

인터넷 VOD 서비스에서 대기시간 감소를 위한 효율적인 사용자 스케줄링 (An Efficient Service Scheduling for Decrease Waiting Time Based on Internet VOD)

최성욱(Seong-Wook Choi)¹⁾

요약

인터넷 VOD 서비스 스케줄링에서의 배칭(Batching) 기법은 서비스를 요청한 사용자들을 일정 시간 동안 그룹화 하여 한꺼번에 서비스하기 때문에 서비스 개시를 위한 지연시간이 발생한다. 그러나 이 지연시간을 효과적으로 제어하지 못하면 대기시간의 불규칙으로 서비스 공정성이 저하되고, 서비스 취소가 발생할 수 있다. 본 논문에서는 기존의 배칭(Batching) 기법인 선입 선출(FCFS) 및 빈도수 우선 방식(MQL)에서 문제시되던 평균 대기시간의 증가, 비인기 비디오의 서비스 지연 문제를 해결하기 위한 효율적인 사용자 스케줄링을 제안한다. 또한 시뮬레이션 한 결과, 기존의 방식들에 비하여 서비스 평균 지연 시간을 약 20% 정도 단축하였으며, 서비스 최대 대기 시간 보장 등 사용자에게 대한 서비스면에서도 향상된 결과를 보임을 입증하였다.

Abstract

The waiting delays of internet VOD users are inevitable in this policy since the services are not taken immediately upon requests but upon every scheduling points. An inefficient management of such delays makes an unfair service to users and increases the possibility of higher renegeing rates. This paper proposes an efficient service scheduling scheme which improves the average waiting time of users requests and reduces the starvation problem of users requesting less popular movies. Experimental results of simulations show that the proposed scheme improves about 20 percentage of average waiting time and reduces significantly the starving requesters comparing with those of conventional methods such as FCFS and MQL.

1) 정회원 : 시립인천전문대학

논문접수 : 2007. 5. 15.

심사완료 : 2007. 5. 30.

본 연구는 시립인천 전문대학의 2006년도 연구지원비에 의한 것임

I. 서론

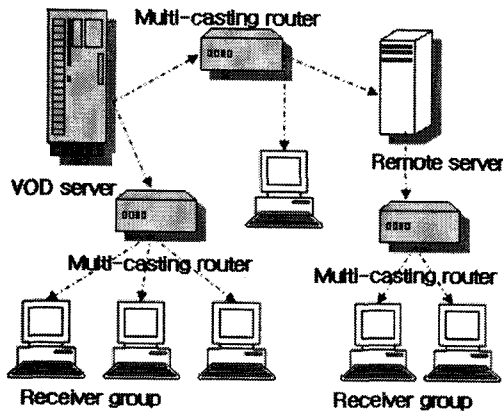
전송 속도의 향상에 따라 대용량 네트워크 대역폭을 기반으로 한 인터넷 웹 서비스는 디지털 비디오, 멀티미디어 데이터 전송을 위한 서비스 품질보장(Quality of Service: QoS), 멀티캐스팅, 화상 전화 시스템, 인터넷 가전, 전자상거래 보안 및 콘텐츠 서비스 등 다양한 응용 서비스가 개발되어 발전하고 있다. 디지털 비디오 서비스는 실시간 서비스 형태와 일괄 수용 서비스 형태로 나누어질 수 있다. 실시간 서비스는 화상 전화나 회의 등과 같이 데이터가 중단간 대략 200ms 이내의 지연시간을 만족시켜야하는 시스템이며, 일괄 수용 서비스는 VOD 서비스 등을 말하는데 이러한 서비스는 스케줄링에 따라 다소간의 지연시간이 발생한다. 인터넷 VOD 서비스 환경에서 중앙 서버는 수시로 사용자의 비디오 상영 요청을 받아들이는데, 이러한 사용자의 요구는 비디오 특성상 응답 주기에 민감하고 연속적인 스트림의 전달이 필요하다. 인터넷 VOD 서버가 다수의 사용자를 수용하여, 연속적으로 서비스하기 위해서 송신 측에서는 동일한 비디오 스트림을 요구하는 사용자들이 존재할 경우, 사용자들에게 전달되는 비디오 스트림들을 시간적으로 동기화 시키는 정책[1, 3, 4]이 필요하다. 이러한 정책을 사용자 그룹화(Grouping) 혹은 일괄 수용 서비스(배칭: Batching)이라고 한다. 이렇듯 VOD 서비스를 위한 배칭 정책은 서비스를 요청하고 난 후 동일 비디오 요청자를 그룹화하기 위한 일정 시간의 지연이 발생하며 한정된 서버의 자원 하에서 이 지연 시간을 어떻게 적절히 조정하는가가 일괄 수용 서비스를 위한 배칭 정책에서의 중요한 문제가 된다. 본 논문은 인터넷 VOD 서비스 환경에서 배칭 정책을 실행할 경우에 사용자의 서비스 평균대기 시간 단축과 서비스의 공정성 확보(최대 대기 시간 보장) 방법을 제시한다. 우선 서비스 평균대기 시간의 단축을 위한 방안으로 서비스 요구 큐에 도착한 각 비디오의 서비스 요구 패

턴을 분석하여 각 비디오에 적합한 효율적인 서비스 요구 큐의 크기를 결정한다. 서로 다른 크기의 서비스 요구 큐에 의해서 각각 비디오의 배칭 간격이 결정되고, 다시 최대 서비스 대기시간과 현재 사용가능한 큐 메모리의 상태에 따라 효율적으로 평활화 된다. 이로써 MQL 스케줄링 시에 문제가 되었던 비인기 비디오의 서비스 문제를 효과적으로 보증할 수가 있다. 2장에서는 본 논문에서 다룰 배칭 서비스 정책에 관한 사례들을 살펴보고, 3장에서는 본 논문에서 제안한 사용자 서비스 정책에 대한 스케줄링 개념에 대하여 논의한다. 4장에서는 본 논문에서 제안한 스케줄링 정책을 시뮬레이션하고, 기존의 방식과 비교 분석한다. 끝으로 5장에서는 결론과 함께 향후 연구 방향에 대하여 간략히 기술한다.

II. 일괄 수용 스케줄링

2.1 VOD 시스템의 구성 및 특징

일괄 수용 서비스를 위한 VOD 시스템의 구성은 그림 1과 같다. 서버는 대용량의 RAID에 비디오 데이터베이스를 저장하고 있으며, 인터넷 통신망에 의해 리시버 그룹(Receiver Group)과 연결되어 있다. 리시버 그룹은 멀티캐스팅 서비스 하에 동일한 비디오를 서비스 받는 사용자의 집합이다. 서버의 비디오 데이터는 멀티캐스팅 라우터를 통과하여 리시버 그룹에 전달되거나 혹은 프록시 서버나 원격 서버로 보내진다. 통신망은 ATM기반의 Overlay 모델, IP over SONET/SDH, IP over WDM, 기가비트 이더넷 등이 활용된다.[14,15,16]



[그림 1] VOD 시스템 구성

[Fig. 1] Architecture of VOD system

2.2 배칭 스케줄링

일괄 수용 스케줄링 정책은 배칭 간격을 결정하는 단계부터 시작한다. 배칭 간격이란 배칭 윈도우의 크기(Size of batching windows)라고도 한다. 배칭 간격을 결정하기 위한 방식은 큐의 크기를 기준으로 한 방식[17]과 시간을 기준으로 한 방식[18]이 연구되었다. 이 두 가지는 모두 고정된 배칭 간격(큐, 시간)을 갖는다.

(1) 큐의 크기를 기준으로 한 배칭

사용자가 서버에 비디오 서비스를 요청할 때 이러한 요청 정보는 일단 서버의 서비스 요청 큐(Service request queue)에 저장되는데, 큐의 크기를 기준으로 한 배칭에서는 서비스 요청 큐에 도착된 사용자 수(큐의 크기)를 기준으로 배칭 간격을 결정한다. B_j 를 미리 기준으로 정의된 j 번째 비디오의 요청 큐의 크기, 비디오 j 의 평균 서비스 도착률을 λ_j 라 하고, 한 번의 배칭에 의해서 절약될 수 있는 I/O 량을 $E[N_j]$, 큐에 도착 후 스케줄링 될 때까지

의 평균대기 시간을 $E[L_j]$ 이라 하면, 각각 (6)(7)의 식으로 표현 될 수 있다.

$$E[N_j] = B_j - 1 \quad (6)$$

$$E[L_j] = (1 / B_j) \sum (B_j - i) / \lambda_j = (B_j - i) / 2\lambda_j \quad < \text{for } i = 1 \text{ to } B_j > \quad (7)$$

이 방식의 대표적인 것이 FCFS 및 MQL[19,20]이며, 이 방식은 서버의 자원을 효율적으로 사용할 수 있으나, 서비스 요청 도착률의 편차가 클 경우에는 배칭 서비스 간격의 편차도 커져서, 최악의 경우에 평균대기 시간이 너무 길어지는 문제가 있다. 그러므로 이 방식을 사용할 경우에는 서비스 도착률이 낮은 인기 없는 비디오를 위하여 별도의 대책이 필요하다.

(2) 시간 간격을 기준으로 한 배칭

일반적으로 많이 활용되는 시간 간격을 기준으로 한 배칭은 서버가 일정한 시간 단위를 기준으로 서비스 요청 큐에 저장된 사용자의 정보를 조사하여 스케줄링 하는 방식이다.

T_j 는 j 번째 비디오의 배칭 간격의 크기, 비디오 j 의 평균 서비스 도착률을 λ_j 라 하고, 한 번의 배칭에 의해서 절약될 수 있는 I/O량을 $E[N_j]$, 큐에 도착 후 스케줄링 될 때까지의 평균 대기 시간을 $E[L_j]$ 이라 하면, 각각 (8)(9)의 식으로 표현 될 수 있다.

$$E[N_j] = \lambda_j \cdot T_j \quad (8)$$

$$E[L_j] = T_j (2 + \lambda_j \cdot T_j) / 2(1 + \lambda_j \cdot T_j) \quad (9)$$

시간 간격을 기준으로 하는 배칭은 단지 일정한 시간 간격으로 서비스를 진행하기 때문에 서비스 메커니즘이 간단하고 사용자 서비스 측면에서는 효과적이나, 서비스 요구 도착률이 수시로 변화되는 환경에서는 서버의 자원을 효율적으로 사용할 수 없다.

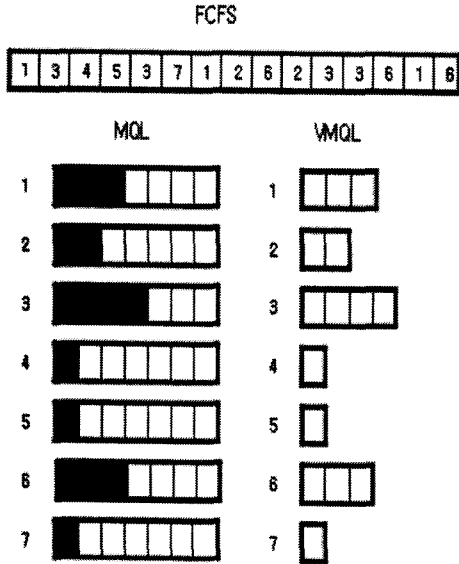
Ⅲ. 효율적인 사용자 스케줄링

일괄 수용 서비스(Batched admission service)는 서버 자원의 한계를 극복하고 사용자 서비스의 수용능력을 증대하기 위하여 연구되었다. 이 방식은 미리 정의된 시간 간격 동안에 모아진 사용자의 동일한 비디오의 요구를 한번의 I/O 수행으로 처리하게 되므로 그에 해당하는 만큼 디스크의 부하를 감소시킬 수 있다. 일괄 수용 서비스 정책 중에서 대표적인 것으로는 배칭(Batching)[1,3,4] 스케줄링이 있는데, 이는 동일한 비디오를 요구하는 사용자의 서비스 개시 시간을 일정한 시간 간격(Time interval)동안 모아서 함께 처리하는 방식이다. 일괄 수용 서비스를 위한 배칭 스케줄링은 사용자의 입장에서 서비스 요청하고 난 후 일정 시간의 지연을 감수해야 하는 불편함이 있다. 본 논문에서 제안한 배칭 스케줄링은 첫째, 비디오 인기도와 시간대에 따라 변화될 수 있는 비디오 요구 패턴에 동적으로 감시하여 각 비디오의 서비스 간격을 결정함으로써 서비스 평균대기 시간을 효과적으로 단축한다. 둘째 서비스의 공정성을 확보하기 위하여 비인기 비디오에 대한 서비스 정책을 준비한다. 이를 위하여 비디오 요구도에 따른 배칭 시간에 만족한 비디오와 최대 대기시간을 초과한 비디오가 존재할 경우 이를 모두 선택한다. 물론, 이에 따른 서버내의 디스크 및 네트워크 대역폭의 증가가 일정범위를 넘어설 경우에는 서비스 중인 비디오의 재 그룹화 스케줄링 정책을 제안한다. 재 그룹화 정책은 버퍼 내에서 흩어져서 서비스 중인 인근 사용자 그룹들을 다시 하나의 그룹으로 통합하는 작업이며, 이로 인하여 버퍼의 사용이나 디스크 대역폭을 절약 할 수가 있다. 인근 서비스 그룹간의 통합은 서비스 처리율을 변경하는 방식으로 처리하는데, 이 방식은 기본적으로 [13]에서 유래되었으며, 이를 Golubchik, Lui에 의하여 사용자 스케줄링 [11,12]에 응용되었다. 서비스 처리율의 조정은 기본적으로 서비스 처리되는 비디오 프레임들

사이에 임의로 새로운 프레임을 삽입하거나, 프레임 중간 중간을 삭제하여 처리율을 조정할 수 있으며 프레임의 삽입, 삭제율이 약 5% 미만일 때는 사용자가 쉽게 인식하지 못한다 [13]고 연구된 바 있다. 본 논문에서는 서비스 처리율 조정으로 현재 서비스 중인 사용자 재 그룹화하여 서버의 자원을 절약하기 위한 방안을 제안한다.

3.1 서비스 개시 스케줄링

그림 2는 배칭 스케줄링을 위한 서비스 요구 큐의 형태를 나타내었다. FCFS(First Come First Service) 전략은 하나의 서비스 요구 큐를 가지고 있으며, 먼저 요구된 비디오를 우선적으로 스케줄링 한다. 그러므로 FCFS 전략은 비디오의 인기도에 관계없이 끝고루 서비스해줄 수 있다는 장점이 있지만, 서비스 요청 빈도가 높은 비디오와 서비스 요청 빈도 낮은 비디오를 동일한 우선순위로 서비스해주기 때문에 평균대기 시간이 증가한다. 반면에 MQL(Multiple Queue List)는 복수(비디오의 수)의 서비스 요구 큐를 준비하여 그중에서 가장 서비스 요구가 많은 비디오 순서대로 서비스하여 준다. MQL 전략은 FCFS와는 다르게 서비스 요청의 빈도가 높을수록 서비스를 자주 해주게 되어 FCFS에 비하여 상대적으로 평균대기 시간을 감소할 수 있지만, 최악의 경우에 요청 빈도가 낮은 비디오는 서비스 대기시간이 너무 길어져 버리는 문제가 발생할 수 있다. 그러므로 효율적인 사용자 서비스를 위하여 서비스 평균 대기시간을 감소시키면서 요청빈도가 낮은 비디오라도 일정시간 안에는 서비스해 주는 전략이 필요하다. 본 논문에서 제안한 VMQL(Variable Multiple Queue List) 방식은 복수개의 서비스 요구 큐를 사용하지만, 큐의 크기가 모두 일정한 것이 아니라 서비스 요구도에 따라 수시로 그 크기를 변화시키는 가변 길이 서비스 요구 큐를 활용한다.



[그림 2.] 배치 스케줄링 큐

[Fig. 2] *Batching Scheduling Queue*

본 논문에서 제안한 비디오에 대한 요구도에 따라 배치 간격의 크기를 결정하는 방식은 기본적으로는 비디오의 인기도 분포인 Zipf distribution을 활용한다. 보통 비디오의 인기도를 나타내는 함수는 식(10)과 같다. 한편 식(11)은 인기도에 대한 접근 확률의 편중도를 높이기 위하여 식(10)을 약간 변형하고 $\theta = 0.271$ 로 적용한 식이며, 비디오에 대한 사용자의 요청률을 나타낼 때 일반적으로 많이 사용된다.

$$f(i) = C / i \text{ where } C = 1 / \sum (1/i) < \text{for } i = 1 \text{ to } I > \quad (10)$$

$$f(i) = C / i^{(1-\theta)} \text{ where } C = 1 / \sum (1/i^{(1-\theta)}) < \text{for } i = 1 \text{ to } I > \quad (11)$$

서버에 도착되는 사용자의 서비스 요구 도착률을 λ 라 하였을 때 j 번째 비디오에 대한 도착률을 인기도 함수 $f(j)$ 를 활용하여 정리하

면 식 (12)과 같다.

$$\lambda_j = \lambda \cdot f(j) \quad (12)$$

이를 이용하여 전체 비디오의 서비스 도착률과 최대 서비스 대기 시간을 포함한 j 번째 비디오에 대한 평활화된 서비스 시간 T_j 는 식 (13)으로 정리될 수 있는데 여기서 T_{max} 는 사용자가 기다릴 수 있는 최댓값이며, $\sum f(n)$ 는 배치에 포함되는 모든 비디오의 도착률의 합이 된다.

$$T_j = T_{max} \cdot ((\sum f(n) - f(j)) / \sum f(n)) \quad (13)$$

j 번째 비디오에 대한 서비스 시간은 식 (14)와 같이 서비스 요구 큐의 수 q_j 로 계산될 수가 있다. 여기서 $\alpha (> 0)$ 는 서비스 요구 큐의 총괄 범위를 결정하는 계수이며, 현재 시스템의 버퍼 활용도를 기준으로 산출한다.

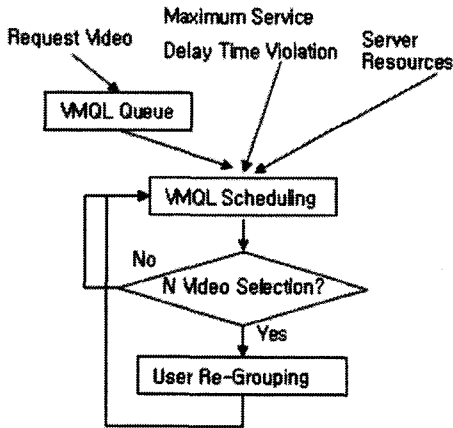
$$q_j = \alpha / T_j \quad (14)$$

이 방식은 비디오 서비스 요구를 토대로 수시로 스케줄링에 적용으로 효율적인 서비스를 할 수가 있다. 또한 최대 대기 시간(T_{max})이 스케줄링 시에 반영되기 때문에 원칙적으로 최대 대기 시간이 보장된다고 할 수가 있다. 한편 서비스 진행 중에 최대 대기 시간에 근접하는 비디오가 있을 경우, 이를 서비스하기 위하여 네트워크나 서버의 자원이 일정치 이상 초과가 예상될 경우에는 서비스 사용자를 재 그룹화 하여 멀티캐스팅의 정도를 효율적으로 높여주는 방안을 제안하였다.

3.2 사용자 재그룹화 스케줄링

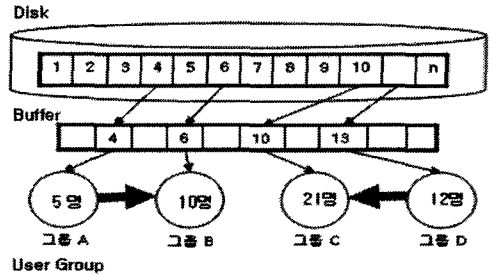
그림 3에는 본 논문에서 제안된 VMQL 스케줄링과 사용자의 재그룹핑 스케줄링의 연관도를 나타내었다. 먼저 사용자 서비스요구는 VMQL 큐에 저장되어지고, 최대 서비스 대기 위반의 정보 및 서버의 디스크 대역폭이나 버

퍼 등의 자원 사용 현황을 참고하여 VMQL 스케줄링을 실행한다. 짧은 시간동안 서비스해야 할 비디오가 복수가 선택되었을 경우, 이를 서비스 하기 위하여 추가로 사용되는 시스템 자원이 전체 자원의 80%를 초과할 경우에는 사용자 그룹을 재조정하기 위한 스케줄링을 실행하게 된다.



[그림 3.] VMQL 스케줄링 연관도
 [Fig. 3] Relationship of VMQL

그림4는 그룹 A, B, C, D의 현재 서비스 중인 프레임 번호가 4, 6, 11, 13라 하고, 그룹 내의 멀티태스킹 인원수가 각각 5, 10, 21, 12 이라 할 때, 재생을 조절에 의한 그룹 병합의 예를 보여준다. 여기서는 재생을 조절의 시간과 그 범위를 최소화하기 위하여 그룹간의 병합은 일단 서비스 시간차가 짧은 두 개의 그룹을 우선으로 선정하고, 서비스 재생률의 조절은 멀티태스킹 인원수가 적은 그룹으로 한다. 결국 4개의 그룹은 각각 그룹B와 그룹C로 통합되며, 다음 차수에서는 두 개의 그룹이 다시 그룹C로 통합될 수가 있다.



[그림4] 서비스 재생을 변경

[Fig.4] Alternative service ratio

서비스 처리율 조절의 방식은 뒤쪽 프레임에서부터 처리율을 증가시켜, 앞쪽 프레임과 병합하는 방식인 처리율 증가 방식(Increase ratio time)과 앞쪽의 서비스 처리율을 감소시켜 뒤이어오는 프레임과 병합하는 처리율 감소 방식(Decrease ratio time)이 있다. 그리고 이 두 방식을 통합한 양방향 조정(Bi-way ratio time)이 있는데, 이 방식은 그룹 병합 속도가 바로 위의 두 방식보다 빠르기 때문에 자원의 부족이 급하게 예상될 때 사용할 수가 있다. 처리율 변경을 위한 초당 삭제 프레임 수를 F_d , 추가 프레임 수를 F_a 라 하고, 초당 프레임 서비스 처리 수를 E_f 라 할 때 그룹 병합에 걸리는 시간은 *forward group merge*와 *backward group merge*, *bi-way group merge*을 각각 식(15), 식(16), 식(17)로 표시할 수 있다.

$$Gmt_{ij_For}(Gst_j, Gst_j) = Ts * Ef / Fd \quad (15)$$

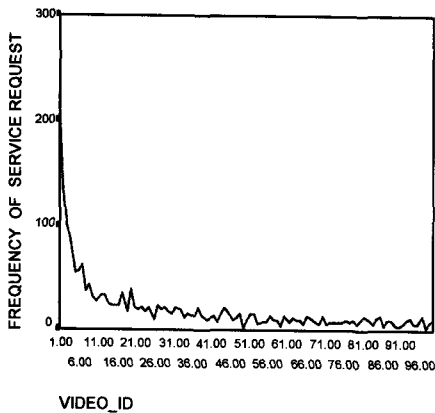
$$Gmt_{ij_Back}(Gst_j, Gst_j) = Ts * Ef / Fa \quad (16)$$

$$Gmt_{ij_way}(Gst_j, Gst_j) = Ts / (1 / Ed + 1 / Fa) \quad (17)$$

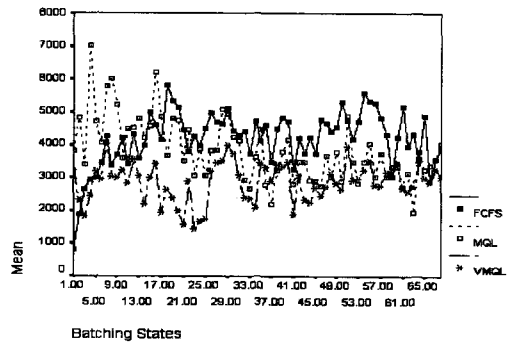
IV. 시뮬레이션 및 분석

본 논문에서 제안한 인터넷 VOD 서버에서 서비스 정책의 주요 관심은 주어진 서버의

능력 하에서 서비스 평균대기 시간의 단축과 서비스 최대 대기 시간 보장에 있으며, 이를 시뮬레이션 하기 위하여 사용자가 요청한 비디오를 서비스 받기까지 평균대기 시간 및 최대 서비스 대기 시간을 대표적인 배칭 기법인 FCFS, MQL 특성을 비교하였다. 또한 다양하고 객관성 있는 분석을 위하여, 시뮬레이션 결과의 표현 및 출력은 통계처리 전용 패키지를 활용하였다. 시뮬레이션 자료 및 기본 파라메타의 내용은 표 1에 각각 나타내었으며 케이스별 파라메타는 시뮬레이션 결과 분석에서 언급하였다. 시뮬레이션에 사용할 자료의 비디오 요구패턴은 Zif Distribution 을 기본으로 하였고, 서비스 도착율은 λ 는 1sec당 한 개의 평균 도착률을 갖는 Poisson Distribution으로 하였다. 그리고 VMQL 스케줄링을 위한 최대 대기 보장 시간은 600sec으로 하였다. 그림 5는 시뮬레이션에 사용된 100개의 비디오에 대한 요구 패턴을 나타낸다. 그림6, 7, 8 및 표2는 스케줄링별 서비스 대기시간을 분석한 결과이다. 스케줄링 종류는 FCFS, MQL, VMQL이다. 그림6은 배칭 단계별 스케줄링 당 평균 대기시간의 변화를 나타내었으며, 표2에는 대기시간의 분포 형태를 테이블로 표시하였다. 그리고 각 그래프 및 표에 있는 모든 시간의 단위는 10msec(0.1sec)이다.



[그림 5] 비디오 요구 패턴
[Fig. 5] Request video pattern



[그림 6] 대기 시간의 변화
[Fig. 6] variation of waiting time

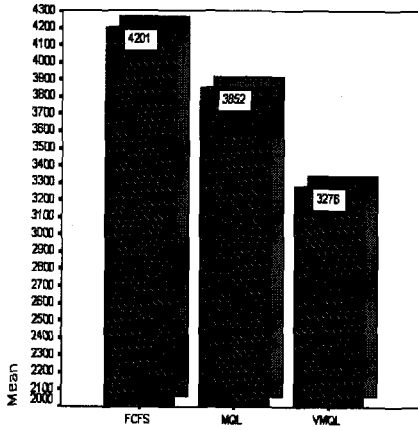
[표 1] 기본 파라메타 테이블
[Table. 1] Basic parameter table

구분	내용	값
비디오 요구 패턴	Zif Distribution	
λ (포아송 분포)	서비스 도착율	1/sec
V_no	비디오 수	100개
V_play_time	비디오 상영 시간	3600sec
Max_wait_time	최대 대기 시간	600sec
Time_interval	배칭 간격	60sec
data count	자료 건수	2000
α	총괄 큐 계수	1

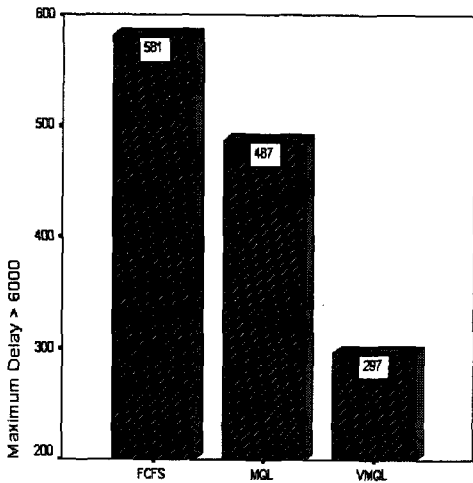
[표 2] 스케줄링별 평균대기 시간 비교표

[Table.2] A comparative table of average waiting time by scheduling polices

		FCFS	MQL	VMQL
N	Valid	2000	2000	2000
	Missing	0	0	0
Mean		4201.2605	3851.7605	3276.4105
Std. Deviation		2509.0600	4633.2404	2056.6035
Minimum		1.00	.00	.00
Maximum		6646.00	25117.00	6299.00



[그림 7] 스케줄링별 평균대기 시간
 [Fig. 7] Scheduling policies and mean of waiting time



[그림 8] 최대 대기 시간 위배 건수
 [Fig. 8] Frequencies over maximum waiting time

그림 6, 7에서 보는 바와 같이 평균대기 시간에서 VMQL은 3276으로 FCFS의 4201, MQL의 3852 보다 20% 정도 짧게 나타나고 있다. 또한 서비스 지연에서 오는 사용자의 서비스 요청 취소와 밀접한 연관이 있는 최대 서비스 대기 보장 시간은, 기본 파라메타 테이블

인 표2에 표시된 것처럼 이를 600sec으로 하였을 경우에 최대 보장 시간을 넘는 서비스 건수를 계산하여 그림 8에 나타나 있다. 그림 8에서 총 위배 건수에서는 VMQL 스케줄링이 297건으로 FCFS의 581건, MQL의 487건 보다 40~50% 정도 단축하고 있다. VMQL 방식은 스케줄링 단계에서 최대 서비스 보장 개념을 사용함으로써 원칙적으로 서비스 대기시간이 정확하게 보장되어야 하겠지만, 자원의 효율적인 활용을 위하여 사용자를 재그룹화하는 시간이 추가되어 있기 때문에 보장 시간이 약간 오버되었다 할 수가 있다. 그러므로 VMQL 스케줄링에서 정확한 최대 서비스 보장 시간은 초기에 결정한 최대 보장 시간에 사용자 재그룹에 소요되는 시간을 합한 시간이 됨을 알 수 있다.

V. 결론

일괄 수용 서비스 환경에서 배칭 정책을 실행할 경우에 사용자의 서비스 평균대기 시간 단축과 서비스의 공정성 확보(최대 대기 시간 보장)를 실현하기 위하여 본 논문에서 제안된 VMQL 스케줄링은 사용자의 서비스 요구패턴과 자원의 사용율에 효과적으로 적응하기 위한 배칭 정책이며, 이는 서비스 요구 큐에 도착한 각 비디오의 요청 빈도(인기도) 및 서비스 대기 시간에 관한 정보를 토대로 서비스 허락 여부 및 개수를 동적으로 스케줄링 한다. 이 정책은 비디오의 인기도 분포(Zipf distribution)와 서비스 요청 빈도에 따라 그때그때 산출되어지는 가변 길이 큐를 활용하여, 배칭 간격 내에서 다중 선택(Zero or many)되어 지기 때문에, 주변의 서비스 환경을 고려하지 않는 종래의 고정적 배칭 정책에 비하여 효율적인 사용자 서비스가 가능하다. 이는 시뮬레이션 결과에서도 나타난 바와 같이 FCFS 나 MQL 등 다른 스케줄링에 비하여 20% 정도의 평균 대기 시간의 감소를 보였으며, 서비스 취소 여부를 판단하는 최대 서비스 보장 위배건수에서도 40~50% 정도 감소하는 것으로 나타났다.

VMQL 스케줄링은 서비스 요구패턴이 다양하게 변화하고 자원 활용이 엄격하게 정의된 VOD 시스템에서 유리한 방식이라 할 수 있다.

참고 문헌

[1] Heek-Young Woo, Chong-Kwon Kim, "Multicast Scheduling for VOD Services". *Multimedia Tools and Applications*, pp 157-171, 1996

[2] Leana Golubchik, John C.S. Lui, "Adaptive piggybacking: a novel technique for data sharing in video-on-demand storage servers". *Multimedia Systems*, pp 140-15, 1996

[3] Asit Dan, Dinker Sitaram, "Dynamic batching policies for an on-demand video server". *Multimedia Systems*, pp 112-121, 1996

[4] Asit Dan, Dinkar Sitaram and Perwez Shahabuddin, "Scheduling Policies for an On-Demand Video Server with Batching". *ACM Multimedia*, pp 15-23, 1994

[5] Y.W.Park, K.H.Baek, and K.D.Chung. "Reducing network traffic using two-layered cache servers for continuous media on the internet" *In Proc.of the IEEE Int'l Conf.on Computer Software and applications*, pages 389-394, 2000.

[6] Duc.A.Tran, Kien.A.Hua, and Simon Sheu, "A new caching architecture for efficient Video-on-Demand services on the internet", *Proceedings of IEEE on Applications and the Internet*, pp.172-181, January 2003

[7] Wen-Jiin Tsai and Suh-Yin Lee, "Dynamic Buffer Management for Near Video-On-Demand Systems". *Multimedia Tools and Applications*, pp 61-83, 1998

[] Kurt Rothermel, Tobias Helbig, "An adaptive protocol for synchronizing media streams", *Multimedia Systems*, pp 324-336,

1997

[9] 박승철, 최양희, "QOS를 고려한 적응형 멀티미디어 동기화 기법" *정보과학회 논문지 (A) 제 22권 제9호*, PP 1307-1318, 1995

[10] Hong-ki Jung, Seung-Kyu Park, "Grouping and Buffer Management Methods in a VOD Server" *Proceeding of ITC-CSCC99, Sado, Niigata, Japan*, pp899-902, July 13-15, 1999

[11] Ohanian TA, "Digital nonlinear editing: new approaches to editing film and video. Focal Press, Boston, Mass, 1993

[12] Leana Golubchik, John C.S. Lui, "Adaptive piggybacking: a novel technique for data sharing in video-on-demand storage servers". *Multimedia Systems*, pp 140-155, 1996

[13] L.Goubchik, J.C.S. Lui. and R. Muntz, "Reducing I/O demand in video-on-dem and stoage servers", in proc. of Intl. Conf. on Measurement and Modeling of Comp. Syst.(SIGMETRICS'95), pp 25-36, 1995

[14] J. Manchester, J. Anderson, B. Doshi, and S. Dravida, "IP over SONET," *IEEE Communications Mag.* May 1998

[15] D. Cavendish, "Evolution of Optical Transport Technologies: From SONET/SDH to WDM," *IEEE Communications*, June 2000

[16] Nen-Fu Huang and Shi-ming Zhao, "A Novel IPRouting Looking Scheme and Hardware Architecture for Multigigabit Switching Routers", *IEEE Journal of*

[17] Ozden B, Biliris A, Rastogi R, Silberschatz A, "A Low-cost storage server for movie-on-demand data bases". *Proceedings of the 20th international Conference on Vary Large Databases*, pp 594-605, 1994

[18] Takagi H "Queueing analysis: a foundation of performance evaluation". vol: vacation and priority systems, part1, North-Holland. NewYork NY, 1991

[19] Asit Dan, Dinker Sitaram, "Dynamic

batching policies for an on-demand video server". *Multimedia Systems*, pp 112-121, 1996

[20] Asit Dan, Dinkar Sitaram and Perwez Shahabuddin, "Scheduling Policies for an On-Demand Video Server with Batching". *ACM Multimedia*, pp 15-23,1994