RAID 5의 성능향상을 위한 쓰기 캐쉬의 동적 적응 반출기법

이 혁(李 赫), 최 상 방(崔 相 昉)
인하대학교 전자공학과
전화 : (032) 860-7417 / 팩스 : (032) 868-3654

New Dynamic Adaptive Threshold Destage Algorithms for Cached RAID 5

Hyeok Yie, Sang-Bang Choi
School of Electronic Engineering Inha University
E-mail : cosy75@lycos.co.kr, sangbang@dragon.inha.ac.kr

Abstract

In this paper, we propose a new destage algorithms, called the Dynamic Adaptive Threshold which determines turn-on and turn-off thresholds dynamically depending on the current write cache occupancy level and the differential rate of the host write requests. For performance evaluation, the proposed algorithm is compared with the wellknown High-Low Water Mark (HLWM) algorithm.

Performance tests are fulfilled with our cached RAID 5 simulator. The simulation results show that the proposed algorithm outperforms the HLWM algorithm in terms of response time of host reads and write cache hit ratio under various workloads.

I. 서 론

디스크 시스템은 운영 체제와 같은 컴퓨터 시스템의 기본적인 동작으로부터 사용자 데이터 관리에까지 컴퓨터 운영에 전반적으로 깊이 관련되어 있으므로 기계 요소 중에서는 컴퓨터 시스템의 성능에 가장 큰 영향을 끼친다. 디스크 시스템의 성능을 향상시키기 위하여 개개의 디스크 성능개선에 관한 연구나 RAID와 같이 다수의 디스크를 사용하는 디스크 배열에 관한 연구가 있었다.[2, 3] 최근의 대형 컴퓨터 시스템에서는 단일의 대용량 디스크를 사용하기보다는 여러 개의 디스크들을 연동 하는 디스크 배열 (disk array)을 이용하는 추세에 있다.

RAID-5에서는 한번의 쓰기를 위해 부가적으로 패리티 정보를 업데이트 하므로 물리적으로 4번의 디스크 접근을 수행하는 작은 쓰기 문제를 지니고 있어 이로 인해 사용자 응답 시간은 크게 늘어난다. 이를 해결하기 위한 다양한 접근 방법들 중, 대부분의 대용량 디스크 시스템들은 입출력 수행시간을 줄이기 위해 디스크 캐쉬를 사용한다. 이때 쓰기 캐쉬를 읽기 캐쉬와 별도로 비휘발성 메모리로 구성하면 쓰기 요구에 대해서도 시간적으로 상당한 이득과 함께 높은 신뢰성을 얻을 수 있다.[4, 5]

본 논문에서 고려한 시스템은 비휘발성 메모리를 디스크 시스템의 쓰기 캐쉬로 사용한 RAID-5이다. 호스트로부터의 모든 쓰기 요구는 우선 쓰기 캐쉬에 저장한다. 디스크 캐쉬에 저장된 블록들은 적절한 반출 알고리즘에 따라, 쓸모 없다고 판단되는 블록들을 디스크로 옮긴다. 이런 반출 알고리즘은 쓰기 캐쉬내의 블록들을 디스크로 스케줄하는 순서와 방법을 결정한다.[1, 2]

시스템이 유휴 상태에 이르게 되면 Dirty 블록들을 정해진 알고리즘에 따라 디스크로 반출하게 된다. 일반적으로 널리 사용하고 있는 알고리즘으로 HLWM [1, 2] 방법이 흔히 사용되고 있다. 이 방법은 캐쉬 내에 사용되고 있는 블록의 개수가 상위 제한 개수(High Water Mark)에 이르면 디스크로 반출을 시작하고 줄어들고 있는 블록의 개수가 하위 제한 개수(Low Water Mark)에 이르면 디스크로의 반출을 멈추게 하는 알고리즘이다. 알고리즘이 매우 간단하기 때문에 널리 사용되고 있으나 쓰기 요구가 급격히 변화하는 상태에서는 알고리즘이 유연하게 대응하지 못하여 쓰기 캐쉬가 쉽게 Overflow하기도 한다.

본 논문에서는 정적인 Threshold를 갖고 동작하는 HLWM 알고리즘과 달리 쓰기 캐쉬로 유입되는 쓰기 요구의 변화율에 따른 offset을 계산하고 그에 부가하여 쓰기 캐쉬에 유지되는 블록의 현재 사용개수에 따라 계산된 offset에 가중치를 적용하는 동적 적응 반출 알고리즘을 제안하였다. 이 Offset값을 기본 Threshold 값에 가감하여 이를 통해 새롭게 적용된 Threshold값이 동적으로 변화하도록 하였다. 비록 유입되는 쓰기 요구의 변화율이 같더라도 offset은 현재 쓰기 캐쉬에 유지되고 있는 블록의 개수에 따른 가중치로 다르게 적용된다. 이는 쓰기 캐쉬를 통해 보다 높은 적중률을 얻을 수 있도록 하며 반대로 쓰기 캐쉬의 오버플로우를 최소화할 수 있도록 한다.

II. RAID-5의 작은 쓰기 문제

OLTP에 적당한 Level 5는 데이터의 결함을 제어하기 위한 데이터 기록 방법으로 패리티 정보를 모든 드라이브에 나누어 기록하고, 나머지 드라이브들 사이에 데이터를 블록 단위로 분산한다. 이를 통해 디스크의 병목현상을 크게 줄었으며 멀티프로세스 시스템에서와 같이 작고 잦은 데이터 기록이 있을 경우 더 빠르다.

RAID Level 5는 트랜잭션 형태의 입출력을 발생시키는 데이터 베이스와 같은 응용에서 널리 사용되고 있다. 구조적으로 새로운 데이터를 한 번 기록 위해 디스크에 접근해야 하는 횟수가 상당히 많다. 새로운 데이터를 한 개의 디스크에 한 번 기록하기 위해서 기록하려고 하는 Disk와 해당 패리티 Disk에서 원래의 데이터를 각각 읽고 새로이 쓸 데이터와 위의 데이터들을 XORing 하여 새로운 패리티를 계산하고 새로이 쓸 데이터를 해당 Disk에 쓰고 새로운 패리티를 Parity Disk에 기록하며 이렇게 총 4번의 디스크 접근을 필요로 한다. 이런 이유로 RAID-5는 작은 쓰기(Small write)에 대해서 성능이 떨어진다는 단점을 갖고 있다. 이를 작은 쓰기 문제라고 일컫고, 문제를 극복하기 위한 방법으로 플로팅(Floating) 패리티, 패리티 로깅, 로그 구조를 갖는 파일 시스템 VSP(Variable Scope Parity Protection) 방법, 그리고 비휘발성 캐쉬를 이용한 빠른 쓰기 방법들이 제시되어왔다.

디스크의 쓰기 캐쉬로 비휘발성 메모리를 사용하는 경우에 사용자의 쓰기 캐쉬에 대한 요구는 단지 비휘발성 쓰기 캐쉬에 쓰는 것으로 처리가 완료됨에 따라서 사용자에게는 매우 빠른 응답을 보장해 주게된다.

Cache를 통해 같은 Block을 연속적으로 Update하는 과정을 통해 시간적인 Locality를 얻을 수 있으며, 공간적으로 근접해 있는 작고 연속적인 Block들을 하나의 커다란 단일 쓰기 Block으로 처리할 수 있다. 이와 같은 이유로 Write Cache의 사용이 Write Request의 오버헤드를 줄임으로서 전체적인 Total Response Time을 줄여주는 효과를 얻을 수 있다. 물론 이런 이점들을 가지고 있지만 위와 같은 뛰어난 효과를 얻기 위해서는 먼저 다음과 같은 여러 가지 문제점들을 해결하여야 한다. 첫째로, Nonvolatile Cache를 Disk Array에 사용함에 있어서 Disk Array가 가지고 있는 안정적인 수행을 방해하지 않아야 하기 때문에 시스템에 부합하도록 새롭게 설계되어야 한다는 점이다. 둘째로, Update를 하는 중에 System Failure가 발생할 경우에도 혹은 System으로의 Datapath에 문제가 발생할 경우에도 Write Cache의 Data 보존 상태가 유지되어 System Failure가 복구된 후에 재생성이 가능할 수 있는 기능이 요구되어진다. 마지막으로 논문에서 다루고자 할 내용으로 Write Cache에서 Disk Array로 Data를 효과적으로 Destage를 할 수 있는 스케쥴링 알고리즘을 결정하는 것이다. 즉 Write Cache에 있는 Data Block들을 Disk Array로 언제 Destage 할 것인가 그리고 Dirty Block 중 어느 것을 다음에 Destage 할 것인가를 결정하는 구체적인 방법을 말한다.

III. 동적 적응 반출 알고리즘

비휘발성 캐쉬에 저장된 데이터들은 궁극적으로 물리적인 디스크에 쓰여져야 한다. Cache에 저장된 Data와 Parity Data는 백그라운드에서 Update되며 이런 과정을 통해 갱신된 Data를 Disk로 옮기는 것을 Destage라고 한다. 이러한 destage 작업은 destage 알고리즘(반출 혹은, destage 스케줄러)에 의해서 수행되며, destage 알고리즘 설계시 중요하게 여기는 이슈는 첫째, 언제 destage 작업을 시작하고 중단할 것인가에 대한 결정 문제와 둘째, 캐쉬에 저장된 어떤 블록을 선택하여 destage 작업을 수행 할 것인가에 대한 결정 문제이다. High/Low Water Mark 알고리즘은 destage가 시작 혹은 중단하기 위한 두 캐쉬 점유율에 해당하는 threshold를 정해 놓은 간단한 알고리즘인 반면에 Linear Threshold 알고리즘은 캐쉬 점유율이 높아짐에 따라서 destage 속도를 빠르게 하는 좀더 지능적인 알고리즘이다. 하지만, Linear Threshold 알고리즘은 디스크의 현재 헤더 위치를 추적하고 destage할 디스크 블록들에 대한 destage 값을 계산해야 하는 부가적인 작업으로 인하여 High/Low Water Mark 알고리즘에 비해서 CPU 오버헤드가 크다.

호스트의 쓰기 요구에 대해 오버플로우를 최소화하고 쓰기 캐쉬의 적중률을 높이기 위해 기존의 HLWM 방법에서 정적인 Threshold로 인한 문제점을 동적으로 대응할 수 있도록 하고 높은 캐쉬 점유율을 유지하게 함으로써 쓰기 요구에 대한 높은 캐쉬 적중률을 얻을 수 있도록 하며 또한, 대량으로 쓰기 요구가 증가하여도 캐쉬의 오버플로우를 최소화 할 수 있도록 하며 그리고 이로 인해 전체적으로 사용자 응답 시간을 줄여

주는 효과를 줄 수 있는 알고리즘을 제안하였다. 제안한 알고리즘은 다음과 같은 처리 과정을 가지고 있다.

첫째, 호스트로부터 쓰기 요구가 발생되면 제어기 내의 캐쉬에서 먼저 처리되고 쓰기 요구에 해당하는 블록이 캐쉬에서 적중되면 이들은 즉시 캐쉬에서 갱신한다. 둘째, 현재 제어기로 요구되는 쓰기 요구의 시간에 따른 변화량과 현재 쓰기 캐쉬에 남아있는 여유 공간의 정도에 따라 반출 알고리즘에 적용되는 높고 낮은 Threshold를 동적으로 변경하여 시스템의 상태에 따른 캐쉬관리를 할 수 있도록 한다. 즉, 시간에 따른 쓰기 요구의 변화량이 기준보다 많고 적음에 따라 그리고 쓰기 캐쉬의 여유 공간이 많고 적음에 따라 Threshold를 변경한다. 셋째, 일반적인 요구상태에서는 HLWM와 동일하게 동작을 한다. 하지만, 쓰기 요구의 증가로 인한 변화량이 기준보다 큰 경우 평소에 비해 낮은 레벨에서 반출을 시작하도록 준비한다. 이때 사용되고 있는 캐쉬의 점유 레벨이 중간을 기준으로 해서 100%에 가까워질수록 반출을 시작하는 레벨을 조금씩 더 낮아지도록 하여 쓰기 캐쉬에 충분한 빈 공간을 만들어 새로운 쓰기 요구에 효과적으로 대응할 수 있도록 한다.

반면 쓰기 요구의 증가량이 기준보다 작아서 반출로 인한 쓰기 캐쉬의 점유율이 비정상적으로 낮아지는 경우에는 앞선 경우와는 반대의 정책을 사용한다. 캐쉬의 점유 레벨이 중간을 기준으로 해서 0%에 가까워질수록 반출을 멈추기 시작하는 레벨을 조금씩 높여서 쓰기 캐쉬에 많은 수의 블록들을 저장하도록 하여 쓰기 요구에 대한 적중률을 높일 수 있도록 한다.

쓰기 캐쉬에 새롭게 쓸 수 있는 여유가 많을 때에는 보통의 부하상태에서는 오버플로우를 방지할 수 있는 여건이 되지만 여유가 적을 때에는 보통의 부하상태에서도 쉽게 오버플로우될 수 있기 때문에 반출을 조금 빨리 시작한다. 반출 요구가 많을 때에는 캐쉬에 새롭게 쓸 수 있는 여유가 많은 반면 호스트의 쓰기 요구가 캐쉬에서 적중될 확률이 떨어지게 된다. 이런 경우에는 반출을 멈추는 시점을 조금 빨리 적용하여 적정량의 블록들이 캐쉬에 저장되도록 하면 적중률을 높일 수 있다. 부하상태 즉, 캐쉬 사용량의 변화율을 통해 현재 사용량과 비교 적용하여 캐쉬의 오버플로우를 최소화하기 위한 방법으로 캐쉬에서 Dirty 블록을 평소보다 많은 양을 반출하도록 하며 반대의 경우 반출을 조기에 중지하게 함으로써 적당한 사용량을 유지하도록 한다.

쓰기 캐쉬의 사용 정도와 쓰기 캐쉬 사용량의 변화율에 따른 동적 Threshold 방법에 대해 간단히 정리하면, HLWM 알고리즘의 *High Threshold*와 Low Threshold이 기본으로 하고, 쓰기 캐쉬 사용량의 변화량을 적용한 간단한 수식을 통해 Offset을 구한다. 얻어진 Offset에 쓰기 캐쉬의 사용정도를 적용하여 새로운 Offset을 구한 후 기존의 HLWM 알고리즘에서의 기본 Threshold값에 적용하여 새로운 Threshold값을 얻고 이를 쓰기 캐쉬에 적용한다.

IV. 시뮬레이션 및 성능비교

반출 알고리즘에 따른 RAID-5의 성능은 기존에 연구된 최소 비용 스케줄링 알고리즘, 상하제한(HLWM) 알고리즘과 본 연구에서 제안된 동적 반출(AT) 알고리즘에 대한 성능을 비교함으로써 평가될 수 있다. 성능평가 요소에는 읽기 요구 응답 시간(Average Host Read Response Time)과 쓰기 캐쉬의 적중률(Write Cache Hit Ratio)등이 있다.

본 논문에서 제안한 동적 적응 반출 알고리즘과 기존에 연구되어진 반출 알고리즘들의 성능을 비교 평가하기 위하여 먼저 반출 알고리즘별로 성능 평가 요소들을 측정하였다. 여기에서 쓰기 캐쉬의 크기는 2, 4, 8MB이고 분할 단위의 크기는 8KB로 하고, 디스크 캐쉬 블록의 크기도 8KB 단위를 사용하였다. 읽기 캐쉬는 쓰기 캐쉬의 2배인 4, 8, 16MB를 사용하였다.

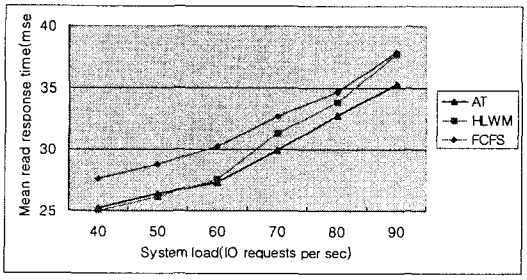

그림 1 시스템 부하에 따른 읽기 응답시간의 변화 비교
Fig. 1. Comparison of mean response time of disk reads *in RAID-5 system with 8MB write cache*

성능 평가 요소 중 컴퓨터 시스템의 성능에 가장 큰 영향을 주는 요소는 읽기 요구에 대한 응답 시간이다. 그림 1은 시스템 부하의 변화에 따른 응답 시간의 변화를 반출 알고리즘별로 나타낸다. 여기서 동적 적응 반출 알고리즘은 FCFS 스케줄링 알고리즘에 비해서는 우수한 성능을 나타내지만 낮은 부하에서는 HLWM 스케줄링 알고리즘과 거의 비슷한 성능을 나타낸다. 이는 본 연구에서 제안된 반출 알고리즘이 기본적으로 HLWM 스케줄링 알고리즘을 적용하고 있기 때문이다.

HLWM 스케줄링 알고리즘은 정적이기는 하지만 캐쉬 점유율을 조절하여 평균적으로 높은 캐쉬 점유율을 유지할 수 있도록 설계되어 있으므로 읽기 요구들이 쓰기 캐쉬에서 적중되거나 쓰기 요구들이 쓰기 캐쉬에 재 저장(re-write) 될 확률이 다른 알고리즘에 비하여 높다. 또한 제안된 알고리즘은 기존에 제안된 반출 알고리즘들에 비해 시스템의 부하가 큰 경우에도 우수한 성능을 보인다. HLWM 스케줄링 알고리즘의 경우에는

상위 하위 Threshold에 의해 반출이 수행되고 쓰기 캐쉬의 점유율은 50% 정도를 유지되었다. 또한, 쓰기 캐쉬의 점유율이 제안된 알고리즘에 비해 상위Threshold에 근접하는 횟수도 잦고 쉽게 도달한다. 반면 제안된 동적 적응 반출 알고리즘의 경우 쓰기 요구의 변화량과 캐쉬의 점유율에 따른 동적인 Threshold를 제안함으로써 다른 알고리즘에 비해 급격히 변화하는 구간이 적어 제안된 알고리즘은 안정적인 쓰기 캐쉬 점유율을 보여주고 있다.

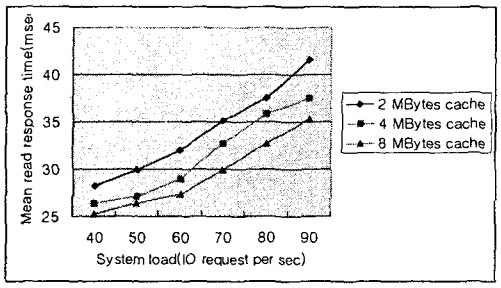

그림 2 쓰기 캐쉬의 크기에 따른 읽기 응답시간의 변화 비교
Fig. 2. Performance(response time) of disk reads with various write cache size

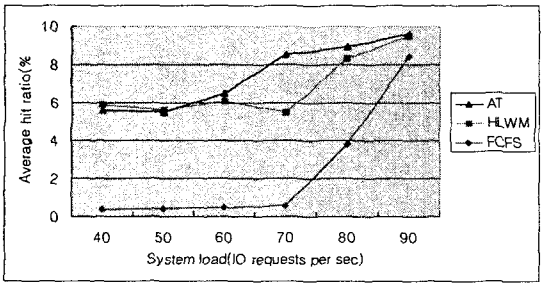

그림 3 시스템 부하에 따른 쓰기 요구의 적중률 비교
Fig. 3. Comparison of write cache hit ratio with various scheduling algorithms

캐쉬 점유율의 변화와 쓰기 요구의 급격한 변화를 통해 곧 캐쉬가 오버플로우할 것을 예측하고 조기에 Dirty block들을 반출함으로써 디스크 시스템의 성능 저하를 막을 수 있다. 실험 결과 제안된 알고리즘은 HLWM 알고리즘에 비해 비교적 높은 캐쉬 점유율을 얻었다. 그림 2는 시스템 부하 즉, 쓰기 요구가 적을 때는 약간 성능이 저하되는 것을 제외하면 동적 적응 반출 알고리즘은 비교적 높은 캐쉬 점유율로 인해 적중률이 보다 높음을 보여주고 있다.

그림 3은 각각 알고리즘별 디스크 읽기 응답 시간의 변화와 동적 적응 알고리즘에서 쓰기 캐쉬의 크기에 따른 응답 시간의 변화를 나타낸다. 쓰기 캐쉬가 증가하면 읽기 요구들이 쓰기 캐쉬에서 적중될 확률이 증가되고, 이에 따라서 평균 응답 시간이 줄어들게 된다. 쓰기 캐쉬에 저장되는 반출 요구들이 증가되면 반출하기 적당한 요구들이 많아지게 된다. 따라서 최적의 반출 요구들을 선별하여 스케쥴할 수 있는 확률이 증가되므로 반출에 필요한 대기시간이 줄어들게 되어 성능이 향상된다.

V. 결 론

동적 적응 반출 알고리즘은 RAID-5의 디스크로의 쓰기 요구를 제어기내의 쓰기 캐쉬에 우선 쓰게 함으로써 물리적인 디스크로의 접근을 최소화하여 호스트 입출력 요구들의 처리시간이 증대되는 문제를 줄일 수 있다.

본 논문에서는 기존의 HLWM 반출 알고리즘에 호스트의 쓰기 요구 변화율과 쓰기 캐쉬의 사용 정도 정보를 부가한 새로운 RAID-5 캐쉬 모델을 구성하고 모의 실험을 통하여 동적 적응 반출 알고리즘이 RAID-5의 쓰기 캐쉬를 동적으로 관리하도록 하여 시스템의 성능을 효과적으로 증대시킬 수 있음을 증명하였다. 특히 기존의 디스크 성능 향상에 관련된 연구들이 구현이 단순하되 정적이거나 혹은 동적으로 구성은 가능했지만 쓰기 요구의 급작스런 변화에 효과적으로 대응하지 못 하였던 점과 비교하여 제안된 알고리즘은 반출 알고리즘이 동적이면서도 구현이 단순한 형태를 지니고 있으며 효율적으로 쓰기 캐쉬를 관리한다. 따라서 동적 적응 반출 알고리즘은 RAID-5의 입출력 성능을 상당히 향상시킬 수 있음을 보였다.

참고문헌

[1] J. Menon and J. Cortney, "The architecture of a fault-tolerant cached RAID controller," *Proceedings of the 20th International Symposium on Computer Architecture*, pp 76-86, May 1993.

[2] A. Varma and Q. Jacobson, "Destage algorithms for disk arrays with nonvolatile caches," *IEEE Trans. on Computers*, Vol. 47, No. 2, pp. 228-235, Feb. 1998.

[3] D. Patterson, G. Gibson, and R. Katz, "A case for redundant arrays of inexpensive disks (RAID)," *Proceedings of IEEE COMPCON*, pp. 112-117, Spring 1989.

[4] D. M. Jacobson and J. Wilkes, "Disk Scheduling Algorithms Based on Rotational Position," *Technical Report HPL-CSP-91-7, Hewlett-Packard Laboratories*, Feb. 1991.

[5] M. G. Baker and et al, "Non-Volatile Memory for Fast, Reliable File Systems," *Proceedings of the 5th International Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 1992.