2000년 한국정보처리학회 추계 학술논문발표집 제7권 제2호

통합 서비스 제공을 위한 이중 큐 디스크 스케줄링

현은실, 이윤정, 김태윤
고려대학교 컴퓨터학과
e-mail:eunsil@netlab.korea.ac.kr

Two queue disk scheduling for Integrated service

Eun-sil Hyun, Yoon-Jung Rhee, Tai-Yun Kim
Dept. of Computer Science and Engineering, Korea University

요 약

응용프로그램을 실행시키는 것은 많은 디스크 대역폭을 필요로 하고 있다. 데이터를 조회하는데 걸리는 시간을 최소로 하는 것이 성능을 향상시키는 가장 큰 문제이다. 디스크 기술은 속도와 용량면에서 많은 발전이 있었음에도 불구하고, 프로그램에서 요구하는 요구사항이 증가함하고 있다. 멀티미디어 서버는 여러 종류의 텍스트나 이미지, 비디오 등의 다양한 데이터 타입을 지원한다. 따라서 다른 데이터 타입들은 다른 마감시간의 요구와 우선 순위를 가지고 있다. 연속적인 오디오나 비디오 같은 것들은 마감시간 제약을 가지고 있다.

이 논문에서 제한한 알고리즘은 모든 요청들을 두 개의 큐로 유지하고 탐색 시간을 최소화 함으로써 디스크 대역폭의 처리율을 극대화시킨다. 즉 다양한 우선 순위의 요청들을 해당 클래스로 나누고 주기적인 요청을 위한 큐와 비 주기적인 요청이나 대화형 요청들을 위한 또 하나의 큐로 나누고 요청의 마감시간과 디스크 헤드의 움직임을 고려하여 스케줄링한다.

1. 서론

멀티미디어를 지원하는 서비스는 컴퓨터의 하드웨어와 소프트웨어의 기술 발전으로 인해 빨리 성장하고 있다. 특히 대용량의 저장공간이나 압축기술, 높은 속도의 네트워크로 인해 브로드 캐스팅, 디지털 라이브러리, VOD 등의 멀티미디어 서비스가 가능하다. 이러한 응용프로그램을 실행시키는 것은 많은 디스크 대역폭을 필요로 한다. 서버의 디스크로부터 해당 데이터 블록을 불러올 때, 데이터를 전달하고(서버에서 클라이언트에게) 데이터를 조회 하는데 시간(탐색시간과 헤드의 회전시간)이 소요된다. 데이터 조회 시간을 최소로 하는 것이 성능을 향상시키는 가장 큰 문제이다. 디스크 기술은 속도와 용량면에서 많은 발전이 있었음에도 불구하고, 프로그램에서 요구하는 요구사항이 증가함하고 있고 멀티미디어 서버는 여러 종류의 텍스트나 이미지, 비디오 등의 다양한 데이터 타입을 지원한다. 따라서 다른 데이터 타입들은 다른 마감시간의 요구와 우선 순위를 가지고 있다. 연속적인 오디오나 비디오 같은 것들은 마감시간 제약을 가지고 있다.

이 논문에서 제한한 알고리즘은 모든 서비스 요청들을 해당 클래스로 분류하고 주기적인 요청을 위한 큐와 비 주기적인 요청이나 대화형 요청들을 위한 또 하나의 큐로 나누고 요청의 마감시간과 우선순위을 고려하여 스케줄링 한다.

2장에서는 기존에 연구되어 왔던 실시간 디스크 스케줄링에 관한 관련 연구를 제시하고, 3장에서는 디스크 서비스를 위한 서비스 요청들을 3개의 클래스로 나누어 두 개의 큐로 관리하고 마감시간을 기반으로 서비스를 제공하는 새로운 알고리즘과 동작과정을 제시한다. 마지막으로 4장에서는 결론과 향후 연구 방향을 제시한다.

2. 디스크 스케줄링에 관한 관련연구

디스크 스케줄링의 대표적인 두 가지 연구 방향은 싸이클 기반과 마감시간 기반의 알고리즘이다.

싸이클 기반의 알고리즘은 서비스 요청을 그룹으로 나누어 디스크 상의 데이터의 물리적인 위치에 따라 각각의 그룹 안에서 요청을 정렬한다. 이 알고리즘은 각 요청의 마감시간을 고려하지 않는다.

두 번째는 마감시간을 고려한 알고리즘이다. 이것은 단지 소프트 리얼타임에 대한 프로그램에 적용된다. 이 알로리즘은 요청의 마감시간을 중요시하지만 모든 작업은 마감

시간내에 도착하도록 하지는 않는다. 일반적으로 마감시간 기반의 알고리즘은 높은 디스크 처리율을 보장한다. 이 계열의 예는 EDF(Earliest DeadLine First), SCAN-EDF 그리고 SCAN-RT 가 있다.

대화형 작업을 위한 요청은 고유한 스케줄링이 존재한다. 그 예는 Bubble up으로 대화형 요청의 응답시간을 최소로 하는 데 목적을 두고 있다. 이 기술을 바탕으로 모든 주기적인 요청은 SCAN의 순서로 서비스를 처리한다. 그러나 새로운 스트림에 대한 서비스가 처음으로 요청될 경우 그것은 즉각적인 서비스를 위해 큐의 헤드에 삽입된다. Wijayaratne 과 Reddy이 알고리즘은 오디오나 비디오 프로그램 같은 주기적인 요청과 대화형 요청 사이에 종류의 서비스를 제공한다. 오디오나 비디오는 디스크에서 일정한 마감시간을 가지고 주기적으로 도착하는 주기적인 요청이다. 대화형 요청은 예상하기 어려운 방법으로 요청이 도착한다. 이 알고리즘은 큐에 SCAN 순서대로 유지하고 있다가 주기적인 요청을 그룹 안에 위치시키고 대화형 요청이 들어오면 큐의 맨 처음으로 위치시키는 것이다.

3. 이중 큐 기반의 통합 서비스 제공을 위한 디스크 스케줄링

본 장에서는 본 논문에서 먼저 해당 서비스를 다른 클래스로 구분하고, 이중 큐 기반의 통합 서비스 제공을 위한 디스크 스케줄링을 제시하여 이를 기반으로 동작 모델을 제안하고자 한다.

3.1 클래스 구분

디스크에서 요청 서비스를 분류하는 있어서 작업을 특성에 따라 속성을 구분하면 3가지의 클래스로 나눈다.

①주기적 요청

주기적 요청은 주기적인 시간 간격으로 서비스를 요구한다. 비디오 플레이어, 디지털 방송, VOD 등과 같이 MPEG2 수준의 데이터를 실시간으로 전송하는 경우에 발생되는 데이터 유형이다. 이러한 클래스의 데이터는 데이터 손실 및 엄격하지 않은 에러 제어가 허용 될 수 있으며, 실 시간성 및 연속성을 요구하므로 빠른 전송을 필요로 할 뿐 아니라 각 데이터 간의 동기화에도 신경을 써야 한다.

②대화형 요청

대화형 요청은 I/O 시스템으로부터 빠른 응답시간을 요구한다. 이 요청은 일정하지 않은 비 동기적 시간 간격을 가지고 도착한다. 인터넷 전화, 웹 비디오 전화, 주문형 뉴스, 인터넷 비디오 광고 등이 해당되며, 실시간 처리 및 스트림 동기화를 필요로 한다. 품질에 크게 지장을 주지 않는 범위의 지연이나, 데이터의 손실은 허용 될 수도 있다.

③비 주기적 요청

일반적인 비동기적 요청 이며 일반적인 작업은 ftp, 전자우편(email), 그래픽 이미지 전송 등의 데이터들로서 동기화나 지연 시간 제어와 같은 복잡한 메커니즘을 필요로 하지 않는다.

3.2 디스크 대역폭 할당

대역폭은 앞에서 정의 한 클래스에 따라 각각의 다른 비율로 할당한다. 각 클래스에 따라 한정적으로 대역폭을 할당한다. 이 단계를 거친 후 할당된 대역폭에 한하여 큐로 관리하고 스케줄링한다. 각 해당 클래스를 위한 admission controller는 요청 풀에 얼마만큼의 요청을 포함할 것인지 결정하고 또한 풀에 포함되는 순서를 결정한다.

admission controller는 각각의 클래스에 합당한 대역폭을 할당한다. 만약 일정한 비율의 대역폭이 할당되지 않는다면 요청들이 비 동기적으로 많이 도착하면, 주기적 요청들에게 제공되는 서비스를 방해하는 요인이 된다. 이 것을 피하기 위해 대화형 요청이나 비 주기적 요청들은 그에 해당하는 admission controller에 의해 주어진 시간 안에 최대한 허용되는 요청들의 수와 최소한 허용되는 요청들을 제한 한다. 비 주기적인 요청과 대화형 서비스의 요청에 기아현상이 일어나는 것을 방지하기 위하여 최소값을 설정함으로 서비스를 보장한다. 또한 비 주기적 요청은 주기적인 요청이나 대화용 요청을 처리한 후의 나머지의 대역폭을 할당받을 수 있다. 만약 주기적 요청이나 대화용 요청이 더 이상 디스크의 대역폭을 요구하지 않는다면, 비 주기적인 요청은 비 주기적인 요청의 처리율을 향상시킬 수 있도록 가용한 대역폭을 사용할 수 있다.

3.3 이중 큐 기반의 통합 서비스 제공을 위한 디스크 스케줄링

디스크 스케줄러는 요청을 개개의 요청 시간이 만족되도록 서비스를 처리해야 한다. 그러나 비 주기적 요청이나 대화용 요청은 언제 서비스를 요청할 지를 예상하기 어렵다. 이 요청들을 처리하기 위해서 스케줄러는 두 개의 큐를 사용한다. 디스크 스케줄러는 주기적 요청에 대한 하나의 큐을 사용하고 비 주기적으로 도착하거나 대화형 요청이 도착했을 경우를 위해 또 하나의 큐를 만든다. 각각의 큐는 기본이 EDF스케줄링을 기본으로 하고 비 주기적것은 마감시간을 고려하여 마감시간이 극박한 경우 처리하고 만약 시간의 여유가 있다면 두 번째 큐에게 제어권을 넘긴다. 대화형 요청이나 비 주기적 작업은 가장 큰 문제가 언제 요청이 일어날 지 모른다는 사실이다. 두 번째 큐에서는 대화형 요청에 우선 순위를 두고 마감시간이 극박한 경우 비 주기적 작업보다 먼저 처리한다.

만약 주기적 요청이 높은 우선 순위를 가지고 우선적으로 처리된다면 비 주기적이거나 대화형 요청은 주기적인 요청이 처리되기를 기다리게 되어 긴 응답시간을 가지게 된다. 디스크 스케줄링은 주기적인 요청의 가용한 여유시간을 고려하여 큐에 EDF 스케줄링에 의하여 스케줄하고 비 동기적으로 도착하는 비 주기적 요청이나 대화형 요청들은 또 하나의 큐에 도착하면 두 개의 큐를 머지 하여 스케줄링을 실시한다.

3.3.1 이중 큐 기반의 통합 서비스 제공을 위한 디스크 스케줄링 알고리즘 설계

다음과 같은 원칙을 기반으로 한다.

1. 주기적 요청들은 하나의 큐에 우선순위, 마감시간을 고려하여 정렬되어 서비스를 할당한다.

2. 비 동기적 요청이나 대화형 작업을 위한 요청이 도착한다면, 다음과 같은 알고리즘을 적용시킨다.

1) 비 주기적 대화형 작업을 위한 큐에 마감시간이 극박한 작업이 있을 경우 하나의 큐에 주기적인 요청들과 다시 우선 순위, 마감시간을 고려하여 머지 한 후 정렬한다.

2) 두 개의 큐를 머지 한 경우 큐에 모든 요청이 마감시간을 어기지 않고 모두 처리 가능한 상태라면 유효한 상태라 한다.

3) 큐가 유효한 상태라면 감지되면 알고리즘이 끝낸다.

4) 만약 요청 Rj(큐에 이미 있던 요청이거나 새로운 요청일 경우 모두 다)가 마감 시간을 넘길 경우,

· Rj 보다 선행하여 실행되는 요청들 중에서 가장 낮은 우선 순위를 가진 요청을 찾아서 큐의 마지막으로 이동시킨다.

· 다수의 후보 요청들이 있다면, 그 중에서 가장 여유가 있는 마감시간을 가진 요청을 선택한다.

· 영향을 받은 요청의 서비스 시간을 계산한다.

· 만약 마감 시간안에 서비스를 받지 못할 경우, 마감시간을 지키지 못했음을 선언하고 큐에서 제거한다.

3.3.2 동작과정

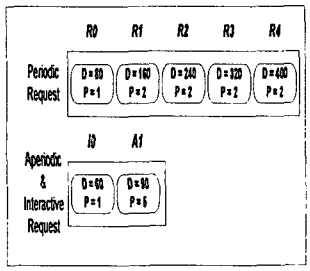
각 클래스는 자신의 admission controller가 있고 요청 큐에 삽입될 수 있는 수와 순서를 결정하고 요청 큐에 삽입시킨다. 요청 큐에서는 디스크 처리율을 극대화시키기 위해서 큐는 EDF 알고리즘을 통하여 처리한다. 만약 비 동기적 작업을 위한 요청이나 대화형 작업의 새로운 작업이 삽입된다면 또 하나의 큐에 저장을 하고 여기에서 비 동기적 작업을 위한 요청에 비해 대화형 작업을 위한 요청에 더 큰 우선 순위를 부여하여 큐의 상단에 위치시킨다. 비 주기적 대화형 작업을 위한 큐에 위치한 작업들의 마감시간이 다가오면 주기적인 작업을 위한 큐에 새로운 알고리즘은 ⓐ 우선 순위 ⓑ 마감시간을 고려하여 서비스의 순서를 재 정렬한다. 주기적인 작업을 위한 큐 안에서 순서를 정할 때는 EDF순서를 기본으로 하고 우선 순위를 고려해 대화형 작업을 위한 요청을 큐에 앞부분에 위치시킨다. 하나의 EDF 싸이클에 속하는 요청은 디스크 헤드의 위치에 관계하여 오름차순이나 내림차순으로 정렬한다. 스케줄링 큐가 만약 모든 요청을 마감시간을 어기지 않고 모두 처리한 상태를 유효하다고 정의하면, 큐는 초기에 유효한 상태이다. 만약 비 동기적 삽입(비 주기적인 작업을 위한 요청이나 대화형 작업을 위한 요청)인 Rnew가 삽입해 유효하지 않은 상태가 된다면, (적어도 하나이상이 마감시간을 지키지 못한다면) 알고리즘은 유효한 상태를 회복시키기 위해서 일련의 단계를 가지고 있어야 하며 다음과 같은 원칙에 의해 정렬이 되어야 한다.

① Rnew의 삽입은 높은 우선 순위를 가진 다른 요청들의 마감시간을 놓치지 말아야 한다.

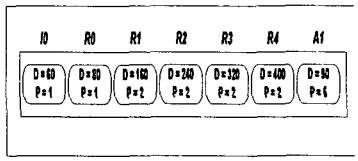
② 마감시간을 지키지 못한 요청의 수를 최소화 시켜야 한다.

③ 유효한 상태를 복구하는데 있어서, EDF 순서를 혼합하든 새로운 EDF순서를 만들간에 대역폭의 효율성을 저하해서는 안된다.

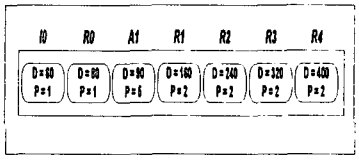
새로운 요청이 도착하여, 만약 Rj (Rj 가 이전에 있던 요청이거나 새로운 요청일 경우)의 마감시간을 놓쳤을 때, 먼저의 위치에서 큐의 마지막으로 이동시킨다. 그러나 큐의 마지막으로 이동시킨 요청은 이동함으로써 마감시간을 지키지 못할 수 있다. 따라서 이동할 요청을 선택할 때에는 낮은 우선 순위를 가진 요청을 선택해야 한다. 그리고 새로운 요청이 낮은 우선 순위를 가지고 있다면 그것을 큐의 마지막으로 이동시킴으로 문제를 해결한다. 이동하는 작업을 선택하는데 있어서는 낮은 우선 순위의 순서대로, 같은 우선 순위를 가진 요청들은 마감시간이 여유가 있는 것을 선택한다. 선택된 요청은 스케줄이 유효한 상태로 될 때까지 큐의 마지막으로 옮겨진다. 결국 큐의 마지막으로 이동하는 것은 마감시간의 여유가 있으며 낮은 우선 순위를 가진 요청들이다. 이동한 요청은 마감시간을 놓치기 쉽다.



(a) before the merge



(b) after the merge : Periodic request violates the deadline of A1



(c) after reorganization

그림 1에서 알 수 있듯이 큐에는 우선 순위가 다른 몇 개의 요청을 있다. 높은 값은 높은 우선 순위를 의미하고, D(Deadline)는 각 요청들의 마감시간을 의미한다. 각 요청

은 20msec 의 정해진 서비스 타임을 사용한다. A) 상태는 비 주기적인 요청이나 대화형 작업이 요청되어 처리 되기 전이며 이 때에는 두 개의 큐 - 주기적 작업 요청을 위한 큐와 비 주기적 작업과 대화형 작업 요청을 위한 큐가 두 개로 나누어 배치된 상태이다. B) 상태는 두 개의 큐가 하나의 큐로 머지가 이루어지고 클래스 구분에 의하여 정렬된 상태이다. 이 경우는 비 주기적인 작업 *A1*이 마감시간을 놓치게 된다. 그러므로 큐의 순서는 재배열해야 한다. A1앞의 다수의 요청들 사이에서 우선 순위가 가장 낮고 마감시간에 여유가 있는 요청을 찾아 선택한다. C) EDF 알고리즘에 의하여 재 정렬된 상태이다.

4. 결론 및 향후 연구 방향

본 논문에서는 다수의 클래스를 지원하기 위한 두 개의 큐 기반 디스크 스케줄링을 제안하였다.

모든 서비스를 주기적인 요청, 비 주기적인 요청, 대화형 요청의 3개의 클래스로 구분하고 해당 요청들을 두 개의 큐로 유지하고 디스크 대역폭의 처리율을 극대화시키기 위해 EDF를 사용하였다.

이 방법은 다른 형태의 서비스를 위한 통합서비스를 제공한다.

향후 연구 과제는 이것을 다른 스케줄링과 비교하기 위해 시뮬에이션 하여 성능을 비교 하는 것이다.

참고 문헌

[1] Ibrahim Kamel and T. Niranjan, A Novel Deadline Driven Disk Scheduling Algorithm for Multi-Priority Multimedia Objects,

[2] R.K Abbot and H.Gracia-Molina. Scheduling I/O Request with Deadlines: A Performance Evalution. In IEEE Real Time Systems Symposium, 1990

[3] R. Wijayaratne and N. Reddy. Intergrated QOS Management for Management for Disk I/O. Proc. of IEEE Multimedia System, pages 487-492, June 1999.

[4] D. Kenchammana and J. Srivastava. I/O Scheduling for Digital Continuous Media. ACM Multimedia Systems, 5: 213-237, 1997.

[5] T. Niranjan, T. Chiueh and G. A Schloss. Implementation and evaluation of a multimedia file systems. Proc of ACM SIGMETRICS, June 1998.

[6] M. M. Budhikot, X. J. Chen, D. Wu, and G. M. Parulkar. Enhancements to 4.4 BSD UNIX for efficient networked multimedia in project MARS. Proc. of IEEE Multimedia Computing and Systems Conf., page 326-337, June 1998.