論文2000-37CI-3-5

효율적인 사용자 서비스를 위한 적응적 배칭 스케줄링 정책

(An Adaptive Batching Scheduling Policy for Efficient User Services)

崔 成 旭 *, 金 鍾 景 **, 朴 升 圭 **, 崔 景 熙 **, 金 東 潤 **, 崔 德 圭 **
(Sung-Wook Choi, Jong-Kyung Kim, Seung-Kyu Park, Kyung-Hee Choi, Dong-Yoon Kim, and Dug-Kyoo Choi)

요 약

일괄 수용 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, 비 인기 비디오의 서비스 지연 문제를 해결하기 위한 적응적 배칭 정책을 제안한다. 적응적 배칭 정책의 개념은 일정한 시간 간격 내에 서비스 요청 패턴을 감시하여 동적으로 서비스 비디오 수를 다중으로 선택하는 방식이며, 이를 위하여 현재의 서버 활용률, 비디오 인기 분포도, 서비스 대기 시간이 활용된다. 또한 이를 시뮬레이션 한 결과, 기존의 방식들에 비하여 서비스 평균 지연 시간을 약 20~30% 정도 단축하였으며, 서비스 최대 대기 시간 보장 등 사용자에 대한 서비스면 에서도 향상된 결과를 보임을 입증하였다.

Abstract

The waiting delays of 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 reneging rates. This paper proposes an adaptive batch scheduling scheme which improves the average waiting time of users requests and reduces the starvation problem of users requesting less popular movies. The proposed scheme selects dynamically multiple videos in given intervals based on the service patterns which reflect the popularity distribution(Zipf-distribution) and resource utilizations. Experimental results of simulations show that the proposed scheme improves about 20-30 percentage of average waiting time and reduces significantly the starving requesters comparing with those of conventional methods such as FCFS and MQL.

* 正會員, 市立 仁川專門大學 電算科
(Dept of Computing, Incheon Junior College)
** 正會員, 亞洲大學校 컴퓨터工學部
(School of Computer Engineering, Ajou Univ)
※ 이 연구를 위해 정보통신연구진흥원 및 아주대의 지원에 감사합니다.
接受日字:1999年 8月 19日, 수정완료일 : 2000年 1月31日

Ⅰ. 서 론

VOD 서비스 환경에서 중앙 서버는 수시로 사용자의 비디오 상영 요청을 받아들이는데, 이러한 사용자의 요구는 비디오 특성상 응답 주기에 민감하고 연속적인 스트림의 전달이 필요하다. 그러므로 서버가 사

용자에 대한 초기 서비스의 개시를 실시간으로 스케줄링 하는 실시간 수용 서비스(Real-time admission service)의 경우에는 서버가 동시에 수용할 수 있는 사용자의 수는 서버의 자원, 즉 사용 가능한 버퍼의 크기나 디스크 대역폭에 따라 제한적일 수밖에 없다. 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 요구들을 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. 그 외에 배칭 스케줄링에서의 초기 서비스 지연을 방지하기 위하여 재생율 조정의 개념을 추가한 adaptive piggybacking 정책[2,7] 연구되었으나 서비스 재생율의 변동에 의한 QOS의 저하가 발생할 수 있으며 영상 및 음성 정보를 재 동기화하기 위한 부가적인 정책이 필요하여 메커니즘 복잡해질 수 있다[10,11]. 일괄 수용 서비스를 위한 배칭 정책은 사용자의 입장에서는 서비스를 요청하고 난 후 일정 시간의 지연을 감수해야 함으로 서비스 요청을 취소하는 경우가 발생한다. 따라서 한정된 서버의 자원하에서 이들 각 요소들을 어떻게 적절히 조정하는가가 일괄 수용 서비스를 위한 배칭 정책에서의 중요한 문제가 된다.

본 논문은 일괄 수용 서비스 환경에서 배칭 정책을 실행할 경우에 사용자의 서비스 평균 대기 시간 단축과 서비스의 공정성 확보(최대 대기 시간 보장) 방법을 제시한다. 이를 위하여 본 논문에서는 서비스 요구 큐에 도착한 각 비디오의 요청 빈도(인기도) 및 서비스 대기 시간에 관한 정보를 토대로 서비스 허락 여부 및 개수를 동적으로 스케줄링 하는 방안을 제안한다. 2장에서는 본 논문에서 다룰 일괄 서비스 환경에 관한 사례들을 살펴보고, 3장에서는 본 논문에서 제안한 적응적 배칭 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

1. VOD시스템의 구성 및 특징

일괄 수용 서비스를 위한 VOD 시스템의 구성은 그림 1과 같다. 서버는 비디오 자료를 관리하고 있는 대용량의 RAID에 비디오 데이터 베이스를 저장하고 있으며, ATM 고속 통신망에 의해 클라이언트들과 연결되어 진다. 사용자의 서비스 요청은 요청 큐(Request queue)에서 일정한 배칭 간격 동안 대기하게 되며, 서비스 스케줄링 시에 서비스에 필요한 자원을 할당받게 된다. 클라이언트에 있는 수신 큐(Receive queue)는 네트워크로부터 입력된 미디어 데이터를 인트라 미디어 동기화(Intra-media-synchronization)하여 지터(Jtter)나 스큐(Skew)를 제거하고 비디오 복원기(Video decoder)에 의하여 비디오 데이터로 복원되기까지 대기하는 장소로 사용된다.

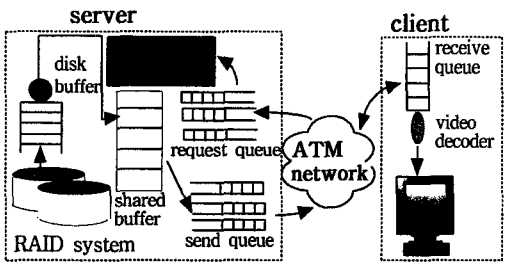

그림 1. VOD 시스템 개요
Fig. 1. VOD system overview.

2. 일괄 수용 스케줄링 정책

일괄 수용 서비스 정책은 외부적으로는 ATM의 멀티케스팅의 특성에 영향을 받지만, 내부적으로는 사용자 그룹핑을 위한 배칭 간격의 결정 방식[2,8,9]이나 사용자의 배칭 스케줄링 전략[3,4]이 그 성능을 결정하는 또 다른 중요한 요소가 된다.

(1) 배칭 간격(Batching interval)

일괄 수용 스케줄링 정책은 배칭 간격을 결정하는 단계와 이를 기본으로 한 스케줄링 단계로 이루어진다. 배칭 간격이란 일괄 수용 스케줄링의 경우에 서버의 관점에서 본 서비스 수용 스케줄링의 간격을 말하는 것이며, 사용자 입장에서 볼 때에는 일정한 서비스 간격내의 동일한 비디오를 요구하는 사용자들이 집단적으로 함께 서비스를 받게되므로 그룹핑(Grouping)[5]이라는 용어로 널리 사용되고 있다. 그러므로 배칭 간격의 결정 시 서버의 입장과 사용자의 입장을 함께 고

려해야만 효과적인 일괄 수용 정책을 실현 할 수 있다. 배칭 간격을 결정하기 위한 방식은 큐의 크기를 기준으로 한 방식과 시간을 기준으로 한 방식이 있다[2,9].

① 큐의 크기를 기준

사용자가 서버에 비디오 서비스를 요청할 때 이러한 요청 정보는 일단 서버의 서비스 요청 큐에 저장되는데, 큐의 크기를 기준으로 한 배칭에서는 서비스 요청 큐에 도착된 사용자 수(큐의 크기)를 기준으로 배칭 시점을 결정한다. 그러나 서비스 요청 수를 기준으로 하는 방식은 서비스 요청 도착률의 편차가 클 경우에는 평균 대기 시간이 너무 길어지는 문제가 있어 적용이 쉽지 않으므로 이 방식을 사용할 경우에는 서비스 도착률이 낮은 인기 없는 비디오를 위하여 별도의 대책이 필요하다.

② 시간 간격을 기준

시간 간격을 기준으로 한 배칭은 서버가 일정한 시간 단위를 기준으로 서비스 요청 큐에 저장된 사용자의 정보를 조사하여 스케줄링 하는 가장 일반적인 방식이다. 이 방식은 요구 도착률의 편차에 관계없이 일정한 시간 간격으로 서비스 할 수 있다는 장점이 있으나 서비스 도착률이 높은 인기 있는 비디오를 효율적으로 서비스하기에는 문제가 있다. 배칭 스케줄링 시에 인기도를 감안하기 위한 방안은, 초기에 배칭 시간 간격의 결정 단계에서 적용하거나[12], 이후에 스케줄링 단계에서 이를 적용하는 방안이 있을 수 있다. 본 논문에서 제안된 적응적 배칭 스케줄링은 두 번째 방안으로써, 현재 비디오 서비스 요구 도착률 및 서비스률의 정보를 토대로 스케줄링 한다.

(2) 배칭 스케줄링

배칭 스케줄링은 배칭 간격을 주기로 하여 서비스 요청 큐에 도착된 사용자 서비스 정보를 토대로 어떤 사용자를 서비스 개시할 것인가를 결정한다. 이러한 배칭 스케줄링 방식으로는 선입 선출 방식(First Come First Service; FCFS)과 다중 큐를 활용한 빈도수 우선 방식(Multi Queue List; MQL)이 있다[3,4].

① FCFS 스케줄링

FCFS은 일괄 수용 정책을 기본으로 하면서, 먼저 들어온 요청을 먼저 서비스해 준다는 전략이며, 이를 위하여 서버는 오직 하나의 서비스 요청 큐를 가진다. 그래서 배칭 시간 간격이 경과한 후 서비스 요청 큐 안에서 먼저 도착된 요청부터 우선 서비스를 해주는데, 이때 이와 동일한 비디오를 요청한 사용자가 서비스 요청 큐 내에 있을 경우에 이들도 함께 서비스해준다. 그림 2에서 배칭 지점 *t0*에서는 비디오 3이 선택되었고, *t1*에서는 다음 순서인 비디오 1이 선택되는데, *t1* 시점까지 요청된 4개의 비디오 1은 일괄 선택된다. FCFS 전략은 비디오의 인기도에 관계없이 골고루 서비스해줄 수 있다는 장점이 있지만, 서비스 요청 빈도가 높은 비디오와 서비스 요청 빈도 낮은 비디오를 동일한 우선 순위로 서비스해주기 때문에 평균 대기 시간이 증가한다.

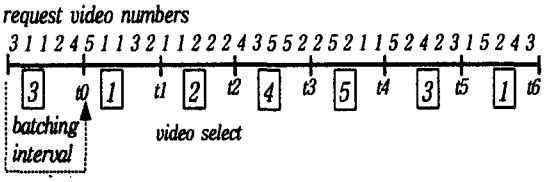

그림 2. FCFS 스케줄링 실행의 예
Fig. 2. Example of FCFS scheduling.

(2) MQL 스케줄링

MQL은 FCFS에서 나타나는 평균 대기 시간의 증가를 개선하기 위해서 고안된 전략이며 서버가 서비스 가능한 영화의 개수만큼 서비스 요청 큐를 가지고 있으며, 각각의 비디오에 대한 사용자의 서비스 요청은 자신의 고유한 큐에 저장된다. MQL은 배칭 시간 간격이 경과한 후 각각의 서비스 요청 큐를 조사한 후, 서비스 요청이 제일 많은 큐를 우선적으로, 그 큐에 있는 사용자 모두를 동시에 서비스한다. 그림 3에서 첫 단계는 *t0*지점까지 요청 빈도가 가장 많은 비디오 1이 선택되었고, 다음 단계는 *t1*지점까지는 비디오 1, 2가 각각 2개씩으로 가장 많이 요청되었으나, 여기에서는 먼저 요청된 비디오 2를 선택하게 된다.

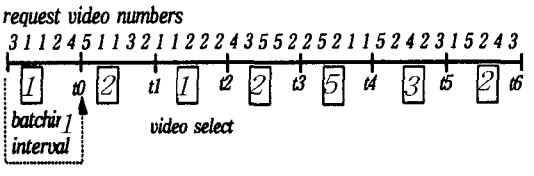

그림 3. MQL 스케줄링 실행의 예
Fig. 3. Example of MQL scheduling.

MQL 전략은 FCFS와는 다르게 서비스 요청의 빈도가 높을수록 서비스를 자주 해주게 되어 FCFS에 비하여 상대적으로 평균 대기 시간을 감소할 수 있지만, 그림 3의 비디오 4와 같이 최악의 경우에 요청 빈도가 낮은 비디오는 서비스 대기시간이 너무 길어져

버리는 문제가 발생할 수 있다.

Ⅲ. 적응적 배칭 전략

본 논문에서 제안된 적응적 배칭 전략(Adaptive Batching Scheme: ABS)의 특징은 첫째, 비디오 인기도와 시간대에 따라 변화될 수 있는 비디오 요구 패턴에 동적으로 작용하고, 둘째, 상호 연관되는 서비스 환경에 대해서도 적응적으로 대처하여, 서비스의 공정성 확보 및 평균 대기 시간을 효과적으로 단축하는데 있다. 적응적 배칭의 주요 특징은 동적 다중 선택(Dynamic multiple selection)과 적응적 스케줄링(Adaptive scheduling)이라 할 수 있다.

적응적 배칭은 그림 4와 같이 서비스 요청 큐를 서비스 요청되는 비디오의 종류만큼 복수 개로 두고 최소 요구 빈도수 k이상인 비디오와 최대 대기시간을 초과한 비디오가 존재할 경우 이를 모두 선택한다. 물론, 요청 빈도 k는 서버의 자원 상태나 전반적인 비디오 요구 도착률 등 필요에 따라 수시로 변경할 수 있으며, 한 리그 내에 각 비디오의 요청 빈도가 모두 k개 미만일 경우, 원칙적으로는 선택이 불가하지만 서버의 자원이 충분할 경우필요에 따라 최소한 1개는 선택하게 할 수 도 있다.

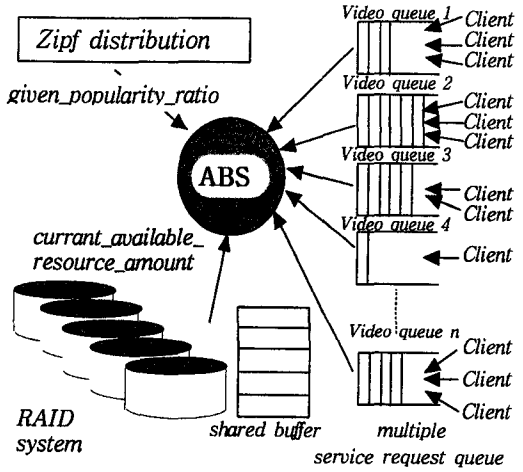

그림 4. ABS 스케줄링 모델
Fig. 4. ABS scheduling model.

1. 동적 다중 선택

비디오에 대한 인기도에 따라 배칭 간격의 크기를 결정하기 위하여 먼저 생각 할 수 있는 방식은 인기도 분포인 Zipf distribution을 활용한 방식이 있을 수 있다. I를 전체 비디오 수라 하면, 보통 비디오의 인기도를 나타내는 Zipf distribution함수 $f(i)$는 식 (5)와 같다. 식 (6)은 인기도에 대한 접근 확률의 편중도를 높이기 위하여 식 (5)을 약간 변형한 형태이며, θ 값은 0.271을 적용하는데, 비디오에 대한 사용자의 요청률을 나타낼 때 일반적으로 많이 사용된다[3].

$$f(i) = C/i \quad where\ C = 1/\sum(1/i) \ <for\ i=1\ to\ I> \tag{5}$$

$$f(i) = C/i^{(1-\theta)}\ where\ C = 1/\sum(1/i^{(1-\mu)}) \ <for\ i=1\ to\ I> \tag{6}$$

이를 이용하여 j번째 비디오에 대한 배칭 시간 간격 T_j는 식 (7)로 정리될 수 있는데 여기서 T_{max}는 사용자가 기다릴 수 있는 최대 값이며, $\sum f(n)$은 배칭에 포함되는 모든 비디오의 도착률의 합이 된다.[12]

$$T_j = T_{max} \cdot ((\sum f(n) - f(j)) / \sum f(n)) \tag{7}$$

그러나 이 방식은 해당 비디오의 인기도를 미리 알고 있어야 하고, 더욱이 수시로 변화되는 비디오의 요구 패턴(인기도) 및 도착률에 따라 그때, 그때 배칭 시간을 결정해야 한다는 점과 비 인기 비디오에 이 방식을 사용하였을 경우 배칭간격이 너무 길어져 버리는 문제가 발생한다. 그러므로 본 논문에서는 비디오의 인기도 및 요구 패턴을 스케줄링에 적용하기 위한 또 다른 방식을 제안한다. 적응적 배칭 스케줄링의 개념은 일정한 시간 간격을 기준으로 배칭 서비스를 수행하되, 비디오 요구 큐를 조사하여 최소 요구 빈도 수 k이상 요청된 비디오 n개를 모두 선택하는 방식이다. 본 논문에서는 빈도 수 k를 결정하기 위하여 두 가지 방법을 제안한다. 첫째, 일정 시간 간격 동안의 비디오 요청 패턴을 감시하고 이를 Zipf distribution의 인기도 분포를 참고로 휴리스틱한 방식으로 결정하는 방식과 둘째, 현재 서버의 자원 활용 상태에 따라 적응적으로 대처하는 방식이 있다. 이는 3.2절의 적응적 스케줄링 방안에서 자세히 언급한다. 또한 비 인기 비디오에 대한 서비스의 공정성 확보 방안 즉, 사용자의 최대 대기시간을 보장하는 문제는 배칭 스케줄링 시에 서비스 큐를 조사하여 최대 대기 시간이 초과된 비디오에 대하여 n개의 비디오 선택 범위에 추가로 포함시킨다. 동적 다중 선택을 실행하는 ABS에서 k_{min}을 서비스가 가능한 최소의 요구 빈도수(k)라 하고, b_{inv}을 배칭 간격, d_{max}를 최대 보장 대기 시간이라 할 때, k_{min} = 3, $d_{max} = b_{inv} * 3$으로 한 ABS의 스케줄링

실행 예가 그림 5에 있다.

요청 비디오 순서란의 *표시는 서비스 요청이 아직 도착하지 않은 경우이다. 배칭 순서 1에서는 k_{min}개 이상 요구된 비디오가 없으므로 아무 것도 선택되지 않았고, 배칭 순서 4에서 비디오 2는 최대 대기 시간이 초과된 경우이다. 배칭 순서 5에서는 비디오 3, 4가 모두 k_{min}개 이상 요청되었으므로 다중 선택되었다.

배칭 sequence	1					2					3					4					5				
요청 비디오 순서	1	2	*	4	*	1	4	1	*	*	4	1	3	1	3	*	4	1	4	2	*	3	4	2	1
FCFS					1					2					4					1					3
MQL					1					4					1					4					3
ABS										1					4					1 2					3 4

그림 5. 배칭스케줄링 실행의 예
Fig. 5. Examples of batching scheduling.

2. 적응적 스케줄링

적응적 스케줄링에서는 서비스 요청된 비디오들을 다중 선택하기 위한 기준이 되는 최소 요구 빈도 수 k, 즉 인기 있는 비디오를 결정하는 문제가 중요 시된다. 만일 k의 값이 너무 큰 경우에는 서비스 대기 시간이 길어지고, 반대의 경우에는 서버의 효율적인 자원 관리를 어렵게 한다. 또한 비디오 서비스 요구 도착률은 시간에 따라 변화 될 수 있기 때문에 k의 값은 현재의 서비스 요구 패턴 및 서버의 자원 현황에 따라 선택적이며 적응적으로 변화되어야 한다. 적응적 스케줄링을 위한 최소 요구 빈도 수 k를 결정하는 방식은 Zipf distribution의 인기도 분포를 기준으로 하는 방식과 자원의 사용 현황을 기준으로 하는 방식이 있을 수 있다.

(1) 인기도 분포를 기준

최소 요구 빈도 수 k를 결정하기 위하여 Zipf distribution의 인기도 분포를 참고하고 이를 휴리스틱한 방식으로 결정한다. 일반적으로 Zipf distribution[3]에 의하면 전체 요구 빈도 중 약 50% 이상이 특정한 비디오 (전체 비디오 수의 약 10~15%)에 편중되어 있다는 점을 감안하여, 이를 인기 비디오로 보고, 그 비율을 *given_popularity_ratio*이라 정한다. 그리고 스케줄링 시에 일정 시간 간격동안에 서버의 서비스 형태를 조사하여, 해당 시간동안의 서비스 중인 총 비디오 수에서 최대 대기 시간의 보장을 위하여 서비스중인 비디오 수를 제외한 값에서 그 시간 동안의 전체 서비스 요청된 비디오 수를 나눈 값을 n이라 하면 $n \fallingdotseq given_popularity_ratio$이 유지될 수 있게 최소 요구 빈도 수 k의 증감을 시도한다. 즉 $n > given_popularity_ratio$인 경우 k값을 증가시키고, $n < given_popularity_ratio$인 경우 k값을 감소시킨다. 또한 *given_popularity_ratio*의 값은 시스템의 서비스 환경에 따라 적절히 조정할 수 있다. 이 방식은 시스템의 활용 자원이 충분한 대규모 VOD서버에 적합한 방식이지만, 서버의 사용 자원에 제약이 있는 소규모 VOD 서버의 경우에는 현재의 자원 상태를 수시로 감시할 필요가 있다. 그림 6에서는 인기도 분포도를 기준으로 최소 요구빈도 수 k를 조정하기 위한 알고리즘 개요를 나타내었다.

```
Algorithm for Popularity
repeat
while (time-interval = null)  do skip;
   k:= previous_minimum_service_frequency;
   n:=(current_service_video_count-service_count_for_maximum_waiting_time)/
      (all_request_video_count);
   if n > given_popularity_ratio
     then k:= k + 1
     else
      if n < given_popularity_ratio
      then k := k - 1;
until false;
```

그림 6. 인기도 분포를 기준으로 한 k의 결정 알고리즘 개요.
Fig. 6. Determine k algorithm with video popularity.

(2) 서버의 자원 현황을 기준

서버의 자원 현황을 기준으로 하여 최소 요구 빈도 수 k를 결정하는 방식은 버퍼 및 디스크의 사용 가능율이 미리 결정된 한계치 미만이 될 경우, 즉 서버 가용 자원의 부족이 예상 될 경우, 최소 요구 빈도수 k를 증가시키며, 반대의 경우에는 이를 감소시킨다.

현재 버퍼, 디스크 대역폭등 사용 가능한 자원 정도를 *current_available_resource_count*라 하고, 자원

의 사용 한계치를 *lower_limit_resource_count, upper_limit_resource_count* 라 하면, 사용가능 자원이 사용 한계치를 지나치게 초과하지 못하도록 k의 값을 조정한다. 또한 사용 한계치는 시스템 서비스 환경에 따라 조정할 수 있다. 그림 7에서는 자원 활용을 기준으로 최소 요구빈도 수 k를 조정하기 위한 알고리즘 개요를 나타내었다.

```
Algorithm for Resource
repeat
while (time-interval = null)  do skip;
   k:= previous_minimum_service_frequency;
   if current_available_resource_count < lower_limit_resource_count
     then k:= k + 1
     else
   if currant_available_resource_count > upper_limit_resource_count
     then k := k - 1;
until false;
```

그림 7. 자원 활용률을 기준으로 한 k의 결정 알고리즘 개요

Fig. 7. Determine k algorithm with resource utilization.

Ⅳ. 시뮬레이션 및 분석

본 논문에서 제안한 효율적 서비스를 위한 적응적 배칭 정책의 주요 관심은 주어진 서버의 능력 하에서 서비스 평균 대기 시간의 단축과 서비스 최대 대기 시간 보장에 있으며, 이를 시뮬레이션 하기 위하여 사용자가 요청한 비디오를 서비스 받기까지 평균 대기 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 λ는 1sec당 한개의 평균 도착률을 갖는 Poisson Distribution으로 하였다. ABS 스케줄링을 위한 최대 대기 보장 시간은 600sec, k값은 서버의 현재 서비스 현황에 따라 변화하므로 특별한 의미는 없으나, 초기 k값은 다른 스케줄링(FCFS, MQL)의 한번 배칭에 의해서 스케줄링 되는 비디오 개수의 평균치에 가까운 6으로 하였다. 또한 Zif Distribution에 의한 인기도 분포는 10~15%이므로, 본 시뮬레이션에서는 임의로 *given_popularity_ratio* 를 10%로 하였다. 그림 8은 시뮬레이션에 사용된 100개의 비디오에 대한 요구 패턴을 나타낸다.

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

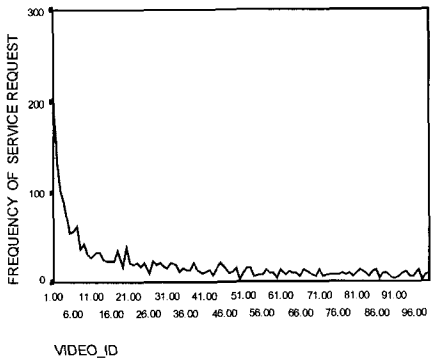

그림 8. 비디오 요구 패턴

Fig. 8. Request video pattern.

1. 서비스 대기 시간 분석

그림 9, 10, 11, 12 및 표 2는 스케줄링별 서비스 대기시간을 분석한 결과이다. 스케줄링 종류는 FCFS, MQL, 인기 분포를 감안한 ABS1, 서버 자원 현황을 감안한 ABS2이다. 그림 9, 10은 배칭 단계별 스케줄링 당 평균 대기시간의 변화를 나타내었으며, 각 그래프의 W_FCFS, W_MQL, W_ABS1, W_ABS2는 각각 FCFS, MQL, ABS1(인기도), ABS2(자원)의 평균대기시간이며, 그래프 및 표에 있는 모든 시간의 단위는 10msec(0.1sec)이다. 평균 대기 시간에서 ABS1(자원), ABS2(인기)는 FCFS, MQL 보다 20~30% 정도 짧게 나타나고 있다.

또한 서비스 지연에서 오는 사용자의 서비스 요청 취소와 밀접한 연관이 있는 최대 서비스 대기 보장 시간에 대하여 이를 600sec으로 하였을 때, 최대 보장 시간을 넘는 서비스 건수가 그림 11에 나타나 있다.

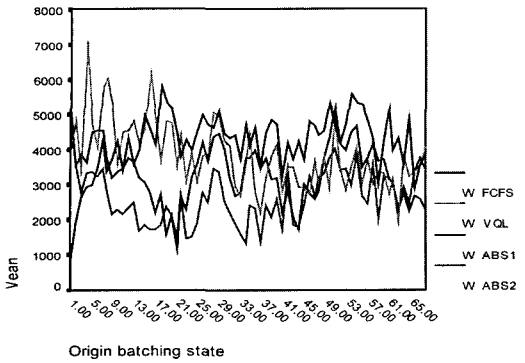

그림 9. 배칭 단계별 평균 대기 시간
Fig. 9. Batching sequence and mean of waiting time.

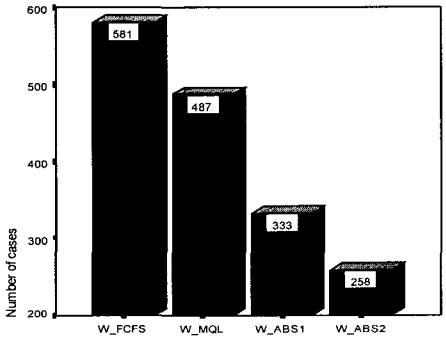

그림 10. 스케줄링별 평균 대기 시간
Fig. 10. Scheduling policies and mean of waiting time.

그림 11에서 총 위배 건수 면에서는 ABS1, ABS2 스케줄링이 다른 스케줄링에 비해 50%정도 단축하고 있으나, 모두 대기 시간을 위배하므로 좋은 결과로 보기 어렵다. 그림 11의 시뮬레이션 결과를 조금 세밀히 분석하기 위하여, 그림 11의 결과를 그대로 두고 서비스 대기 시간이 650sec을 초과하는 건수를 계산해 보면, 그림 12에서 보는 바와 같이 FCFC나 MQL의 경우 별다른 변화가 없지만, 두개의 ABS 스케줄링에서는 초과 건수가 모두 없어졌음을 볼 수 있다. 이는 ABS 스케줄링에서 최대 보장 시간(600sec)을 초과한다 하더라도, 최대 보장 시간 ±10% 안에서 모두 서비스됨을 알 수 있다. 그 이유는 배칭 시간 간격 안에서 어느 사용자가 현재 최대 보장 시간을 초과하였다 하더라도 그 사용자는 다음 배칭 스케줄링 시간까지는 기다려야한다. 그러므로 ABS 스케줄링에서 정확한 최대 서비스 보장 시간은 초기에 결정한 최대 보장 시간에 배칭 시간 간격을 합한 시간이 됨을 알 수 있다.

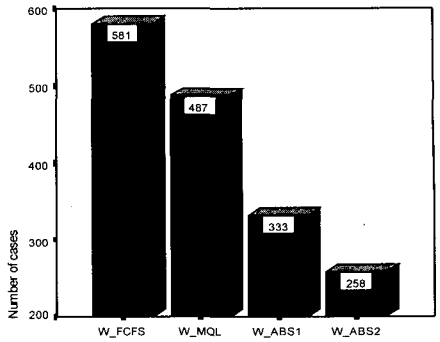

그림 11. 600sec을 초과 한 건수(최대 대기 시간 보장 시간 600sec)
Fig.11. Frequencies over 600sec.(Guaranteed maximum waiting time 600sec).

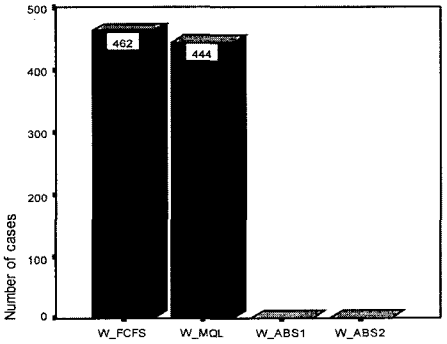

그림 12. 650sec을 초과한 건수(최대 대기 보장 시간 600sec)
Fig. 12. frequencies over 650sec.(Guaranteed maximum waitsing time 600sec).

2. 자원 사용도 분석

본 논문에서 제안된 적응적 배칭 전략(ABS)의 자원 사용도를 다른 스케줄링과 분석한 결과를 그림 13, 14 및 표 3에 표시하였다. 정확한 자원 사용도를 계산하기 위해선 현재 서버의 디스크 스케줄링과 버퍼 스케줄링에 관한 전반적인 자원 관리 정책이 감안되어야 하지만, 본 논문의 관심은 사용자에게 효율적인 서비스를 하기 위한 평균 대기 시간 단축 및 서비스 최대 대기 시간의 보장에 있고, 일반적으로 서버가 처리하여야 할 프로세스의 개수와 자원의 사용도는 비례한다 할 수 있으므로, 본 논문에서는 스케줄링별 현재 서버 내의 프로세서 개수를 기초로 하였다. 그리고 자원 현황을 감안한 ABS2 스케줄링에서는 초기에 적정 자원의 수(총 자원의 60~70%)를 입력하는데, 적정 자원 수의 변화에 따른 실제 사원 사용도 및 대기 시간의

관계를 알아보기 위해, 시뮬레이션 마지막 부분에서 적정 자원의 수를 200으로 하였을 경우(ABS3)과 300으로 하였을 경우(ABS2)로 각각 나누어서, 자원 사용에 따른 두 스케줄링간의 대기시간 및 자원 사용도를 비교 분석하여 보았다. 먼저 FCFS, MQL, ABS1(인기도 중심), 300을 적정 자원 수로 한 ABS2(자원 중심)의 자원 사용 현황을 각각 R_FCFS, R_MQL, R_ABS1, R_ABS2로 표시한 결과가 그림 13, 14에 있다. 먼저 그림 13, 14에서 평균자원의 사용은 R_ABS1 즉 인기도 중심 ABS가 가장 좋은 결과를 나타내고 있으며, 배칭 단계별 사용 자원에서는 R_ABS2를 제외한 세 개의 스케줄링이 비슷한 사용율을 보이고 있다. 마지막으로 R_ABS2의 평균 점유 자원이 159로 가장 높은 이유는 초기에 적정 자원 량을 300으로 하였기 때문이며, 대신에 전체적인 평균 대기시간은 표2에서 보는 바와 같이 가장 낮게 나타나고 있다. 여기서 다시 적정 자원을 200으로 한 ABS3의 경우와 비교 한 결과가 표 3에 있다. 적정 자원을 200 (ABS3)으로 하면 자원의 사용도가 급격히 줄어드는 대신에 대기 시간이 늘어남을 볼 수 있다.

표 2. 스케줄링별 대기 시간 비교표
Table.2. A comparative table of waiting time by scheduling polices.

	Wait_FCFS	Wait_MQL	Wait_ABS1	Wait_ABS2
Valid	2000	2000	2000	2000
Mean	4201.2605	3851.7605	3329.9605	2630.8105
S.D	2509.0600	4633.2404	2033.4491	2165.4727
Min	1.00	.00	1.00	1.00
Max	8646.00	25117.00	6299.00	6299.00

표 3. 한계 자원을 200, 300으로 한 경우 ABS(자원 중심)의 자원 사용
Table.3. Resource utilization of ABS with limited 200 or 300.

	Wait_ABS2	Wait_ABS3	Resource_ABS2	Resource_ABS3
N	2000	2000	2000	2000
Mean	2630.8105	4418.9605	159.7430	98.6330
S.D	2165.4727	4122.1931	97.6768	66.3778
Min	1.00	.00	.00	.00
Max	6299.00	12296.00	345.00	257.00

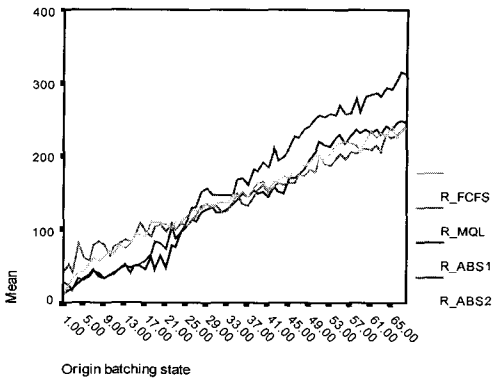

그림 13. 배칭 단계별 자원 사용도
Fig. 13. Resource utilization of batching sequence.

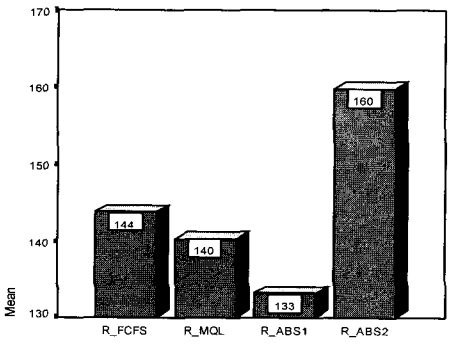

그림 14. 평균 자원 사용도
Fig. 14. Mean of resource utilization.

V. 결 론

일괄 수용 서비스 환경에서 배칭 정책을 실행할 경우에 사용자의 서비스 평균 대기 시간 단축과 서비스의 공정성 확보(최대 대기 시간 보장)를 실현하기 위하여 본 논문에서 제안된 적응적 배칭 정책은 사용자의 서비스 요구패턴과 자원의 사용율에 동적으로 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)나 현재의 자원 활용 정보에 따라 그때그때 적응적으로 산출되어지는 최소 허락 빈도 수 인 k를 활용하여 배칭 간격 내 에서 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 20~30% 정도의 평균대기 시간의 감소를 보였으며, 서비스 취소 여부를 판단하는 최대 서비스 보장 시간에서도 ±10% 범위 안에서 모두 충족시키는 것으로 나타났다. 또한 중요한 요소라고 할 수 있는 자원의 사용도면에서 기존 스케줄링과 별 차이가 없음을 보였다. 따라서 적응적 배칭 전략은 서비스 요구패턴이 다양하게 변하고 자원 활용이 엄격하게 정의된 VOD 시스템에서 유리한 방식이라 할 수 있다. 앞으로의 연구 방향은 적응적 배칭 정책의 서비스 효율성을 더욱 증대시키고 서버의 서비스 수용 능력의 증대를 하여, 적응적 배칭 전략에 디스크 및 버퍼 스케줄링 정책을 연계한 일관화 된 통합 스케줄링 정책을 마련하는 것이다.

참 고 문 헌

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

[6] 박호균, 유황빈 " 주문형 비디오 서비스 시스템에서 VCR 기능을 위한 Batching 전송", *한국통신 학회*, vol.22, no.12, pp. 2852-2859, 1997

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

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

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

[10] Kurt Rothermel, Tobias Helbig, "An adaptive protocol for synchronizing media streams", *Multimedia Systems*, pp. 324-336, 1997.

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

[12] Hong-ki Jung, Seung-Kyu Park,"Grouping and Buffer Management Methods in a VOD Server" *Proceeding of ITC-CSCC '99, Sado, Niigata, Japan*, pp. 899-902, July 13-15, 1999.

저 자 소 개

崔 成 旭(正會員)

1984년 광운대학교 이공대학 전산학 (이학사). 1987년 경희대학교 대학원 전산학 (공학 석사). 1997년 아주대학교 대학원 컴퓨터공학 (박사과정 수료). 1992년~현재 시립 인천 전문 대학 전산과 부교수. 주관심분야는 멀티미디어 응용 및 시스템 구조, 소프트웨어 시스템 디자인,이동 컴퓨팅 시스템 등

金 鍾 景(正會員)

1990년 호원 대학교 전산학 (공학사). 1993년 경희대학교 산업정보 대학원 전산학 (공학 석사). 1999년 아주대학교 대학원 컴퓨터공학 (박사 수료). 1999년~현재 원 인포텍 대표. 주관심분야는 멀티미디어 응용 및 시스템 구조, 소프트웨어 시스템 디자인 등

朴 升 圭(正會員)
1974년 서울대학교 공과대학 응용수학(공학사). 1976년 한국과학원(KAIST) 전산학(석사). 1982년 프랑스 INP de Grenoble 전산학 (공학박사). 976년~1977년 한국과학기술연구소(KIST) 연구원. 1978년~1992년 한국전자통신연구소(ETRI) 부장. 1984년 3월~1985년 9월 미국 IBM 왓슨 연구소 연구원. 1992년~현재 아주대학교 정보 및 컴퓨터공학부 교수. 주관심분야는 멀티미디어 처리구조, 다중처리및 MPP 시스템, 실시간시스템, 이동컴퓨팅 시스템 등

金 東 潤(正會員)
1974년 서울대학교 문리대 수학과(학사). 1976년 한국과학원(KAIST) 전산학(석사). 1985년 Massachusetts Institute of Technology 응용수학박사. 1976~1991년 국방과학연구소 연구원. 1988~1989년 Maryland대 Computer Vision Lab SEP fellow. 1991년~현재 아주대 정보및컴퓨터공학부 교수. 1994~1995년 Cambridge대 인촌기념교수. 주관심분야는 Computer Vision, Algorithm, Simulation

崔 景 熙(正會員)
1976년 서울대학교 사범대학 수학교육과(학사). 1979년 그랑제꼴 ENSEEIHT(엔지니어, 공학석사). 1982년 Paul Sabtier대(공학박사). 1982년~현재 아주대학교 정보및컴퓨터공학부 교수. 주관심분야는 운영체제, 분산시스템, 실시간시스템, 프로그래밍방법론

崔 德 圭(正會員)
1966년 서울대학교 공대 원자력공학과(석사). 1984년 Wright State University 전산과(석사). 1989년 University of Massachusetts 전산학 박사. 1993년~현재 아주대학교 정보및컴퓨터공학부 교수.주관심분야는 고속 LAN, ATM and WATM protocol, 이동통신네트워크