(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년01월15일
(11) 등록번호 10-0794969
(24) 등록일자 2008년01월09일

(51) Int. Cl.
G06F 15/16 (2006.01) *G06F 17/30* (2006.01)
(21) 출원번호 10-2007-0014180
(22) 출원일자 2007년02월12일
심사청구일자 2007년02월12일
(56) 선행기술조사문헌
KR100231220 B1

(73) 특허권자
인하대학교 산학협력단
인천 남구 용현동 253 인하대학교
(72) 발명자
송민석
서울 서초구 방배3동 래미안 아트힐 107동 402호
(74) 대리인
김건우

전체 청구항 수 : 총 2 항 심사관 : 노영철

(54) 클러스터링된 비디오 서버에서의 부하 공유를 위한 적응형데이터 검색 방법

(57) 요 약

본 발명은 클러스터링된 비디오 서버에서의 부하 공유를 위한 적응형 데이터 검색 방법으로서, 보다 구체적으로는 라운드 길이를 동적으로 변화시킴으로써 데이터의 이동 또는 복사 없이도 클러스터들 간에 부하 공유를 달성할 수 있는 적응형 데이터 검색 방법에 관한 것이다. 본 발명은, 고객이 새롭게 비디오 스트림을 요청하는 경우, 해당 비디오 스트림을 포함하는 클러스터에 대하여 라운드 길이를 확장시킬 수 있는지 여부를 판단하는 단계와, 라운드 길이를 확장시킨 경우의 디스크 대역폭 이용도를 통해 지터가 없는 라운드 길이의 확장이 가능한지를 판단하는 단계와, 이 때 버퍼 이용도가 1보다 작은지를 확인하여 충분한 버퍼 공간이 있는지를 판단하는 단계와, 실제로 라운드 길이를 확장하는 단계를 포함하는 것을 그 특징으로 한다. 또한, 본 발명은, 라운드 길이가 최대로 설정되어 있는 클러스터의 디스크 이용도들의 집합으로부터 값이 가장 작은 원소를 선택하여 제거하는 단계와, 이 경우에 디스크 대역폭 이용도 제약을 만족하는지를 판단하는 단계와, 디스크 대역폭 이용도 제약을 만족하는 것으로 판단되면 라운드 길이를 감소시키고, 그렇지 않으면 상기 제거 단계 및 판단 단계를 반복하는 단계를 포함하는 것을 그 특징으로 한다.

본 발명에 따르면, 라운드 길이를 동적으로 변화시키는 적응형 데이터 검색 방법을 제안함으로써, 데이터의 이동 또는 복사 없이도 클러스터들 간에 부하 공유를 달성할 수 있다.

대표도 - 도4

Algorithm 1 Admission Decision Procedure

Set of utilizations of $DS_k(j)$ and $BS_k(j)$, $(j = 1, ..., ND)$
A client requests a stream from cluster p, and the server recalculates $DS_p(j)$ and $BS_p(j)$;
A set of disk utilizations: $SM = \{DS_m(ND) | m = 1, ..., C \text{ and } SL_m = ND\}$;
BOOLEAN: FLAG ← FALSE;
if $SL_p \leq ND - 1$ and $\frac{r_{SL_p}}{r_{SL_p+1}} - \frac{\epsilon}{r_{SL_p}} < DS_p(SL_p + 1) \leq \frac{r_{SL_p}}{r_{SL_p+1}}$ **then**
  if $\sum_{k=1}^{C} BS_k(SL_k) + BS_p(SL_p + 1) - BS_p(SL_p) \leq 1$ **then**
    $SL_p \leftarrow SL_p + 1$; { Round length increases }
  end if
end if
if $DS_p(SL_p) \leq 1$ **then**
  if $\sum_{k=1}^{C} BS_k(SL_k) \leq 1$ **then**
    FLAG ← TRUE;
  else
    while FLAG = FALSE and $SM \neq \phi$ **do**
      Find the lowest value, $DS_l(ND) \in SM$;
      $SM \leftarrow SM - \{DS_l(ND)\}$;
      if $DS_l(ND - 1) \leq 1$ **then**
        $SL_l \leftarrow ND - 1$;
        FLAG ← TRUE;
      end if
    end while
  end if
end if
if FLAG = TRUE **then**
  The new client passes the admission test;
else
  The new client is rejected;
end if

특허청구의 범위

청구항 1

전송 속도(transfer rate)가 tr이고, 디스크 검색 시간(disk seek time)이 T_s로서 각각 상수이며, 버퍼 크기는 B이고,

시간이 라운드(rounds)라고 지칭되는 동일한 크기의 주기들로 분할되며, 각각의 인가된 고객은 각각의 라운드에서 한 번씩 서비스를 받게 되는 라운드-기반 스케줄링(round-based scheduling)이 사용되며,

비디오 스트림 V_i가 유한 개수의 연속적인 세그먼트들로 나누어지고, 각각의 세그먼트들은 NS개의 서브-세그먼트들로 다시 나누어지며(여기서, 각각의 서브-세그먼트의 크기는 길이 BR인 기본 라운드 동안 검색되는 데이터 크기에 해당함), 각각의 세그먼트들은 라운드-로빈 방식으로 디스크들에 저장되며, 데이터 재생 속도는 dr_i로서 그 단위는 bits/sec이며,

C개의 클러스터로 구성되고 각각의 클러스터는 Q개의 동종 디스크들로 구성되고, 고객들을 C개의 고객 그룹(CG_1, …, CG_C)으로 분할한 후, 그룹 CG_k 내의 고객들이 대응하는 클러스터 k로부터 스트림을 수신 받도록 구성되며,

고객 CL_i^m 이 상기 비디오 스트림 V_i를 요청하고 있는 클러스터링된 비디오 서버에서,

라운드 길이가 fr_j인 경우의 클러스터 k에 대한 디스크 대역폭 이용도 $DS_k(j)$와 버퍼 이용도 $BS_k(j)$가 각각 다음 수학식 1 및 2와 같이 주어지는 경우에,

<수학식 1>

$$DS_k(j) = \frac{ST_k(j)}{fr_j \times Q}$$

$$ST_k(j) = \sum_{CL_i^m \in CG_k} (T_s + fr_j \times \frac{dr_i}{tr})$$

(여기서,)

<수학식 2>

$$BS_k(j) = \sum_{CL_i^m \in CG_k} \frac{2 \times fr_j \times dr_i}{B}$$

클러스터링된 비디오 서버에서 부하 공유를 위해 데이터를 적응적으로 검색하는 방법으로서,

(1) 모든 클러스터 k(k = 1, …, C)에 대하여 상기 $DS_k(j)$와 $BS_k(j)$의 값을 계산하여 그 집합들을 각각 유지하는 단계;

(2) 가능한 라운드 길이를 오름차순으로 포함하고 있는 집합(FS)의 SL_m번째 원소, fr_{SLm}가 클러스터 m에 대한 라운드 길이로서 선택되는 것을 나타내는 선택 파라미터 SL_m이 ND인 경우의 디스크 이용도 $DS_m(ND)$(m = 1, …, C)들의 집합 SM을 구하는 단계;

(3) 고객이 클러스터 p에 저장된 비디오 스트림을 요청하는 경우, 새로운 고객이 인가될 것이라고 가정하고 j = 1, …, ND에 대하여 상기 클러스터 p에 대한 이용도들 $DS_p(j)$와 $BS_p(j)$를 다시 계산하는 단계 - 여기서, ND는 상기 NS에 대한 약수의 개수임 - ;

(4) 상기 클러스터 p에 대한 선택 파라미터 SL_p가 $SL_p \leq ND - 1$인지를 확인하여, 상기 클러스터 p에 대하여 라운드 길이가 확장될 수 있는지 여부를 판단하는 단계;

(5) 상기 클러스터 p에 대한 선택 파라미터를 증가시킬 경우의 디스크 대역폭 이용도 $DS_p(SL_p + 1)$이 다음 수학식 3을 만족하는지 여부를 확인하여 지터가 없이 라운드 길이의 확장이 가능한지를 판단하는 단계;

<수학식 3>

$$\frac{fr_{SL_p}}{fr_{SL_p+i}} - \frac{\epsilon}{fr_{SL_p}} < DS_p(SL_p + 1) \leq \frac{fr_{SL_p}}{fr_{SL_p+i}}$$

(여기서, ε는 고객그룹 CG_i에 속하는 고객 CL_i^m에 대한 최대 서비스 시간을 나타냄)

(6) 상기 단계 (4) 내지 (5)에서 라운드 길이가 지터가 없이 확장 가능한 것으로 판단되는 경우, 상기 클러스터 p에 대한 라운드 길이를 확장한 경우의 버퍼 이용도가 1보다 작은지를 다음 수학식 4에 의해 확인하여 충분한 버퍼 공간이 있는지를 판단하는 단계;

<수학식 4>

$$\sum_{k=1}^{C} BS_k(SL_k) + BS_p(SL_p + 1) - BS_p(SL_p) \leq 1$$

(7) 상기 단계 (6)에서 버퍼 공간이 충분한 것으로 판단되는 경우, SL_p를 SL_p + 1로 증가시켜 라운드 길이를 확장시키는 단계;

(8) SM에서 값이 가장 작은 원소 DS(ND)을 선택하여 SM로부터 제거하는 단계;

(9) 클러스터 1에 대하여 라운드 길이를 줄인 경우의 디스크 대역폭 이용도인 $DS_1(ND - 1)$이 1이하인지를 확인하여 디스크 대역폭 이용도를 만족시키는지 확인하는 단계; 및

(10) 상기 단계 (9)에서 $DS_1(ND - 1) \leq 1$ 인 것으로 판단되면 SL_1을 ND - 1로 하여 라운드 길이를 감소시키고, 그렇지 않으면 SM이 공집합이 아닌 동안 상기 단계 (8) 내지 (9)를 반복하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

(11) 고객이 클러스터 k에 저장된 비디오 스트림을 닫는 경우, $DS_k(j)$와 $BS_k(j)(j = 1, \cdots, ND)$를 다시 계산하는 단계;

(12) 다음 수학식 5를 만족하는지 판단하는 단계;

<수학식 5>

$$DS_k(SL_k) \leq \frac{fr_{SL_k-1}}{fr_{SL_k}} - \frac{\epsilon}{fr_{SL_k-1}} \quad (SL_k = 2, \cdots, ND)$$

(13) 상기 단계 (12)에서 상기 수학식 5를 만족하는 것으로 판단되면, 다음 수학식 6을 만족하는지 확인하여 라운드 길이를 감소시키는 것이 디스크 대역폭 제약을 위반시킬 것인지를 판단하는 단계; 및

<수학식 6>

$$DS_k(SL_k - 1) \leq 1$$

(14) 상기 단계 (13)에서 디스크 대역폭 제약을 위반하지 않는 것으로 판단되면, SL_k를 SL_k - 1로 감소시켜 클러스터 k의 라운드 길이를 감소시키는 단계

를 더 포함하는 방법.

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<16> 본 발명은 클러스터링된 비디오 서버에서의 부하 공유를 위한 적응형 데이터 검색 방법으로서, 보다 구체적으로는 라운드 길이를 동적으로 변화시킴으로써 데이터의 이동 또는 복사 없이도 클러스터들 간에 부하 공유를 달성할 수 있는 적응형 데이터 검색 방법에 관한 것이다.

<17> 멀티미디어 및 네트워크 기술에서의 최근의 진보로 인해, 주문형 비디오(video-on-demand 이하 "VOD"라고 함) 서비스를 고객들에게 제공하는 것이 가능하게 되었다. VOD 서비스가 가능하려면, 비디오 서버가 수천 명의 고객들에게 비디오를 전송할 수 있는 능력을 가져야 한다. 비디오 데이터는 높은 대역폭 및 큰 저장 공간을 필요로 하기 때문에, 비디오 서버는 일반적으로 수백 개의 디스크들로 구성될 수 있는 디스크 어레이들 상에 만들어진다. 서버의 디자인에 있어서 주된 관심사는, 동시 발생하는 스트림의 수를 최대화시킬 수 있도록 비디오 데이터를 어떻게 위치시키고 검색하느냐는 것이다.

<18> 다수의 고객들을 지원하기 위한 데이터 검색 기법으로서는, 라운드 기반 스케줄링(round-based scheduling) 기법이 일반적으로 사용된다. 라운드 기반 스케줄링에서는, 시간이 라운드(rounds)라고 지칭되는 동일한 크기의 주기들로 분할되며, 각각의 인가된 고객은 각각의 라운드에서 한 번씩 서비스를 받게 된다. 연속적인 재생을 보장하기 위해, 현재 재생이 요구되는 데이터는 이전 라운드에서 디스크로부터 판독되는 반면에, 다음 라운드에서 재생되어야 할 데이터는 현재 라운드에서 판독된다. 그렇지 않으면, 재생이 왜곡되거나, 비디오 데이터의 시간제한 위반 때문에 일시 정지(pause)가 발생할 수 있는데, 이 현상을 지터(jitter)라고 지칭한다.

<19> 디스크 대역폭을 효과적으로 사용하기 위하여, 비디오 오브젝트는 흔히 세그먼트로 분할되어 다수의 디스크에 걸쳐 분배된다. 이러한 스킴은 스트라이핑(striping)이라고 지칭되는데, 세그먼트는 하나의 디스크 상에 저장되는 연속적인 데이터의 최대 양이다. 너무 많은 디스크에 걸친 스트라이핑은 추가적인 복잡도(complexity)를 수반하게 되며, 그 결과 신뢰도 문제를 초래할 수 있다. 따라서 하나의 디스크 어레이를 몇 개의 클러스터로 분할하여 스트라이핑의 범위를 제한한 후, 각각의 클러스터가 독립적으로 스트라이핑 그룹을 구성하게 하고 나서, 각각의 비디오가 하나의 클러스터 내에 스트라이핑 되도록 하는 것이 현실적이다.

<20> 일반적으로 고객들의 요청은 소수의 인기 비디오들로 지나치게 치우치는 경향이 있어서, 해당 인기 비디오들의 사본을 포함하고 있는 클러스터들은 다른 클러스터들보다 더 많은 요청을 수신하게 되어 과부하가 걸리기 쉽다. 미래의 고객들에 의해 요청될 비디오에 대한 접근 확률을 정확하게 예측하는 것은 어렵기 때문에, 비디오 데이터를 균형 잡힌 방식으로 배치하는 것은 불가능하다. 더욱 어려운 점은, 요청 패턴들이 시간 단위로 변할 수도 있다는 것이다. 예를 들어, 아동용 비디오는 흔히 초저녁에는 인기가 있으나, 늦은 밤에는 거의 요청되지 않는다.

<21> 클러스터링된 비디오 서버에서 부하 밸런싱(load balancing)을 달성하기 위해서, 기존의 접근법들은 동적 데이터 이동(dynamic data migration) 또는 복사(replication) 기법을 사용한다. 이들 스킴의 주된 아이디어는, 클러스터들 사이에서 비디오 데이터를 이동시키거나 복사하여, 일정 정도의 부하 균형을 달성한다는 것이다. 그러나 이들 스킴은 이동 또는 복사를 위해 추가적인 디스크 대역폭 및 저장 공간과 같이 보다 많은 리소스를 필요로 하며, 이로 인해 중단(downtime)을 야기하거나 VOD 시스템을 액세스하는 애플리케이션들의 성능에 영향을 미칠 수 있다. 게다가, 이러한 이동은 고객들의 미래 수요의 평가에 기초할 필요가 있는데, 부정확한 예측은 시스템 성능을 크게 떨어뜨릴 수도 있다. 그러므로 이들 접근법은 비디오 서버에 그다지 적합하지 않다.

발명이 이루고자 하는 기술적 과제

<22> 본 발명은 상기와 같은 인식에서 비롯된 것으로서, 데이터의 이동 또는 복사 없이도 클러스터들 간에 부하 공유를 달성할 수 있는 적응형 데이터 검색 방법을 제안하는 것을 그 목적으로 한다.

발명의 구성 및 작용

<23> 상기한 목적을 달성하기 위한 본 발명의 특징에 따른, 클러스터링된 비디오 서버에서의 부하 공유를 위한 적응형 데이터 검색 방법은 다음과 같다.

<24> 전송 속도(transfer rate)가 tr이고, 디스크 검색 시간(disk seek time)이 T_s로서 각각 상수이며, 버퍼 크기는 B이고,

<25> 시간이 라운드(rounds)라고 지칭되는 동일한 크기의 주기들로 분할되며, 각각의 인가된 고객은 각각의 라운드에서 한 번씩 서비스를 받게 되는 라운드-기반 스케줄링(round-based scheduling)이 사용되며,

<26> 비디오 스트림 V_i가 유한 개수의 연속적인 세그먼트들로 나누어지고, 각각의 세그먼트들은 NS개의 서브-세그먼트들로 다시 나누어지며(여기서, 각각의 서브-세그먼트의 크기는 길이 BR인 기본 라운드 동안 검색되는 데이터 크기에 해당함), 각각의 세그먼트들은 라운드-로빈 방식으로 디스크들에 저장되며, 데이터 재생 속도는 dr_i로서 그 단위는 bits/sec이며,

<27> C개의 클러스터로 구성되고 각각의 클러스터는 Q개의 동종 디스크들로 구성되고, 고객들을 C개의 고객 그룹(CG_1, …, CG_C)으로 분할한 후, 그룹 CG_k 내의 고객들이 대응하는 클러스터 k로부터 스트림을 수신 받도록 구성되며, 고객 CL_i^m 이 상기 비디오 스트림 V_i를 요청하고 있는 클러스터링된 비디오 서버에서,

<28> 라운드 길이가 fr_j인 경우의 클러스터 k에 대한 디스크 대역폭 이용도 $DS_k(j)$와 버퍼 이용도 $BS_k(j)$가 각각 다음 수학식 1 및 2와 같이 주어지는 경우에,

수학식 1

<29>
$$DS_k(j) = \frac{ST_k(j)}{fr_j \times Q}$$

$$ST_k(j) = \sum_{CL_i^m \in CG_k} (T_s + fr_j \times \frac{dr_i}{tr})$$

<30> (여기서,)

수학식 2

<31>
$$BS_k(j) = \sum_{CL_i^m \in CG_k} \frac{2 \times fr_j \times dr_i}{B}$$

<32> 클러스터링된 비디오 서버에서 부하 공유를 위해 데이터를 적응적으로 검색하는 방법으로서,

<33> (1) 모든 클러스터 k(k = 1, …, C)에 대하여 상기 $DS_k(j)$와 $BS_k(j)$의 값을 계산하여 그 집합들을 각각 유지하는 단계;

<34> (2) 가능한 라운드 길이를 오름차순으로 포함하고 있는 집합(FS)의 SL_m번째 원소, fr_{SLm}가 클러스터 m에 대한 라운드 길이로서 선택되는 것을 나타내는 선택 파라미터 SL_m이 ND인 경우의 디스크 이용도 $DS_m(ND)$(m = 1, …, C)들의 집합 SM을 구하는 단계;

<35> (3) 고객이 클러스터 p에 저장된 비디오 스트림을 요청하는 경우, 새로운 고객이 인가될 것이라고 가정하고 j = 1, …, ND에 대하여 상기 클러스터 p에 대한 이용도들 $DS_p(j)$와 $BS_p(j)$를 다시 계산하는 단계 - 여기서, ND는 상기 NS에 대한 약수의 개수임 - ;

<36> (4) 상기 클러스터 p에 대한 선택 파라미터 SL_p가 $SL_p \leq ND - 1$인지를 확인하여, 상기 클러스터 p에 대하여 라운드 길이가 1만큼 확장될 수 있는지 여부를 판단하는 단계;

<37> (5) 상기 클러스터 p에 대한 선택 파라미터를 증가시킬 경우의 디스크 대역폭 이용도 $DS_p(SL_p + 1)$이 다음 수학식 3을 만족하는지 여부를 확인하여 지터가 없이 라운드 길이의 확장이 가능한지를 판단하는 단계;

수학식 3

<38> $$\frac{fr_{SL_p}}{fr_{SL_p+1}} - \frac{\epsilon}{fr_{SL_p}} < DS_p(SL_p + 1) \leq \frac{fr_{SL_p}}{fr_{SL_p+1}}$$

<39> (여기서, ε는 고객그룹 CG$_p$에 속하는 고객 CL_i^m 에 대한 최대 서비스 시간을 나타냄)

<40> (6) 상기 단계 (4) 내지 (5)에서 라운드 길이가 지터가 없이 확장 가능한 것으로 판단되는 경우, 상기 클러스터 p에 대한 라운드 길이를 확장한 경우의 버퍼 이용도가 1보다 작은지를 다음 수학식 4에 의해 확인하여 충분한 버퍼 공간이 있는지를 판단하는 단계;

수학식 4

<41> $$\sum_{k=1}^{C} BS_k(SL_k) + BS_p(SL_p + 1) - BS_p(SL_p) \leq 1$$

<42> (7) 상기 단계 (6)에서 버퍼 공간이 충분한 것으로 판단되는 경우, SL_p를 SL_p + 1로 증가시켜 라운드 길이를 확장시키는 단계;

<43> (8) SM에서 값이 가장 작은 원소 DS(ND)을 선택하여 SM로부터 제거하는 단계;

<44> (9) 클러스터 l에 대하여 라운드 길이를 줄인 경우의 디스크 대역폭 이용도인 $DS_l(ND - 1)$이 1이하인지를 확인하여 디스크 대역폭 이용도를 만족시키는지 확인하는 단계; 및

<45> (10) 상기 단계 (9)에서 $DS_l(ND - 1) \leq 1$ 인 것으로 판단되면 SL_l을 ND - 1로 하여 라운드 길이를 감소시키고, 그렇지 않으면 SM이 공집합이 아닌 동안 상기 단계 (8) 내지 (9)를 반복하는 단계

<46> 를 포함하는 것을 그 특징으로 한다.

<47> 또한, 본 발명의 또 다른 특징에 따른 클러스터링된 비디오 서버에서 부하 공유를 위해 데이터를 적응적으로 검색하는 방법은,

<48> (11) 고객이 클러스터 k에 저장된 비디오 스트림을 닫는 경우, $DS_k(j)$와 $BS_k(j)(j = 1, \cdots, ND)$를 다시 계산하는 단계;

<49> (12) 다음 수학식 5를 만족하는지 판단하는 단계;

수학식 5

<50> $$DS_k(SL_k) \leq \frac{fr_{SL_k-1}}{fr_{SL_k}} - \frac{\epsilon}{fr_{SL_k-1}} \quad (SL_k = 2, \cdots, ND)$$

<51> (13) 상기 단계 (12)에서 상기 수학식 5가 만족되는 것으로 판단되면, 다음 수학식 6을 만족하는지 확인하여 라운드 길이를 감소시키는 것이 디스크 대역폭 제약을 위반시킬 것인지를 판단하는 단계; 및

수학식 6

<52> $$DS_k(SL_k - 1) \leq 1$$

<53> (14) 상기 단계 (13)에서 디스크 대역폭 제약을 위반하지 않는 것으로 판단되면, SL_k를 SL_k - 1로 감소시켜 클러스터 k의 라운드 길이를 감소시키는 단계

<54> 를 더 포함하는 것을 그 특징으로 한다.

<55> 이하에서는 첨부된 도면들을 참조하여, 본 발명의 실시예에 대해 상세하게 설명하기로 한다.

<56> 1. 시스템 모델

<57> 본 발명에서 사용되는 비디오 서버는 하나의 디스크 어레이를 클러스터들로 분할하며, 각각의 클러스터는 독립적으로 스트라이핑 그룹을 형성한다. 비디오 서버는 C개의 클러스터들로 분할되며, 각각의 클러스터는 Q개의 동종 디스크로 구성될 수 있다. 클러스터 k의 i번째 디스크를 지칭하기 위해서 D_k^i라고 표시하기로 한다.

<58> 도 1은 클러스터의 수가 2개이고, 각각의 클러스터가 4개의 동종 디스크를 포함하는 비디오 서버의 일예의 구성을 도시한다. 비디오 V_i는 유한 개수의 연속적인 세그먼트들($S_{i,1}$, $S_{i,2}$, …)로 나누어지는데, 도 1에서는, V_1은 클러스터 1에 저장되며, V_2는 클러스터 2에 저장된다.

<59> 고객이 비디오 스트림을 요청하는 경우, 서버는 해당 비디오의 일부분을 저장하기 위하여 일정 양의 버퍼 공간을 할당한다. 또한, 서버는 디스크들로부터 비디오 데이터를 연속적으로 검색하기 위하여 각각의 스트림에 대하여 디스크 대역폭을 할당한다. 디스크 검색 시간의 불가피한 오버헤드를 줄이기 위하여, 디스크 헤드가 디스크의 표면을 따라 앞뒤로 스캔하면서 자신들이 지나치는 블록들을 검색하는 SCAN 스케줄링을 사용한다.

<60> 2. 라운드길이에 따른 리소스 이용도의 변화

<61> 라운드 길이의 선택은 동시 발생하는 사용자의 최대 수를 실질적으로 결정하기 때문에 중요하다. 한 라운드 동안 많은 양의 데이터를 검색하는 것은 디스크 지연(disk latency)의 영향을 감소시키기 때문에, 길이가 긴 라운드는 디스크 이용도의 효율을 향상시킨다. 반면에, 검색된 데이터를 저장하기 위해 많은 버퍼 공간이 필요하기 때문에, 길이가 긴 라운드는 버퍼 요구조건을 증가시킨다.

<62> 본 발명에서 제안하는 적응형 데이터 검색 방법에서는, 각각의 클러스터가 고유한 라운드 길이 및 버퍼 공간을 가진다. 과부하가 걸린 클러스터에 의해 야기되는 디스크 대역폭 상의 스트레스를 극복하기 위해, 본 발명에 따른 방법은 라운드 길이를 증가시킨다. 이것은 버퍼 공간에 대한 요구조건을 증가시키지만, 보다 적게 사용되는 클러스터들로부터 버퍼 공간을 빼어올 수 있다. 그러나 전체 버퍼 공간은 제한된 리소스이며, 더욱이 라운드 길이를 변화시키는 프로세스는 지터를 발생시킬 수도 있다. 이러한 밸런싱 행위(balancing act)를 효과적으로 수행하기 위해, 먼저 리소스 이용도가 라운드 길이에 따라 얼마나 변화하는지를 알 필요가 있다.

<63> 새로운 라운드 동안 검색되는 데이터가 인접하여 저장되지 않을 수도 있기 때문에, 라운드 길이를 변화시키는 것은 추가적인 검색 오버헤드를 발생시킬 수 있다. 이것을 해결하기 위해, 각각의 데이터 세그먼트 $S_{i,m}$을 NS개의 서브-세그먼트 $ss_{i,m}^n$(n = 1, …, NS)로 분할하는데, 여기서 각각의 서브-세그먼트의 크기는 길이 BR의 기본 라운드 동안 검색되는 데이터 크기에 해당한다. NS개의 서브-세그먼트들은 인접하게 저장되어 하나의 세그먼트를 구성하며, 세그먼트들은 도 1에 도시된 바와 같이 라운드-로빈(round-robin) 방식으로 배치된다(여기서 라운드-로빈 방식이란 그룹 내에 있는 모든 요소들을 한 번씩 순서대로 돌아가면서 처리하는 방법을 의미함).

<64> NS의 j번째 약수를 나타내기 위해 dv_j(j = 1, …, ND)라는 표기를 사용할 것인데, 여기서 ND는 NS의 약수의 개수이다. 가능한 라운드 길이들의 집합인 FS는 $fr_j|fr_j = BR * dv_j$이다. FS의 원소들이 오름차순으로 정렬된다고 가정한다. 예를 들어, 만약 NS = 6이면, FS는 $fr_1 = BR$, $fr_2 = 2BR$, $fr_3 = 3BR$, $fr_4 = 6BR$이다. FS에 해당하지 않는 라운드 길이는, 한 번의 읽기를 수행하기 위해 2번의 검색을 요구할 수도 있기 때문에 허용되지 않는다. 예컨대, 4BR을 라운드 길이로 선택했다고 가정하면, 서버는 도 1에서의 배치를 사용하여 $S_{1,3}$을 검색하기 위해 D_1^3을 액세스한다. 만약 NS = 6이고, $ss_{1,3}^1$부터 $ss_{1,3}^6$까지 6개의 서브-세그먼트가 하나의 세그먼트 $S_{1,3}$를 구성한다고 하면, 서버는 한 라운드 동안 4개의 인접한 서브-세그먼트 $ss_{1,3}^1$, $ss_{1,3}^2$, $ss_{1,3}^3$, $ss_{1,3}^4$를 D_1^3로부터 검색할 것이다. 그러나 $ss_{1,3}^5$와 $ss_{1,3}^6$는 D_1^3 상에 저장되어 있고, $ss_{1,4}^1$와 $ss_{1,4}^2$는 D_1^4 상에 저장되어 있기 때문에, 다음 라운드 동안, 서버는 2개의 디스크 D_1^3와 D_1^4에 액세스할 필요가 있을 것이다.

<65> dr_i(bits/sec)의 데이터 재생 속도를 갖는 비디오 V_i에 대한 버퍼와 디스크 대역폭 요구조건이 라운드 길이 fr_j(j = 1, …, ND)에 어떻게 영향을 받는지 알아보자. 인접한 데이터에 대한 한 번의 판독에 대해 T_s의 일정한 검색 오버헤드(검색 시간 + 회전 지연(rotational delay))가 요구되는 전형적인 검색 시간 모델(typical seek

time model)이 사용된다. 비디오 스트림 V_i를 검색하는 것은 T_s의 오버헤드와 $fr_j \times \frac{dr_i}{tr}$의 판독 시간을 발생시키는데, 여기서 tr은 디스크의 데이터 전송 속도이다. 결과적으로, 비디오 스트림 V_i를 서비스하는 것은 $T_s + fr_j \times \frac{dr_i}{tr}$의 서비스 타임을 증가시킨다. 고객 CL_i^m이 비디오 스트림 V_i를 요청한다고 가정하자. 고객들은 C개의 클라이언트 그룹(CG_1, …, CG_C)으로 분할될 수 있는데, 여기서 그룹 CG_k 내의 고객들은 클러스터 k로부터 스트림을 수신한다. 만약 라운드 길이가 fr_j이면, 클러스터 k에 대한 전체 서비스 타임 $ST_k(j)$를 다음 수학식 7과 같이 얻을 수 있다.

수학식 7

$$ST_k(j) = \sum_{CL_i^m \in CG_k} (T_s + fr_j \times \frac{dr_i}{tr})$$

<66>

<67> 설명의 편이를 위해, 디스크 부하들이 동일한 클러스터 내의 디스크들 사이에 균등하게 분배된다고 가정한다. 이러한 부하 밸런싱은 고객들에 대한 인가를 지연시킴으로써 쉽게 달성될 수 있다. 디스크에 대한 대역폭 이용도는 흔히 라운드 길이에 대한 전체 서비스 타임의 비로서 정의된다. 한 클러스터 내에 Q개의 디스크가 존재하므로, 이제 클러스터 k에 대한 디스크 대역폭 이용도 $DS_k(j)$를 앞서 언급한 다음 수학식 1과 같이 결정할 수 있다.

<68> <수학식 1>

$$DS_k(j) = \frac{ST_k(j)}{fr_j \times Q}$$

<69>

<70> B를 전체 버퍼 크기라고 하자. 더블 버퍼링(double buffering)이 SCAN 스케줄링을 위해 사용되므로, 비디오 스트림 V_i를 서비스하는 것은 버퍼 이용도를 $\frac{2 \times fr_j \times dr_i}{B}$ 만큼 증가시킨다. 이제 클러스터 k에 대한 버퍼 이용도 $BS_k(j)$를 앞서 언급한 다음 수학식 2와 같이 얻을 수 있다.

<71> <수학식 2>

$$BS_k(j) = \sum_{CL_i^m \in CG_k} \frac{2 \times fr_j \times dr_i}{B}$$

<72>

<73> 3. 지터가 없는 라운드 조정을 위한 조건

<74> 라운드 길이를 증가시키면 데이터 블록이 정시에 도달할 수 없을 수도 있는데, 이에 의해 일부 비디오 스트림에서는 지터가 발생할 수 있다. 이러한 현상이 발생하는 것은, 너무 적은 데이터 블록들이 판독되어서 새로운 라운드 길이를 지원할 수 없었기 때문이다.

<75> 도 2는 Q = 1, $fr_{(j+1)} = 2 \times fr_j$ 이며, 시간 (m + 2) * fr_j에서 라운드 길이가 fr_j에서 fr_{j+1}로 변화하는 경우의 스트림들의 검색 및 재생을 도시한다. 지터가 없는 서비스를 유지하기 위해, 한 라운드 동안 검색된 데이터는 다음 라운드 동안 재생을 위해 사용될 수 있어야 한다. 예를 들어, 도 2(a)에서, 시간 m * fr_j 및 시간 (m + 1) * fr_j 사이에 소모된 데이터의 모든 아이템은 시간 m * fr_j에서 버퍼 내에서 사용 가능하게 될 필요가 있다. 도 2(a)는 라운드의 길이를 늘이는 것이 현재 진행 중인 스트림들에서 어떻게 지터를 발생시킬 수 있는지를 예시한다. 도면으로부터, 스트림들 S3 및 S4에 의해 요청되는 블록들이 시간 (m + 3) * fr_j에서 사용할 수 없기 때문에 현재 진행 중인 스트림들에 대하여 시간 (m + 3) * fr_j부터 시간 (m + 4) * fr_j까지 데이터가

없는 것을 관찰할 수 있다.

<76> 도 2(b)는 라운드 길이의 확장이 지터를 야기하지 않는 상황을 예시한다. 이것은 시간 (m + 3) * fr_j과 시간 (m + 5) * fr_j 사이에 재생될 필요가 있는 데이터의 모든 아이템이 시간 (m + 3) * fr_j에서 사용 가능하기 때문이다. 이러한 관찰결과로부터, 지터가 없는 동작을 보장하기 위해서는, 하나의 디스크로 구성된 클러스터에서 새로운 라운드 길이 fr_{j+1}에 대한 서비스 시간이 fr_j를 초과해서는 안 된다는 것을 쉽게 알 수 있다. Q개의 디스크가 존재하고, 디스크 부하가 동일한 클러스터 내의 디스크들 사이에 균등하게 분배되기 때문에, fr_j로부터 fr_{j+1}로의 지터가 없는 라운드 확장에 대한 조건을 다음 수학식 8과 같이 얻는다(j = 1, …, ND-1).

수학식 8

<77> $$ST_k(j+1) \leq fr_j \times Q$$

<78> 수학식 1을 사용하면, 상기 부등식을 다음 수학식 9와 같이 다시 적을 수 있다.

수학식 9

<79> $$DS_k(j+1) \leq \frac{fr_j}{fr_{j+1}}$$

<80> 버퍼 공간을 얻기 위해 서버는 라운드 길이를 감소시킬 필요가 있을 수 있다. 버퍼 공간은 디스크 대역폭의 대가로 얻어지기 때문에, 서버는 디스크 대역폭 제약을 체크할 필요가 있다. 즉, 라운드 길이를 fr_{j+1}에서 fr_j로 감소시키는 것은 $DS_k(j) \leq 1$(j = 1, …, ND-1)을 만족시켜야 한다.

<81> 도 3은 Q = 1, $fr_{(j+1)} = 2 \times fr_j$ 이며, 시간 (m + 2) * fr_j에서 라운드 길이가 fr_{j+1}에서 fr_j로 감소하는 경우의 데이터 검색 및 재생을 예시한다. 도 3으로부터, 다음과 같은 점들을 관찰할 수 있다. (1) 감소된 라운드 길이 fr_j에 대한 데이터 검색이 시간 (m + 3) * fr_j에서 시작한다는 점과 (2) 라운드 길이를 fr_{j+1}로부터 fr_j로 감소시키는 것은, 시간 (m + 4) * fr_j과 시간 (m + 5) * fr_j 사이에 재생되는 모든 데이터가 시간 (m + 4) * fr_j에서 사용 가능하기 때문에, 지터를 발생시키지 않는다는 점을 관찰할 수 있다.

<82> 4. 적응형 데이터 검색 스킴

<83> SL_k를, FS의 SL_k번째 원소, fr_{SLk}가 클러스터 k에 대한 라운드 길이로서 선택되는 것을 나타내는 선택 파라미터라고 하자. 예를 들어, FS = BR, 2BR, 3BR, 6BR이고 SL_k = 2인 경우, 2BR이 라운드 길이로서 선택된다. 수학식 1 및 2로부터, 더 높은 값의 fr_j는 $DS_k(j)$를 감소시키나 $BS_k(j)$는 증가시킨다는 것을 쉽게 알 수 있다. 이러한 관찰결과로부터, ADR 스킴은 라운드 길이를 다음과 같은 방식으로 조정한다.

<84> (1) 디스크 대역폭에 대하여 과부하가 걸린 클러스터들의 요구조건에 대처하기 위하여, 상대적으로 높은 값의 SL_k가 이들 클러스터들에 할당되며,

<85> (2) 과부하가 걸린 클러스터들의 라운드 길이를 증가시키기 위한 버퍼 공간을 얻기 위하여, 상대적으로 낮은 값의 SL_k가 가벼운 부하가 걸린 클러스터들에 할당된다.

<86> 여기서 어려운 점은, 지터가 없는 방식으로 라운드 길이를 변화시킬 수 있는 경우를 정확하게 결정하는 것이다. 본 발명에서는 디스크 대역폭 이용도에서의 변화에 의해 라운드 길이의 변화가 야기되는 임계값-기반의 접근법을 사용한다. 실제로, 이것은 고객이 비디오 스트림을 요청하거나 닫는 경우에 라운드 길이가 변화될 수 있다는 것을 의미한다. 각각의 경우에 어떤 일이 발생하는지 설명한다.

<87> 4.1 고객이 비디오 스트림을 요청하는 경우

<88> 서버는 모든 클러스터 k(k = 1, …, C)에 대하여 이용도들 $DS_k(j)$와 $BS_k(j)$의 집합을 유지한다. 고객이 클러스터 p에 저장된 비디오 스트림을 요청하는 경우, 서버는 새로운 고객이 인가될 것이라고 가정하고 j = 1, …, ND

에 대하여 클러스터 p에 대한 이용도들 $DS_p(j)$와 $BS_p(j)$를 다시 계산한다. 그 후, 서버는 필요하다면 라운드 길이를 확장하며, 이어서 새로운 고객에 대하여 충분한 디스크 대역폭 및 버퍼 공간이 존재하는지를 체크한다.

<89> 도 4는 본 발명에 따른 적응형 데이터 검색 방법을 구체적으로 구현한 알고리즘을 나타내는 도면이다. 설명의 편이를 위하여, ND ≥ 2라고 가정한다. 먼저, 서버는 SL_p가 ND보다 작다는 것, 즉 라운드 길이가 확대될 수 있다는 것을 확인하여야 한다. 특정 클러스터에 대한 디스크 대역폭 이용도가 임계값을 초과하는 경우, 서버는 라운드 길이를 확장하여 증가된 부하를 반영한다. 이를 위해, 서버는 $\frac{fr_{SL_p}}{fr_{SL_p+1}} - \frac{\epsilon}{fr_{SL_p}} < DS_p(SL_p+1) \leq \frac{fr_{SL_p}}{fr_{SL_p+1}}$ 인지를 체크한다(여기서, ε는 고객 그룹 CGp에 속하는 고객 CL_i^m에 대한 최대 서비스 시간, 즉 $CL_i^m \in CG_p$ (i.e. $\max_{CL_i^m \in CG_p}(T_s + fr_{SL_p} \times \frac{dr_i}{tr})$을 나타냄)(제5행). 수학식 9로부터, 상기 조건이 지터가 없는 라운드 확장을 보장한다는 것을 알 수 있다. 상기 조건이 만족되는 경우, 서버는 충분한 버퍼 공간이 라운드 확장을 위해 사용 가능한지를 체크한다(제6행). 버퍼 공간이 충분한 경우, 서버는 SL_p의 값을 1만큼 증가시킨다(제7행). 그렇지 않으면, SL_p의 값은 변화하지 않는다.

<90> 다음으로, 서버는 새로운 고객이 인가될 수 있는지 여부를 검사하는데, 이것은 라운드 길이가 확장된 경우에는 인가 기준이 $DS_p(SL_p+1) \leq \frac{fr_{SL_p}}{fr_{SL_p+1}}$ 및 $\sum_{k=1}^{C} BS_k(SL_k) \leq 1$와 같이 이미 만족하기 때문에 불필요하다. 인가에 대해 결정하기 위하여, 서버는 먼저 $DS_p(SL_p) \leq 1$인지 여부를 검사한다(제10행). 제10행의 조건을 위반할 경우, 새로운 고객은 디스크 대역폭의 부족 때문에 거절된다. 아니라면, 서버는 $\sum_{k=1}^{C} BS_k(SL_k) \leq 1$인지 여부를 검사한다(제11행). 이 조건을 만족하면, 새로운 고객이 인가된다. 그렇지 않으면, 서버는 한 클러스터의 라운드 길이를 감소시킴으로써 새로운 고객에 대한 버퍼 공간을 획득할 수 있다. 가장 큰 라운드들(즉, SL_k = ND)이 가장 많은 버퍼 공간을 사용하기 때문에, 본 발명에 따른 방법은 이들을 가지고 클러스터들을 테스트한다. SM을 SL_m이 ND인 디스크 이용도 $DS_m(ND)$(m = 1, …, C)의 집합이라고 하자. SM이 공집합이라면, 새로운 고객은 버퍼 공간의 부족 때문에 거절된다. 아니라면, SM 내에서 가장 가볍게 부하가 걸린 클러스터를 찾기 위하여, 서버는 SM 내에서 $DS_l(ND)$의 가장 작은 값을 찾아서, SM으로부터 제거한다(제15-16행). 이것이 디스크 대역폭 제약을 위반하지 않는다면(제17행), 서버는SL_l의 값을 ND로부터 ND-1로 감소시킨다(제18-19행). 라운드 길이에서의 이러한 감소로 인해 새로운 고객이 수용 가능하게 된다.

<91> 4.2 고객이 비디오 스트림을 닫는 경우

<92> 시기적절한 방식으로 라운드 길이를 감소시킴으로써 서버는 버퍼 공간을 얻을 수 있는데, 이는 이후 과부하가 걸린 클러스터들의 라운드 길이를 확장시키는 데 사용될 수 있다. 이를 위하여, 고객이 클러스터 k에 저장된 비디오 스트림을 닫는 경우, 서버는 $DS_k(j)$와 $BS_k(j)$(j = 1, …, ND)를 다시 계산한 후, $DS_k(SL_k) \leq \frac{fr_{SL_k-1}}{fr_{SL_k}} - \frac{\epsilon}{fr_{SL_k-1}}$ (SL_k = 2, …, ND) 여부를 체크한다. 이 조건이 만족되면, 서버는 라운드 길이를 감소시키는 것이 디스크 대역폭 제약을 위반시킬 것인지(즉, $DS_k(SL_k-1) \leq 1$) 여부를 체크한다. 이 조건이 유지될 수 있다면, 서버는 SL_k의 값을 SL_k-1로 감소시킨다.

<93> 5. 실험 결과

<94> 본 발명의 유효성을 평가하기 위하여, tr과 T_s가 각각 55 MB/s와 11.2 ms인 40개의 IBM Ultrastar36Z15 디스크들을 가진 서버를 시뮬레이션하였다. 서버는 10개의 클러스터로 분할되며, 각각의 클러스터는 4개의 디스크로 구성된다. 고객 요청의 도착은 푸아송 분포(Poisson distribution)를 따르는 것으로 가정한다. 또한, 모든 비디오는 90분 길이이며, MPEG 1 재생에서 전형적인 1.5 Mbps의 동일한 비트-속도를 가진다고 가정한다. NS는 6으로 설정되며, FS는 BR=0.5, 2BR=1, 3BR=1.5, 6BR=3이다. 전체 버퍼 크기 B는 2GB이다. 각각의 영화의 클러스터 위치는 임의로 선택된다. $p_i(t)$를 시간 t에서 비디오 V_i(i = 1, …, NV)의 액세스 확률이고, 여기서 NV는 영화들의 수이며, $\sum_{i=1}^{NV} p_i(t) = 1$라고 하자. 다음과 같은 2가지 서비스 시나리오를 고려한다.

<95> (1) 인기도의 점진적인 변화(gradual change of popularities: GCP)

<96> 액세스 확률은 Zipf 분포를 따른다($p_i(0) = \frac{1}{\sum_{m=1}^{NV} \frac{1}{m^{(1-\theta)}}} \times \frac{1}{i^{1-\theta}}$). Θ = 0.0으로 설정하는데, 이것은 실제 VOD 애플리케이션에 대한 실 측정값에 해당한다. 처음에, $p_1(0) > \cdots > p_{NV}(0)$이며, 시간 τ에서, $p_{(i+1) \bmod NV}(\tau) = p_i(0)$이다. 영화의 인기도는 τ의 주기로 시간에 대해 순환하는데, 예컨대 $p_{(i+1) \bmod NV}((j+1) \times \tau) = p_i(j \times \tau)$이다. 이러한 패턴은 하루의 서로 다른 시간들에서 관객의 유형들을 변화시키는 것에 해당할 수 있다.

<97> (2) 인기도의 급격한 변화(drastic change of popularities: DCP)

<98> 이 경우에, 특정 영화의 인기도가 τ의 모든 주기 동안 α까지 급격하게 증가한다. $p_1(0)$ = α로 설정한다. 그러면, 영화들의 인기도는 τ의 주기로 순환한다. 예를 들어, 시간 τ에서, $p_2(\tau)$는 α까지 증가한다. 이 시나리오는 새로운 영화 또는 드라마의 주기적인 배급에 해당한다.

<99> 우리는 적응형 라운드 길이 조정을 허용하지 않는 CV1, CV2, CV3의 3가지 기존의 방법들과 본 발명에 따른 방법을 비교할 것이다. CV1, CV2, CV3의 라운드 길이는 각각 1, 1.5, 3초로 가정된다. 그 후, 우리는 GCP 및 DCP 시나리오 하에서 20시간 동안 인가된 고객들의 수를 조사했다. τ의 값은 30분으로 가정된다.

<100> 도 5는 GCP(인기도의 점진적인 변화) 시나리오 하에서 고객의 비율, 즉 인가 비(admission ratio)가 고객 요청들의 도착 속도에 따라 어떻게 달라지는지를 도시한다. 이 경우에, 본 발명에 따른 방법(ADR)은, 1%와 28% 사이의 더 많은 고객들을 인가하면서, 모든 작업부하들 하에서 기존의 스킴들보다 더 나은 성능을 보였다.

<101> 도 6 및 도 7은 도착 속도가 20 및 30 요청/분인 경우에 DCP(인기도의 급격한 변화) 시나리오 하에서 인가 비가 α의 값에 따라 어떻게 달라지는지를 도시한다. 도 6 및 도 7에 따르면, 본 발명에 따른 방법(ADR)이 모든 작업부하 하에서 가장 좋은 성능을 나타낸다. 본 발명에 따른 방법은 CV1보다는 21%와 33% 사이의 더 많은 고객들을 인가하였고, CV2보다는 10%와 21% 사이의 더 많은 고객들을 인가하였으며, CV3보다는 1%와 19% 사이의 더 많은 고객들을 인가하였다. 도면들로부터, 본 발명에 따른 방법과 CV2 사이의 성능 차이가 α의 값이 증가함에 따라 증가하는 것을 관찰할 수 있는데, 이것은 다음과 같이 설명될 수 있다. 즉, α의 값이 증가함에 따라, 요청들은 인기 클러스터들로 집중되어 디스크 대역폭이 더 짧은 시간 내에 포화되게 된다. 또한, CV3은 도착 속도가 20 요청/분인 경우에는 본 발명에 따른 방법보다 3%까지 더 나쁜 성능을 보이나, 도착 속도가 30 요청/분인 경우에는 21%까지 더 나쁜 성능을 보이는 것으로 관찰된다. 이것은 다음과 같이 설명될 수 있다. 즉, 도착 속도가 20 요청/분인 경우에, 버퍼 리소스는 대부분의 경우에 문제가 되지 않으며, 따라서 CV3은 본 발명에 따른 방법보다 단지 조금만 나쁜 성능을 보인다. 그러나 도착 속도가 증가함에 따라, 서버는 보다 많은 고객들을 수용하기 위해 보다 많은 버퍼 공간을 필요로 하나, 이 효과는 CV3 스킴에 대해 보다 뚜렷하며, 그 결과 증가된 성능 차이를 야기한다.

발명의 효과

<102> 본 발명에 따르면, 라운드 길이를 동적으로 변화시키는 적응형 데이터 검색 방법을 제안함으로써, 데이터의 이동 또는 복사 없이도 클러스터들 간에 부하 공유를 달성할 수 있다.

도면의 간단한 설명

<1> 도 1은 클러스터의 수가 2개이고, 각각의 클러스터가 4개의 동종 디스크를 포함하는 비디오 서버의 일예의 구성을 도시하는 도면.

<2> 도 2는 Q = 1, $fr_{(j+1)} = 2 \times fr_j$ 이며, 시간 (m + 2) * fr_j에서 라운드 길이가 fr_j에서 fr_{j+1}로 변화하는 경우의 스트림들의 검색 및 재생을 도시하는 도면.

<3> 도 3은 Q = 1, $fr_{(j+1)} = 2 \times fr_j$ 이며, 시간 (m + 2) * fr_j에서 라운드 길이가 fr_{j+1}에서 fr_j로 감소하는 경우의 데이터 검색 및 재생을 예시하는 도면.

<4> 도 4는 본 발명에 따른 적응형 데이터 검색 방법을 구체적으로 구현한 알고리즘을 나타내는 도면.

<5> 도 5는 GCP(인기도의 점진적인 변화) 시나리오 하에서 고객의 비율, 즉 인가 비(admission ratio)가 고객 요청들의 도착 속도에 따라 어떻게 달라지는지를 도시하는 도면.

<6> 도 6 및 도 7은 도착 속도가 20 및 30 요청/분인 경우에 DCP(인기도의 급격한 변화) 시나리오 하에서 인가 비가 α의 값에 따라 어떻게 달라지는지를 도시하는 도면.

<7> <도면의 주요 부분에 대한 부호의 설명>

<8> D_k^i : 클러스터 k의 i번째 디스크

<9> $S_{i,m}$: 비디오 V_i의 m번째 세그먼트

<10> $ss_{i,m}^n$: 서브-세그먼트

<11> Data read : 데이터 판독

<12> S_m : 세그먼트 m

<13> Playback of S_m : 세그먼트 S_m에 대한 재생

<14> CV : 적응형 라운드 길이 조정을 허용하지 않는 기존의 데이터 검색 방법

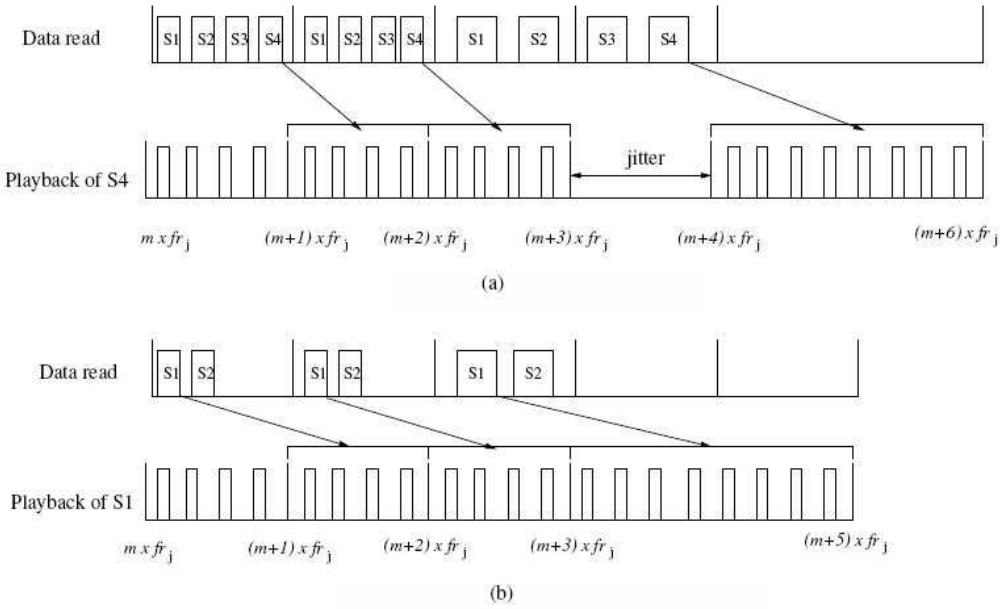
<15> ADR : 본 발명에 따른 적응형 데이터 검색 방법

도면

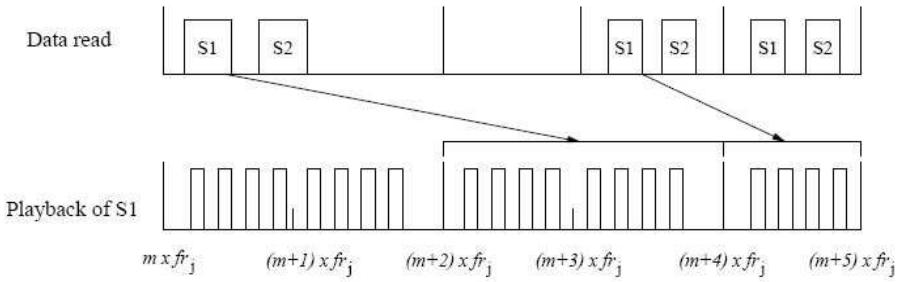
도면1

cluster 1				cluster 2			
D_1^1	D_1^2	D_1^3	D_1^4	D_2^1	D_2^2	D_2^3	D_2^4
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$S_{2,4}$
$S_{1,5}$	$S_{1,6}$	$S_{1,7}$	$S_{1,8}$	$S_{2,5}$	$S_{2,6}$	$S_{2,7}$	$S_{2,8}$
...	...	...	...	...	...	...	...

도면2



도면3

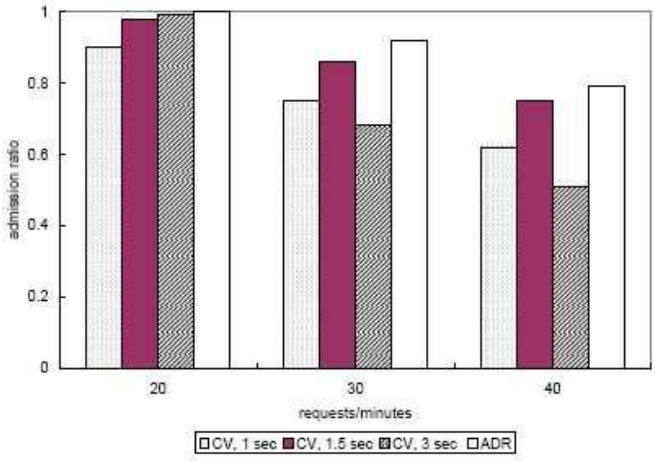


도면4

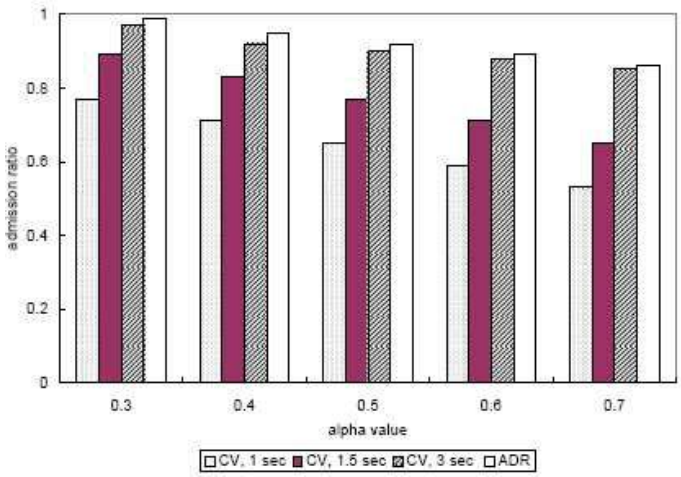
Algorithm 1 Admission Decision Procedure

1: Set of utilizations of $DS_k(j)$ and $BS_k(j)$, $(j = 1, ..., ND)$
2: A client requests a stream from cluster p, and the server recalculates $DS_p(j)$ and $BS_p(j)$;
3: A set of disk utilizations: $SM = \{DS_m(ND) | m = 1, ..., C \text{ and } SL_m = ND\}$;
4: BOOLEAN: FLAG ← FALSE;
5: **if** $SL_p \leq ND - 1$ and $\frac{fr_{SL_p}}{fr_{SL_p+1}} - \frac{\epsilon}{fr_{SL_p}} < DS_p(SL_p + 1) \leq \frac{fr_{SL_p}}{fr_{SL_p+1}}$ **then**
6: **if** $\sum_{k=1}^{C} BS_k(SL_k) + BS_p(SL_p + 1) - BS_p(SL_p) \leq 1$ **then**
7: $SL_p \leftarrow SL_p + 1$; { Round length increases }
8: **end if**
9: **end if**
10: **if** $DS_p(SL_p) \leq 1$ **then**
11: **if** $\sum_{k=1}^{C} BS_k(SL_k) \leq 1$ **then**
12: FLAG ← TRUE;
13: **else**
14: **while** FLAG = FALSE and $SM \neq \phi$ **do**
15: Find the lowest value, $DS_l(ND) \in SM$;
16: $SM \leftarrow SM - \{DS_l(ND)\}$;
17: **if** $DS_l(ND - 1) \leq 1$ **then**
18: $SL_l \leftarrow ND - 1$;
19: FLAG ← TRUE;
20: **end if**
21: **end while**
22: **end if**
23: **end if**
24: **if** FLAG = TRUE **then**
25: The new client passes the admission test;
26: **else**
27: The new client is rejected;
28: **end if**

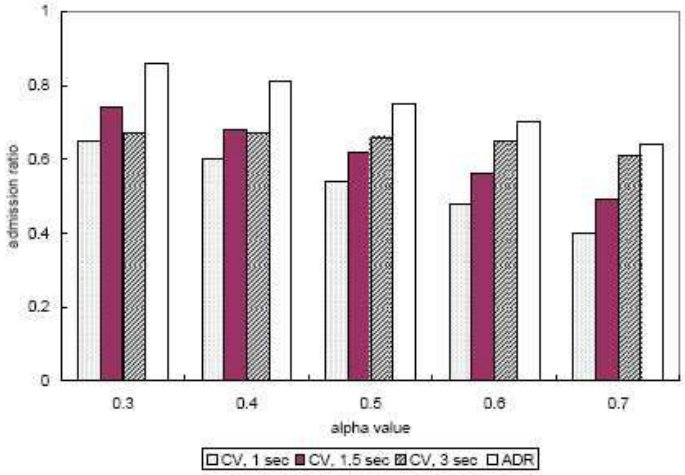
도면5


admission ratio
1
0.8
0.6
0.4
0.2
0
20
30
40
requests/minutes
CV, 1 sec
CV, 1.5 sec
CV, 3 sec
ADR

도면6


admission ratio
1
0.8
0.6
0.4
0.2
0
0.3
0.4
0.5
0.6
0.7
alpha value
CV, 1 sec
CV, 1.5 sec
CV, 3 sec
ADR

도면7


admission ratio
1
0.8
0.6
0.4
0.2
0
0.3
0.4
0.5
0.6
0.7
alpha value
CV, 1 sec
CV, 1.5 sec
CV, 3 sec
ADR