

# 스토리지 클래스 메모리를 활용한 즉각 구동 시스템의 개발 (Development of an Instant On System Using Storage Class Memory)

문 영 제 <sup>†</sup>                      도 인 환 <sup>†</sup>  
(Young Je Moon)            (In Hwan Doh)

박 정 수 <sup>†</sup>                      노 삼 혁 <sup>\*\*</sup>  
(Jung Soo Park)             (Sam H. Noh)

**요 약** 스토리지 클래스 메모리 (SCM)는 비휘발성 속성과 바이트 단위의 임의 접근이 가능한 속성을 동시에 보유하고 있는 차세대 메모리 기술으로써 그 활용 방안이 있어서 귀추가 주목된다. 기존 시스템에 SCM을 도입하면 시스템의 수행 속도와 안전성을 크게 향상할 수 있을 뿐만 아니라 기존의 시스템에서는 불가능했던 새로운 특징들을 제공할 수 있다. 본 연구는 혁신적인 용도로의 SCM 활용 가능성에 주목하며, 그 일환으로 SCM을 메인 메모리로 활용하여 종료 상태의 시스템에 전원이 인가되는 즉시 종전의 시스템 상태로 되돌아갈 수 있는 SOONN을 제안한다. 본 논문에서는 실제 임베디드 시스템 환경에서 프로토타입 시스템을 개발함으로써 SOONN의 실현 가능성을 제시한다.

**키워드** : 스토리지 클래스 메모리, 즉각 구동 시스템

**Abstract** Storage Class Memory (SCM) has both characteristics of non-volatility and random byte addressability. The advent of SCM can bring about novel and innovative features that are not possible in conventional computing systems. This paper suggests a new system design that turns on/off a system instantly. To do this, we replace the main memory with SCM to retain the volatile system states as the system is turned off. We implement our prototype in an embedded environment and measure its system on/off time.

**Key words** : Storage Class Memory, Instant On System

## 1. 서 론

스토리지 클래스 메모리(Storage Class Memory; SCM)는 MRAM(Magnetoresistive RAM), PCM(Phase-change RAM), 그리고 FeRAM(Ferroelectric RAM)과 같은 다양한 형태로 연구 개발되고 있는 차세대 메모리 기술의 하나이다. SCM은 비휘발성 속성과 바이트 단위의 임의 접근이 가능한 속성을 동시에 보유한다. SCM의 기술이 계속 발전되어감에 따라, SCM이 가까운 미래에는 일상적인 컴퓨팅 시스템 요소로써 자리하게 될 것으로 예상된다[1,2].

SCM의 도입과 이를 효과적으로 활용하는 시스템 소프트웨어의 개발은 수행 속도와 안정성 측면에서 시스템의 성능을 혁신할 수 있을 뿐만 아니라 기존의 시스템에서는 불가능했던 특징들을 새롭게 제공할 수 있을 것으로 기대된다. SCM을 활용하는 기존 연구들은 SCM이 메모리 계층 구조에서 효과적으로 활용되었을 때 시스템의 성능과 안정성이 향상될 수 있음을 보여주었다[3,4]. 지금까지 대부분의 연구들은 단순히 빠른 저장 매체로써의 SCM 활용을 고려하고 있지만 최근 SCM의 새로운 활용 가능성에 대한 연구들이 등장하고 있다. 본 연구는 이처럼 SCM의 새로운 활용 가능성에 주목하여 SCM을 메인 메모리로 활용함으로써 시스템의 종료와 구동이 즉각적으로 이루어질 수 있는 시스템인 SOONN(System On/Off iNstaNtly)을 제안한다.

본 연구에서 고려하는 SCM을 사용한 SOONN은 전원 인가 시 이전의 종료 직전 시스템 상태로 즉각적으로 되돌아갈 수 있는 시스템이다. 휘발성 메인 메모리를 사용하는 기존의 시스템은 전원이 차단되면 항상 초기 시스템 상태로 전환되기 때문에 종료 상태에서 시스템이 운용될 수 있는 상태가 되기 위해서는 다수의 연산들을 필수적으로 수행해야 한다. 하지만 SCM이 메인 메모리의 일부분 혹은 전체로 사용되어 종전의 시스템 상태를 전원이 차단된 후에도 유지할 수 있다면, 해당 시스템은 즉각적으로 종전의 시스템 상태로 복귀될 수 있으며 종료 과정 또한 단순화 될 수 있다.

- 이 논문은 2007년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. ROA-2007-000-20071-0)
- 이 논문은 2009 한국컴퓨터종합학술대회에서 '스토리지 클래스 메모리를 활용한 즉각 구동 시스템의 개발'의 제목으로 발표된 논문을 확장한 것입니다

<sup>†</sup> 학생회원 : 홍익대학교 컴퓨터공학과  
lemonkenya@naver.com  
cmzace@naver.com  
inhwando@hotmail.com

<sup>\*\*</sup> 종신회원 : 홍익대학교 컴퓨터공학과 교수  
samhnoh@hongik.ac.kr

논문접수 : 2009년 8월 14일  
심사완료 : 2009년 11월 10일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제2호(2010.2)

본 논문은 SOONN 프로토타입 시스템을 개발하여 즉각적인 시스템 복귀 및 종료가 실현 가능함을 제시한다. 실제 임베디드 시스템 환경에서 개발된 SOONN 프로토타입 시스템의 성능 평가 결과, 프로토타입 시스템은 1초 이내로 종전 시스템 상태로의 복귀가 가능하며 종료 과정은 복귀 과정보다 더 빠르게 진행됨을 확인할 수 있다.

2. 관련 연구

시스템의 부팅 속도를 향상시키려는 연구들에는 다양한 연구들과 상용화된 기법들이 존재한다. 먼저, 시스템 부팅 시에 실행되는 운영체제와 시작 프로그램의 실행 파일들을 일반 하드디스크보다 읽기 접근 속도가 뛰어난 메모리 기반 SSD(Solid State Disk) 장치에 캐시함으로써, 시스템 부팅 과정에서 실행 파일을 메인 메모리로 읽어 들이는 시간을 단축하고자 하는 기법이 제안되었다[5]. 이 기술은 종래의 하드디스크보다 단순히 더 빠른 저장매체를 사용했기 때문에 부팅 속도 향상에 대한 기대 효과가 극히 제한적이라고 할 수 있다.

범용 운영체제의 부팅 과정은 많은 시간 지연을 유발하기 때문에 범용 운영체제보다 작은 실행 이미지를 가지면서 최소한의 서비스 기능을 갖는 운영체제로 시스템을 부팅함으로써 부팅 시간을 단축하고자 하는 기법들도 널리 쓰이고 있다[6]. 그렇지만 이러한 기법들의 가장 큰 제약 사항은 최소화된 새로운 운영체제가 필요하다는 것과 소형화된 운영체제가 지원하는 사용자 서비스가 매우 제한적이기 때문에 해당 운영체제가 지원하는 것 이상의 서비스를 원하는 사용자에게는 범용 운영체제로의 재부팅이 불가피하다는 것이다. 본 연구에서 제안되는 시스템은 SCM이라는 새로운 형태의 하드웨어의 도입과 혁신적인 활용을 통해서 즉각적으로 종전 시스템 상태로의 복귀가 가능한 시스템을 제안한다는 점에서 기존의 연구 또는 방식들과 차별된다.

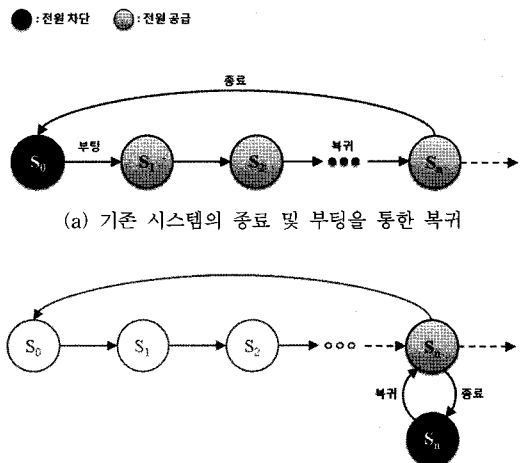
3. SOONN(System On/Off iNstaNtly)

본 연구에서는 수행중인 시스템이 종료 요청을 수신하면 즉각적으로 전원 공급이 완전히 중단된 상태로 전환되며, 반대로 시스템이 구동 요청을 수신하면 즉각적으로 종전의 시스템 상태로 복귀가 가능한 시스템(SOONN)을 제안한다. SOONN은 시간 지연 없이 사용자의 요청에 따라서 언제든지 시스템을 구동하고 종료할 수 있는 차세대 시스템이다. 이에 따른 다양한 부가적 이득에는 시스템이 불필요하게 켜져 있는 시간의 감소로 에너지를 절약할 수 있고 시스템의 소음 혹은 전자파 노출과 같은 문제를 최소화할 수 있다. 본 연구에서는 이러한 매력적인 SOONN을 임베디드 시스템의 프

로토타입에서 직접 실현한다. 앞으로의 논의에서, ‘부팅’이라는 용어는 시스템이 사용자의 요청을 처리할 수 있는 초기 상태가 되는 과정을 의미하며, ‘복귀’라는 용어는 종료 상태의 시스템을 이전의 종료 직전 상태로 되돌리는 것을 의미한다.

그림 1은 전원의 인가와 종료에 따른 기존 컴퓨팅 시스템 상태 변화와 이상적인 형태의 SOONN의 상태 변화를 도식한 상태 천이 다이어그램이다. 그림에서 흑색과 회색으로 채워진 원형은 각각 전원 공급이 완전히 중단된 상태와 전원이 공급되고 있는 상태를 의미하며 흰색으로 채워진 회미한 원형은 기존 시스템에서 SOONN으로 변경됨으로써 생략이 되는 상태를 의미한다. 또한 S0상태는 시스템이 어떠한 상태도 가지고 있지 않은 상태를 나타내고 Sn상태는 외부로부터 시스템이 종료 요청을 수신하기 직전의 시스템 상태를 의미한다.

그림 1(a)에서 기존 시스템은 시스템 종료 과정을 수행한 후에 전원 공급이 차단되면 휘발성 메모리에서 관리되는 시스템 상태 유지에 필수적인 정보들이 소멸되기 때문에 항상 시스템의 상태가 소실된 S0상태가 된다. S0상태에서 전력이 재공급되었을 때, 종료 직전의 Sn상태로 복귀하기 위해서는 부팅 과정에서 다수의 연산들을 순서대로 수행하면서 S1상태와 S2상태 같은 특정한 시스템 상태들을 거친 후 종전에 실행되던 시스템 상태인 Sn상태로 천이된다. 이처럼 기존의 시스템에서 복귀 과정은 부팅 과정을 포함한다. 만약 기존의 시스템에서 휘발성 속성을 가진 시스템 상태 정보들이 모두 비휘발성 속성을 가지게 된다면, 그림 1(b)처럼 해당 시스템은 전력의 공급 유무와 상관없이 항상 시스템 상태 유지에 필수적인 정보들을 유지할 수 있게 된다. 이러한



(a) 기존 시스템의 종료 및 부팅을 통한 복귀

(b) 이상적인 방식으로 동작하는 SOONN의 종료 및 복귀  
그림 1 전원의 공급/차단에 따른 상태 천이 다이어그램

시스템은 시스템 종료 후 전원 공급이 중단된 상황에서도 여전히 종료 직전의 Sn 상태를 유지하고 있으므로 전원이 재공급될 때 즉시 종전의 상태로 복귀하는 것이 가능해진다.

SCM의 도입과 이를 효과적으로 활용하는 시스템 소프트웨어의 사용은 기존의 시스템에서 휘발성 속성을 가진 시스템 상태 정보들을 모두 비휘발성 속성으로 전환하여 SOONN의 실현을 가능하게 한다. SOONN을 실현하기 위해서 여러 가지 접근 방식이 존재한다. 먼저 가장 이상적인 형태로 메인 메모리 및 프로세서와 주변 장치의 레지스터를 모두 비휘발성 매체로 전환하는 방식이 있다. 이러한 하드웨어적인 접근 방식은 기존 시스템들과의 호환성과 하드웨어 제조업체들의 참여도를 고려할 때 현실적으로 거의 불가능하다. 둘째, 메인 메모리를 비휘발성 속성으로 변경하고 메인 메모리를 제외한 나머지 휘발성 속성의 시스템 상태 정보들을 종료 직전에 SCM 혹은 저장매체에 기록해 두는 방식이다.

메인 메모리를 비휘발성 속성으로 변경하는 방법에는 세 가지 현실성 있는 방식이 존재한다. 먼저, 종료 전에 메인 메모리의 내용을 모두 비휘발성 저장장치에 압축하여 저장하는 방식이다. 메모리 간의 데이터 전송이기 때문에 디스크나 플래시 메모리에 저장하는 것보다 빠른 시간 내에 쓰기 작업이 완료될 수 있다. 둘째, 메인 메모리를 SCM으로 대체하는 것이다. 그렇게 되면 비휘발성 메인 메모리에 커널 및 프로세스 관리 테이블, 페이지 캐시와 같이 이전 상태에서 사용되던 모든 메모리 상태 정보를 포함하기 때문에 종료 과정에서 추가적인 작업을 필요로 하지 않는다. 셋째, 커널을 수정하여 메인 메모리에 들어갈 시스템 상태 정보 중에 시스템 복귀를 위해서 반드시 유지해야 하는 모듈과 그렇지 않은 모듈을 구분하여 복귀를 위해 꼭 필요한 모듈만 비휘발성 메모리에 저장하고 나머지는 휘발성 메모리에 저장하는 방식이 있다. 첫 번째 방식은 현재의 비휘발성 저장매체의 접근 속도를 고려할 때 종료와 부팅 시 메인 메모리와 비휘발성 저장장치 간의 복사 오버헤드가 발생하므로 즉각적인 복귀를 제공하기 힘들다. 세 번째 방식은 SCM을 적게 사용할 수 있지만 구현의 측면에서 복잡도가 매우 높기 때문에 본 논문에서는 두 번째 접근 방식을 고려하여 SOONN 프로토타입 시스템을 개발하고자 한다.

#### 4. 프로토타입 시스템 개발

##### 4.1 시스템 동작 방식

SCM을 사용한 SOONN 프로토타입 시스템의 동작 방식은 기존 시스템의 절전 모드 가운데 하나인 STR(Suspend To RAM) 기법에 그 기반을 둔다. 기존 시

스템의 STR 기법은 시스템의 유휴 상태가 특정 기간 이상으로 지속되면 메인 메모리와 특정한 입출력 장치에만 전력을 공급하고 나머지 장치들은 전원 공급을 차단하거나 최소화함으로써 전력 사용을 줄이기 위한 기술이다. 메인 메모리와 특정 입출력 장치에만 전력을 공급하는 것은 각각 휘발성 메인 메모리에서 관리되고 있는 현재의 시스템 상태를 보존하고 절전 모드에서 실행 모드로의 전환을 요구하는 사용자의 요청을 감지하기 위해서이다. STR 기법을 적용한 시스템은 절전 모드 상태에서 실행 모드 상태로 혹은 그 역으로 즉각적으로 전환될 수 있다.

따라서 비휘발성 메인 메모리를 가진 시스템에서 STR 기법의 동작 방식이 활용된다면, 해당 시스템은 시스템의 종료 시에 전력 공급을 완전하게 차단하고 전력 공급이 재개되었을 때 종전의 시스템 상태로 즉각적으로 복귀하는 SOONN이 될 수 있다.

본 연구에서 고려하는 SOONN 프로토타입 시스템은 그림 2에 나타난 바와 같이 종료와 복귀 시에 기존의 시스템과 차별되는 방식으로 동작한다. 종료 과정은 시스템이 종료되기 전에 현재 시스템의 상태를 저장하는 과정이다. 종료 과정은 크게 3 가지 단계로 동작한다. 첫째, 종료 명령을 내리서 동작중인 프로세스들을 모두 중지시킨다. 예를 들면 임계 구역에서 프로세스가 수행 중인 상황에서는 그 상황이 해소된 후에 해당 프로세스를 중지시킨다. 둘째, 프로세서의 레지스터 정보와 같은 휘발성 메모리의 내용들을 비휘발성 메인 메모리로 복사한다. 전력 공급이 중단되면 프로세서 및 장치의 레지스터들은 휘발성 메모리이므로 그 내용을 소실하는데 이러한 정보들도 시스템의 복귀 과정에서 꼭 필요하므로 SCM에 보존되어야 한다. 셋째, 현재 시스템 상태를 나타내는 모든 정보가 SCM에 저장되었으므로 시스템 전원을 차단한다.

복귀 과정은 시스템에 전원이 다시 공급되면 종료 과정에서 저장해 둔 데이터를 이용하여 시스템을 종전 상태로 되돌려 놓는 과정이다. 복귀 과정이 실행되기 위해

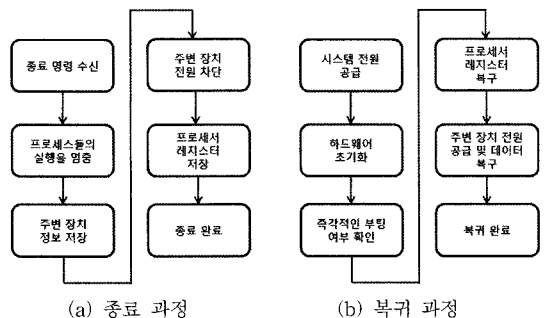


그림 2 SOONN의 종료와 복귀과정 상태 그래프

서는 복귀 코드가 실행되기 위한 기본적인 환경이 필요하므로 부트로더에서 복귀 과정을 실행하도록 설계하였다. 부트로더에서 하드웨어 초기화 작업이 이루어진 후에 다음과 같은 과정을 거쳐서 시스템 복귀가 이루어진다. 먼저 프로세서 및 주변 장치들을 초기화한다. 이어서 종전에 SCM에 저장해 둔 휘발성 속성의 정보들을 프로세서 및 장치의 레지스터에 복사함으로써 복귀 과정이 완료된다.

#### 4.2 프로토타입 시스템 구현

SCM을 활용하는 SOONN 프로토타입 시스템은 실제 임베디드 시스템 환경에서 개발된다. 임베디드 시스템 환경에서 프로토타입을 구현한 이유는 메인 메모리 용량이 적고, 장착되어 있는 주변 장치들이 복잡하지 않아서 프로토타입 구현의 복잡도를 낮출 수 있기 때문이다. 임베디드 시스템 개발 보드는 PXA255 프로세서와 64MB의 SDRAM, 그리고 64MB의 FeRAM SCM을 탑재하고 있다. SCM은 SDRAM처럼 연속적인 물리 메모리 주소 공간 내에 장착되어 프로세서가 직접 접근할 수 있기 때문에 메인 메모리로 동작하는 것이 가능하다.

프로토타입의 종료 과정과 복귀 과정의 코드는 리눅스에 포함되어 있는 STR 기법의 ARM용 패치를 기반으로 구현된다. 프로토타입의 구현에서 종료 과정과 복귀 과정으로 진입하기 위한 인터페이스 부분과 프로세서와 장치의 정보를 따로 보관하는 부분이 활용되었다.

### 5. 성능 평가

본 절에서는 개발된 SCM을 활용하는 SOONN이 기존의 시스템과 비교해서 얼마나 빠른 복귀 및 종료율을 제공하는지에 대해서 정량적으로 평가한다. 평가는 실제 임베디드 시스템 환경에서 이루어지며, 비교 대상으로는 세 가지 방식의 서로 다른 복귀 및 종료 메커니즘을 고려한다.

#### 5.1 실험 환경

타겟 임베디드 시스템은 ARM용 리눅스 2.6.21 버전을 기반으로 동작하며, SOONN이 종전 시스템 상태로 즉각적인 복귀가 가능한지 확인하기 위해서 다음과 같은 추가적인 소프트웨어들이 제작 및 포팅된다. 먼저 GUI 프로그램을 실행시키기 위해 타겟 시스템은 Embedded Qt 3.3.4 라이브러리를 포함한다. GUI 어플리케이션으로써 숫자 증가 프로그램을 제작한다. GUI 숫자 증가 프로그램은 숫자를 0부터 100ms 마다 1씩 증가하는 것을 LCD에 출력하는 프로그램으로써 SOONN이 복귀 과정을 제대로 수행하는지를 확인하려는 목적으로 개발된다. 또한, 숫자 증가 프로그램은 임베디드 시스템 환경에서 동작하기에 상대적으로 큰 프로그램으로 구동

하는데 오랜 시간이 소모되는 데, 기존 시스템과 SOONN에서 이 프로그램을 실행시킨 상태에서 종료 및 복귀 시간의 차이를 살펴봄으로써 SOONN의 잠재적인 이득을 확인해 보고자 한다.

실험에서 사용된 비교 대상으로써 SDRAM을 사용한 전형적인 기존의 종료 및 부팅 메커니즘, 최대 절전 모드로 알려진 STD 기법의 저장매체로 NAND 플래시 메모리를 사용한 종료 및 복귀, 그리고 STD 기법의 저장매체로 SCM을 사용한 종료 및 복귀 시스템을 고려한다. 논의의 편의를 위해서 이들 비교대상을 각각 순서대로 Normal, NAND-STD, 그리고 SCM-STD로 지칭하고 본 연구의 SOONN 프로토타입 시스템은 SOONN으로 명명한다.

#### 5.2 실험 결과

SOONN을 비롯한 비교 대상 시스템들이 복귀 과정과 종료 과정을 수행하는데 소요하는 시간이 각각 그림 3과 그림 4에 나타나 있다. 각 그래프에서 x축은 평가 대상 시스템들이고 y축은 복귀 및 종료 과정을 수행하는데 소요된 시간을 초단위로 표현하고 있다. 또한 각 그래프에서 검은 색(Kernel-only) 막대로 표현된 값은 커널 프로세스만 동작하고 있는 상태, 그리고 빗금 쳐진 (Kernel + User program) 막대로 표현된 값은 커널과 사용자 프로세스(숫자 증가 프로그램)가 동시에 동작하고 있는 상태에서 시스템 복귀와 종료를 의미한다.

그림 3과 4에 보여지는 바와 같이 복귀와 종료 과정에서 본 연구에서 개발된 SOONN은 1초 이내로 가장 빠르게 복귀와 종료를 완료한다. 각 그래프는 사용자 프로그램을 실행시킨 상태가 SOONN의 복귀와 중단에 소요되는 시간에 크게 영향을 끼치지 않음을 보여준다. 이는 SOONN의 복귀 및 중단이 이루어질 때 각각 비휘발성 메인 메모리로부터 읽거나 메인 메모리에 저장되어야 하는 휘발성 정보들이 사용자 프로그램의 동작 유

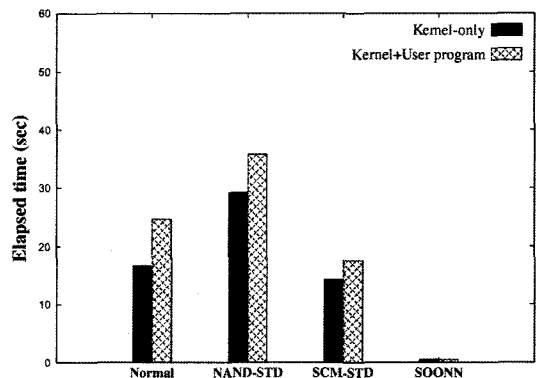


그림 3 복귀 시간 비교 그래프

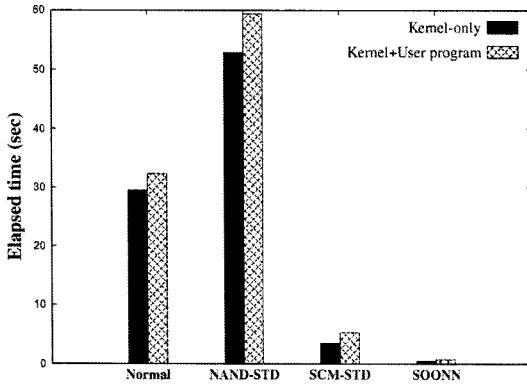


그림 4 종료 시간 비교 그래프

무에 상관없이 적은 용량의 프로세서와 주변 장치의 정보에 국한되기 때문이다. 따라서 메인 메모리에 상주되는 사용자 프로그램의 존재 유무는 복귀와 종료 과정에 영향을 미치지 않는다. 이에 반해서 SOONN을 제외한 비교 대상 시스템들은 커널 프로세스만 실행된 상태의 복귀와 중단보다 사용자 프로그램을 실행시킨 상태에서 더 많은 시간을 소비한다. Normal의 경우에는 부팅 이후에 사용자 프로그램을 실행시키는 시간이 포함되어 복귀 시간이 더 늘어나며 종료 과정에서는 사용자 프로그램을 종료시키는 시간이 추가되어 커널 프로세스만 실행된 상태보다 중단을 처리하는 작업에 더 많은 시간이 필요하게 된다. 나머지 시스템은 둘 다 STD 기법을 사용하기 때문에 메인 메모리로 복귀할 데이터의 크기가 사용자 프로그램으로 인해 더 늘어난다.

본 실험 결과에서 비교 대상 시스템들 간의 성능 비교 결과도 흥미롭다. 먼저 STD 기법을 사용하는 두 시스템들의 경우를 살펴보면, 빠른 저장장치를 사용한 SCM-STD가 NAND-STD보다 복귀와 중단이 더 빠르게 진행된다. 특히 종료 과정에서는 NAND 플래시 메모리가 읽기보다 쓰기에 더 많은 연산 시간이 필요하기 때문에 SCM을 사용한 SCM-STD보다 더 약화된 성능을 보여 주게 된다. 이어서 Normal과 SCM-STD간의 복귀 및 종료 과정을 비교해 보면, 복귀할 때는 비슷한 시간이 걸리지만 중단할 때는 SCM-STD가 더 좋은 성능을 보이고 있다. 그 이유는 STD의 매커니즘에 기인한다.

Normal의 복귀 과정은 하드웨어 초기화, 커널 초기화, 파일 시스템 인식, 초기 프로세스 생성으로 크게 네 단계로 나타낼 수 있다. STD가 Normal과 다른 점은 파일 시스템 인식 후에 초기 프로세스를 생성하지 않고, 비휘발성 저장 장치에 저장해 둔 데이터로 시스템 상태를 복구한다는 점에 있다. 비록 SCM-STD의 시스템 상태 복구가 Normal의 프로세스 생성 단계보다 빠르게

이루어질 수 있지만 SCM-STD 역시, 복귀 과정의 처음 세 단계(하드웨어 초기화, 커널 초기화, 파일시스템 인식)가 선행되어야 하기 때문에 복귀 과정은 Normal과 큰 차이가 없게 된다. 하지만 중단의 경우 Normal이 프로세스를 종료시키고 자원 할당을 해제하는 데에 반해 STD에서는 단순히 프로세스를 멈춘 후, 시스템 상태를 비휘발성 저장 장치에 저장하는 과정만 필요하기 때문에 SCM-STD가 Normal보다 더 좋은 성능을 보여 주게 된다.

## 6. 결론

본 논문에서는 SCM을 메인 메모리로 대체하여 종전의 상태를 즉각적으로 복귀하는 SOONN에 대해서 논의하였다. SOONN의 실현 가능성을 살펴보기 위해서 프로토타입 시스템을 실제 임베디드 시스템 환경에서 개발하고 그 성능을 평가하였다. 성능 평가 결과, SOONN은 1초 이내로 복귀와 종료 과정이 즉각적으로 완료되는 것을 확인하였다.

향후 연구 과제로는 본 연구에서 개발된 프로토타입 시스템을 보완하고 실제 시스템 환경에서 적용하는 연구들이 진행될 필요가 있다. 먼저 프로토타입 시스템의 완성도를 높이기 위한 연구가 선행되어야 한다. 특히, SOONN은 시스템 상태가 초기화되지 않고 지속적으로 유지되는 특성을 가지기 때문에 기존의 시스템에 비해서 갑작스런 시스템 붕괴 상황에 더욱 취약하게 된다. 따라서 향후 연구 과제으로써 SCM을 활용하여 시스템의 안정성을 극대화하는 연구의 수행이 절실하다.

## 참고 문헌

- [1] R. F. Freitas and W. W. Wilcke, "Storage-class memory: The next storage system technology," *IBM Journal of Research and Development*, vol.52, no.4/5, pp.439-447, 2008.
- [2] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. G., and R. S. Shenoy, "Overview of candidate device technologies for storage-class memory," *IBM Journal of Research and Development*, vol.52, no.4/5, pp.449-464, 2008.
- [3] E. L. Miller, S. A. Brandt, and D. D. E. Long, "HeRMES: High-Performance Reliable MRAM-Enabled Storage," In *Proceedings of HOTOS-VIII*, pp.95-99, 2001.
- [4] M. Baker, S. Asami, E. Deprit, J. Ousterhout, and M. Seltzer, "Non-Volatile Memory for Fast, Reliable File Systems," In *Proceedings of ASPLOS-V*, pp.10-22, 1992.
- [5] M. Moshayedi and P. Wilkison, "Enterprise SSDs," *ACM Queue*, vol.6, no.4, pp.34-39, 2008.
- [6] Splashtop, "<http://www.splashtop.com/overview.php>"