효율적인 정보 검색 시스템 구축을 위한 새로운 프로세스 구조

고 형 대[†] · 유 재 수[††] · 김 병 기[†††]

요 약

기존 정보 검색 시스템의 프로세스 구조는 사용자를 위한 클라이언트 프로세스와 정보 검색 시스템을 위한 서버 프로세스가 일대일로 대응하는 간단한 구조이다. 그러나 사용자 마다 사용자 인터페이스, 검색 모델, 자동 색인 및 저장 시스템으로 구성되어 있는 큰 서버 프로세스가 하나씩 할당되기 때문에 많은 수의 사용자가 정보 검색 시스템을 사용할 때 시스템 오버헤드가 커져 시스템을 사용할 수 없는 문제를 발생시킨다. 본 논문에서는 이와같은 기존 정보 검색 시스템의 프로세스 구조가 갖는 문제점을 해결한 효율적인 정보 검색 시스템 구축을 위한 새로운 프로세스 구조를 제안한다. 제안된 프로세스 구조는 정보 검색 시스템의 전체적인 동작 성능 및 컴퓨터 시스템 자원의 효율적인 활용에 기여할 수 있게 된다. 제안된 프로세스의 구축은 프로세스 오버헤드를 최소화하여 많은 수의 사용자 환경을 효율적으로 지원할 수 있는 다중 스레드와 전체 시스템의 성능을 향상시키기 위해 제공되는 트랜잭션 처리 모니터에 근거한다.

A New Process Structure for Constructing Efficient Information Retrieval Systems

Hyung Dae Koh[†] · Jae Soo Yoo[††] · Byung Ki Kim[†††]

ABSTRACT

Many information retrieval systems have a simple process structure that a client process for a user is mapped to a server process for information retrieval. That is, when using information retrieval systems, each user is allocated a big process that consists of user interfaces, retrieval systems, automatic indexing systems and storage systems. Therefore when many users use the information retrieval systems, it might be difficult to use the information retrieval systems. This is because the system overhead is increased as enormously much as users cannot use them. In this paper, we propose a new process structure for constructing efficient information retrieval systems that solves the problem resulting from the process structure. The proposed process structure contributes to the whole operational performance improvement of information retrieval systems and the efficient use of computer system resources. It is constructed based on a multi-threading scheme and a transaction processing monitor.

† 종신회원 : 목포대학교 전산통계학과
†† 정 회 원 : 충북대학교 공과대학 전기전자공학부
††† 종신회원 : 전남대학교 전산학과
논문접수 : 1996년 7월 18일, 심사완료 : 1996년 11월 20일

1. 서 론

1950년대 초에 제 1세대 컴퓨터가 출연한 이후로 컴퓨터를 이용하여 대용량의 문서를 효율적으로 검색할 수 있는 정보 검색 시스템에 관한 많은 연구가 이루어져 왔다[3]. 정보 검색 시스템의 사용은 원하는 정보에 대한 접근을 용이하게 함으로써 여러 분야에 있어서 정보 수집에 대한 시간과 노력을 단축시키게 된다. 특히 관리할 정보의 양이 기하 급수적으로 증가하고 있는 정보화 시대인 오늘날에는 효율적인 정보 검색 시스템에 대한 요구는 더욱 절실하다.

정보 검색 시스템은 몇몇의 기본적인 연산을 제공해야 한다. 즉 데이타베이스에 문서들을 삽입 하고, 문서를 변경하며 필요에 따라 삭제할 수 있는 수단을 제공해야 한다. 또한 원하는 문서를 찾을 수 있는 방법과 이 문서들을 사용자에게 표시해 줄 수 있는 수단을 제공해야 한다. 이러한 정보 검색 시스템은 몇 가지 요소들로 나뉘어질 수 있다. 사용자의 요구를 받고 결과를 제시하는 사용자 인터페이스 부분, 사용자의 요구를 해석하여 적절한 검색 방법등을 결정하며, 필요한 동작을 수행하는 정보 검색 엔진 부분, 대용량의 문서를 저장 관리하는 저장 시스템 등으로 구분할 수 있다.

이에 따라 최근 정보 검색의 성능 및 유용성을 향상시키기 위해 정보 검색 시스템의 각 구성 부분에 대한 연구가 계속적으로 활발히 진행되고 있다. 그래픽 사용자 인터페이스와 같은 편리하고 인식도가 높은 수단을 개발하고 멀티미디어를 이용한 검색 결과 제시 등을 고려하고 있으며, 보다 정확도가 높은 정보 검색을 위해 문서들의 저장 방법 및 검색 모델을 제안하고, 정보 검색 성능 향상을 위한 검색 방법 및 저장 시스템 구조, 인덱싱 방법 등을 제시하는 연구들이 계속되고 있다[3, 4, 5, 8].

그러나, 이들 기존의 정보 검색 관련 연구는 정보 검색 시스템의 각 구성 요소 단위로 정보 검색 효율의 향상을 위한 방안들을 제시하는 것으로서, 전체적인 정보 검색 시스템의 실질적인 구현 구조와 실제의 컴퓨터 시스템 환경에서의 요구사항에는 미흡한 상태이다. 다수의 사용자가 온라인으로 이용하게 되는 정보 검색 시스템은 주어진 컴퓨팅 환경에 적절한 프로세스 구조가 구현되어야 정보 검색 시스템의 전체적인 동작 성능 및 컴퓨터 시스템 자원의 효율적인 활용에 기여할 수 있게 된다.

본 논문에서는 트랜잭션처리 모니터[2]와 다중 스레드기법[1]을 이용한 정보 검색 시스템의 새로운 프로세스구조를 제안한다. 이를 위해 먼저 정보 검색 시스템과 유사한 특성을 갖는 온라인 트랜잭션 처리 시스템에 이용하는 트랜잭션 처리 모니터와 프로세스 관리에 따른 부하를 줄이는 다중 스레드에 대해 살펴본다. 제안된 정보 검색 시스템의 프로세스 구조인 정보 검색 처리 모니터를 구성하는 모듈들에 대한 설계 방안을 제시한다. 또한 제안된 프로세스 구조의 타당성을 제시하기 위해 벤치마크를 통해 정보 검색 처리 모니터를 사용하지 않는 시스템과 정보 검색 처리 모니터를 사용하는 시스템과의 성능평가를 수행한다.

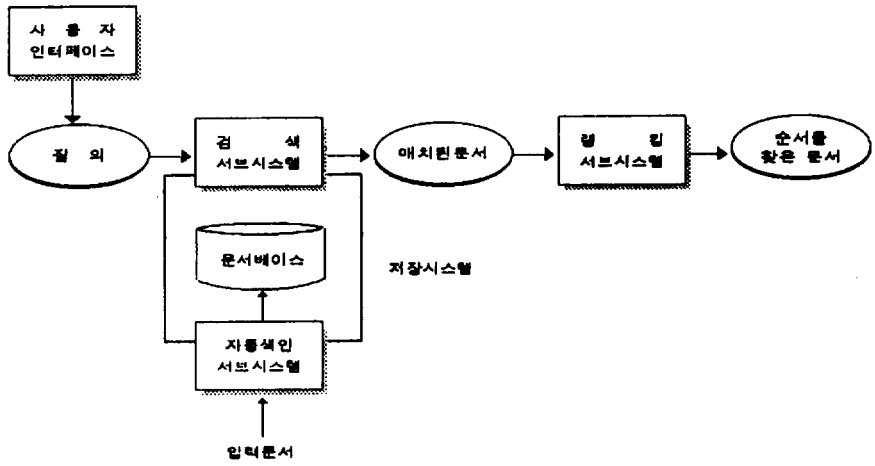
본 논문은 다음과 같이 구성된다. 2장에서는 기존 정보 검색 시스템의 구조 및 운용 환경을 기술한다. 3장에서는 다중 스레드의 기본 개념과 특성을 분석하고 트랜잭션처리 모니터의 개념 및 기능에 대해서 살펴본다. 4장에서는 효율적인 온라인 정보 검색 시스템을 위한 프로세스 구조를 제안하고 정보 검색 처리 모니터를 구성하는 모듈들에 대한 설계 방안을 제시한다. 5장에서는 제안된 프로세스 구조의 핵심 요소인 정보 검색 처리 모니터의 성능 평가를 수행한다. 마지막으로 본 논문의 결론을 6장에서 기술한다.

2. 기존 정보 검색 시스템 구조

본 장에서는 정보 검색 시스템의 동작 구조를 판단하고 기능별 적절한 프로세스 구조를 결정하기 위한 바탕을 제공하기 위해 정보 검색 시스템의 구체적인 구조 및 기존 정보 검색 시스템 사용을 위한 운영 환경을 살펴본다. 정보 검색 시스템은 사용자 인터페이스, 자동 색인 기법, 검색 모델, 저장 시스템으로 구성된다. 한편, 대용량의 정보 검색 시스템의 운영 환경, 즉 아키텍춰는 중형/대형 호스트 기반 시스템, PC LAN을 기반으로 하는 시스템, 클라이언트/서버 모델에 기반한 방법 등으로 구별될 수 있다.

2.1 정보 검색 시스템의 구성

먼저 정보 검색 시스템의 운영 구조를 살펴보기 전



(그림 2.1) 정보 검색 시스템 구성도
(Fig. 2.1) Architecture of information retrieval systems

에 일반적인 정보 검색 시스템의 소프트웨어 구조를 살펴본다. 그 이유는 일반 정보 검색 소프트웨어 구성이 시스템 아키텍춰와 밀접한 연관을 가지기 때문이다. (그림 2.1)은 일반 정보 검색 소프트웨어의 구성을 보여준다. (그림 2.1)에서 보여주는 바와같이 정보 검색 시스템은 사용자 인터페이스, 자동 색인 기법, 검색 모델, 저장 시스템으로 구성된다.

사용자 인터페이스는 사용자가 원하는 정보를 검색할 수 있도록 정보 검색 시스템에 질의를 하기 위해 사용자에게 제시되는 인터페이스로서, 메뉴 인터페이스, 그래픽 인터페이스, 자연어처리 인터페이스 등이 있다[12]. 자동 색인 서브 시스템은 입력되는 문서들을 분석하고 그 문서에서 키워드를 추출하여 사용자 질의를 만족하는 문서를 검색할 때, 질의에 나타난 단어와 추출된 단어들과의 유사성을 계산하여 사용자가 만족하는 문서를 검색할 수 있도록 한다. 정보 검색 서브 시스템은 특정 검색 모델을 지원한다. 질의를 만족하는 문서들을 사용자에게 적합한 형태로 제공하는 검색 모델에 관한 많은 연구들이 진행되어 왔다. 널리 알려진 검색 모델로는 불리안 모델[11], 퍼지 집합 모델[10], 벡터 공간 모델[5], 확장된 불리언 모델[13], 확률 모델[9], 인공 지능 모델[12] 등이 있다. 이와 더불어 시소러스를 이용한 검색된 문서들을 원하는 정도에 따라 정렬(ranking)해 주는 연구가 있다[8].

저장시스템은 실제적으로 데이타가 저장되어 있는 (그림 2.1)의 문서 베이스 부분으로 대부분의 상용 정보 검색 시스템에서는 역 색인 방법을 기반으로 구성되어 있다. 따라서, 저장 시스템을 크게 두부분으로 나누면 인덱스 부분과 텍스트 화일 부분으로 나눌수 있다[3, 12].

2.2 정보 검색 시스템 운영 환경

2.2.1 호스트 기반 정보 검색 시스템 구조

기존의 대용량 정보 검색 시스템은 대부분 중형/대형 호스트 기반 아키텍춰이다. 2.1절에서 언급한 모든 검색 S/W들은 호스트에 존재하고 사용자는 단지 문자 기반 dummy 터미날들을 사용해서 원하는 정보를 검색하는 아키텍춰이다.

한편, LAN 또는 WAN을 통하여 여러 시스템들이 호스트 시스템과 연결되어 있는 경우에도 결국 사용형태는 정보 검색 시스템이 존재하는 호스트로 원격 로그인(remote login)을 통해 접속한 뒤, 직접 연결된 dummy 터미날의 사용자와 마찬가지로 데이타베이스를 검색하는 형태의 검색이 이루어지게 된다. 이때는 정보 검색 시스템이 존재하는 호스트 컴퓨터가 터미날 관리 및 통신 제어의 부담을 떠맡게 되어 다수의 사용자가 사용하게 되면 전체 시스템의 성능이 급격히 떨어지게 된다. 또한, 현재 정보 검색 시스템은 사용자당 하나의 프로세스가 형성되어 서비스를 담

당하게 되어 있어서, 사용자 수 만큼의 프로세스 관리가 대단한 시스템의 부하로 작용하게 된다. 따라서 이와같은 구조상에서 정보를 관리하면, 정보가 하나의 호스트에 집중되어 데이타베이스가 대용량이 되어 이에 따르는 인덱스 정보의 오버헤드 및 검색 효율의 저하는 심각한 상태에 이르게 된다.

2.2.2 PC/LAN 기반 정보 검색 시스템 구조

PC/LAN 기반의 네트워크 정보 검색 시스템 구조는 PC의 하드웨어적인 처리 속도와 저장 용량이 급속하게 발전하고 있고, 경제적인 측면에서도 저렴하기 때문에 사용이 급속하게 증가하고 있는 추세이다. PC/LAN 기반의 정보 검색 시스템 구조의 급속한 증가는 서버와 클라이언트의 접속을 용이하게 해주는 브리지(bridge)와 게이트웨이(gateway)와 같은 하드웨어 기술이 인텔리전트한 기능을 갖춘 허브(hub)를 출현시켰기 때문이다. PC/LAN 기반 구조는 실제로 데이타가 저장되어 있는 데이타베이스가 노드들에 분산되어 있어 호스트 기반 정보 검색 시스템 구조의 시스템 오버헤드를 줄일 수 있다. 그러나 대용량의 정보를 저장하고 검색할 때 분산된 노드들로부터 원하는 정보를 얻기 위해서는 많은 검색시간이 소요된다는 단점이 있다.

3. 정보 검색 시스템의 프로세스 구조 구축을 위한 주요 개념들

3.1 다중 스레드

각 단말기마다 하나의 프로세스를 할당하는 단일 스레드의 단점은 하나의 지역 시스템에 연결된 모든 단말기를 관리하는 하나의 프로세스를 만들어서 해결할 수 있다[1]. 즉, 각 단말기는 각기 독자적인 제어 스레드를 가지고 그 프로세스내의 다른 스레드와 함께 주소 공간을 공유하는 것이다. 이러한 독자적인 스레드는 응용 프로그램에 의해 구현될 수도 있지만 대개의 경우 트랜잭션 처리 모니터나 운영체제에 의해 구현된다. 한 프로세스의 각 스레드는 지역 변수를 위하여 공유되지 않는 데이타 공간을 가져야만 한다.

운영체제로 구현할 경우에 이 공간은 private 스택과 레지스터 저장 공간으로 구성되며, 트랜잭션 처리 모니터로 구현할 경우 각 스레드마다 지역 메모리로 구성되어야 한다. 다중 스레드가 제공되면 시스템은 프로세스를 스위칭하지 않고 스레드를 스위칭함으로써 여러 단말기들을 동시에 관리할 수 있다. 스레드 스위칭은 주소 공간 사상 테이블, 캐쉬 메모리 등의 대치를 유발하지 않으므로 프로세스 스위칭 보다 훨씬 효율적이다.

다중 스레드 프로세스의 단점은 다음과 같다. 첫째, 모든 스레드가 프로세스 메모리를 접근하므로 단일 스레드에 비하면 보호가 약하다. 이러한 문제는 각 스레드가 지역 데이타에 대하여 private 스택을 가질 수 있는 기계 구조를 이용하거나 프로그램이 메모리 접근을 함부로 하지 않는 강력한 프로그래밍 언어를 이용하여 해결할 수 있다. 둘째, 스케줄링을 프로세스와 스레드 두 단계에 걸쳐서 해야만 한다. 따라서 다른 프로세스에 있는 스레드간의 우선순위 조절이 어려워진다. 하지만 이러한 단점에도 불구하고 다중 스레드의 장점으로 인하여 대부분의 트랜잭션 처리 모니터들은 다중 스레드를 이용한다.

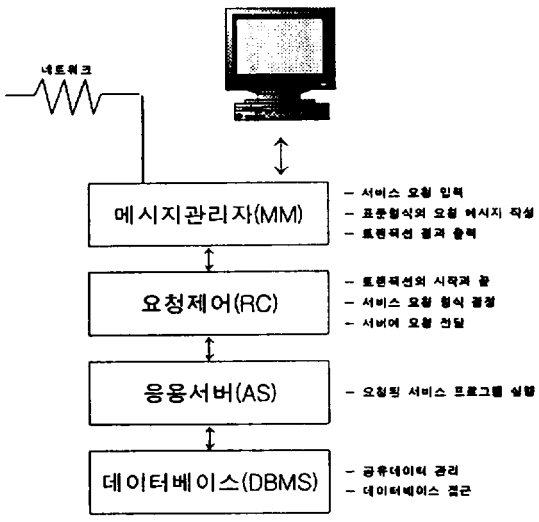
3.2 트랜잭션 처리 모니터

다수의 사용자가 온라인으로 이용하게 되는 정보 검색 시스템은 주어진 컴퓨터 환경에 적절한 프로세스 구조를 갖추어야 한다. 이와 같은 적절한 프로세스 구조를 제안하기 위하여, 프로세스를 효율적으로 관리하고, 네트워크상의 많은 프로세서 및 단말기 등과 같은 많은 자원들을 용이하게 제어할 수 있는 트랜잭션 처리 모니터를 살펴본다.

대부분의 트랜잭션 처리 시스템은 온라인 트랜잭션 응용의 특성을 가지며, 업무의 분업화 및 다양화로 자연스럽게 분산되어 있는 여러 노드 간에 분산 트랜잭션을 처리할 수 있도록 요구된다. 트랜잭션 처리 기술에 있어서 대부분의 관심은 데이타베이스 관점에 있지만, 데이타베이스가 트랜잭션 처리에 매우 중요하더라도 DBMS의 성능 향상만으로는 부족하다. 이것은 전체 시스템의 성능과 용이성은 운영체제와 통신 기법에 의해 영향을 받기 때문이다.

트랜잭션 모니터는 프로세스를 관리해야 하고, 네트워크에 대한 상세한 지식없이 편리하게 통신할 수 있는 기능을 제공해야 하며 네트워크상의 많은 프로세서와 단말기들을 용이하게 제어할 수 있도록 해야한다. 그러므로 트랜잭션 처리 모니터는 각각의 독립된 요

소들을 관리하면서 전체시스템의 통합적인 관리 기능도 가지고 있어야 한다. (그림 3.1)을 전형적인 트랜잭션 처리 모니터 모형으로서 대부분의 트랜잭션 처리 모니터가 이러한 모형을 기반으로 만들어진다[2].



(그림 3.1) 트랜잭션 처리 모니터 모델
(Fig. 3.1) A model for TP monitors

트랜잭션 처리 모니터의 기능은 다음과 같다.

- 메시지 관리자(Message Manager)
  - MM은 여러 종류의 단말기로부터 입력과 출력을 처리해 주어야 한다. MM의 주요기능들로는 표준형식의 요청 메시지 작성, 폼 관리, 입력 값의 유효성 검사, 결과 출력 및 사용자 위주의 보안기능을 수행하는 것 등이다.
- 요청 제어(Request Control)
  - MM에서 처리된 요청 메시지는 RC에 보내어진다. RC에서는 요청된 일을 처리할 수 있는 AS를 찾아 요청 메시지를 보낸다. RC는 요청 메시지의 헤더에 저장된 일을 해당서버에게 입력 변수를 보내어 수행시킨다. 또 트랜잭션의 시작과 끝을 관리한다.
- 응용서버(Application Server)
  - 각 AS는 DB를 참조하는 여러 개의 응용프로그램들로 구성된다. AS와 RC를 연결시키는 것뿐만 아니라 트랜잭션 처리 모니터는 AS에게 단말기가 직접 연결되도록 하는 기능을 제공한다. 이러한 점은 프로세스 관리와 통신 면에서 운영체제의 단점을 보완한다.

4. 정보 검색 시스템을 위한 효율적인 프로세스 구조

4.1 주요 기능 설계

기존 정보 검색 시스템의 프로세스 구조는 사용자를 위한 클라이언트 프로세스와 정보 검색 시스템을 위한 서버 프로세스가 일대일로 대응하는 간단한 구조이다. 하지만 사용자마다 사용자 인터페이스, 검색 모델, 자동 색인 및 저장 시스템으로 구성되어 있는 큰 서버 프로세스가 하나씩 할당되기 때문에 UNIX 시스템 상에 많은 수의 사용자가 정보 검색 시스템을 사용할 때 시스템 오버헤드가 커져 시스템을 사용할 수 없는 문제를 발생시킨다. 본 논문에서는 이와 같은 문제점을 해결하기 위해 클라이언트/서버 기반 정보 검색 처리 모니터를 설계하고 구현한다.

클라이언트/서버 기반 구조는 데이터베이스와 인덱싱, 랭킹(ranking) 기능등을 서버에서 수행하고, 사용자 인터페이스 부분은 각각의 클라이언트에 위치한다. 클라이언트/서버 기반 구조의 장점은 클라이언트들은 데이타를 어떻게 찾고 또 각각의 프로그램들이 어떻게 실행되는지를 알 필요가 없이 데이타베이스와 모든 검색 S/W들이 존재하는 것처럼 모든 정보 검색 서비스를 동일한 하나의 이미지로서 제공 받을 수 있다. 따라서 정보 검색 시스템에서도 분산된 클라이언트/서버 기반 정보 검색 시스템 구조를 사용함으로써 쉬운 확장성, 간단한 재구성 그리고 다양한 상호 연결을 지원할 수 있다.

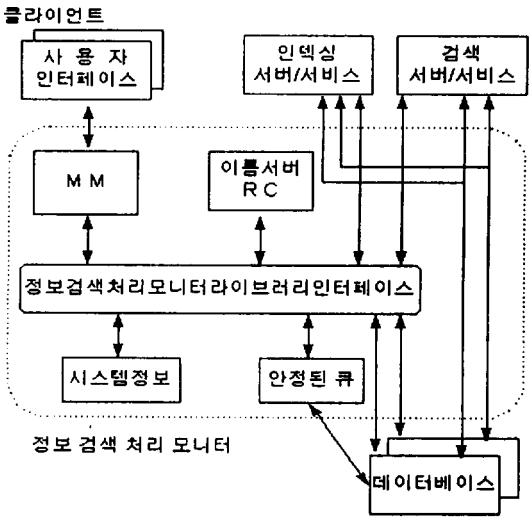
본 논문에서 제안하는 정보 검색 처리 모니터는 클라이언트 프로세스와 서버 프로세스간의 자료 교환, 제한된 자원인 데이타베이스의 효율적인 관리, 처리 요구의 라우팅, 그리고 처리 요구를 서버들에게 균형있게 분산시키는 일을 담당하는 새로운 프로세스 구조로 설계하여 높은 성능을 보장할 수 있도록 하였다. 또한 시스템의 오버헤드를 줄이기 위해 기존의 하나의 프로세스에 포함되어 있던 사용자 인터페이스, 검색 시스템, 자동 색인 및 저장 시스템을 기능에 따라 클라이언트 프로세스와 서버 프로세스로 나누어 그

역할을 수행한다. 즉 사용자 인터페이스 부분을 클라이언트 프로세스로 할당하고, 검색 연산을 위한 서버 프로세스, 자동색인을 위한 서버 프로세스로 분할한다. 또한, 이 클라이언트 프로세스와 서버 프로세스들을 다중 스레드로 구현하여 시스템 성능을 높인다.

본 논문에서 설계하는 정보 검색 시스템 프로세스 구조인 온라인 정보 검색 처리 모니터의 주요기능은 우선적으로 클라이언트들과 서버들 간의 안정된 요청을 주고 받는 것이다. 이를 위하여 정보 검색 처리 모니터는 클라이언트들에 의해 요구된 요청을 잃어버리지 않기 위해 데이타베이스 트랜잭션의 ACID 특성을 유지하는 안정된 큐(durable queue)를 제공한다. 또한 요청 제어 및 부하 균형을 위하여 관련 정보를 관리하여 시스템을 동적으로 재구성할 수 있는 기능을 제공한다. 마지막으로 온라인 정보 검색연산들의 스케줄링을 위하여 요청 큐에서 요청 데드라인에 따른 요청 메시지의 우선 순위를 제공할 수 있도록 한다.

4.2 시스템 구성

온라인 정보 검색 시스템을 위한 정보 검색 처리 모니터는 검색 요청 메시지 관리 모듈(MM), 요청 서비스 명을 기반으로 대응하는 응용 서버로 라우팅하는 이름 서비스기능(Naming Service), 정보 검색 서버



(그림 4.1) 시스템 구성도
(Fig. 4.1) System architecture

간의 부하 분산과 요청 흐름을 제어하기 위한 요청 제어 모듈(RC), 응용 서버 관리 모듈, 그리고 요청 메시지를 효율적으로 전달하기 위한 메시지 큐 관리 모듈과 마지막으로 시스템 정보관리를 위한 모듈로 구분할 수 있다. 정보 검색 시스템을 위한 정보 검색 처리 모니터의 대략적인 구성도는 (그림 4.1)과 같다.

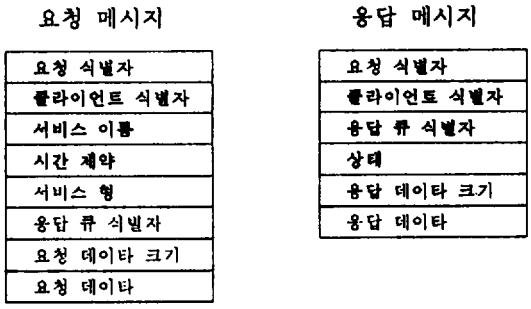
그림에서의 검색 서버와 자동색인 서버는 UNIX에서 다중 스레드를 지원하는지의 여부에 따라 생성되는 서버의 수가 조정되어야 한다. UNIX에서 다중 스레드를 지원할 경우, 각 서버는 하나의 다중 스레드 프로세스로 구현된다. UNIX가 다중 스레드를 지원하지 않을 경우, 정보 검색 시스템의 대부분의 연산이 I/O bound이기 때문에 이러한 서버들을 다중 스레드로 구현시 이점이 사라지므로 단지 하나의 스레드를 갖는 다중 서버로 구현되어야 한다.

4.3 주요 모듈 설계 및 구현 일반

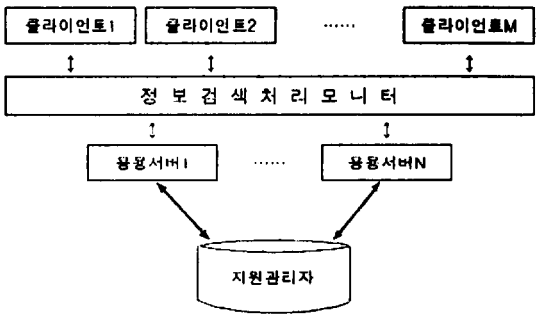
4.3.1 메시지 관리 모듈(Message Manager)

정보 검색 처리 모니터는 정보 검색 요청과 응답을 올바르게 전달하기 위하여 정규화된 메시지 형식을 필요로 한다. 본 논문에서는 요청 메시지와 응답 메시지를 위하여 (그림 4.2)와 같은 두 가지 메시지 형식을 사용한다.

클라이언트의 요청을 해당 서버에게 전달하기 위한 메시지로써 요청 메시지를 사용한다. 각 요청 메시지는 요청 식별자(request id)와 클라이언트 식별자(client id)를 통하여 유일하게 구분되며 서비스 이름(service name) 필드를 사용하여 해당요청 서비스를 인식할 수 있도록 하며 시간 제약(time constraint)필드를 사용하여 실 시간 요청을 위한 제한 시간을 요청할 수 있도록 한다. 그리고 실질적인 서비스 정보를 요청 데이타(request data)필드를 통하여 전달한다. 특히, 응답 큐 식별자(response queue id)는 서버로 부터 응답을 받는 큐 식별자이다. 그 외 서비스 형(service type)과 요청 데이타 크기(request data size)는 요청 관리를 효율적으로 하기 위한 필드이다. 또한 응답 메시지는 요청을 처리한 후 서버에 의해 클라이언트로 전달되는 메시지로 상태(status) 필드는 요청 트랜잭션이 정상적으로 commit 또는 abort되었는지를 지시하는 필드이다. 그리고 나머지 필드의 사용은 요청 메시지의 경우와 동일하다.



(그림 4.2) 요청/응답 메시지 구조
(Fig. 4.2) Request/reply message



(그림 4.3) M : N 클라이언트/서버구조
(Fig. 4.3) M : N client/server

4.3.2 통신 관리 모듈

본 논문에서 중심이 되는 통신 관리 요소는 클라이언트로 부터의 정보 검색 요청이 정보 검색 처리 모니터를 통해 응용 서버에게 전달되고, 처리된 결과가 다시 해당 클라이언트로 보내지는 프로세스간 통신으로 효율적인 프로세스 관리를 위하여 다 대 다 클라이언트/서버 모형을 기반으로 한다.

제안된 정보 검색 시스템의 프로세스 구조 다음과 사항들을 고려하여 설계되었다.

- 짧은 정보 검색 연산들을 효율적으로 처리해야 한다.
- 같은 종류의 짧은 검색 연산들이 요청될 때마다 이를 처리해 주기 위한 새로운 프로세스를 매번 생성시키는 오버헤드없이 프로세스들이 검색 연산 요청을 기다리면서 계속 수행되게 해 준다.
- 같은 응용을 사용하는 사용자들을 대부분 같은 종류의 검색 연산을 요청하므로 많은 사용자가 동시에 사용할 때 발생할 수 있는 오버헤드를 줄일 수 있는 방법을 제공해야 한다.
- 중요한 검색 연산들은 먼저 처리해 주어야 한다.

본 논문에서 설계한 정보 검색 처리 모니터에서는 위의 사항들을 고려하여 정보 검색 시스템에 적합한 클라이언트/서버 모형을 기반으로 (그림 4.3)과 같이 구성된다.

(그림 4.3)과 같은 M:N 클라이언트/서버 모형은 모듈성, 확장성, 분산의 용이성들과 같은 장점을 갖는다. 즉, 검색 연산 요청을 위한 프로세스가 시작될 때마다 이에 대응하는 응용서버가 일대일로 시작되도록 하는 일반적인 시스템 응용이나, 정보 검색시스템을 사용하지 않는 종래의 정보 검색 시스템의 클라이언트와 서버간의 관계를 개선한 형태이다. 고정된 수의 응용서버를 항상 수행 상태로 대기하도록 하고, 정보 검색 처리 모니터가 클라이언트로 검색 연산 처리 요청을 서버에게 배분하는 교통정리 기능을 제공한다.

이와 같은 방법으로 구성된 개선된 클라이언트/ 서버 모형은 다음과 같은 장점을 가진다.

- 높은 성능 : 서버는 클라이언트의 서비스 요청을 기다리면서 항상 수행되고 있기 때문에 서버의 start up 오버헤드가 없으며 처리율(throughput)이 증가된다.
- 위치 은폐성 : 클라이언트는 서버의 주소를 몰라도 서비스 명(name)으로 name server를 통해 정보를 얻을 수 있다.
- 유연성, 규모 조절 능력 : 응용에 맞게 서버의 갯수를 조절할 수 있고 여러 새로운 기능의 서버를 추가할 수 있으므로 시스템의 확장, 축소가 가능하다.

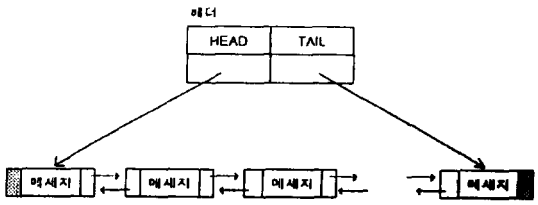
4.3.3 메시지 큐 관리 모듈

응용 서버의 부하가 일시적으로 높을 때 새로운 요청에 대해 새로운 서버 프로세스를 생성시키는 것 보다 응용 서버에 대응하는 메시지 큐를 두어 큐에 요청 메시지를 저장하는 것이 경제적이다. 이러한 메시지 큐는 검색 연산 요청 방식에 따라 두 가지 즉, 동기 검색을 위한 휘발성 메시지 큐와 비 동기 정보 검색을 위한 안정된 메시지 큐로 분류된다. 먼저 휘발성 메시지 큐는 메인 메모리에 유지되며 따라서 빠른 응답 시간을 요하는 실시간 정보 검색 응용에 적합하

다. 이에 반해 안정된 메시지 큐는 비동기 정보 검색을 지원한다. 비동기 정보 검색을 긴 지연 시간을 포함할 수 있으며 이때 모니터로 요청한 요청을 잃어버리지 않기 위해 요청 메시지는 안정된 기억 장치에 유지되어야 하며 일반적으로 정보 검색 처리 모니터가 접근 가능한 정보 검색용 데이타베이스를 통해 구현된다.

큐잉(queueing)이 정보 검색 처리 모니터에 사용되는 이유는 다음과 같이 요약해 볼 수 있다.

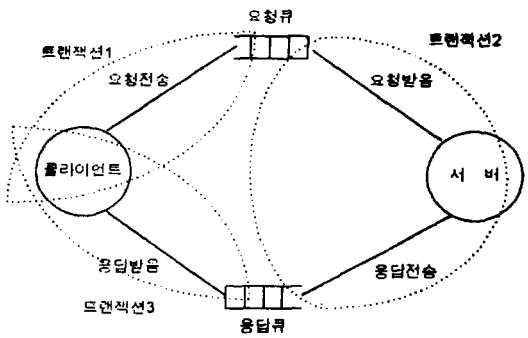
- 부하조절(load control): 만약 응용 서버의 부하가 일시적으로 높을 때 새로운 요구에 대해 새로운 프로세스를 생성시키기는 것보다 응용 서버에 대응하는 메시지 큐를 두어 큐에 요청을 두는 것이 것이 경제적이다.
- 최종결과 조절(end-user control): 사용자가 하나의 트랙잭션 요구에 대해 최종 결과에 대한 가시적인 응답이 있을 때까지 비동기 트랜잭션의 결과는 큐에 유지되어야 한다.
- 회복 가능성: 사용자가 빠른 응답 시간을 요구할 때라도 시스템이 요청을 잃어버리지 않는 것을 보장해야 한다. 따라서 사용자에게 요청을 잘 받았다는 신호를 보내기 전에 디스크 같은 안전한 기억장소에 요청을 저장하는 기능이 중요하다. 즉, 요청이 수행되기 전에 시스템 고장이 나더라도 요청을 잃어 버려서는 안된다.



(그림 4.4) 휘발성 메시지 큐
(Fig. 4.4) Volatile message queue

본 정보 검색 처리 모니터에서는 온라인 검색요청과 요청의 안정성을 위하여 위의 두 가지 형태의 큐를 모두 제공한다. 먼저 휘발성 메시지 큐는 모든 프로세스가 접근 가능하도록 UNIX System V에서 제공하는 공유 메모리를 이용하여 구현한다. 또한 실시간을 요하는 요청을 지원하기 위하여 휘발성 큐는 클라이언트가 제시한 데드라인에 기초한 우선 순위 큐로 구성된다. 따라서 우선 순위에 따라 요청을 큐에 삽입하기 위해 우선 순위 큐는 양방향 링크드 리스트로 구현하였다. (그림 4.4)는 구현된 메시지 큐의 구조를 보여준다. 그림에서 볼 수 있듯이 우선 순위 큐의 헤더 부분에 HEAD와 TAIL 필드를 두어 각각 최고 우선 순위와 최저 순위 요청 메시지를 가리키고 있다. 요청 메시지는 HEAD부터 검색을 하여 데드라인에 따라서 제 위치를 선정하여 삽입한다. 또한 응용 서버로 부터 메시지 추출은 TAIL 포인터가 가리키는 메시지로 부터 시작된다.

안정된 메시지 큐는 일단 모니터로 요청된 요청 메시지를 시스템 고장이 있을 경우에라도 그 내용을 잃어 버리지 않기 위해서 사용된다. 안정된 큐를 이용하는 비동기 요청은 다음과 같은 독립적인 세개의 트랜잭션으로 구성될 수 있다. (그림 4.5)는 구현된 안정된 큐를 나타낸다. 첫번째 트랜잭션인 트랜잭션 1은 요청 메시지를 생성하여 해당 서버의 메시지 큐에 삽입한다. 두번째 트랜잭션은 서버 요청 큐에 있는 요청에 대하여 요청을 수행한 후 그 결과를 해당 응답 큐에 집어 넣는다. 세번째 트랜잭션인 트랜잭션 3은 응답 큐에서 결과를 추출하여 클라이언트에게 결과를 전달한다. 안정된 메시지 큐에서 수행되는 모든 작업이 트랜잭션 이므로 만약 시스템 고장이 난 경우 정보 검색 시스템이 회복을 마치면 고장 당시 수행되고 있는 요청 메시지의 상태를 알 수 있다.



(그림 4.5) 안정된 메시지 큐
(Fig. 4.5) Nonvolatile message queue

4.3.4 이름 서비스 요청제어

이름 서버는 Tuxedo System의 bulletin board와 유사한 방법으로 설계/구현된다. 즉, 클라이언트와 서버 사이에 이름서버/요청제어 모듈을 둠으로써 다 대 다 통신을 가능하게 한다.

이름 서버는 공유 메모리 시스템 테이블에 클라이언트와 서버에 대한 각종 정보를 저장해 둔다. 따라서 클라이언트는 자기가 원하는 서비스를 처리해 줄 서버에 대한 자세한 정보를 알 필요없이 서비스 이름만을 통해 서비스를 요청할 수 있다. 클라이언트가 서비스를 요청할 때 우선 이름서버를 통해 요청된 서비스를 제공하는 서버의 정보를 검색한 후 해당 서버에게 요청을 보낸다. 정보 검색 시스템을 위한 서버에는 자동 색인을 위한 서버와 검색 연산을 서비스하기 위한 서버로 구성된다. 또한, 부하의 분산을 위해 한 개의 응용서버는 다수개의 서버를 포함하며, 여러 개의 동일서버는 한 개의 서버그룹을 이루는 응용서버의 계층구조 및 한 그룹내의 다중서버가 단일 큐를 공유하는 Multiple Server Single Queue 구조를 사용한다.

5. 성능 평가

제안된 프로세스 구조의 타당성을 제시하기 위해서 본 논문에서 설계한 정보검색처리 모니터의 성능을 평가한다. 성능 시험 수단으로는, 데이터베이스 시스템 및 온라인 트랜잭션 처리 시스템의 성능 시험에 널리 사용되고 있는 산업계 표준인 TPC-A 벤치마크를 사용한다. TPC-A 벤치마크에서 온라인 트랜잭션 시스템의 성능을 평가하기 위해 사용되는 척도에는 반응시간(response time)과 초당 처리되는 트랜잭션의 수를 나타내는 처리율(throughput)이 있다.

본 논문에서 수행한 성능 평가는 시스템 및 DBMS의 TPS 성능을 고려하여 기본 크기의 TPC-A 벤치마크 데이터베이스를 사용하여 수행하였으며, 벤치마크를 수행한 시스템 환경은 다음과 같다.

- 사용기종:SUN-center 1000
- Informix-Online DBMS
- Solaris 2.4 운영체제(64 MB 주기억용량)

정보 검색 처리 모니터를 사용하지 않는 정보 검색 시스템의 프로세스 구조는 두 경우를 고려하였다. 하나는 트랜잭션의 요청을 위한 프로세스가 시작될 때마다 이에 대응하는 응용 서버가 1 대 1로 시작되도록 하는, 종래의 정보 검색 시스템의 프로세스 구조의 클라이언트와 서버간의 통신을 위해 파이프를 사용한 시스템이고, 다른 하나는 클라이언트와 서버간의 통신이 지역 프로시져 호출(local procedure call)로 수행되는 시스템이다. 성능 시험은 정보 검색 처리 모니터를 사용하지 않은 종래의 정보 검색 시스템과 고정된 수의 응용 서버를 항상 수행 상태로 대기하도록 하여, 클라이언트로 부터의 트랜잭션 요청을 서버에게 배분하는 정보 검색 처리 모니터를 갖는 정보 검색 시스템의 프로세스 구조의 성능 비교를 위해 평균 반응 시간, 최대 반응 시간, 초당 처리되는 트랜잭션의 수 관점에서 수행하였다.

〈표 5.1〉은 벤치마크를 수행한 결과를 나타낸다. 수행된 결과를 살펴보면, 1 tps 환경에서는 대체로 TPC-A 기준에 가깝게 나타나고 있지만 사용자의 수가 많아지면(30개) 프로세스 개수의 증가로 인한 시스템의 프로세스 스위칭, 프로세스의 생성 및 삭제 등과 같은 오버헤드로 처리율이 낮고 15분간의 측정 시간동안 완료되는 많은 트랜잭션들중 90%가 2초 이내에 트랜잭션을 완료해야 한다는 TPC-A의 시간 제약을 만족하지 못하고 있다.

〈표 5.1〉 정보 검색 처리 모니터 벤치마크 결과
〈Table 5.1〉 Benchmark results

프로세스 구조	수행된 트랜잭션수	TPS	평균 RT	최대 RT
정보검색처리모니터를 사용하지 않는 기존 프로세스 구조 Ⅰ (지역 프로시져)	3225	3.58	5.99	15.59
정보검색처리모니터를 사용하지 않는 기존 프로세스 구조 Ⅱ (파이프)	1244	1.38	8.63	22.15
정보검색처리모니터를 사용하는 새로운 프로세스 구조	4740	5.27	3.58	9.49

하지만 정보 검색 처리 모니터를 갖는 정보 검색 시스템의 프로세스 구조는 정보 검색 처리 모니터를 사용하지 않으면서 서버와 클라이언트간의 통신이 파이프를 통해 이루어지는 기존의 정보 검색 시스템

의 프로세스 구조에 비해 반응 시간이 141%, 133%, 초당 처리되는 트랜잭션의 수가 282% 만큼 성능이 향상되었다. 또한 제안된 프로세스 구조는 정보 검색 처리 모니터를 사용하지 않으면서 서버와 클라이언트간의 통신을 지역 프로시져 호출을 통해 이루어지는 정보 검색 시스템의 프로세스 구조에 비해서는 반응시간이 67%, 64%, 초당 처리되는 트랜잭션의 수가 47% 만큼 성능이 향상되었다.

이와같은 결과는 제안된 프로세스 구조가 서버를 부하에 따라 그 수를 고정하여 수행 상태로 대기하므로, 시스템의 프로세스 수를 감소시켜 프로세스 스위칭을 경감시킬 수 있으며, 트랜잭션이 요청될 때마다 이를 처리해 주기 위한 새로운 프로세스를 매번 생성시키는 오버헤드도 없기 때문이다. 제안된 프로세스 구조는 효율적이어서 통신 오버헤드 역시 지역 프로시져 호출 경우에 비해 많이 떨어지지 않는 장점을 갖는다.

결과적으로 제안된 정보 검색 처리 모니터를 갖는 새로운 프로세스 구조는 정보 검색 처리 모니터를 사용하지 않는 기존 정보 검색 시스템의 1:1 프로세스 구조에 비해 성능이 모든 측면에서 향상되었음을 확인할 수 있었다.

6. 결론 및 향후 연구 방향

본 논문에서는 정보 검색 시스템과 유사한 특성을 갖는 온라인 트랜잭션 처리 시스템에서 많은 프로세스, 단말기 등과 같은 자원들을 용이하게 제어할 수 있는 트랜잭션 처리 모니터와 프로세스 관리에 따른 부하를 줄이는 다중 스레드에 대해 살펴보고 이를 기반으로 효율적인 정보 검색 시스템을 위한 프로세스 구조를 제안하였다. 설계된 정보 검색 시스템을 위한 프로세스 구조는 전체적인 정보 검색 시스템의 실질적인 구현구조와 시스템 환경 및 동작특성을 고려하여 정보 검색 시스템의 프로세스 오버헤드를 최소화하고 많은 수의 사용자 환경를 효율적으로 지원한다.

또한 제안된 프로세스 구조의 타당성을 제시하기 위해 벤치마크를 통해 정보 검색 처리 모니터를 사용하지 않는 시스템의 1:1 프로세스 구조와 정보 검색 처리 모니터를 사용하는 시스템의 프로세스 구조와의 성능평가를 수행하였다. 성능 평가 결과, 정보 검색 처리 모니터를 갖는 정보 검색 시스템의 프로세스 구조가 정보 검색 처리 모니터를 사용하지 않으면서 서버와 클라이언트간의 통신이 파이프를 통해 이루어지는 기존의 정보 검색 시스템의 프로세스 구조와 정보 검색 처리 모니터를 사용하지 않으면서 서버와 클라이언트간의 통신을 지역 프로시져 호출을 통해 이루어지는 정보 검색 시스템의 프로세스 구조에 비해 반응 시간이 각각 168%(134%), 23%(78%), 초당 처리되는 트랜잭션의 수가 272%(291%), 35%(64%) 만큼 성능이 향상되었다.

앞으로 설계된 정보 검색 시스템의 정보 검색 처리 모니터를 실질적인 정보 검색 시스템에 적용하여 UNIX 시스템 상에서 많은 사용자가 정보 검색 시스템을 사용할 때 시스템의 오버헤드를 줄여 효율적인 사용을 도모하고자 한다.

참 고 문 헌

[1] A. S. Tanenbaum, "Modern Operating System," Prentice-Hill, Inc., pp. 362-393, 1992.

[2] C. Brian, "UNIX and Online Transaction Processing," Datapro Research, APRIL 1990.

[3] D. B. McCarn, "MEDLINE: An Introduction to On-Line Searching," Journal of the American Society for Information Science, Vol. 40, No. 5, pp. 304-310, 1989.

[4] F. W. Lancaseter, "Information Retrieval Systems: Characteristics, Testing and Evaluation," John Wiley & Sons, Inc., 1979.

[5] G. Salton, A. Wong and C. S. Yang, "A Vector Space Model for Automatic Indexing," Communications of the ACM, Vol. 18, No. 11, pp. 613-620, 1975.

[6] ISO/IEC JTC1/SC21.31, "Information Technology-Open Systems Interconnection-Distributed Transaction Processing-Part 1:OSI TP Model," May 1991.

[7] P. A. Berstein, "Transaction Processing Monitors," CACM, Vol. 33, No. 11, Nov. 1990, pp. 75-86.

[8] R. Rada and E Bicknell, "Ranking Documents

with a Thesaurus," Jounal of the American Society for Information Science, Vol. 40, No. 5, pp. 304-310, 1989.

[9] W. B. Croft, "Boolean Queries and Term Dependencies in Probabilistic Retrieval Models," Journal of the American Society for Information Science, Vol. 37, No. 2, pp. 71-77, 1986.

[10] T. Radecki, "Fuzzy Set Theoretical Approach to Document Retrieval," Information Processing and Management, Vol. 15, No. 5, pp. 247-259.

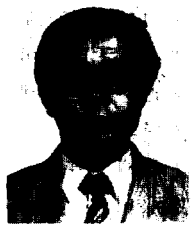
[11] T. Radecki, "Similarity Measures for Boolean Search Request Formulations," Journal of the American Society for Information Science, Vol. 33, No. 1, pp. 8-17, 1982.

[12] G. Salton, "Recent Trends in Automatic Information Retrieval," In Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, pp. 1-10, 1986.

[13] G. Salton, E. A. Fox and H. Wu, "Extended Boolean Information Retrieval," Communications of the ACM, Vol. 26, No. 11, pp. 1022-1036, 1983.

[14] J. H. Lee. "Document Ranking Methods for Thesaurus-Based Boolean Retrieval Systems." Ph. D. thesis, Dept. of Computer Science, KAIST, 1993.

[15] M. H. Kim, "Design and Implementation of Real Time TP Monitor Prototype," ETRI Final Report, 1992.



고 형 대

1982년 전남대학교 계산통계학과(학사)
1984년 전남대학교 대학원 전산통계학과(석사)
1991년 전남대학교 대학원 전산통계학과(박사수료)
1985년~현재 목포대학교 전산통계학과 부교수
1992년~1996년 2월 목포대학교 정보산업연구소 소장
관심분야:소프트웨어공학, 프로젝트 관리

유 재 수

1989년 전북대학교 공과대학 컴퓨터공학과(학사)
1991년 한국과학기술원 전산학과(공학석사)
1995년 한국과학기술원 전산학과(공학박사)
1995년~1996년 8월 목포대학교 전산통계학과 전임강사
1996년 8월~현재 충북대학교 공과대학 전기전자공학부 전임강사
관심분야:데이타베이스 시스템, 정보검색, 멀티미디어 데이타베이스, 분산 객체 컴퓨팅 등

김 병 기

1978년 전남대학교 수학교육과(학사)
1980년 전남대학교 대학원 수학과(이학석사)
1981년~현재 전남대학교 전산학과 교수
관심분야:소프트웨어 공학 신경망 컴퓨터, 초고속 정보통신 등