

효율적이고 안정적인 DB-IR 통합시스템을 위한 서버 구조

Design of Server Structure for Efficiency and Stability of DB-IR Integration System

이민호, 김진숙, 최윤수, 정창후, 서정현, 윤화목
한국과학기술정보연구원

Lee min-ho, Kim jin-suk, Choi yun-soo,
Jeong chang-hoo, Seo jeong-hyun, Yoon hwa-mook
Korea Institute of Science and Technology
Information (KISTI)

요약

정보 갱신 주기가 빨라짐에 따라, 정보검색시스템에 트랜잭션 처리, 조인, 뷰, 트리거와 같은 데이터 관리 기능이 필요하다는 요구가 커지고 있다. 그러므로, DB-IR (데이터베이스 - 정보검색) 통합은 데이터 베이스 분야와 정보검색 분야 모두에 관심 대상이 되고 있다. DB-IR 통합 시스템은 기존의 DBMS 혹은 IRS(정보검색시스템)과는 다른 역할을 가지고 있기 때문에, 새로운 구조의 디자인이 필요하다. 본 논문에서는 효율적이고 안정적인 DB-IR 통합 시스템을 위한 서버 구조에 대해서 논의한다.

Abstract

As information renewal period comes to be quick recently, the need that IRS must have data management facilities such as transaction processing, join, view, trigger is extending. Therefore, DB-IR(Database ? Information Retrieval) Integration is a recent budding area of interest in the research fields of database area and information retrieval area. Since the role of DB-IR integration systems is different from those of DBMS or IRS, It is necessary to be a new design of structure for DB-IR integration system. We have designed a new server structure for efficiency and stability of DB-IR integration system.

I. 서론

정보가 많아지고 동적으로 변하다 보니 DBMS의 구조화된 데이터 관리와 정보검색 시스템의 빠른 검색을 필요로 하는 분야가 늘어나고 있다. 따라서 DB(Database)-IR(Information Retrieval) 통합은 데이터베이스 분야 [1,2,3]와 정보 검색 분야 [4,5,6]에서 새로운 이슈로 떠오르고 있는 연구 분야이다. 이러한 DB-IR 기능 통합에는 몇 가지 접근 방법이 있다.

첫 번째 방법은, 기존의 DBMS와 IRS를 애플리케이션 혹은 미들웨어 레벨에서 연결하는 방법[3]으로 가장 많이 사용되어 온 방식이다. 이러한 접근법에서는 DBMS(Database Management System)는 문서를 관리하고 IRS(Information Retrieval System)는 색인을 생성하고 사용자 질의를 처리하는 역할을 담당하게 된다. 미들웨어 구조가 실제 시스템에서 많이 사용되고 있기는 하지만 DBMS의 문서 내용과 IRS의 색인 정보간 동기화가 매우 어려운 문서-색인 갭이라는 문제가 발생하며, IRS와 DBMS의 관점의 차이로 인하여 두 시스템의 모든 기능을 최대로 이용하기 어렵다.

두 번째 접근법은 DBMS에 IR기능을 느슨하게 결합하여 확장한 것이다. 오라클의 데이터데이터블, DB2의 카드리지와 인포믹스의 extender와 같은 것들로, 이러한 접근법은 쉽게 구

현이 가능하나, 고성능이 필요한 대용량 데이터베이스에서 새로운 데이터 타입과 연산을 구현하는데 어려움이 많아 선호되지 않는다.[7]

세 번째 접근방법은 DBMS와 IR을 밀접합하는 방법이다 [7]. DBMS의 core 엔진에 IR 기능을통합하는 방법과 (Odysseus[7]가 대표적이다.) IRS의 core 엔진에 몇 가지 DBMS 기능을 통합하는 방법이 있다. 후자를 우리는 IRMS (Information Retrieval Management System)이라고 부른다[8]. 이 논문에서는 IRMS를 위한 서버 구조를 설계할 것이다.

마지막 접근법은 DB-IR을 처음부터 새롭게 설계하는 것이다. 구조적 데이터 독립성, 스코어생성, 유연하고 강력한 질의 언어 등 여러가지 장점을 갖는 새로운 DB-IR 통합 시스템이 연구 중이다[2,3].

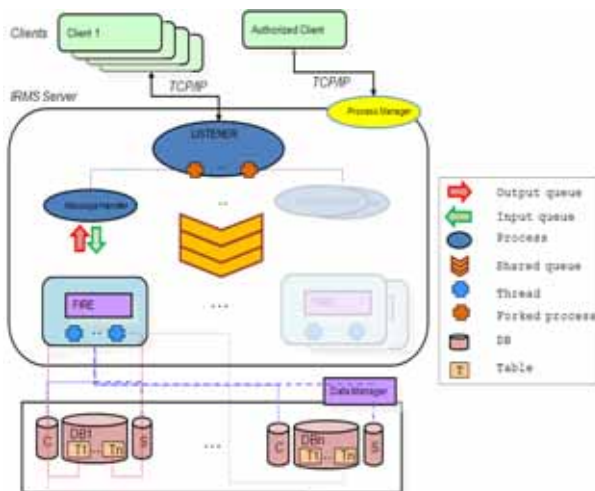
II. DB-IR 시스템 설계의 필요성

IRS는 빠른 검색과 비정형 문서의 랭킹 기능으로 현재 많은 곳에서 사용되고 있다. 하지만, 최근에는 정보 갱신 주기가 빨라짐에 따라 트랜잭션 처리, 조인, 뷰, 트리거 등과 같은 DBMS의 데이터 관리 능력을 요구하는 경우가 늘어나고 있

다. 우리는 웹 응용분야에서 빠른 검색을 주로 필요로 하고 일부 데이터 관리만을 필요로 하는 경우가 많다고 생각한다. 이러한 방식의 DB-IR 통합을 위해 기존과는 다른 서버 설계가 있어야 한다. 기존 IRS는 사용자의 요청이 들어올 때마다 검색 프로세스를 생성하지만 IRMS는 색인과 문서 내용을 하부 저장 구조에 저장하므로 미리 띄워 놓고 사용해야만 한다. 또한 빠른 검색을 위한 역과일 구조를 가짐으로 기존 DBMS의 설계처럼 복잡하고 무거우면 안된다. 이러한 점들이 IRMS에 맞는 새로운 구조가 필요한 이유이다.

III. IRMS 구조 설계

본 연구에서는 안정성과 확장성을 최우선적으로 고려하면서 시스템을 설계하였다. 이에 따라, 통신 부분과 검색 부분을 분리하는 방식으로 하고 이를 모두 관리하는 프로세스 관리기를 따로 두기로 하였다.



▶▶ 그림 1. IRMS 전체 구조

1. IRMS 시스템 전체 구조

1.1 리스너

리스너는 데이터 관리 혹은 검색 요청을 클라이언트로부터 받아 검색 프로세스인 FIRE(Fast Information Retrieval Engine)에게 넘기고 FIRE로부터 결과를 받아 이를 다시 사용자 클라이언트에게 전달하는 역할을 한다. 사용자의 요청이 들어오면 우선 이를 전달할 자식 프로세스인 메시지 핸들러를 생성하고 메시지 핸들러는 FIFO를 이용하여 사용자 요청을 담아 두는 입력 큐와 FIRE로부터 결과를 받을 출력 큐를 만든다. 그 후, 자신의 프로세스 ID를 공용 큐에 담아서 FIRE가 데이터를 주고받을 입출력 큐의 이름을 알 수 있도록 한다. FIRE가 요청을 처리한 후 큐에 담으면, 메시지 핸들러는 결과

를 클라이언트에게 전송하고 입출력 큐를 삭제한 후 메시지 핸들러는 종료된다. 각 사용자의 요청마다 이러한 작업이 수행되는데, 리스너는 공용 큐가 오버플로우되거나 시스템 성능이 떨어지지 않도록 메시지 핸들러의 수를 조절하는 역할도 한다.

1.2 FIRE(Fast Information Retrieval Engine)

FIRE는 IRMS에서 실질적으로 검색과 삽입, 삭제, 수정 등의 데이터 관리를 수행한다. FIRE는 내부적으로 쓰레드를 발생하여 동시에 여러 테이블에 접근하며, 색인 정보와 포스팅 정보를 읽어와서 랭킹을 하거나 문서 관리를 수행한다.(이에 대한 자세한 설명은 [8]에서 자세히 살펴볼 수 있다.) FIRE는 공용 큐에 값이 들어오면 하나를 꺼내어 메시지 핸들러의 ID를 알아내고 해당되는 입력 큐에서 클라이언트의 요청을 받아 처리한 후 결과 큐에 담아준다.

1.3 프로세스 관리기

프로세스 관리기는 DB-IR시스템의 기동, 종료를 수행하며 전체 프로세스들을 안정적인 상태가 되도록 유지 관리한다. 또한 시스템 운영 또는 상태 점검 명령어를 사용자로부터 요청받아 처리하는 역할을 한다. 사용자는 미리 동시에 수행 가능한 요청의 수, 리스너와 FIRE의 실행파일 위치, 색인 및 저장된 문서의 위치, 공용 큐 이름 등을 환경설정 파일에 기록해 두는데, 프로세스 관리기는 환경설정파일로부터 이 값을 가져와 리스너와 검색 프로세스를 띄운다. 또한, 리스너와 FIRE간 사용되는 공용 큐를 생성하며, 수행 불능 상태에 빠진 FIRE나 메시지 핸들러는 제거 한 후 다시 띄우는 방법을 통해 전체 시스템의 안정적인 운영을 지속하게 한다.

2. 프로세스간 통신 설계

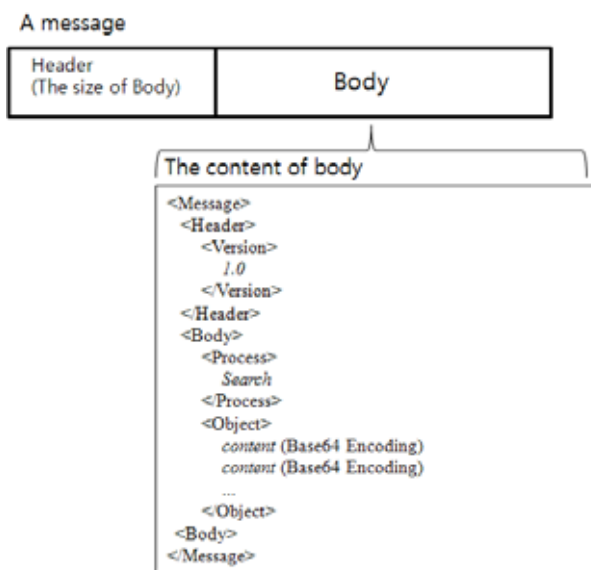
통신을 담당하는 프로세스와 검색을 담당하는 프로세스로 나눔에 따라서, 이들 프로세스간의 정보 전달방식을 선택하는 것도 매우 중요한 문제 중 하나가 되었다. 그리하여 다음과 같은 여러 가지 방법이 논의되었다.

2.1 공용 입력/출력 큐 사용 방식

메시지 핸들러가 사용자의 요청을 받으면, 이 요청을 공용 입력 큐(FIRE 입장에서 입력)에 넣게 된다. 비작업중인 FIRE는 공용 큐로부터 사용자의 요청을 하나씩 받아 처리한 후 결과를 공용 출력 큐에 쌓는다. 마지막으로, 요청을 보낸 메시지 핸들러는 공용 출력 큐로부터 결과를 꺼내어 사용자 클라이언트에게 전송한다.

이러한 방식은 몇가지 문제를 가지고 있다. 첫째, 메시지 핸들러는 지속적으로 공용 출력 큐에 결과가 입력되어있는지 채

크해야만 한다. 이것은 심각한 시스템 성능 감소를 야기시키며 구현을 복잡하게 한다. 두번째 문제는 검색 프로세스 혹은 통신 프로세스 중 하나가 죽거나 데드락 상태처럼 처리 불능 상태가 되면 전체 시스템의 안정성에 영향을 끼치게 된다. 또한, 프로세스들 사이에 전송되는 하나의 메시지는 헤더와 바디로 이루어져 있다. 이것은 입력/출력 메시지가 매우 클수 있기 때문에 두 번에 걸쳐 나누어 전송하기 위함이다. 만약 두 개 이상의 요청이 거의 동시에 서버에 들어오면 헤더와 바디가 나누어져 서로 다른 FIRE와 메시지 핸들러에 전송되어 서버 전체가 블록킹될 수 있다.



▶▶ 그림 2. 메시지 구조

2.2 개별 큐 사용 방식

위의 문제를 해결하기 위하여, 공용 입출력큐를 사용하지 않고 각 메시지 핸들러마다 고유의 ID 값이 붙어있는 입출력 큐를 가지는 방법을 고안하였다. FIRE가 사용자의 요청을 처리하고 결과를 해당 메시지 핸들러의 출력 큐에 넣어주면, 메시지 핸들러는 이것을 클라이언트에게 전송한다. 자세한 처리절차는 1.1절에서 이미 설명하였다.

이렇게 하여 공용 입출력 큐 사용에서의 문제점을 해결하였지만 또다른 문제가 생겼다. 두 번째 문제는 입출력 큐로 시스템 메시지 큐를 사용할 수 없다는 것이었다. 시스템 메시지 큐는 시스템의 한정된 자원이기 때문에, 정해진 수 이상으로 만들 수 없으며, 일부 프로세스가 Deadlock에 빠지거나 죽으면 할당되어진 메시지 큐를 즉각 없애야지 전체 시스템이 다운되는 일을 막을 수 있다. 그러므로, 실제 구현에서는 시스템 메시지 큐 대신에 FIFO를 사용하여 입출력 큐를 만들었다.

FIFO의 특성때문에 몇 가지 문제가 또 나타났는데, 몇 개의 프로세스가 동시에 FIFO 버퍼 크기를 넘어서는 크기의 데이터를 보내면 FIFO의 데이터 원자성을 보장할 수 없다. 만약 많은 수의 사용자가 동시에 요청을 보내면, 많은 수의 메시지 핸들러가 띄워져 공용 입력 큐에 자신의 ID를 넣는다면, 공용 큐에 저장된 ID들은 FIFO 원자성 속성 때문에 조각날 수 있다. 그러므로 우리는 메시지 핸들러 ID의 크기 X 메시지 핸들러의 수가 FIFO 버퍼의 크기를 벗어나지 않도록 메시지 핸들러의 수를 조절하는 메카니즘을 설계했다. 또한 FIFO의 특성상, 입력 FIFO와 출력 FIFO사이에 상대가 없으면 메시지 핸들러와 FIFO는 무한 블록킹 상태에 빠진다. 이점은 DOS(Denial of Service) 공격을 유발할 수 있으므로, 우리는 블록킹 상태를 빠져나갈 수 있도록 프로세스에 타임아웃 시간을 설정하였다.

IV. 결론

DB-IR 시스템은 기존 정보검색 시스템의 빠른 검색과 DBMS의 데이터 관리 기능의 장점을 합친 시스템이다. 우리는 DB-IR 시스템 통합의 여러가지 방법 중 IRS를 기본으로 하고 DBMS의 관리 기능을 밀결합한 IRMS 방식을 연구하였다. 기존 DBMS나 IRS와는 기능적인 차이가 있기때문에 IRMS를 위한 새로운 서버 설계가 필요하다. 본 논문에서는 IRMS 시스템에 적합한 서버 구조와 프로세스간 통신 방법을 설명하였다. 향후 연구계획으로는 이러한 서버 구조의 효율성을 DBMS, IRS와 비교하여 객관적인 자료를 만들 생각이며, 더 나아가 시스템의 효율성을 높이기 위하여 사용자 요청의 수, 질의 처리 시간 혹은 시스템 부하에 따라 동적으로 검색 프로세스의 수를 조절하는 기능과 물리적으로 두 개 이상의 서버를 이용하여 하나의 시스템이 다운되었을 때 자동으로 스워칭하는 기능을 추가할 예정이다.

■ 참고 문헌 ■

[1] A. P. de Vries , A. N. Wilschut "On the integration of IR and Databases", 8th International IFIP 2.6Working Conference on Database Semantics, 제 138권, pp.16-31, 1999.
 [2] S. Amer-Yahia, P. Case, T. Roelleke, J. Shanmugasundaram, G. Weikum., Report on the DB/IR panel at SIGMOD, Vol. 34, No. 4, pp. 71-74, December 2005.
 [3] S. Chaudhuri, R. Ramakrishnan, G. Weikum, "Integrating DB and IR technologies: What is the sound of one hand clapping?", In Proceedings of the Second Biennial

- Conference on Innovative Data Systems Research (CIDR 05), pp 1-12, 2005.
- [4] R. Baeza-Yates, Y. S. Maarek, T. Roelleke, A. P.de Vries, Third edition of the “XML and information retrieval” workshop. SIGIR 04. Vol. 38, No 2, pp-24-30, 2004.
- [5] M. P. Consens, R. A. Baeza-Yates, “Database and information retrieval techniques for XML”, In Advances in Computer Science - ASIAN 2005, Data Management on the Web, 10th Asian Computing Science Conference Proceedings, vol. 38. No. 18, pp. 22-27, 2005.
- [6] H. Bast , I. Weber, “The Complete Search Engine: Interactive, efficient, and towards IR&DB integration”, In CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, pp. 88-95, 2007.
- [7] K.-Y. Whang, M.-J. Lee, J.-G. Lee, M.-S. Kim, W.-S. Han, “Odysseus: A high-performance ORDBMS tightly coupled with IR features”, In ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05), pp. 1104-1005, 2005.
- [8] GIIS : “KRISTAL-IRMS: Information Retrieval & Management System”, <http://www.kristalinfo.com/>