

센서 노드를 위한 플래시 메모리 저장 시스템에 대한 고찰

송준영^o 이기혁 한형진 최원철 한경훈 한지연 손기락

한국 외국어 대학교 컴퓨터 및 정보통신 공학부

{ slyking^o, knockin81, nom96ny, wc.choi, myway767, hanjiyeon, ksohn}@hufs.ac.kr

Survey on Flash Memory based Storage System for Sensor Nodes

Jun-young Song^o, Ki-hyuk Lee, Hyung-jin Han, Won-chul Choi, Kyoung-hoon Han,

Ji-yeon Han, Kirack Sohn

Hankuk University of Foreign Studies

요 약

본 논문에서는 현재 많은 분야에 적용되어 데이터 수집을 위해 사용되는 센서 노드들의 데이터 저장에
위한 플래시 메모리를 기반으로 한 저장 시스템들을 비교해 보고 각각의 특징과 취약점을 비교 분석한
다. 그리고 대용량 플래시 메모리 기반 저장시스템을 위한 개발 방안에 대한 앞으로 나아갈 방향을 제시
한다.

1. 서 론

현재 주위 환경에 대한 온도, 습도, 조도, 중력, 자기 등의 데이터를 실시간으로 감지하고, 관리할 수 있게 해 주는 센서들이 많은 분야에 적용되고 있다. 그 센서들의 부착한 센서 노드들의 크기가 작아지고 생산 비용 또한 감소하고 있는 추세이다.

이렇게 많은 분야에서 사용되고 있는 센서 노드들의 가장 큰 제약은 소모되는 전력이다. 타 기계에 부착되어 사용된다면 전력 문제에 크게 영향을 받지 않겠지만 데이터 수집을 목적으로 하기 때문에 보통 단독으로 설치 사용된다.

센서 노드의 가장 큰 전력 사용은 측정된 데이터를 전송할 때 발생한다. 그 소모량이 측정된 데이터를 저장할 때 소모되는 에너지 보다 100배에 이른다. 측정된 데이터를 전송하는 것보다 저장하는 것이 에너지 소모의 측면과 데이터의 안정성을 위해 더욱 효율적인 것이다.

현재 센서 노드의 데이터 저장을 위해서는 센서 노드 내부에 설치되어 있는 ROM을 사용한다. 하지만 ROM의 용량의 크기가 한계가 있다. 그래서 현재 새로운 저장매체로 플래시 메모리(flash memory)가 각광받고 있는 추세이다. 플래시 메모리는 전력공급이 없어도 안전한 비휘발성, 빠른 접근속도, 전력의 소모가 작고 크기가 작다는 장점을 가지고 있다. 플래시 메모리는 낸드(NAND) 플래시와 노어(NOR) 플래시로 두 가지의 종류가 있다 [1].

낸드(NAND) 플래시는 디스크와 같이 일정 크기의 섹터 단위로 어드레싱한다. 이런 이유로 플래시 메모리 상에서 직접 프로그램을 실행하지 못하고 DRAM으로 옮겨서 실행해야 한다. 그러나 한 비트를 저장하기 위한 셀의 크기가 작고 특정한 블록에 문제가 있더라도 해당 블록을 배드 블록으로 설정한 후 사용하면 되므로 수율이 높아싼 가격으로 고집적화를 할 수 있어 용량에 있어 대용량으로 만들기가 쉬운 장점을 가지고 있다.

반면 노어(NOR) 플래시는 저장된 데이터를 읽어올 때 바이트 단위로 어드레싱이 가능하다. 그래서 노어(NOR) 플래시 메모리는 프로그램을 적재하여 플래시 메모리 상에서 실행할 수 있는 특징이 있어 휴대전화에

본 연구는 21 세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크원천기술개발사업의 지원에 의한 것임

많이 사용되고 있다. 하지만 모든 소자를 어드레싱해야 하기 때문에 수율이 낮아 비트당 가격이 비싼 단점을 지니고 있다. 최근 낸드(NAND) 플래시 메모리는 단가가 낮아지고 대용량화 되고 있다는 이유로 단순 데이터 저장용을 위해서 센서 노드용 저장시스템으로 적용되고 있다.

하지만 현존하는 센서 노드용 저장시스템의 대부분이 노어(NOR) 플래시 메모리용으로 개발되어 있어 대용량의 낸드 플래시용으로 적용할 수 있는 시스템이 거의 없는 것이 문제점이다.

본 논문에서는 플래시 메모리를 이용한 저장 시스템을 비교하고 대용량의 센서 노드용 저장시스템을 위한 앞으로의 방향에 대하여 제시한다.

앞으로 많은 분야에 센서들이 적용될 것이다. 하지만 지금까지 센서 노드용 플래시 기반 저장 시스템에 대한 고찰이 없었다. 본 논문에서는 현재까지 보고된 네 가지 센서 노드용 플래시 기반 저장 시스템을 비교 분석하고자 한다.

2. NAND 플래시와 NOR 플래시

플래시 메모리용 저장 시스템의 기본이 되는 플래시 메모리는 ROM처럼 전원이 제거되어도 정보를 그대로 유지하는 비휘발성 기억 장치로, 메모리 셀들의 한 부분이 선평(flash)처럼 단 한번의 동작으로 지워질 수 있도록 마이크로 칩이 조직화되었기 때문에 플래시 메모리라 부르게 되었다. 플래시 메모리는 EPROM과 EEPROM의 두 기술을 기초로 제작되었는데 EPROM의 단순한 회로와 EEPROM의 전기적 소거 방식을 채택하였다.

하드디스크와 같은 보조기억장치보다 실행속도가 훨씬 빠르고 전력소모가 매우 적고, 물리적 충격에 강해 노트북과 PC 등 각종 휴대용 단말기에 적합한 기록매체로 각광 받고 있다.

하지만 RAM 이나 하드디스크와 같은 기억장치가 데이터를 읽고 쓰는 횟수가 거의 무한대인 반면에 플래시 메모리는 그 횟수에 제한이 있다는 단점이 있다. 플래시 메모리는 처리속도가 빠른 NOR 플래시와 저장용량이 큰 NAND 플래시로 구분되는데 메모리를 만드는 기본 원리는 동일하지만 셀을 어떻게 구성했느냐에 따라 구분이 되어 진다.

2.1 NOR 플래시의 특징

코드 저장 형이라고 불리는 NOR 플래시는 셀이 병렬 구조로 어드레싱라인과 데이터라인으로 연결되어 있기 때문에 셀에 바로 접근할 수 있어서 데이터의 읽기 속도가 빠르다. 그리고 데이터안정성이 뛰어나다는 장점이 있지만 가격이 비싸다는 단점이 있다. 또한 데이터 및 어드레스 라인이 있어 회로가 복잡하고 차지하는 공간

이 넓기 때문에 이로 인해 반도체를 집적하기 힘들다는 단점이 있다.

플래시 상에서 프로그램 실행이 가능하여 hand phone, settop box등에 사용된다. 제조사는 인텔-샤프, AMD-후지쯔 등이 대표적이다. NOR 플래시는 병렬 구조이기에 순서에 관계없이 임의의 셀을 액세스 하는 Random Access가 가능하며 쓰기, 읽기 동작 시 내부의 어드레스를 인식하는 과정이 간단하여 컨트롤러의 제작이 간단하다.

2.2 NAND 플래시의 특징

데이터 저장 형이라 불리는 NAND 플래시는 셀이 직렬로 연결되어 있어 먼저 해당 블록으로 이동 후 직렬로 연결된 각 셀 가운데 자신이 필요로 하는 데이터에 접근하기 위해서는 다른 셀을 거쳐야만 하기 때문에 읽는 속도가 느리다는 단점이 있지만 가격이 싸고 대용량이 가능하다는 장점 때문에 많은 인기를 누리고 있다.

Read, write, erase, copyback의 4가지 주요기능이 있는데, Read는 플래시 메모리로부터 내용을 읽고, write는 플래시 메모리로 내용을 쓰는 것이다. 이 작업들은 모두 페이지 단위로 진행된다. Erase 기능은 read, write와 다르게 블록단위로 작업되며 플래시 메모리에 새롭게 써야 할 작업이 발생하면 이를 위해 메모리 내용을 제공한다. copyback 기능은 플래시 메모리 한 페이지에서 다른 페이지로 복사하는 기능이다.

그리고 NAND 플래시는 플래시상에서 프로그램 실행이 불가능하기 때문에 저장용으로 사용되는데 이러한 특징으로 단순 데이터 저장을 위해 SDcard, memory stick, 디지털 카메라, mp3에 사용된다. 제조사로는 삼성과 도시바가 대표적이다. 구조의 특성상 읽기 동작 시에 block 선택시간이 필요하고, random access 동작측면에서 크게 불리하기 때문에 단일 칩 형 품목 대신 card 형 제품에 주력하고 있다.

3. 플래시 메모리 저장 시스템

Flash Memory를 사용한 파일 시스템은 JFFS, YAFFS 등 여러 가지가 존재한다. 본 논문에서는 센서 노드용이라는 장치적 제약을 전제하여 센서 노드용 Flash Memory 저장 시스템에 대하여 다루도록 한다.

센서 노드를 기반으로 하여 Flash Memory를 저장장치로 설계된 파일시스템은 그 수가 많지 않다. 대표적인 파일시스템은 TinyOS에서 기본적으로 제공하는 파일시스템인 Matchbox, ELF 그리고 Capsule이 있다. 이 절에서는 각 파일시스템의 특징을 비교하고 설명한다.

3.1 플래시 메모리 저장 시스템의 비교

아래 표 1은 Matchbox, ELF, Capsule 그리고

MicroHash을 비교한 것이다[2,3,4,5]. 저장소로 사용하는 장치를 비교하면 Matchbox와 ELF의 경우 NOR Flash를 사용하고 Capsule은 NOR, NAND 모두에 동작하도록 설계 되었고, MicroHash는 MMC에 기반으로 동작한다.

표 1 플래시 메모리 파일 시스템의 비교

	Matchbox	ELF	Capsule	MicroHash
Storage Device	NOR	NOR	NOR, NAND	MMC
Energy Optimized	X	X	O	O
Memory Optimized	O	O	O	X
Wear Leveling	X	O	O	O
Check pointing	X	X	O	X
Abstraction	Filesystem	Filesystem	Object	Stream/Index
Operation	Open Create Append Read Delete Read Delete	Open Create Append Read Delete Seek Rename Mkdir, rmdir	Open Create Append Read Delete Move	Initialization Growing Repartition Deletion
EEPROM에 저장하는 내용	Freelist	Metadata Index	X	X

에너지 효율적인 측면은 Capsule과 MicroHash만이 고려하고 있으며 메모리 효율적인 측면은 Matchbox와 ELF, Capsule이 고려하였다.

플래시 사용량의 균일성은 Matchbox를 제외한 두 파일시스템이 고려하고 있으며, Crash Recovery에 대한 정책은 ELF는 EEPROM에 기록하는 snapshot을 사용하고 Capsule은 checkpointing을 사용한다.

데이터의 추상화는 Capsule은 객체로 이루어지고 MicroHash는 Stream과 인덱스로 이루어지며 Matchbox와 ELF는 파일시스템으로 이루어진다.

Matchbox, ELF, Capsule은 모두 Open/Create, Write(Append), Read, Delete를 지원하며 ELF는 Write-Modify, seek, rename과 Mkdir, Rmdir과 같은 디렉토리 명령어도 지원하고 Capsule은 Move도 지원한다. MicroHash는 Initialization, Growing, Repartition, Deletion 작업을 지원한다.

EEPROM은 ELF가 메타데이터와 인덱스를 저장하는데 사용한다.

3.2 플래시 메모리 저장 시스템의 특징

3.2.1 Matchbox

TinyOS에서 기본적으로 장착된 파일 시스템이다. 동시에 여러 file을 열수 있고 file관리를 위한 원격 접근이 가능하다. 코드 크기는 10K 정도이며, 메모리 최소 사용량은 362byte 정도이나 작업하는 파일 수에 비례해 증가한다. Write-append 연산만을 지원하고 회복 중 파일의 무결성을 위해 CRC를 사용한다.

Free 페이지 list 관리를 위해 각 페이지당 1 비트를 할당하여 작성된 free list는 메인 메모리에 저장한다.

3.2.2 ELF

EEPROM과 NOR 플래시의 특성을 이용하여 Matchbox와 달리 Write-modify를 지원한다. 가능한 모든 write 연산에 대해 새로운 노드를 생성하지 않고 write-modify 연산에만 로그 적용하여 노드를 생성하고, 센서노드 환경을 위해 메모리 소비시간을 줄였으며 best-effort 신뢰성을 제공한다.

EEPROM과 RAM에 파일시스템이 가진 파일과 그 데이터를 가진 노드를 연결하는 추상화 영역을 사용하여 데이터에 대한 접근 속도를 향상 시켰다. 또한 seek과 디렉토리 명령어를 지원한다.

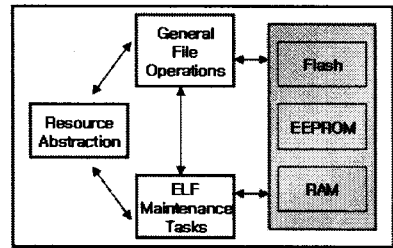


그림 1. ELF의 시스템 구조

그림 1은 ELF의 시스템 구조이다. 가장 위에 있는 "Resource Abstraction"은 run-time 메모리에 자리잡는다. 이는 open 된 파일에 대한 정보, cleaning 정책, 그리고 다른 ELF 설정을 저장한다. "General File Operations"는 파일이나 디렉토리 명령에 대한 논리적 추상영역이다. "ELF Maintenance Tasks"는 EEPROM에 있는 디렉토리과 파일 메타데이터의 snapshot의 관리 등 시스템 관리 Task의 추상화 영역이다. 이는 garbage collection task도 포함한다.

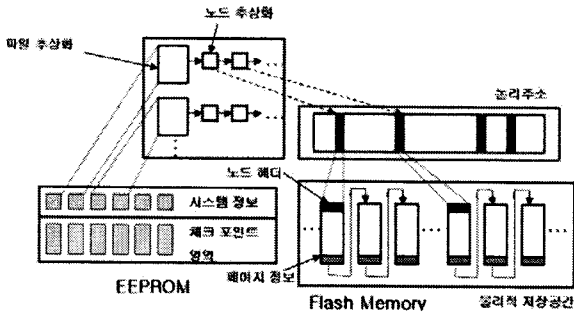


그림 2. ELF의 분산된 구조도

그림 2는 ELF의 EEPROM, RAM, Flash를 사용하여 분산된 구조도이다. 플래시 메모리에는 데이터가 담기는 physical 노드와 per-page 메타데이터가 저장된다. 메인 메모리에는 파일 추상화 영역과 노드 추상화 영역 그리고 파일 descriptor가 저장되며, EEPROM에는 빠른 액세스를 위하여 디렉토리 영역으로 사용하고 데이터의 snapshot을 저장하여 재실행 시 시스템을 복구한다.

3.2.3 Capsule

Capsule은 Matchbox나 ELF와 달리 NAND flash Memory에서도 사용 가능하게 설계되었다.

다양한 객체를 기반으로 하여 추상화를 수행하는 파일시스템으로 에너지, 메모리 절약적 디자인에 중점을 두었다.

Compaction을 지원하고 checkpointing을 사용한 failure 처리를 지원한다. Stack을 사용한 pointing으로 stack, queue 등의 다양한 자료 구조를 저장시스템 단위에서 지원한다.

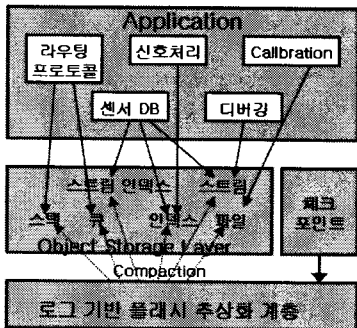


그림 3. Capsule의 객체 저장 구조

그림 3은 Capsule의 객체 저장 구조를 나타낸 그림이다. Capsule은 객체 저장 계층과 로그 기반 플래시 추상화 계층을 제공한다. Object Storage Layer로 다양한 목적으로 사용할 수 있는 스택, 큐, 인덱스 등을 지원한다.

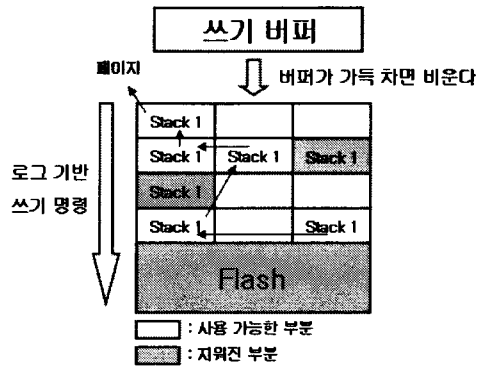


그림 4. 플래시 추상화 계층

플래시 추상화 계층은 그림 4와 같은 구조를 가지고 있다. 저장의 단위는 Chunk이며 log structure 기반으로 저장된다. 쓰기 횟수를 줄이기 위하여 쓰기 버퍼를 두고 버퍼가 가득 찰 때만 플래시 메모리에 기록한다.

또한 메모리 반환의 개념으로 garbage collection을 지원한다. Garbage collection은 플래시 추상화 계층의 몫이지만, 자동으로 수행하지 않고 지정한 임계 값만큼 플래시 메모리가 사용되고 compaction 명령이 상위에서 주어진 때 삭제를 수행한다.

3.2.4 MicroHash의 소개

MicroHash는 파일시스템보다는 인덱스에 의한 데이터 저장을 지원하는 저장 시스템이라 할 수 있다.

Timestamp나 value를 통한 플래시에 저장된 레코드로의 효율적 접근 방법을 제공한다. 일반적인 DB 인덱스처럼, 인덱스를 기반으로 다양한 query를 수행할 수 있다. 2~64KB 크기의 SRAM을 지원한다.

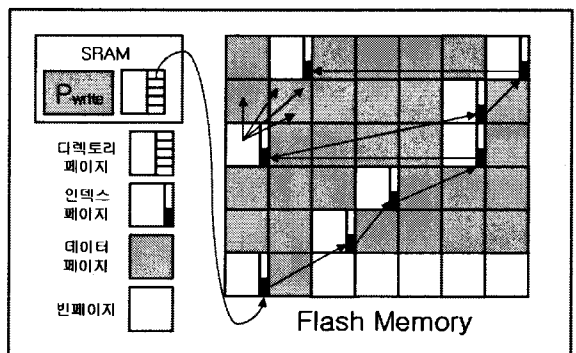


그림 5. MicroHash

MicroHash에서 사용하는 페이지의 종류는 그림 5

에서 보여주는 바와 같이 SRAM에 들어가는 Root 페이지와 Directory, 인덱스, 데이터 페이지이다. 인덱스 변화로 인해 발생하는 phase는 Initialization, Growing, Repartition, Deletion의 네 가지가 있다.

4. 각 저장 시스템의 취약점

각각의 파일 시스템은 그 개발 환경이나 목적이 달라 여러 가지 취약한 부분을 가지고 있다. 가장 먼저 개발된 Matchbox가 불리할 수 있으나 사용될 환경이 제약이 강한 센서 노드라는 부분을 생각할 때 단순함이 강점이 될 수 있다.

4.1 Matchbox

Matchbox는 NOR 플래시 기반이지만, write-append나 seek 명령을 지원하지 않는다. Free-list를 사용하여 공간 관리를 하지만 wear-leveling을 고려하지 않았으므로 앞 부분의 삭제가 자주 발생할 경우 앞부분만 과도하게 사용하게 된다.

Free-list가 메인 메모리에서 관리되므로 전원을 소실하였을 때 free-list 정보도 소실하게 되어 재구성을 해야 하는 문제가 있다.

4.2 ELF

ELF는 Atmel 데이터 Flash 장치에 특성화하여 구현 되어 다른 장치를 대상으로 코드를 재사용할 수 없다.

ELF에서 사용하는 circular 버퍼 형태의 기술은 센서 네트워크에서 사용하는 다양한 응용 프로그램을 만족시키지 못한다.

센서 네트워크의 특징상 자주 발생하는 연산(reboot, file open, file create)을 잘 처리하도록 디자인 되지 않는다. 버퍼링은 ELF의 성능과 효율성에 중요한 역할을 하지만 신뢰성 측면에서 비용을 부담해야 한다. 센서 노드에서 파일의 신뢰성을 보장하기 위한 latency 비용 혹은 에너지 비용을 측정하지 않는다.

4.3 Capsule

capsule에서 제공하는 인덱스 기능은 compile시에 크기가 고정된다. Forward chaining이 되지 않아 앞부분을 읽으려면 가장 뒤부터 전체를 읽어야 한다.

수정 자체는 가능하지만 인덱스가 고정되어 있어 중간 부분이 페이지크기를 초과할 경우 처리를 할 수 없다. 인덱스를 메모리에도 유지시키기 때문에 고정된 크기를 벗어날 수 없으며, 인덱스의 사용이 파일시스템 용도로 사용할 경우에도 사용되면서 파일 크기나 fragment 수의 제한이라는 제약 조건을 만든다.

Compaction에 많은 시간이 요구되어 사실상 이용이 쉽지 않다. 그리고 Capsule에서 제안하는 객체 이외의 자료구조에는 적용이 되지 않아 일반적으로 사용하기 어렵다는 단점이 있다.

4.4 MicroHash의 취약점

MicroHash는 삭제 명령이 LRU(Least Recently Used)에 의해 수행되어, 순차적인 데이터 저장에 의해 가장 오래된 데이터를 버려도 되는 경우가 아니라면 그대로 적용하기 힘들다.

그리고 저장하는 단위인 인덱스의 크기가 일정하게 제한하여야 한다는 단점이 있다.

5. 결론

본 논문에서는 플래시 메모리를 기반으로 한 저장 시스템의 특징과 취약점에 대해서 알아보았다. Match Box는 TinyOS에서 기본적으로 제공하는 기본적으로 제공하는 저장 시스템으로 단순 Append 작업을 지원해주고 코드 크기도 작지만 사용할 수 있는 기능이 적고 용량이 상대적으로 작은 NOR를 기반으로 하고 있다.

ELF는 사용할 수 있는 많은 기능들을 지원해 주고 메모리 소비 시간을 줄였지만 Matchbox와 같이 NOR를 기반으로 Atmel 데이터 Flash 장치에 특성화 되어 있어 다른 장치를 대상으로는 그 코드를 재사용할 수 없고 성능과 효율성 향상을 위해 사용한 버퍼링은 신뢰성 측면에서 단점을 가지고 있다.

Capsule은 객체 기반으로 에너지, 메모리 절약적 디자인에 중점을 두고 설계 되었지만 인덱스의 기능이 compile시에 크기가 고정되어 되고, Forward chaining이 되지 않아 앞부분을 읽기 위해서는 전체를 읽어야 하고 Capsule에서 지원해주는 객체 외에 다른 자료 구조는 지원이 되지 않아 일반적으로 쉽게 사용할 수 없는 단점을 지니고 있다.

그리고 파일시스템보다는 인덱스에 의한 데이터 저장을 지원하는 저장 시스템인 MicroHash는 Timestamp나 value를 통한 플래시에 저장된 레코드로의 효율적 접근 방법을 제공하지만 기반이 되는 인덱스의 크기가 일정해야 된다는 제약점을 지니고 있어 일반적으로 데이터의 양이 가변하는 환경에는 적용하기 어려운 단점을 가지고 있다.

지금까지 비교 분석해본 저장 시스템들은 모두가 대용량 저장 시스템이 아니라는 공통점을 지니고 있다. 그리고 일반적으로 적용하기 어려운 제약점들을 지니고 있다.

앞으로 많은 분야에서 더 많은 데이터를 수집하고 관리하기 위해서는 대용량의 센서 노드용 저장 시스템의 개발이 필요하다. 더욱 일반적이고 적용하기

쉽고 하드웨어에 의존적이지 않은 센서 노드용 저장 시스템을 위해서는 낸드(NAND) 플래시 메모리를 기반으로 한 저장 시스템을 개발해야 할 것이다.

참고 문헌

- [1] Memory Technology Device Subsystem for Linux, <http://www.linux-mtd.infradead.org/doc/nand.html>
- [2] D.Gay. Design of Matchbox : The simple Filing system for Motes. In TinyOS 1.x distribution, www.tinyos.net, Aug2003.
- [3] H.Dai, M.Neufeld, and R.Han. ELF: An efficient Log-structured Flash file system for micro sensor nodes. In SenSys, page 176-187, New York NY, 2004
- [4] Gaurav Mathur, Peter Desnoyers, Deepak Ganesan and Prashant Shenoy, "Capsule: An Energy-Optimized 객체 Storage System for Memory-Constrained Sensor Devices", Proceedings of the Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys), Boulder CO, November 1-3, 2006.
- [5] D. ZeFinalipour-Yazti, S. Lin, V. Kalogeraki, D. Gumopoulos and W. Najjar "MicroHash : An Efficient Index Structure for Flash-Based Sensor Devices", 4th USENIX Conference on file and Storage Technologies (FAST'2005), San Francisco, CA, December 14-16 pp.31-44, 2005.