

분산 XML 저장 시스템에서 질의 컴파일 시 시스템 카탈로그 관리 기법의 성능 평가

장건업[†], 홍의경^{**}

요 약

XML은 W3C(World Wide Web Consortium)에서 제안한, 인터넷 환경에서의 데이터 표현과 교환을 위한 표준 마크업 언어이다. 최근에는 XML 문서의 사용과 XML 문서의 양이 급속하게 증가하여 언제 어디서나 쉽게 필요한 XML 문서에 액세스할 수 있어야 한다. 이에 따라 분산 환경에서의 XML 문서의 처리가 요구되면서 분산 XML 저장 시스템(Distributed XML Repository System)이 개발되고 있다. 이를 위해 분산 XML 저장 시스템에서의 시스템 카탈로그 관리 기법에 대한 연구가 필요하다. 본 논문에서는 분산 XML 저장 시스템에서 CPU 비용, I/O 비용, 동시성 제어, 이단계 완료 프로토콜, 큐잉 지연 등을 모두 고려하고, 합리적으로 매개변수를 설정하고 시뮬레이션을 수행함으로써 중앙 집중식 카탈로그, 완전 중복식 카탈로그, 분할식 카탈로그 관리 기법의 성능을 평가하였다.

Performance Evaluation of System Catalog Management Schemes for Distributed XML Repository Systems at the Query Compilation Time

Gun Up Jang[†], Eui Kyeong Hong^{**}

ABSTRACT

XML is recommended by W3C(World Wide Web) and widely accepted as a standard markup language for expression and sharing data. Recently, increasing the use and the amount of XML documents, there is required that easy methods for accessing XML documents at anytime and anywhere, therefore, the distributed XML repository systems is developed. For this, distributed XML repository systems is necessary to research catalog management schemes. In this paper, we design distributed XML database model considering CPU costs, I/O costs, concurrency control, two-phase commits a protocol, queuing delay and so on. And we implement a simulator in order to execute performance evaluation centralized catalog method, fully replicated catalog method, and partitioned catalog method using reasonably setting up environments.

Key words: XML, Distributed Database(분산 데이터베이스), System Catalog(시스템 카탈로그), XQuery, Query Compilation(질의 컴파일)

1. 서 론

급격히 발전하는 인터넷 시스템에서 사용자의 요

구가 다양해짐에 따라, 이러한 요구들을 충족시킬 수 있는 방안으로 다양한 데이터를 표현하고 교환할 수 있는 XML(Extensible Markup Language)[1]이라는

* 교신저자(Corresponding Author) : 홍의경, 주소 : 서울특별시 동대문구 전농동 90(130-743), 전화 : 02)2210-2497, FAX : 02)2210-5275, E-mail : ekhong@venus.uos.ac.kr
접수일 : 2008년 10월 7일, 완료일 : 2008년 12월 5일

[†] 준회원, 서울시립대학교 컴퓨터과학부 박사과정

(E-mail : atlas@venus.uos.ac.kr)

^{**} 정회원, 서울시립대학교 컴퓨터과학부 교수

※ 본 연구는 2007년도 서울시립대학교 학술연구조성비의 지원을 받아 수행하였음

새로운 문서 표준이 제안되었다. XML은 W3C(World Wide Web Consortium)에서 제안한, 인터넷 환경에서의 데이터 표현과 교환을 위한 표준 마크업 언어이다. XML은 HTML과 마찬가지로 SGML의 부분 집합이지만, HTML 태그가 데이터 표현에 중점을 둔 것이라면, XML 태그는 데이터 자체를 기술한다. 따라서 XML의 자기 서술적인 특징(self-describing)을 바탕으로 문서의 내용을 여러 형태로 보여줄 수 있고, 내용을 기반으로 하여 필터링하거나 애플리케이션의 목적에 맞게 재구성하는 것이 가능하다.

최근에는 컴퓨팅 환경이 인터넷의 웹(World Wide Web)을 기반으로 하는 분산 컴퓨팅 환경으로 변화하였고, XML 문서의 사용이 급속하게 증가하였다. 이에 따라, 분산 환경에서 XML 문서를 저장하고, 저장된 XML 문서를 검색하는 분산 XML 저장 시스템에 대한 연구가 이루어지고 있다[2,3]. 하지만 분산 XML 저장 시스템에서 사이트 자치성(site autonomy), 질의 최적화(query optimization), 뷰 관리(view management), 권한 관리 기법(authorization mechanism), 데이터의 투명성 등에 큰 영향을 미치는 시스템 카탈로그의 관리 기법에 대한 연구는 현재까지 이루어지지 않았다.

시스템 카탈로그는 시스템 데이터베이스라고도 하며, 이는 데이터베이스 시스템이 관리하는 여러 객체인 테이블, 뷰, 인덱스, 사용자 액세스 모듈(수행 가능한 코드), 액세스 권한 등에 대한 정보를 포함하고 있다[4]. 시스템 카탈로그 자체도 사용자의 데이터 테이블과 같이 테이블들로 구성되어 있기 때문에, 시스템 카탈로그도 질의어를 사용하여 액세스할 수 있고, 사용자 테이블에 사용되는 것과 동일한 동시성 제어와 회복 기법에 의해 유지된다. 또한 질의 최적화, 사용자 데이터베이스 객체에 대한 권한, 데이터베이스 객체들 간의 의존성 등에 대한 정보를 저장한다.

분산 XML 저장 시스템의 시스템 카탈로그 관리 기법은 크게 중앙 집중식 시스템 카탈로그 관리 기법, 완전 중복식 시스템 카탈로그 관리 기법, 분할식 시스템 카탈로그 관리 기법으로 구분할 수 있다. 중앙 집중식 시스템 카탈로그 관리 기법은 분산된 모든 사이트의 카탈로그 정보를 하나의 중앙 사이트에 저장하는 방법이며, 완전 중복식 시스템 카탈로그 관리 기법은 분산된 각 사이트의 시스템 카탈로그 전체가

모든 사이트에 중복해서 저장되는 방법이다. 분할식 시스템 카탈로그 관리 기법은 원격 데이터 객체를 참조하는 질의가 들어오면 원격 데이터 객체를 저장하고 있는 사이트로부터 시스템 카탈로그 정보를 찾는 방법이다. 분할식 시스템 카탈로그 관리 기법은 원격 데이터 객체에 대한 시스템 카탈로그 정보를 지역 사이트에 캐시하고 관리하는 방법에 따라 여러 개의 변형으로 분류할 수 있다.

본 논문에서는 중앙 집중식 시스템 카탈로그 관리 기법, ROWA(Read One Write All) 방법을 사용한 완전 중복식 시스템 카탈로그 관리 기법, 원격 사이트의 카탈로그 정보를 지역 사이트에 캐시를 하지 않는 분할식 시스템 카탈로그 관리 기법, 원격 사이트의 카탈로그 정보를 지역 사이트에 완전히 캐시하는 분할식 시스템 카탈로그 관리 기법을 성능 평가의 대상으로 하였다. 또한, CPU 비용, I/O 비용, 동시성 제어, 이단계 완료 프로토콜, 큐잉 지연 등을 모두 고려한 분산 XML 저장 시스템 모델을 설계하였다. SimJava[5]를 이용하여 이 모델에 대한 시뮬레이터를 구현하였다. 합리적인 매개변수 설정을 통해 시뮬레이션을 수행함으로써 질의 컴파일 시 시스템 카탈로그 관리 기법들의 성능을 평가하였다.

성능 평가 결과, 모든 실험에서 중앙 집중식 카탈로그 관리 기법에서 읽기 질의와 쓰기 질의의 평균 응답시간이 가장 길었다. 중앙 집중식 카탈로그 관리 기법은 모든 사이트의 시스템 카탈로그가 하나의 사이트에만 저장되어 있으므로 그 사이트가 병목 지점이 되기 때문이다. ROWA 방법을 사용하는 완전 중복식 카탈로그 관리 기법은 읽기 질의의 비율이 높은 응용에서 컴파일하는데 걸리는 시간이 짧게 나타났다. 분할식 카탈로그 관리 기법은 각 실험마다 캐시의 유무에 따라 읽기 질의와 쓰기 질의의 평균 응답시간이 다르게 나타남을 알 수 있었다.

본 논문의 구성은 다음과 같다. 2절에서는 분산 XML 저장 시스템에서 질의를 수행하는 단계에 대해서 설명하고, 3절에서는 이 논문에서 실험한 시스템 카탈로그 관리 기법에 대하여 설명하고, 4절에서는 분산 XML 저장 시스템 모델을 소개하고, 시뮬레이션 환경과 시뮬레이션 매개변수들에 관해서 설명한다. 5절에서는 시뮬레이션 모델을 사용하여 실험한 조건과 그에 따른 실험 결과에 관해 설명하고, 마지막으로 6절에서 결론을 맺는다.

2. 질의 컴파일 단계

어떤 유형의 사용자 질의든 질의 컴파일 과정에서 데이터베이스를 액세스하기 전에 시스템 카탈로그를 먼저 액세스하게 된다. 일반적으로 사용자 질의는 데이터에 대해 읽기 연산을 수행하는지 쓰기 연산을 수행하는지에 따라 읽기 질의 또는 쓰기 질의로 구분하지만, 시스템 카탈로그에 대한 읽기 질의는 전형적인 읽기 질의뿐만 아니라, 쓰기 질의도 포함된다. 본 논문에서는 시스템 카탈로그에 대한 쓰기 액세스가 있는 경우에만 쓰기 질의로 간주한다. 따라서 데이터베이스 자체를 읽거나 쓰는 것에 상관없이 질의 처리 과정에서 시스템 카탈로그에 대하여 읽기 연산이 필요하면 *읽기 질의*라고 정의하고, 질의 처리 과정에서 시스템 카탈로그에 대하여 쓰기 연산이 수반되면 *쓰기 질의*라고 정의한다. 이러한 구분에 따라 표 1에서 SQL문을 시스템 카탈로그에 대한 읽기 질의와 쓰기 질의로 구분하였다.

2.1 객체 이름 지정

여러 개의 사이트로 이루어진 분산 XML 저장 시스템에서 질의에 사용된 객체를 시스템 내에서 식별 가능한 이름으로 바꾸어주는 방법은 시스템 설계 시 중요한 의미를 가진다. 시스템 카탈로그 구조의 성능

을 일관성 있게 비교하기 위해서 상용 데이터베이스 시스템인 Oracle사의 분산 데이터베이스에서 사용하는 객체 이름 지정 방식을 사용하였다[6]. 분산 XML 저장 시스템을 이루는 각 사이트는 처음 연결될 때 다음과 같은 고유한 식별자를 갖는다.

```
<global_domain_name> =
<database_name>.<site_name>
```

여기에서 *global_domain_name*은 분산 XML 저장 시스템에서 하나의 사이트를 고유하게 식별하기 위한 이름이며, *database_name*과 *site_name*의 쌍으로 표현된다. *database_name*은 지역 사이트의 데이터베이스 이름이다. *site_name*은 분산 XML 저장 시스템을 구성하고 있는 지역 사이트의 이름, IP 주소, 도메인 주소 등이 될 수 있다.

한 사이트 내에서 고유한 사용자 식별자와 *global_domain_name*과 결합되어 전체 분산 XML 저장 시스템 내에서 고유한 *userid*를 이룬다.

```
<userid> = <local_userid>@<global_domain_name>
```

분산 XML 저장 시스템 내의 객체는 다음과 같이 구성된다. 이것은 분산 XML 저장 시스템에서 고유하다.

```
<object_name> =
```

표 1. 시스템 카탈로그에 대한 읽기 질의 및 쓰기 질의

구분	명령어	기능
읽기	SELECT	테이블로부터 데이터를 검색
	INSERT	테이블에 튜플을 삽입
	DELETE	테이블로부터 튜플 삭제
	UPDATE	테이블의 컬럼 값 갱신
쓰기	CREATE TABLE	테이블 생성
	DROP TABLE	테이블 삭제
	ALTER TABLE	테이블에 컬럼 추가/삭제 또는 제약조건 변경
	CREATE INDEX	인덱스 생성
	DROP INDEX	인덱스 삭제
	CREATE VIEW	뷰 생성
	DROP VIEW	뷰 삭제
	GRANT	사용자에게 권한 부여
	REVOKE	사용자로부터 권한 취소
	CREATE SYNONYM	테이블이나 뷰 등에 대한 별명 정의
	DROP SYNONYM	테이블이나 뷰 등에 대한 별명 삭제
	UPDATE STATISTICS	데이터베이스가 관리하는 통계 정보 갱신

`<local_userid>.<local_object>@<global_domain_name>`

`object_name`은 지역 사이트의 사용자 아이디와 지역 사이트의 객체 이름, `global_domain_name`이 결합하여 구성된다.

2.2 분산 XML 저장 시스템의 시스템 카탈로그의 이름과 기능

시스템 카탈로그는 DBMS에 관련된 정보를 저장하게 되므로 DBMS에 따라서 시스템 카탈로그의 구조와 개수가 다르다. 표 2에 있는 시스템 카탈로그가 [7]의 분산 XML 저장 시스템에 필요한 최소한의 시스템 카탈로그라고 가정한다.

2.3 분산 XML 저장 시스템의 질의 컴파일 단계

일반적으로 읽기 질의는 응용 프로그램에 포함되어 반복적으로 수행된다. 응용 프로그램 내의 읽기 질의는 컴파일되어 질의를 포함하고 있는 응용 프로그램에 대한 액세스 모듈의 한 부분으로 저장된다.

한 질의에서 참조된 사용자 테이블을 저장하고 있는 각 사이트는 액세스 모듈 내에 그 사이트가 질의 처리 단계에서 수행해야 할 부분을 나타내는 액세스 구조를 저장한다. 한 개 이상의 내포 질의를 포함한 응용 프로그램은 질의 컴파일 단계와 실제 수행 단계의 순서에 따라 처리된다. 이 논문에서는 질의의 실제 수행 단계는 고려하지 않았다.

분산 XML 저장 시스템에서 분산 저장된 XML 문서에 질의할 때에는 SQL 질의와 XQuery 질의를 사용한다.

본 논문에서는 성능을 비교한 모든 시스템 카탈로그 관리 기법에 대해 그림 1과 같이 사용자의 XQuery 질의를 분산 SQL 질의로 변환하고, 변환된 분산 SQL 질의를 컴파일하여 액세스 모듈을 생성한다고 가정한다[7-12]. XQuery 파싱, XQuery Context Processing, XQuery 정규화, XQuery-to-SQL 변환의 4단계를 거쳐서 XQuery 질의가 분산 SQL 질의로 변환된다. 변환된 분산 SQL 질의를 컴파일하여 액세스 모듈을 생성하기 위해서는 SQL 파싱, 이름 도출, 카탈로그 조회, 권한 검사, 최적화, 액

표 2. 분산 XML 데이터베이스의 시스템 카탈로그

시스템 카탈로그 이름	기능
SYSCATALOG	분산 XML 저장 시스템의 각 테이블이나 뷰에 대한 정보를 저장한다. 테이블이나 뷰 등을 생성한 사용자 정보와 테이블이나 뷰의 데이터에 대한 통계 정보를 저장한다. 이 통계 정보는 질의 최적화 과정에 사용된다.
SYSCOLUMNS	분산 XML 저장 시스템의 각 테이블이나 뷰의 컬럼에 대한 정보를 저장한다.
SYSACCESSES	응용 프로그램에 의해 생성된 액세스 모듈 정보를 저장한다.
SYSINDEXES	분산 XML 저장 시스템에 존재하는 인덱스에 대한 정보를 저장한다.
SYSSYNONYMS	분산 XML 저장 시스템에 존재하는 별명들을 기술하고, 별명을 정의한 사용자에 대한 정보를 저장한다.
SYSTABAUTHS	테이블이나 뷰 등에 액세스할 수 있는 권한을 기록한다.
SYSUSAGES	분산 XML 저장 시스템의 객체에 대한 액세스 모듈의 중속성을 기록한다.
SYSXMLDOCUMENTS	분산 XML 저장 시스템에 저장된 XML 문서에 대한 정보 저장. XML 문서의 사용자 정보와 XML 스키마나 DTD 문서에 대한 정보와 수정 여부에 대한 정보를 저장한다.
SYSXMLSCHEMAS	분산 XML 저장 시스템에 저장된 XML 문서의 구조 정보를 가지고 있는 XML 스키마나 DTD 문서를 저장한다. 분산 XML 문서의 구조에 대한 검증과 재구성 과정에 사용된다.
SYSXMLELEMENTS	분산 XML 저장 시스템에 저장된 XML 문서들의 엘리먼트들을 저장하고 있는 테이블에 대한 정보를 저장한다.
SYSXMLATTRIBUTES	분산 XML 저장 시스템에 저장된 XML 문서들의 컬럼들을 저장하고 있는 테이블에 대한 정보를 저장한다.
SYSXMLCONTENTS	분산 XML 저장 시스템에 저장된 XML 문서들의 콘텐츠 정보를 저장하고 있는 테이블에 대한 정보를 저장한다.

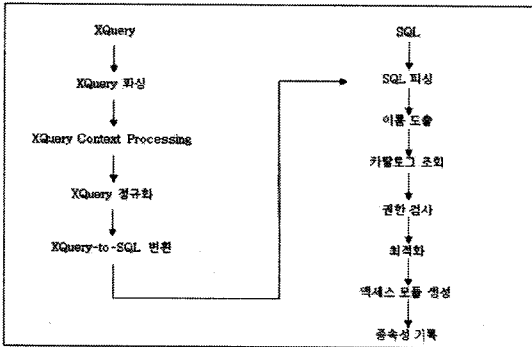


그림 1. 분산 XML 저장 시스템에서의 질의 컴파일 단계

세스 모듈 생성, 종속성 기록 등 6단계를 거치게 된다. XQuery Context Processing 단계와 XQuery-to-SQL 변환 단계는 [7,8]에 따라서 이루어지며, XQuery 질의를 정규화하는 것은 [9]의 규칙을 사용한다. 분산 SQL 질의로부터 액세스 모듈을 생성하는 6단계는 [10,11,12]를 따른다.

2.3.1 XQuery 질의를 분산 SQL 질의로 변환

XQuery 파싱 단계에서는 사용자의 XQuery 질의가 정확한 구문을 갖는지 검사한다. XQuery 질의에 대한 파싱이 이루어지면서 SQL 질의로의 변환에 필요한 모든 정보를 추출하여 파스 트리를 만든다. 이 단계는 주기억장치에서 이루어지므로 시스템 카탈로그에 대한 액세스가 일어나지 않는다[7].

XQuery Context Processing 단계는 XQuery 파싱 단계에서 발생한 토큰을 FLOWR절별로 분류하여 처리한다[7,8]. 이 단계에서는 FLOWR절을 처리하기 위하여 표 2의 SYSXMLDOCUMENTS, SYSXMLSCHEMAS 시스템 카탈로그를 한 번씩 액세스한다.

XQuery 정규화 단계에서는 XQuery 질의를 SQL로 쉽게 변환하기 위해서 XQuery 질의를 단순화하고, XQuery 질의의 중복을 제거한다[9]. 이 단계에서는 [9]의 XQuery 질의 정규화 규칙을 사용한다. 이 단계는 주기억장치에서 이루어지므로 시스템 카탈로그에 대한 액세스가 발생하지 않는다.

XQuery-to-SQL 단계는 XQuery 파싱 단계와 XQuery Context Processing 단계에서 분류된 토큰과 심볼 테이블, 처리된 FLOWR 표현식을 분산 SQL로 변환한다. 이 단계에서는 XQuery 질의를 분산 SQL로 변환하기 위해서 표2의 SYSCATALOG,

SYSXMLDOCUMENTS, SYSXMLSCHEMAS, SYSXMLELEMENTS, SYSXMLATTRIBUTES 시스템 카탈로그를 한 번씩 액세스하여 필요한 카탈로그 정보를 참조한다. 그리고 이 단계에서 원격 사이트의 카탈로그에 대한 액세스가 필요할 수 있다[7].

2.3.2 분산 SQL 처리 단계

SQL 파싱 단계는 변환된 분산 SQL 질의가 정확한 구문을 갖는가를 조사한다. 이 단계는 주기억장치에서 이루어지므로 시스템 카탈로그를 액세스하지 않는다[10].

이름 도출 단계에서는 사용자의 SYNONYMS 카탈로그나 목시적 규칙에 의해 변환된다. 이름 도출시 global_domain_name이 지정되지 않은 경우에 각 테이블이나 뷰마다 표 2의 SYSSYNONYMS 시스템 카탈로그에 대해 한 번의 액세스가 필요하다[11].

카탈로그 조회 단계에서는 질의에서 참조된 각 테이블에 대해 카탈로그 엔트리가 검색된다. 시스템 카탈로그 관리 기법에 따라 이 단계에서 원격 사이트 액세스가 필요할 수 있다[10]. 테이블이나 뷰마다 표 2의 SYSCATALOG 시스템 카탈로그에 대해 한 번의 액세스, 각 컬럼에 대해 SYSCOLUMNS 시스템 카탈로그에 대한 한 번의 액세스, 테이블에 존재하는 각 인덱스에 대해 SYSINDEXES 시스템 카탈로그에 대한 한 번의 액세스를 하게 된다.

읽기 질의의 컴파일에 관련된 액세스 권한에는 테이블에 대한 읽기 권한(SELECT할 수 있는 권한)과 쓰기 권한(INSERT, DELETE, UPDATE할 수 있는 권한)이 있다. 권한 검사 단계에서는 질의를 입력하는 사용자가 질의에서 참조된 객체들에 대하여 필요한 액세스 권한을 가지고 있는가를 검사한다. 이 단계에서는 각 테이블마다 한 번씩 SYSAUTHS 시스템 카탈로그의 조회가 필요하다.

최적화 단계에서는 분산 XML 저장 시스템에 구현된 최적화 알고리즘에 의하여 모든 사이트에서의 총 비용을 최소로 하는 액세스 계획을 만든다[11,12].

액세스 모듈 생성 단계에서는 최적화된 액세스 계획이 주기억장치에서 액세스 모듈로 컴파일되어 표 2의 SYSACCESSES 시스템 카탈로그에 저장된다. 액세스 모듈을 SYSACCESSES 시스템 카탈로그에 저장할 때 보통 한 번의 액세스가 필요하다. 만일 응용 프로그램이 많은 질의를 포함하여 액세스 모듈이

4K바이트 디스크 페이지보다 크면 그 이상의 액세스가 필요하다.

액세스 모듈 생성 단계에서 생성된 액세스 모듈은 액세스 모듈이 생성될 시점의 객체, 액세스 경로, 권한의 계속적인 존재 여부에 의존한다[10]. 이러한 종속성이 종속성 기록 단계에서 시스템 카탈로그에 기록된다. 종속성은 SYSACCESS, SYSUSAGE 시스템 카탈로그에 기록된다.

사이트 자치성을 보장하는 것을 중요한 목적으로 하는 시스템 카탈로그 관리 기법에서는 조정자(coordinator)와 참여자(participant) 사이에 위의 컴파일 단계가 부분적으로 중복되어 수행된다. 조정자는 항상 처음 질의가 입력된 사이트이고, 참여자 사이트는 질의에서 참조된 테이블을 하나 이상 저장하고 있는 사이트를 말한다[13].

쓰기 질의는 응용 프로그램 내에 내포시켜 사용해도 무방하지만 보통 대화식으로 사용되며, 컴파일되지 않고 인터프리트된다. XQuery[14]에서는 아직까지 수정 기능을 지원하지 않으며, 본 논문에서는 XML 문서에 대한 수정을 위하여 [15]의 방법을 따른다고 가정한다. [15]의 방법을 따르면 위의 질의 처리 단계 중 최적화, 액세스 모듈 생성, 종속성 기록의 세 단계는 불필요하다. 그러므로 쓰기 질의는 XQuery 파싱, XQuery Context Processing, XQuery 정규화, XQuery-to-SQL 변환, SQL 파싱, 이름 도출, 카탈로그 조회, 관련된 카탈로그의 실제 갱신 등의 단계로 이루어진다[10].

3. 분산 XML 저장 시스템에서 카탈로그 관리 기법

3.1 중앙 집중식 시스템 카탈로그 관리 기법(Centralized System Catalog Management)

중앙 집중식 시스템 카탈로그 관리 기법은 분산 XML 저장 시스템을 구성하고 있는 모든 사이트의 시스템 카탈로그 전체를 하나의 중앙 사이트에 저장하고 관리한다. 중앙 집중식 시스템 카탈로그 관리 기법은 높은 신뢰도와 가용성, 처리 부하의 분산 등의 면에서 분산 XML 저장 시스템의 이점을 활용하지 못한다. 이 관리 기법은 지역 사이트에 있는 데이터를 액세스할 때조차도 중앙 사이트에 시스템 카탈로그 정보를 요청해야 한다. 따라서 시스템 카탈로그

를 저장하고 있는 중앙 사이트에 고장이 발생하면 전체 분산 XML 저장 시스템을 사용할 수 없게 되고, 중앙 사이트에 과도한 오버헤드가 발생하여 병목 지점이 될 가능성이 높다.

3.2 완전 중복식 시스템 카탈로그 관리 기법(Fully Replicated System Catalog Management)

완전 중복식 시스템 카탈로그 관리 기법은 분산 XML 저장 시스템을 구성하는 모든 사이트에 시스템 카탈로그 전체를 중복해서 저장하는 것이다. 읽기 질의의 빈도가 높을 경우 시스템 카탈로그 정보를 참조하는 성능이 좋다. 그러나 중복 저장된 모든 사이트의 시스템 카탈로그를 일관성 있게 갱신해야 하기 때문에 쓰기 질의가 비효율적이고, 분산 XML 저장 시스템에 새로운 사이트를 추가하기 어려워 확장성이 떨어진다. 또한 한 사이트에서 결코 참조하지 않을 사이트 또는 테이블에 대한 시스템 카탈로그까지 유지하고 있어야 하기 때문에 카탈로그 저장 공간의 낭비가 초래된다.

완전 중복식 시스템 카탈로그 관리 기법은 시스템 카탈로그의 일관된 상태를 유지하는 것이 중요하다. 따라서 본 논문에서는 카탈로그의 일관성을 유지하는 방법으로 ROWA(Read One Write All) 알고리즘을 사용하였다.

3.3 분할식 시스템 카탈로그 관리 기법

3.3.1 캐시를 사용하지 않는 분할식 시스템 카탈로그 관리 기법(Partitioned System Catalog Management with No Cache)

분산 XML 저장 시스템을 구성하는 각 사이트는 지역 사이트에 저장된 데이터 객체에 대한 시스템 카탈로그 정보만을 관리한다. 원격 사이트에 저장된 데이터 객체들에 대한 시스템 카탈로그에 대한 캐시를 로컬 사이트에 유지하지 않으므로 원격 사이트의 객체를 액세스하는 질의가 입력될 때마다 매번 원격 사이트의 시스템 카탈로그 정보를 참조하며, 그 정보는 질의를 수행한 후 삭제된다.

3.3.2 캐시를 사용한 분할식 시스템 카탈로그 관리 기법(Partitioned System Catalog Management with Full Cache)

분산 XML 저장 시스템을 구성하는 각 사이트는

그 사이트에 저장된 데이터 객체의 시스템 카탈로그를 관리한다. 원격 데이터 객체를 처음 액세스할 때 그 데이터 객체를 저장하고 있는 원격 사이트로부터 시스템 카탈로그를 읽어 와서 로컬 사이트에 캐시로 저장한다. 다음에 동일한 원격 데이터 객체를 참조하는 질의가 입력되면, 원격 사이트의 시스템 카탈로그를 참조하지 않고 지역 사이트에 캐시된 내용을 참조하여 질의를 컴파일한다. 완전 중복식과는 달리, 원격 사이트에서 시스템 카탈로그가 갱신되더라도 로컬 사이트에 캐시해둔 시스템 카탈로그가 함께 갱신되지는 않는다.

4. 시뮬레이션

분산 XML 저장 시스템의 시스템 카탈로그 관리 기법들의 성능을 평가하기 위하여 우선 분산 XML 저장 시스템을 구성하고 있는 사이트 모델을 정의하였다. 이 모델 상에서 합리적인 가정을 한 후, 각 카탈로그 관리 기법의 성능을 시뮬레이션을 통해 평가하였다. 본 절에서는 시뮬레이션 모델, 시뮬레이션 수행 시 가정, 매개변수에 대해서 설명한다.

4.1 시뮬레이션 모델

분산 XML 저장 시스템은 LAN에 연결되어 있는 여러 개의 사이트로 구성되며, 각 사이트는 그림 2와 같은 구조를 갖는다. Java 6.0과 SimJava[5] 패키지를 사용하여 각 시스템 카탈로그 관리 기법마다 시뮬레이터를 구현하였다. SimJava는 프로세스 중심의

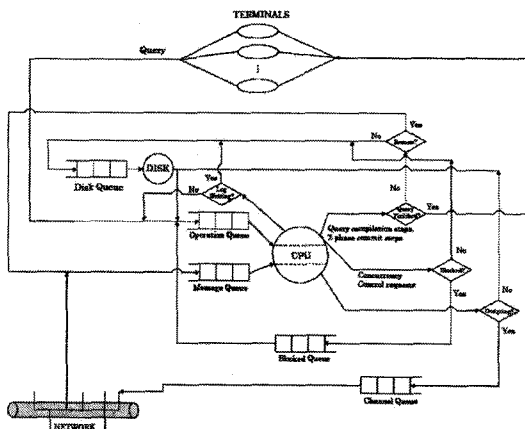


그림 2. 분산 XML 저장 시스템의 사이트 모델

패키지로서 모든 프로세스를 동시에 실행할 수 있도록 한다. 프로세스를 특정 시간 동안 정지시킬 수 있고, 특정 이벤트가 발생할 때까지 대기하도록 할 수도 있다.

한 사이트는 여러 명의 사용자 또는 터미널, CPU와 디스크, 5개의 큐로 구성된다. CPU는 일반 연산 큐(operation queue)나 메시지 큐(message queue)에 있는 요청들을 처리한다. 일반 연산 큐에서 대기하고 있는 요청들은 질의 컴파일, 동시성 제어, 2단계 완료 처리, 시스템 카탈로그의 읽기/쓰기 등이다. 메시지 큐에서 대기하는 요청들은 일반 연산 큐에서 대기하는 요청보다 우선순위가 높다. 따라서 CPU는 메시지 큐에 대기하는 요청이 없는 경우에 일반연산 큐에서 대기하는 요청을 처리한다.

지연 큐(blocked queue)에서는 동시성 제어로 인해 지연되는 요청들이 대기하며, 이러한 요청들은 지연 조건이 해결되면 일반 연산 큐로 옮겨진다. 채널 큐(channel queue)에는 네트워크를 통해 다른 원격 사이트로 보내질 요청이 대기한다. 디스크 큐(disk queue)에는 레코드를 디스크에서 읽거나 디스크로 쓰려는 요청들과 로그를 디스크에 쓰려는 요청들이 대기한다.

완전 중복식 시스템 카탈로그 관리 기법과 분할식 시스템 카탈로그 관리 기법을 사용한 분산 XML 저장 시스템의 사이트 모델은 같지만 중앙 집중식 시스템 카탈로그 관리 기법은 하나의 중앙 사이트에만 시스템 카탈로그가 저장되므로, 시스템 카탈로그에 대한 동시성 제어 요청을 처리하는 부분(지연 큐 등)이 중앙 사이트에만 있고, 다른 사이트에는 필요 없다.

4.2 시뮬레이션 수행 시 가정과 매개변수

본 논문에서는 시뮬레이션을 위해 다음과 같은 가정을 바탕으로 분산 XML 저장 시스템의 시스템 카탈로그 관리 기법들의 시뮬레이션을 수행하였다.

- 응용 프로그램에 내포된 질의는 시스템 카탈로그에 대한 읽기 질의, 쓰기 질의를 포함한다.
- 질의는 XQuery와 SQL 형식으로 이루어질 수 있는데, XQuery 질의는 그림 2에서와 같이 10단계의 수행 단계를 거치며, SQL 질의는 6단계의 수행 단계를 거친다. 또한 사이트에 입력되는 XQuery 질의와 SQL 질의의 비율은 각각 70%, 30%라고 가정한다.
- 분산 XML 저장 시스템의 동시성 제어를 위해

2단계 로킹 알고리즘(Two-phase locking algorithm: 2PL)을 사용한다. 또한 각 트랜잭션의 원자적인 완료를 위해서 2단계 완료 프로토콜(Two-phase commit protocol: 2PC)을 사용한다.

- 네트워크는 완전히 연결되어 있으며, 분산 XML 저장 시스템을 구성하는 사이트들의 고장이나 연결 정지는 발생하지 않는다.

시뮬레이션의 입력 값으로 사용되는 매개변수는 시스템을 위한 매개변수, 트랜잭션의 생성을 위한 매개변수, 자원 관리를 위한 매개변수로 구성된다. 표 3은 이 매개변수를 나타내며, 매개변수의 값은 [7,11,16]을 근거로 하였다. XQuery 질의를 컴파일할 때는 [16]과 같이 XQuery 질의를 네 가지 유형으로 나누고 동일한 비율로 질의가 입력되도록 하였다.

5. 시스템 카탈로그 기법의 성능 평가

시스템 카탈로그 관리 기법들의 성능을 평가하기

위해 분산 XML 저장 시스템을 구성하는 사이트 수, 하나의 사이트 내의 터미널 수, 하나의 사이트에서의 읽기 질의와 지역 질의의 비율을 변화시켜 가면서 실험하였다. 중앙 집중식 시스템 카탈로그 관리 기법은 CC로, ROWA 방법을 사용한 완전 중복식 시스템 카탈로그 관리 기법은 ROWA로, 캐시를 사용하지 않는 분할식 시스템 카탈로그 관리 기법은 PC로, 캐시를 사용한 분할식 시스템 카탈로그 관리 기법은 PCWC로 표현한다.

5.1 사이트 수의 변화에 따른 성능 평가

이 실험에서는 분산 XML 저장 시스템을 구성하는 사이트의 수를 3~15로 변화시켜 가면서 시스템 카탈로그에 대한 읽기/쓰기 질의를 컴파일하는데 걸리는 평균 시간을 측정하였다. 이 실험에서 분산 XML 저장 시스템을 구성하는 한 사이트 내의 터미널의 수는 25개이며, 시스템 카탈로그에 대한 읽기 질의 비율과 지역 질의 비율은 모두 80%로 가정하였다. 이 실험의 결과는 그림 3과 그림 4와 같다. 표

표 3. 시뮬레이션을 위한 매개변수

NumberOfSites	3~15	분산 XML 저장 시스템을 구성하는 사이트의 수
NumberOfTerminals	5~50	하나의 사이트에 연결된 터미널의 수
NumberOfXMLDocuments	1000	하나의 사이트에 저장된 XML 문서의 수
NumberOfTables	5000	하나의 사이트에 저장된 테이블의 수
TransactionCreationTime	3000ms	터미널에서 트랜잭션을 생성하는 평균 시간
ReadXactRatio	0~100%	시스템 카탈로그에 대한 읽기 질의의 비율
LocalXactRatio	0~100%	지역 사이트의 객체를 액세스하는 질의의 비율
MessageProcessingTime	2ms	메시지 처리 시간
LockAcquireTime	1ms	로크 획득 시간
LockReleaseTime	1ms	로크 반환 시간
DiskSeekTime	5ms	디스크 탐색 시간
LockingTime	2ms	동시성 제어, 데드록 등의 처리에 필요한 시간
XQueryParsingTime	30~100ms	XQuery 질의를 파싱하는데 걸리는 시간
XQueryContextProcTime	50~250ms	XQuery 질의의 FLOWR절을 처리하는데 걸리는 시간
XQueryNormalizationTime	250ms~500ms	XQuery 질의를 정규화하는데 걸리는 시간
XQueryToSQLTime	500ms~2000ms	XQuery 질의를 SQL 질의로 변환하는 시간
SQLParsingTime	30ms	SQL 질의를 파싱하는데 걸리는 시간
CatalogLookupTime	5ms	시스템 카탈로그를 조회하는데 걸리는 시간
NamingLookupTime	2ms	분산 질의 수행 시 이름을 도출하는데 걸리는 시간
AuthenticationTime	2ms	사용자 권한을 검사하는데 걸리는 시간
OptimizationTime	10ms	질의를 최적화하는데 걸리는 시간
CodeGenerationTime	2ms	코드 생성 시간

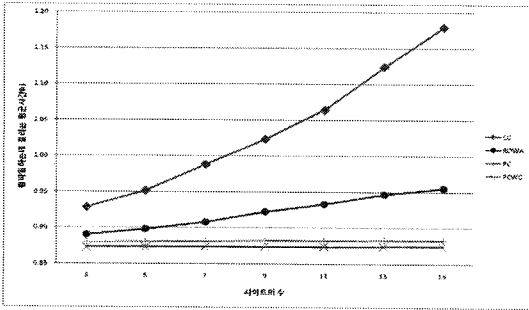


그림 3. 사이트 수의 변화에 따른 읽기 질의 컴파일에 걸리는 평균 시간

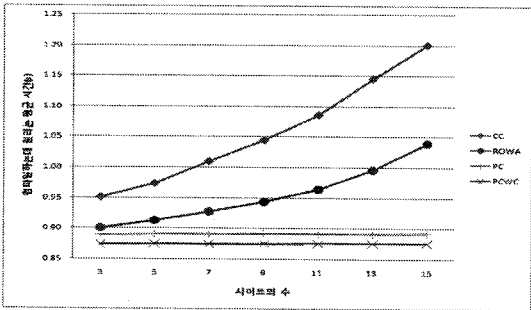


그림 4. 사이트 수의 변화에 따른 쓰기 질의 컴파일에 걸리는 평균 시간

4와 표 5는 사이트의 수가 각각 3, 9, 15인 경우의 자원 사용 비율과 큐 대기 시간의 비율을 나타낸다.

그림 3과 그림 4와 같이 읽기 질의와 쓰기 질의를 컴파일하는데 걸리는 시간이 CC에서 가장 길고, PCWC의 평균 시간이 가장 짧다. CC의 경우에는 중앙 사이트에서 시스템 카탈로그를 참조하여야 하기 때문에 큐잉 지연이 발생하고 카탈로그를 저장하고 있는 중앙 사이트의 오버헤드가 크기 때문에 사이트의 수가 증가할수록 컴파일하는데 걸리는 평균 시간이 길어진다.

ROWA의 경우에는 지역 사이트에 시스템 카탈로그의 사본이 저장되어 있기 때문에 읽기 질의의 경우에는 지역 사이트의 카탈로그를 읽고 질의를 컴파일하면 되지만, 쓰기 질의의 경우에는 모든 사이트의 시스템 카탈로그를 갱신하기 위해 배타 로크를 얻기 위해 대기하는 큐잉 지연이 발생하고, 2PC를 위한 메시지 처리와 데이터 전송을 위한 오버헤드가 발생하기 때문에 읽기 질의의 응답 시간보다 쓰기 질의의 응답 시간이 길다.

PC에서 각 사이트는 원격 사이트의 시스템 카탈

표 4. 사이트의 수에 따른 자원 사용 비율 (단위:%)

사이트의 수가 3인 경우 자원의 사용 비율				
	CC	ROWA	PC	PCWC
CPU 시간	1.02	3.68	20.33	26.22
디스크 시간	0.97	9.94	39.93	34.87
통신 시간	0.04	0.12	0.41	0.38
대기 시간	97.97	86.26	39.33	38.53

사이트의 수가 9인 경우 자원의 사용 비율				
	CC	ROWA	PC	PCWC
CPU 시간	0.98	3.50	18.88	24.08
디스크 시간	0.94	10.07	40.90	35.72
통신 시간	0.06	0.07	0.45	0.39
대기 시간	98.02	86.32	39.77	39.81

사이트의 수가 15인 경우 자원의 사용 비율				
	CC	ROWA	PC	PCWC
CPU 시간	0.93	3.46	17.68	23.01
디스크 시간	0.93	10.11	41.97	36.69
통신 시간	0.10	0.08	0.47	0.41
대기 시간	98.04	86.26	39.88	39.89

표 5 사이트 수에 따른 큐 대기 시간 비율 (단위:%)

사이트의 수가 3인 경우 자원의 사용 비율				
	CC	ROWA	PC	PCWC
일반 연산 큐	0.03	0.46	8.92	7.68
메시지 큐	0.09	0.18	1.01	0.62
블록 큐	2.92	2.67	0.13	0.08
디스크 큐	96.95	96.67	89.90	91.59
채널 큐	0.01	0.02	0.04	0.03

사이트의 수가 9인 경우 자원의 사용 비율				
	CC	ROWA	PC	PCWC
일반 연산 큐	0.02	0.40	8.12	7.14
메시지 큐	0.07	0.20	1.02	0.70
블록 큐	2.72	2.47	0.13	0.09
디스크 큐	97.18	96.91	90.70	92.04
채널 큐	0.01	0.02	0.04	0.03

사이트의 수가 15인 경우 자원의 사용 비율				
	CC	ROWA	PC	PCWC
일반 연산 큐	0.01	0.38	8.01	7.03
메시지 큐	0.08	0.19	1.01	0.76
블록 큐	2.68	2.67	0.14	0.11
디스크 큐	97.22	96.73	90.79	91.06
채널 큐	0.01	0.03	0.05	0.04

로그가 필요한 경우마다 원격 사이트의 시스템 카탈로그를 액세스하므로, 데이터 전송과 메시지 처리 처리를 위한 오버헤드가 발생하지만, PCWC에서는 읽기 질의의 경우에 캐시되어 있는 시스템 카탈로그를 사용할 수 있기 때문에 원격 사이트의 시스템 카탈로그를 읽어오는 오는 경우가 PC보다 적어서 평균 시간이 짧다.

5.2 터미널 수의 변화에 따른 성능 평가

각 사이트의 부하 증가가 시스템 카탈로그 관리 기법들의 성능에 미치는 영향을 분석하기 위해 한 사이트의 터미널 수를 5에서 50까지 변화시키면서 읽기 질의를 컴파일하는데 걸리는 평균 시간과 쓰기 질의를 컴파일하는 평균 시간을 측정하였다. 이 실험에서 분산 XML 저장 시스템의 사이트의 수는 7개이며, 시스템 카탈로그의 읽기 질의 비율과 지역 질의 비율은 각각 80%로 가정하였다. 이 실험의 결과는 그림 5~그림 8과 같다.

그림 5~그림 8이 보여주듯이 읽기 질의, 쓰기 질의 모두에서 CC에서 컴파일하는데 걸리는 평균 시간이 가장 길고 PC에서 컴파일하는데 걸리는 평균 시간이 가장 짧은 것을 알 수 있다. 또한 모든 시스템 카탈로그 관리 기법에 대해서 터미널의 수가 증가할수록 응답시간이 급격히 증가하는 것을 볼 수 있다. 이것은 터미널의 수가 증가할수록 시스템 전체의 부하가 증가하기 때문이다.

분할식 시스템 카탈로그 관리 기법의 PC와 PCWC에서 컴파일하는데 걸리는 평균 시간이 다르다. 이것은 디스크 액세스 빈도와 재컴파일 비율의 차이 때문이다. 재컴파일 비율은 전체 읽기 질의 중에서 재컴파일이 발생한 질의의 비율이다. PC에서는 원격 사이트의 시스템 카탈로그 정보가 필요할 때마다 원격 사이트에서 시스템 카탈로그 정보를 가져오게 되므로 재컴파일이 적게 일어난다.

PCWC의 경우에는 원격 사이트의 시스템 카탈로그 정보를 지역 사이트에 캐시로 저장하므로 디스크 액세스가 상대적으로 빈번하게 일어난다. 로컬 사이트의 디스크에 저장된 원격 사이트의 시스템 카탈로그 정보는 다음에 지역 사이트에서 동일한 원격 사이트의 객체를 액세스할 때 사용할 수 있다. 하지만 원격 사이트의 객체가 수정되었다면, 지역 사이트의 시스템 카탈로그 정보가 최신의 정보가 아니므로 원격

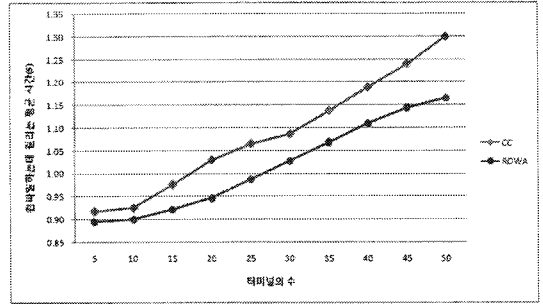


그림 5. 터미널 수의 변화에 따른 쓰기 질의 컴파일에 걸리는 평균 시간 - CC, ROWA

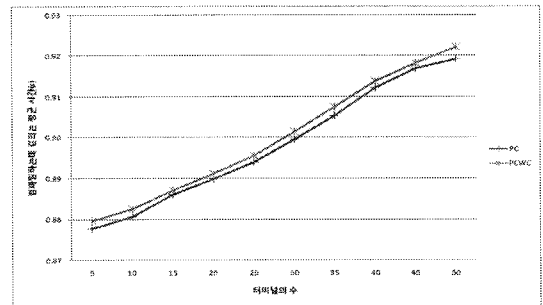


그림 6. 터미널 수의 변화에 따른 읽기 질의 컴파일에 걸리는 평균 시간 - PC, PCWC

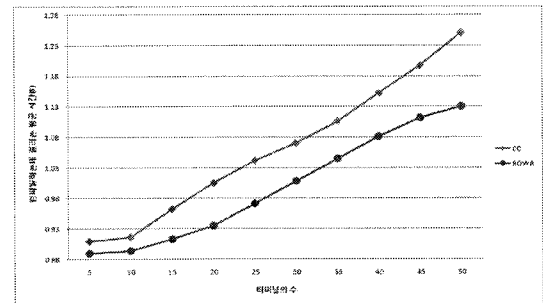


그림 7. 터미널 수의 변화에 따른 읽기 질의 컴파일에 걸리는 평균 시간 - CC, ROWA

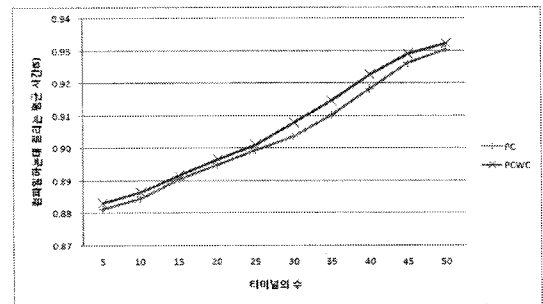


그림 8. 터미널 수의 변화에 따른 읽기 질의 컴파일에 걸리는 평균 시간 - PC, PCWC

사이트의 최신의 시스템 카탈로그 정보를 지역 사이트로 가져와 다시 컴파일해야 한다. 또한 재컴파일할 때 CPU, 디스크, 네트워크 등의 자원과 로킹 테이블을 추가로 액세스하게 되므로 여러 큐에서의 대기 시간이 증가하게 된다. 그러므로 재컴파일 비율이 증가하면 여러 큐에서의 대기 시간의 증가 때문에 읽기 질의의 성능뿐만 아니라 쓰기 질의의 성능도 나빠지게 된다.

5.3 읽기 질의의 비율 변화에 따른 성능 평가

이 실험은 읽기 질의의 비율이 시스템 카탈로그 관리 기법들의 성능에 미치는 영향을 알아보기 위해 수행하였다. 읽기 질의의 비율을 0%에서 100%까지 변화시키면서 실험하였다. 이 때 분산 XML 저장 시스템을 구성하고 있는 사이트의 수는 7개이며, 각 사이트마다 10개의 터미널이 있고, 지역 사이트에 대한 질의의 비율은 80%라고 가정하였다. 읽기 질의의 비율은 전체 질의의 중에서 읽기 질의가 차지하는 비율을 의미한다. 즉, 읽기 질의의 수가 0이면 0%, 반대로 모든 질의가 읽기 질의인 경우는 100%가 된다.

이 실험의 결과는 그림 9~그림 12가 나타낸다. 이 실험 역시 CC에서 컴파일하는데 걸리는 평균 시간이 가장 길고, PCWC에서 컴파일하는 데 걸리는 평균 시간이 가장 짧다.

읽기 질의의 비율이 높아질수록 대부분의 읽기 질의는 질의가 입력된 사이트에서 컴파일되고 액세스 모듈만 원격 사이트로 보내져서 저장되므로 ROWA에서는 평균 시간이 줄어든다. 분할식 시스템 카탈로그 관리 기법의 경우에는 읽기 질의의 비율이 100%이면 카탈로그 정보를 검색만 하므로 재컴파일이 발생하지 않는다. 그림 10에서 읽기 질의의 비율이 100%일 때, PCWC는 이미 모든 원격 사이트의 시스템 카탈로그 정보를 지역 사이트에 캐시로 유지하고 있으므로 읽기 질의를 컴파일하는데 걸리는 평균 시간이 가장 짧다. 하지만 PC의 경우에는 원격 사이트의 객체를 액세스하는 읽기 질의를 컴파일할 때마다 원격 사이트의 카탈로그 정보를 가져와야 하므로 읽기 질의를 컴파일하는데 걸리는 평균 시간이 PCWC보다 길다. 모든 질의가 쓰기 질의일 때, 즉 읽기 질의 비율 0%일 때는 PC와 PCWC에서 쓰기 질의를 컴파일하는데 걸리는 평균 시간이 같다.

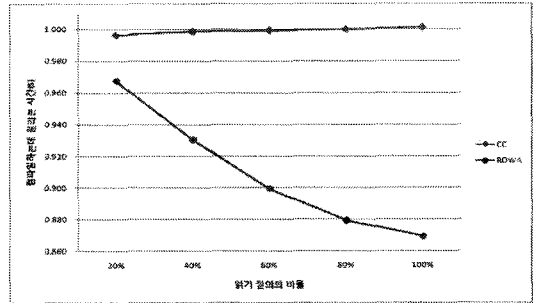


그림 9. 읽기 질의의 비율에 따른 읽기 질의 컴파일에 걸리는 평균 시간 - CC, ROWA

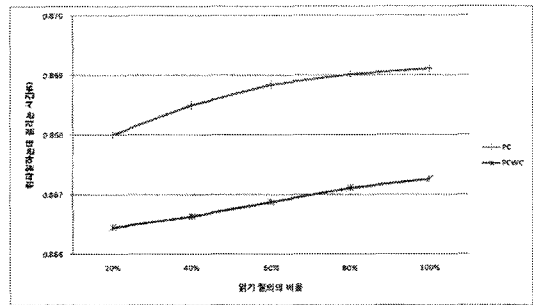


그림 10. 읽기 질의의 비율에 따른 읽기 질의 컴파일에 걸리는 평균 시간 - PC, PCWC

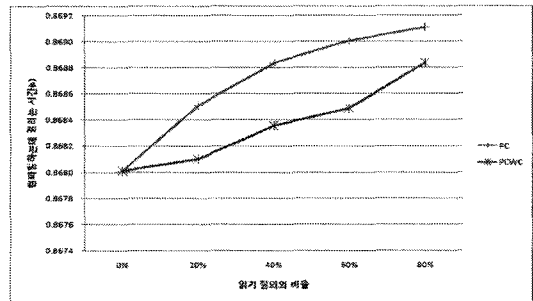


그림 11. 읽기 질의의 비율에 따른 읽기 질의 컴파일에 걸리는 평균 시간 - PC, PCWC

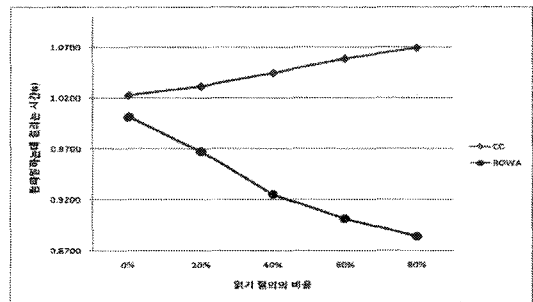


그림 12. 읽기 질의의 비율에 따른 쓰기 질의 컴파일에 걸리는 평균 시간 - CC, ROWA

5.4 지역 질의 비율 변화에 따른 성능 평가

이 실험은 지역 질의 비율이 시스템 카탈로그 관리 기법들의 성능에 미치는 영향을 조사하기 위해 수행하였다. 지역 질의 비율을 0%에서 100%까지 변화시켰다. 이 때 분산 XML 저장 시스템을 이루고 있는 사이트의 수는 7개이며, 각 사이트마다 10개의 터미널이 있고, 시스템 카탈로그에 대한 읽기 질의의 비율은 80%라고 가정하였다. 지역 질의 비율은 전체 질의에서 지역 사이트의 객체만을 액세스하는 질의의 비율을 나타낸다.

이 실험의 결과는 그림 13와 그림 14와 같다. 이 실험에서는 읽기 질의, 쓰기 질의 모두에 대해 CC에서 컴파일하는데 걸리는 평균 시간이 가장 길고, PCWC에서 컴파일하는데 걸리는 평균 시간이 가장 짧다. CC와 ROWA의 경우에는 지역 질의의 비율에 상관없이 컴파일하는데 걸리는 평균 시간이 비교적 일정하다. 지역 질의가 100%일 때는 캐시 비율에 따라 PC와 PCWC 간의 성능 차이가 발생하지 않으므로 읽기 질의와 쓰기 질의를 컴파일하는데 걸리는

평균 시간이 같음을 알 수 있다. 또한 지역 질의의 비율이 높아짐에 따라 읽기 질의와 쓰기 질의를 컴파일하는데 걸리는 평균 시간이 짧아진다. 지역 질의 비율이 100%일 때를 제외하고, PC가 읽기 질의와 쓰기 질의 모두에서 PCWC보다 컴파일하는데 걸리는 평균 시간이 짧게 나타난다.

6. 결 론

단일 XML 저장 시스템을 분산 XML 저장 시스템으로 확장하기 위해서는 중복 제어 및 시스템 카탈로그의 확장이 필수적이다. 본 논문에서는 분산 XML 저장 시스템의 시스템 카탈로그 관리 기법들 중에서 중앙 집중식 시스템 카탈로그 관리 기법, ROWA 방법을 사용하는 완전 중복식 시스템 카탈로그 관리 기법, 캐시를 사용하지 않는 분할식 카탈로그 관리 기법, 캐시를 사용하는 분할식 카탈로그 관리 기법을 성능 평가 대상으로 선정하였다.

각 시스템 관리 기법의 성능을 평가하기 위해 분산 XML 저장 시스템을 구성하는 하나의 데이터베이스 사이트의 모델을 제안하고, 각 시스템 카탈로그 관리 기법에 대해서 합리적으로 매개변수를 설정하고 시뮬레이션을 수행하여, 읽기 질의와 쓰기 질의를 컴파일하는데 걸리는 평균 시간을 측정하였다.

시뮬레이션 결과, 모든 실험에서 중앙 집중식 카탈로그 관리 기법에서 읽기 질의와 쓰기 질의의 평균 응답시간이 가장 길었다. 중앙 집중식 카탈로그 관리 기법은 모든 사이트의 시스템 카탈로그가 하나의 사이트에만 저장되어 있으므로 그 사이트가 병목 지점이 되기 때문이다. ROWA 방법을 사용하는 완전 중복식 카탈로그 관리 기법은 읽기 질의의 비율이 높은 응용에서 컴파일하는데 걸리는 시간이 짧게 나타났다. 분할식 카탈로그 관리 기법은 각 실험마다 캐시의 유무에 따라 읽기 질의와 쓰기 질의의 평균 응답 시간이 다르게 나타났다. 캐시의 유무에 따라 분할식 카탈로그 관리 기법들에서 디스크 액세스 횟수와 질의 재컴파일 비율이 달라지기 때문이다.

참 고 문 헌

[1] W3C, Extensible Markup Language(XML) 1.0 (Fourth Edition), <http://www.w3.org/TR/xml>,

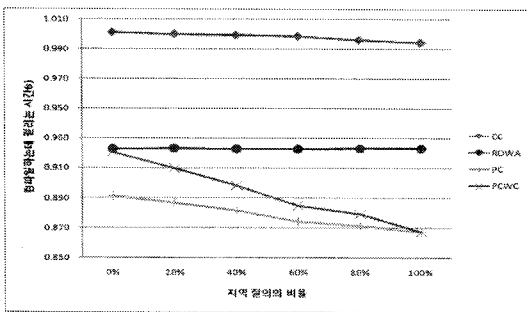


그림 13. 지역 질의의 비율에 따른 읽기 질의 컴파일에 걸리는 평균 시간

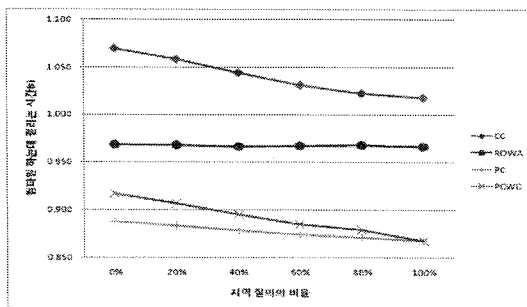


그림 14. 지역 질의의 비율에 따른 쓰기 질의 컴파일에 걸리는 평균 시간

Aug. 2006.

[2] J. Bremer and M.Gertz, "On Distributing XML Repositories," Proc. of the 6th Int'l Workshop on Web and Databases, San Diego, pp. 73-78, June 2003.

[3] S. Abiteboul, et al. "A Framework for Distributed XML Data Management," Proc. of the 10th Int'l Conf. on Extending Database Technology, Munich, Germany, pp. 1049-1058, Mar. 2006.

[4] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 5th Edition, Addison-Wesleyh, Boston, MA, 2006.

[5] R. McLab and F. W. Howell, "Using Java for Discrete Event Simulation," Proc. 12th UK Computer and Telecommunication Performance Engineering Workshop, Edinburgh, Scotland, pp. 219-228, Sept. 1996.

[6] S. Forgel, et al. Oracle Database Administrator's Guide, 11g Release 1(11.1), Oracle Corporation, Oct. 2007.

[7] 이재민, 장건업, 홍의경, "객체 관계 데이터베이스 기반에서 XML 문서를 검색하는 분산 XQuery 질의 처리기의 설계 및 구현," 데이터베이스소사이터 데이터베이스연구회지, 제24권, 제1호, pp. 1-16, Apr. 2008.

[8] D. Suci, "Distributed Query Evaluation on Semistructured Data," ACM Transactions on Database Systems, Vol.27, No.2, pp. 1-62, Mar. 2002.

[9] I. Manolescu, et al., "Answering XML Queries over Heterogeneous Data Sources," Proc. of the 27th VLDB Conf., pp. 241-250, Sept. 2001.

[10] G. M. Lohman, et al., "Query Processing in R*: A Distributed Database Manager," IBM Research Report RJ3720, San Jose, California, Jan. 1983.

[11] D. Kossmann, "The State of the Art in Distributed Query Processing," ACM Computing Surveys, Vol. 32, No. 4, pp.

422-429, Dec. 2000.

[12] P. G. Selinger, et al., "Access Path Selection in a Relational Database Management System," Proc. of ACM SIGMOD Int'l Conf. on Management of Data, Boston, Massachusetts, pp. 23-34, 1979.

[13] M. T. Ozs, and P. Valduriez, Principles of Distributed Database Systems, 2nd Edition, Prentice Hall, New Jersey, 1999.

[14] W3C, XQuery 1.0: An XML Query Language, <http://www.w3.org/TR/xquery>, Jan. 2007.

[15] I. Tatarinov, et al., "Updating XML," Proc. of ACM SIGMOD Int'l Conf., on Management of Data, Philadelphia, Pennsylvania, pp. 413-424, May 2001.

[16] E. K. Hong, and J. W. Cho, "Performance Evaluation of Catalog Management Schemes in Distributed Database Systems," Information Systems Vol. 16, No. 2. Apr. 1991.



장 건 업

2005년 서울시립대학교 컴퓨터 통계학과 학사
 2007년 서울시립대학교 컴퓨터 통계학과 컴퓨터학전공 석사
 2008년 서울시립대학교 컴퓨터 과학부 박사과정 재학 중

관심분야 : XML, 데이터베이스 성능 평가, 분산 데이터베이스, 정보 검색, 데이터 마이닝



홍 의 경

1981년 서울대학교 사범대학교 수학교육과 학사
 1983년 한국과학기술원 전산학과 석사
 1991년 한국과학기술원 전산학과 박사
 1984년 3월~현재 서울시립대학교 컴퓨터과학부 교수

관심분야 : XML, 데이터베이스 성능 평가, 분산데이터베이스, 데이터 마이닝