동사항이 있을 때 마다 수행된다. 이처럼 각 피어에 존재하는 클러스터인덱스는 클러스터내의 다른 피어들의 정보를 지닌다. 이 정보는 피어들의 사용 스키마, 클래스, 프로퍼티 등이다. 각 클러스터에서는 질의에 존재하는 요소들에 대하여 질의에 응답이 가능한 피어를 결정하기 위하여 클러스터인덱스의 내용들과 일치하는 지를 판단한다. 일치하는 내용이 있는 피어가 결정되면 해당 피어는 해당 질의에 응답하도록 한다. 각 피어마다 존재하는 클러스터인덱스는 일반적인 질의처리 절차뿐만 아니라 캐싱 메카니즘에서 검색범위를 줄이기 위하여 참조되는 중요한 요소로 사용될 것이다.

2. 클러스터환경의 캐싱 메카니즘

본 연구의 시맨틱 캐싱은 튜플 캐싱이나 페이지 캐싱과 달리 질의와 질의 결과를 저장하여 다음의 질의 처리에 이용하는 개념으로 정의한다. 시맨틱 캐싱은 실제로 발생한 질의에 대한 결과를 캐싱하기 때문에 인터넷 환경에서의 캐싱 기법으로 가장 적합하다.

클러스터 환경에서 질의의 처리 절차는 질의가 입력되면 우선 자신의 캐쉬에 존재하는 데이터들을 접근하며 캐쉬에 데이터가 없는 경우에는 자신 피어에 존재하는 데이터를 접근하며, 자신의 피어에 존재하지 않는 데이터는 이를 지니는 특정 피어에게 데이터를 요청한다. 질의에서 요청한 데이터가 캐쉬에 존재하는 부분은 부분 결과(Partial Result)라고 정의하고, 나머지 수행해야 할 부분을 잔여 질의(Remainder Query)라고 정의한다. 성공적인 캐싱은 잔여 질의가 최소한으로 발생하도록 하는 것이다. 본 연구의 환경에서는 잔여 질의의 발생이 동일 클러스터에 발생한 경우와 다른 클러스터의 데이터를 필요로 하는 경우를 구분하여 처리하도록 한다.

잔여 질의의 발생을 줄이는 여러 가지 요소 중에 하나로 캐쉬의 적중률(Hit Ratio)를 높이기 위한 효율적인 캐쉬 데이터 대체 정책을 세우는 방법이 있다. 일반적으로 대체 선택은 시간적, 공간적, 의미적 지역성을 바탕으로 선택한다. 시간적인 지역성은 최근에 접근했던 데이터를 가까운 미래에 다시 접근할 가능성이 많다는 가정을 원칙으로 하며, 공간적 지역성은 어떤 데이터가 접근된 경우에 해당 데이터와 인접한 데이터들을 가까운 미래에 접근할 가능성이 많다는 가정을 원칙으로 구성된다. 의미적 지역성은 공간적 지역성과 유사한 개념이나, 정적으로 묶여있는 페이지에 존재하는 튜플들을 대상으로 하는 것이 아니라 질의 접근 패턴에 근거한 동적인 접근 방법이다.

잔여 질의의 발생을 최소화하기 위한 직접적인 방법으로 피어마다 존재하는 캐쉬 데이터를 통하여 최대한으로 잔여 질의를 줄이는 메카니즘이 필요하다. 효율적인 데이터 대체 정책으로 자주 접근할 데이터가 캐싱되어 있다하더라도 현재 입력된 질의와 캐쉬에 저장된 질의가 정확히 일치하는 경우 외에는 이들간의 포함 관계를 정확히 판단하여 최소한의 잔여 질의 발생이 되도록 하여야 한다. 이 절차를 필터링(Filtering)이라 하며, 본 연구에서는 이를 데이터 필터링(Filtering with data)과 스키마필터링(Filtering with schema)으로 분류하였다. 데이터 필터링은 데이터 자체가 지니는 값들을 가지고 판단하는 방법이며, 스키마 필터링은 데이터의 스키마를 가지고 판단하는 방법이다.

- 데이터 필터링 : 전통적인 데이터베이스 캐싱 기법에서 많이 사용하는 방법이다. 예를 들어, 캐싱되어 있는 질의가 50살이상의 교직원에 대한 데이터를 지니고 있다면, 질의 결과가 지니는 나이 데이터를 필터링하면 60세 이상의 교직원에 대한 데이터를 캐쉬에서 해결할 수 있다는 것이 유추된다.
- 스키마 필터링 : 본 연구에서 제안한 스키마 필터링은 피어에 존재하는 스키마 정보를 이용하여 잔여 질의를 결정하도록 하는 방법이다. 온톨로지가 피어마다 구축되어 있는 환경에서 시맨틱 캐싱은 온톨로지의 스키마 정보를 이용하면

더욱 효율적인 캐싱이 가능하다. 예를 들어, SUV자동차에 대한 질의가 발생하면 이 정보를 지니는 특정 피어를 통하여 질의 결과를 가져온다. 일반적인 시맨틱 캐싱에서는 발생한 질의와 그 결과만을 캐싱한다. 그러나, 다음에 현대자동차의 SUV자동차에 대한 질의가 발생하면 캐쉬에 결과 데이터가 존재함에도 불구하고 현대자동차 제품을 구분할 수 있는 정보가 없다면 다시 질의를 발생시켜야 한다. 이때, 질의 결과 인스턴스의 클래스정보, 하위클래스 정보가 있다면 캐쉬의 정보로도 질의를 처리 할 수 있다. 마찬가지로 프로퍼티의 정보를 지니면 캐싱을 효율적으로 처리할 수 있다.

잔여질의의 발생을 최소화하기 위하여 캐쉬에 존재하는 질의와 입력질의와의 관계를 판단할 때 다음과 같이 분류할 수 있다.

· 일치(Equivalence) - 입력 질의에서 요구하는 내용과 일치하는 질의와 질의 결과가 캐쉬에 존재하는 경우이다.

· 전부 포함(Fully containment) - 입력 질의에서 요구하는 내용을 모두 포함하는 질의와 질의 결과가 캐쉬에 존재하는 경우이다. 이 경우에 캐쉬에 존재하는 질의 결과로부터 새로운 질의가 요구하는 결과를 구분하여 추출하는 과정이 필요하다.

· 일부 포함(Partially containment) - 입력 질의에서 요구하는 내용을 일부 포함하는 질의와 질의 결과가 캐쉬에 존재하는 경우이다. 이 경우에 캐쉬에 존재하는 질의 결과로부터 새로운 질의가 요구하는 결과를 구분하여 추출하고 나머지 부분에 대한 잔여 질의를 발생시켜 전체 결과를 조합하는 과정이 필요하다. 혹은, 캐쉬에 존재하는 두 개 이상의 질의 결과를 조합(합집합, 교집합)하여 새로운 질의 결과를 생성할 수 있는 경우에는 잔여 질의의 생성없이 질의를 처리할 수 있다.

· 불일치(Disagreement) - 입력 질의에서 요구하는 내용을 포함하는 질의와 질의 결과가 캐쉬에 존재하지 않는 경우이다.

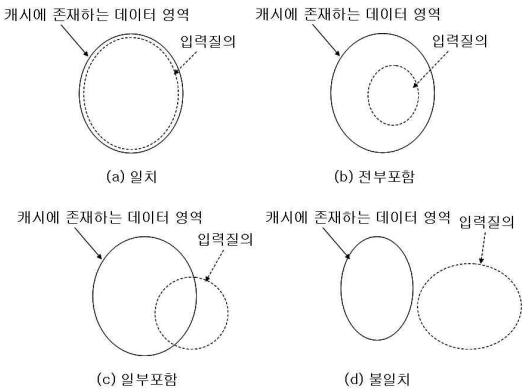

그림 2. 캐쉬와 입력질의 관계
Fig. 2. Relation of cache and input query

위의 네 가지 경우중에서 일부 포함의 경우에는 부분 결과를 산출하고, 잔여 질의를 발생시키기 위하여 필터링을 수행한다. 앞에서 언급한 바와 같이 필터링은 데이터 필터링과 스키마 필터링으로 나누어 수행된다. 데이터 필터링은 캐쉬에 존재하는 데이터를 가지고 수행하므로 바로 수행된다. 스키마 필터링의 경우는 다음과 같은 세가지 경우로 나누어 수행하며, 필터링을 수행하기 위해서는 캐쉬에 존재하는 질의의 결과에 해당 리소스의 클래스와 프로퍼티와 같은 스키마 정보를 첨가하여 지니도록 한다.

· 셀프 스키마 필터링 : 필터링하고자 하는 데이터가 자신의 피어에 존재하는 데이터이므로 자신이 지니는 스키마를 사용하여 필터링을 수행한다.

· 클러스터 스키마 필터링 : 필터링하고자 하는 데이터가 자신이 속한 클러스터에 존재하는 데이터이므로 자신이 지니는 클러스터 스키마를 사용하여 필터링을 수행한다.

· 외부 클러스터 스키마 필터링 : 필터링하고자 하는 데이터가 외부 클러스터에 존재하는 데이터이므로 직접 필터링을 수행하지 않고 질의를 해당 피어로 직접 전송한다.

위 세 가지 필터링을 수행하는 경우는 각각 필터링을 수행하지 않는 경우보다 데이터의 전송량과 수행 시간에 있어서 효율적이다. 이와 같이, 본 연구에서는 새로운 질의가 발생하였을 때 캐쉬에 존재하는 질의와 질의 결과를 비교하여 일치, 전부 포함, 부분 포함을 판단하고 잔여 질의의 발생을 결정하는 방법으로 데이터 자체가 지니는 값을 가지고 비교하는 방법과 함께 온톨로지의 특성을 이용하여 클러스터 별로 지니고 있는 각 피어의 스키마 정보를 이용하였다. 스키마 정보는 각 피어에 존재하는 온톨로지에 대한 클래스, 프로퍼티 등에 관한 상하위 관계등을 지니므로, 캐쉬의 내용과 새로운 질의를 비교하여 최소한의 잔여 질의가 발생하도록 한다. 또한, 클러스터 환경을 고려하였기 때문에 입력된 질의의 내용이 자신의 피어에서 해결되는 경우와 자신이 소속된 클러스터에서 가능한 경우, 다른 클러스터의 데이터가 필요한 경우로 나누어 해결하는 방식을 사용하였다.

3. 입력질의와 캐쉬 비교 분석

질의가 입력되면 시맨틱 캐쉬 처리기는 캐쉬의 내용과 입력질의를 비교한다. 새로운 질의와 캐쉬의 내용과의 관계에 대하여 부분 결과와 잔여 질의의 처리 내용은 다음과 같다.

· 일치(Equivalence) - 부분 결과가 모든 결과를 포함하고, 잔여 질의는 존재하지 않는다. 따라서, 부분 결과를 입력된 질의의 최종 결과로 산출한다. 예를 들어, 입력된 질의가 “현재 세계적으로 생산되고 판매된 SUV 자동차에 대한 자료

를 제출하라"라고 가정했을 때, 이에 대한 모든 내용이 캐쉬에 저장되어 있어서 별다른 질의 처리 없이 결과를 산출할 수 있는 경우이다.

· 전부 포함(Fully containment) – 부분 결과가 모든 결과를 포함하고, 잔여 질의는 존재하지 않는다. 그러나, 추가적으로 캐쉬에 존재하는 질의 결과 중에서 새로이 입력된 질의가 원하는 결과를 분류하기 위하여 스키마 정보를 이용하여 클래스와 프로퍼티의 상하 관계를 참조하여 정확한 결과를 산출한다. 단, 앞에서 언급된 세가지 경우의 필터링을 고려하여, 입력된 질의의 내용이 자신의 피어에서 해결되는 경우와 자신이 소속된 클러스터에서 가능한 경우, 다른 클러스터의 데이터가 필요한 경우로 나누어 처리한다. 예를 들어, 입력된 질의가 "2011년에 현대자동차에서 생산되고 판매된 SUV 자동차에 대한 자료를 제출하라"라는 질의가 있고, 캐쉬에는 SUV 자동차에 대한 모든 결과가 있을 때, 캐쉬에 존재하는 SUV 자동차에 대한 질의 결과중에서 "2011년"과 "현대"라는 조건을 필터링하며, 이때 데이터 필터링과 스키마 필터링이 발생한다.

· 일부 포함(Partially containment) – 부분 결과를 산출할 수 있으며, 잔여 질의가 발생한다. 전부 포함의 경우와 마찬가지로, 입력된 질의의 내용이 자신의 피어에서 해결되는 경우와 자신이 소속된 클러스터에서 가능한 경우, 다른 클러스터의 데이터가 필요한 경우로 나누어 필터링을 수행한다. 예를 들어, 입력된 질의가 "현재까지 현대자동차에서 생산되고 판매된 SUV 자동차에 대한 자료를 제출하라"라는 질의가 있을 때, 캐쉬에는 2005년 이후의 자료만이 존재한다면, 2005년 이전의 자료를 처리하는 부분이 잔여 질의로 생성되어 처리 되도록 한다.

· 불일치(Disagreement) – 입력된 질의에 대한 질의 결과가 캐쉬에 존재하지 않는 경우이므로 부분 결과가 존재하지 않으며 입력된 질의가 모두 잔여 질의로 생성된다.

4. 유사도를 이용한 캐쉬 대체 방법

본 연구에서 제안하는 캐쉬 대체 방법은 온톨로지의 특성인 구조적 관계에 기반하여 자신의 피어의 온톨로지와의 유사도를 측정하고 측정 결과에 따라 캐쉬의 대체를 결정하는 기법이다. 이 방법의 전제는 해당 사이트의 온톨로지와 유사한 유사도를 지닌 결과가 다음에도 질의될 가능성이 높다는 가정이다. 다음은 온톨로지와의 유사도를 측정하기 위한 요소들을 나열한 것이다.

표 1. 유사도 측정 요소
Table 1. Similarity measure items

항목	내용
스키마	질의의 결과가 해당 피어의 온톨로지와 동일한 스키마를 사용하는 경우
클래스	질의의 결과에 해당 피어의 온톨로지가 지니는 동일한 혹은 동치인 클래스가 존재하는 경우와 질의의 결과에 해당 피어의 서브클래스관계의 클래스가 존재하는 경우
프로퍼티	질의의 결과에 해당 피어의 온톨로지가 지니는 동일한 혹은 동치인 프로퍼티가 존재하는 경우와 질의의 결과에 해당 피어의 서브프로퍼티관계의 클래스가 존재하는 경우

<표1>에서 언급한 바와 같이 유사도 측정을 위하여 크게 스키마, 클래스, 프로퍼티의 3가지 요소와 세부적으로 5개의 요소를 가지고 유사도를 측정하도록 한다. 특징적인 것은 서브클래스관계와 서브프로퍼티 관계에 대한 정보도 유사도 측정을 위하여 고려한다. 본 연구에서는 식(1)에서와 같이 두개의 온톨로지 간의 유사도를 측정하기 위한 함수를 제시한다.

$$S(O,Q) = a_1 \cdot S_s(O,Q) + a_2 \cdot S_c(O,Q) + a_3 \cdot S_p(O,Q)$$
$$\text{단, } a_1, a_2, a_3 >= 0 \quad (1)$$

유사도 함수 S는 <표1>의 유사도 관계 측정에서 제시한 관계에 따라 두개의 온톨로지간의 유사도를 수치 값으로 표현하도록 하였다. $S(O,Q)$는 온톨로지 O와 질의결과 Q간의 유사도를 나타낸다. $Ss(O,Q)$는 두 온톨로지간의 스키마 유사도, $Sc(O,Q)$는 두 온톨로지간의 클래스 유사도, $Sp(O,Q)$는 두 온톨로지간의 프로퍼티 유사도이며, a_1, a_2, a_3 는 각각의 가중치이다. $Sc(O,Q)$와 $Sp(O,Q)$는 관계 측정 요소 내용을 참조하여 다음과 같은 식에 따라 산출된다.

$$S_c(O,Q) = w_1 \cdot S_e(O,Q) + w_2 \cdot S_u(O,Q)$$
$$S_p(O,Q) = x_1 \cdot S_e(O,Q) + x_2 \cdot S_u(O,Q)$$
$$\text{단, } w_1, w_2, x_1, x_2 \geq 0 \quad (2)$$

$S_e(O, Q)$는 동일한 클래스나 프로퍼티의 일치여부를 나타내며, $Su(O, Q)$는 서브클래스나 서브프로퍼티의 일치 여부를 정도를 나타낸다. 각 가중치는 <표 1>에 나열된 요소들의 중요도에 따라 결정된다. 질의의 결과는 해당 피어의 온톨로지와 비교되어 유사도가 산출되며, 캐쉬에 존재하는 질의들의 유사도 수치와 비교하여 대체 여부를 결정한다.

질의의 결과를 토대로 산출된 유사도가 캐쉬에 존재하는 유사도보다 높은 유사도를 지니는 경우에는 캐쉬에 존재하는

질의 결과 중에서 가장 낮은 유사도를 지니는 질의와 대체된다. 본 연구의 대체 단위는 페이지 단위가 아닌 논리적인 질의 결과 단위이기 때문에, 대체될 두 개의 질의 결과 데이터의 크기를 고려하여야 한다. 따라서, 질의 결과 크기에 따라 하나의 질의 결과를 캐쉬로 대체하기 위하여 캐쉬에 현재 존재하는 2개 이상의 질의 결과를 제거 할 수 있다. 단, 이러한 경우에도 반드시 유사도를 바탕으로 산출되어야 하며, 대체되는 2개 이상의 유사도보다 높은 유사도가 유지될 경우에만 대체가 가능하다. 따라서, 경우에 따라서는 캐쉬로의 대체가 불가능 할 수 있다.

5. 캐쉬메카니즘을 이용한 질의처리 과정

시맨틱 캐쉬를 이용한 시맨틱 웹 데이터의 P2P 질의 처리 과정은 다음과 같다.

단계 1 : 입력 질의와 캐쉬 비교 분석

한 피어에서 온톨로지를 기반으로 한 질의가 발생하면, 피어는 자신의 시맨틱 캐쉬에 존재하는 질의와 비교한다.

case 1 : 일치하는 질의가 캐쉬에 존재하는 경우, 캐쉬에 존재하는 해당 질의의 결과 데이터를 반환하고 종료한다.

case 2 : 입력된 질의 결과를 모두 포함하는 질의가 존재하는 경우, 캐쉬에 존재하는 질의 결과를 비교하여 입력된 질의의 결과를 산출하고 종료한다.

case 3 : 일부 포함의 경우에는 필터링을 통하여 캐쉬에서 부분 결과를 산출하고, 잔여 질의를 질의 처리기로 보낸다. 외부 클러스터 스키마 필터링의 경우에는 질의 처리기를 통하여 부분 결과를 산출한다.

case 4 : 불일치의 경우에는 그대로 질의 처리기로 입력 질의를 보낸다.

단계 2 : 질의처리기

질의 처리기는 캐쉬 비교 분석기로 입력된 질의를 분석하여 캐쉬에서 처리하지 못한 잔여 질의를 처리한다.

일단 자신의 피어에 존재하는 데이터에 대하여 잔여 질의를 처리한다. 다음으로, 클러스터내의 다른 피어의 정보를 지니는 피어 인덱스를 사용하여 클러스터내의 클래스와 프로퍼티의 내용을 검색한다. 검색된 결과를 바탕으로 자신의 피어에서 처리가 가능한 경우에는 자체 질의 모듈로 질의를 처리하고 종료한다. 다른 피어로의 질의 전송이 요구되는 경우에는 각 피어에 전송될 별도의 질의를 산출한다.

단계 3 : 외부 클러스터로 질의 전송

소속된 클러스터내에서 질의의 처리가 수행되지 못한 경우에는 외부 클러스터로 질의가 전송된다.

단계 4 : 부분 질의 전송 및 결과 취합

단계 2, 3을 통해 각 지역 피어로 잔여 질의가 전송되면 지역 피어는 전송 받은 질의를 실행하여 결과 데이터를 추출한다. 최종적인 부분 질의의 처리가 끝나면 결과 데이터는 다시 호출 피어로 전송되고, 호출 피어는 전송 받은 결과를 취합하여 결과를 반환한다.

단계 5 : 캐쉬 대체

단계 4을 통하여 나온 결과를 가지고, 소스피어의 온톨로지와 비교하여 유사도를 산출한다. 산출된 유사도를 토대로 캐쉬의 내용을 대체할 것인지를 결정하여 처리한다.

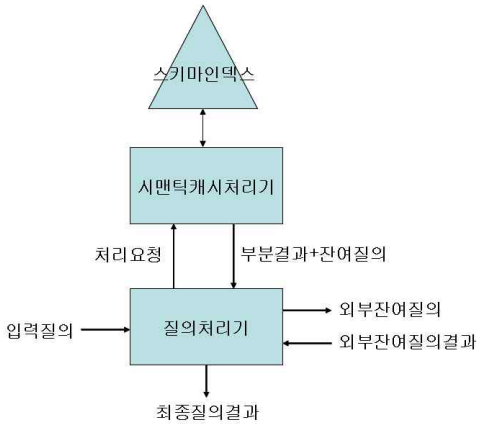

그림 3. 시맨틱 캐쉬를 이용한 질의처리절차
Fig. 3. Query processing using semantic cache

6. 유사도 측정

본 논문에서 제안한 캐쉬 대체를 위한 유사도 측정 함수를 실험하기 위해 OWL문서로 생성한 온톨로지와 다양한 응용 도메인에서 사용되는 12개의 OWL 문서를 질의 결과로 하여 실험을 진행하였다. 실험에 사용된 OWL 문서들은 가족, 책과 음악, 사람, 야구와 쇼핑에 대한 주제에 대한 결과이다. 결과문서별로 OWL 문서의 특성은 표 2와 같다. 연구의 환경이 유사한 피어들을 클러스터로 구분하여 P2P 방식으로 질의를 처리하므로, 질의가 발생한 피어에서 질의 결과를 모두 산출한 경우와, 클러스터내에서 결과를 산출한 경우, 클러스터 밖에서 결과를 산출한 경우, 그리고 이들이 혼합된 경우를 분리하여 실험을 진행하였다. 알고리즘은 C 언어로 구현하였으며, 1GB의 메인메모리, 윈도우 XP 운영체제를 플랫폼으로 한 3.4GHZ Pentium 4 PC 환경에서 실험하였다.

표 2. 실험 대상 OWL 질의결과문서의 특성
Table 2. Characters of query result documents

분류 \ 특성	클래스 개수	프로퍼티 개수	비고
질의결과1	15	13	동일 피어의 데이터결과
질의결과2	8	15	
질의결과3	12	21	
질의결과4	9	21	다른 클러스터의 데이터결과
질의결과5	25	12	
질의결과6	13	4	
질의결과7	7	5	동일 클러스터내의 데이터결과
질의결과8	8	6	
질의결과9	9	12	
질의결과10	14	17	동일 클러스터와 다른클러스터의 질의 결과 혼합
질의결과11	22	11	
질의결과12	12	7	

본 논문에서 제안한 유사도 함수를 이용해서 OWL 문서들 간의 유사도를 측정하기 위해서는 4절의 식(1),(2)에서 언급된 7개 가중치의 값을 결정하는 것이 중요하기 때문에 표 3과 같이 각 가중치를 변화시켜 실험을 진행하였다.

표 3. 실험별 가중치 값
Table 3. Weight values

분류 \ 가중치	a1	a2	a3	w1	w2	x1	x2
실험1	0.4	0.3	0.3	0.5	0.5	0.5	0.5
실험2	0.4	0.3	0.3	0.7	0.3	0.7	0.3
실험3	0.4	0.3	0.3	0.3	0.7	0.3	0.7
실험4	0.2	0.5	0.3	0.7	0.3	0.7	0.3
실험5	0.2	0.3	0.5	0.5	0.5	0.5	0.5
실험6	0.2	0.3	0.5	0.3	0.7	0.3	0.7
실험7	0.2	0.3	0.5	0.7	0.3	0.7	0.3

실험별 데이터를 정리하여, 각 실험별 유사도를 그림 4에 그래프로 표시하였다. 질의결과 1~12에 대하여 유사도를 계산하여 유사도 1~12를 각각 산출하였다.

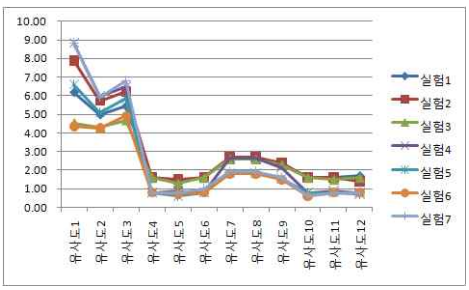

그림 4. 질의결과와의 유사도 측정결과
Fig. 4. Reaults of similarlity measure

그래프의 유사도1, 2, 3에서 보는 바와 같이 동일한 피어에서 결과를 산출한 경우 즉, 다시 말해서 유사한 스키마를 지니는 질의결과와의 유사도가 높은 것을 알 수 있다. 또한, 다음으로는 같은 클러스터에 존재하는 데이터를 가지고 질의결과가 산출된 경우인 유사도 7, 8, 9가 다음으로 높은 유사도를 보인다. 이는 캐쉬의 내용을 대체하는 기본적인 개념인 자신의 피어로부터 산출된 결과는 앞으로도 반복적으로 질의에 사용될 가능성이 많으며, 이를 우선적으로 캐쉬에 존재하도록 하자는 기본적인 아이디어에 부합된다.

계산된 유사도 값은 캐쉬에 존재하는 모든 질의 결과가 각각 지니며, 캐쉬의 크기와 질의 결과의 크기를 상호 비교하여 캐쉬를 대체하는데 적용한다.

유사도 측정과 관련된 부분 이외에 현재 제안된 연구의 성능을 추가적으로 평가하기 위하여 P2P 환경에서의 실험 테스트 시스템을 구축하고 있으며, 이러한 환경에서 캐쉬 대체 정책에 따른 효율성을 비교하여 다른 연구들과의 정량적인 성능 비교를 수행할 예정이다.

IV. 결 론

효율적인 질의 처리를 위하여 질의 응답 시간을 줄여주는 캐쉬 메카니즘은 운영체제에서 데이터베이스 분야에 이르기까지 계속적으로 연구되어 왔다. 인터넷 상의 질의 처리가 일반화 되어가고 있는 요즘에는 웹 환경에 더욱 적합한 개념의 캐싱의 필요성이 대두되었다. 시맨틱 캐싱의 개념은 P2P 네트워크 시스템과 같은 웹 환경에서, 질의가 발생하면, 소스 피어로의 데이터 전송량을 줄여서 효율적인 질의처리가 되도록 하는 중요한 부분이다. 시맨틱 캐싱은 캐쉬를 의미적인 영역들로 이루어진 공간으로 관리하는 개념이며, 논리적인 캐싱 단위가 질의와 질의 결과이므로 웹 환경에서 적합한 개념이다. 본 연구에서는 온톨로지와 같은 시맨틱 웹 정보가 클러스터 단위로 여러 피어에 분산되어 있는 경우에 캐쉬 메카니즘을 이용하여 효율적인 질의 처리가 이루어 지도록 하는 방법을 제시하였다. 특히, 캐쉬를 유지하고 처리하는 방법으로 스키마를 이용한 캐쉬 영역필터링과 스키마의 유사도를 이용한 캐쉬 대체 방법을 제시하였다. 발생된 질의를 가지고, 캐쉬의 내용과 비교하여 추가 질의를 발생시키고, 결과를 산출하는 경우의 처리 과정을 서술하였고, 캐쉬 대체 수식을 이용한 알고리즘을 실험하여 온톨로지와 결과 문서들간의 유사도를 나타내고 비교하여 효율적이고 능동적인 질의 대처 능력이 있음을 알 수 있었다. 본 연구는 현재까지 발생한 질의에 대한 유사도 측정을 통하여 대체와 필터링을 수행하도록 하였다. 미

래에 발생할 수 있는 질의를 예측하여 좀 더 효율적인 시스템 구축이 되도록 추후 연구 부분에서 다루도록 할 것이다

본 연구 과제의 수행을 통해 웹 환경에서의 캐쉬 적용에 대한 구체적인 방법이 구현될 수 있으며, 나아가 차세대 지능형 정보 검색 분야, 온톨로지 기반의 정보 교환 시스템 개발 분야의 경쟁력을 높일 수 있다. 또한, 국내 연구의 활성화로 관련 산업에 대한 기업들의 집중적인 연구 개발 투자를 유도하여 향후 지식 정보 기술 산업을 주도할 새로운 시맨틱 웹 환경의 응용 분야들에 대한 국내 기업들의 상품화를 지원할 수 있을 것으로 예상된다.

참고문헌

[1] Lee, Soonmi, "Design of Relational Storage Schema and Query Processing for Semantic Web Documents", Journal of the Korea Society of Computer and Information, v.14, n.1, pp. 35-45, 2009

[2] R. Alonso, D. Barbara and H. Garcia-Molina, "Data caching issues in an information retrieval system", ACM TODS 15, pp. 359 - 384, 1990

[3] Y. Arens and C.A. Knoblock, "Intelligent caching: selecting, representing, and reusing data in an information server", Proc. CIKM'4 Conference, Gaithersburg, MA, pp. 433 - 438, 1994

[4] Soo-Mok Jung, Kyung-Taeg Rho, "Advanced Disk Block Caching Algorithm for Disk I/O sub-system", Journal of the Korea Society of Computer and Information, v.12, n.6, pp. 139-146, 2007

[5] Dar, S., Franklin, M.J., J´onsson, B., Srivastava, D., Tan, M, "Semantic data caching and replacement.", Proc. 22th VLDB, Morgan Kaufmann Publishers Inc., pp. 330 - 341, 1996

[6] Godfrey, P., Gryz, J., "Answering queries by semantic caches.", Proc. 10th DEXA, Florence, Italy, 1999

[7] Luo Li, Birgitta K¨onig-Ries, N.P., Makki, K. " Strategies for semantic caching.", Proc. 12th DEXA. Volume 2113 of Lecture Notes in Computer Science., Springer, pp. 99 - 106, 2001

[8] Sitaram Iyer, Antony Rowstron, Peter Druschel, "Squirrel: a decentralized peer-to-peer web cache", Proceedings of the twenty-first annual symposium on Principles of distributed computing, pp. 213-222, 2002

[9] Ingo Brunkhorst, Hadhami Dhraief, "Semantic Caching in Schema-Based P2P-Networks.", Lecture Notes in Computer Science, Springer Berlin, Heidelberg, Volume 4125, pp. 179-186, 2007

저 자 소 개

김 병 곤

1990년 : 홍익대학교 공과대학 전자계산학과 이학사
1992년 : 홍익대학교 공과대학 전자계산학과 이학석사
2001년 : 홍익대학교 공과대학 전자계산학과 이학박사
1992년~1998년 : 국방과학연구소 연구원
2001년~현재 : 부천대학 e-비즈니스과 부교수
관심분야 : 다차원 인덱싱, 데이터웨어하우스, 시맨틱 웹 등
E-mail : bgkim@bc.ac.kr

오 성 균

1981년 : 홍익대학교 이공대학 전자계산학과 이학사
1984년 : 연세대학교 산업대학원 전자계산학과 공학석사
1999년 : 홍익대학교 공과대학 전자계산학과 이학박사
1987년~현재 : 서일대학 소프트웨어과 교수
관심분야 : 능동데이터베이스, XML 모델링, 소프트웨어공학
E-mail : skoh@seoil.ac.kr