A Simulator for None-Volatile Memory

Shanhe Yi

Department of Computer Science
College of William and Mary

December 8, 2013

W&M Computer Science

Outline

W&M Computer Science

Outline

W&M Computer Science

Intro: NVM

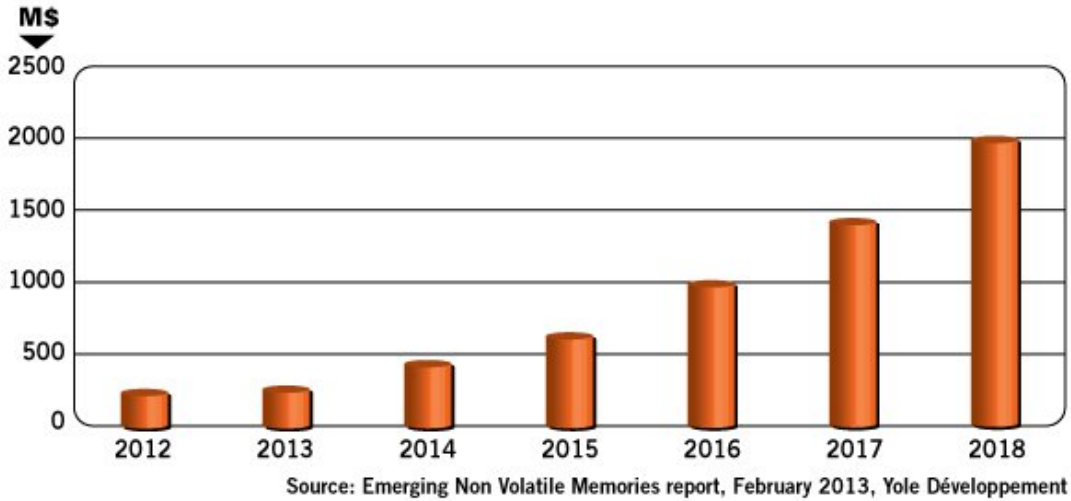


Figure : http://www.mram-info.com/[1]

- None-Volatile Memory(NVM)
  - Persistent
  - Byte addressable
  - Short access time
  - Low power consumption

[1] http://www.mram-info.com/

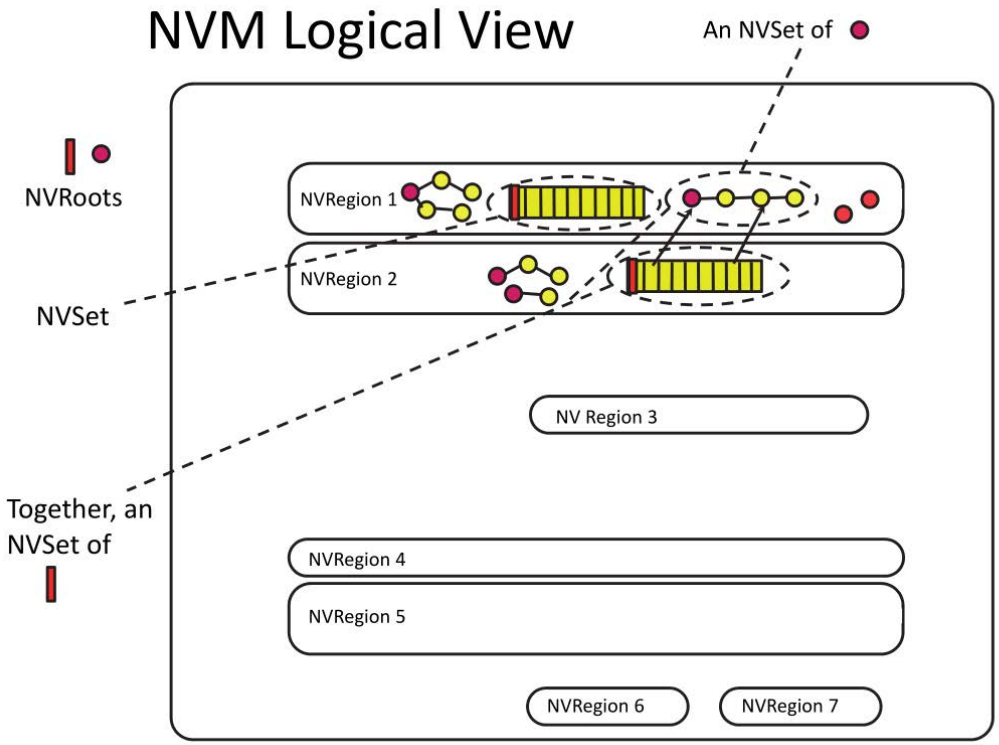
W&M Computer Science

Problem to address

- Problem: How to simulate NVM
- Solution: Using a shared chunk of main memory
  - Global access: Shared memory ✓
  - Transparent to application: APIs ✓
  - Privacy: Permission control
  - Persistent: Assume that processes won't crash, and there is no power outage. ✓
  - Consistent: Read-write lock

W&M Computer Science

Organization of NVM



- `NVRegions`: the basic block in NVM, contains `NVRoot` and `NVSet`
- `NVSet`: graph, array, linked list, individual variables.
- `NVRoot`: leads the `NVSet`.

W&M Computer Science

Outline

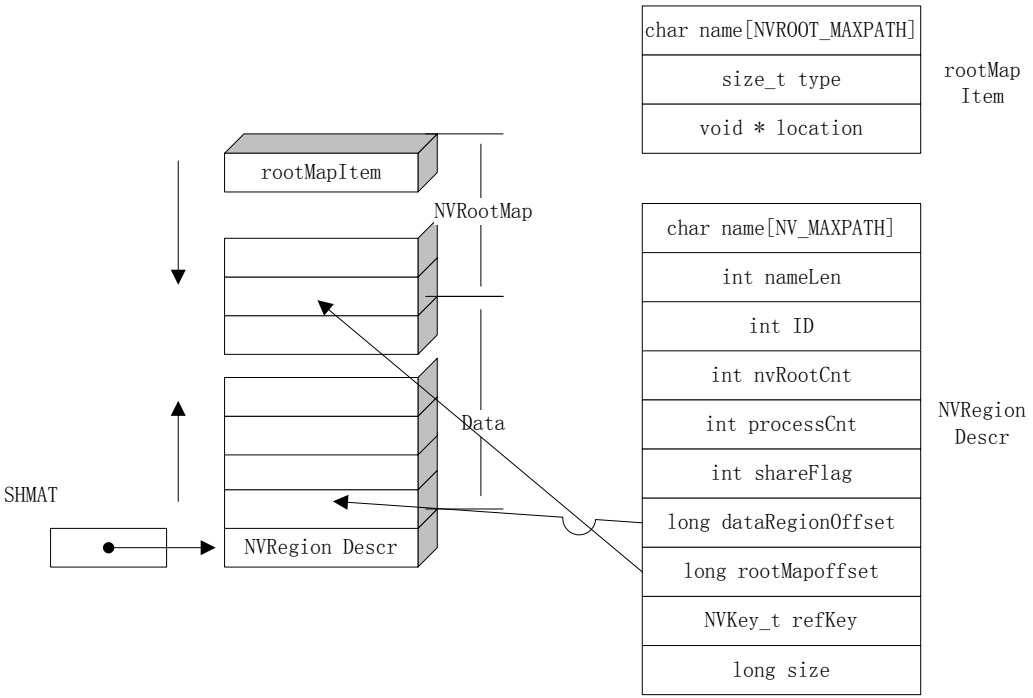
W&M Computer Science

Data Structure

- The NVRegion segments: [MetaData|Data|RootMap]
- RootMap is located at high address and grows into lower address
- NVRegion Metada is located at the lowest address
- Data Region grows from low address to high address.

W&M Computer Science

Data Structure Cont'd



- The NVRegion Descriptor Data Structure
- RootmapItem Data Structure

Architecture

DRAM simulated as NVM

- NVM management
  - Inside NVM Management.
    - Use a big chunk of main memory and put all NVRegions in it.
    - Need to implement NVM management inside the memory chunk.
  - ✓ Outside NVM Management. NVRegions are logically together but physically distributed.
    - flexible, process only need to open the region they need
    - offload partial management to kernel or OS (System V Shared Memory APIs)
    - get some metadata directly((`shm_ds`, `file stat`).)

W&M Computer Science

Outline

W&M Computer Science

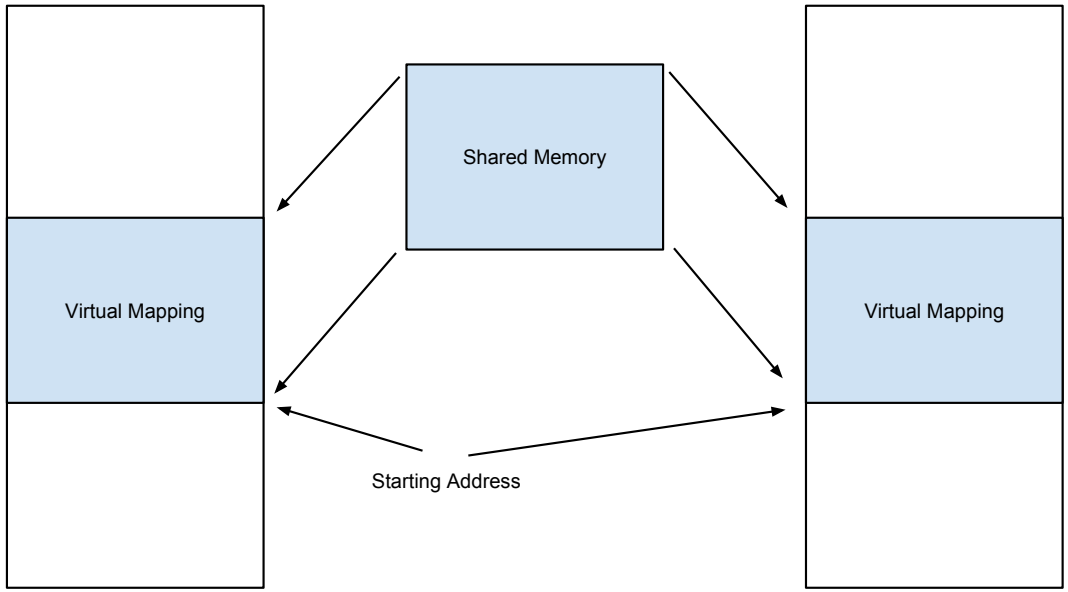
Shared Memory

- System V Shared Memory APIs
  - `shmget` can be used to fetch an existed shm or create a new one in kernel.
  - `shmat` attach the shm to a certain address of process address space
  - `shmdt` deattach the shm from a process's address space

W&M Computer Science

Shared Memory Cont'd

- SHMs are all attached at the same starting address(so we can use pointer directly.).
- SHM size is current fixed (30MB).



W&M Computer Science

Memory Mangement

- Naive, Implicite Free List,
- ✓ Explicit Free List[2](`malloc`,`free` and `realloc`).
  - Aligmment
  - Block header: size and tags
  - Explicit pointers below block header to next and previous free blocks
  - Boundary tag at the block tail indicate the size(free block coalescing)
  - First fit

[2] http://csapp.cs.cmu.edu/

W&M Computer Science

Outline

W&M Computer Science

Evaluation

- 2k LoC C [3]
- test: Call each API functions.
- Two examples
  - Array Assignment
  - Sort

[3] https://github.com/yishanhe/nvm-simulator

Outline

W&M Computer Science

Conclusion

- A NVM simulator is designed and implemented by using main memory
- The NVM management take advantage of the shared memory management offered by kernel.
- A dynamic memory allocation using explicit free lists is implemented to manage the nvm-inner data region memory.

W&M Computer Science

Future Work

- Transform between pointer and address offset
- Improve the RootMap managements(linked list, validate tag bit.).
- Provide an define option to use `mmap` and `munmap` instead of `shmget`, `shmat` and `shmdt`
- Multiple processes with fine grained read-write lock.
- Process crash and power outage(write through, write back).
- Thread-safe `NVMalloc` and `NVFree`

W&M Computer Science

End.
Thanks
Any Question?

W&M Computer Science