Storage Class Memory as a Fast Paging Device

Presented at NVRAMOS 2011 Spring
April 19, 2011
Joo-Young Hwang, Yang-Woo Roh, Min-Chan Kim
S/W선행연구-PJ, Flash 개발실,
Memory Division, Samsung Electronics

Agenda

- Introduction
- Swap IO characteristics
- OS overheads for SCM paging device
  - Swap out overhead
  - Swap in overhead
- SCM vs. NAND as a paging device: performance comparison
  - Swap workload pattern analysis
  - Performance simulation results
  - Lifetime estimation
- Summary

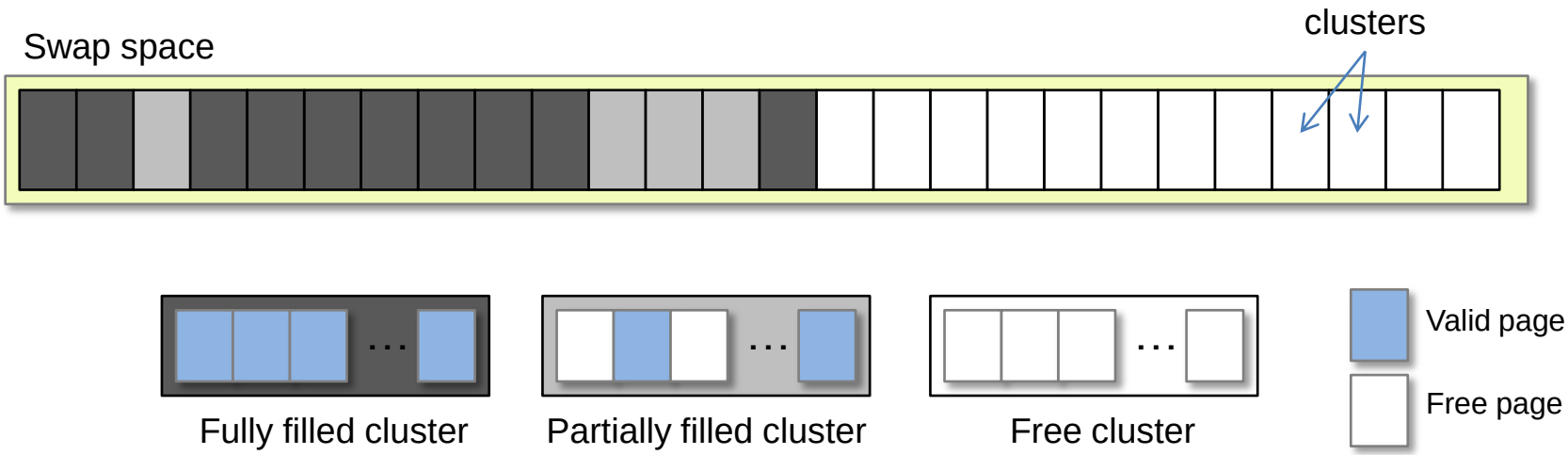
Introduction

- Multi-tasking in smart phone
  - We want more apps running together.
- More and more memory?
  - Cost (and power) for memory
- Why not paging?
  - Latency in interaction
- Which device to use for paging?
  - Do we need high random write performance?
  - Endurance requirement?

Swap IO Characteristics

- Access Localities
  - Pages swapped out together are likely to be swapped in together
  - Recently swapped out pages are unlikely to be accessed for a short time

- Conventional Swap Mechanism for HDD
  - Read-ahead (usually 32KB)
    - Read-ahead size can be configured via /proc/swap/cluster_size
  - Allocate a cluster (1MB size), write sequentially within a cluster
  - Forward scanning for free cluster until *swap cluster utilization** is lower than 50%.
  - Scanning for free cluster in the first half when swap cluster utilization is higher than 50%.
  - Reuse obsolete pages (swapped-in pages) in a cluster if there is no free cluster.



* Swap cluster utilization: touched clusters / total clusters

Swap Optimizations for SSD

- Linux Swap Optimizations for SSD (Linux 2.6.29 ~)
  - Don’t reuse obsolete pages until swap device is full
    - Seek latency is meaningless
    - Reduce overwrite, increase sequential writes
  - Issue TRIM command for obsolete pages
    - Remove garbage collection overhead due to obsolete page copy

- Possible Optimizations
  - Block-aligned swap-in
    - Writes are aligned to block if swapped-in pages are reused.

Storage Class Memory (SCM)

- SCM definition
  - Byte addressable
  - Can be updated in-place without erase
  - (expected to be) Faster than NAND
  - (expected to be) More reliable than NAND
  - (expected to be) Cheaper than DRAM/NAND

Samsung PRAM

Issues

- OS overheads for SCM paging device
- SCM vs. NAND as a paging device: Performance Comparison

OS Overhead Analysis

- Experimental Environment: SMDKV210
  - CPU: S5PV210 (Samsung Cortex-A8) – 800MHz
  - Memory: 8 Gb x DDR2

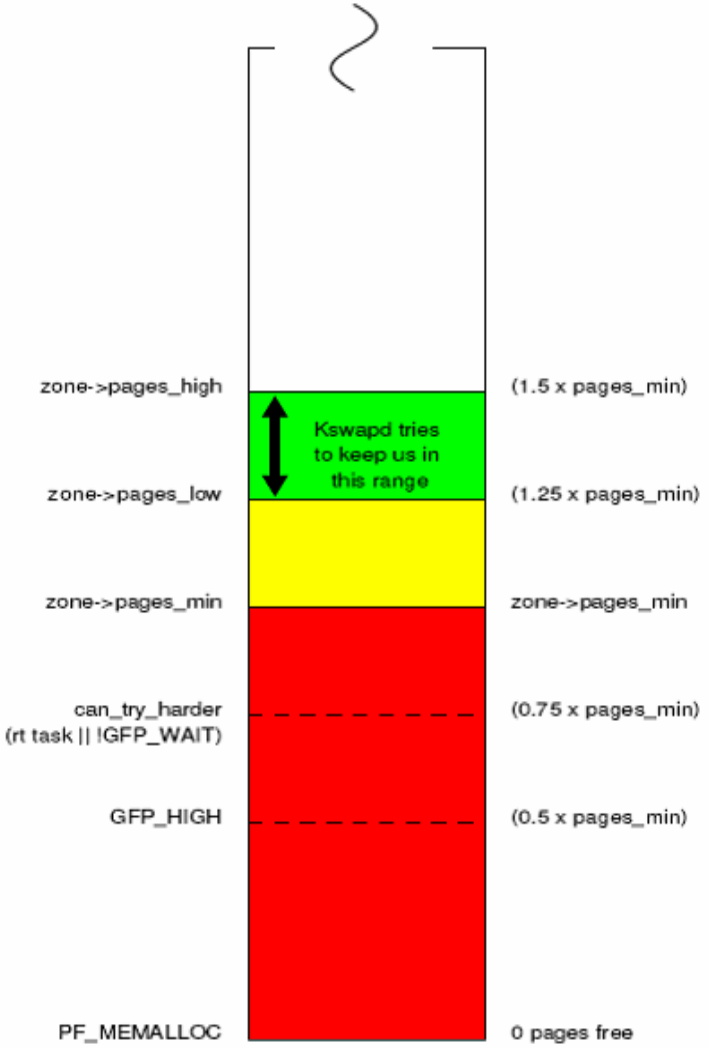
- OS: Linux 2.6.32

Page Reclaim

- Maintain the maximum working set in memory while maintaining sufficient free pages to ensure critical operations won't block.

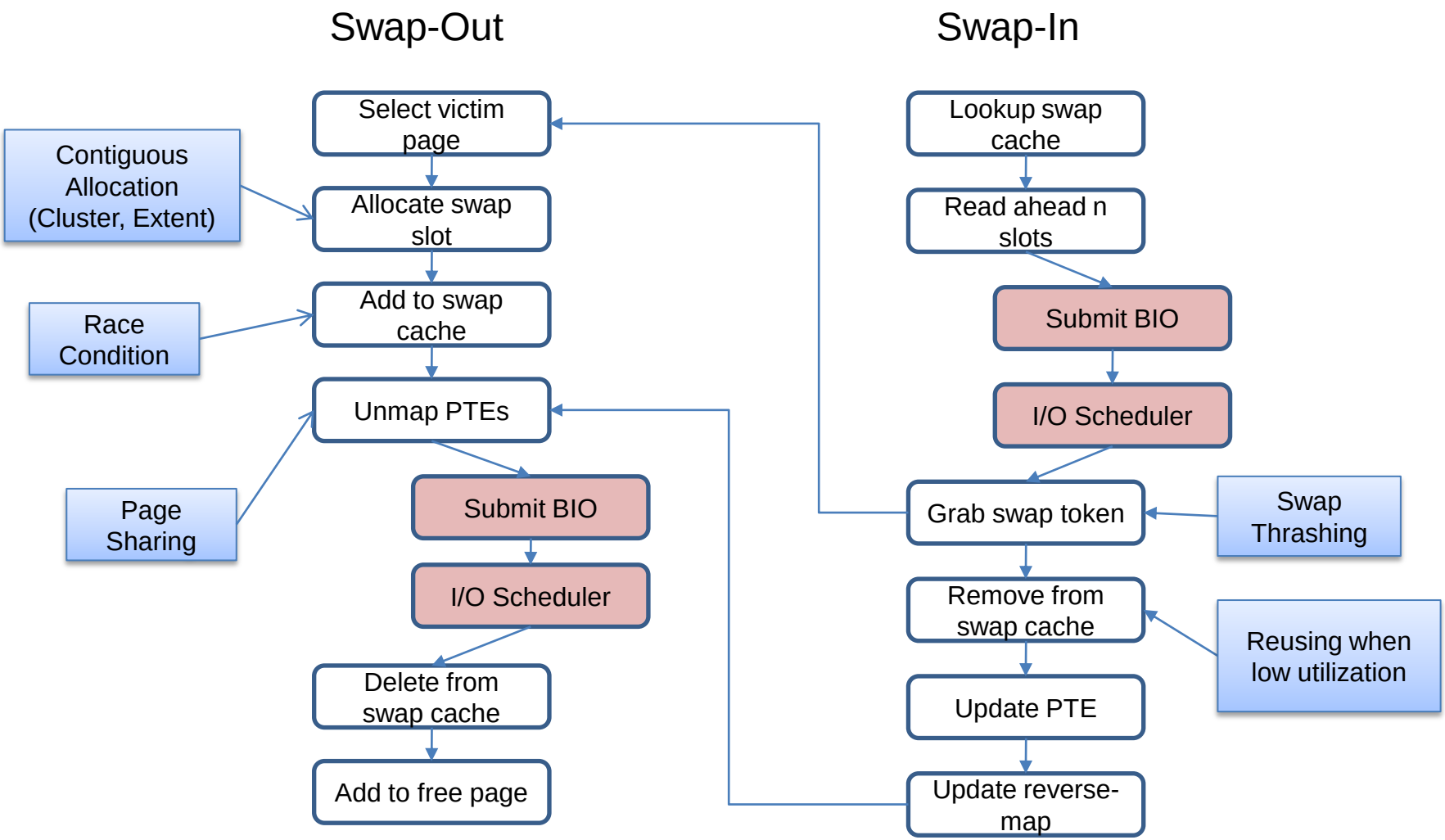
- Background Reclaim Path
  - Wake up kswapd if
    - Zone->free < pages_low
  - Kswapd stops if
    - zone->free > pages_high
  - Why need it ?
    - When allocation of interrupt context

- Direct Reclaim Path
  - Execution point
    - zone->free < page_min
  - Stops if
    - generally free_pages >= 32



* This slide is excerpted from Kim Min-Chan's Linux MM lecture slides.

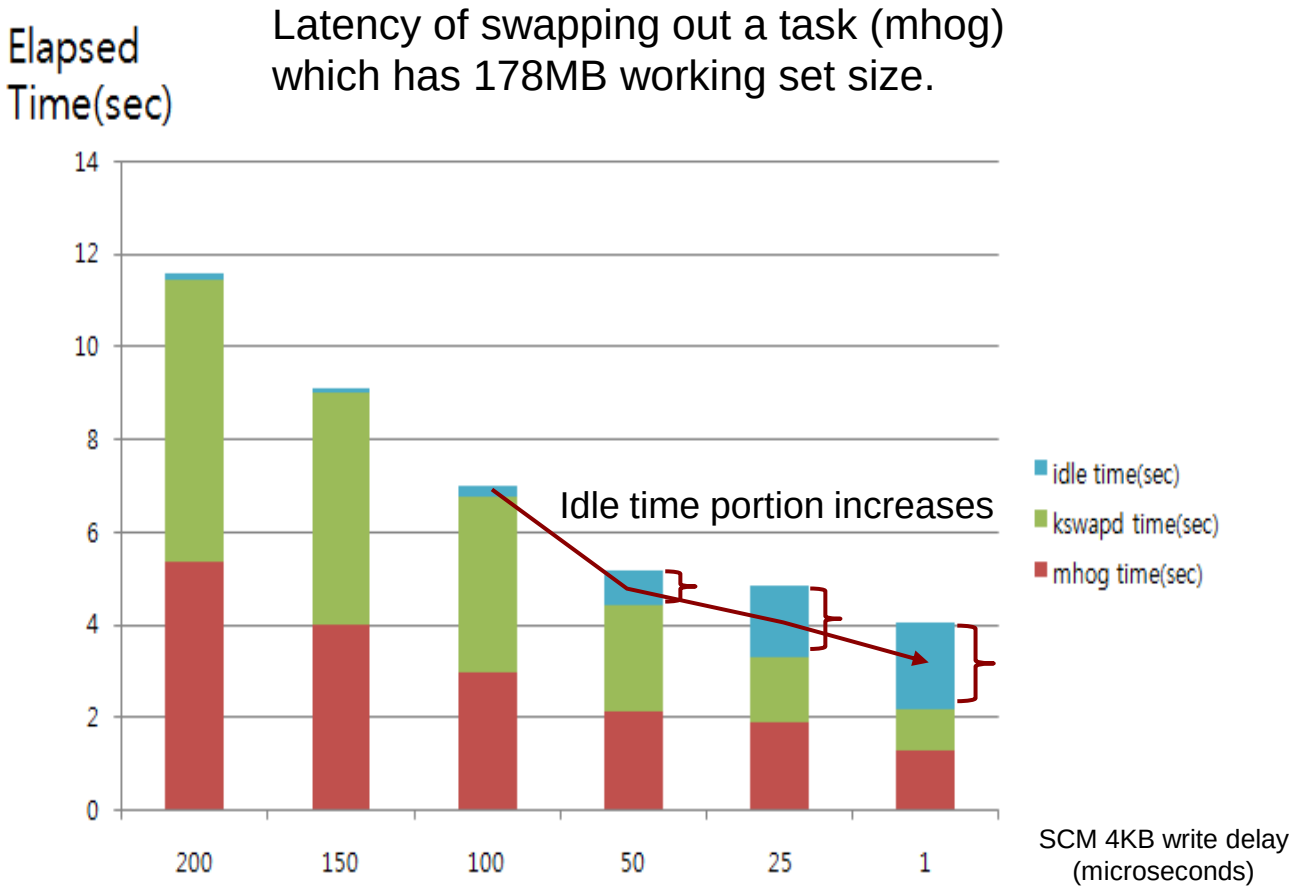
Linux Swap Subsystem



* This slide is excerpted from Yang-woo Roh's Linux swap lecture slides.

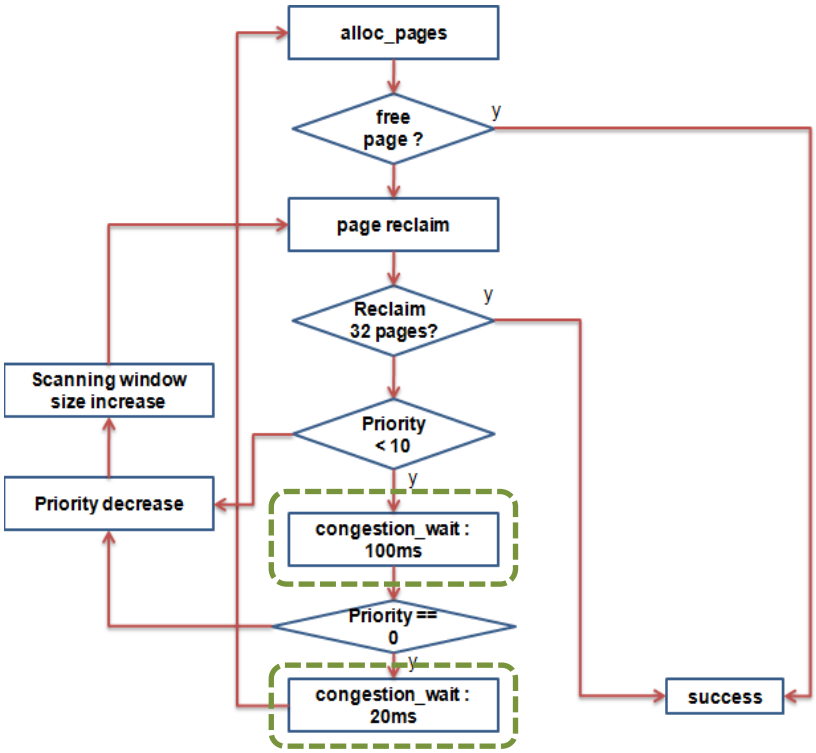
Swap Out Overhead Analysis

- OS Overhead Issue
  - SCM의 write speed를 증가시키더라도 일정 수준 이후에는 OS overhead로 인해서 더 이상 swap out 성능이 개선되지 않는 현상이 있음.
  - SCM random write IOPS > 20K: idle time overhead 비중이 큼.

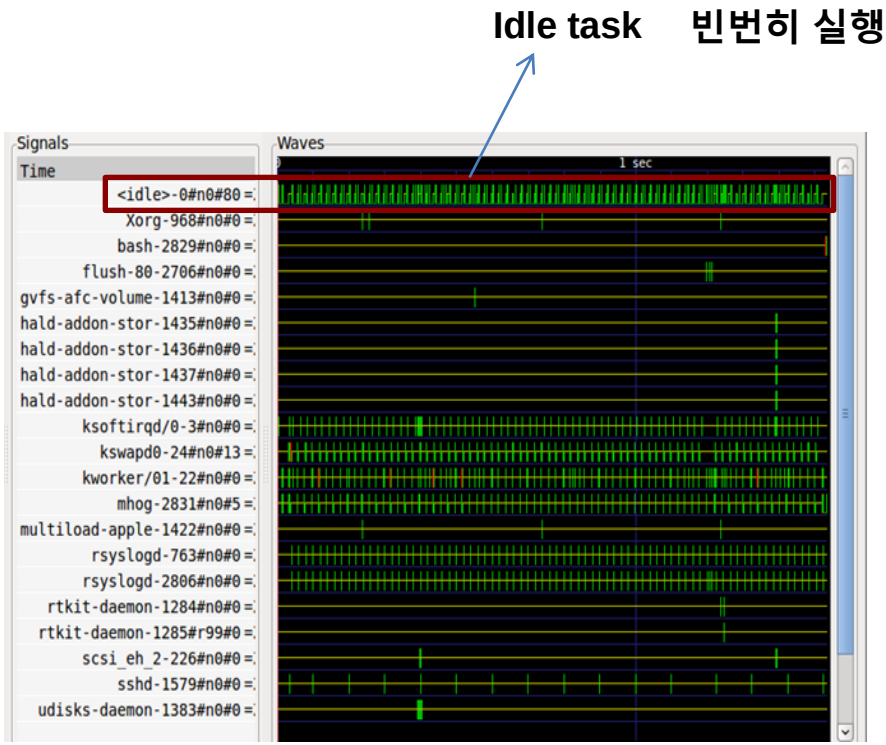


Congestion Control

- Direct reclaim path에서 page reclaim이 성공하지 못한 경우, device congestion 발생한 것으로 간주하여 일정 시간의 timeout (congestion wait)을 준 뒤 다시 retry함.
- 현 congestion wait timeout 값은 SCM과 같은 fast device에게는 불필요하게 큼.

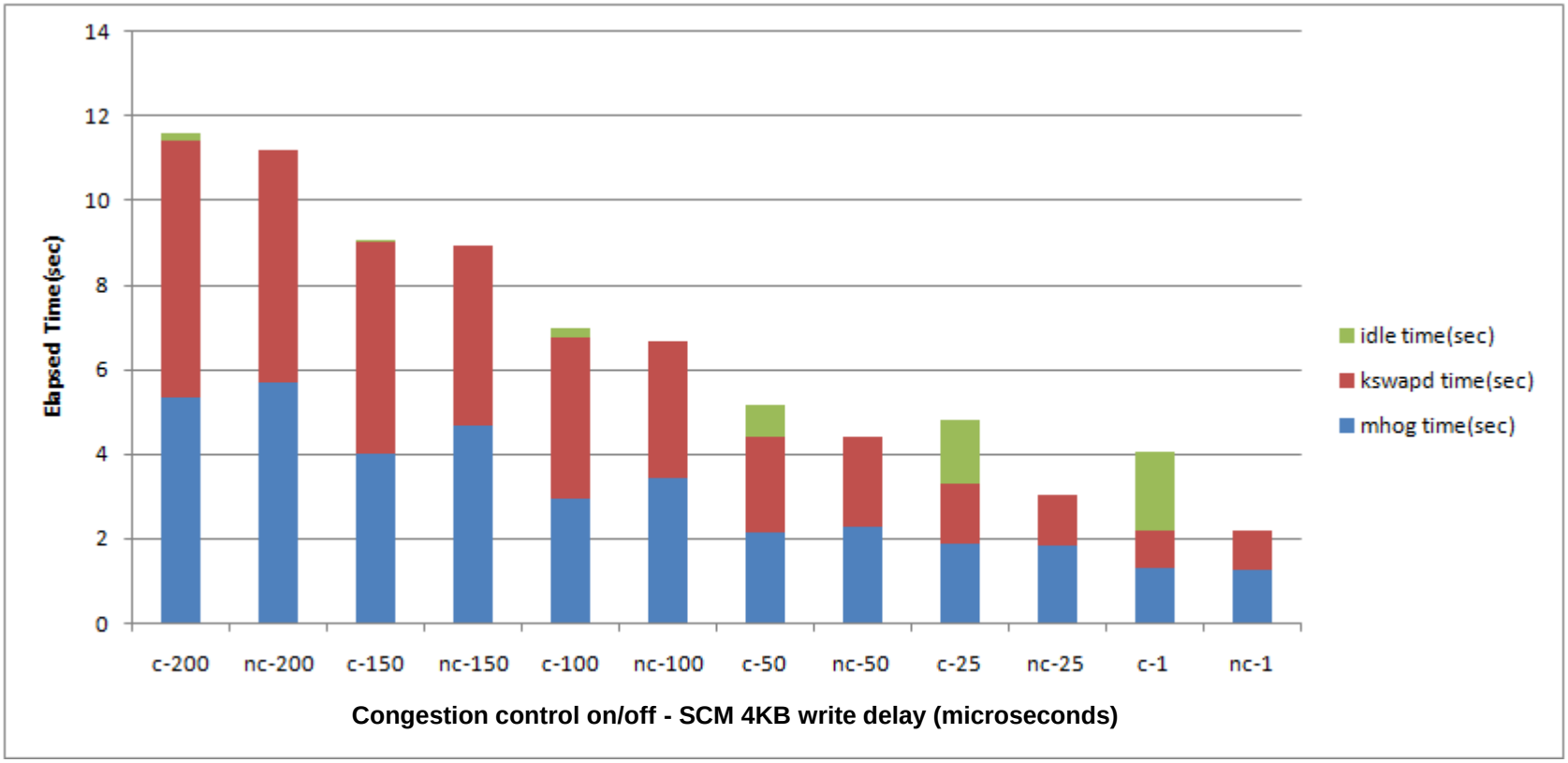


Direct reclaim procedure



Congestion Control (cont'd)

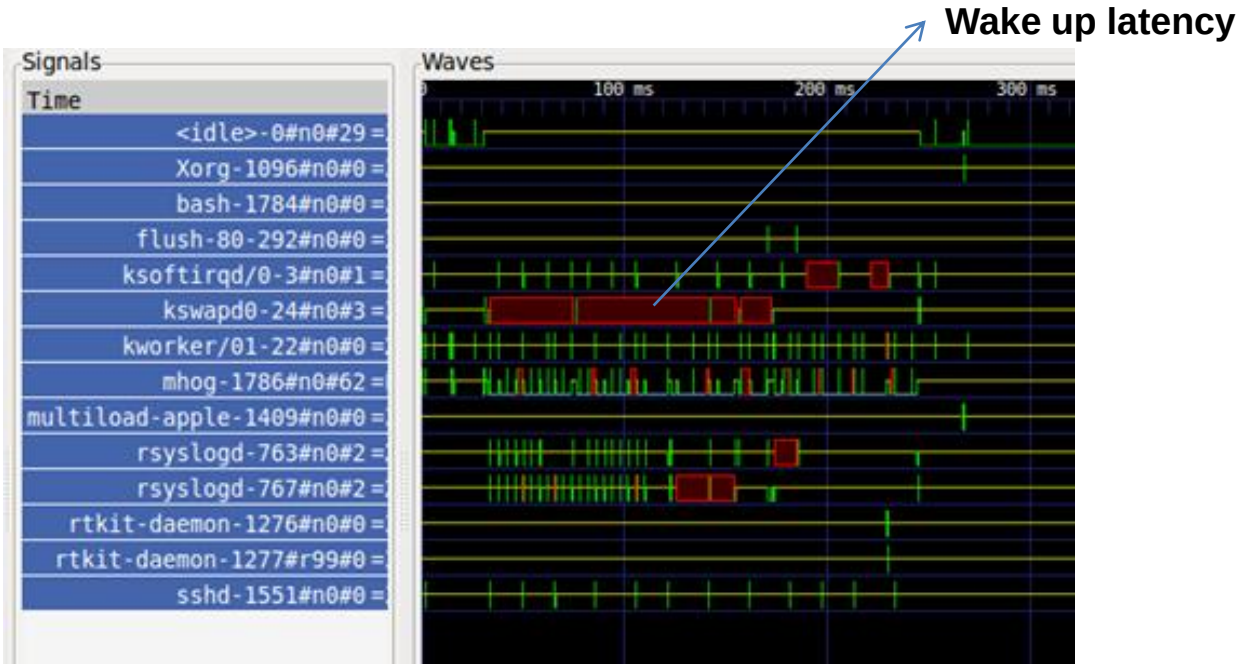
- Turning off congestion control ➔ idle time 감소, swap out 시간 단축



nc: congestion control is off.
c: congestion control is on.

Congestion Control (cont'd)

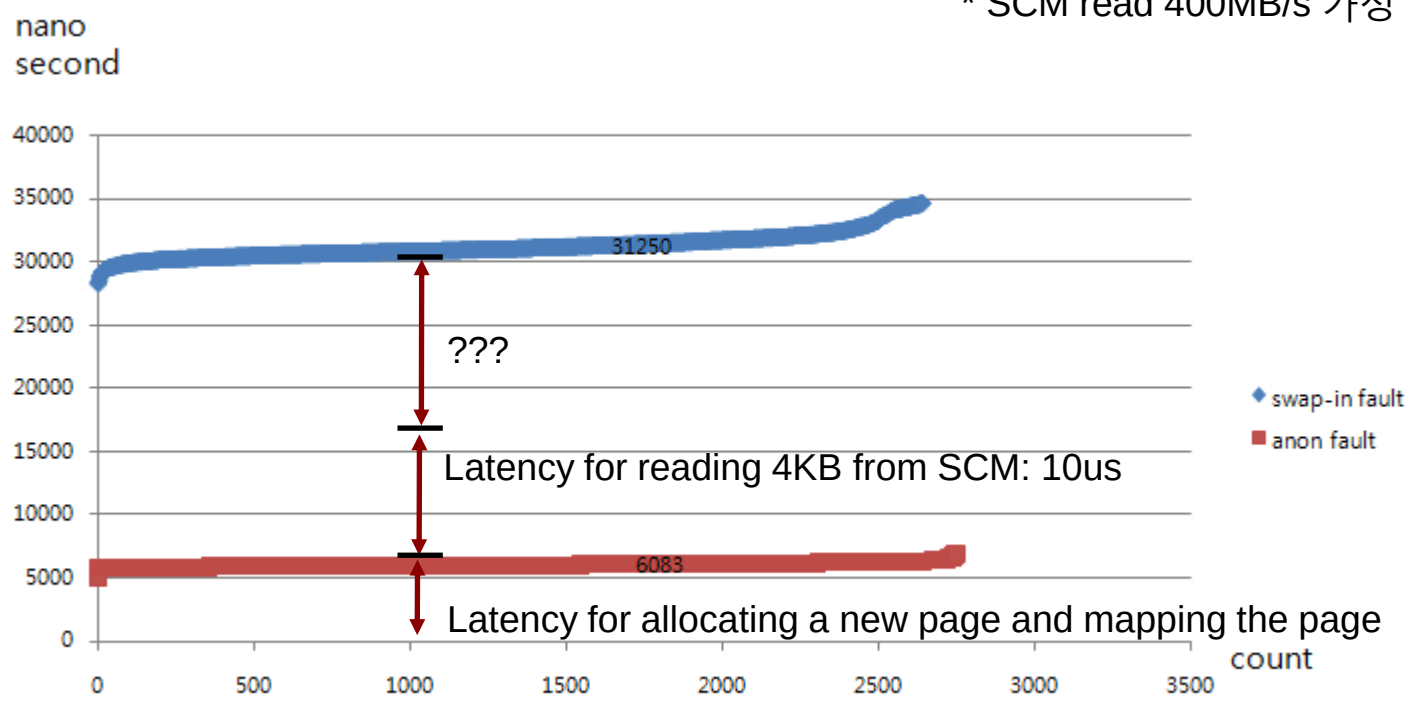
- Congestion control을 off하는 경우 side-effects
  - Swap out efficiency 저하 문제: small random writes to swap space
    - Kswapd는 bulk 단위로 swap out,
    - Direct reclaim은 긴급히 적은 양의 free memory 확보하는 데 중점.
  - Device congestion 상황에서 시스템 응답성 저하



Swap In Overhead

- Swap in fault handling latency: 약 31us
  - Cf. minor page fault handling latency는 약 6 us: new page 할당 및 mapping

* SCM read 400MB/s 가정



Swap In Overhead Analysis (cont'd)

- Data transfer 시간 (SCM read + dcache flush)을 제외한 OS overhead는 53.2%.

Function	Latency (us)	Ratio (%)
lookup swap cache	0.7	2.3
swap page read	12	38.7
SCM read	10	32.3
swap page reuse	2.2	7.0
page mapping	0.4	1.3
swap slot free	1.2	3.9
update MMU cache	4.5	14.5
total	31	100

- **Swap layer: 53%**
  - Swap cache searching
  - Allocate a new page
  - Swap slot management
  - LRU page management
  - Etc.
- **Block layer: 31%**

ARM CPU cache aliasing avoidance를 위한 dcache flushing

Issues

- OS overheads for SCM paging device
- SCM vs. NAND as a paging device

Swap Workload Generation

- System configuration
  - Memory:1GB, Swap: 1GB
  - I/O scheduler: CFQ
  - SOLID_STATE MODE: forward scan
  - Aging time: about 20 hours
- Simulated a typical PC usage scenario:

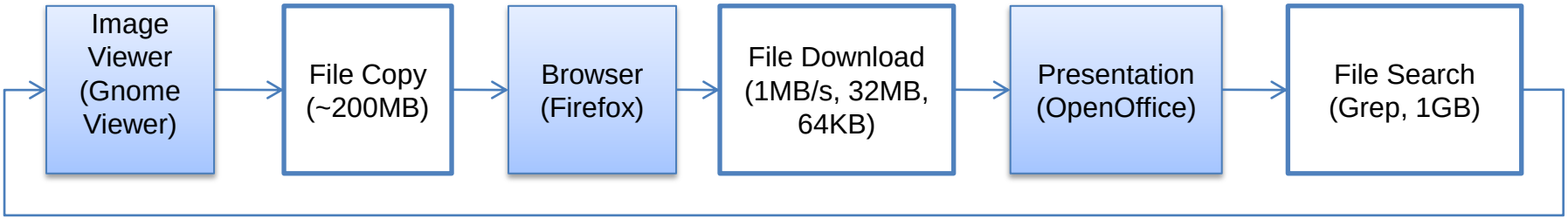
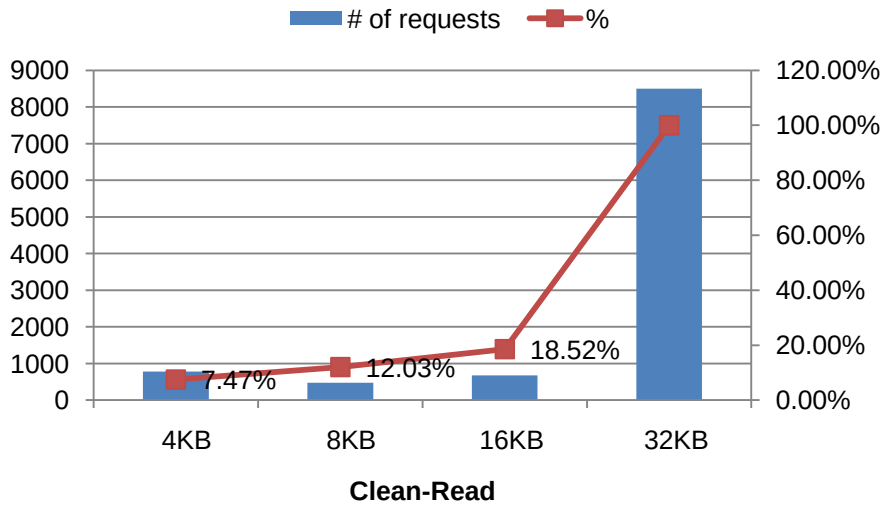
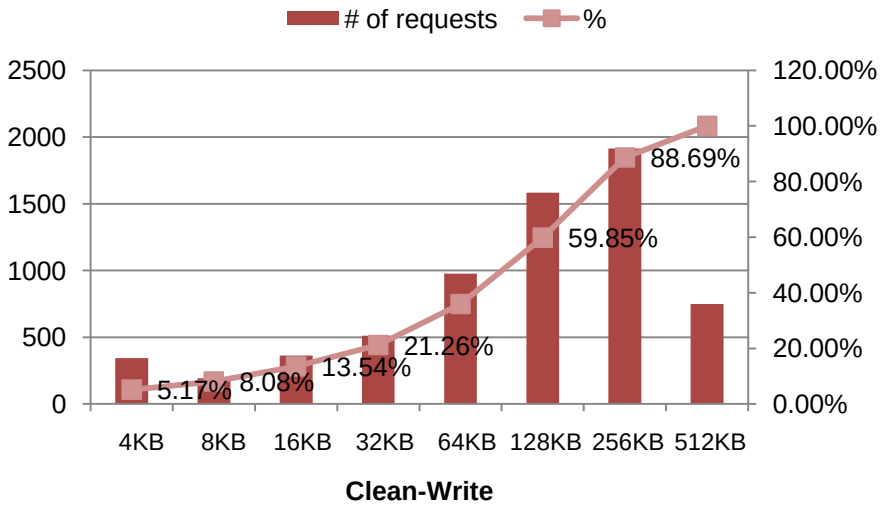


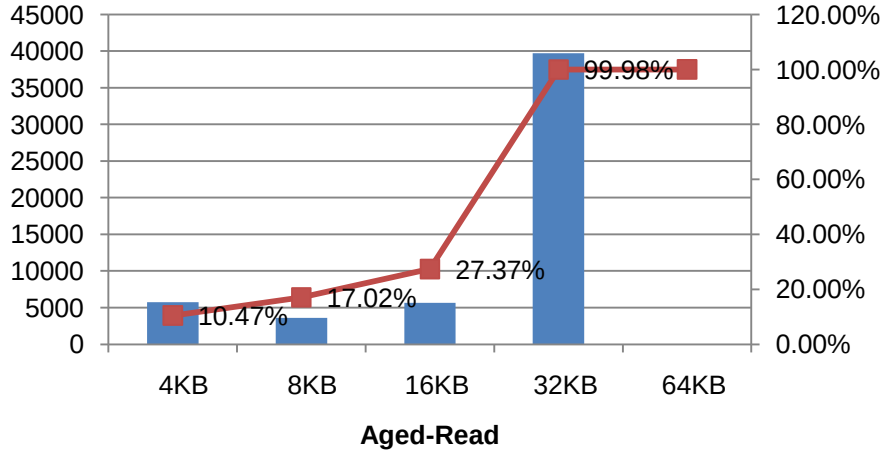
image viewer : 10MB * 7
File Copy : 160MB
Brower page multi-tab 8ea open : u-tube, twitter , damn. CNN (Browsing + scroll)
File Download : 1MB/s chunk size :64KB , 32MB – File Download.
Presentation – Sun.(Linux) open office file (3.2MB PPT) , PAGE ..PPT Mode.. Texting. ..
File search – 1.4GB

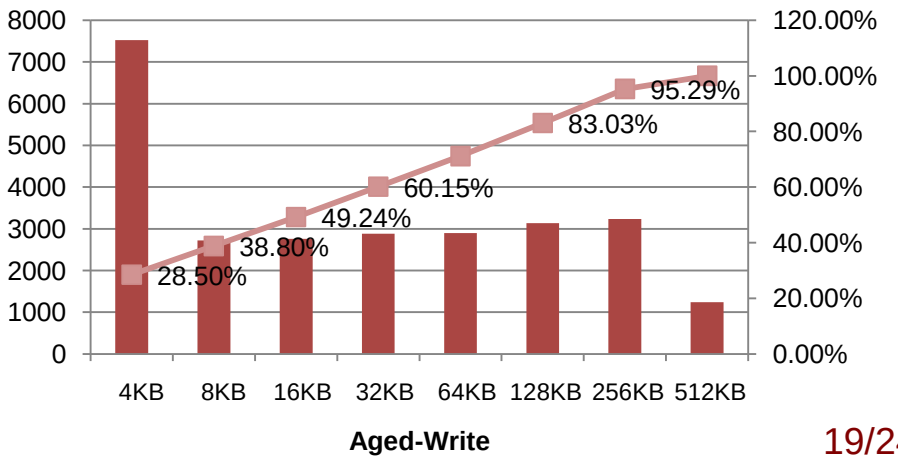
Swap I/O Workload Pattern

- Request size distribution (NOOP scheduler 사용 시)
  - Read의 경우 aging에 관계 없이 32KB가 많음 (read-ahead 동작)
  - Write size는 clean 상태에서 128, 256KB가 많지만, aged 상태에서는 4KB가 대다수임.



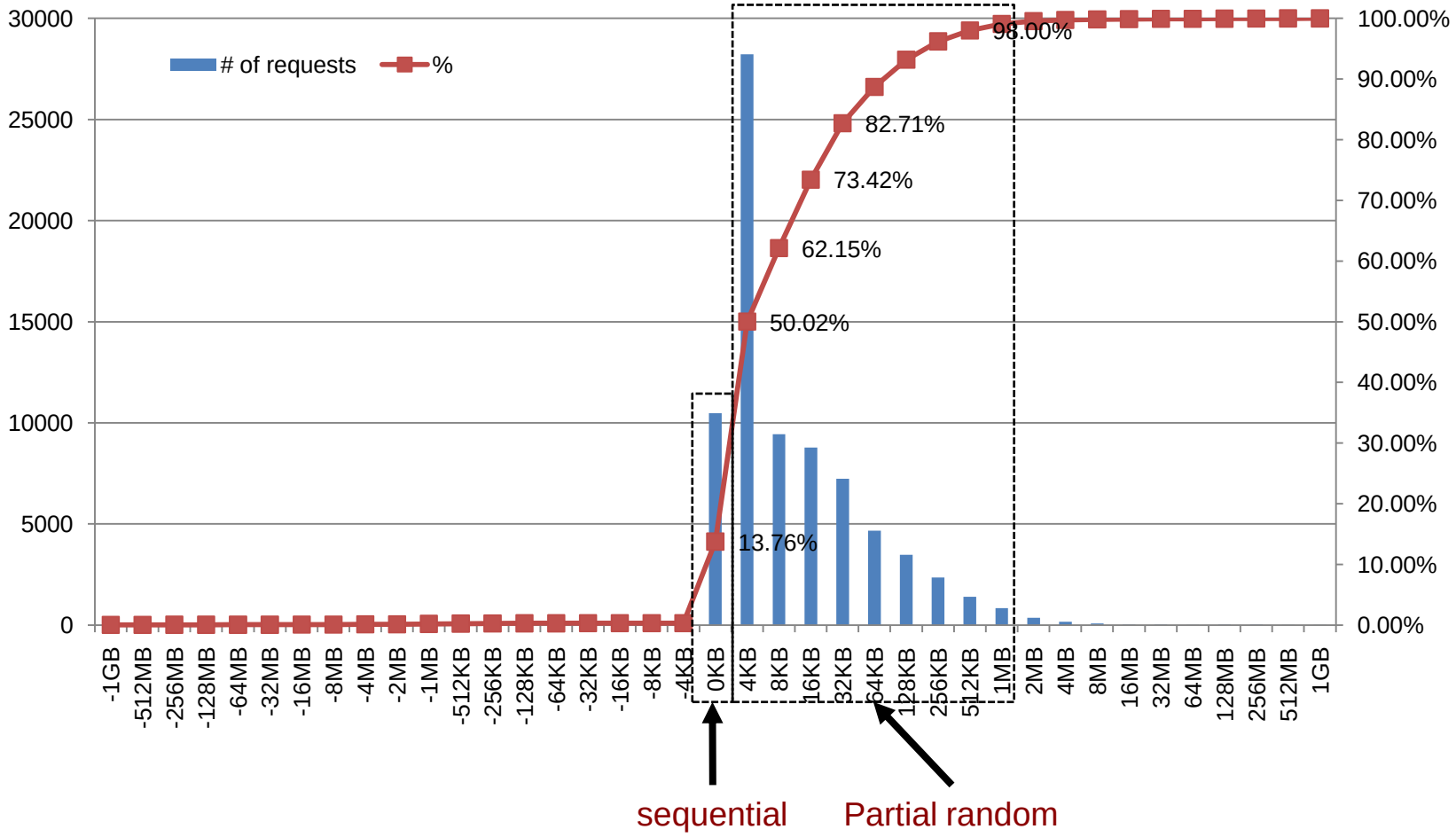






Swap I/O Workload Pattern (cont'd)

- Inter-request distance (IRD) distribution



Performance Simulation Method

- Workload pattern에 따른 NAND 성능 특성 고려
  - Sequential workload (IRD = 0): merge 없음.
  - Full random workload (IRD >= block size): each request마다 merge cost 추가
  - Partial random workload (IRD < block size): each N request마다 merge cost 추가

Parameter	SCM-133	SCM-400
Read Speed	133MB/s	400MB/s
Write Speed	40MB/s	40MB/s

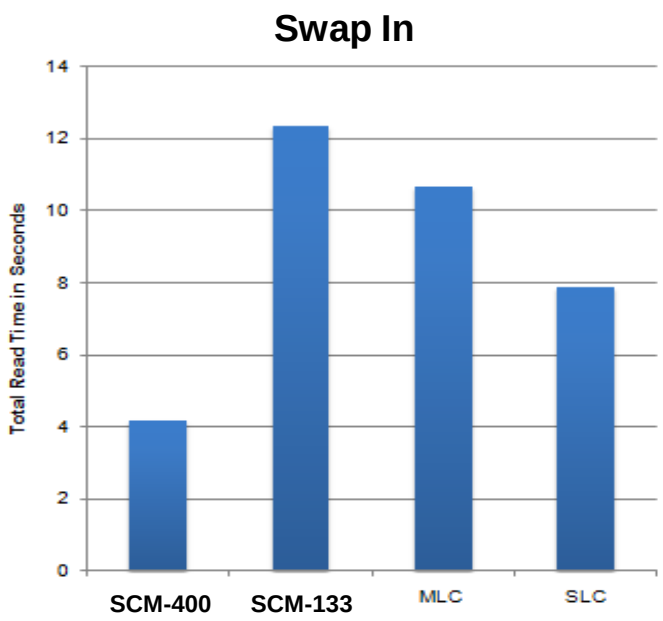
SCM parameters

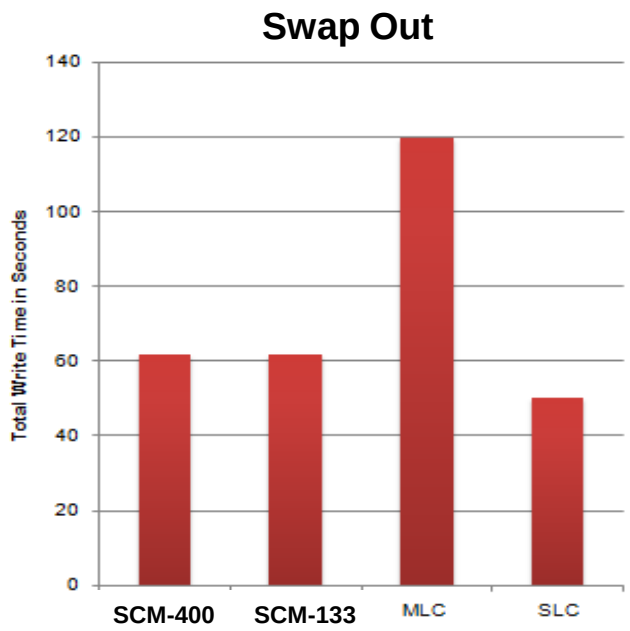
Parameter	SLC Flash	MLC Flash
Read Latency (tR)	50 us	80us
Program Time (tProg)	500us	1300us
Pages Per Block	64	128
Block Merge Time	43000us	187000us
Page Size	8KB	
Planes Per Chip	4	
Page Transfer Time (tTrans)	20us (40MHz)	

NAND parameters

Performance Simulation Results

- Swap In
  - Read size는 read-ahead (8개 page를 prefetch)로 인해 32KB의 비중이 큼
  - SCM-400이 SLC/MLC NAND에 비해 우세한 성능을 보임.
    (SCM-133은 SLC/MLC NAND 대비 열세)
- Swap Out
  - Aged 환경에서 workload pattern: small size 및 random write request 비중이 높음.
  - 10K IOPS write 성능의 SCM이 MLC NAND보다 성능 우세함. (SLC NAND 대비 다소 열세)





Lifetime Estimation

- Test time: 2hr
- Write amount: 880MB
- # total writes: 215,022
- # written pages: 163,177
- # writes per page: max 4, avg. 1.32

	SCM	NAND	
		SLC	MLC
Endurance	10^6	100k	10k
Life time (w/o WL)	**500,000 (h)= 57.1 (y)**	**5.7 (y)**	**0.57 (y)**
Life time (w/ WL)	1000,000 (h)= 11.4 (y)	11.4 (y)	1.14 (y)

Summary

- SCM paging device 사용을 위해서 OS overhead 개선 필요
  - SCM과 같은 고속의 메모리를 전통적인 HDD I/O 처리 방식으로 사용시 효과적이지 않음.
    - e.g. congestion control, swap subsystem overhead.

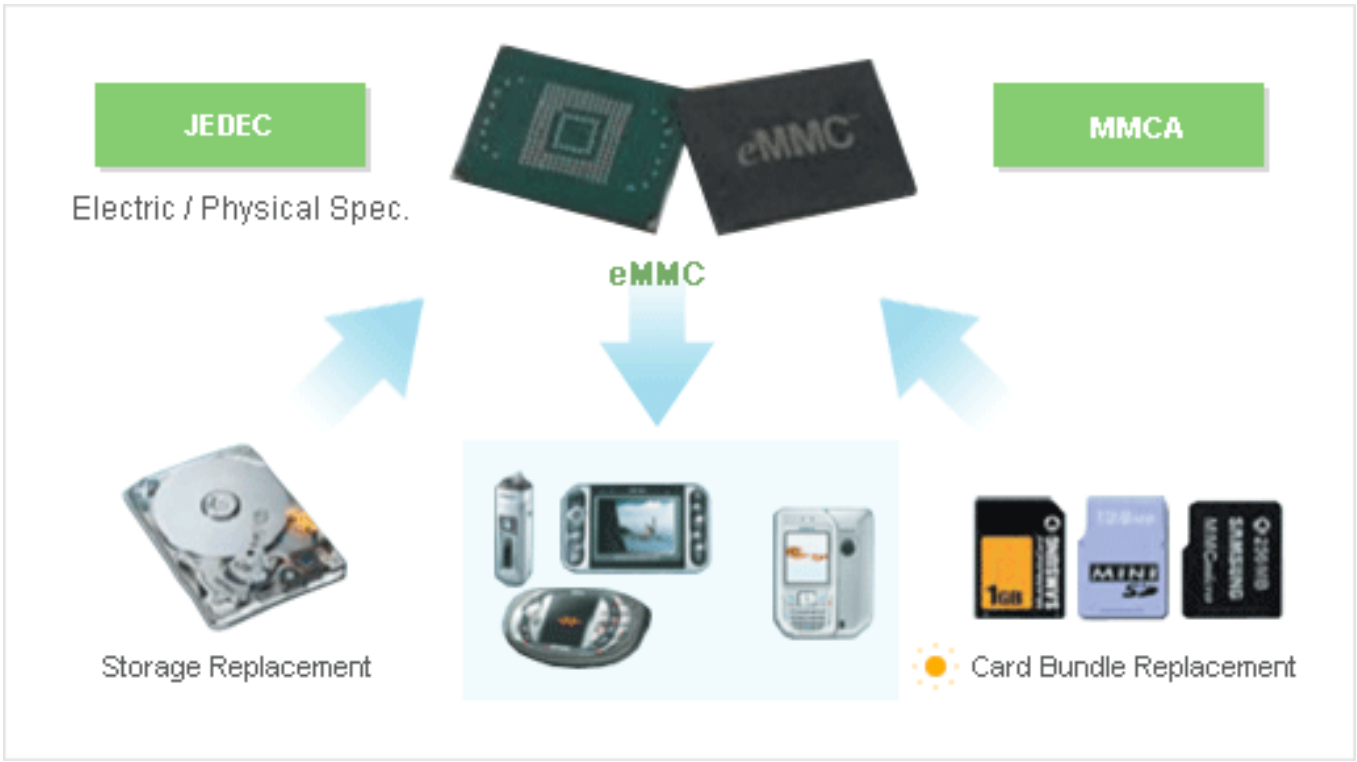
- Paging device: SCM or NAND
  - SCM이 NAND에 비해 다소 유리해 보임
  - 비용과 전력 소모 추가 고려가 필요

- Things to do
  - smart phone/tablet user scenarios에서의 paging workload 패턴 조사
  - User interaction latency 평가 필요
  - Prototyping
    - SCM emulation
    - Swap on real NAND storages (eMMC, SD card, …)

Appendix

Samsung eMMC

Samsung eMMC



- 32GB eMMC 4.3: CS available, Max 52MHz, x8 width.
- 16GB eMMC 4.3: MP, Max 52MHz, x8 width.

NcPRAM

Samsung PRAM

		*NcPRAM	NOR
Interface		Mux	Mux / Demux
Available Density		512 Mb	512 / 256 / 128 / 64 Mb
Package size		8mm x 9.2mm	Same
PKG type		56FBGA	Same
Voltage		1.8V	Same
Endurance / Retention		100K / 10Years	100K / 1Year or 10K / 10Years
Performance	Read(max.)	100ns/word	Same
	Program(typ.)	3us/word	80us/word
	Block Erase(typ.)	80ms/block	600ms/block