

TC-SSD: 파일 버전을 위한 타임 캡슐 SSD

채지원¹, 안진우¹, 민동현¹, 이종희², 김영재¹

¹서강대학교 컴퓨터공학과, ²고려대학교 사이버국방학과

{jeewon9708, jinu37, mdh38112, youkim}@sogang.ac.kr, j_lee@korea.ac.kr

TC-SSD: Time Capsule SSD for File Version Control

Jeewon Chae¹, Jinwoo Ahn¹, Donghyun Min¹, Junghee Lee², Youngjae Kim¹

¹Dept. of Computer Science and Engineering, Sogang University, Seoul, South Korea

²School of Cybersecurity, Korea University, Seoul, Republic of Korea

요약

민감한 데이터에 대한 위협이 증가하고 있다. 본 논문에서는, 커널이 침해당한 환경에서도 민감한 파일을 선별적으로 백업하기 위한 파일 버전 관리 스토리지 시스템(TIME CAPSULE SSD)을 제안한다. TIME CAPSULE SSD는 SSD에서 스토리지의 모든 상태를 백업하는 Project Almanac을 확장하여, SSD에서 파일 별로 세밀하게 백업을 관리하는 시스템을 제공한다. 이를 위해, 첫째로 SSD에게 파일 시멘틱을 알려주기 위한 파일 보존 기간 핸들러를 구현한다. 둘째로, 보존기간이 만료된 파일을 효율적으로 수거 하기 위한 파일 백업 존을 구현한다. 우리는 리눅스 환경 및 Jasmine OpenSSD 플랫폼을 활용하여 TIME CAPSULE SSD의 프로토타입을 제작하였다. 실험에서는 Time Capsule SSD가 20%의 파일을 선별적으로 백업 할 경우에, Almanac과 같이 100%를 백업하는 경우보다 1.71배 향상된 쓰기 성능을 보였다.

1 서론

데이터의 중요성이 커짐에 따라, 데이터를 변조하기 위한 다양한 위협이 증가하고 있다 [1, 2]. 대표적으로 랜섬웨어 [1]는 사용자의 파일을 암호화하고 복호화를 위한 금전을 요구하는 멀웨어이다. 또한, 해킹을 통해 무차별적으로 데이터를 삭제하거나 [2], 기업 내부자의 실수 또는 고의로 데이터가 손실될 위험이 있다. 이로 인해, 법적으로 일정기간 보존해야 하는 컴플라이언스 파일 데이터(진료 기록, 개인 정보 등)나, 조직의 핵심 디지털 자산(계약서, 설계데이터, 소스코드, 중요 문서 파일 등)이 변조되면 피해는 갈잡을 수 없이 커진다.

데이터 변조 공격에 방어하기 위해서 애플리케이션 또는 운영체제 레벨에서 소프트웨어 기반의 백업 솔루션이 연구되었다 [3]. 하지만, 강력한 멀웨어가 침입하여 커널 권한을 획득한다면, 소프트웨어 기반의 솔루션은 더 이상 안전을 보장하지 못한다. 예를 들어, Shamoon2 멀웨어 [2]는 사용자 루트킷 권한을 탈취하고, 커널 드라이버를 이용하여 OS의 파일 시스템이 제공하는 보안을 우회하고 파일을 파괴한다.

이를 방어하기 위해, Project Almanac [4]은 SSD 펌웨어 위에서 스토리지의 모든 상태를 백업 및 복구하는 시스템을 제공한다. Almanac은 다음과 같은 이유로 SSD 펌웨어를 백업하는 곳으로 선정한다. 첫째로, SSD는 out-of-place 업데이트를 수행한다. 따라서, 이전 버전의 데이터는 Garbage Collection이 수행되기 전까지는 안전하게 보존된다. 둘째로, SSD 펌웨어는 애플리케이션이나 운영체제로부터 고립되며, 상대적으로 매우 작은 Trusted Computing Base(TCB)를 가진다. 사용자는 Almanac이 제공하는 API를 이용하여 스토리지 상태의 최소 백업 보존 기간을 설정한다. 뿐만 아니라, 사용자는 API를 사용하여 특정 시점으로 복구를 수행한다.

하지만 Almanac은 몇가지 문제점을 가진다. 첫째로 Almanac은 중요한 파일을 장기간 보존하기에는 한계가 있다. Almanac은 모든 스토리지 상태의 보존 기간을 동일하게 설정하기 때문이다. 예를 들어, 법적으로 5년간 저장해야 하는 컴플라이언스 데이터보다 Almanac의 백업 보존 기간이 짧으면, 예기치 않게 백업본이 삭제 될 수 있다. 이를 막기 위해, 백업 보존 기간을 늘려서 5년으로 설정하면 공간상의 오버헤드가 폭발적으로 증가한다. 단일 파일을 보호하기 위해 모든 스토리지 상태를 5년간 보관해야 하기 때문이다. 둘째로 많은 양의 백업본을 보존으로 SSD의 성능이 저하된다. SSD는 용량이 찰수록 GC로 인한 성능 오버헤드가 증가하기 때문이다.

이러한 문제를 해결하기 위해, 본 연구는 SSD에서 선별적으로 중요한 파일 백업과 복구를 지원하는 TIME CAPSULE SSD를 제안한다. TIME CAPSULE SSD는 컴플라이언스나 조직의 디지털 자산과 같은 중요한 파일을 선별적으로 백업한다. 여기서 사용자는 TIME CAPSULE SSD가 제공하는 API를 통해서 파일의 보존 기간을 설정할 수 있다. SSD는 사용자가 선정한 보존기간동안 파일을 백업하다가, 보존기간이 끝나면 데이터를 자동적으로 삭제한다. TIME CAPSULE SSD는 중요한 파일만 선별적으로 백업하기 때문에, SSD의 백업량 및 성능 오버헤드가 감소한다.

하지만 스토리지는 파일 시멘틱을 모르기 때문에, 스토리지에서 파일 별로 백업을 관리하는 것은 쉬운 문제가 아니다. 이를 해결하기 위해, TIME CAPSULE SSD는 운영체제 단에서 SSD에게 파일 시멘틱을 알려주기 위한 파일 보존 기간 핸들러를 구현한다. 사용자가 파일 보존 기간을 설정하면, 파일 보존 기간 핸들러는 파일 시스템이 SSD에 페이지쓰기를 수행할 때, 보존 기간을 packing하여 함께 전송한다. 이를 통해 SSD는 파일 보존 기간별로 페이지를 관리하는 백업 메커니즘을 수행한다. 둘째로, TIME CAPSULE SSD는 SSD에서 보존 기간이 만료된 파일을 효율적으로 수거하는 메커니즘을 제시한다. SSD에서 파일 보존기간이 같은 페이지는 서로 같은 페이지 수명을 가진다. 이를 활용하기 위해, TIME CAPSULE SSD는 파일의 보존 기간에 따라 페이지를 분리하여 파일 백업 존(file backup zone)에 저장한다. 백업 존에서 파일의 보존 기간이 끝난 블록은 GC 수행 때 일괄적으로 수거된다. 수거될 때에는 추가적인 페이지 복사가 발생하지 않으므로, 성능 오버헤드가 거의 발생하지 않는다.

우리는 우리의 시스템을 증명하기 위해 리눅스 환경에서 파일 보존 기간 핸들러를 구현하고, Jasmine OpenSSD [5] 플랫폼 내부에 TIME CAPSULE SSD를 개발하여 프로토타입으로 제작하였다. 그리고, 파일을 100% 백업하는 Almanac과 비교하기 위해, 백업 파일의 비율에 따른 성능 실험을 수행하였다. 실험 결과, 파일을 모두 백업하는 정책을 가진 시스템(Almanac)에 비해, 20%만 백업할 경우 1.71배 향상된 쓰기 성능을 보였다.

2 배경

NAND 플래시 메모리 기반의 Solid-State Drives (SSDs)는 저장장치의 하나로서, 내장 CPU를 활용한 디바이스 레벨 계산 능력, 하드 디스크 보다 높은 성능 덕분에 널리 사용되고 있다. SSD는 다음의 세 가지 주요한 컴포넌트를 갖는다: (1) Flash Translation Layer (FTL)의 주소 변환 테이블,

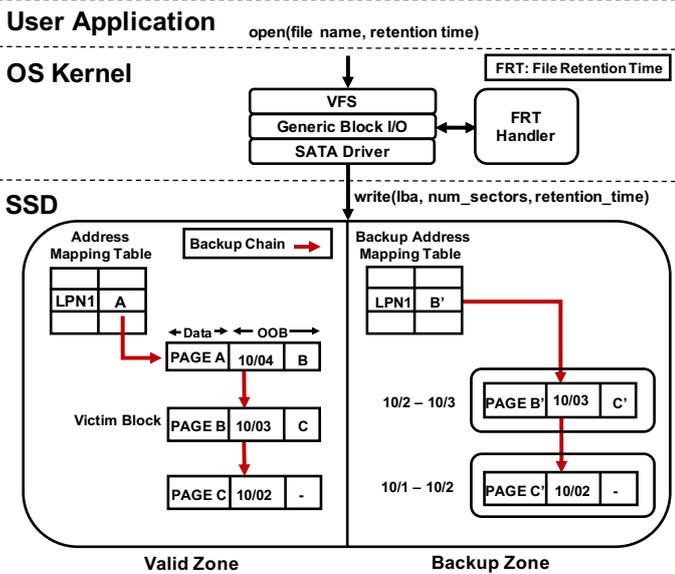


그림 1: Time Capsule SSD (TC-SSD) 개요

(2) Out-of-Band (OOB) 영역 그리고 (3) Garbage Collection (GC).

FTL 은 데이터가 저장된 플래시 메모리의 물리 주소 (PPN) 와 호스트가 볼 수 있는 논리 주소 (LPN) 사이의 변환 역할을 한다. SSD 에서는 페이지 (page) 를 기본 단위로 하여 읽기와 쓰기 연산이 수행된다. SSD 는 플래시 메모리 설계 때문에 한 번 쓰기가 수행된 Valid 상태의 페이지에 제 자리 덮어쓰기 (In-place update) 를 허용하지 않는다. 대신 다른 자리 덮어쓰기 (Out-of-place update) 를 수행한다. 따라서 특정 LPN 의 페이지에 덮어 쓰기 연산이 들어온다면, SSD 는 다른 Free 상태의 페이지에 새로운 내용을 쓰고 새로운 페이지의 PPN 과 원래의 LPN 을 연결시킨다. 그리고 이전 PPN 의 페이지는 Invalid 상태가 된다. 새로운 LPN-PPN 연결 정보는 FTL 에서 맵핑 테이블 (mapping table) 형태로 기록된다.

다음으로, 페이지에 대한 메타 데이터는 OOB 공간에 저장된다. OOB 공간은 물리 페이지의 일부 공간을 빌려 할당할 수 있다. 기존의 OOB 공간에는 해당 물리 페이지와 연결되어 있는 LPN, 페이지 쓰기 연산이 수행된 시간 등이 저장되어 있어 SSD 전원 실패에 대비할 수 있다.

마지막으로, SSD 가 다음 쓰기 연산을 수행할 페이지가 부족해진 경우에는 GC 작업을 시행하여 다시 쓸 수 있는 페이지를 만든다. SSD 는 GC 과정을 통해 여러 페이지들로 이루어진 블록을 단위로 하는 지우기 연산을 수행하여 여러 새 페이지들을 확보할 수 있다. 이러한 지우기 연산을 때 GC 마다 빅티엄 (victim) 블록을 선정하고 이 빅티엄 블록을 지워서 다음 쓰기 연산에 대비한다. 한 가지 GC 의 큰 특징은 빅티엄 블록 내의 Valid 페이지는 다른 Free 상태의 물리 페이지에 옮겨 보관되고 Invalid 페이지는 그대로 남겨진다. 따라서 GC 수행 시, Valid 페이지를 재배치하는 오버헤드가 생긴다. 따라서 이 오버헤드를 줄이기 위해 기존의 여러 GC 는 Valid 페이지 개수가 적은 블록을 빅티엄으로 선정하는 정책을 갖는다.

3 설계

3.1 TIME CAPSULE SSD개요

본 섹션에서 우리는 SSD 레벨에서 파일 별로 다양한 보존 기간을 설정하여 선별적인 백업과 복구를 지원하는 TIME CAPSULE SSD를 설명한다. TIME CAPSULE SSD는 다음과 같은 세 가지 핵심 구성 요소로 구성된다: (1) 파일 보존기간 핸들러, (2) 백업 체인 (Backup chain), (3) 백업 존 (Backup zone). 파일 보존기간 핸들러는 파일시스템에서 파일 별 보존 기간을 관리하고 SSD 로 보내는 소프트웨어이다. 이 핸들러는 응용 단

로부터 파일 별 보존 기간 정보를 받아와서 SSD 로 보내주어 스토리지 레벨의 파일 별 보존 기간 설정을 가능하게 만든다. TIME CAPSULE SSD는 보존 기간이 정해진 특정 논리 페이지에 덮어쓰기 요청이 들어오게 되면 이전 물리 페이지의 상태를 Invalid 로 바꾸는 대신 Backup 으로 바꾸어 보존 기간 만큼의 유통 기한을 보장한다. 이를 위해, TIME CAPSULE SSD는 SSD 내부에 논리 페이지 (LPN) 당 백업 체인을 만들어 백업 페이지 물리 주소를 연결한다. TIME CAPSULE SSD는 백업 페이지도 유효한 상태로 인식하므로 GC 과정 동안 페이지 복사가 일어난다. 따라서 백업 페이지 주소가 바뀌면서 백업 체인이 끊어질 수 있다. TIME CAPSULE SSD는 GC 에 의해 끊어지는 백업 페이지와 그 이전 백업 페이지 모두를 백업 존이라고 하는 다른 플래시 메모리로 옮겨 새로운 체인 연결을 유지한다. 이 백업 존에 있는 페이지는 GC 의 빅티엄에 선정되지 않기 때문에 파일의 요구되는 보관 기간을 보장할 수 있다.

3.2 파일 보존 기간 핸들러

호스트의 운영체제에서 구현되는 파일 보존 기간 핸들러의 기능은 다음과 같다. 첫째, 사용자에게 파일의 보존 기간을 설정하기 위한 API를 제공한다. 사용자는 API를 통해 파일을 최초 생성할 때 파일의 보존 기간을 파라미터로 파일 보존 핸들러에게 전송한다. 이 때, 파일 보존 핸들러는 보존 기간을 파일 시스템의 아이노드 구조체에 영구적으로 저장한다. 둘째, 파일 보존 기간 핸들러는 SSD에게 블록 단위의 보존 기간을 알려주는 역할을 수행한다. 사용자가 파일 쓰기를 수행하면, 파일 시스템은 쓰기를 수행할 논리 블록 주소(LBA)를 선정한다. 그리고 SATA 드라이버는 파일시스템으로부터 받은 LBA, size, 데이터 버퍼를 포장 (packing) 하여 SSD 에 쓰기 명령을 전송한다. 이 때, 파일 보존 핸들러는 아이노드로부터 보존 기간을 받아온다. 그리고, SATA 드라이버가 전송하는 파라미터의 사용하지 않는 공간 (Reserved space)에 보존 기간을 포함하여 전송한다 [6]. 따라서 SSD 는 새롭게 쓰는 페이지의 보존 기간을 호스트로부터 함께 받아오게 된다.

3.3 백업 체인

TIME CAPSULE SSD는 파일에 해당하는 페이지의 백업을 보관하는 것을 목표로 한다. 이를 위해, TIME CAPSULE SSD는 백업 체인 (Backup chain) 을 만들어 하나의 논리 페이지에 대한 백업 체인을 만든다. 각 플래시 페이지는 내부적인 메타데이터를 위해 사용되는 Out-of-Band (OOB) 공간을 갖는다. TIME CAPSULE SSD는 SSD 내부 DRAM 절약과 SSD 절전 대비를 위해 OOB 공간을 활용하여 다음과 같은 정보를 저장한다:(1) 해당 플래시 메모리 주소와 대응되는 LPN 정보, (2) Backup 페이지 물리 주소를 가리키는 포인터 정보, (3) 쓰기 타임 스탬프 (WT), (4) 보존 기간 (RT). TIME CAPSULE SSD에 보존 기간을 보장해야 하는 페이지에 대해 덮어쓰기가 들어오면, Valid 상태 페이지는 Backup 상태 페이지가 되고, 새로운 Valid 페이지의 포인터에 백업 페이지의 물리 주소를 기록하여 체인을 만든다. 특히, 동일한 논리 페이지에 여러 번의 덮어쓰기가 요청되면 여러 백업 페이지가 포인터로 연결되어 있어 여러 버전이 유지된다.

TIME CAPSULE SSD의 백업 체인은 Valid, Invalid 그리고 Backup 상태의 페이지가 혼재되어 있다. 가장 최신 버전의 데이터를 의미하는 Valid 페이지들은 GC 수행 시, 데이터 내용을 지우지 않고 다른 Free 상태의 페이지로 복사된다. 반면, Backup 상태 페이지는 GC 시, 정해진 백업 유통 기한에 따라 두 가지 처리가 가능하다. 보관 기간이 아직 지나지 않은 백업 페이지는 GC 수행 시, Valid 상태 페이지와 마찬가지로 페이지 복사를 수행한다. 반면 보관 기간이 이미 지난 페이지는 GC 수행 시 바로 지워진다. TIME CAPSULE SSD는 한 백업 페이지의 유통 기한을 계산하기 위해, 식 1 을 사용한다.

$$Compare(CurrentTime, BackupTime + BackupRT) \quad (1)$$

식 1 에서 $CurrentTime$ 는 SSD 에서 내부 타이머의 현재 시간, $BackupTime$ 은 페이지가 Backup 상태가 된 시간, $BackupRT$ 는 백업 페이지의 보존 기간을 의미한다. TIME CAPSULE SSD는 $CurrentTime$ 가 $BackupTime + BackupRT$ 의 값 보다 같거나 크면 해당 Backup 페이지의 유통 기한이 지난 것으로 판단한다.

보관 기간이 지나지 않은 백업 페이지는 GC 수행 시, 페이지 복사를 수행하고 해당 백업 페이지는 Invalid 상태가 된다. 따라서 백업 페이지 물리 주소가 바뀐다. 만약 백업 체인 중간의 백업 페이지가 GC 의 빅팀에 포함 되는 경우에는 체인이 끊기는 문제가 발생한다. 그림 1에서 백업 체인을 보면 GC 수행시 선정된 빅팀 블록에 Page B가 존재한다. 이 때 이를 다른 free 페이지에 옮길 경우 데이터 체인이 끊어지게 된다. 따라서 백업 체인의 올바른 연결 유지를 위해 TIME CAPSULE SSD는 백업존을 제안한다.

3.4 백업 존

LPN 당 유지되는 백업 체인이 GC 수행에 의해 끊어지는 것을 방지하기 위해 TIME CAPSULE SSD는 플래시 메모리를 백업 존 (Backup Zone) 과 유효 존 (Valid Zone) 을 구분한다. 백업 존은 특정 LPN 의 백업 체인에 존재하는 백업 페이지 중 GC 빅팀이었던 페이지부터 백업 체인 끝까지의 모든 백업 페이지들이 복사 및 저장되는 플래시 메모리 공간이다. 백업 존에 있는 모든 백업 페이지들은 각 고유 유통 기한에 도달할 때 까지 GC 에 의해 절대 사라지지 않는다. 반면, 유효 존은 GC 수행이 아직 이루어지지 않은 그 외 백업 페이지들의 체인이 존재하는 공간이다.

예를 들어, 그림 1 에서 Data Page B 는 유효 존의 백업 체인에 존재한다. TIME CAPSULE SSD는 Data page B 가 GC 빅팀으로 선정이 되면 그 체인의 끝인 Data page C 까지 모든 백업 페이지를 전부 백업 존으로 옮긴다. 따라서 하나의 백업 체인은 유효 존과 백업 존으로 서브 백업 체인 (Sub-Backup chain) 으로 나누어진다. TIME CAPSULE SSD는 FTL 의 주소 변환 테이블 (mapping table) 을 통해 유효 존에 있는 서브 백업 체인을 접근할 수 있다. 마찬가지로 백업 존에 있는 서브 백업 체인을 접근하기 위해서, TIME CAPSULE SSD는 백업 주소 변환 테이블 (Backup Address Mapping Table) 을 DRAM 에 두어 체인 연결 정보를 유지한다. 따라서 두 서브 체인에 접근하여 돌아가려는 데이터 버전으로 복구가 가능하다.

TIME CAPSULE SSD는 백업 체인을 백업 존으로 옮기는 과정에서 유사한 유통 기한 시기를 갖는 백업 페이지들을 하나의 백업 블록에 모은다. 우리는 하루 단위로 유통 기한 시기를 분류하였지만 이는 조정 가능한 값이다. 한 백업 블록에는 유통기한이 유사한 페이지들이 있기 때문에 유통 기한이 다하여 삭제 되는 경우에 추가적인 페이지 복사가 필요하지 않다. 그림 1 에 나타난 Page B' 와 Page C' 는 유통 기한이 서로 다르므로 다른 백업 블록에 배치된 모습을 보인다. 만약 10/2 이 지난다면 Page C' 가 있는 백업 블록은 삭제될 수 있다.

4 실험

우리는 실험을 위해 Intel i7-8700 CPU와 16GB RAM을 제공하는 컴퓨터를 이용하였다. 우리는 호스트 운영체제 (Linux 4.10.16)의 VFS와 블록 레이어, 그리고 드라이버 (SATA.2.0)코드를 일부 수정하여 파일 보존 기간 핸들러를 구현하였다. 그리고 우리는 Jasmine OpenSSD 플랫폼 (ARM7TDMI-S core, 96KB SRAM, 64MB DRAM, 64GB NAND Flash) 을 활용하여, SSD 펌웨어 단에서 TIME CAPSULE SSD를 프로토타입으로 제작하였다. 여기서 우리는 Jasmine에 선별적인 Flush를 구현하여 쓰기 최대 성능을 약 20MB/s에서 약 65MB/s로 개선하였다. OpenSSD는 64GB를 지원하지만, 우리는 GC으로 인한 성능 오버헤드를 측정하기 위해 2GB만 파티션으로 지정해 1GB는 유효 존, 1GB는 백업 존으로 할당하였다.

실험을 위해, 우리는 단일 스레드 기반의 종합적인 워크로드를 개발하였다. 워크로드는 다음과 같이 작동한다. 먼저, 워크로드는 200MB파일을 5

표 1: 백업 파일 비율에 따른 Throughput 비교 (MB/s)

0%	20%	40%	60%	80%	100%
66.26MB/s	42.89MB/s	34.07MB/s	30.14MB/s	28.41MB/s	24.98MB/s

개 생성하여, SSD로 최초 쓰기를 수행한다. 그리고, 각 파일에 대해 연속적인 덮어쓰기를 다시 수행하여, 성능을 측정한다. 덮어쓰기를 수행하는 시점에서 SSD의 유효존이 가득차기 때문에, GC가 유발된다. 우리는 파일의 일부를 민감한 파일로 선정하고, 보존 기간을 충분히 길게 설정하여 실험을 수행하였다.

Table 1은 전체 파일 대비 백업하는 파일의 비율에 따른 성능을 비교한 실험 결과를 보여준다. 0%는 민감한 파일이 없는 상태로, 백업을 제공하지 않는 보통의 SSD와 같은 효과를 보인다. 반면 100%는 모든 파일에 보존 기간을 설정한 상태로, 사실상 스토리지의 모든 상태를 저장하는 Almanac과 같은 효과를 보인다. 그 외 20%부터 80%까지는 파일을 선별적으로 백업하는 TIME CAPSULE SSD를 나타낸다. 백업 비율이 0%일때의 성능은 기존 OpenSSD(65MB/s)와 거의 동등한 성능이 나타났다. 하지만, 백업 비율이 증가함에 따라, 쓰기 성능은 감소하였다. 이는 SSD의 백업량이 늘어남에 따라, GC로 인한 페이지 카피량이 증가하기 때문인 것으로 보인다. 특히, 백업 비율이 100%에 다다르면, 성능이 가장 낮아져서, 0%일 때보다 약 66% 하락하는 결과를 나타냈다.

5 결론

본 논문에서는 SSD에서 선별적으로 중요한 파일 백업과 복구를 지원하는 TIME CAPSULE SSD를 제안한다. TIME CAPSULE SSD는 SSD에서 컴플라이언스나 핵심 디지털 자산과 같은 중요 파일의 백업 보존 기간을 설정하여, 보존 기간 내에서 복구를 보장한다. 우리는 SSD에게 파일 시멘틱을 알려주기 위하여 호스트 단에서 파일 보존 기간 핸들러를 개발하였다. 또한, SSD에서 파일 별 백업을 관리하고, 효율적으로 만료된 파일을 수거하기 위하여 파일 백업 존을 구현하였다. 우리는 우리의 시스템을 증명하기 위해, TIME CAPSULE SSD프로토타입을 제작하였다. 실험 결과 TIME CAPSULE SSD는 20%의 파일을 백업 할 경우에, 모든 상태를 백업하는 정책을 가진 시스템(Almanac)보다 1.71배 향상된 쓰기 성능을 보였다.

6 사사문구

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학 지원사업의 연구결과로 수행되었음(2015-0-00910)

참고 문헌

- [1] C. SAM, "2017-2019 ransomware statistics and facts." <https://www.comparitech.com/antivirus/ransomware-statistics/>, 2018.
- [2] D. Jain, "Shamoon 2: Back on the prowl!" <https://nsfocusglobal.com/shamoon-2-back-on-the-prowl/>, 2017.
- [3] A. Continella, A. Guagnelli, G. Zingaro, G. De Pasquale, A. Barengi, S. Zanero, and F. Maggi, "ShieldFS: a self-healing, ransomware-aware filesystem," in *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC)*, 2016.
- [4] X. Wang, Y. Yuan, Y. Zhou, C. C. Coats, and J. Huang, "Project almanac: A time-traveling solid-state drive," in *Proceedings of the Fourteenth EuroSys Conference 2019*, p. 13, ACM, 2019.
- [5] S. P. Lim, "The jasmine openssd platform: Technical reference manual (v1.4, in english)," 2016.
- [6] D. Anderson, "Sata storage technology," 2007.