

# 광주크박스 저장장치에 기반한 비디오 서버

## (Video Server with Optical Mass Storage)

이종민<sup>\*</sup>    차호정<sup>\*\*</sup>    오재학<sup>\*\*\*</sup>    박병준<sup>\*\*\*\*</sup>  
 (Jongmin Lee)    (Hojung Cha)    (Jaehak Oh)    (Byungjoon Park)

**요약** 본 논문은 광 주크박스과 테잎 라이브러리 같은 Tertiary 저장 장치를 미디어 저장 장치로 사용하는 비디오 서버의 설계와 구현에 대하여 기술한다. Tertiary 저장 장치를 스트리밍 서버의 미디어 저장 장치로 사용하기 위해서는 Tertiary 저장 장치의 하드웨어적인 특성을 고려한 적절한 스트리밍 서비스 모델이 필요하다. 논문에서는 연속 미디어의 특성을 고려한 스트림 스케줄링, 디스크 캐싱, 사용자 수용 제어 정책등을 제안하고 이를 핵심으로 하는 스트리밍 서비스 모델을 제시한다. 제안된 시스템은 광주크박스 저장장치를 장착한 Windows 2000 플랫폼에서 구현되었고, 다양한 실험 및 성능 분석을 통해 제안된 정책들이 올바르게 동작함을 검증하였다.

**키워드** : 비디오 서버, 광 주크박스

**Abstract** This paper discusses the design and implementation of a video server which uses tertiary device, such as magneto-optical jukebox or tape library, as a source of media archiving. In order to handle the tertiary device in the framework of disk-based stream service model, a sophisticated streaming model is required for server and it should consider the device-specific performance characteristics of tertiary storage. We have carefully designed the streaming mechanism for server whose key functionalities include stream scheduling, disk caching and admission control. The proposed system has been implemented on Windows 2000 platform which is equipped with a magneto-optical jukebox. The design principles of the server are validated with the experiments and the performance characteristics is analyzed.

**Key words** : Video Server, Optical Mass Storage

### 1. 서론

최근 컴퓨터 하드웨어와 네트워크 기술의 발전으로 인터넷을 이용한 주문형 비디오서비스가 활성화되고 있다. 주문형 비디오 서비스는 통신망을 통하여 다양한 비디오를 원하는 방식대로 시청할 수 있는 대화형 미디어 서비스로 안정된 서비스를 위한 연속된 스트리밍 보장

과 다양한 사용자의 욕구를 만족시키기 위한 많은 컨텐츠의 확보가 필요하다[1]. 기존의 디스크 기반의 비디오 서버는 고품질의 서비스와 안정성을 보장하는 반면 저장 장치의 구성에 많은 비용이 든다. 대용량 비디오 서버의 구성은 현실적인 가격대 성능비를 고려하여 저가의 저장 공간을 제공하는 Tertiary 저장 장치의 활용이 필요하다[2, 3].

테잎 라이브러리나 광주크박스와 같은 Tertiary 저장 장치는 미디어를 보관할 수 있는 수 십개에서 수 백개의 슬롯과 미디어를 읽기 위한 다수개의 드라이브, 그리고 슬롯과 드라이브간의 미디어 이동을 위한 드라이브로 구성된다. Tertiary 저장 장치에 대한 입출력 요구는 로 봇암이 슬롯으로부터 저장 미디어를 선택하여 드라이브에 로드함으로써 입출력 기능을 수행한다. Tertiary 저장 장치는 테라에서 페타 바이트에 이르는 대용량의 저장 공간을 제공하는 반면, 입출력 대역폭이 낮은 단점이

\* 본 연구는 정보통신부 대학기초 연구지원 사업(과제번호: 2001- 076-3) 과 2001년도 광운대학교 교내학술 연구비 지원에 의해 연구되었음.

<sup>\*</sup> 비 회 원 : 연세대학교 컴퓨터과학과  
 jmlee@cs.yonsei.ac.kr

<sup>\*\*</sup> 종신회원 : 연세대학교 컴퓨터과학과 교수  
 hjcha@cs.yonsei.ac.kr

<sup>\*\*\*</sup> 학생회원 : 광운대학교 컴퓨터과학과  
 ojh@cs.kwangwoon.ac.kr

<sup>\*\*\*\*</sup> 종신회원 : 광운대학교 컴퓨터과학과 교수  
 bjpark@cs.kwangwoon.ac.kr

논문접수 : 2001년 4월 14일

심사완료 : 2002년 3월 8일

있다. 요구하는 자료가 드라이브에 로드되어 있지 않은 경우 해당 미디어를 드라이브에 로드하기 위한 지연시간이 소요되며 이러한 지연시간은 보통 수십초에 이른다. 이러한 Tertiary 저장 장치를 대용량 비디오 서버의 저장 장치로 사용하기 위하여 몇가지 고려해야 할 사항이 있다.

첫째, 계층적 자료관리이다. 계층적 자료 관리는 저장 장치를 각각의 특성에 맞게 계층화하여 이를 관리하는 방법을 말한다[4, 5]. 계층적 자료 관리를 구성하는 방법은 하드디스크와 같은 높은 자료 전송능력을 가지나 상대적으로 고가인 저장 장치를 계층구조의 상위에 두어 자주 요청되는 자료를 저장하고, 광주크박스과 같은 전송 능력은 떨어지나 가격이 저렴한 저장 장치를 계층구조의 하단에 위치시켜 상대적으로 요구가 적은 자료를 저장한다. 계층적 자료 관리를 함으로써 같은 용량의 저장 용량을 확보할때 고가인 저장 장치로만 구성하는 것보다 가격이 저렴하며, 자주 요청되는 자료는 계층구조의 상단에 저장되므로 빠른 제어가 가능하다. 둘째, 디스크의 효율적인 사용이다. 계층적 자료 관리의 핵심적인 구성 요소인 하드디스크는 Tertiary 저장 장치와의 데이터 전송을 위한 디스크 캐쉬로 사용될 수 있다. 디스크 캐쉬의 적절한 구성 방법 및 관리는 전체 시스템의 성능을 결정한다. 디스크 캐쉬의 관리는 범용 운영체제에서의 버퍼 캐쉬와는 다르게 관리되어야 한다. 버퍼 캐쉬에서와 같이 자료 요구 확률이 높은 랜덤한 일부 블록의 보장보다는 연속 미디어의 특성을 반영하는 새로운 캐쉬 관리 방법이 필요하다. 셋째, 스트림 스케줄링 정책이다. 스케줄링 정책은 사용자 QoS 를 보장하기 위한 프로세스 스케줄링과 미디어 스테이징을 위한 스테이징 스케줄링이 있다. 사용자 요구는 서버에 수용된 후 프로세스와 스테이징에 대한 스케줄링을 구성하고 각각의 제한 조건을 만족시킴으로써 QoS가 보장된다. 스테이징 스케줄링 정책은 Tertiary 저장장치에서 디스크 캐쉬로의 자료 전송에 관한 입출력 스케줄링을 의미하며 비디오 서버의 성능을 위한 중요한 변수로 사용자의 자료 요구율과 디스크 캐쉬의 크기 및 Tertiary 저장장치의 입출력 대역폭을 고려해야 한다. 넷째, 적절한 사용자 수용 제어 정책이다. 다양한 사용자의 요구에 대한 시스템 자원 사용 변동량을 동적으로 측정하여 이를 사용자 수용 제어의 기준으로 삼아야 한다. 정확한 시스템 자원의 사용량을 사용자 수용 제어의 기준으로 하여 시스템 자원을 최대한 활용하여 서비스 요청이 허가된 사용자에게는 QoS 를 보장할 수 있도록 해야한다.

Tertiary 저장 장치를 주문형 비디오 서비스에 응용

하려는 연구는 다양하게 진행되었다. 특히, 연속 비디오 자료를 Tertiary 저장 장치의 미디어에 저장하는 방법, 저장된 자료를 디스크 저장 장치로 옮기는 방법, 이러한 작업들을 효과적으로 수행하기 위한 스케줄링 방법, 한정된 디스크 저장 장치를 보다 효율적으로 사용하기 위한 방법등에 관한 연구들이 활발히 이루어졌다. Tertiary 미디어 저장 및 배치에 관한 연구로는 비디오의 초기 부분과 중간의 여러 부분들을 선반입하는 방법, Tertiary 저장 장치의 대역폭을 적절히 활용할 수 있도록 미디어의 자료를 블록 단위로 엇갈리게 저장하는 방법, 비디오 화일을 페이지 단위로 관리하는 방법등의 연구가 있다[5, 6, 7, 8]. Tertiary 저장 장치에서 디스크 저장 장치로 자료를 이동하는 방법에 관한 연구로는 일반적인 파이프라인 방법, Tertiary 저장 장치의 자료를 디스크 저장 장치로 바로 이동시키지 않고 디스크 저장 장치가 Idle 상태일 때 이동시키는 Delayed Disk Write 등의 방법이 있다[5, 9]. Tertiary 저장 장치를 효율적으로 사용하기 위한 스케줄링에 관련된 연구로는 최근에 사용된 미디어들을 Tertiary 저장 장치의 드라이브 가까운 곳에 위치시켜 해당 미디어에 대한 접근 시간을 줄이는 방법, Tertiary 저장 장치의 드라이브에 로드된 미디어에 대한 모든 요구를 해당 미디어가 언로드 되기 전에 수행하는 방법등이 있다[10, 11]. 한정된 디스크 저장 장치를 효율적으로 사용하기 위한 연구로는 디스크 캐쉬 교체 정책에 대한 연구가 있다. 비디오의 크기와 선호도, 디스크 캐쉬에 있는 자료의 양, 현재 서비스 여부 등을 고려한 PIRATE (Partial Replacement Technique) 알고리즘이나 다중 스트림을 지원하는 PIRATE 등이 있고 LFU(Least Frequently Used)와 LEAT(Longest Expected Access Time)을 이용한 방법이 있다[2,6,12].

이와같은 기존의 연구는 Tertiary 저장 장치를 주문형 비디오 서버에 응용하기 위한 부분적인 정책의 연구에 중점을 두고 있다. 즉, 미디어의 저장 및 재생, 디스크 캐싱, 사용자 수용 제어, 스트림 스케줄링과 같은 지역적인 정책의 연구에 국한되어 있고 다른 정책들에 대한 고려가 이루어지고 있지 않으며 연구의 검증이 모의 실험등을 통해서만 이루어져 이러한 지역적인 정책들을 종합하여 통합된 하나의 실제 시스템을 구성하는데 있어서 실시간성을 만족시키지 못하는 등의 여러가지 문제점이 있을 수 있다. 실제의 서버를 구현하기 위해서는 정확히 정의된 시스템 모델을 바탕으로 여러 개별적인 정책들이 일관성있게 정의되고 설계되어야 한다. 본 논문에서는 광주크박스 저장 장치를 사용한 대용량 비디오

오 서버의 설계와 구현에 대하여 기술한다. 안정된 스트림 서비스를 위하여 하드디스크 저장 장치를 캐쉬 장치로 사용하며 사용자 QoS를 보장하기 위한 다양한 서버 운영 및 관리 정책을 제시하고 성능을 분석한다. 많은 저장 공간을 제공하지만 입출력 대역폭이 낮은 Tertiary 저장장치를 최대한 활용하기 위하여 Tertiary 저장장치에서 디스크 캐쉬로의 스테이징 작업을 위한 스케줄링 방법이 필요하다. 스테이징 스케줄링은 주기 모델을 적용하며 서비스의 안정성을 위해 위급도가 높은 스테이징 작업이 먼저 수행될 수 있도록 하며, 여유 스테이징 대역폭의 관리를 통해 스테이징 대역폭을 최대한 활용한다. 대용량 비디오 서버를 위한 디스크 캐쉬는 서비스의 안정성을 위해 고정 크기의 캐쉬 블록 단위로 관리되며, 연속 미디어의 특성을 고려한 캐쉬 교체 정책을 통하여 서비스의 안정성과 연속성을 보장한다. 또한 적절한 사용자 수용 제어 정책을 통하여 스트림 스케줄링 및 디스크 캐싱 정책이 올바르게 작동 할 수 있도록 한다. 논문에서 제시하는 스트림 스케줄링 정책, 디스크 캐쉬 관리 정책, 사용자 수용 제어 정책들은 일관되게 설계되었으며 각각의 정책들이 다른 정책들의 고려를 통해 통합된 하나의 시스템에서 잘 동작할 수 있도록 하였다. 논문에서 제시된 스트림 스케줄링 정책, 디스크 캐쉬 관리 정책, 사용자 수용 제어 정책들을 범용 운영체제상에서 실제 구현하고 성능 분석함으로써 설계된 서버의 기능 및 성능 특성을 파악한다.

논문의 구성은 다음과 같다. 2장에서 서버의 설계와 구현에 대하여 기술하고 3장에서 구현된 서버의 성능 분석을 하며 4장에서 결론을 맺는다.

## 2. 서버 설계 및 구현

다음에는 Tertiary 저장 장치를 사용하는 비디오 서버의 설계와 구현에 대해 설명한다. 서버의 핵심 구성 요소인 스트림 스케줄링 정책, 디스크 캐쉬 관리 정책, 사용자 수용 제어 정책을 기술한다.

### 2.1 서버 구조

서버는 크게 서비스와 프로세스 관리 서버 시스템, 입출력 관리 서버 시스템, 자원 관리 서버 시스템의 세 가지 관리 서버 시스템으로 구성된다. 서비스와 프로세스 관리 서버 시스템은 사용자의 서비스 요구를 받아들이며 서비스 요구에 대한 사용자의 수용 제어와 사용자들에게 연속적이고 안정적인 서비스를 하기 위한 실시간 스케줄링 정책을 담당한다. 입출력 관리 서버 시스템은 입출력 순서등을 정돈하는 입출력 스케줄링과 실제 입출력 작업을 하기 위해 저 레벨의 디바이스 드라이버를

제어하는 작업을 한다. Tertiary 저장 장치에서 디스크 캐쉬로의 자료 이동을 관리하는 스테이징 스케줄링과 계층적 자료 관리, 디스크 캐쉬 관리가 모두 입출력 관리 서버 시스템에서 이루어진다. 자원 관리 서버 시스템은 시스템 자원의 종합적인 관리를 담당한다. 자원 관리 서버 시스템은 세분화된 여러 개의 자원 관리자들로 구성되며 각각의 자원 관리자들은 세분화된 특정 시스템 자원의 관리를 담당한다.

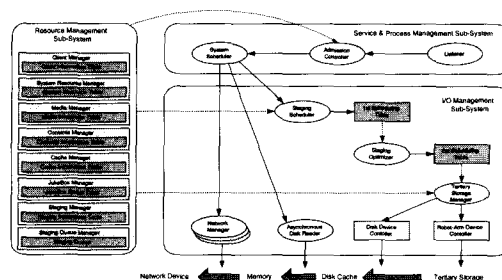


그림 1 서버 구조

그림 1은 서버의 세부 구조를 보여준다. 각 관리 서버 시스템은 세분화된 작업을 담당하는 여러 가지 모듈들로 구성되어 있다. 각각의 관리 서버 시스템의 구성 요소와 각 구성 요소의 역할은 다음과 같다. 서비스와 프로세스 관리 서버 시스템은 사용자의 서비스 요구를 처리한다. 또한 사용자의 수용 제어 및 안정된 스트리밍 서비스를 위한 쓰레드 스케줄링 정책을 수행하며 사용자의 서비스 요구를 받아들이는 Listener, 사용자 수용 여부를 결정하는 사용자 수용 제어기, 사용자의 자료 요구에 따른 위급도를 판단하여 서비스 순서를 결정해주는 시스템 스케줄러로 구성된다. 입출력 관리 서버 시스템은 Tertiary 저장 장치와 디스크 캐쉬 간의 입출력에 관련된 부분을 담당한다. 입출력을 위한 저 레벨의 디바이스 드라이버를 제어하는 역할과 입출력 순서를 정돈하는 입출력 스케줄링을 수행한다. 저 레벨 디바이스를 제어하기 위한 모듈로는 로보탐 제어기, 디스크 제어기 등이 있고, 입출력 스케줄링에 관계된 모듈로는 스테이징 스케줄러, 스테이징 최적화 모듈 등이 있다. 자원 관리 서버 시스템은 시스템 자원의 종합적인 관리를 담당한다. 자원 관리 서버 시스템은 세부 특성화된 자원 관리자들로 구성되며 특성화된 자원 관리자들은 각각의 세부 자원을 독립적으로 관리한다. 자원의 관리는 사용자 수용 제어의 중요한 기준이 되며 정확한 시스템 자원의 측정이 곧 사용자 QoS 보장과 연결된다. 대부분

의 관리자들이 관리하는 자료들은 리스트 형태로 관리되며 자료의 종류나 양에 따라 트리나 테이블 형태로도 관리된다. 관리되는 자료들은 주기적으로 로컬 디스크에 저장되어 시스템 비정상 작동 등의 경우 복구가 가능하도록 한다.

사용자의 요구를 시작으로 최종 Tertiary 저장 장치의 입출력 스케줄링까지의 과정은 다음과 같다. 사용자 요구는 Listener 를 통해 접수된다. 접수된 요구는 사용자 수용 제어기로 보내어지며 사용자 수용 제어기에서는 자원 관리 서버 시스템으로 부터 얻은 현재 시스템 자원 사용 상황을 고려하여 새로운 스트림의 수용 여부를 결정한다. 수용이 허락된 사용자 정보는 자원 관리 서버시스템에 등록되며 다음 스케줄링 대상에 포함된다. 시스템 스케줄러는 일정 주기로 실행되는 주기 모델로 실시간 스케줄러이다. 시스템 스케줄러는 사용자의 자료 요구율에 따라 위급도를 계산하고 EDF 방법에 의해 서비스 실행 순서를 결정한다. 결정된 서비스 순서에 의해 네트워크 관리자의 실행 순서를 변경하여 위급도가 높은 사용자의 요구가 먼저 처리될 수 있도록 한다. 스테이징 스케줄러도 일정 주기로 실행되는 주기 모델이다. 스테이징 스케줄러는 사용자의 자료 요구율과 디스크 캐쉬량, Tertiary 저장 장치의 입출력 대역폭을 고려하여 스테이징 작업의 위급도를 판단하고 EDF 방법에 의해 스테이징 작업의 순서를 결정한다. 스테이징 스케줄러에서 생성된 스테이징 작업 순서는 스테이징 최적화 모듈에서 최적화 되며, 최적화를 통해서 여유 스테이징 대역폭을 최대한 확보할 수 있다. Tertiary 저장 장치 관리자는 스테이징 최적화 모듈에서 나온 결과를 Tertiary 저장 장치의 로봇암 제어 부분과 디스크 저장 장치 제어 부분으로 나누어 해당 모듈을 제어하며 이들 작업간의 동기화를 담당한다. Tertiary 저장 장치에서 디스크 캐쉬로의 자료 이동은 스테이징 스케줄러의 결과인 스테이징 스케줄 작업 목록에 따라 이루어지며 디스크 캐쉬에서 주 메모리로의 자료 이동은 비동기 디스크 읽기 모듈에서 담당하고 주 메모리에서 네트워크 장치를 통한 자료 전송은 네트워크 관리자가 담당한다.

**2.2 스케줄러**

스트리밍 서버에서의 스트림 스케줄러는 사용자에게 연속되고 안정된 서비스를 제공하기 위하여 필수적인 요소이다. Tertiary 저장 장치를 사용한 비디오 서버의 스케줄링 모델은 시스템 스케줄링과 스테이징 스케줄링으로 구분된다.

**2.2.1 시스템 스케줄러**

본 논문에서의 비디오 서버는 클라이언트가 자료를

필요로 할때 서버로 자료를 요청하여 이 때 서버가 미리 정해진 네트워크 패킷 단위로 자료를 전송해 주는 Pull 모델이다. 시스템 스케줄러는 일정 시간 간격으로 실행되는 주기 모델이며 실시간 스케줄러이다. 시스템 스케줄러는 클라이언트의 자료 요구율에 따라 위급도를 결정하고 결정된 위급도에 따라 실행 순서를 결정한다. 결정된 실행 순서에 따라 해당 네트워크 관리자들을 실행시켜 위급도가 높은 사용자의 요구가 먼저 실행될 수 있도록 한다. 또한 비동기 디스크 읽기 모듈을 실행시켜 디스크 캐쉬에서 주 메모리로의 자료 이동이 이루어지도록 하며 스테이징 스케줄러가 실행될 수 있도록 한다.

시스템 스케줄러의 실행 주기는  $T_{system}$ 이고, 각 클라이언트의 데이터 요구 주기는  $T_i$ 이다.  $T_i$ 는 클라이언트의 대역폭을 의미하는 데이터율( $R_i$ )에 따라 결정되며 데이터율이 높은 클라이언트는 데이터율이 낮은 클라이언트에 비해 짧은 주기를 갖는다.  $T_{system}$ 는 식 1, 식 2에 의해 결정된다.

$$T_{system} \leq \min(T_1, T_2, \dots, T_n) \tag{1}$$

$$T_{system} = \begin{cases} k \times T_{disk} & \text{if } B_{disk} < B_{net} \\ k \times T_{net} & \text{otherwise} \end{cases} \tag{2}$$

$T_i$ 는 i번째 클라이언트의 데이터 요구 주기를 의미한다. 식 1은  $T_{system}$ 주기 내에서  $T_i$ 가 한번의 주기를 가짐을 의미하며  $T_{system}$ 주기 내의 클라이언트의 요구들을 위급도에 따라 우선 순위를 정하게 된다.

$T_{system}$ 내에 스케줄링된 요구들은 다른 요구에 의해서 선점되지 않으며 자신의 수행이 끝난 후에 다음 요구를 수행한다. 시스템 스케줄러의 주기  $T_{system}$ 에서 실행할 수 있는 요구의 갯수는 식 2에 의해 결정된다.

디스크 저장장치의 대역폭( $B_{disk}$ )과 네트워크 대역폭( $B_{net}$ )간의비교를 통해 작은 대역폭을 기준으로 동시 허용 가능한 클라이언트의 갯수를 정함으로써 시스템 자원량 초과를 방지한다. 디스크에서 메모리로의 자료이동은 입출력 관리 서버 시스템의 비동기 디스크 읽기 모듈에서 수행되며  $T_{system}(i)$ 주기에서 디스크 입출력을 실행함과 동시에 네트워크 관리자의 요구를 요구 큐에 수용하고  $T_{system}(i+1)$ 내에 수행함으로써 네트워크 관리자의 요구율을 만족시킨다.

**2.2.2 스테이징 스케줄러**

스테이징 작업은 Tertiary 저장장치에서 디스크 캐쉬로의 자료 이동을 의미한다. 스테이징 작업은 미리 정해진 블록 단위로 수행된다. 스테이징 작업은 현재 서비스 중인 Active 상태의 비디오를 대상으로 하며, 사용자의 자료 요구율과 디스크 캐쉬에 저장되어 있는 비디오의 캐쉬되어 있는 양을 고려하여 어떠한 비디오가 먼저 스

테이징 되어야 하는지의 스테이징 작업의 위급도를 결정하고 위급도가 높은 스테이징 작업이 먼저 처리될 수 있도록 한다. 스테이징 작업의 위급도는 디스크 캐쉬에 저장되어 있는 자료의 양( $S_{cached(i)}$ )과 미디어 데이터율( $R_i$ )에 의해 결정된다( $S_{cached(i)}/R_i$ ). 즉, 디스크 캐쉬에 저장되어 있는 자료의 양으로 재생 가능한 시간이 짧을수록 해당 비디오에 대한 스테이징 작업의 위급도는 높아지게 된다. 디스크 캐쉬에 저장되어 있는 양으로 재생 가능한 시간은 해당 비디오에 대한 스테이징 작업의 마감 시간을 의미하며 이 마감 시간을 넘기게 되면 서비스의 일시 중지 혹은 종료로 가져온다. 한편,  $S_{cached(i)}$ 을  $i$ 번째 스트림이 향후 필요로 하는 캐쉬양이라고 정의할때 이 미디어 스트림의 스테이징 대역폭( $R_{cached(i)}$ )은 해당 미디어에 대한 스테이징 작업을 수행하는데 소모되는 스테이징 대역폭을 의미하며 식 3에 의해 계산된다. 스테이징 대역폭은 스테이징 스케줄링 및 사용자 수용 제어 위한 기준으로 쓰인다.

$$R_{staging(i)} = \frac{S_{cached(i)} \times R_i}{S_{cached(i)} + S_{caching(i)}} \quad (3)$$

스테이징 스케줄링의 가장 기본적인 방법으로 매 스테이징 작업이 완료될 때마다 서비스중인 Active 상태의 비디오들의 위급도를 계산하여 위급도가 가장 높은 스테이징 작업이 먼저 수행되도록 하는 단순 스테이징 스케줄링 방법이 있다. 이러한 마감 시간 우선 수행 정책은 각각의 서비스중인 비디오에 대한 스테이징 작업의 예측과 현재 사용중인 스테이징 대역폭의 측정이 어렵다. 이러한 문제는 여유 스테이징 대역폭의 계산을 어렵게 하며 사용자 수용 제어 정책에 여유 스테이징 대역폭을 반영하지 못하여 한정된 스테이징 대역폭 이상의 사용자에게 서비스 요구를 허용할 수 있다. 이는 곧 스테이징 작업의 마감 시간 초과를 가져오며 결과적으로 서비스의 일시 중지 혹은 종료를 가져온다.

단순 스테이징 스케줄링의 문제점인 스테이징 작업의 예측과 사용중인 스테이징 대역폭의 측정이 어려운 문제를 해결하기 위해 스테이징 스케줄링에 주기 모델을 적용할 수 있다. 각각의 비디오는 자신의 미디어 요구율을 가지고 있고 스테이징 작업 또한 각각의 비디오가 필요로 하는 요구 스테이징 대역폭을 가지며 이를 주기 모델로 일반화 할 수 있다. 이때 가장 높은 자료

요구율 ( $R_{max}$ )을 가지는 사용자를 기준으로 하여 하나의 고정 크기 캐쉬 블록( $S_{cacheblock}$ )을 재생하는데 걸리는 시간을 시스템 주기 ( $T_{system}$ )의 정수배가 되도록 하여 주기를 정할 수 있는데 이를 지역 주기 ( $T_{local}$ )라고 한다. 즉, 서비스 중인 Active 상태의 비디오를 위한

스테이징 작업을 하나의 지역 주기에 하나씩 수행하도록 하였을 때 모든 스테이징 작업이 마감 시간을 만족하며 수행될 수 있는 시스템 주기 정수배의 최대 주기를 지역 주기로 정하며, 식 4에 의해 구한다.

$$T_{local} = \frac{S_{cacheblock}}{R_{max}} \times T_{system} \quad (4)$$

지역 주기단위 스테이징 스케줄링은 지역 주기 단위로 스테이징 스케줄러가 실행되어 하나의 지역 주기에 실행 될 수 있는 스테이징 작업의 순서를 마감 시간 우선 정책에 의해 정한다. 이러한 지역 주기 단위 스테이징 스케줄링은 Active 상태의 비디오에 대한 스테이징 작업의 예측이 가능하나 여유 스테이징 대역폭의 측정이 어렵고 이를 지속적으로 보장하면서 한정된 스테이징 대역폭을 활용하기가 어렵다. 이를 해결하기 위한 방법으로 전역 주기 스케줄링 모델을 도입한다. 전역 주기는 서비스 보장 주기라고도 하며 Active 상태의 비디오들에 대해 지역 주기 모델을 적용했을 때 나타나는 스테이징 작업 순서가 동일한 형태가 반복되는 주기를 구하여, 이 주기들의 최소 공배수로 전역 주기 ( $T_{global}$ )를 정한다.

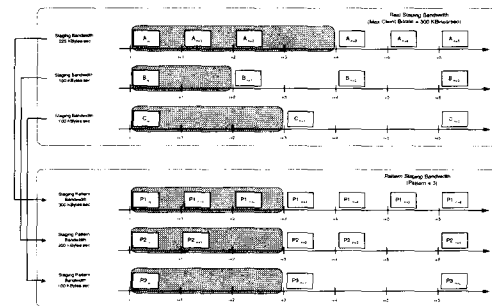


그림 2 스테이징 패턴

그림 2에서 Active 상태 비디오들의 요구 스테이징 대역폭에 따른 지역주기가 A 비디오의 경우 4, B 비디오의 경우 2, C 비디오의 경우 3 주기마다 반복되는 것을 알 수 있으며 이럴 경우 전역주기는 4, 2, 3의 최소공배수인 12가 된다. 전역 주기가 결정되면 전역 주기 단위로 모든 Active 상태의 비디오들을 위한 스테이징 작업이 같은 형태로 반복되고 전역 주기 단위로 사용 중인 스테이징 대역폭 측정이 용이하며 여유 스테이징 대역폭의 계산이 쉽다. 이렇게 얻어진 여유 스테이징 대역폭을 사용자 수용 제어 정책에 반영함으로써 스테이징 대역폭을 초과하는 사용자의 서비스 요구를 거절할 수 있으며

로 서비스중인 스트림의 안정성을 보장할 수 있다.

Active 상태의 비디오들에 대해 스테이징 작업의 순서가 동일하게 반복되는 형태가 나오는 주기들의 최소 공배수로 전역 주기를 정하게 되면 다양한 대역폭의 비디오들을 서비스 할 경우 각각의 Active 상태 비디오들에 대한 스테이징 작업 형태가 반복되는 주기들이 다양해지면서 전역 주기도 길어지게 된다. 전역 주기는 사용자가 새로운 서비스를 요구했을 때, 기존 사용자가 서비스를 종료했을 때, 그리고 Active 상태의 비디오의 요구 스테이징 대역폭이 바뀌었을 때 초기화되어 그 시점에서 다시 스테이징 스케줄링을 수행한다. 전역 주기가 길어지면 스테이징 작업이 완료 될 때마다 해당 비디오의 요구 스테이징 대역폭이 바뀌어 스테이징 작업이 반복되는 형태가 변할 확률이 커지며 이로 인해 잦은 전역 주기의 초기화가 일어난다. 최악의 경우 매 스테이징 작업이 완료 될 때마다 전역주기가 초기화되어 단순 스테이징 스케줄링 방법과 동일하게 될 수도 있다. 빈번한 전역 주기의 초기화로 인하여 스테이징 스케줄링의 효과가 충분히 나오지 않는 것을 보완하기 위해 Active 상태의 비디오가 요구하는 요구 스테이징 대역폭을 고정된 몇 개의 스테이징 대역폭에 매핑할 수 있다. 그림 2에 Active 상태 비디오의 요구 스테이징 대역폭을 3개의 일반화된 패턴 요구 스테이징 대역폭에 대응하는 예를 보여주고 있으며 전역 주기도 3으로 고정 됨을 알 수 있다. 이러한 고정 대역폭을 갖는 스테이징 패턴을 사용함으로써 전역 주기가 길어지는 것을 방지할 수 있고 짧은 전역 주기로 인한 스테이징 스케줄링의 오버헤드가 감소되며 전역 주기의 잦은 갱신을 피할 수 있으므로 전역 주기 단위의 스테이징 스케줄링 효과를 충분히 얻을 수 있다.

전역 주기 모델에서 패턴의 갯수는 하나의 전역 주기 안에 포함되는 지역 주기의 갯수를 의미하며, 또한 패턴 요구 스테이징 대역폭의 갯수를 의미하기도 한다. 패턴의 갯수가 적을수록 패턴 요구 스테이징 대역폭의 갯수가 적어지므로 다양한 대역폭의 사용자들을 몇 개의 패턴 요구 스테이징 대역폭에 대응함으로써 스테이징 대역폭의 낭비를 가져올 수 있다. 패턴의 갯수가 많아지면 다양한 대역폭의 사용자들을 수용하여 스테이징 대역폭을 잘 활용할 수 있으나 상대적으로 전역 주기가 길어지게 되어 스케줄링 작업이 복잡해지며 잦은 패턴 요구 스테이징의 변화로 스테이징 스케줄링 작업이 비효율적이 될 수 있다. 패턴 스테이징 대역폭은 식 5에 의해 계산된다.

$$R_{stagingpattern(n)} = R_{max} \times \frac{n}{N_{pattern}} \quad (5)$$

where  $1 \leq n \leq N_{pattern}$

식 5에서  $N_{pattern}$ 은 고정 대역폭 패턴의 개수이며 패턴의 갯수 만큼의 패턴 스테이징 대역폭이 계산된다. 그림 2의 예에서  $N_{pattern}=3$ ,  $R_{max}=300\text{KBytes/sec}$ 이므로  $R_{stagingpattern(1)}=100\text{KBytes/sec}$ ,  $R_{stagingpattern(2)}=200\text{KBytes/sec}$ ,  $R_{stagingpattern(3)}=300\text{KBytes/sec}$ 이 됨을 알 수 있다.

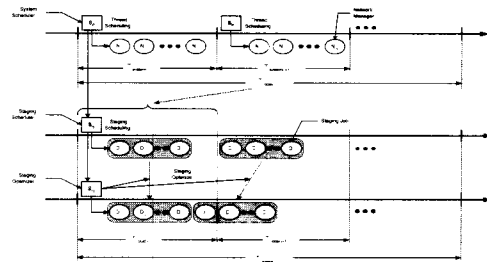


그림 3 스트림 스케줄링의 주기 모델

그림 3은 전역 주기 모델을 도식화 한 것이다. 시스템 스케줄러는 시스템 주기 ( $T_{system}$ ) 단위로 실행되며 사용자의 자료 요청에 대해 자료 요구율에 따른 위급도를 바탕으로 위급도가 높은 작업이 사용자가 먼저 서비스 받을 수 있도록 하는 EDF 방법에 의해 서비스 되어야 할 우선 순위를 결정하여 이에 해당되는 네트워크 매니저들의 실행 순서를 변경한다. 전역 주기의 스테이징 스케줄링은 스테이징 스케줄러가 전역 주기 ( $T_{global}$ ) 단위로 실행되며 서비스중인 비디오가 필요로 하는 요구 스테이징 대역폭을 계산하여 이를 패턴 스테이징 대역폭에 대응하여 하나의 전역 주기 안에 수행되어야 할 스테이징 작업의 갯수를 구한다. 하나의 전역 주기에 실행되어야 할 스테이징 작업의 갯수가 정해지면 이를 전역 주기 단위로 위급도에 근거한 스케줄링을 하여 작업의 순서를 결정한다. 전역 주기 단위 스테이징 스케줄링에서 전역 주기는 스테이징 스케줄러가 실행되는 주기이며 스테이징 작업의 예측과 사용중인 스테이징 대역폭의 측정, 여유 스테이징 대역폭 계산의 기준이 된다. 또한 서비스의 안정성을 위해 위급도가 높은 스테이징 작업이 먼저 수행되는 EDF 방법에 의해 전역 주기내 스테이징 작업의 순서를 결정한다. 전역 주기 스테이징 스케줄링에서 지역 주기는 서비스 중인 비디오의 요구 스테이징 대역폭을 패턴화 하고 전역 주기내에 들어갈 스테이징 작업의 갯수를 구하는 기준을 마련해 준다. 전체적인 스테이징 스케줄링 세부 과정은 다음과 같다.

전역 주기 단위의 스테이징 스케줄링을 하면 하나의

```

Staging_Scheduler () {
  // Global period staging scheduling
  for (all active streams under service) {
    calculate required_staging_bandwidth;
    map calculated_bandwidth
      to predetermined_staging_bandwidth;
    add staging job to staging_list; }
  for (all staging jobs in staging_list)
    calculate staging emergency;
    sort staging_list by staging emergency;
  // Staging Optimization
  calculate available_staging_bandwidth
    in the current period;
  calculate additional_number_of_stagings;
  predict the staging jobs in the next global period;
  for (additional_number_of_stagings)
    add predicted staging job to staging_list;
}

```

전역 주기 안에 필요한 스테이징 작업이 위급도에 따라 수행 순서가 정해지게 된다. 스테이징 스케줄링은 현 시점에서 안정된 서비스를 보장하기 위해 반드시 필요한 스테이징 작업만을 수행하게 된다. 따라서 전역 주기의 모든 스테이징 작업을 수행하더라도 그 다음 전역 주기가 시작되기 전까지 여분의 시간이 존재할 수 있다. Tertiary 저장 장치의 낮은 대역폭을 최대한 활용하기 위하여 전역 주기 단위의 여분의 시간에 추가의 스테이징 작업을 할당할 수 있다. 이러한 적극적인 스테이징 스케줄링 작업으로 Tertiary 저장 장치의 대역폭을 최대한 활용할 수 있고 추가의 스테이징 작업 수행으로 인한 해당 비디오의 요구 스테이징 대역폭을 낮출 수 있으므로 여유 스테이징 대역폭을 보다 많이 확보할 수 있다. 이러한 스테이징 최적화 기법으로는 스테이징 스케줄링이 수행중인 현재 전역 주기의 다음 주기를 예측하여 여분의 시간이 허락하는 동안 미리 앞당겨 수행하는 방법이 있다. 스테이징 작업 순서의 형태는 전역 주기 단위로 반복되므로 다음 전역 주기의 스테이징 작업을 예측하는 작업은 쉽다. 스테이징 최적화 작업을 통해 현재 전역 주기내에서의 스테이징 대역폭을 최대한 활용할 수 있다. 예측된 스테이징 작업을 미리 수행하므로 해당 비디오에 대한 요구 스테이징 대역폭이 낮아질 확률이 커지며 이로 인해 전체적인 여유 스테이징 대역폭이 커지게 되어 스테이징 최적화를 하지 않았을 때 보다 더 많은 사용자들을 수용할 수 있게 된다.

### 2.3 디스크 캐쉬

대용량 스트리밍 서비스를 위한 하드디스크와 Tertiary 저장장치간의 계층적 자료 이동을 위하여 본 논문에서는 하드디스크를 디스크 캐쉬로 사용한다. 디스크 캐쉬는 안

정적이고 연속적인 스트리밍 서비스를 위하여 적절히 스테이징되고 다양한 정책을 통하여 관리되어야 한다. 따라서 디스크 캐싱 시스템의 핵심은 사용자가 요구하는 미디어 스트림을 하드디스크 캐쉬내에 항상 유지하여 서비스의 연속성을 보장하는 것이다. 논문에서 제안하는 디스크 캐쉬는 고정 크기의 블록 단위로 관리되며, 캐쉬 블록의 크기는 Tertiary 저장장치의 대역폭 ( $B_{tertiary}$ ) 및 사용자의 최대 자료 요구율 ( $R_{max}$ )을 고려하여 식 6과 같이 결정된다. 여기서,  $T_{mount}$ 와  $T_{dismount}$ 는 각각 Tertiary 저장 장치의 미디어 loading 시간과 unloading 시간을 의미하며,  $N_{staging}$ 은 지역주기내에 수행 가능한 최대 스테이징의 갯수를 의미한다.

$$S_{cacheblock} = \frac{(T_{mount} + T_{dismount})(N_{staging} + 1)R_{max}}{B_{tertiary} - R_{max}(N_{staging} + 1)} \quad (6)$$

사용자가 요청한 자료가 디스크 캐쉬 내에 없고 요청된 자료가 있는 미디어가 드라이브 내에 없다면 Tertiary 저장장치내의 미디어 교환을 통해 해당 저장 미디어를 드라이브로 이동시키는 작업이 필요하다. 이러한 미디어 교체 시간은 사람의 시각을 통해 충분히 지각할 수 있는 시간이며 요청된 자료가 디스크에 캐쉬되지 않았을 경우 해당 사용자에 대한 서비스는 일시 중지 또는 종료되게 된다. 사용자의 서비스 요구에 대해 비디오의 초기 부분 부재로 인한 서비스 거부를 방지하기 위해 모든 비디오에 대해 비디오의 초기 부분 일정량을 선반입하여 보관한다. 선반입하는 자료의 양은 캐쉬 블록 크기의 정수 배로 한다. 선반입된 자료는 캐쉬 교체 정책에 의해 선택되어지지 않도록 하여 사용자의 서비스 요구에 대해 비디오의 초기 부분이 항상 디스크 캐쉬 안에 있음을 보장한다.

디스크 캐쉬 블록은 *None*, *Request*, *Caching*, *Prefetched*, *Cached*, *Using*, *Complete*의 7가지 상태를 갖는다. *None* 상태의 캐쉬 블록은 현재 어떠한 유효한 자료도 가지고 있지 않음을 의미한다. 디스크 캐쉬 내에 *None* 상태의 캐쉬 블록이 없을 경우는 디스크 캐쉬가 모두 사용되고 있음을 의미하며 이 시점부터 디스크 캐쉬 교체 정책이 적용된다. *Request* 상태의 캐쉬 블록은 해당 캐쉬 블록이 새로운 자료를 저장하기 위하여 예약되었음을 나타낸다. *Caching* 상태의 캐쉬 블록은 현재 Tertiary 저장장치에서 해당 디스크 캐쉬 블록으로 자료가 이동중임을 의미한다. *Prefetched* 상태의 캐쉬 블록은 Tertiary 저장장치에서 디스크 캐쉬 장치로 자료 이동이 종료되었음을 의미하는 동시에 캐쉬 교체 정책에 의해 선택되지 않음을 나타낸다. 이는 디스크 캐쉬 안에 있는 모든 비디오들의 초기 부분이 *Pre*

fetched 상태로 되어 항상 디스크 캐쉬 안에 있음을 보장하여 사용자의 서비스 요구시 비디오의 초기 부분 부재에 의한 서비스 일시 중지 혹은 종료를 방지하기 위함이다. *Cached* 상태의 캐쉬 블록은 Tertiary 저장장치에서 디스크 캐쉬로 자료 이동이 끝났음을 의미하며 사용자에게 의해 앞으로 사용될 블록임을 나타낸다. *Using* 상태의 캐쉬 블록은 사용자가 현재 사용 중인 캐쉬 블록임을 의미하고, *Complete* 상태의 캐쉬 블록은 사용자가 사용을 마친 캐쉬 블록임을 나타낸다. *Complete* 상태의 캐쉬 블록은 캐쉬 교체 정책에 의해 선택되어지는 유일한 블록이다. 그림 5는 현재 서비스중이지 않은 Inactive 상태의 비디오와 현재 서비스 중인 Active 상태의 비디오들에 대해 캐쉬 블록의 상태가 어떻게 전이 되는지를 보여주며 사용자의 서비스 요구, 종료 등에 따른 Active 비디오와 Inactive 간의 캐쉬 블록 상태변화가 어떻게 이루어지는지 보여주고 있다.

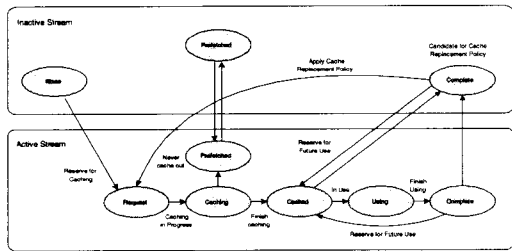


그림 5 캐쉬 블록의 상태 전이도

스트리밍을 고려한 캐쉬 교체 정책은 디스크 캐쉬에 저장된 비디오 데이터의 안정성과 연속성을 보장할 수 있어야 한다. 캐쉬 교체 정책은 현재 서비스 중이지 않은 Inactive 상태의 비디오를 대상으로 한다. Active 상태의 비디오를 캐쉬 교체 정책의 대상으로 할 경우 해당 비디오를 서비스 받고 있는 사용자에게 대한 서비스의 안전성을 보장하지 못하거나 이미 캐쉬되어 사용이 예약된 캐쉬 블록이 교체되므로 나중에 그 블록을 위한 추가의 스테이징 대역폭을 할당해야 한다. 이는 곧 스테이징 대역폭의 낭비를 가져온다.

캐쉬 교체 정책에 의해 희생 블록을 선택하는 과정은 다음과 같다. 현재 사용자에게 의해 서비스중이지 않은 Inactive 상태의 비디오를 대상으로 하여 검색을 시작한다. 이때 완전 캐쉬가 된 비디오는 제외한다. 완전 캐쉬가 된 비디오는 이전 사용자가 해당 비디오를 끝까지 재생하였음을 의미하며, 연속 미디어의 특성과 사용자의 선호도를 고려하였을 때 어떤 사용자가 끝까지 재생을 마친 비

디오는 다음 사용자가 같은 비디오를 서비스 요청했을 경우 끝까지 재생할 확률이 높다. 이러한 완전 캐쉬된 비디오들을 캐쉬 교체 대상에서 제외함으로써 차후에 발생할 수 있는 추가의 스테이징 대역폭 낭비를 방지하여 여유 스테이징 대역폭을 최대한 활용할 수 있도록 한다. 완전 캐쉬된 비디오를 제외한 Inactive 상태 비디오중 사용자에게 의해 사용이 완료된 *Complete* 상태의 캐쉬 블록이 있는 비디오를 선택하고, 캐쉬 거리를 계산한다. 캐쉬 거리 ( $T_{cachedistance}$ )는  $S_{cached(i)}/R_i$ 에 의해 계산되며 디스크 캐쉬에 저장된 자료의 양으로 재생 가능한 시간을 의미한다. 각각의 Inactive 상태 비디오중 *Complete* 상태의 캐쉬 블록이 있는 비디오의 캐쉬 거리가 가장 긴 비디오를 선택하여 해당 비디오에 대한 지막 *Complete* 상태의 캐쉬 블록을 희생자 블록으로 선택한다. 위의 방법으로 선택된 희생 블록이 없다면 현재 서비스 중이지 않은 Inactive 상태의 비디오를 대상으로 하여 완전 캐쉬된 비디오를 포함하여 *Complete* 상의 캐쉬 블록이 있는 비디오 중 캐쉬 거리가 가장 큰 비디오의 가장 마지막 캐쉬 블록을 희생 블록으로 선택하게 된다. 이러한 경우는 완전 캐쉬된 비디오의 마지막 *Complete* 상태의 블록이 우선적으로 선택되어지게 된다. 캐쉬 교체 정책에 의해 희생 블록을 선택하지 못했다면 이는 곧 해당 비디오에 대한 서비스 중지를 의미한다. 캐쉬 교체 정책에 의한 희생 블록을 선택하지 못하는 경우는 디스크 캐쉬의 크기가 충분하지 않을 경우 발생할 수 있으며 이러한 상황을 방지하기 위하여 디스크 캐쉬의 크기를 고려한 적절한 사용자 수용 제어 정책을 수행하여야 한다. 그림 6은 이러한 캐쉬 교체 정

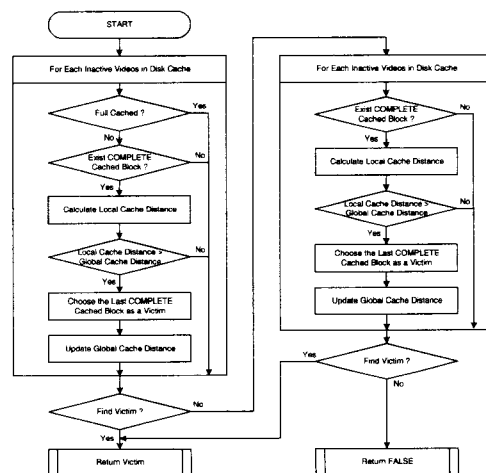


그림 6 캐쉬 교체 정책



책 과정을 상세히 보여준다.

#### 2.4 사용자 수용 제어

시스템의 성능을 초과하여 사용자를 수용하면 기존의 서비스중인 사용자에게 대한 서비스의 연속성과 안전성을 보장 할 수 없다. 그러므로 서비스중인 사용자의 QoS 보장을 위해 시스템 자원 사용 상황을 고려한 사용자 수용 제어 정책이 필요하다. 사용자 수용 제어기는 새로운 사용자 서비스 요청 요구가 있으면 현재 시스템 자원의 사용량을 기준으로 새로운 사용자를 위한 자원을 할당 할 수 있는지를 판단하여 사용자의 수용 여부를 결정하게 된다. 수용 제어 판단의 기준이 되는 시스템 자원은 디스크 대역폭, 네트워크 대역폭, 메모리 사용량이 있고 이외에 디스크 캐쉬량, 여유 스테이징 대역폭을 고려하여 사용자 수용 제어를 수행하며 각각 요소의 판단 방법은 다음과 같다.

시스템 자원 사용량에 근거한 사용자 수용 제어의 판단 방법은 전체 시스템 자원의 한도 내에서 새로운 사용자를 위한 추가의 시스템 자원을 할당 할 수 있는지 검사한다. 사용자 수용 제어기에서 수용 제어의 기준으로 사용하는 시스템 자원으로는 디스크 대역폭과 네트워크 대역폭, 주 메모리 사용량이 있다. 메모리 사용량은 서버 초기 실행시 설정된 값으로 전체 주 메모리의 일부를 할당받아 수용 제어가 허락된 사용자에게 일정량을 분배하며 식 7를 만족해야 한다. 즉, 시스템에 할당된 메모리의 양( $M_{system}$ )이 현재 서비스중인 사용자들이 사용하고 있는 메모리의 사용량( $\sum_{i=1}^n M_i$ )과 새로운 사용자를 위한 메모리 사용량( $M_{(i+1)}$ )의 합보다 크거나 같아야 한다. 서비스중인 사용자의 데이터의 합( $\sum_{i=1}^n R_i$ )과 새로운 사용자가 사용할 데이터의 합이 시스템의 디스크 대역폭 ( $B_{disk}$ )과 네트워크 대역폭 ( $B_{net}$ ) 중 최소값을 넘지 않아야 하며 식 8의 조건을 만족해야 한다. 시스템 자원은 자원 관리 서버 시스템의 시스템 자원 관리자에 의해 관리된다. 서비스 중인 사용자가 사용하는 자원의 양은 입출력 관리 서버 시스템의 네트워크 관리자에 의해 동적으로 측정되며 자원 관리 서버 시스템의 사용자 관리자에 의해 관리된다.

$$M_{system} \geq \sum_{i=1}^n M_i + M_{(i+1)} \quad (7)$$

$$\min(B_{disk}, B_{net}) \geq \sum_{i=1}^n R_i + R_{(i+1)} \quad (8)$$

디스크 캐쉬량에 근거한 사용자 수용 제어의 판단 방법은 디스크 캐쉬 안에 새로운 사용자를 위해 할당할 수 있는 디스크 캐쉬 공간이 충분히 있는지를 검사한다. 디스크 캐쉬 교체 정책에 의해 현재 디스크 캐쉬에서

선택 가능한 블록의 갯수 ( $N_{complete}$ )가 현재 서비스중인 사용자가 서비스를 완료할 때까지 필요한 디스크캐쉬 블록 갯수의 합( $\sum_{i=1}^n N_{caching(i)}$ )과 새로운 사용자를 위해 필요한 디스크 캐쉬 블록의 개수 ( $N_{caching(i+1)}$ )의 합보다 크거나 같아야 하며 식 9의 조건을 만족해야 한다. 이는 새로운 사용자를 위하여 디스크 캐쉬 공간을 할당하는 것을 의미하며 수용이 허락된 사용자는 비디오의 끝까지 서비스 받을 동안 해당 비디오를 위한 디스크 캐쉬 공간이 충분히 있음을 보장받는다.

$$N_{complete} \geq \sum_{i=1}^n N_{caching(i)} + N_{caching(i+1)} \quad (9)$$

스테이징 대역폭에 근거한 사용자 수용 제어의 판단 방법은 새로운 사용자를 위한 여분의 스테이징 대역폭 캐쉬로의 스테이징 대역폭 ( $B_{tertiary}$ )은 현재 스테이징 작업중인 비디오들이 사용중인 스테이징 대역폭의 합 ( $R\sum_{i=1}^n R_{staging(i)}$ ) 과 새로운 사용자가 사용할 스테이징 대역폭 ( $R_{staging(i+1)}$ )의 합보다 커야 하며 식 10을 만족해야 한다.

$$B_{tertiary} \geq \sum_{i=1}^n R_{staging(i)} + R_{staging(i+1)} \quad (10)$$

$$T_{global} \geq \sum_{i=1}^n T_{staging(i)} + T_{mount} + T_{dismount} + T_{moveblock} \quad (11)$$

스테이징 대역폭이 허용하더라도 전역 주기 내에 새로운 사용자를 위한 스테이징 작업을 할당할 수 있어야 하며 식 11의 조건을 만족해야 한다. 즉, 현재 전역 주기 내에 스테이징 작업을 수행하는데 걸리는 시간( $\sum_{i=1}^n T_{staging(i)}$ )과 새로운 사용자를 위한 스테이징 시간( $T_{mount} + T_{dismount} + T_{moveblock}$ )의 합이 전역 주기( $T_{global}$ )를 넘지 않아야 한다. 스테이징 대역폭에 대한 정보는 자원 관리 서버 시스템에 있는 스테이징 관리자에 의해 관리된다. Tertiary 저장 장치의 로봇암을 움직이는 시간과 자료 이동 시간은 동적으로 측정되며 이를 스테이징 스케줄링에 반영하여 시스템 자원을 최대한 활용할 수 있도록 한다. 스테이징 관리자는 스테이징 스케줄러에서의 여유 스테이징 대역폭을 계산하고 관리하며, 스테이징 최적화 모듈에서 사용할 수 있도록 한다. 또한, 전역 주기 단위의 여유 스테이징 대역폭도 관리하여 여유 스테이징 대역폭 한도내의 추가의 스테이징 작업을 가능하게 한다.

### 3. 성능 분석

제안된 비디오 서버 시스템은 광주크박스 저장장치를 장착한 범용 플랫폼에서 구현되었다. 다음은 구현된 시스템을 실제 환경에서 실험하고 분석한 결과를 기술한다. 또한, 제안한 여러 정책들을 실험적으로 검증하고

구현된 시스템의 성능 특성을 파악한다.

3.1 실험 환경

실험 시스템은 100BaseT 이더넷을 기반으로 하는 지역 네트워크에 연결된 서버와 다수개의 사용자 시스템으로 구성된다. 실험에 사용된 Tertiary 저장 장치는 Hewlett Packard 사의 Hp-80fx 광주크박스 (Magnetooptical Jukebox)로 16 개의 슬롯, 1개의 드라이브, 그리고 1 개의 로봇암으로 구성된다. 서버는 Windows 2000 운영체제를 수행하는 Pentium-III 550 시스템이고 디스크 캐싱을 위한 충분한 크기의 하드디스크 저장 장치를 가지고 있다. 실험에 사용된 미디어 데이터는 MPEG-1 표준으로 인코딩된 파일이다.

그림 7과 그림 8은 구현된 서버가 작동하는 모습을 보여준다. 그림 7은 스테이징 모니터 화면으로 최종 단계의 스테이징 스케줄 결과를 보여주며, 그림 8은 캐쉬 모니터 화면으로 현재 디스크 캐쉬 블럭의 상태를 시각적으로 보여주어 캐쉬 교체 정책이 제대로 작동되고 있는지 알 수 있다.

실험의 주 목적은 구현된 시스템의 성능 특성을 밝히는 것이다. 특히, 서버의 스트림 스케줄링, 디스크 캐싱, 사용자 수용 제어 정책들을 실험적으로 검증한다. 실험

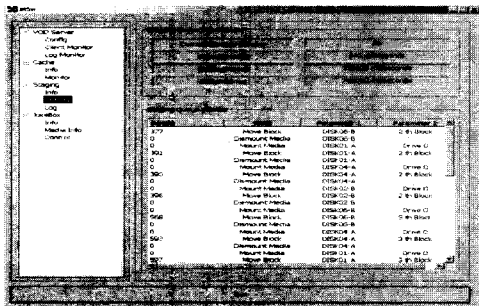


그림 7 구현된 서버의 스테이징 모니터 화면

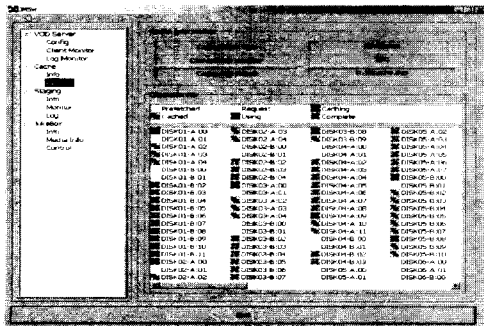


그림 8 구현된 서버의 캐쉬 모니터 화면

에 사용한 비디오 종류는 30개이며 다수의 사용자들이 1분 간격으로 비디오 재생을 연속적으로 요구한다. 사용자의 비디오 선호도에 따른 서버의 성능 특성을 분석하기 위하여 사용자의 비디오 선호도가 Zipf 분포를 따른다고 가정하였다[13]. Zipf 분포는 0과 1 사이의 값을 갖는  $\theta$  변수 값을 가지며  $\theta$ 가 1이면 균일 분포를 이루고, 값이 0에 근접할수록 편중 분포가 된다. 실험에서는 세가지  $\theta$  값 (0.0, 0.5, 1.0)을 사용하였다. 한편, 디스크 캐쉬 크기에 따른 서버의 성능 특성을 파악하기 위하여 6GB와 12GB 크기의 하드디스크 캐쉬로 캐쉬 크기를 변경하면서 실험을 하였다. 그림 9는 실험에 사용된 Zipf 분포에 따른 사용자들의 비디오 선호도 분포를 보여주며, 표 1은 이밖의 각종 실험 환경값을 보여준다.

표 1 실험 환경 변수 값

Maximum media rate ( $R_{max}$ )	200(KB/sec)
$S_{cacheblock}$	30(MB)
Total cache blocks	200
Prefetched media per video	60(MB)
Number of Videos	30
Video length	600(MB)
$T_{mount}$	10(sec)
$T_{dismount}$	5(sec)
Tertiary transfer bandwidth( $B_{terian}$ )	6(MB/sec)
$T_{system}$	640(us)
$T_{local}$	153,600(us)
$T_{global}$	768,000(us)
Number of staging patterns( $N_{pattern}$ )	5
Maximum tertiary access streams( $N_{streams}$ )	5

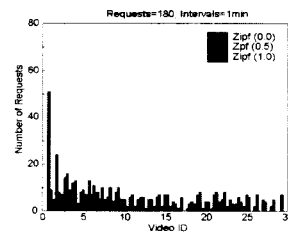


그림 9 사용자 비디오 선호도

3.2 결과 분석

서버의 성능 특성은 다양한 성능 지표를 기준으로 분석된다. 특히, 서버의 최대 수용 가능 스트림 수, 광주크박스 저장장치의 가용 대역폭 트레이스, 그리고 스테이징 작업의 마감 시간 트레이스등을 중점적으로 실험하고 분석하였다.

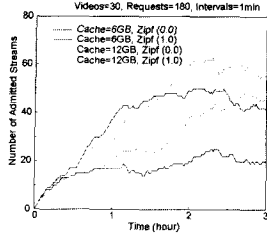


그림 10 서비스 중인 스트림 수

그림 10은 상이한 사용자 선호도와 캐쉬 크기에 따른 3시간 동안의 트래이스 결과로 서비스가 허용된 Active 스트림의 수의 변화를 보여준다. 그림에서 보듯이 편중 분포를 갖으며 캐쉬 디스크가 커짐에 따라 사용자 요구의 수용이 증가됨을 알 수 있다. 즉, 비디오 요구가 편중될수록 캐쉬 적중률이 높아지고 이에 따라 스테이징 대역폭이 확보되어 추가의 사용자 요구가 수용될 수 있다. 이러한 상황에서 디스크 캐쉬의 증가는 캐쉬 적중률의 더욱 증가시킨다.

표 2 스트림 수용 제어 분석

Zipf ( $\theta$ )	캐쉬 크기	요청 횟수	허용 횟수	허용 비율	거절 횟수	스테이징 대역폭 부족	캐쉬 부족	시스템 자원부족
0.0	6GB	180	114	63.3%	66	16	50	0
	12GB	180	134	74.4%	46	33	0	13
0.5	6GB	180	60	33.3%	120	13	107	0
	12GB	180	96	53.3%	84	82	2	0
1.0	6GB	180	54	30.0%	126	17	109	0
	12GB	180	92	51.1%	88	82	6	0

표 2는 전체 사용자 서비스 요청 횟수와 서버의 사용자 수용 제어를 통해 실제 서비스 받은 사용자의 수, 서비스 요구가 거절당한 사용자의 수 그리고 서비스 거절의 근거를 보여준다. 표에서 알 수 있듯이 사용자의 비디오 선호도가 편중 분포되었을 때 상대적으로 보다 많은 사용자가 서비스를 받는다. 또한 디스크 캐쉬의 양이 많을수록 서비스 스트림의 수가 증가함을 알 수 있으며,

디스크 캐쉬의 크기와 사용자의 비디오 선호도가 서버 성능의 중요한 변수임을 확인할 수 있다. 서비스 요구 거절의 이유는 스테이징 대역폭 부족, 캐쉬 부족, 그리고 시스템 자원 부족으로 구분된다. 표에서 보듯이, 디스크 캐쉬의 양이 적을 경우 (6GB) 여유 스테이징 대역폭 부족으로 인한 서비스 거부는 서버 초기 실행시 디스크 캐쉬의 캐쉬 되어 있는 자료량 부족으로 인한 비디오의 요구 스테이징 대역폭이 높아져있기 때문이다. 서버 초기 실행시 짧은 시간대에 요구 스테이징 대역폭이 높은 비디오를 다수의 사용자가 요청하므로 여유 스테이징 대역폭 부족으로 인한 서비스 거부가 발생하게 된다. 시간이 지남에 따라 디스크 캐쉬에 비디오가 풀고 루 캐쉬 되고 각각의 비디오에 대한 요구 스테이징 대역폭이 낮아져 전체적인 여유 스테이징 대역폭이 높아 지므로 여유 스테이징 대역폭 부족으로 인한 서비스 거부가 줄어들게 된다. 반면, 디스크 캐쉬의 양이 커졌을 경우 (12GB), 대부분의 서비스 거부가 캐쉬 양의 부족으로 인한 서비스 거부보다는 여유 스테이징 대역폭 부족으로 인한 서비스 거부임을 알 수 있다. 즉, 광주크박스의 전송 대역폭 한계가 서비스 거부의 주 요인이 된다. 시스템 자원 부족으로 인한 서비스 거부는 디스크 대역폭, 네트워크 대역폭, 메모리 사용량등의 시스템 자원부족으로 인한 서비스 거부를 의미하며 실험 시작후 2시간이 경과했을때 동시 사용자 수가 많아서 네트워크 자원 대역폭 부족할 때 중점적으로 발생하였다.

그림 11은 사용자의 비디오 선호도가 편중 분포되어 있을 경우의 디스크 캐쉬 크기에 따른 여유 스테이징 대역폭의 변화를 보여주고 그림 12는 사용자의 비디오 선호도가 균일 분포되어 있을 경우의 디스크 캐쉬 크기에 따른 여유 스테이징 대역폭의 변화를 보여준다.

서비스 초기에는 디스크 캐쉬 안에 캐쉬 되어 있는 미디어가 적은 반면 서비스 요청된 비디오의 요구 스테이징 대역폭이 높으므로 여유 스테이징 대역폭이 급격히 감소한다. 서비스가 진행됨에 따라 디스크 캐쉬안의 비디오들이 캐쉬 적중되고 이에 따라 서비스를 요구하는 비디오의

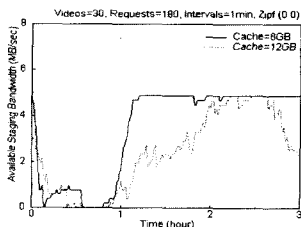


그림 11 스테이징 여유 대역폭 변화 (Zipf 0.0)

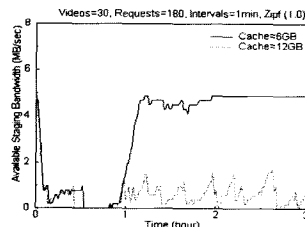


그림 12 스테이징 여유 대역폭 변화 (Zipf 1.0)

요구 스테이징 대역폭이 상대적으로 낮아 저 여유 스테이징 대역폭의 감소 정도가 적어진다. 이때, 디스크 캐쉬의 크기가 적을 경우(6GB)는 사용자 수용 제어 정책으로 인해 새로운 비디오를 요청하는 사용자의 요구를 수용하지 못하고 여유 스테이징 대역폭을 현재 서비스 중인 비디오들에게 할당함으로써 일정 수준의 여유 스테이징 대역폭이 유지됨을 확인 할 수 있다. 반면, 디스크 캐쉬의 크기가 클 경우(12GB), 편중 분포에 의한 캐쉬 적중률이 커짐에 따라 여유 스테이징 대역폭이 점차 증가함을 알 수 있다. 그림 12는 사용자 의 비디오 선호도가 균일 분포되어 있을 경우 여유 스테이징 대역폭의 변화를 보여준다. 사용자의 비디오 선호도가 균일 분포이므로 디스크 캐쉬의 캐쉬 적중률이 상대적으로 낮다. 그림 11과 마찬가지로 디스크 캐쉬의 크기가 작을 경우(6GB)는 현재 서비스 중인 비디오들의 서비스 안정성을 위해 디스크 캐쉬가 사용되고 새로운 사용자의 요구가 거절되어 여유 스테이징 대역폭이 일정 수준의 대역폭을 유지한다. 그러나, 디스크 캐쉬 크기가 클 경우(12GB), 균일 분포에 의한 사용자 요구가 캐쉬 적중률을 감소시키며 따라서 여유 스테이징 대역폭이 전반적으로 낮아진다. 이로써 디스크 캐쉬 정책이 여유 스테이징 작업을 확보하고 연속 미디어의 특성을 잘 반영하여 서비스의 안정성을 보장 해준다는 것을 확인 할 수 있으며 사용자 수용 제어 정책이 제대로 동작함을 알 수 있다.

그림 13은 사용자의 비디오 선호도가 편중 분포되어 있을 경우, 그림 14는 균일 분포되어 있을 경우에 각각 디스크 캐쉬 크기에 따른 스테이징 작업의 마감 시간의 변화를 보여 준다. 그림에서 보듯이, 모든 스테이징 작업이 마감 시간을 만족하여 서비스의 안정성이 보장되고 스테이징 스케줄러가 올바르게 동작하고 있음을 확인 할 수 있다. 서비스 초기에는 디스크 캐쉬 안에 캐쉬 되어 있는 자료가 적기 때문에 상대적으로 서비스 요청된 비디오에 대한 스테이징 작업의 마감 시간이 짧으나, 서비스가 진행됨에 따라 미디어 캐싱이 발생하여 스테이징 작업의 마감 시간이 증가한다. 이때 디스크 캐쉬의 크기에 따라 마감 시간의 변화가 다르게 된다. 이 실험을 통하여 스테이징 스케줄러와 사용자 수용 제

어 정책이 기존 스트림에 대한 서비스의 안정성을 보장하며 올바르게 동작하고 있음을 확인 할 수 있다.

#### 4. 결론

본 논문에서는 광주크박스 저장 장치를 사용한 비디오 서버를 설계하고 구현하였다. 대용량의 저장 공간을 제공하지만 입출력 대역폭이 낮은 광주크박스 저장 장치를 비디오 서버에 사용하기 위하여 디스크 저장 장치를 캐쉬 장치로 사용하는 디스크 캐쉬 모델을 사용하였다. 또한 광주크박스 저장 장치를 효율적으로 활용하기 위하여 새로운 입출력 스케줄링 정책과, 연속 미디어의 특성을 고려한 디스크 캐쉬 관리 정책, 적절한 사용자 수용 제어 정책을 제시하고 구현하였다.

기존의 연구는 스테이징 스케줄링, 디스크 캐쉬등의 일부 지역적인 정책의 연구에 국한되어 있으며 연구의 검증이 모의 실험 등을 통해서만 이루어져 실제 시스템에 적용하기에는 여러 가지 어려움이 있었다. 본 논문에서는 Tertiary 저장 장치를 대용량 비디오 서버에 적용하기 위한 여러 가지 정책들을 새로이 제시하고 이들을 종합하여 하나의 작동하는 시스템을 구현하였다. 이로써 실제 상황에서 발생할 수 있는 여러 문제점들을 파악하고 구현된 서버의 성능을 평가하며 서버에 사용된 여러 정책들이 올바르게 동작함을 검증하였다. 서버 성능의 중요한 시스템 변수로는 Tertiary 저장 장치의 미디어 교체 시간, 입출력 대역폭, 디스크 캐쉬의 크기 등이 있다. 이러한 변수들은 시스템 의존적이며 변경에 제한이 있다. 실험을 통해 디스크 캐쉬 크기와 같은 시스템 변수와 사용자의 시간대 선호도나 비디오 선호도와 같은 시스템 외적 변수가 서버 성능의 중요한 변수임을 알 수 있었다.

향후 연구 과제로는 현재의 디스크 캐쉬 교체 정책이 서비스중인 스트림을 우선으로 하여 미디어의 연속성을 보장하는 반면, 한정된 디스크 캐쉬 공간이 몇 개의 서비스중인 스트림에 모두 할당될 수 있다는 단점이 있는 점을 보완할 새로운 방법의 연구가 있다. 또한, 다중 드라이브를 지원하기 위한 새로운 입출력 스케줄링 정책,

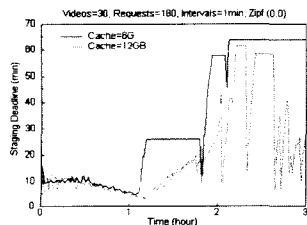


그림 13 스테이징 작업마감시간(Zipf 0.0)

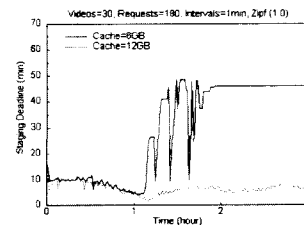


그림 14 스테이징 작업마감시간(Zipf 1.0)

스태이징 작업의 최적화를 위한 Tertiary 저장 장치의 미디어 교체 횟수를 감소시키는 방법, 확보된 여유 스테이징 대역폭을 지속적으로 유지할 수 있는 방법, 단순 재생이 아닌 VCR 기능 지원을 위한 스테이징 정책과 디스크 캐쉬 관리 정책의 개발이 필요하다.

### 참 고 문 헌

- [1] T.D.C. Little, D. Venkatesh, "Prospects for Interactive Video-on-Demand," *IEEE Multimedia*, Fall 1994, Vol. 1, No. 3, pp. 14-24.
- [2] Shahram Ghandeharizadeh, Cyrus Shahabi, "On Multimedia Repositories, Personal Computers, and Hierarchical Storage Systems," *ACM Multimedia*, San Francisco, California, October 1994, pp. 407-416.
- [3] Shung-Han Chan, Fouad A. Tobagi, "Hierarchical Storage Systems for On-Demand Video Servers," *Proceedings of the SPIE*, 1996, Vol. 2604, pp. 103-120.
- [4] Kian Lee Tan, Beng Chin Ooi, Tat Seng Chua, "On Video-On-Demand Servers with Hierarchical Storage," *The 5th International Conference on Database Systems for Advanced Applications*, Melbourne, Australia, April 1997, pp. 491-500.
- [5] Peter Triantafyllou, Thomas Papadakis, "On-Demand Data Elevation in a Hierarchical Multimedia Storage Server," *The 23th International Conference on Very Large Data Bases*, Athens, Greece, August 1997, pp. 226-235.
- [6] Siu-Wah Lau, John C.S. Lui, "Designing a Hierarchical Multimedia Storage Server," *The Computer Journal*, November 1997, Vol. 40, No. 9, pp. 529-540.
- [7] Sung Bae Jun, Won Suk Lee, "Video Allocation Methods in a Multi-level Server for Latge-scale VOD Services," *IEEE Transactions on Consumer Electronics*, Vol. 44, No. 4, November 1998, pp. 1309-1318.
- [8] Tatsuo Mori, Hideharu Suzuki, Kazutoshi Nishimura, Hiroataka Nakano, "Playback Techniques for a Video-on-Demand System Using an Optical Mass Storage System," *Japanese Journal of Applied Physics*, Vol. 35, Part. 1, No. 1B, January 1996, pp. 495-499.
- [9] Shahram Ghandeharizadeh, Ali Dashti, Cyrus Shahabi, "Pipelining Mechanism to Minimize The Latency Time in Hierarchical Multimedia Storage Manager," *Computer Communications*, Vol. 18, No. 3, March 1995, pp. 170-184.
- [10] Makoto Mizukami, Shigetaro Iwatsu, Nobuyoshi Izawa, "Real-Time Staging in Optical Disk Library," *Japanese Journal of Applied Physics*, Part. 1, No. 1B, January 1997, pp. 568-571.
- [11] S. Prabhaker, D. Arrawal, A. El Abbadi, A. Singh, "Efficient I/O Scheduling in Tertiary Libraries," *Technical Reports*, Computer Science Department, University of Columbia, TRCS96-26, October 1996.
- [12] Jihad Boulos, Kinji Ono, "VOD Data Storage in Multimedia Environments," *IEICE TRANS. COMMUN*, Special Issue on Multimedia Communications in Heterogeneous Network Environments, Vol. E81-B, No. 8, August 1998, pp. 1656-1665.
- [13] L. BresLau, P. Cao, L. Fan, Gl. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and Implications," *Proceedings of IEEE infocom*, New York, March 1999, pp. 126-134.



이 종 민

1999년 광운대학교 컴퓨터과학 학사.  
2001년 광운대학교 컴퓨터과학 석사.  
2002년 현재 연세대학교 컴퓨터과학과 박사과정. 관심분야는 멀티미디어 시스템, 운영체제, 컴퓨터 네트워크



차 호 정

1985년 서울대학교 컴퓨터공학 학사.  
1987년 서울대학교 컴퓨터공학 석사.  
1991년 영국 University of Manchester 전산학 박사. 1993년~2001년 광운대학교 컴퓨터과학과 부교수. 2001년~현재 연세대학교 컴퓨터과학과 부교수. 관심분야는 멀티미디어 시스템, 운영체제, 분산 컴퓨팅



오 재 학

1997년 광운대학교 컴퓨터과학 학사.  
1999년 광운대학교 컴퓨터과학 석사.  
2002년 현재 광운대학교 컴퓨터과학과 박사과정. 관심분야는 멀티미디어 시스템, 실시간 시스템, 대용량 저장장치



박 병 준

1984 서울대학교 컴퓨터공학 학사. 1988 미국 미네소타대학교 컴퓨터과학 석사. 1997 미국 일리노이대학교(UIUC) 컴퓨터과학 박사. 미국 Epic Systems, SPSS 등에서 연구원으로 활동하였으며, 현재 광운대학교 컴퓨터과학과 교수. 관심분야는 인공지능, 데이터마이닝, 기계학습 등