Power Analysis for Flash Memory SSD

Dongkun Shin
Sungkyunkwan University
dongkun@skku.edu

Introduction

- SSD requires low power/energy than does HDD.
  - attractive to mobile systems and power-hungry data centers
- Recent SSDs use
  - Intensive parallel schemes ➔ Peak Power
  - Large amount of DRAM buffer
  - Energy(SSD) ≈ Energy(HDD) ?
- many researches on power analysis and optimization for HDD.

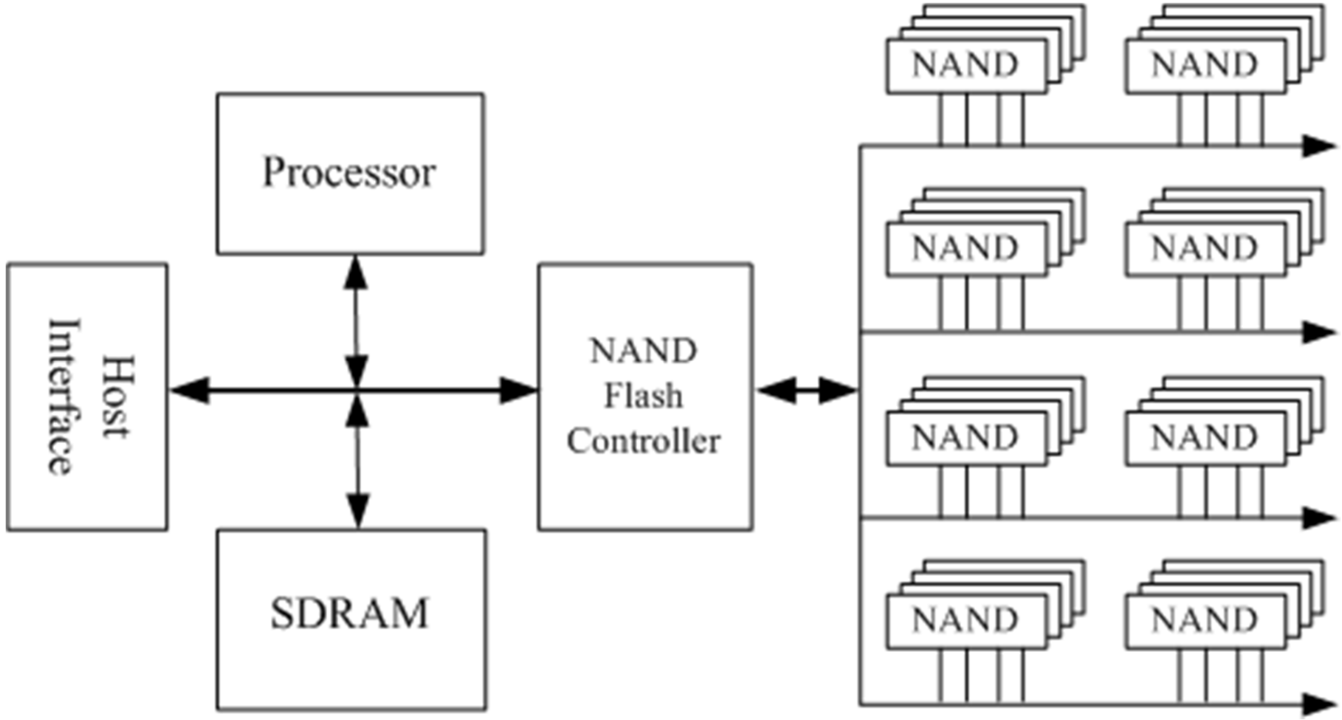
Introduction

- To optimize Energy(SSD), need to characterize the power/energy consumption of SSD.
- depending on the I/O request patterns
- several hints on energy optimization.
- enables to extract several architectural features of target SSD which vendors do not provide to users

Related Works

- Seo: Empirical analysis on energy efficiency of flash-based SSDs. HotPower'08
  - only simple access patterns, no detailed analysis
- Park: Power modeling of solid state disk for dynamic power management policy design in embedded systems. SEUS '09
  - power consumption simulator for SSD.
  - consider parallel flash chip accesses
  - too simple power model
- Lee: Advances in flash memory SSD technology for enterprise database applications. SIGMOD '09
  - Single SSD can outperform several HDDs comprising RAID for both power consumption and I/O performance.
- Mohan: FlashPower: A detailed power model for NAND flash memory. DATE '10

SSD Architecture



Target SSDs

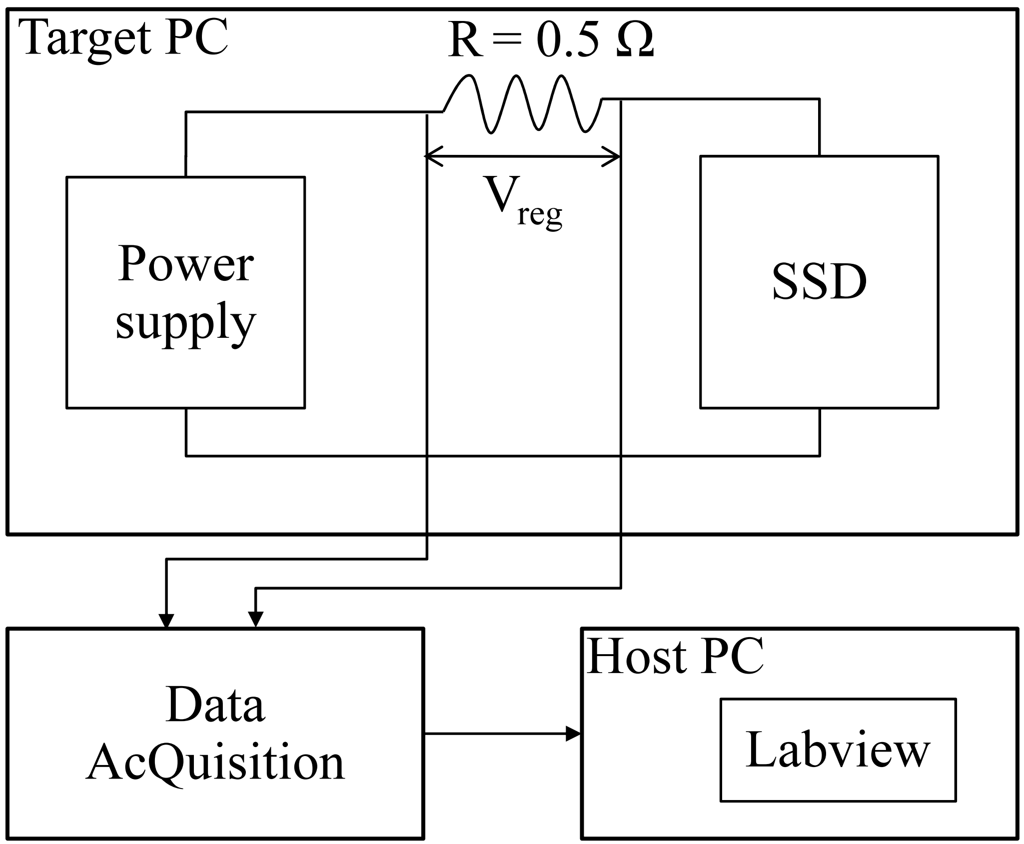
- The performance/power of SSD is determined by
  - Cell type of flash memory
  - DRAM buffer size
  - # of parallel flash chips
  - Firmware
- SSD(H) provides higher I/O performances
  - uses a larger internal buffer and a more intelligent FTL.

	SSD(L)	SSD(H)
Capacity	64 GB	30 GB
Flash Chips	K9NCG08U5M × 8	K9GAG08U0M × 16
Memory Type	SLC	MLC
DRAM	32 MB	64 MB
Max Read	100 MB/s	220 MB/s
Max Write	80 MB/s	130 MB/s

Target SSDs

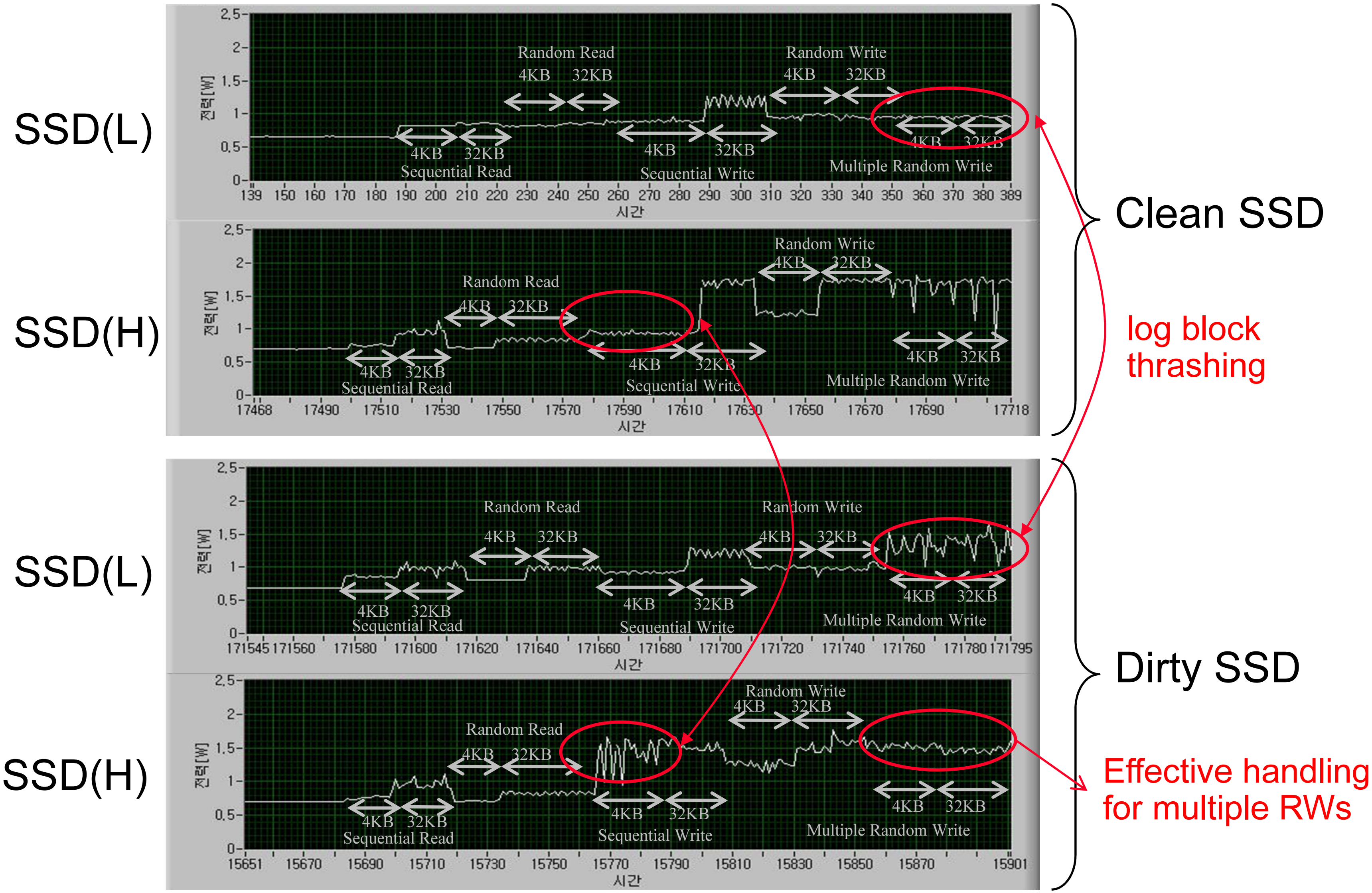
- Dirty SSD
  - write files up to the amount of SSD capacity
  - delete all the files in file system level
  - each write operation will invoke garbage collection
- Clean SSD
  - All flash memory blocks are erased thus data can be written at the blocks without GC
  - HDDerase tool which executes the secure erase command in order to erase flash blocks.

Power Measurement



NI USB 6221
250 kS/s

Power Changes



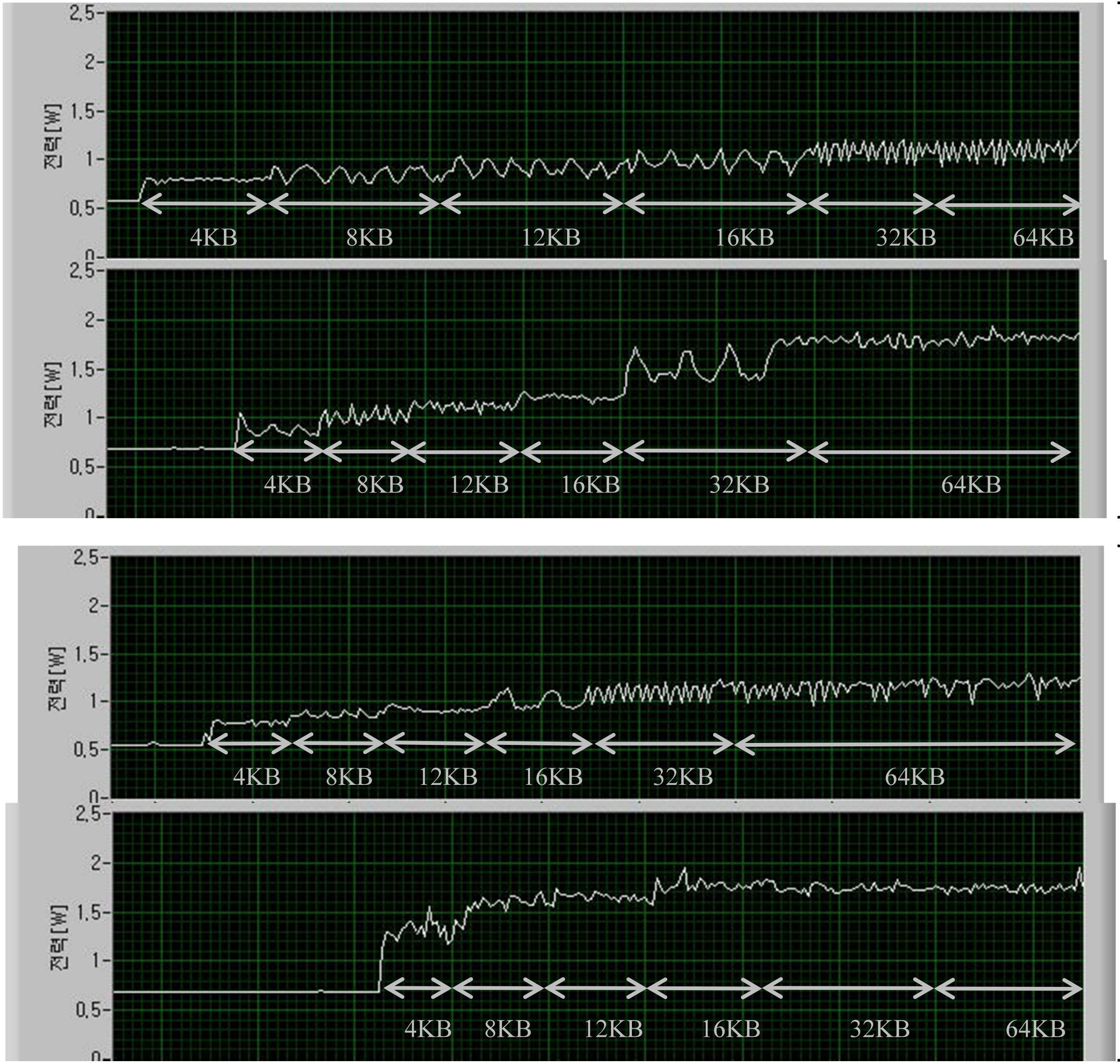
Power Changes

SSD(L)

SSD(H)

SSD(L)

SSD(H)



Clean SSD

Dirty SSD

Power Changes

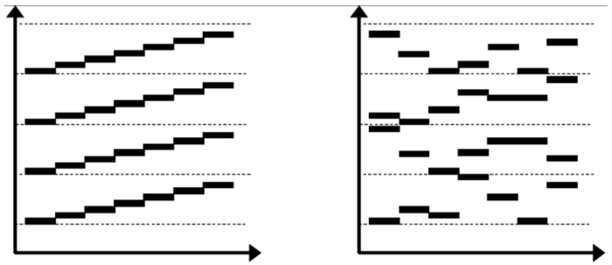
- Power(MLC)>Power(SLC), En(SSD-H)<En(SSD-L)
- Idle state requires 0.6 watt
  - about 40% of the peak power consumption in the active state
  - about twice the power consumption of a typical application processor (≈0.3 watt)
  - about 60% of the standby power of netbook (≈ 1 watt)
  - it is necessary to shut down SSD when it is idle.
  - OCZ datasheet: 2W in operation, 0.5W in stand by
- SSDs have DRAM write buffers
  - But, precipitous increases at the moment the requests are sent to SSD
  - The data from host does not remain at the DRAM buffer during a long time

Micro-Benchmark

- Measure Joule/MB and watt of SSDs
- 4 benchmarks from uFlip
- Alignment benchmark
  - Effects of unaligned I/O requests
  - Shift the start address of the baseline requests that have the I/O size of 32 KB and are aligned by the I/O size
  - to know the address mapping unit of target SSD as well as the adverse effect of unaligned requests.
- Granularity benchmark
  - I/O requests with different I/O sizes from 1 KB to 4 MB.
  - to identify the number of parallel flash chips accessed simultaneously

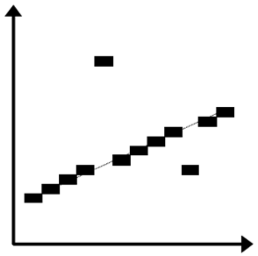
Micro-Benchmark

- Parallelism benchmark
  - 32 KB-sized I/O requests from parallel processes each of which accesses a different region of storage space



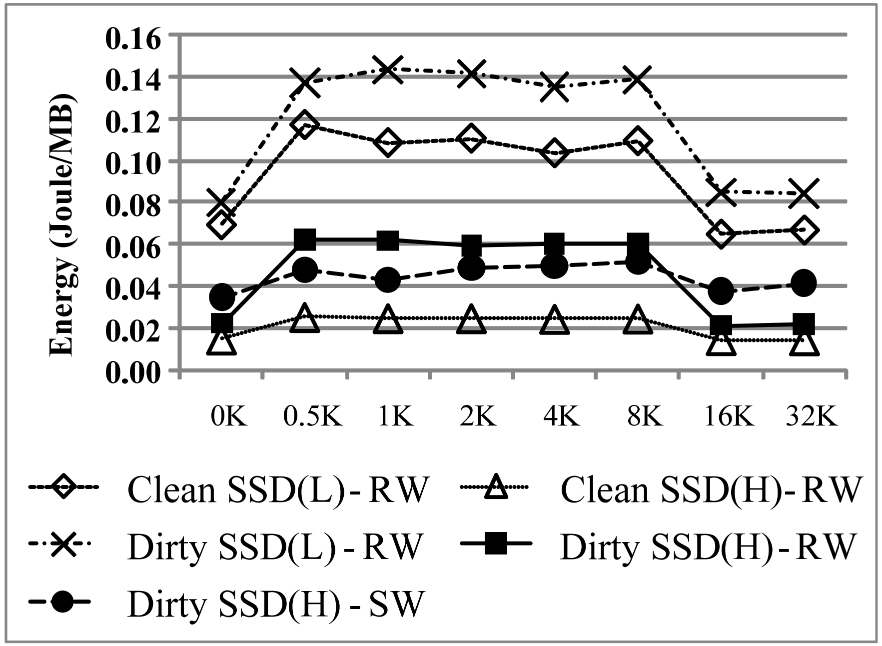
SW (*ParallelDegree* = 4) RW (*ParallelDegree* = 4)

- Mixture benchmark
  - 4 KB-sized write requests by interposing RW requests between SW requests.
  - RW:SW = 1:64
    - 64 number of RW requests are interposed between each SW request (random access pattern)
  - RW:SW = 64:1
    - one RW request is interposed at every 64 number of SW requests (sequential access pattern)

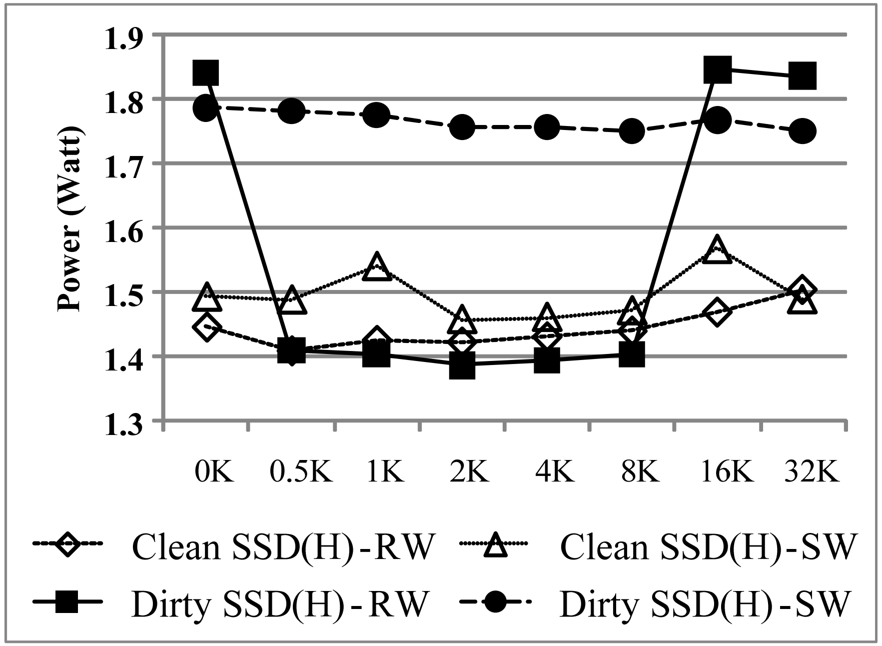


4 SW / 1 RW

Alignment (32KB)



(a) Energy consumption



(b) Power consumption

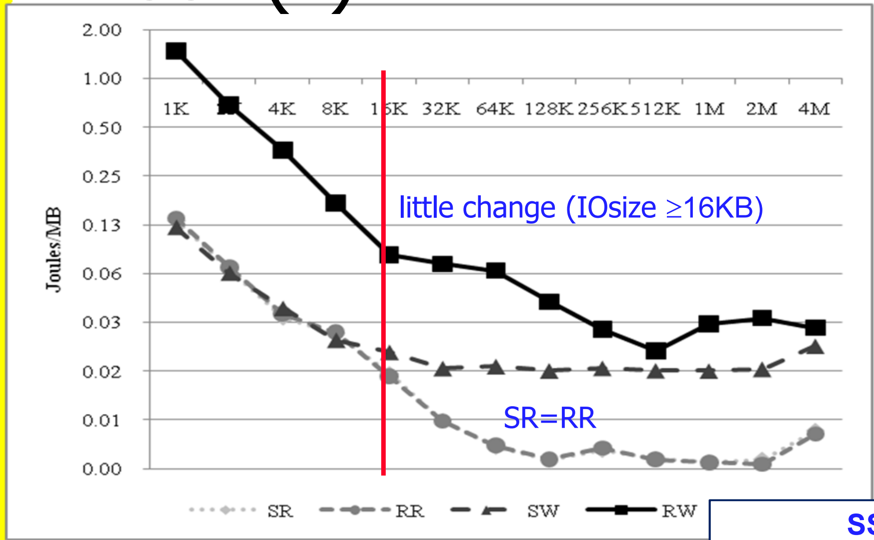
- Observation
  - RW patterns with the shift size between 0.5 KB and 8 KB need more energy
  - If shift size is a multiple of 16 KB, similar to the aligned baseline pattern.
  - SW requests show little changes

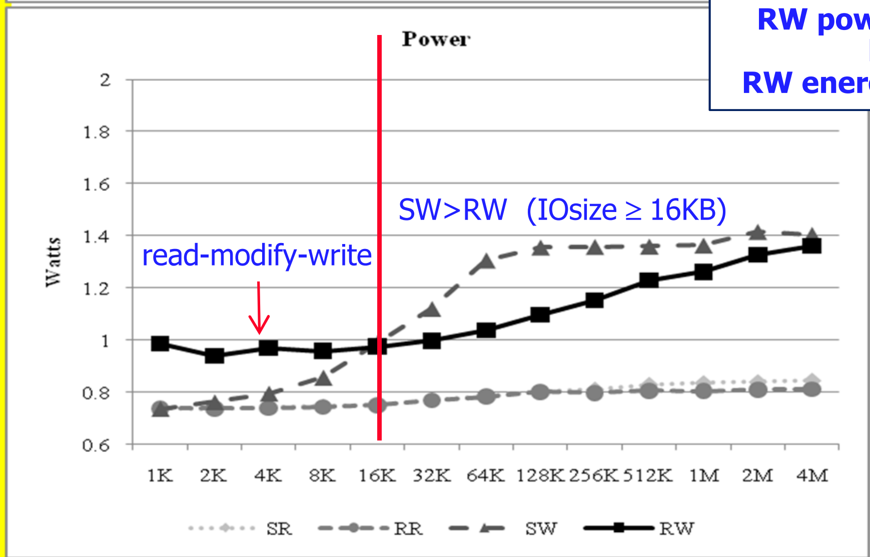
Alignment (32KB)

- SSD performs additional works if requests are not aligned by 16 KB unit.
  - If not aligned, SSD reads two 16 KB units, modifies a part of them and writes them.
  - Read-modify-write operation invokes the read operation that consumes a smaller power than the write operation, the average power consumption of unaligned RW requests is smaller than that of aligned requests
- Clean SSD has little changes
  - does not require the read operation.
- Examined SSDs use 16 KB address mapping unit
  - the whole 16 KB unit should be modified even when only a portion of the unit is modified.

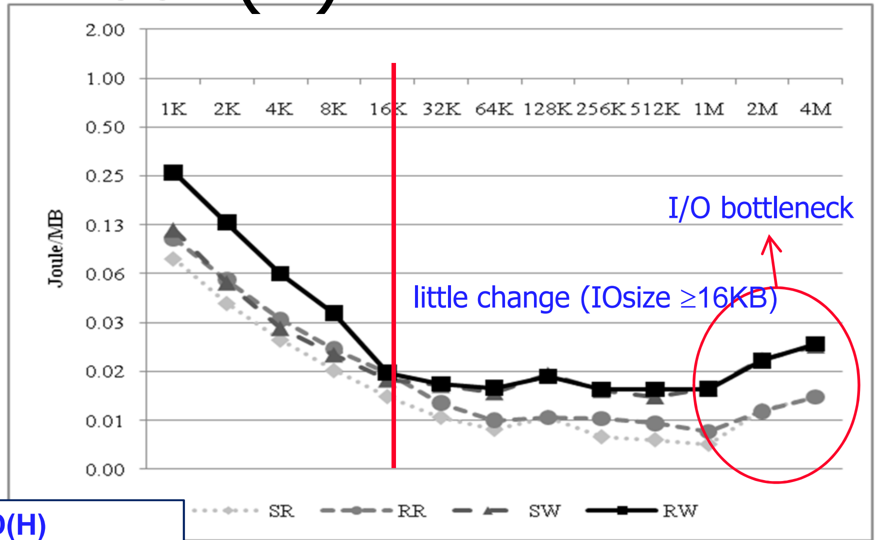
Granularity (Clean SSD)

- SSD(L)

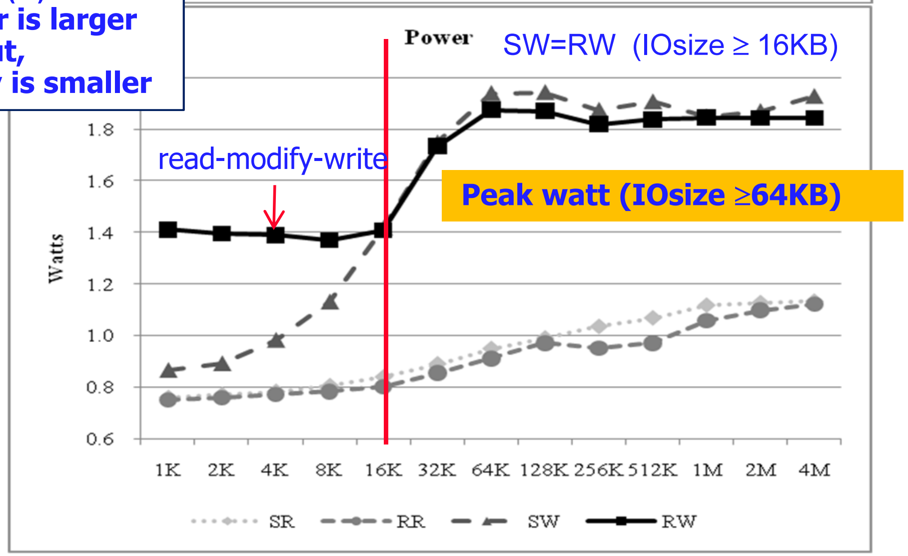




- SSD(H)



SSD(H)
RW power is larger
but,
RW energy is smaller

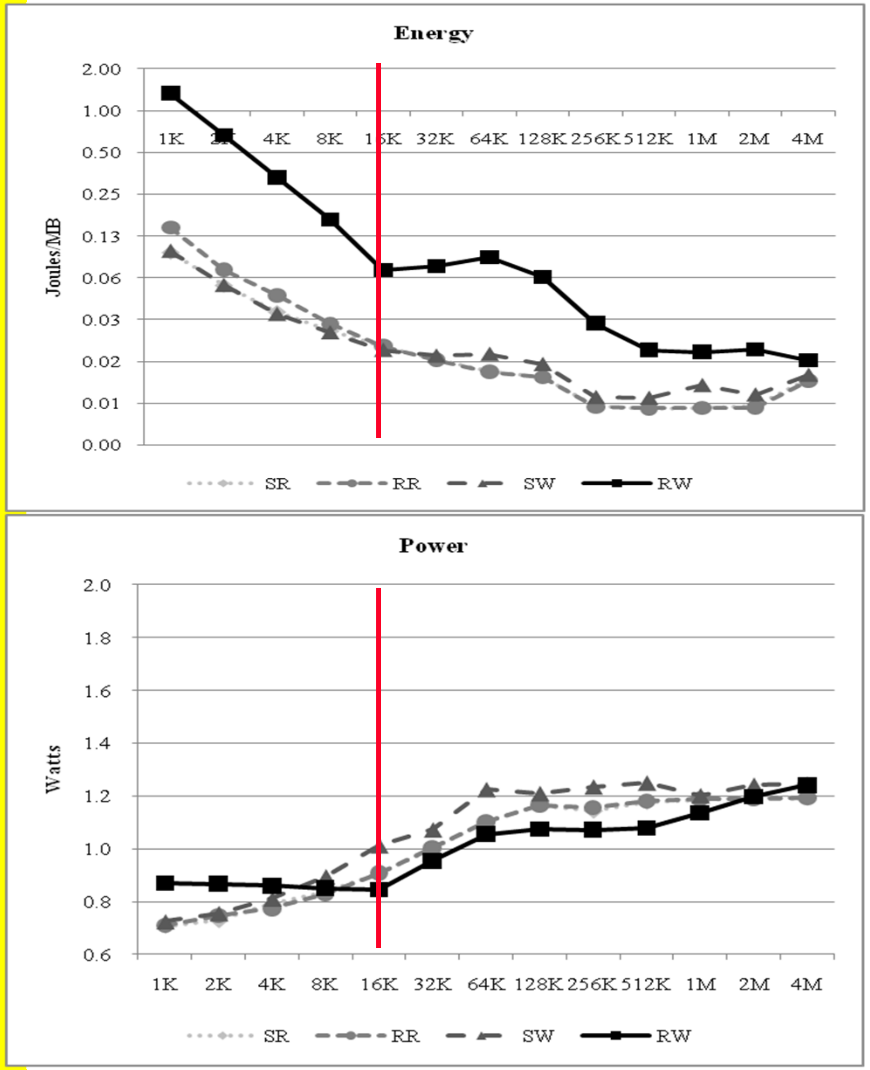


Granularity (Clean SSD)

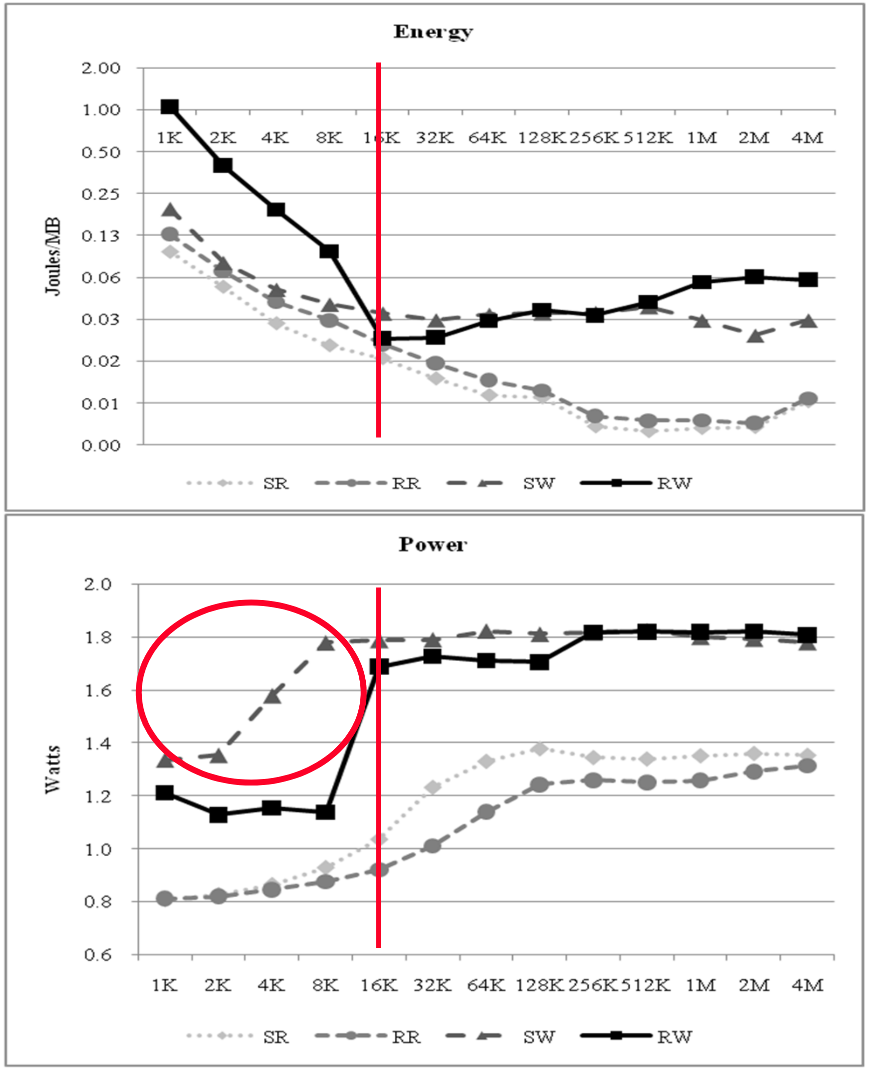
- Power(SSD-MLC) > Power(SSD-SLC) , Energy(SSD-H) < Energy(SSD-L)
  - SSD(L) : lower performance for the random write requests.
  - SSD(H) : intelligent FTL algorithm against RW requests
- The most outstanding change occurs when I/O size ≥ 16 KB.
  - I/O size <16 KB: read-modify-write operation and therefore the energy consumption is significantly high but the power consumption has no change.
  - I/O size ≥ 16 KB: the power of both RW and SW increase as the I/O size increases.
  - I/O size ≥ 64 KB: both SW and RW requests have little change on the power consumption in SSD(H).
- The largest I/O size which can be handled in parallel is 64 KB
  - # of parallel flash chips is 16 (= 64 KB / 4KB)
- Power(RW) < Power(SW) when I/O size ≥ 16 KB in SSD(L)
  - cannot utilize the parallel chips efficiently for random requests.

Granularity (Dirty SSD)

- SSD(L)



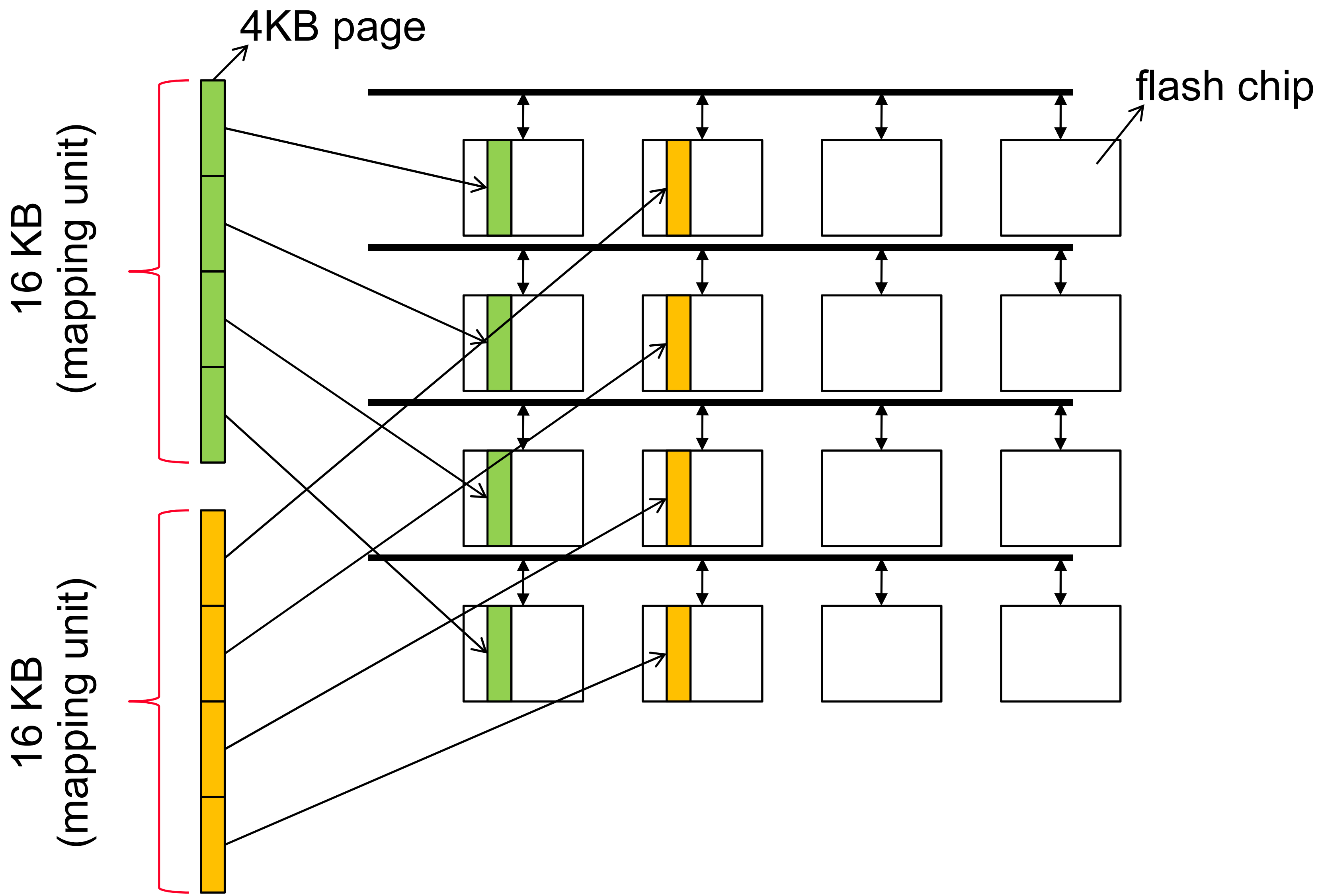
- SSD(H)



Granularity (Dirty SSD)

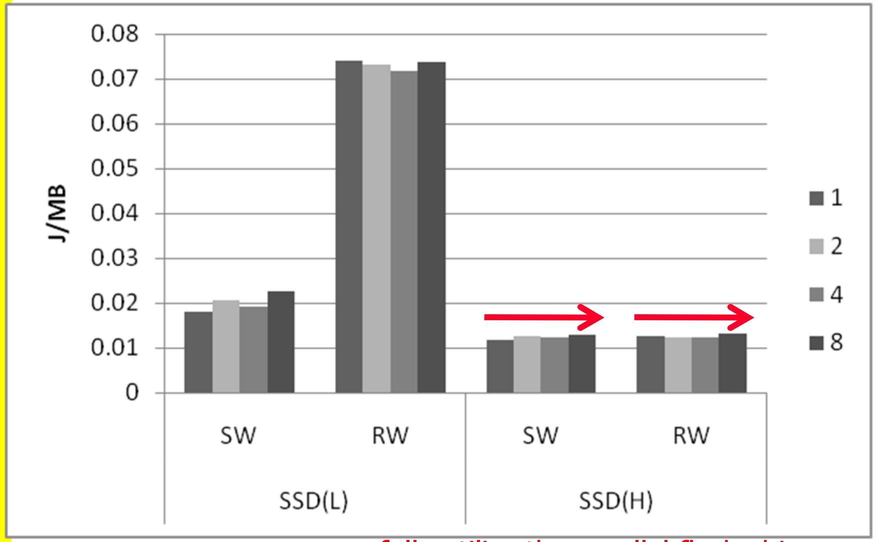
- SW requests consume higher energy/power at the dirty SSD(H).
- Energy/Power increases for read operations

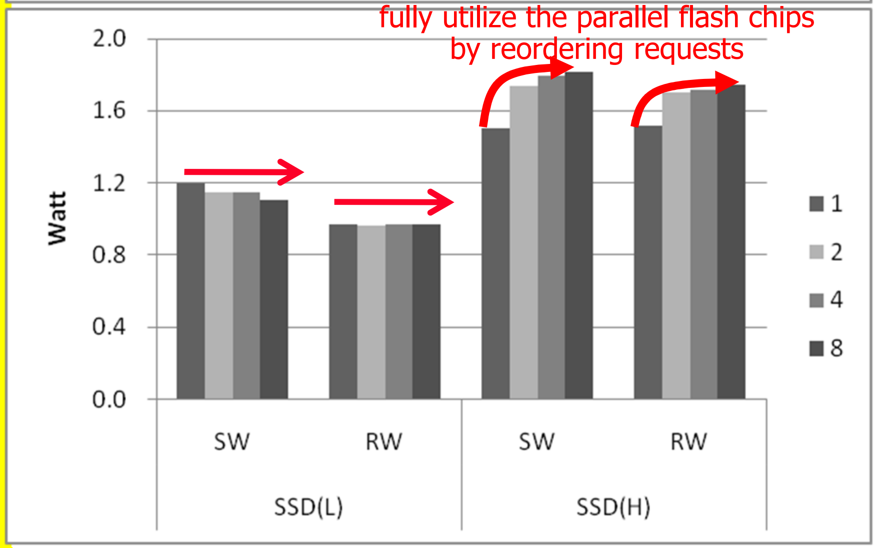
Imagine SSD Internals



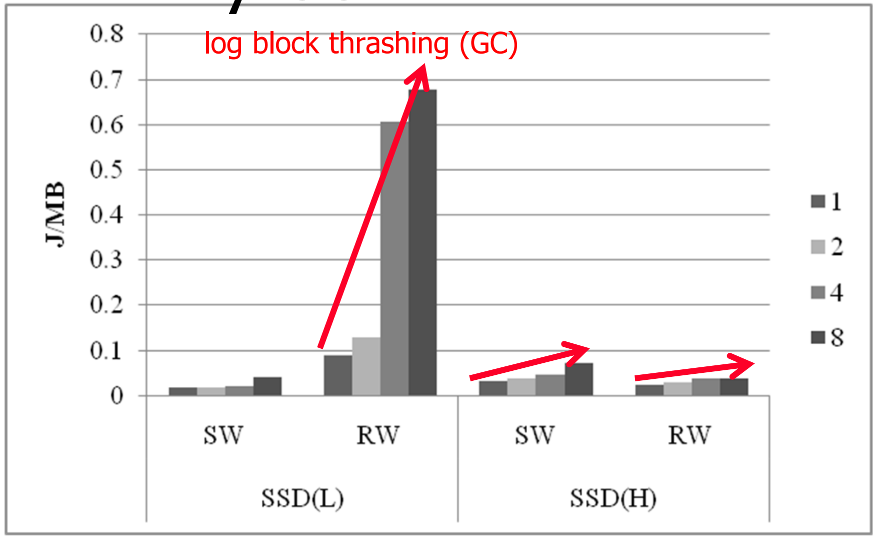
Parallelism (32KB)

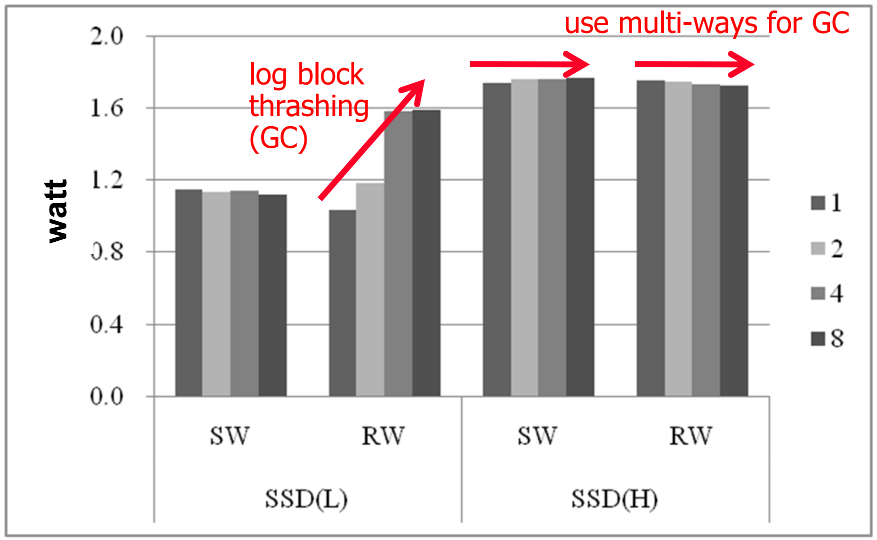
- Clean SSD





- Dirty SSD





Parallelism

- Clean SSDs
  - no large changes on the energy consumption depending on the number of parallel I/O requests.
  - The power consumption of SSD(H) increases as the number of parallel requests increases.
    - more flash chips are accessed for the parallel requests.
    - Even for random requests, SSD(H) can fully utilize the parallel flash chips by reordering requests.
  - Power consumption of clean SSD(L) shows no change since it cannot utilize the parallel flash chips efficiently.
- Dirty SSDs
  - Energy consumptions increase as the parallelism increases.
  - garbage collections, long latency of erase

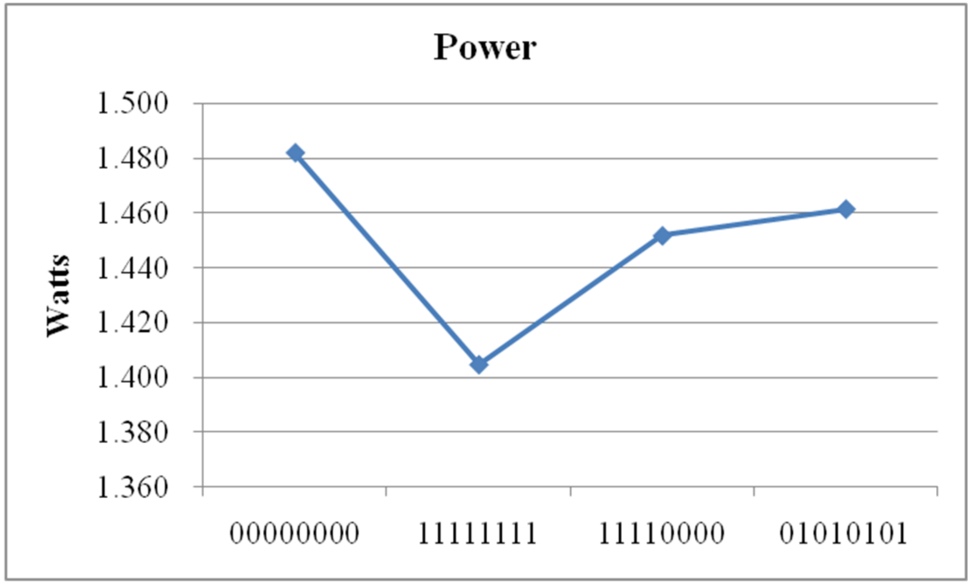
Mixed Pattern (4KB)

(a) Energy consumption (b) Power consumption

Mixed Pattern

- SSD(L) and SSD(H) show completely different results.
- As the portion of SW increases, Energy of SSD(L) decreases.
- Energy of SSD(H) is highest when the portions of SW and RW are similar rather than when the access pattern is SW or RW dominant.
  - We presume that SSD(H) uses the sequential log block (SLB) where data are written by the in-place manner.
  - The SLB is used for efficient handling sequential write requests.
  - When random and sequential write requests are mixed, SLB cannot present its advantage since the interposed random requests obstruct the in-place write.
- SSD(H) can handle the random requests efficiently
  - Energy consumption gap between SSD(H) and SSD(L) is large for random-dominant request patterns.

Data Pattern



No significant difference depending on data pattern (max 5%)

Macro-Benchmark

Postmark

smaller than mapping unit (SS, SL)

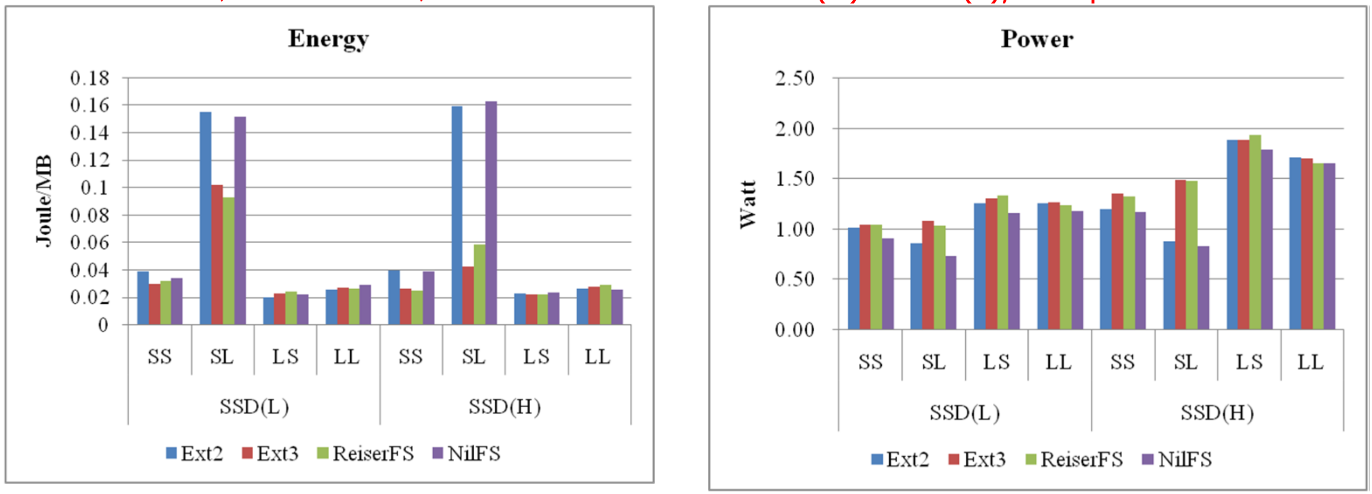
Workload	File Size	Work Set	Transactions	Data (Read/Write)
SS	9k ~ 15k	10,000	100,000	630M/755M
SL	9k ~ 15k	100,000	100,000	600M/1.8G
LS	100k ~ 3M	1,000	10,000	9.7G/12G
LL	100k ~ 3M	4,250	10,000	10G/20G

- Linux File Systems
  - Ext2
  - Ext3
  - ReiserFS
  - NilFS

Linux File System (Read&Write)

ext2, nilfs >> ext3 , reiserfs

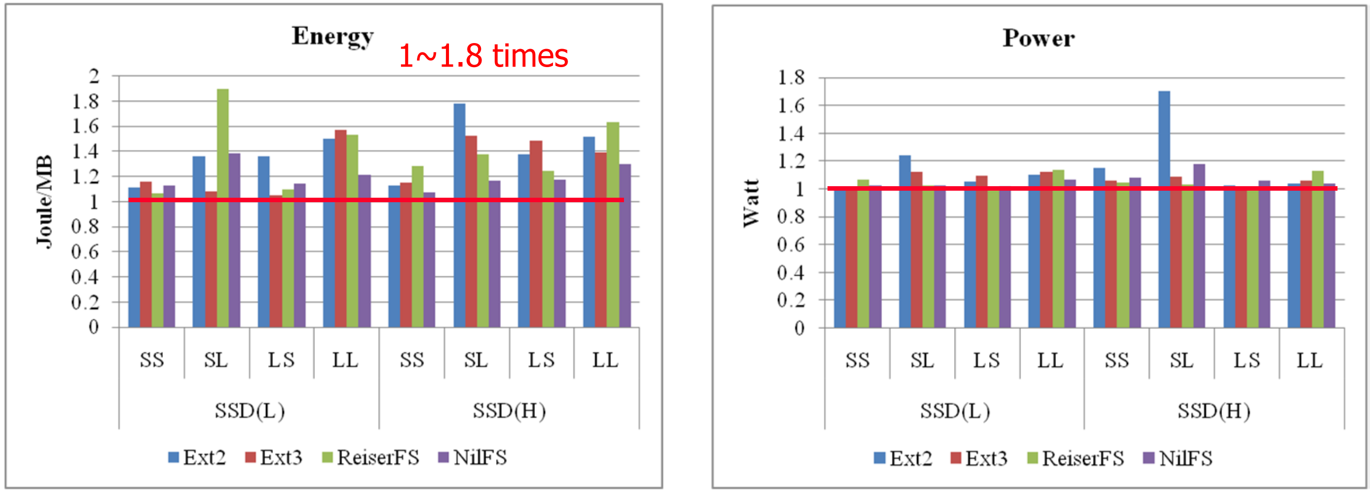
SSD(H) > SSD(L), low power at NILFS



low average power
due to idle times

ext3:ordered

(a) Clean SSD



(b) Relative values of dirty SSD

Conclusions

- Power consumption at idle state is not negligible
  - need an aggressive dynamic power management technique
- Power(MLC)>Power(SLC), but Energy is determined by performance
  - Smart FTL can reduce the energy consumption
- The address of write request should be aligned by the mapping unit
- The size of write request should be a multiple of the mapping unit
  - merge small writes
- Peak power is determined by the parallelism (maximum number of parallel flash chips)
  - can control the peak power
- Random writes on wide address range or multiple requests ➔ log block thrashing
- Mixed pattern can deteriorate the energy efficiency.
- Little difference depending on data pattern

Conclusions

- Different file systems require different energy consumptions
  - Ext3 is best at clean SSD
  - NILFS (log-structured FS) is best at dirty SSD
  - Different idle times
- Dirty SSD requires a larger energy than clean SSD (about 40%)
- Little difference depending on I/O scheduler