

가상 머신을 위한 I/O 요청 예측 기법

이재면, 이승한, 강경태
 한양대학교 컴퓨터공학과
 e-mail : {Jaemyoun, shlee10, ktkang}@hanyang.ac.kr

A I/O Request Prediction Mechanism for Virtual Machines

Jaemyoun Lee, Seunghan Lee, Kyungtae Kang
 Dept. of Computer Science & Engineering, Hanyang University

요 약

클라우드 서비스와 크로스 플랫폼이 확산되는 추세와 맞물려 가상 머신의 File I/O 성능 개선을 위한 노력이 다각도로 모색되고 있다. 본 논문에서는 운영 오버헤드는 줄이면서도 정확도가 높은 File I/O 예측 기법을 유전알고리즘 기반으로 제안한다. 특히, 선택 연산만이 대기 시간에 포함되고, 기타 연산은 후면 처리로 수행함으로써 워크로드에 따라 대기 시간을 크게 줄일 수 있을 것으로 기대한다.

1. 서론

Virtual Machine (가상 머신)은 Cloud Services (클라우드 서비스)에서 필수 요소이다[1]. 더구나 모바일 플랫폼 확산과 Internet of Things (IoT) 디바이스의 등장으로 Cross Platforms (크로스 플랫폼)을 지향하는 움직임이 커지고 있다. 상대 플랫폼과 통일성을 갖기 어려운 플랫폼들을 묶은 크로스 플랫폼을 지원하기 위해, 가상 머신은 필요 불가결하다. 가상 머신은 Full-Virtualization (전가상화)와 Para-Virtualization (반가상화)으로 나뉘고, 각각 Guest Machine (게스트 머신)이 Host Machine (호스트 머신) 내부에 존재하거나, 호스트 머신의 자원을 게스트 머신에게 공유하는 방식을 취한다. 두 방식 모두 직접 연결되는 단일 머신에 비해 오버헤드가 있어, 성능이 저하된다. 한 컴퓨터 당 동작 가능한 머신 수는 경제적 문제와 맞닿기 때문에 성능 향상은 큰 도전 과제이다.

가상 머신의 성능 저하의 주된 원인은 입출력 (Input/Output, I/O) 장치이다. File I/O 요청의 latency time (대기 시간)은 사람이 인지할 정도로 길다. 또한, I/O 요청 프로세스는 Mutual Exclusion (상호 배제)를 요구하기 때문에 한 게스트 머신이 디스크를 선점할 경우 다른 게스트 머신은 대기해야 한다. 이로 인해, 한 게스트 머신이 디스크를 선점하는 시간을 최소화하는 것이 필요하다.

게스트 머신이 디스크를 선점하는 시간을 줄이기 위해, 가장 적합한 방법은 I/O 요청처리 속도를 빠르게 하는 것이다. 물리장치를 이용하는 하드디스크는 속도를 향상시키는데 한계가 있으며, Solid State Disks (SSDs) 역시 최적의 해는 아니다. 그러므로 절대적 성능을 향상시키기 보다 효율을 높이는 Prefetching (프리페칭)이나 SSD 캐시 기법이 제안된다[1]. 프리페칭이나 캐싱 기법은 오랫동안 연구했으나, 아직 가상

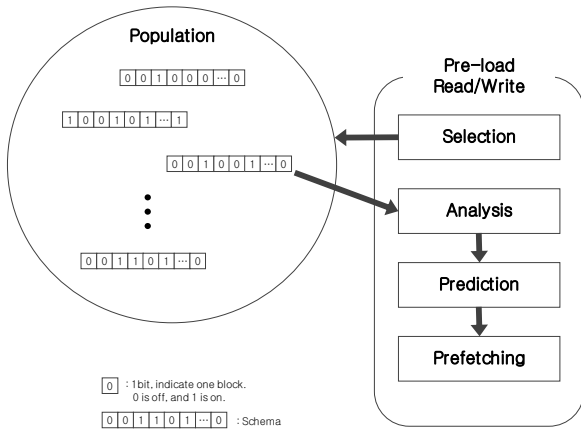
머신에서 뚜렷한 성과가 나오지 않았다. 이는 게스트 머신의 요청이 캡슐화되어 호스트 머신에서 I/O 패턴을 예측하기 어렵기 때문이다.

본 연구는 적은 오버헤드를 가지고 효율이 높은 예측 알고리즘을 제안한다. 스키마 집합에서 스키마 하나를 선택하고, 선택한 스키마를 기반으로 예측한다. 스키마는 과거 I/O 접근 패턴이 기록되어 있다. 제안한 모듈은 스키마로부터 연관된 데이터를 추출하고 프리페칭 또는 캐싱하는데 도움을 준다. 스키마는 유전 알고리즘 기법으로 관리된다. 사용자가 응답시간이라고 느끼는 요청 수행 완료까지는 선택 연산만 수행하기 때문에 오버헤드가 적으며, 고효율의 정책을 수립할 수 있다.

2. 배경

하드디스크의 경우 보통 7200 RPM (Revolutions per Minute)으로 회전하며, 최대 15,000 RPM 까지 올라가나, 고속의 RPM은 많은 열과 큰 진동을 유발하여 현실적으로 사용하기가 어렵다. 실례로 SSDs가 등장하면서 서버용 디스크로 제작되던 10,500 RPM 하드디스크는 발열, 진동, 소음, 소비 전력 문제로 현재 수요가 없어지고 공급 또한 중단되는 상황이다. SSD는 반도체를 이용하여 디스크를 구성한 것으로써, 하드디스크에 비해 진동, 소음, 소비 전력이 적으며, 빠른 처리 속도를 보인다. 그러나 운영체제 호환성, 반도체 수명, 가격은 단점으로 지적 받는다. 특히, Giga bytes 당 가격이 2 배 이상 비싸기 때문에 대용량 서버를 구성하는 클라우드 시스템에서 주된 디스크로 사용하기는 TOC (Total of Costs) 관점에서 적합하지 않다[1].

이런 속도 향상의 한계를 극복하기 위한 기존 연구는 순차 접근이 빠른 하드디스크의 장점과 Native Command Queuing (NCQ)를 이용하여, 앞으로 읽을 것



(그림 2) Selection Operation

으로 예상되는 블록을 미리 읽는 프리페칭이나 소규모의 SSD 를 캐시로 사용하는 SSD 캐시 기법을 사용한다. 이런 연구는 모두 앞으로 일어날 요청을 예측한다는 공통점이 있다. 그러나 I/O 요청은 수 ms 단위로 이루어지기 때문에 복잡한 연산을 통한 예측 알고리즘은 사용할 수 없다.

3. 방법론

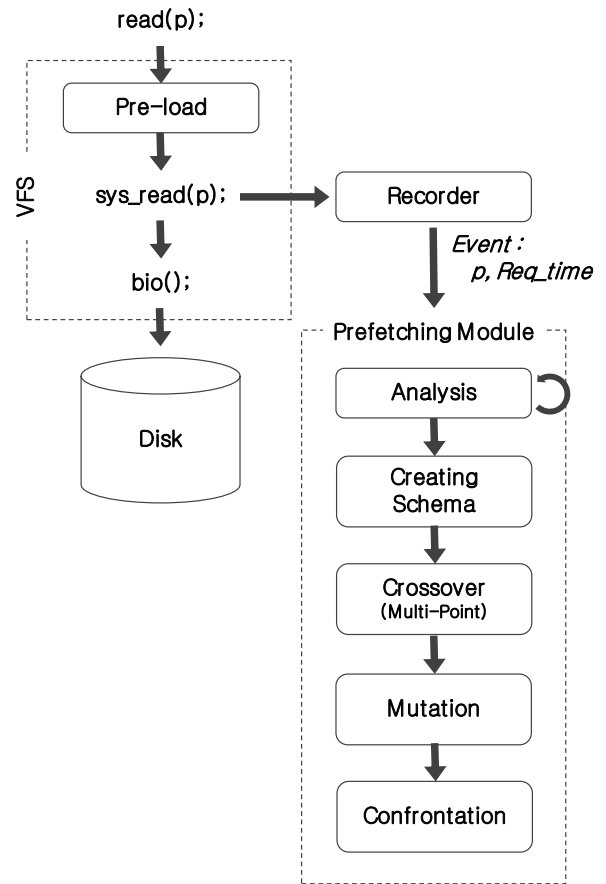
유전 알고리즘 기법을 사용하여 스키마를 관리한다. I/O 요청이 들어왔을 때 선택 연산이 수행된다. 그 후 I/O 요청을 수행하면서 요청 패턴을 기록한다. 그리고 요청이 끝난 후 background 프로세스로 교차 연산과 변이 연산, 대치를 수행한다.

스키마는 저장 장치 블록 전체를 인코딩하고 있으며, 한 비트당 한 블록을 가리킨다. 다시 말해, 한 블록이 128 KB 인 1.0 TB 저장 장치의 스키마 길이는 1,048,576 bytes 이다. 스키마의 개수는 세대 수이며, 정책에 따라 가변이다.

선택 연산은 sys_read 시스템콜의 전처리기로 호출되는데, 세대 중 스키마 하나를 선택한다. 선택한 스키마는 요청한 블록을 대표하는 비트가 on 상태이어야 한다. 선택한 스키마를 기반으로 프리페칭 또는 캐싱할 블록을 선별한 후 가상파일시스템(Virtual File Systems)에 알린다. 다음으로 sys_read 시스템콜을 수행하면서, Recorder 에 로그를 남긴다. Recorder 는 로그를 기록하고, Prefetching Module (PM)에 이벤트를 발생시킨다. PM 은 분석 과정을 거친 후 조건에 따라 스키마 생성, 교차 연산, 변이 연산, 대치를 수행한다.

PM 은 Recorder 가 기록한 요청 블록 주소와 요청 시간을 기준으로 짧은 시간 내에 연속해서 요청한 내역을 한 쌍으로 묶어 스키마를 만든다. 이렇게 생성한 스키마와 선택 연산과는 반대로 관련성이 적은 스키마를 가지고 교차 연산을 수행한다. 교차 연산은 2n 점 교차를 수행한다. 여기서 n 은 정책에 따라 달라진다. 그 후 일정 확률로 변이 연산을 수행한 후 선택 연산에서 선택한 스키마와 교체하여 대치한다.

변이 연산의 확률 역시 정책에 따라 달라지며, 방법은 유사 스키마를 세대에서 하나 더 선택하여 두 스키마를 병합한다. File I/O 의 스키마의 특징이 전혀 관련 없는 블록과 연관될 필요가 없으므로 유사한 스



(그림 1) Proposed Prediction Genetic Algorithm
키마와 병합하는 방법을 취했다. 병합할 때 서로 배타적인 부분은 모두 on으로 반영할지, off로 반영할지를 확률로 정하고, 정책을 수립할 때 결정한다. 프리페칭 또는 캐싱 공간이 클 경우 on 확률이 높은 것이 좋으며, 공간이 부족할 경우 낮은 것이 좋다.

제안한 방법 중 선택 연산만이 사용자 요청 대기 시간에 속하고 다른 연산은 background 프로세스에 의해 수행하기 때문에 적은 오버헤드를 가지면서 지능적 예측 알고리즘을 적용할 수 있다. 본 연구를 통해 가상 머신의 성능 향상을 볼 수 있을 것으로 기대한다.

4. 결론

가상 머신의 성능을 향상시키기 위해 프리페칭이나 캐싱을 사용하고, 두 모듈에서 I/O 요청을 예측하기 위한 기법을 제안한다. 제안한 기법은 적은 오버헤드를 가지며 유전 알고리즘을 사용하여 정확도를 향상시켰다. 앞으로 제안한 기법을 구현 및 실험하여 오버헤드를 분석하고 워크로드에 맞는 정책을 수립할 필요가 있다.

참고문헌

[1] J. Lee and K. Kang, "Assessment of DM-Cache Running on Virtual Linux," in Proc. 4th IEEE International Conference on Big Data and Cloud Computing (BDCloud), pp. 275-276, Dec. 2014.