

# Database를 이용한 VOD 버퍼 스케줄러의 설계 및 구현

이영주<sup>o</sup>  
창원대학교  
p2smf@kyungnam.ac.kr

이수현  
전자계산학과  
suhyun@sarim.changwon.ac.kr

## Design and Implementation of a VOD Buffer Scheduler Using Database

Young-Joo Lee<sup>o</sup> Su-Hyun Lee  
Dept. of Computer Science, Changwon National University

### 요약

VBR로 압축된 멀티미디어 데이터는 비트 변화율이 매우 큰 편차로 변하기 때문에 자원 예약관리가 매우 힘들다. 따라서 MPEG데이터의 참조 패턴을 오프라인으로 분석한 메타 테이블을 근거로 과부하 구간에서 미리 선반입하여 시스템의 활용율을 높이는 선반입 기법을 사용한다. 그러나 기존의 선반입 기법은 상영실패는 발생하지 않으나, 과부하 구간에서 선반입 임계 슬롯의 증가로 인하여 버퍼 점유량이 증가하며 이는 선반입 기법에서 가장 중요한 평가 요소인 최소의 적재시간과 적재비용을 유지할 수 없다. 따라서 본 논문에서는 선반입 구간을 고정적으로 사용하여, 적재비용과 적재시간을 일정한 범위 이하로 유지하면서 시스템 자원의 활용율을 높이는 방법을 제안한다. 아울러 VOD서버에서 수백편의 영화를 저장하여 다수의 사용자에게 서비스하는 상황에서 선반입 기법을 사용할 경우, 메타 테이블의 관리가 힘들어지는데 본 논문에서는 메타 테이블과 버퍼 스케줄러를 데이터베이스를 이용함으로써 메타 테이블의 관리를 편리하게 하였다.

### 1. 서론

멀티미디어 시스템의 일반적인 형태는 멀티미디어 데이터를 저장하고 있는 저장서버와 저장서버에게 원하는 데이터를 요구하는 사용자 시스템 및 이들을 연결하는 통신망으로 구성되는데, 이 중 저장서버가 멀티미디어 시스템의 성능에 큰 영향을 끼친다. 따라서 멀티미디어 데이터를 저장하고 있는 저장 서버의 효율적인 데이터 관리를 위한 많은 연구들이 진행되고 있다.[1],[2]

멀티미디어 시스템에서는 데이터의 전달 경로 상에 필요한 자원들을 예약함으로써 연속매체의 실시간 처리를 보장할 수 있다. 그러나 MPEG과 같은 압축기법에 의해 가변 비트율로 압축된 데이터들은 데이터 내용의 변화로 인해 단위 시간마다 재생해야 할 프레임의 크기가 일정하지 않고 데이터 량의 가변성으로 인해 VOD서버에서의 자원 예약관리를 매우 어렵게 한다.

이러한 문제를 해결하기 위하여 MPEG 데이터의 참조 패턴을 오프라인으로 분석한 메타 테이블을 근

거로 과부하가 발생할 경우, 미리 선반입하는 기법을 사용하는데 기존의 선반입 기법은 과부하 구간에서 선반입 임계 슬롯의 증가로 인하여 버퍼 점유량이 증가하며 이는 선반입 기법에서 가장 중요한 평가 요소인 최소의 적재시간과 적재비용을 유지할 수 없게 된다.

따라서 본 논문에서는 VOD서버에서 연속 매체의 실시간 처리 조건을 만족하면서 시스템 자원을 최대한 활용하여 더 많은 사용자를 수용할 수 있는 고정 선반입 스케줄러를 제안하고, 선반입 기법에서 필요로 하는 메타 테이블과 버퍼 스케줄러를 데이터베이스를 이용하여 구현하였다.

### 2. 관련연구

#### 2.1 선반입 정책

서버의 버퍼에 데이터 적재 및 예약 작업은 일정한 간격인 슬롯(Slot)으로 나누어 수행되며, 한 슬롯 동안에 다수의 프레임으로 구성된 GOP의 단위로 적재한다. 선반입 정책에서는 이러한 데이터의 적재

를 오프라인으로 분석된 MPEG 데이터 참조 패턴을 이용하여 디스크의 대역폭 ( $D_{min}$ )에 의한 사용자 제한을 서버에 의한 제한으로 전환함으로써 더 많은 사용자를 수용할 수 있다. 그리고 데이터 요구량이 디스크 대역폭을 초과하는 과부하 슬롯이 발생할 경우, 데이터 초과량을 이전 슬롯에서 여분의 디스크 대역폭을 이용하여 미리 서버의 버퍼로 읽어 둘 수 있다. 또한 선반입 되는 데이터 블록들은 최소한의 시간동안만 적재되어 있도록 선반입 되어야 한다.

따라서 과부하 슬롯의 데이터 초과량을 최소한의 시간동안 버퍼에 적재하기 위한 선반입 시작 슬롯 ( $\psi$ )은 식(1)을 이용하여 구한다. 슬롯  $t(a \leq t \leq \omega)$ 가 영역  $[a, \omega]$ 에서의 과부하 슬롯일 때 식(1)을 만족하는 최대의  $\psi (\leq t)$ 를 과부하 슬롯  $t$ 의 선반입 시작 슬롯이라 한다.[1]

$$(t - \psi + 1) \times D_{min} \geq \sum_{\tau=\psi}^t \alpha(\tau) \dots\dots\dots (1)$$

### 2.2 Drop정책

일반적으로 사용자들은 연속적인 미디어 스트림들의 내재된 중복성과 인간 지각상의 오차문제로 인하여 짧은 왜곡현상이나 간혹 발생하는 정보의 손실에 대해서는 내성을 지니고 있다.[5] 이러한 특성을 이용하여 과부하 구간에서 의도적인 미디어 Drop을 수행하는 스케줄러가 있다.

ACME(Abstraction for Continuous MEdia)는 캘리포니아 대학에서 제안한 오디오와 비디오 입출력을 제어하는 네트워크 서버로써, 시간 제약을 어기는 경우 전송될 프레임의 일부를 스킵핑(Skipping) 혹은 퍼징(Pausing)하는 기법을 사용하고 있다.[3]

HeiTS(Heidelberg Transport System)는 IBM 유럽 네트워킹 센터에서 개발한 멀티미디어 시스템으로 현재의 네트워크 상태를 파악하여 유용한 대역폭으로 트래픽을 적응시키는 네트워크의 응답성에 중점을 두고 있다. 따라서 수신측 시스템의 자원이 과부하 상태로 임계영역에 도달했다면 서비스를 중단하는 것이 아니라 미디어 스트림의 질을 점진적으로 감소시켜 조절함으로써, 사용자에게 의미있는 미디어 스트림을 제공해 줄 수 있다.[4]

### 2.3 승인제어 정책

새 사용자의 수용여부를 결정하는 승인제어 방법에는 결정적인 방법, 관측에 기반한 방법과 통계적인 방법이 있다. 결정적인 승인제어 방법은 서비스

질을 완벽하게 보장해 주지만, VOD시스템의 자원을 효율적으로 사용하지 못하는 단점이 있으며, 관측에 기반한 승인제어 방법은 시스템의 자원을 효율적으로 사용할 수 있지만, 서비스 질을 보장할 수 없다.

한편 통계적인 승인제어는 결정적인 승인제어 방법과 관측에 기반한 승인제어와의 절충적인 효과를 나타내지만, 결정적인 승인제어 방법에 비해 미디어 스트림의 일부에 대한 상영실패가 나타날 수 있다. 이러한 상영실패는 인간이 감지할 수 없을 정도의 일시적인 성능 저하로써 무리없이 사용자에게 이해되어 질 수가 있을 것이다.

통계적인 승인 제어 기법에서는 새로운 상영이 요청될 때, 그 상영이 평균적으로 요구할 버퍼 요구량의 적정  $\alpha$ 를 결정하여 기존의 수용된 사용의 예측 버퍼 요구량의 합과  $\alpha$ 의 총합이 시스템의 버퍼 크기를 넘지 않을 때 요청된 상영은 승인되는 것이다. 여기서는 단순한 합과 비교 연산만 있으므로 알고리즘의 자체 복잡도는  $O(1)$ 이 되는 것이다.[2]

## 3. 고정 선반입 VOD 버퍼 스케줄러 구현

### 3.1 시스템 구조

본 논문에서 구현한 고정 선반입 VOD 버퍼 스케줄러의 구성은 그림 1과 같다.

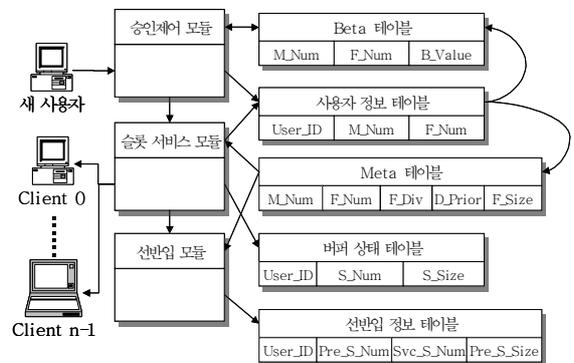


그림 1. 고정 선반입 VOD버퍼 스케줄러의 구성

각 테이블에서 사용한 필드에 대한 설명은 표 1과 같다.

본 논문에서 사용한 테이블은 사용자 정보 테이블, 메타 테이블, 베타 테이블, 버퍼 상태 테이블, 선반입 정보 테이블이다. 사용자 정보 테이블은 승인제어 모듈에서 승인한 사용자들의 사용자 번호, 영화번호, 요구한 프레임 번호를 저장하고, 슬롯 서비스 모듈에서 슬롯마다 갱신된다. 메타 테이블은 서비스 할 각 영화에 대해 영화 번호, 프레임 번호, 프레임 구분, 상영 실패시 이용하는 상영실패 우선순

표 2. 각 테이블의 필드 설명

필드 명	설명	필드 명	설명
M_Num	영화번호	F_Num	프레임 번호
F_Div	프레임 구분	D_Prior	Drop 우선 순위
F_Size	프레임 크기	User_ID	사용자 번호
B_Value	β값	F_Num	프레임 번호
S_Num	슬롯번호	S_Size	슬롯별 버퍼 요구양
Svc_S_Num	서비스 슬롯번호	Pre_S_Size	슬롯별 선반입 양
Pre_S_Num	선반입 슬롯		

위, 프레임 크기가 오프라인으로 분석하여 저장되어 있다. 베타 테이블은 영화 번호, 프레임 번호, 베타 값이 오프라인으로 분석되어 저장되어 있으며 승인 제어 모듈에서 사용한다. 버퍼 상태 테이블에는 각 슬롯에서 사용자에게 서비스될 슬롯 번호, 사용자 번호, 요구량이 저장되어 있으며 슬롯 서비스 모듈에서 슬롯마다 자료를 갱신한다. 선반입 정보 테이블에는 과부하 슬롯 발생시 선반입 해야 할 정보가 기록되며 선반입 슬롯 번호, 선반입된 데이터가 서비스 될 슬롯 번호, 사용자 번호, 선반입량이 저장되어 있으며 선반입 모듈에서 기록을 한다.

3.2 슬롯 서비스 모듈

슬롯 서비스 모듈은 본 논문에서 구현된 VOD서버의 버퍼 스케줄러 모듈 중 메인에 해당된다. 이 모듈은 매 슬롯마다 수행되며 사용자 정보 테이블, 메타 테이블, 버퍼 상태 테이블을 사용한다.

슬롯 서비스 모듈은 데이터 베이스에 연결된 후, 사용자 정보 테이블에서 승인된 사용자  $C_i$ 가 요구한 영화  $M_i$ 의 프레임  $F_i$ 에 대한 대상 프레임 정보 ( $\gamma$ )를 읽어온다. 읽어온 정보를 이용하여 메타 테이블에서 식(2)의 범위에 있는 프레임의 메타 데이터를 읽어온다. 식(2)의  $k$ 는 한 슬롯에서 읽을 GOP의 수로써 본 논문에서는 시스템 파라미터로 사용한다.

$$F_i \leq \gamma \leq F_i + (GOP \times k) - 1 \quad (k=2,4,6 \dots) \dots\dots (2)$$

식(2)를 이용하여 읽은 메타 데이터 합은 사용자  $C_i$ 가 현재 슬롯  $t$ 에서 시스템 파라미터로 정의된 고정 선반입 구간 ( $\psi$ )를 더한 슬롯  $t+\psi$ 에서 요구한 데이터량이다. 모든 사용자들이 슬롯  $t+\psi$ 에서 요구한 데이터량은 식(3)과 같으며 버퍼 상태 테이블에 슬롯  $t+\psi$ ,  $C_i$ ,  $C_i(t+\psi)$ 의 값으로 저장된다.

$$\alpha(t+\psi) = \sum_{i=0}^{n-1} C_i(t+\psi) \dots\dots\dots (3)$$

이때 식(3)의  $\alpha(t+\psi)$  값이 디스크 대역폭을 초과하는 지를 식(4)를 이용하여 알 수 있다.

$$\Delta = \alpha(t+\psi) - D_{min} \dots\dots\dots (4)$$

$\Delta$ 의 값이 0보다 클 경우는 슬롯  $t+\psi$ 에서 사용자들이 요구한 데이터량이 디스크 대역폭을 초과한다는 것이며 이때의 슬롯을 과부하 슬롯이라 한다. 과부하 슬롯에서는 모든 사용자들의 요구를 수용할 수 없다. 그러므로 과부하 슬롯  $t+\psi$ 에서 발생한 초과된 양 ( $\Delta$ )을 슬롯  $t+\psi$ 이전의 슬롯에서 선반입해야 하므로 슬롯 서비스 모듈은 슬롯  $t+\psi$ ,  $\Delta$ 값을 선반입 모듈에 전달하여 호출한다.

3.3 선반입 모듈

선반입 모듈은 슬롯 서비스 모듈의 호출에 의해 활성화되며, 사용하는 테이블은 사용자 정보테이블, 메타 테이블, 버퍼상태 테이블, 선반입 정보 테이블이다.

선반입 해야 할 데이터 ( $\Delta$ )는 식(5)를 만족하는 고정된 선반입 가능구간에 선반입한다.

$$t+1 \leq \text{선반입 가능 구간} \leq t+(\psi-1) \dots\dots\dots (5)$$

선반입 가능 슬롯의 선택기준은 최소의 적재비용과 적재 시간을 유지하기 위하여  $t+(\psi-1)$ 에서  $t+1$ 슬롯 순으로 선반입 한다.

슬롯  $t+(\psi-1)$ 에서  $\Delta$ 의 선반입은 두 가지 경우로 나뉘어 진다.  $\Delta$ 가 슬롯 ( $t+\psi-1$ )에 모두 선반입 되는 경우 ( $D_{max} \geq \alpha(t+\psi-1)+\Delta$ )와  $\Delta$ 의 일부분만 슬롯 ( $t+\psi-1$ )에 들어가는 경우 ( $D_{max} < \alpha(t+\psi-1)+\Delta$ )가 있다. 전자의 경우에는 슬롯  $t+\psi$ 에서 발생한  $\Delta$ 값을  $t+\psi-1$ 슬롯에서 모두 선반입함으로써 과부하 문제가 해결되지만, 후자의 경우에는  $t+\psi-2$ 까지 혹은  $t+1$  슬롯까지 선반입이 계속될 수 있다.

또한 슬롯  $t+(\psi-1)$ 부터  $t+1$ 구간 내에서  $\Delta$ 값을 모두 선반입하지 못할 경우가 발생한다. 이런 경우는 선반입 되지 못한 데이터를 해당 사용자들에게 강제적으로 상영실패를 유발한다. 초당 24프레임의 영상 데이터가 사용자 앞에 나타났다가 사라지는 상황에서, 연속해서 프레임에 대한 서비스가 되지 않으면 사용자가 감지할 수 있지만, 일정한 간격을 두고 B프레임 1~2개 정도의 상영실패는 사용자가 감지하기 어려울 뿐만 아니라, 설사 감지된다 하더라도 사용자는 허용할 수 있으리라 본다.[2]

따라서 본 논문에서는 고정된 선반입 구간에서  $\Delta$ 의 값을 모두 적재하지 못할 경우, MPEG 복원에 미치는 영향이 적은 B프레임 중에서 가장 데이터 크기가 큰 B프레임의 순으로 해당 사용자에게만 상

영실패를 유발한다.

### 3.4 승인제어 모듈

본 논문에서 제안하는 승인제어 기법은 통계적인 승인 제어 기법을 사용하는데 새 사용자의 통계적인 데이터 요구량( $\alpha$ )과 승인된 사용자의 통계적인 데이터 요구량( $\beta$ )을 이용한다.

새로운 사용자  $C_n$ 이 요구한 영화  $M_i$ 의 마지막 슬롯이  $m$ 이고, 상영을 요청한 시작 슬롯이  $t$ 일 때  $\alpha$ 의 값은 식(6)에 의해 구해진다.

$$\alpha = \frac{\sum_{i=t}^m C_n(i)}{m-t} \dots\dots\dots(6)$$

식(6)에 의한  $\alpha$ 값은 새로운 사용자가 슬롯  $t$ 에서 요청한 영화의 마지막 슬롯인  $m$ 까지의 슬롯 당 요구하는 평균 데이터 요구량으로써 베타 테이블에 슬롯 별로 오프라인으로 분석되어 있다.

$\beta$ 의 값은 이미 승인된 각 사용자의 평균 데이터 예상 요구량으로 베타 테이블을 이용하여 식(7)에 의해 구한다.

$$\beta = \sum_{i=0}^{n-1} \alpha_i \dots\dots\dots(7)$$

이로써 새로운 사용자에 대한 승인은 식(8)을 만족할 때 승인이 된다.

$$\alpha + \beta < D_{\min} \dots\dots\dots(8)$$

승인이 되고나면 새로운 사용자  $C_n$ 의 서비스 요구 시작 슬롯  $t$ 부터 슬롯  $t+\psi$ 까지의 서비스 요구 정보를 사용자 정보 테이블과 버퍼 상태 테이블에 기록한다.

### 4. 구현 및 평가

본 논문에서 제안한 고정 선반입 버퍼 스케줄러의 구현은 DG/Unix 4.2 시스템에서 오라클 RDBMS 7.3.4, 클라이언트 서버용 프로그램 개발 도구인 파워빌더 6.5.1과 Pro\*C/C++ Precompiler언어를 이용하였다.[6],[7]

그림 2는 구현된 고정 선반입 버퍼 스케줄러의 동작 화면이다.

본 논문에서 사용한 시스템 파라미터는 20명의 사용자에게 한 슬롯 당 2 GOP를 서비스하면서 고정 선반입 기법에서 사용한 선반입 구간을 10슬롯으로 고정시켰다.

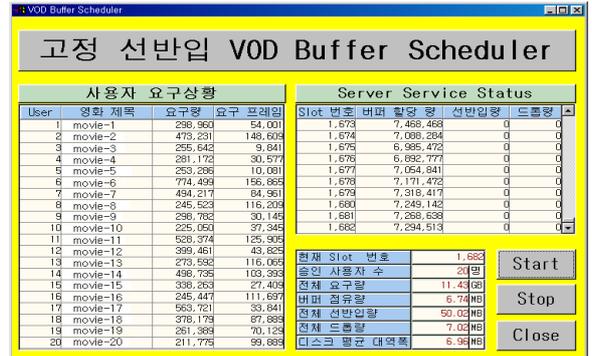


그림 2. 고정 선반입 버퍼 스케줄러 동작 화면

선반입 구간을 가변적으로 한 가변 선반입 기법과 고정 선반입 기법을 이용한 실험의 결과로 각 기법의 버퍼 점유량은 그림 3과 같다.

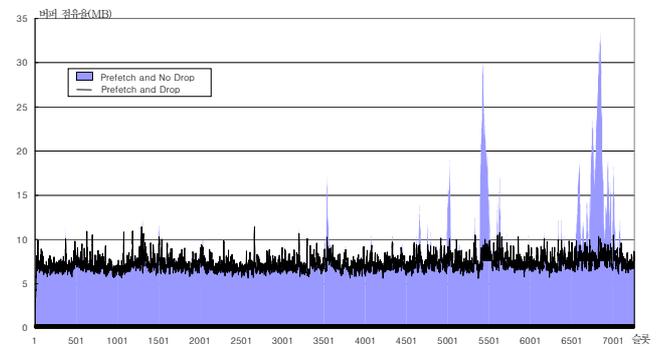


그림 3. 가변 선반입과 고정 선반입 기법의 버퍼 점유량

표 3. 가변 선반입과 고정 선반입 구간의 세부 항목 비교

	가변 선반입 구간	고정 선반입 구간
전체 데이터 요구량	51.71 GB	51.71 GB
전체 선반입량	431.8 MB (0.83%)	258.2 MB (0.49%)
전체 Drop량	-	148.7 MB (0.29%)
선반입 최대 구간	275 Slot	9 Slot
선반입 평균 구간	14.8 Slot	3 Slot
버퍼 평균 점유량	8.23 MB/Slot	7.27 MB/Slot
디스크 대역폭 평균 활용량	7.13 MB/Slot	7.11 MB/Slot

과부하가 발생하지 않은 구간과 과부하가 적게 발생하는 구간에서는 고정 선반입 기법과 가변 선반입 기법의 버퍼 점유율은 큰 차이가 없다. 그러나 과부하가 심한 구간에서는 고정 선반입 기법의 버퍼 최대 점유량은 11.08 MB이고, 가변 선반입 기법의 버퍼 최대 점유량은 33.88MB이다.

따라서 고정 선반입 기법은 MPEG 데이터 복원에 영향이 적은 B프레임을 총 서비스 데이터 양의

표 3. 사용자별 Drop발생 수, Drop 양 및 서비스 받은 양

사용자	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Drop 발생 수	77	77	77	77	77	77	77	77	77	77	77	77	76	76	76	76	76	76	76	76
Drop양 (MB)	9.77	7.94	7.18	9.75	7.28	6.88	6.52	5.99	8.41	6.10	8.10	9.05	6.73	6.01	6.32	7.85	8.21	7.26	7.45	5.90
서비스 받은 양(%)	99.97	99.97	99.97	99.97	99.97	99.96	99.96	99.96	99.97	99.96	99.97	99.91	99.96	99.96	99.96	99.97	99.97	99.97	99.97	99.96

0.29%를 Drop시키면서 버퍼 점유량을 3배 정도 줄일 수 있음을 보여 준다. 표 2는 고정 선반입 기법과 가변 선반입 기법의 세부 항목에 대한 비교이다.

고정 선반입 기법에서 각 사용자에게 슬롯 당 2 GOP, 선반입 임계 구간( $\psi$ )을 10슬롯으로 서비스한 결과 각 사용자의 Drop의 발생 수, Drop 량 및 서비스 받은 양은 표 3과 같다.

5. 결론

가변 선반입 기법은 과부하 구간에서 선반입 임계 슬롯의 값이 증가함으로써 최소의 적재시간과 적재비용을 유지하지 못한다. 따라서 본 논문에서는 선반입 구간을 고정하는 고정 선반입 기법을 제안하였으며, 선반입 기법의 사용에서 필요한 메타 테이블을 데이터베이스를 이용하여 테이블의 관리 용이 및 속도를 향상시켰다.

본 논문에서 제안하는 고정 선반입 기법의 실험 결과, 과부하가 발생하지 않는 상황에서는 가변 선반입 기법과 비슷한 성능을 나타내며, 과부하가 집중적으로 발생하는 상황에서는 전체 서비스 량의 0.29%의 B 프레임 Drop이 발생하지만 가변 선반입 기법에 비해 버퍼 요구량을 3배 정도를 낮출 수 있는 것으로 나타났다. 또한 각 사용자별 전체 서비스 요구량에 평균적으로 약 0.03%정도의 B프레임 Drop이 발생하고 있으나 이 정도의 상영실패는 사용자에게 이해되리라 본다.

향후의 연구과제는 고정 선반입 기법의 사용으로 인한 Drop의 발생을 전체 사용자로 분산시키는 Drop 모듈과 VCR기능의 추가가 요구되며, Drop의 발생 건수를 줄이기 위한 버퍼 공유기법의 적용이 필요하다.

참고문헌

[1] 김순철, “주문형 비디오 서버에서의 효율적인 버퍼 관리 기법” 서울대학교 박사학위 논문  
 [2] 조경운, “라운드를 바탕으로 한 비디오 서버에서

의 버퍼 할당 기법” 서울대학교 석사학위 논문

[3] R. Govindan, David P. Anderson, “Scheduling and IPC Mechanism For Continuous Media”, ACM, selected Area in Computer, pp. 68-80, 1991  
 [4] L. C. Wolf, R. G. Herrtwich, “The System Architecture of the Heidelberg Transport System”, ACM Operating System Review, Vol. 28, No. 2, pp51-64, April, 1994  
 [5] H. M. Vin, P. Goyal, Anshuman Goyal, “A Statistical Admission Control Algorithm for Multimedia Server”, In Proceeding of the ACM Multimedia '94, SanFrancisco, pp33-40, October, 1994  
 [6] Oracle Corporation, Oracle 7 Server Concept Release 7.3, 1996  
 [7] PowerSoft, PowerBuilder 6.0 Application Techniques, 1996