

KRISTAL 데이터베이스 관리를 위한 리스너 설계 및 구현에 관한 연구

이석형^o 최성필 윤화욱
한국과학기술정보연구원 시스템개발팀
{skyi^o, spchoi, hmyoon}@kisti.re.kr

Development of KRISTAL-Listener for KRISTAL Database Management

Seok-Hyoung^o S.P. Choi H.M.Yoon
System Development Team, KISTI

요 약

일반적으로 Oracle과 같은 RDBMS에서 클라이언트가 데이터베이스를 접근하거나 관리하기 위해서는 RDBMS 서버에 리스너가 실행되어야 한다. 리스너는 네트워크를 이용해 클라이언트에서 서버로 연결하기 위한 네트워크 관리자로서, 수행정보의 전송 및 관리 기능을 수행한다. 이에 반해, 정보검색시스템에서의 시스템 기동 및 폐쇄, 데이터의 관리는 온라인이 아닌 오프라인 작업으로 이루어지는 것이 보통이다. 이는 각 시스템이 추구하는 것이 관리와 검색으로 각각 다르기 때문에 구조적인 차이가 존재한다. 정보검색관리시스템 KRISTAL은 기본적으로 정보검색엔진기반의 DBMS를 적용하였기 때문에, 데이터베이스 관리도구 자체가 엔진과 밀접함된 형태가 아닌, 별도의 API로 제공되기 때문에 클라이언트가 온라인상으로 서버에게 관리 기능을 요청할 수 없는 단점이 있다. 이에 본 논문에서는 KRISTAL기반 데이터베이스 관리를 위한 KRISTAL 리스너를 제안한다. KRISTAL 리스너는 RDBMS 리스너의 일반적인 기능뿐만 아니라 KRISTAL 데이터베이스 검색 데몬 프로세스 관리, 스키마 및 테이블 관리, 백업관리, 분산검색 데몬 등 온라인으로 KRISTAL 서버를 제어할 수 있는 관리 기능을 제공한다.

1. 서 론

인터넷의 발달로 인해 사용자들은 웹을 이용하여 많은 정보를 습득할 수 있게 되었고, 정보 수요가 증가함에 따라 다양하고 양질의 정보를 제공하기 위한 많은 방법이 연구되고 활용되고 있다. 이러한 고급정보를 종합적으로 관리하고, 이용자에게 빠르고 편리한 검색서비스를 제공하기 위해 정보 시스템에서의 정보의 저장 및 검색부분에 관한 많은 연구가 이루어지고 있으며 이러한 연구를 바탕으로 한 응용시스템들이 개발 및 이용되고 있다[1]. 일반적으로 데이터베이스를 구축하고 서비스하기 위한 대부분의 방식은 상용 DBMS를 이용하여 정보의 관리를 수행하고, 사용자 검색은 정보검색 시스템을 사용하는 DBMS와 정보검색 시스템의 연동 방식을 채택하고 있다[2]. 이러한 방식은 DBMS의 장점인 대용량 데이터 트랜잭션의 용이성과 정보검색 시스템의 장점인 빠른 정보 검색 시간 제공이 가능하다는 점에서 많이 이용되고 있으나, DBMS와 정보검색 시스템을 이중으로 운영해야 하는 불편함이 있으며, 두 시스템간의 데이터의 일관성을 유지하는데 많은 어려움이 따른다. 특히, 최근의 정보는

재가공 및 재사용이 매우 높기 때문에, DBMS와 정보검색시스템의 연동 방식의 정보시스템에서 이를 실시간으로 반영하기 위해서는 많은 노력과 위험성이 존재한다. 이를 해소하기 위해, 정보검색시스템에 DBMS의 기능을 적용하기 위한 노력이 시도되었는데 그 대표적인 시스템이 KRISTAL이다[1]. KRISTAL은 기본적으로 정보검색엔진기반에 DBMS적인 요소를 결합시킨 정보검색관리시스템으로 다양한 형태의 관리 도구를 제공한다. 그러나, 저장 엔진자체가 검색시스템에 맞도록 구성되어 있기 때문에, 관리도구 자체가 엔진과 밀접함된 형태가 아닌, 별도의 API로 제공되기 때문에 클라이언트가 온라인상으로 서버에게 관리 기능을 요청할 수 없는 단점이 있다. 이에 본 논문에서는 KRISTAL기반 데이터베이스 관리를 위한 KRISTAL 리스너를 제안한다. KRISTAL 리스너는 RDBMS 리스너의 일반적인 기능뿐만 아니라 KRISTAL 데이터베이스 검색 데몬 프로세스 관리, 스키마 및 테이블 관리, 백업관리, 분산검색 데몬 등 온라인으로 KRISTAL 서버를 제어할 수 있는 관리 기능을 제공한다.

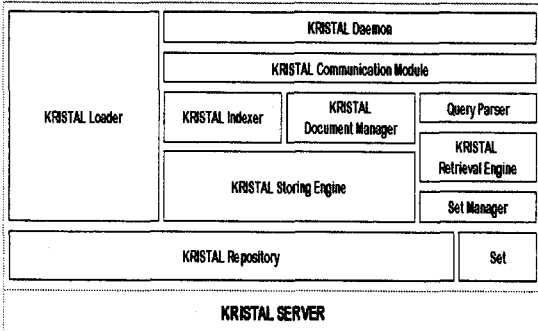
본 논문의 구성은 다음과 같다. 2장에서는 정보검색관리시스템 KRISTAL의 주요 기능과 구조, 그리고 관리요

소에 대해 설명하고, 3장에서는 KRISTAL 리스너에 대해 기술한다. 4장에서는 클라이언트를 위한 KRISTAL 리스너 API에 대해 설명하고 5장에서 결론을 맺는다.

2. KRISTAL 관리요소

2.1 KRISTAL

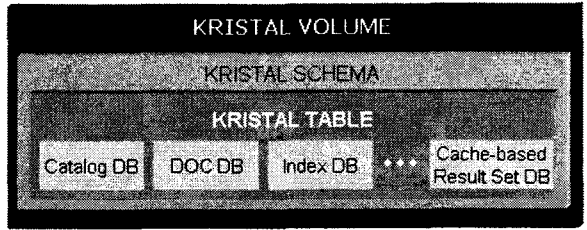
KRISTAL은 한국과학기술정보연구원에서 개발한 정보검색관리시스템으로, UNIX 및 LINUX 기반에서 데이터에 대한 저장, 관리 및 검색에 대한 처리를 효율적으로 수행할 수 있는 시스템이다[3]. 최근 대표적인 포털사이트 등에서 자체 개발하여 서비스 하고 있는 검색엔진과 달리 KRISTAL은 정보검색엔진에 데이터베이스 관리기능(DBMS)을 추가한 정보검색관리시스템(IRMS)이다. 현재 KRISTAL은 국내의 대표적인 과학기술포털사이트인 yeskisti 를 비롯한 국가편찬위원회, 한국역사통합정보시스템 등 과학기술정보와 공공기술정보서비스에 적용 중에 있다. [그림 1]은 KRISTAL의 주요 컴포넌트들을 계층적으로 도식화 한 그림이다.



[그림 1] KRISTAL 데이터베이스 구성요소

2.2 KRISTAL 구조

KRISTAL에서 하나의 DB에는 개별 테이블 구조를 정의하는 스키마들이 존재한다. 스키마에는 데이터베이스의 물리적인 위치 및 구성요소 뿐만 아니라 적용 테이블 리스트, 그리고 색션 정보(자료형, 색인 타입, 색인옵션) 등이 존재한다. 검색은 검색 데몬 프로세스를 통해 클라이언트의 요청에 응답하는 형태로 이루어지며, 여러 개의 DB를 통한 검색할 수 있는 분산통합검색 데몬 프로세스가 존재할 수 있다. 분산통합검색 데몬 프로세스는 각 원천DB의 데몬 프로세스와 통신하면서, 검색결과를 요청한다. [그림 2]는 KRISTAL 데이터베이스(Volume), 스키마, 테이블간의 상관관계를 나타낸 그림이다.



[그림 2] KRISTAL Repository 구조

2.3 KRISTAL 관리요소

정보검색관리시스템의 DB관리 요소는 다음과 같다

- DB Start-UP/Shut-Down(DB개방/폐쇄)
- 스키마, 테이블 관리
- 분산정보검색데몬의 관리
- 데이터 백업/복구 등

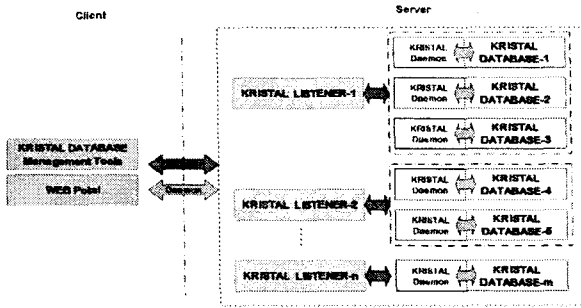
일반적으로 Oracle과 같은 RDBMS에서 클라이언트가 데이터베이스를 접근하거나 관리하기 위해서는 RDBMS 서버에 리스너가 실행되어야 한다. 리스너는 네트워크를 이용해 클라이언트에서 서버로 연결하기 위한 네트워크 관리자로서, 수행정보의 전송 및 관리 기능을 수행한다 [4]. 이에 반해, 정보검색시스템에서의 시스템 기동 및 폐쇄, 데이터의 관리는 온라인이 아닌 오프라인 작업으로 이루어지는 것이 보통이다. KRISTAL은 기본적으로 정보검색엔진기반의 DBMS를 적용하였기 때문에, 오프라인 중심의 데이터 관리를 주로 지원하였으며, 위와 같은 DB 관리 요소 등을 수행하기 위해서는 사용자가 직접 서버에 접근하여 수행하는 불편함이 있다.

3. 시스템 구현

3.1 KRISTAL 리스너

2장에서 살펴본 바와 같이, KRISTAL과 같은 정보검색관리시스템에서 데이터베이스 관리자와 개발자가 손쉽게 온라인 상에서 시스템과 데이터를 효율적으로 관리하기 위한 KRISTAL 리스너가 필요하다. KRISTAL 리스너는 RDBMS에서 적용하는 리스너의 일반적인 기능뿐만 아니라 KRISTAL 데이터베이스 검색 데몬 프로세스 관리, 스키마 및 테이블 관리, 백업관리, 분산검색 데몬 등 온라인 상으로 KRISTAL 서버를 직접적으로 제어할 수 있는 관리 기능을 제공하는 서버 프로세스이다. 이 프로세스는 KRISTAL 이 기본적으로 제공하는 검색데몬프로세스와 독립적으로 작동되며, 클라이언트의 KRISTAL 관리도구 실행 요청을 받아 KRISTAL 관리 API를 실행하고, 그 결과를 클라이언트로 전송하는 역할을 수행한다. [그림

3]은 클라이언트와 KRISTAL 리스너, 그리고 KRISTAL 엔진과의 관계를 표현한 그림이다.



[그림 3] KRISTAL 리스너와 KRISTAL 엔진

3.2 주요기능

KRISTAL 리스너는 다음과 같은 기능을 제공한다.

- KRISTAL 서버정보 관리
- KRISTAL 서버기동/폐쇄
- KRISTAL 데이터베이스(Volume) 관리(생성/삭제)
- KRISTAL Schema/Table 관리(생성/수정/삭제)
- 대용량 데이터 일괄적재 및 색인
- 분산통합검색DB 관리(서버 기동/폐쇄)
- KRISTAL DB 백업/이주

3.3 시스템

3.3.1 클라이언트- 리스너 메시지 전송

KRISTAL 리스너와 클라이언트간 통신 메시지는 TCP/IP 기반으로 하는 XML 형태의 메시징 프로토콜을 사용하여 전송한다[5]. 이는 OAGIS 8.0[6]의 메시지 형식과 비슷한 구조로 이루어졌으며, 기존의 KRISTAL 검색에 사용되는 메시지 방식과의 호환을 위해 같은 프로토콜을 적용하였다. 메시지는 [그림 4]와 같이 XML 기반으로 구성되며, 크게 KRISTAL 프로토콜 정보를 저장하는 Header와 요청 프로세스 정보를 저장하는 Body 부분으로 이루어진다. Version은 KRISTAL 서버 버전이며 이 버전이 KRISTAL 리스너 클라이언트 버전과 불일치 되면 에러로 처리한다. Body의 Process는 서비스 명을 나타내는 것으로 클라이언트로부터의 요청메시지는 4장에서 설명되며, 요청에 대한 응답은 "_RESPONSE"를 붙여 클라이언트로 전송된다. Object는 Process 요청시에 필요한 입력 및 출력 파라미터들을 나타내며, 보통 KRISTAL 데이터베이스를 관리하는 데 필요한 정보들이 기술된다. 객체에서 사용하는 파라미터 값 중에서 문자열 타입의 파라미터는 Base64 코드를 사용하여 인코딩

된 메시지를 전송하며, 수신부에서 Base64 디코딩을 수행하여 원래의 문자열로 복원된다.

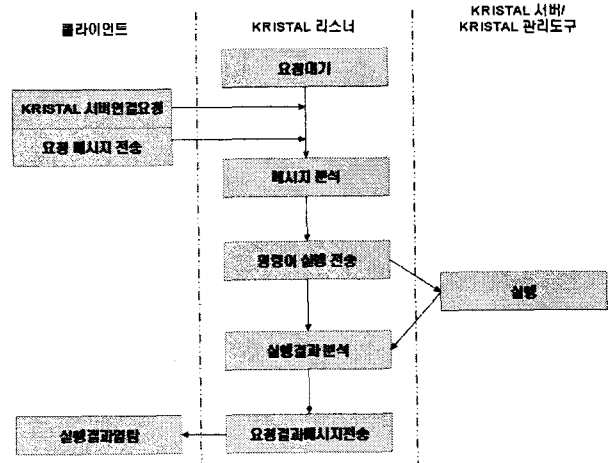
```

<Message>
  <Header>
    <Version>KRISTAL-3.0.1</Version>
  </Header>
  <Body>
    <Process>
      KRISTAL_DAEMON_START
    </Process>
    <Object>
      <Server_ID> blue</ServerID>
      <ConfigName>blue.xml</ConfigName>
      ....
    </Object>
  </Body>
</Message>
    
```

[그림 4] 클라이언트-리스너 메시징 프로토콜 구조

3.3.2 시스템 흐름

클라이언트로부터의 요청을 수행하기 위한 흐름은 [그림 5]와 같다. KRISTAL 리스너가 클라이언트로부터 요청을 받으면 클라이언트 메시지의 인코딩을 수행하여 KRISTAL 프로토콜 정보와 프로세스 정보를 분리한다. 그리고 요청 프로세스에 따라 해당 KRISTAL API를 실행하고 실행 결과를 분석하여, 그 결과를 메시징 작업을 통해 다시 클라이언트로 전송하는 단계를 거친다.



[그림 5] KRISTAL 리스너의 클라이언트 요청 처리

3.4 KRISTAL 리스너의 실행

KRISTAL 리스너의 실행을 위해서는 Oracle 리스너의 listener.ora와 같은 설정 파일이 존재해야 한다. 설정 파

일에는 KRISTAL 리스너의 정보와 [그림 3]과 같이 리스너와 연관된 KRISTAL 서버의 검색데몬정보, 물리적인 데이터베이스 위치정보, 적재스키마 정보, 로그정보등을 저장한다. [그림 6]은 KRISTAL 리스너 설정파일을 보인 것이다.

```
<ServerConfig>
  <ListenerServer ip_address="127.0.0.1" port="34333"
listener_root="/home/k2002/K2002/src/LISTENER"
log_dir="/home/k2002/K2002/src/LISTENER/log" />
  <KristalServerList server_id="blue01" server_type="1"
database_name="BLUE01"
kristalserver_root="~k2002/K2002/src/LISTENER/TEST/BLU
E01/" config_name="blue.xml"
config_dir="~k2002/K2002/src/LISTENER/TEST/BLUE01/"
volume_dir="~k2002/K2002/src/LISTENER/TEST/BLUE01/v
olume/"
schema_dir="~k2002/K2002/src/LISTENER/TEST/BLUE01/
schema/"stopword_dir="/home/k2002/K2002/src/LISTENER
/TEST/BLUE01/stopword/" last_create_schema=""
last_delete_schema="" last_load_schema=""
last_log_file_name="" last_daemon_start_time=""
last_daemon_stop_time="" />
  <ListenerAccessList ip="150.183.111.1" />
</ServerConfig>
```

[그림 6] KRISTAL 리스너 설정 파일

4. 클라이언트를 위한 KRISTAL 리스너 API

KRISTAL 리스너 API는 3장에서 설명한 관리기능을 모두 지원하며 그 명세는 [표 1]과 같다.

[표 1] KRISTAL 리스너 API 명세

- KRISTAL 서버 자원관리

GET_KRISTAL_SERVER	KRISTAL서버정보 획득
SET_KRISTAL_SERVER	KRISTAL서버정보 삽입, 삭제, 갱신

- KRISTAL 검색데몬 및 분산검색데몬 구동/멈춤

KRISTAL_DAEMON_START	KRISTAL 검색데몬구동
KRISTAL_DAEMON_STOP	KRISTAL 검색데몬멈춤
DKSI_DAEMON_START	KRISTAL 분산검색데몬일괄구동
DKSI_DAEMON_STOP	KRISTAL 분산검색데몬일괄멈춤

- KRISTAL 데이터베이스/스키마/테이블/데이터 관리

CREATE_KRISTAL_VOLUME	KRISTAL 데이터베이스 생성
CREATE_KRISTAL_SCHEMA	KRISTAL 스키마 생성
CREATE_KRISTAL_TABLE	KRISTAL 테이블 생성
ADD_SECTIONS	섹션추가

DROP_SECTIONS	섹션삭제
DROP_KRISTAL_TABLE	KRISTAL 테이블 삭제
DROP_KRISTAL_SCHEMA	KRISTAL 스키마 삭제
DROP_KRISTAL_VOLUME	KRISTAL 데이터베이스 삭제
LOAD_KRISTAL_TABLE	대용량 데이터 일괄적재
INDEX_KRISTAL_TABLE	대용량 데이터 일괄/부분 색인

- KRISTAL DB 백업

BACKUP_KRISTAL_DOC	KRISTAL 문서DB 백업
BACKUP_KRISTAL_IDX	KRISTAL 색인DB 백업
KRISTAL_EXPORT	KRISTAL 문서DB 다운로드

5. 결론

KRISTAL에 적용된 DBMS적인 관리 기능은 코어 엔진 특성상 매우 제한적이거나, 일반적인 검색 서비스를 위한 데이터베이스 관리를 위해서는 손색이 없다. 이에 붙여 본 논문에서 제안한 KRISTAL 리스너는 RDBMS에서의 리스너의 일반적인 기능뿐만 아니라 KRISTAL 데이터베이스 검색 데몬 프로세스 관리, 스키마 및 테이블 관리, 백업관리, 분산검색 데몬 등 온라인상으로 KRISTAL 서버를 제어할 수 있는 관리 기능을 제공하기 때문에 KRISTAL과 같은 정보검색관리시스템에서 데이터베이스 관리자와 개발자가 손쉽게 온라인상에서 시스템과 데이터를 효율적으로 관리할 수 있을 것으로 기대된다. 또한, 클라이언트와 서버간 통신에 사용된 XML 메시징 프로토콜은 플랫폼 독립성 뿐만 아니라 처리의 명확성과 수월성을 제공하여, KRISTAL 리스너 기능 확장이 용이하다. 향후 연구에서는 대용량 데이터 및 테이블 처리시 발생되는 서버 부하에 대한 KRISTAL 리스너의 서버 제어 기능 등이 필요하다.

참고문헌

[1] 주원균,정창후,이민호, "KRISTAL-2002를위한 JAVA 사용자 API의 설계 및 구현", 정보과학회 가을 학술발표논문집 Vol.31, No.2, p433-435, 2003

[2] 이민호,진두석,김광영,주원균,서정현,류범종 "정보검색관리시스템 KRISTAL-2000 설계 및 구현", 정보과학회 봄 학술발표논문집 Vol.30, No.1, p653-655, 2003

[3] KRISTAL-2002 User's manual V.1.1.5.pdf, GIS, KIST, 2002

[4] Oracle 리스너, <http://www.e-server.co.kr/>

[5] 김창수, 정희경, "XML 응용개발환경", 정보과학회지, 제19권, 제1호, pp.15-23, 2001

[6] OAGIS, <http://www.openapplications.org/index.htm>