

비 휘발성 메모리를 사용한 애플리케이션 시작 시간 개선 기법

조용운*, 김태석*
*광운대학교 컴퓨터공학과
e-mail : ssangkong@kw.ac.kr

Application Launching Time Reduction Technique with Non-volatile Memory

Yongwoon Cho*, Taeseok Kim*
*Dept. of Computer Engineering, Kwangwoon University

요 약

본 논문에서는 애플리케이션 실행에 필요한 파일들을 검출하고, 그 파일들을 저 용량의 비 휘발성 메모리에 옮겨 시작 시간을 단축시킨다. 또한 각각의 파일들은 전체가 필요하지 않고 파일 중 일부만 필요하기 때문에, 필요한 부분만 주 메모리에 선 적재 함으로써 시작 시간을 크게 개선하였다.

1. 서론

최근 PCRAM, RRAM, STTMRAM 등 비 휘발성 메모리에 대한 연구가 활발하게 이루어지고 있다. 하지만 아직 이 장치가 상용화되지 않았기 때문에, 해당 장치를 활용한 소프트웨어를 개발하기가 쉽지 않다. 그래서 개발자들을 위해 비 휘발성 메모리를 애플리케이션 해주는 장비도 등장하고 있다.[1] 또한 DRAM 을 램디스크로 활용하여 소프트웨어적으로 시뮬레이션 할 수 있다.

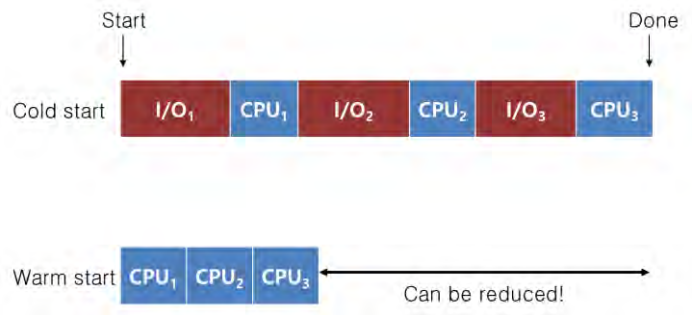
2. 배경지식

비 휘발성 메모리는 용어 그대로 비 휘발성의 특성을 가진 메모리를 의미한다. 또한 비 휘발성의 특성과 바이트 단위의 접근이 가능하다는 특성 있다. 그래서 이 비 휘발성 메모리를 활용하여 주 메모리처럼 사용할 수도 있고, 저장장치처럼 사용할 수도 있다. 궁극적으로는 주 메모리와 저장장치 모두를 대체하는 구조가 될 것이다. 하지만 지금과 같이 직접도가 높지 않은 상황에서는 소량의 비 휘발성 메모리만 가지고 성능을 극대화 시키는 것이 중요한 이슈이다.

애플리케이션 시작 시간은 충분히 기다릴만한 정도로 짧다. 그렇기 때문에 보통은 애플리케이션을 시작시켜 놓고 온전히 그 시간을 아무 작업도 하지 않으며 지나가는 경우가 많다. 따라서 애플리케이션의 시작 시간을 개선하는 것은 중요하다.

애플리케이션의 시작 시간은 대표적으로 두 가지 경우로 나눌 수 있다. 1) “Cold start”, 애플리케이션이 필요로 하는 파일들이 주 메모리에 전혀 적재되어 있지 않은 경우, 필요한 파일의 부분을 적재할 때까지 애플리케이션은 자원을 기다리며 CPU 작업을

진행할 수 없다. 따라서 파일을 주 메모리에 적재하는 과정과 CPU 계산이 교차되며 일어난다. 2) “Warm start”, 애플리케이션이 필요로 하는 파일들이 모두 주 메모리에 적재되어 있는 경우, 오직 CPU 계산만 일어난다. 이는 각각 애플리케이션 시작 시간의 상한값과 하한값으로 볼 수 있다.



(그림 1) Cold start 와 Warm start

애플리케이션 시작 시간 개선은 크게 두 가지로 볼 수 있다. 하나는 CPU 시간을 줄이는 것. 다른 하나는 I/O 시간을 줄이는 것이다. CPU 시간은 더 좋은 계산 능력을 가진 CPU 로 하드웨어를 교체하는 방법과 애플리케이션 코드를 작성할 때, 좀 더 좋은 알고리즘으로 설계하여 소프트웨어를 개선하는 방법이 있다. I/O 시간도 마찬가지로 더 빠른 저장장치로 교체하는 하드웨어적인 방법과 애플리케이션 코드를 최적화하여 I/O를 없애는 소프트웨어적인 방법이 있다.

소스를 공개한 애플리케이션은 코드를 분석하고, 코드를 수정하여 개선 할 수도 있다. 하지만 코드를

분석하는 작업도 만만한 작업이 아니고, 코드를 수정하는 작업 또한 힘든 작업이다. 게다가 애플리케이션마다 따로 작업을 수행 해 줘야 하는 단점이 있고, 소스를 공개하지 않은 애플리케이션은 시도조차 해 볼 수 없다.

따라서 애플리케이션의 코드를 수정하지 않고, 한 가지 방법으로 여러 애플리케이션에 적용하여 시작 시간을 개선하는 방법을 제안한다.

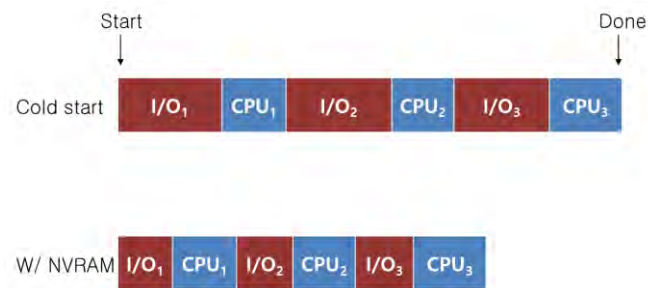
3. 구현내용

3-1. 하드웨어적인 개선

앞서 말한 바와 같이 하드웨어를 개선하여 애플리케이션의 시작 시간을 줄일 수 있다. I/O 시간을 단축 할 수 있도록 기존의 저장장치 대비 전송속도 및 접근속도가 빠른 비 휘발성 메모리를 사용하여 I/O 시간을 단축시키려 한다. 기존의 컴퓨터 구조에서는 애플리케이션의 실행파일 및 관련 라이브러리들이 디스크에 저장되어 있다. 이런 구조에 비 휘발성 메모리를 추가하여 속도를 개선할 수 있다. 하지만 비 휘발성 메모리의 크기가 현재 수준에서는 굉장히 한정적이다. 따라서 전체 디스크를 비 휘발성 메모리로 대체하기는 무리가 있다. 그러므로 우리는 특정 애플리케이션의 최적화를 목표로 그 애플리케이션을 분석하여 애플리케이션이 필요로 하는 파일을 알아내어, 그 파일들만 비 휘발성 메모리에 옮겨서 속도를 향상시킨다.

이때, 실행파일이 필요로 하는 파일을 알아내기 위해 어떤 system call 을 호출하는지 분석할 수 있는 툴인 strace[2]를 사용한다. 이 툴을 이용하여 파일 경로가 인자로 들어가는 system call 인 open, access, execve 등을 추출한다. 이 파일들을 모두 비 휘발성 메모리로 이동시켜야 한다. 하지만 이 파일들은 목표 애플리케이션만 단독으로 사용하는 파일들이 아니다. 따라서 새로운 경로(비 휘발성 메모리)에 해당 파일들을 복사하고, 기존 경로에 소프트 링크를 생성하여 기존 시스템과의 호환성을 유지한다.

이 과정을 거치면, 목표 애플리케이션을 실행할 때, 기존 디스크에 대한 I/O는 없어지게 된다.

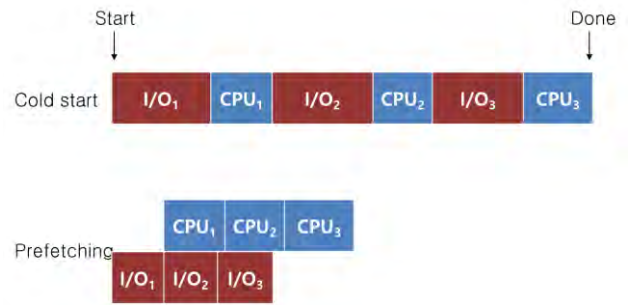


(그림 2) Cold start 와 With NVRAM

3-2. 소프트웨어적인 개선

하드웨어적인 개선을 하여 I/O 시간이 짧아졌지만, 여전히 존재한다. 따라서 이를 소프트웨어적으로도 개선하고자 한다. 애플리케이션에서 필요로 하는 파일의 부분을 선 적재하여 실행속도를 개선하는 방법은 FAST[4]의 방법을 사용한다.

blktrace[3] 라는 툴을 사용하여 디스크의 I/O 를 모두 분석할 수 있다. 이 툴을 사용하여 디스크에 read 요청의 주소 및 크기를 파악하여 선 적재 시킨다. 선 적재는 posix_fadvise 라는 system call 을 사용하여 할 수 있다.



(그림 3) Cold start 와 Prefetching

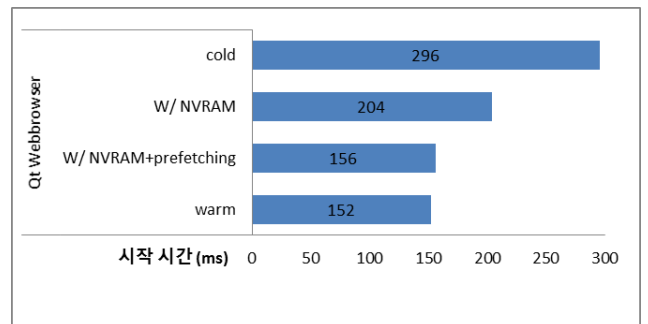
4. 실험 환경 및 결과

4-1. 실험 환경

실험에 사용한 PC 는 Intel Core i7 6 Cores @ 2 GHz CPU, Samsung 840 Pro 128 GB SSD, 16 GB RAM 을 사용한 환경에서 ramdisk 를 사용하여 NVRAM 을 512 MB 시뮬레이션 하였으며, Ubuntu 12.04 64 bit 운영 체제에서 linux kernel v3.19 를 사용하였다. 또한 실험에 사용한 애플리케이션은 Qt Webbrowser 를 사용하였다.

4-2. 실험 결과

cold start 와 warm start 의 시간을 각각 측정하였으며, 하드웨어적인 개선을 한 W/ NVRAM 과 하드웨어 와 소프트웨어적인 개선을 모두 한 W/ NVRAM + prefetching 의 시간을 측정하였다.



(그림 4) Qt Webbrowser 시작 시간

5. 결론

이 방법을 사용하여 하드웨어적인 개선과 소프트웨어적인 개선 모두 활용하여 종합적으로 애플리케이션의 시작 시간을 개선하였다. Qt Webbrowser 로 실험을 하였는데, warm start 와 근접한 결과를 보였다. 아직 다른 애플리케이션을 실험하지 못하였는데, 추후에 실험을 더 진행 할 예정이다.

6. 감사의 글

본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천 기술개발 사업(정보통신)의 일환으로 수행하였음. [No.10041608, 차세대 메모리 기반의 스마트 디바이스용 임베디드 시스템 소프트웨어]

참고문헌

- [1] “TUNA NVM-H2 NVRAM Emulation Board”, <http://www.mangoboard.com/admin/bbs/down.php?code=NVM&idx=8147&no=2>
- [2] “strace”, <http://sourceforge.net/projects/strace>
- [3] “blktrace User Guide”, <http://www.cse.unsw.edu.au/~aaronc/iosched/doc/blktrace.html>
- [4] Joo, Y., Ryu, J., Park, S., & Shin, K. G. (2011, February). FAST: Quick Application Launch on Solid-State Drives. In FAST (pp. 259-272).