

KRISTAL-2002를 위한 JAVA 사용자 API의

설계 및 구현

주원균^o 정창후 이민호 양명석 최윤수 최기석

한국과학기술정보연구원 정보시스템개발실

{joo^o, chjeong, cokeman, msyang, armian, kschoi}@kisti.re.kr

Development of Java API for KRISTAL-2002

Wonkyun Joo^o C.H. Jeong M.H. Lee M.S. Yang Y.S. Choi K.S. Choi

Dept. of Information System Development

KISTI(Korea Institute of Science and Technology Information)

요 약

본 논문에서는 트랜잭션 기반의 정형/비정형/XML 데이터 관리 및 검색 기능을 갖춘 정보 관리 시스템 KRISTAL-2002를 대상으로 하여, JAVA를 구현언어로 사용하는 개발자가 해당 시스템의 기능을 원활하게 사용할 수 있도록 하기 위한 KRISTAL-2002 사용자 프로토콜 및 JAVA 기반의 API 설계를 목표로 한다. KRISTAL-2002 프로토콜은 TCP/IP 기반 XML 메시징 프로토콜에 기반을 둔 것으로서, 클라이언트와 서버 모두에서 처리 효율을 높일 수 있다. 이러한 프로토콜을 기반으로 3개 그룹에 걸쳐 총 24개의 KRISTAL-2002 JAVA API를 설계/구현하였다.

1. 서 론

정보 시스템에서 다루는 전문 데이터의 수와 용량이 증가함에 따라 대용량 데이터의 처리에 초점을 맞춘 연구들이 많이 수행되었는데, 그 중 정보의 저장 및 검색 부분에서 많은 발전이 있었다.

이런 과정에서 전체 데이터를 여러 개의 DB로 나눈 뒤, 한 시스템 혹은 여러 시스템에 분산 저장하고 이를 통합 검색하는 연구들이 수행되었다. 이런 일련의 연구들 중 하비스트[1]는 초기에 제안되었던 시스템으로, 분산 저장 검색 시스템의 기본적인 구조에 대해 설계/구현하였다. 검색 측면에서 본다면, 에이전트 기반의 분산 검색 모델을 선보인 Harness, 웹을 기반으로 하여 통합 검색 방법을 제시한 여러 메타 검색 엔진들도 있었다. 정보 저장 및 검색을 동시에 지원하기 위한 밀-결합 형태의 연구로서는 국내의 대표적인 예로 바다-3[9]와 오디세우스[5]를 들 수 있다. 최근에는 Google이 파일 시스템(GFS: Google File System) 기반의 분산 저장 검색 시스템을 선보였다[8]. 이러한 연구들은 현재 분산 검색, 통합 검색, 분산 통합 검색, 분산 저장, 분산 저장 검색이라는 용어들로 대표되고 있다.

본 논문에서 대상으로 한 KRISTAL-2002 시스템은 분산 저장 검색의 맥락에서, 대용량 데이터의 분산 저장 검색 기능을 기본으로 제공한다. 무엇보다도, 그 동안 검색 시스템의 문제점으로 지적되었던 데이터 관리기능을 개선하고 적당한 규모의 온라인 데이터 처리에 대한 신뢰성을 보장하기 위해, 데이터의 삽입, 수정, 삭제 등의 기능 수행 시 보다 향상된 트랜잭션

처리와 회복 기법을 도입하여 제공한다.

본 논문에서는 분산 정보 관리 시스템 KRISTAL-2002의 기능을 사용자에게 제공하기 위한 JAVA API의 설계와 그 구현에 초점을 맞추었다. 2장에서는 KRISTAL-2000에 비해 달라진 KRISTAL-2002 시스템의 새로운 구조 및 주요 기능들에 대해서 설명하고, 3장에서는 XML 메시징 프로토콜을 기반으로 한 JAVA API의 설계 및 구현 과정에 대해 설명하고, 4장에서 결론을 맺는다.

2. 새로운 KRISTAL-2002 시스템 구조 및 기능

KRISTAL-2002를 위한 JAVA 사용자 API 설계에 앞서, 기존 2000 버전에 비해 향상된 구조와 기능을 살펴볼 필요가 있다. 많은 변경 사항들이 있지만, 몇 가지의 큰 변경 사항만을 언급한다. 이러한 내용들은 통합적으로 반영되어 그림 1에 자세히 도식되어 있다.

□ 하부 저장 엔진의 변경

기존 Shore저장 엔진[2]은 트랜잭션 지원 측면에서는 우수했지만 대용량 저장과정에서 구조상의 문제점으로 인해 많은 메모리 문제를 보였고, KRISTAL-2002에서는 트랜잭션과 대용량 처리에 모두 효율적인 Berkeley DB[3]를 하부 저장 엔진으로 채택하였다.

□ 프로세스 구조의 변경

이전 구조에서는 메인 프로세스, 데이터관리 프로세스, 셋 관리 프로세스가 모두 별도의 프로세스로 동작함으로써 프로세스의 비효율적 관리를 야기하였기 때문에, 이러한 문제의 해결

을 위해 단일 프로세스 다중 쓰레드 구조로 변경하였다.

□ XML 문서의 지원

XML 문서의 저장 및 검색을 지원하기 위해서 구조문서변환기 모듈을 설계/개발하였다. 이 변환기는 원본 XML 문서를 KRISTAL에서 처리할 수 있는 중간 XML 문서로 변환하여 KRISTAL로의 적재 및 서비스를 가능하게 한다.

□ BLOB 데이터의 지원

BLOB을 지원하기 위한 데이터 타입을 제공한다.

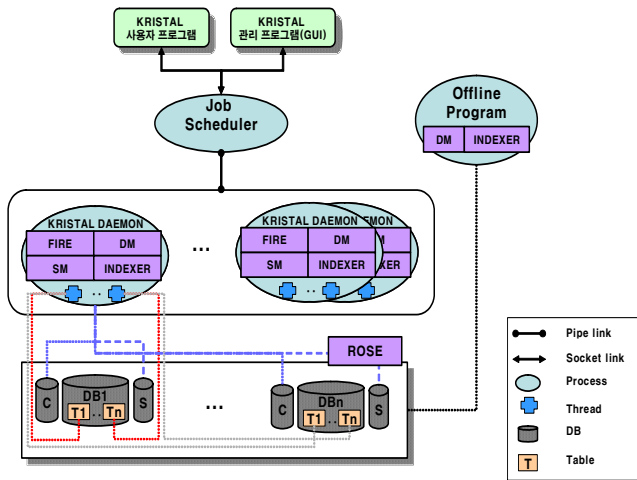
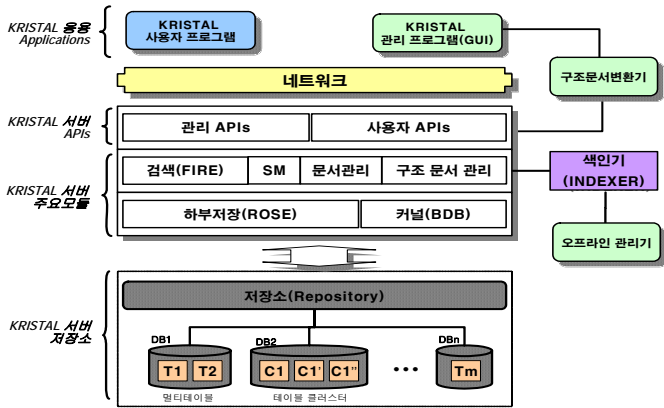


그림 1 KRISTAL-2002 시스템의 구조

3. KRISTAL-2002 JAVA API 설계

본 장에서는 KRISTAL-2002를 위한 JAVA API 설계 시 고려한 점, 프로토콜 구조, API 명세, 클래스들에 대해서 설명한다.

3.1 KRISTAL-2002 API 설계의 초점

- 서버와 클라이언트 모두 프로토콜의 처리가 용이해야 한다.
- 제공 API를 이용해서 KRISTAL 응용 프로그램의 작성이 용이해야 한다.
- 새로운 API의 추가가 용이해야 한다.
- XML 형식의 온라인 데이터 처리와 검색이 가능해야 한다.

3.2 KRISTAL-2002 프로토콜 구조

KRISTAL-2002 프로토콜에서 서버와 클라이언트간의 메시지

는 TCP/IP 기반으로 하는 XML 형태의 메시징 프로토콜을 사용하는데, 이것은 "OAGIS 8.0[10]"의 메시지 형식과 비슷한 구조로 이루어져 있다(그림 2 참조). KRISTAL-2002 프로토콜에서 XML 메시지는 크게 KRISTAL-2002 프로토콜에 관한 정보를 담고 있는 부분(Header)과 각 서비스 파라미터와 관련된 부분(Body)으로 구성되어 있다.

3.2.1 Header의 내용

□ Version

프로토콜의 버전(version)을 표시한다. 만약 프로토콜 버전이 맞지 않을 경우에는 버전 불일치 에러로 처리한다. 버전의 변경은 프로토콜의 구조가 변하는 Major 변환과 서비스명이나 파라미터가 변경되는 Minor 변환이 있을 수 있다. 버전의 표시는 "version 1.0" 과 같이 "Major번호.Minor 번호" 로 표기한다.

3.2.2 Body의 내용

Body는 서비스와 관련된 내용이 들어가며, Process와 Object로 이루어져있다. 즉, 'Object를 가지고 Process를 하라' 라는 의미이다.



그림 2 KRISTAL-2002 XML 메시징 프로토콜 구조

□ 프로세스(Process)

서비스명을 나타낸다. 요청 메시지의 경우 영문자로 된 서비스명을 사용하고, 서버로부터 요청에 대한 응답 메시지인 경우에는 요청 메시지의 영문자 서비스명 뒤에 "_RESPONSE"를 붙인다. 예를 들어 요청 서비스가 검색일 경우, 요청 서비스명은 "RETRIEVE"이고, 응답 서비스명은 "RETRIEVE_RESPONSE" 이다. 에러가 발생할 경우에는 요청 서비스명에 "_RESPONSE"를 붙이는 것이 아니라 에러를 나타내는 "ERROR" 을 사용한다.

□ 객체(Object)

서비스에 관계된 파라미터를 나타낸다. 요청 메시지만 경우에는 입력 파라미터를 나타내며, 응답메시지인 경우에는 서버로부터의 요청 결과 값을 나타낸다. 파라미터 값들은 구조적인 형태를 가지고 각 서비스 별로 다르게 표현되는데, 이 값들은 KRISTAL-2002 개발자 매뉴얼[6]의 서비스별 파라미터 항목에 자세히 기술되어 있다. 이때의 파라미터 값들은 3.4의 클래스들과 밀접한 관계가 있다.

객체에서 사용하는 파라미터 값 중 문자열 타입의 파라미터는 Base64 코드를 사용하여 인코딩(encoding) 된 메시지를 전송하고, 메시지 수신부에서는 다시 원래의 문자열로 복원하기 위해서 Base64 디코딩(decoding)을 수행한다.

3.3 KRISTAL-2002 API 명세

KRISTAL-2002 API는 크게 3 그룹으로 구분될 수 있는데, 각 그룹 내에 속한 API에 대한 설명은 표 1에서 언급한다.

표 1 KRISTAL-2002 API 명세

그룹 1: 서버 관리 API	
CHECK_STATUS	서버 상태 확인
그룹 2: 정형/비정형 문서 관련 API	
GET_DB_INFO	DB정보 구함(스키마/테이블 등)
META_INFO_QUERY	질의 정보 구함
RETRIEVE	기본 검색
RETRIEVE_SIMILAR_DOCUMENTS	유사 문서 검색
RETRIEVE_IN_RESULT	결과내 검색
GET_DOCUMENTS_FROM_RESULT	결과셋으로부터 문서내용 얻기
GET_DOCUMENTS_WITH_IDS	문서 ID로 문서내용 얻기
GET_DOCUMENTS_WITH_PRIMARY_KEY	기본키로 문서내용 얻기
GET_DOCUMENTS_ID_WITH_DOCUMENT	문서에 해당하는 문서 ID 구함
SORT_BY_SECTION	결과셋을 특정 섹션명으로 정렬
PROCESS_DATABSE_SCHEMA	데이터베이스 스키마 처리
APPEND_DOCUMENT	문서 삽입
DELETE_DOCUMENT	문서 삭제
UPDATE_DOCUMENT	문서 수정
그룹 3: 정형/비정형 문서 관리 API	
GET_XML_NODES_FROM_RESULT	결과셋에서 노드들 얻기
GET_XML_NODES_WITH_IDS	문서 ID로 노드들 얻기
GET_XML_NODES_INFO	특정 노드에 대한 정보 구함
GET_XML_TREE	XML 트리 정보 구함
RETRIEVE_DOCUMENTS_WITH_FOREIGN_KEY	외부 키로 문서 검색
APPEND_XML_NODE	XML 문서 삽입
UPDATE_XML_NODE	XML 문서 수정
DELETE_XML_NODE	XML 문서 삭제
MOVE_XML_NODE	XML 문서 이동

3.4 KRISTAL-2002 JAVA API를 위한 CLASS들

위에서 명세한 API는 그 쓰임새에 따라 여러 클래스 객체들을 필요로 한다. KRISTAL-2002 JAVA API에서 필요로 하는 모든 객체들을 그림 3에 명세하였다. 가장 처음에 도식된 입출력 파라미터 클래스가 가장 상위의 클래스로서 모든 종류의 클래스를 포함하고 있고, 하위 클래스들은 각 쓰임새에 맞게 조

합되어 사용된다.

3.3에서 명세한 API들은 모두 2개의 파라미터를 가지는데, 가장 상위의 입출력 파라미터 클래스를 각각 입력과 출력으로 사용한다.(KRISTAL-2002 개발자 매뉴얼[6]에 보다 상세히 기술되어 있음)

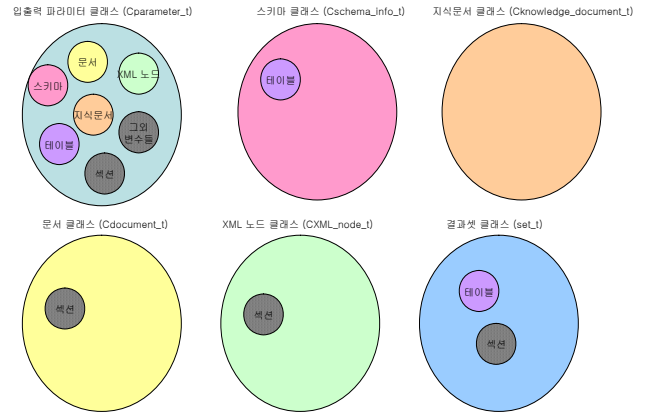


그림 3 KRISTAL-2002 JAVA API 파라미터 클래스들

4. 결론

본 논문은 JAVA를 개발 언어로 사용하는 개발자가 분산 정보 관리시스템인 KRISTAL-2002를 이용하기 위한 JAVA 기반의 사용자 API 설계 및 구현에 그 목적이 있다. 사용자와 서버를 연결시키는 과정에서 TCP/IP 기반 XML 메시징 프로토콜을 매개체로 삼았다. XML 메시징은 플랫폼 독립성뿐만 아니라 프로토콜 처리의 명확성과 수월함을 함께 제공한다. 또한, 서버 관리, 정형/비정형/XML 문서의 저장 및 관리, 검색 서비스를 제공하기 위하여 여러 API들을 설계/구현하였다.

참고문헌

[1] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. Harvest: A Scalable, Customizable Discovery and Access System Technical Report CU-CS-732-94, August 26, 1994 Department of Computer Science, University of Colorado - Boulder.
 [2] Shore, <http://www.cs.wisc.edu/shore/>
 [3] Berkeley DB, <http://www.sleepycat.com/>
 [4] Google, <http://www.google.com/>
 [5] 오디세우스, <http://odysseus.kaist.ac.kr/>
 [6] KRISTAL-2002 기술 매뉴얼, <http://giis.kisti.re.kr/>
 [7] 주원균, 이민호, 강우영, "분산 정보 관리 시스템을 위한 사용자 API 설계 및 구현", 정보과학회 추계 학술 발표 논문집, 2002
 [8] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System", ACM SOSP'03, p29-43, 2003
 [9] 백광진, "바다-III/자바 API 설계", 한국통신학회 종합학술발표회 논문집, 97권, P90-95쪽
 [10] OAGIS, <http://www.openapplications.org/oagis/index.htm>