

---

# 쓰기 참조 특성을 고려하는 플래시메모리 기반 시스템의 가상메모리 페이지 회수 기법

---



이화여자대학교  
반효경

## 연구의 구성

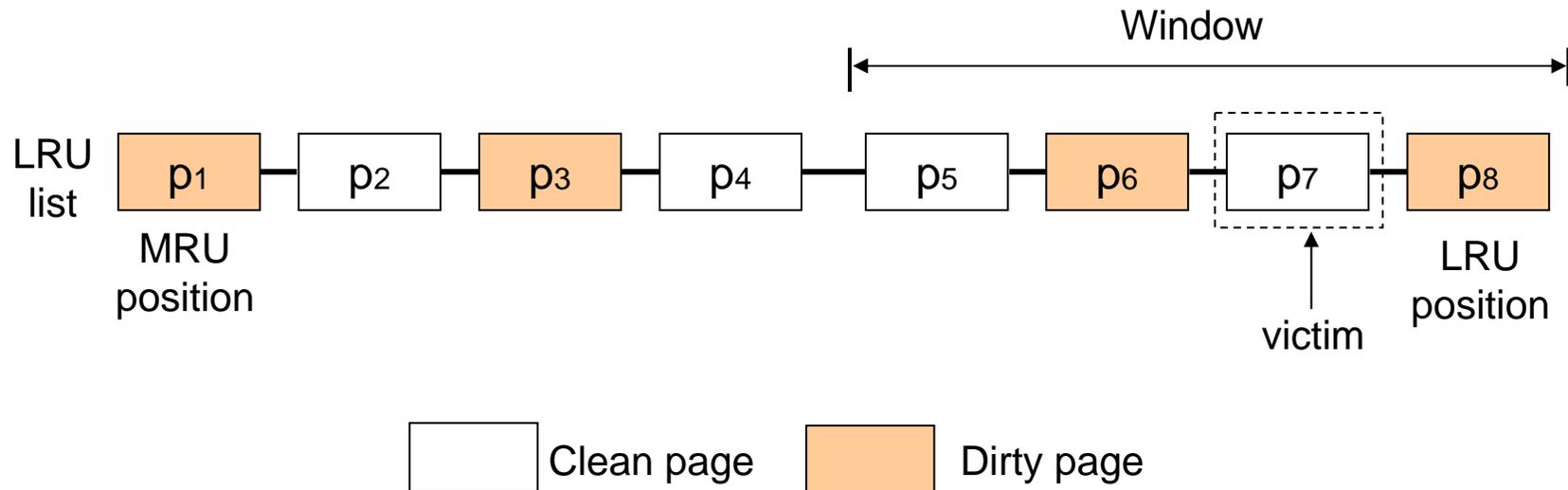
- 1) 가상메모리의 쓰기 참조 특성 고려
- 2) 압축 파일 시스템에서의 인출 비용 고려

## 가상메모리의 페이지 참조

- “최근에 참조된 페이지가 가까운 미래에 다시 참조될 가능성이 높다”
  - 시간지역성(temporal locality)
- LRU
  - 가상메모리 시스템에서 그대로 사용될 수 없음
  - 메모리 참조시 마다 리스트 조작
- Clock
  - 메모리 직접 참조 시에는 하드웨어적 비트 세팅
    - access bit, dirty bit
  - 메모리 부재 시 커널의 리스트 조작

# CFLRU (Clean-First LRU)

- 플래시메모리의 연산 비용
  - 읽기 << 쓰기



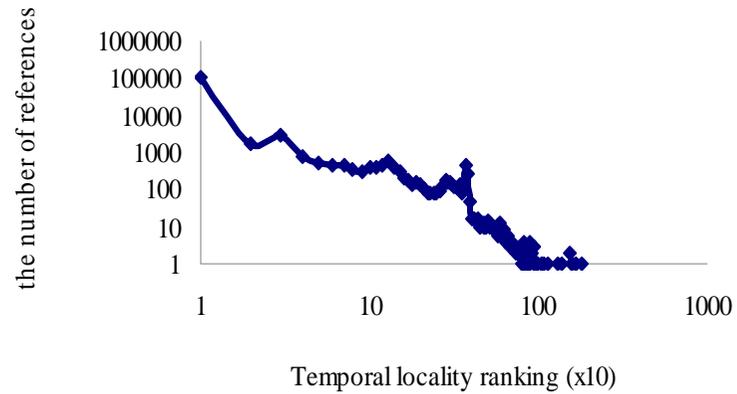
- 윈도우 내부: clean page를 우선 회수
- 윈도우 외부: 단순 LRU

## 기존 연구의 한계

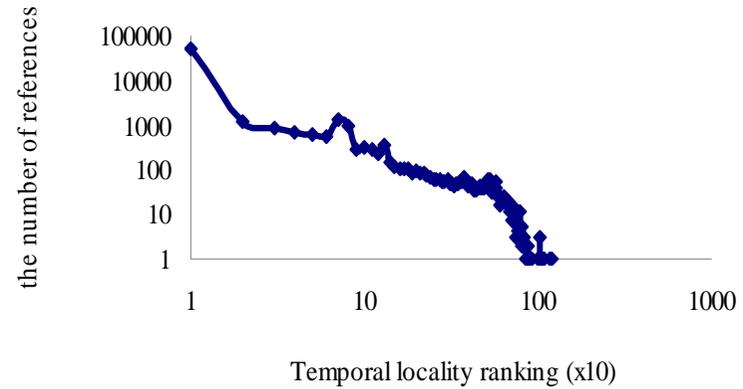
- 과거의 쓰기 기록을 정교하게 활용하지 않음
  - 쓰기 시점, 쓰기 빈도
- 읽기/쓰기 요청 비율 및 패턴 변화에 적응적이지 않음
- 리스트 조작 오버헤드

# 페이지 참조의 시간 지역성

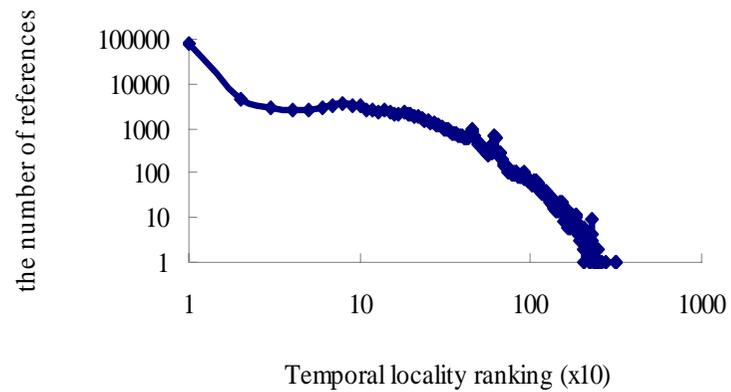
(a) xmms



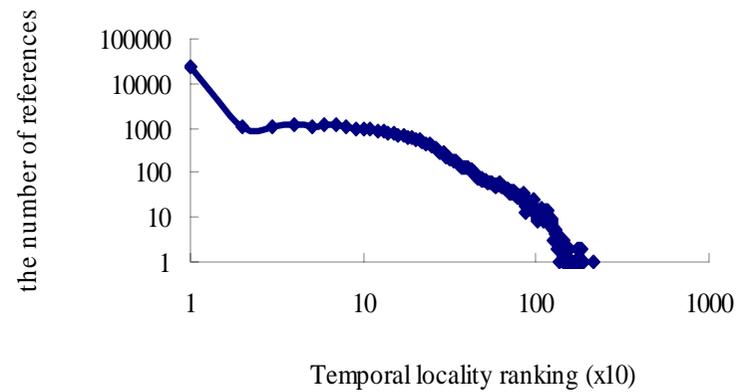
(b) gqview



(c) gedit

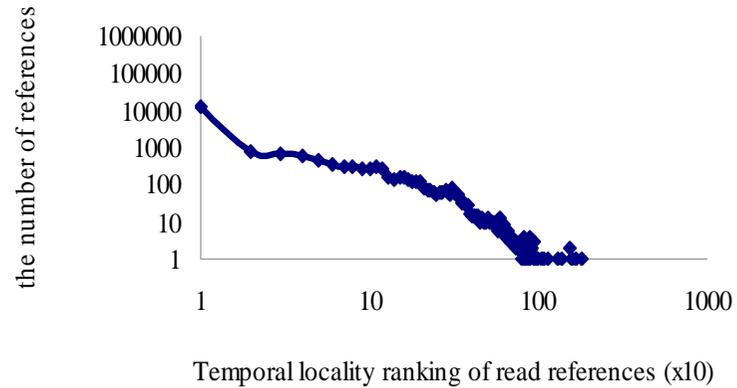


(d) freecell

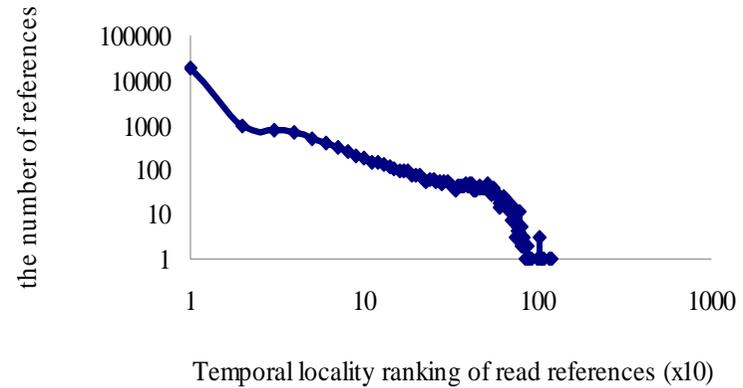


# 읽기 참조의 시간지역성

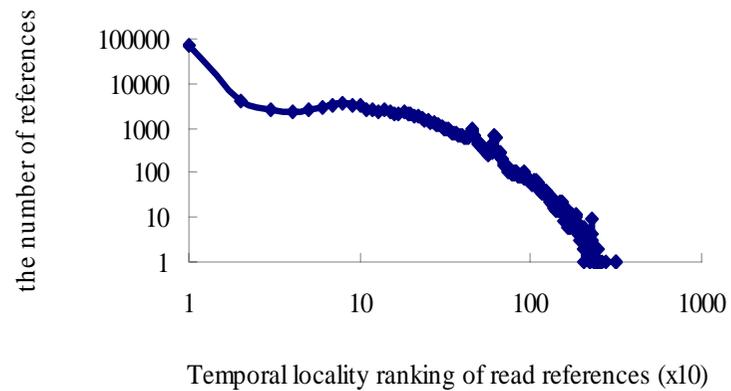
(a) xmms



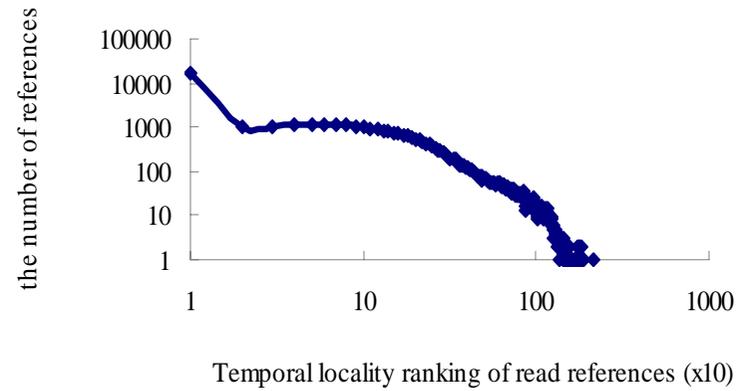
(b) gqview



(c) gedit

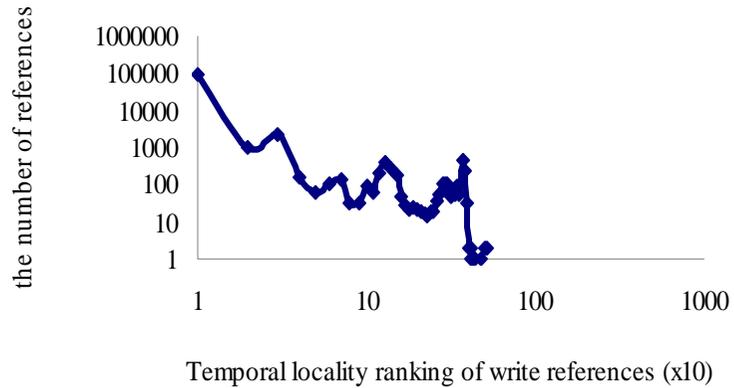


(d) freecell

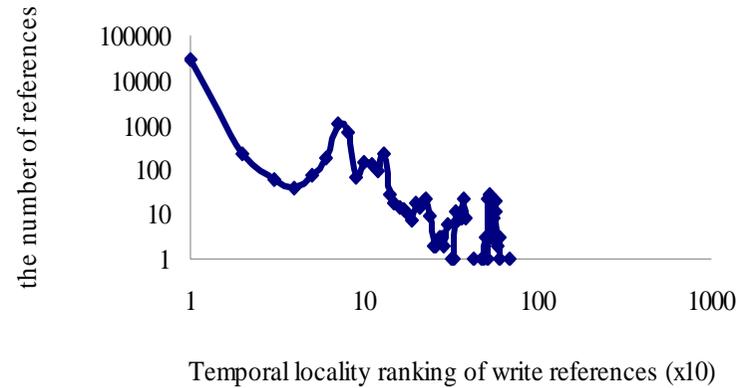


# 쓰기 참조의 시간지역성

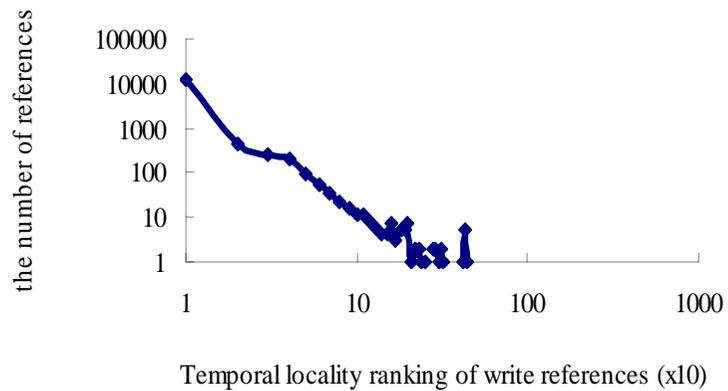
(a) xmms



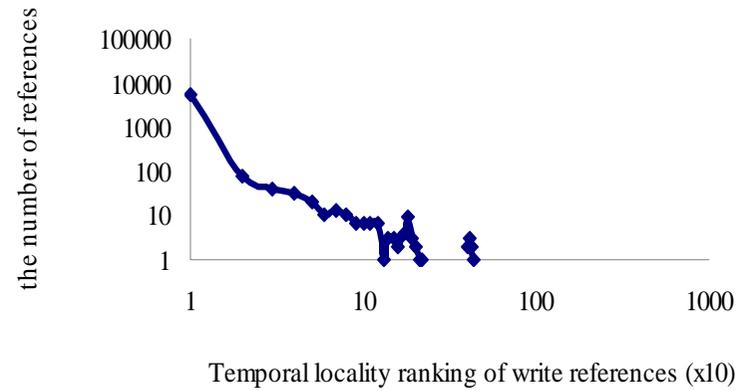
(b) gqview



(c) gedit

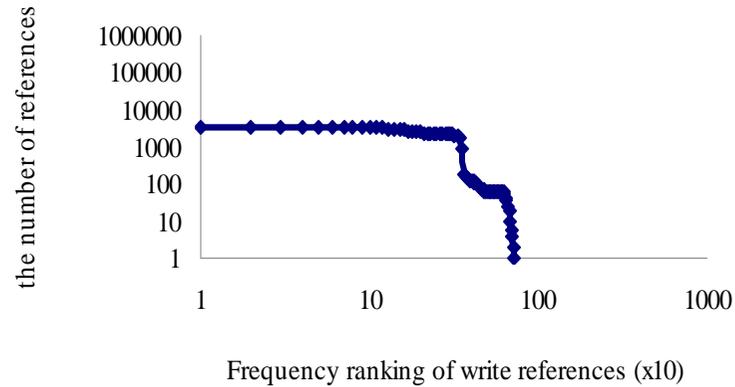


(d) freecell

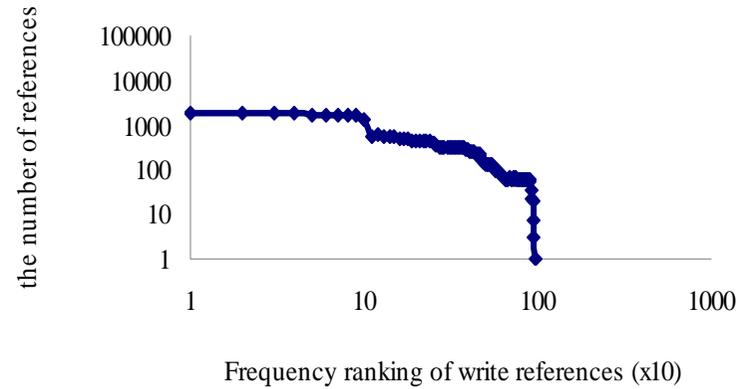


# 쓰기 참조의 빈도

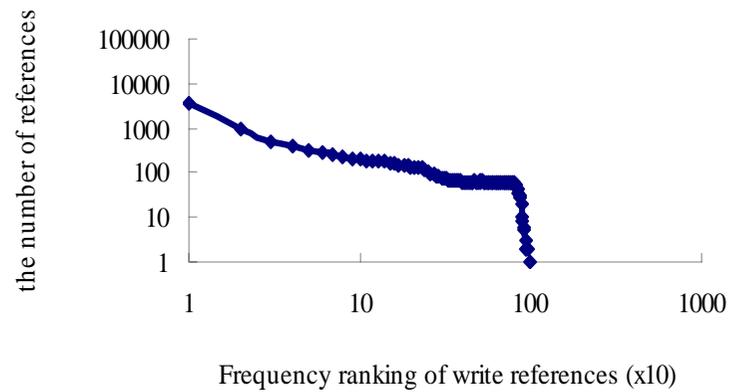
(a) xmms



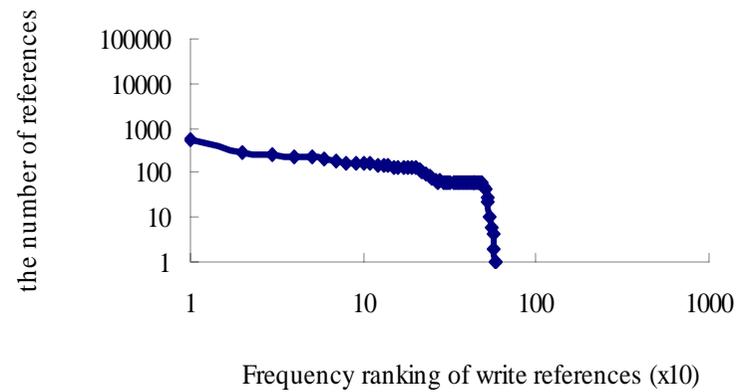
(b) gqview



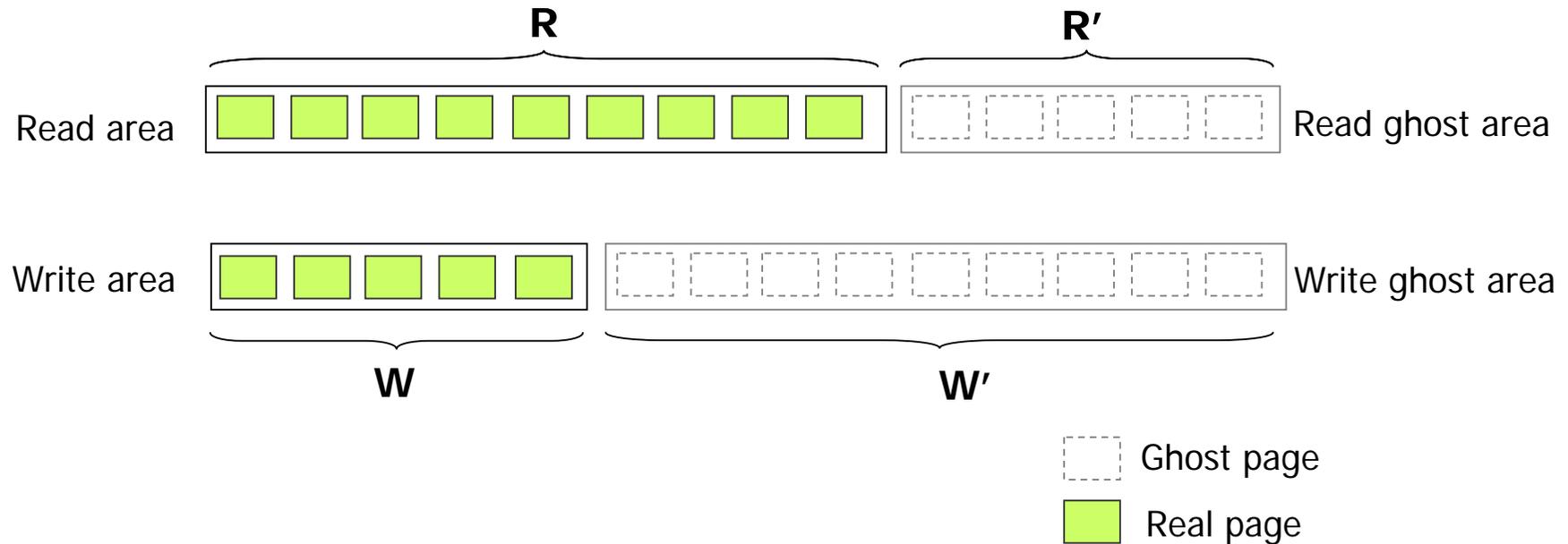
(c) gedit



(d) freecell

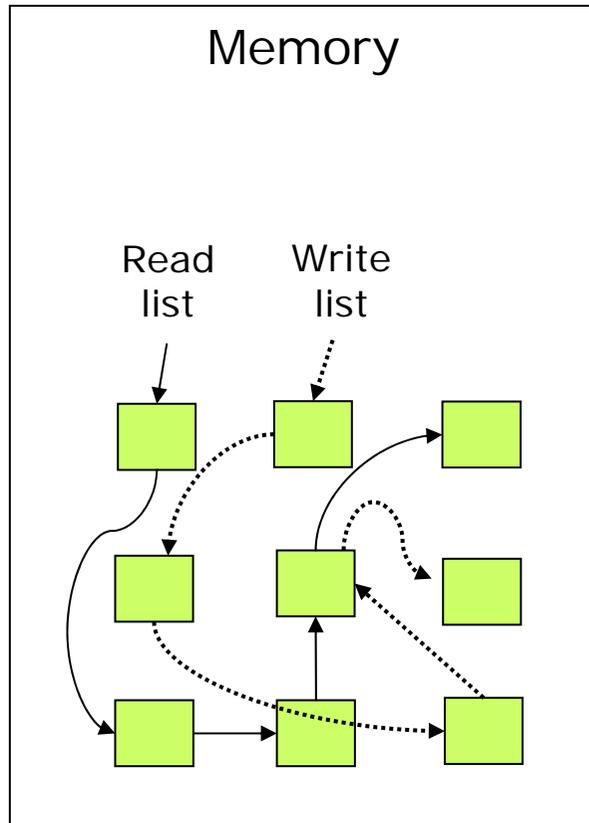


# 읽기 영역과 쓰기 영역



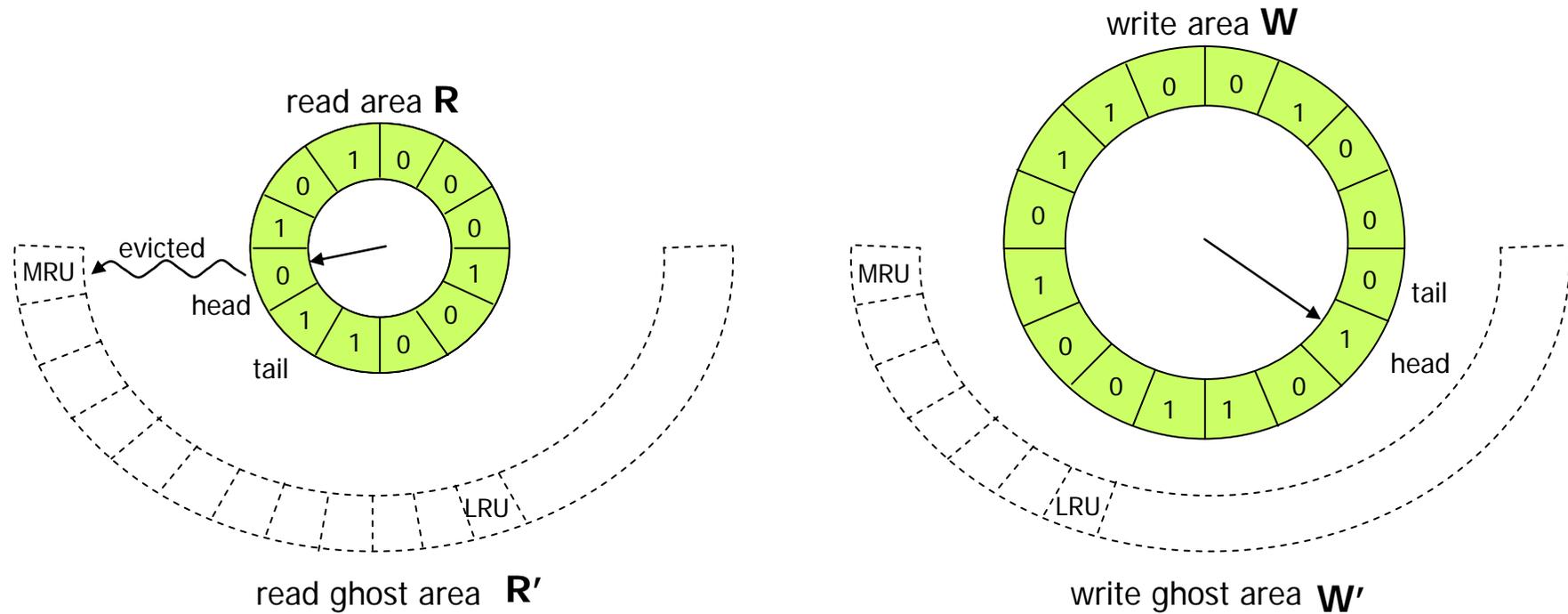
- 메모리 페이지를 읽기 영역과 쓰기 영역으로 관리
- I/O 성능 향상치에 근거해서 영역의 크기 동적 조절
  - 가상 영역에서의 페이지 참조
  - 읽기와 쓰기의 비용 차이 고려

# 읽기 영역과 쓰기 영역



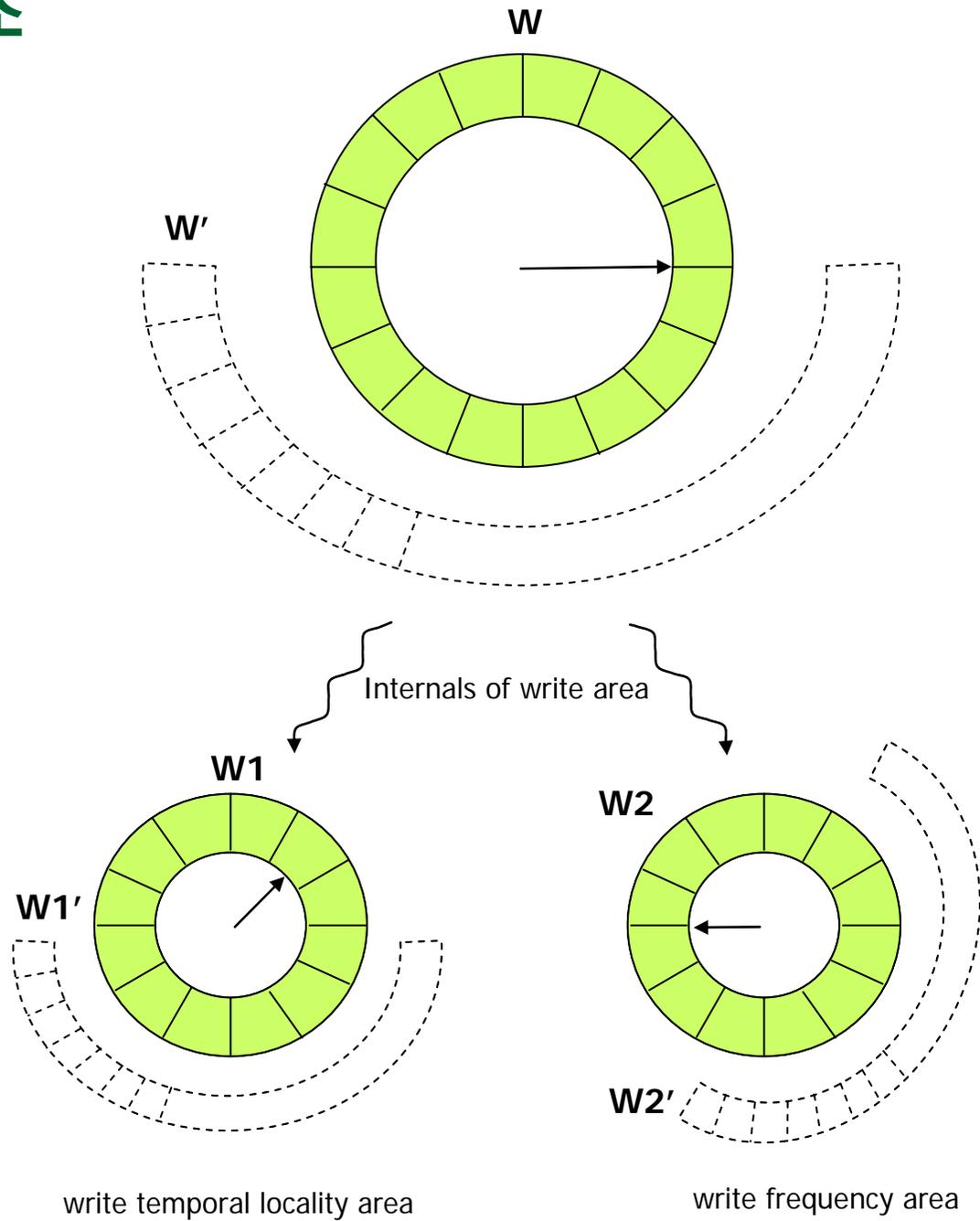
- 읽기 영역과 쓰기 영역의 중복 관리 가능
  - 두 영역에서 모두 회수된 경우 물리적 메모리에서 회수

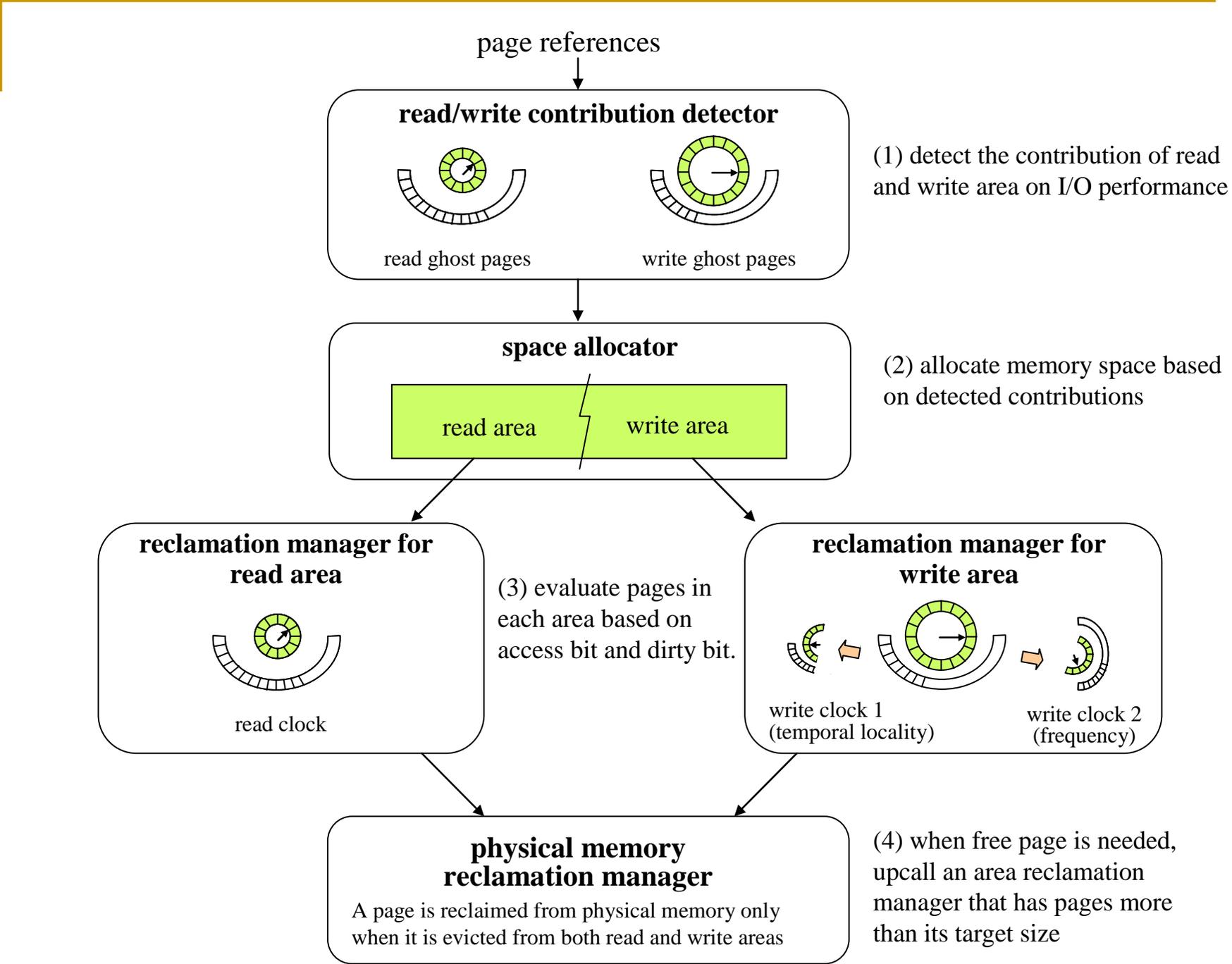
# 읽기 영역과 쓰기 영역



- 메모리에서 직접 참조 시 비트 세팅만 이루어짐
- 페이지 폴트 발생시
  - 각 영역의 목표 크기에 기반해 회수할 영역 결정
  - 회수 페이지는 해당 영역에서만 회수
  - 상대 영역의 비트가 세팅된 페이지의 리스트 조작 필요

# 쓰기 영역의 내부 구조



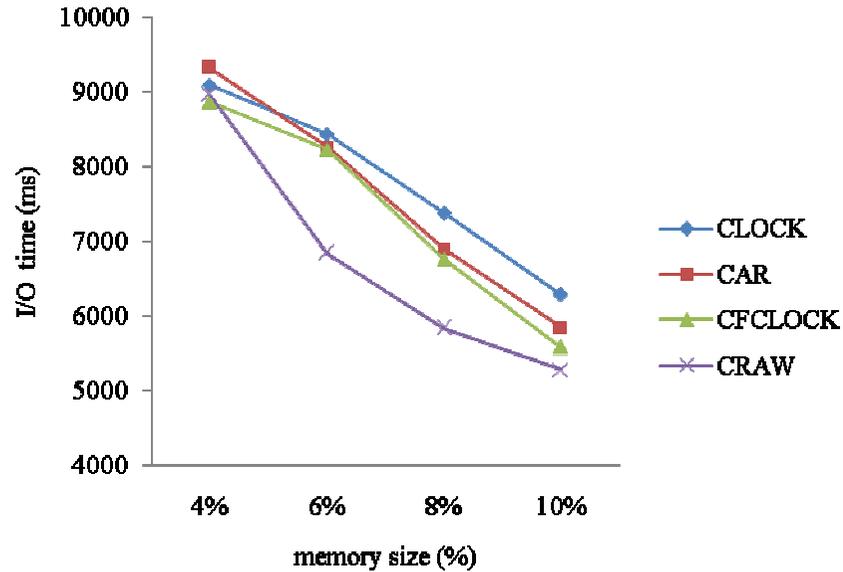


## 실험에 사용된 워크로드의 특성

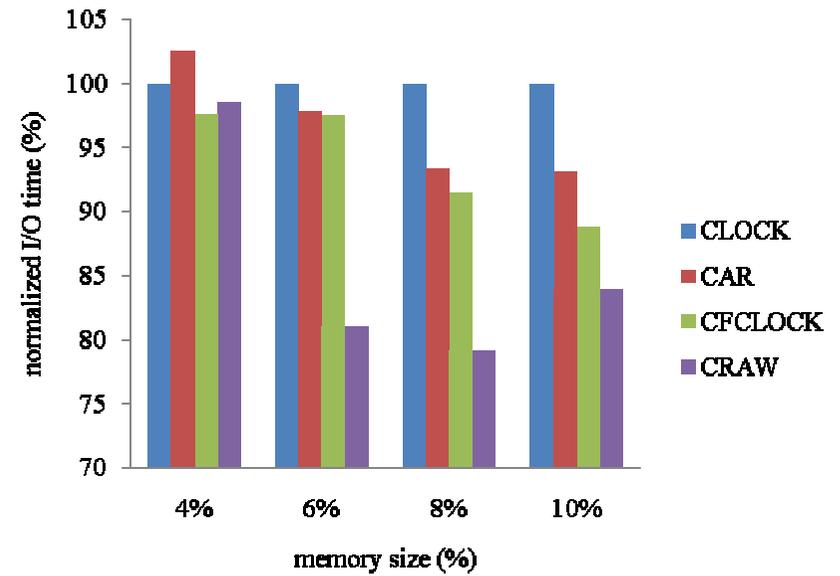
워크로드	메모리 사용량 (MB)	메모리 참조 횟수 워크로드메모리 사용량 (MB)			
		전체 횟수	명령 읽기	데이터 읽기	데이터 쓰기
xmms	8.05 MB	1,168,939	65,048	125,649	978,242
			읽기 : 쓰기 = 1 : 5.13		
gqview	7.43 MB	610,685	93,242	172,044	345,399
			읽기 : 쓰기 = 1 : 1.30		
gedit	14.46 MB	1,733,763	649,500	951,441	132,822
			읽기 : 쓰기 = 12.05 : 1		
freecell	10.08 MB	490,175	114,233	315,902	60,040
			읽기 : 쓰기 = 7.16 : 1		
kghostview	17.39 MB	1,546,135	380,609	1,061,986	103,540
			읽기 : 쓰기 = 13.93 : 1		

# xmms

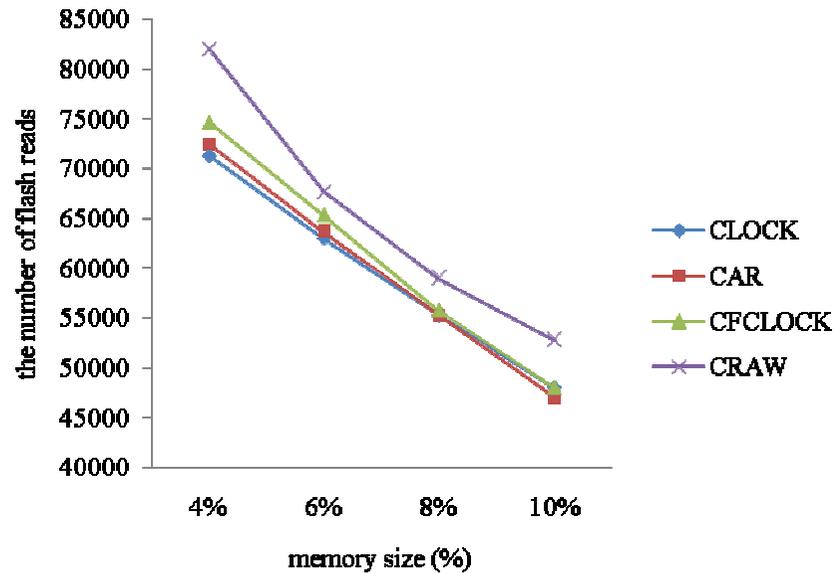
## I/O time



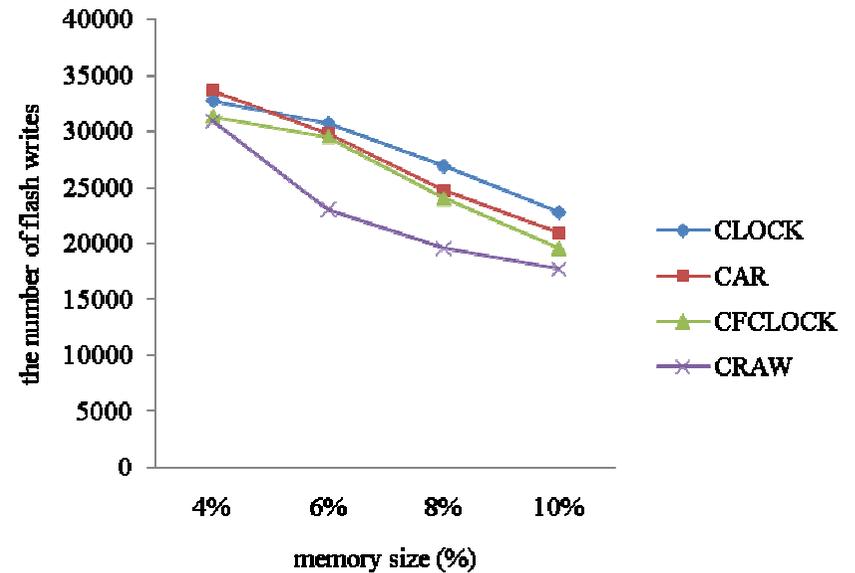
## normalized I/O time



## read operation

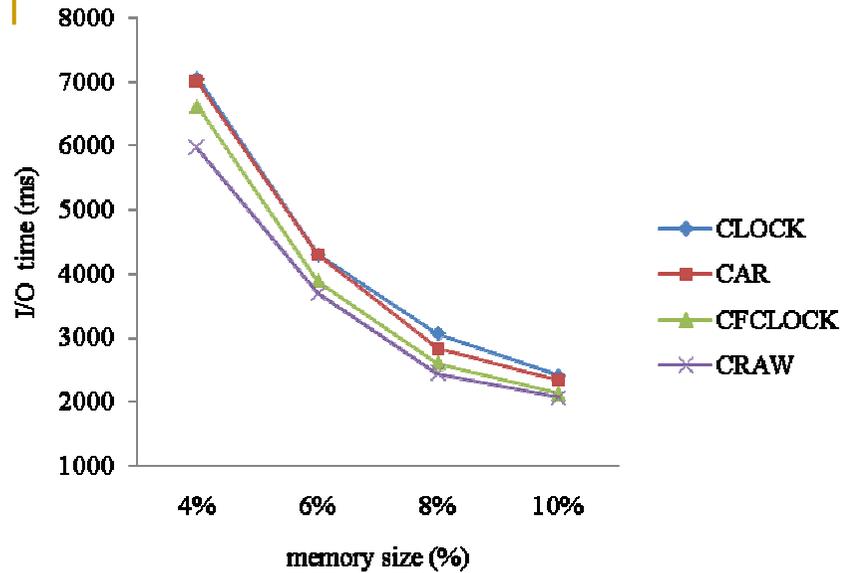


## write operation

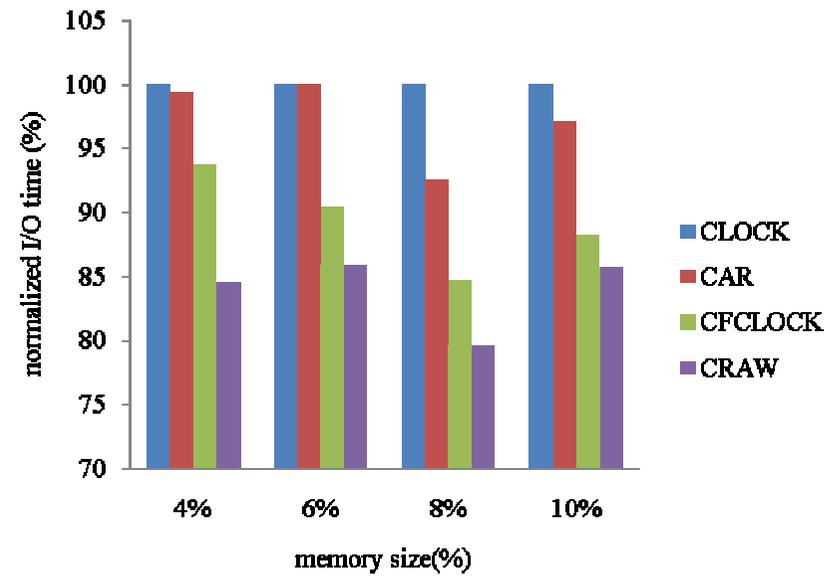


# gqview

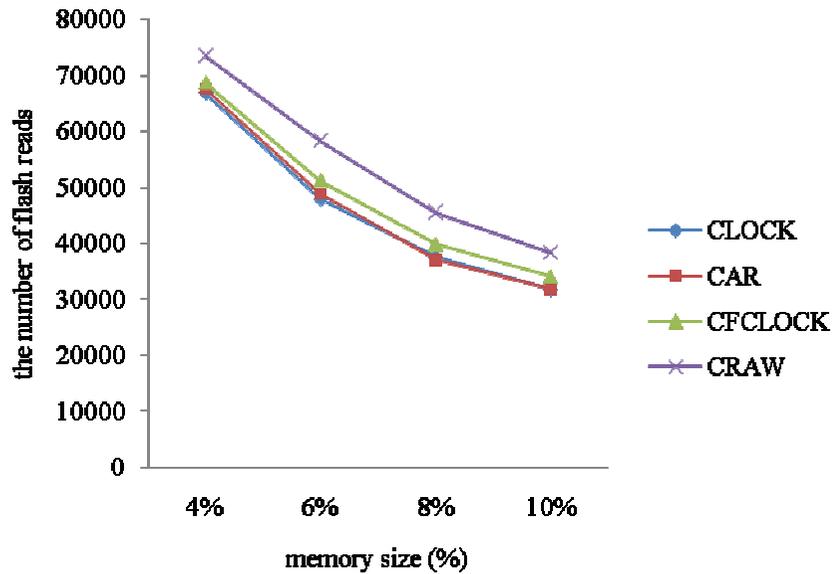
## I/O time



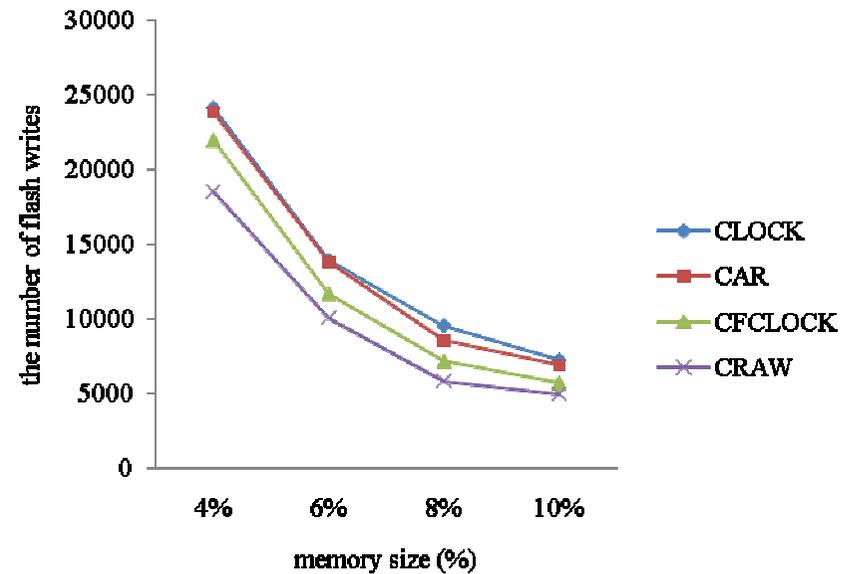
## normalized I/O time



## read operation

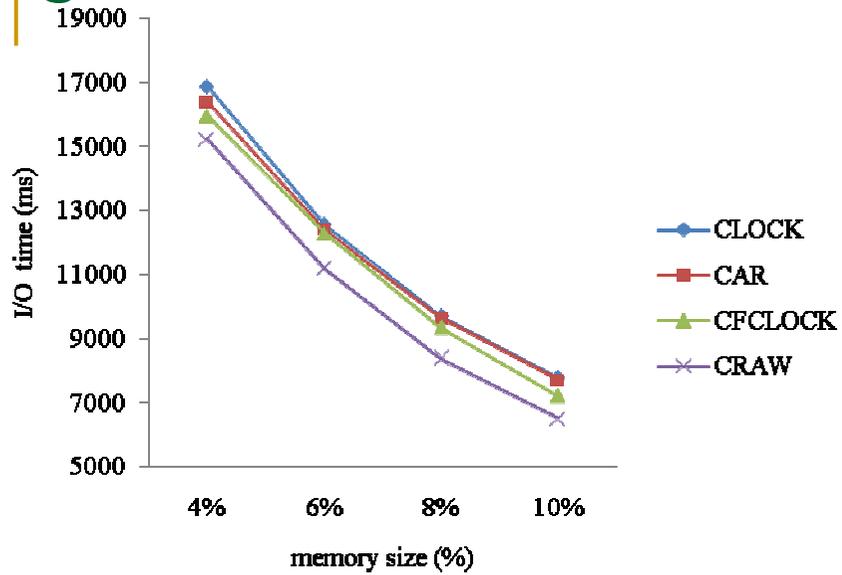


## write operation

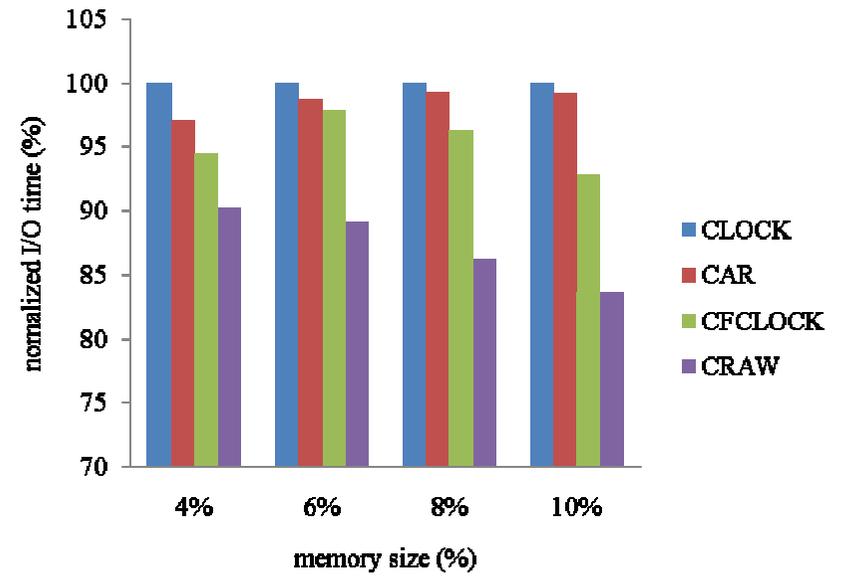


# gedit

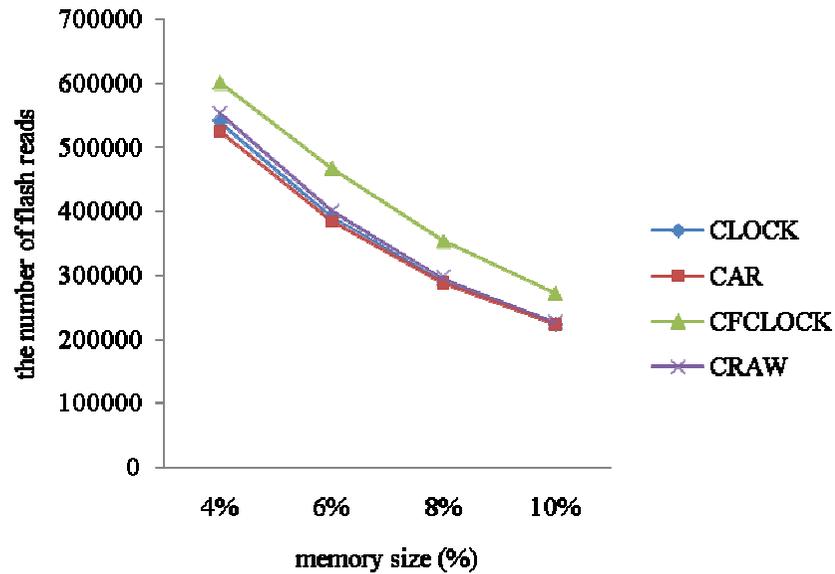
## I/O time



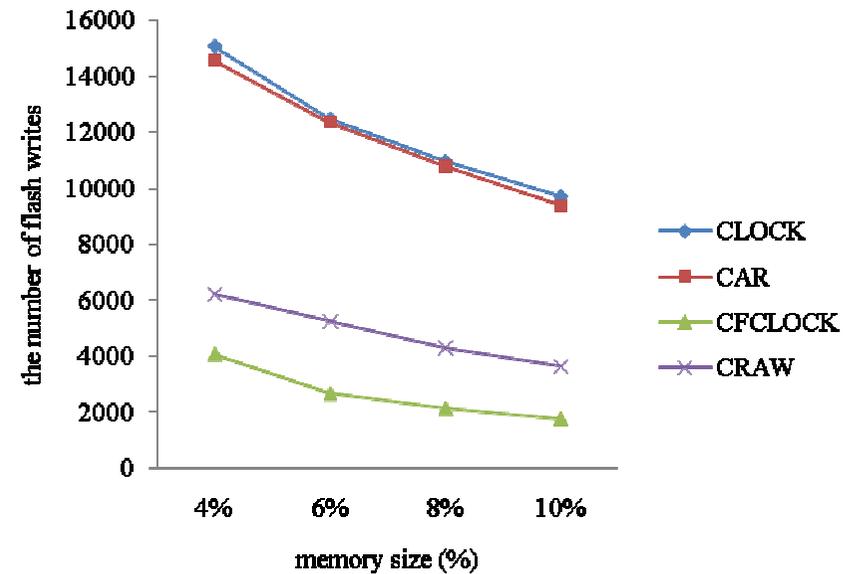
## normalized I/O time



## read operation

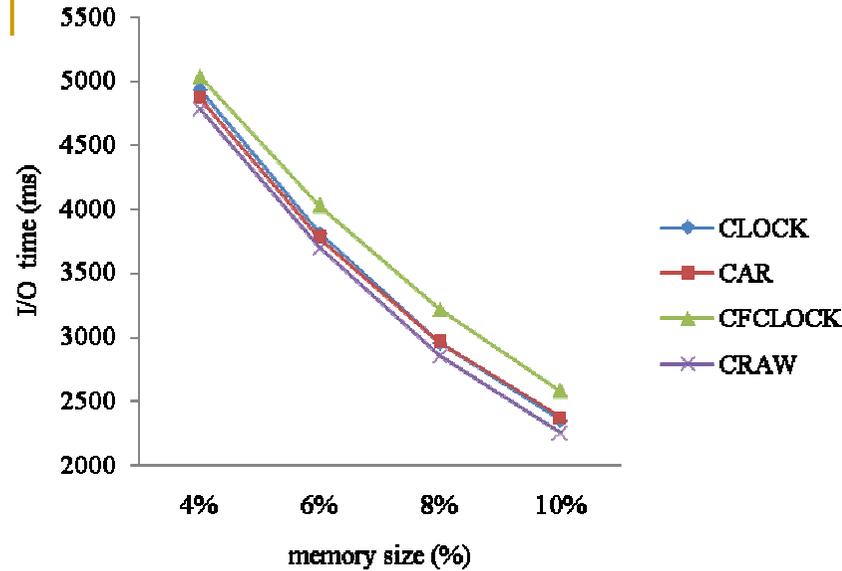


## write operation

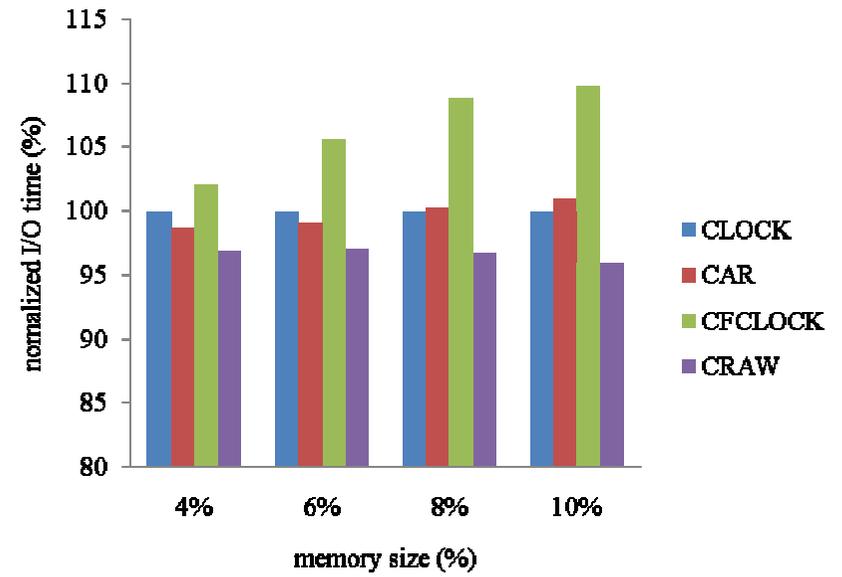


# freecell

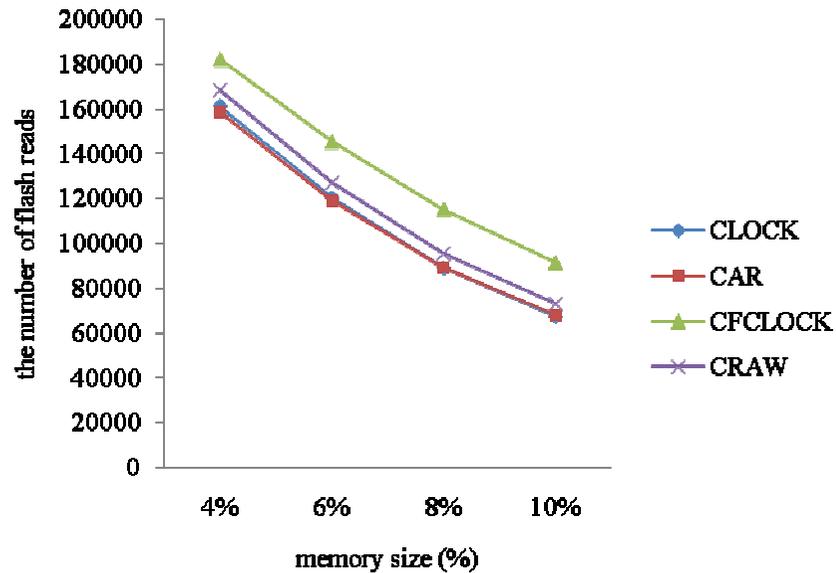
## I/O time



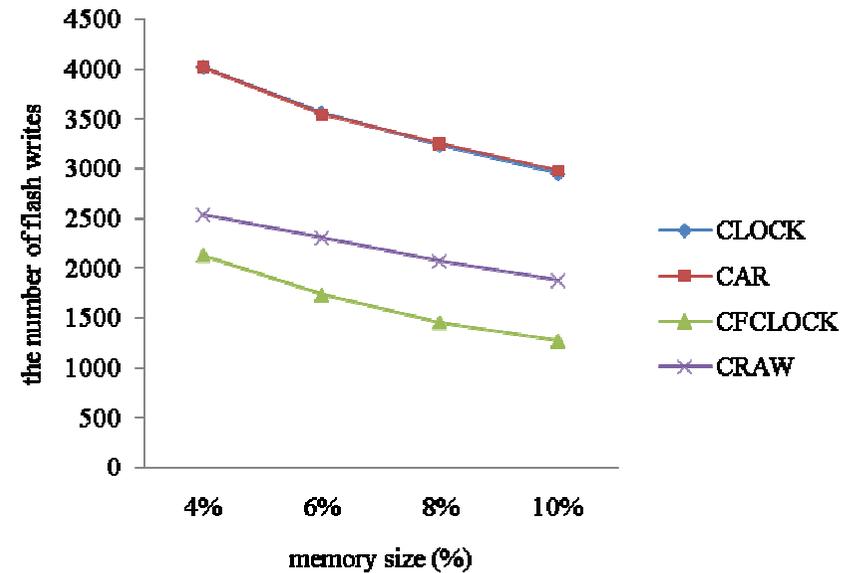
## normalized I/O time



## read operation

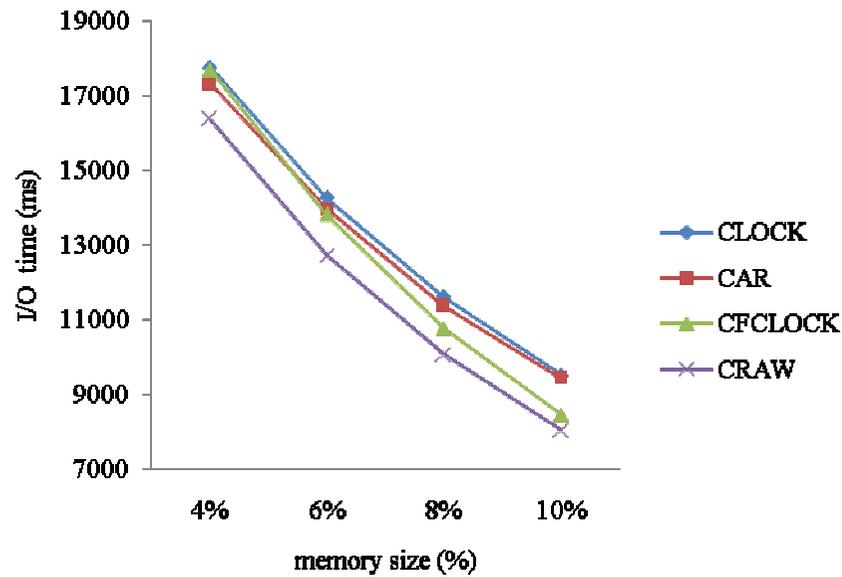


## write operation

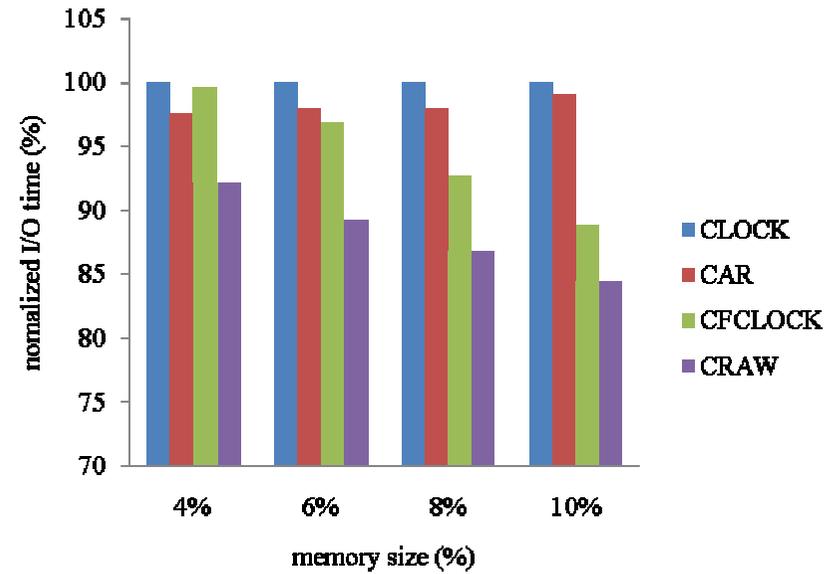


# kghostview

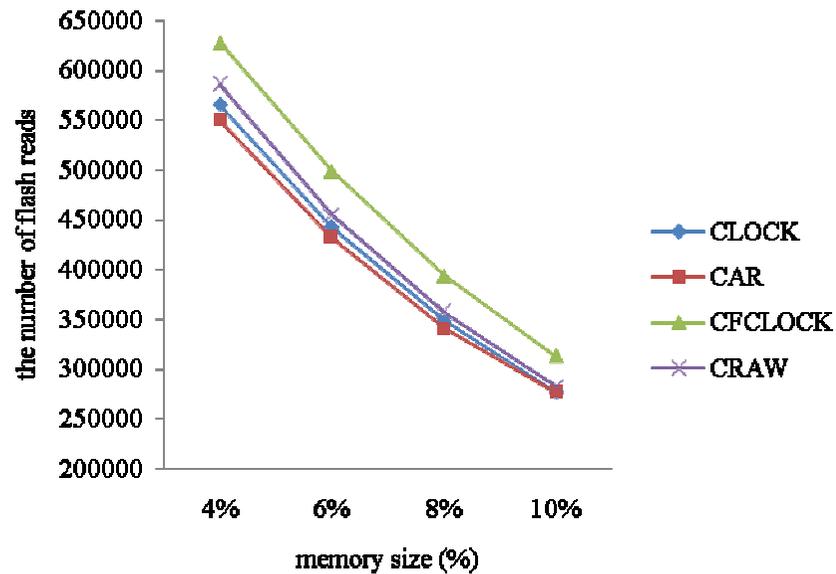
### I/O time



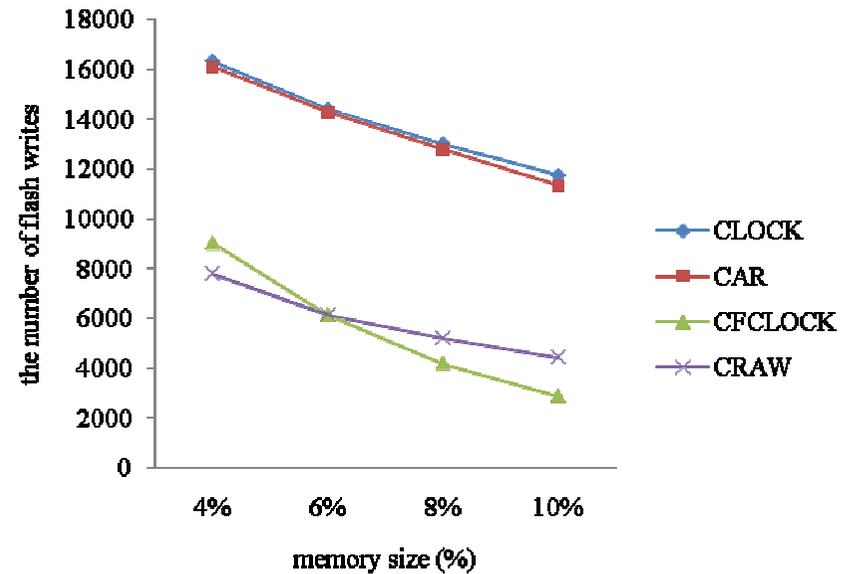
### normalized I/O time



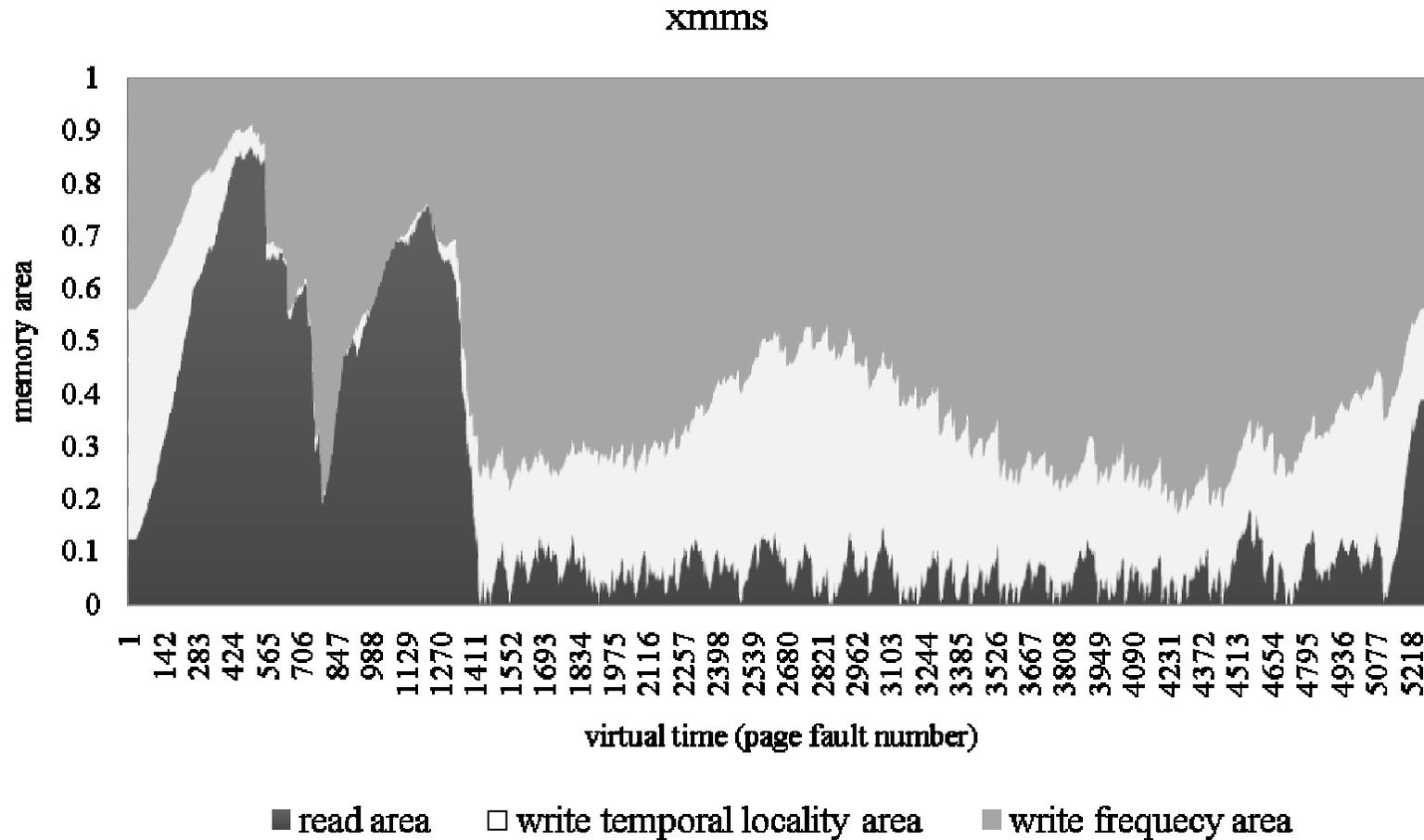
### read operation



### write operation



# 영역별 공간 할당



## 연구 내용의 요약

- 가상메모리의 페이지 참조
  - 읽기는 시간지역성이 강함
  - 쓰기는 시간지역성 역전 현상 발생
- 읽기 영역과 쓰기 영역의 독립적 할당
  - 읽기 영역과 쓰기 영역 크기의 동적 자동 조절 기능
    - 가상 영역의 참조 패턴 변화
    - 읽기/쓰기 연산의 비용 차이
- 가상메모리에서 사용될 수 있는 최적의 조건
- Clock, CAR, CFLRU에 10-20% 성능 향상

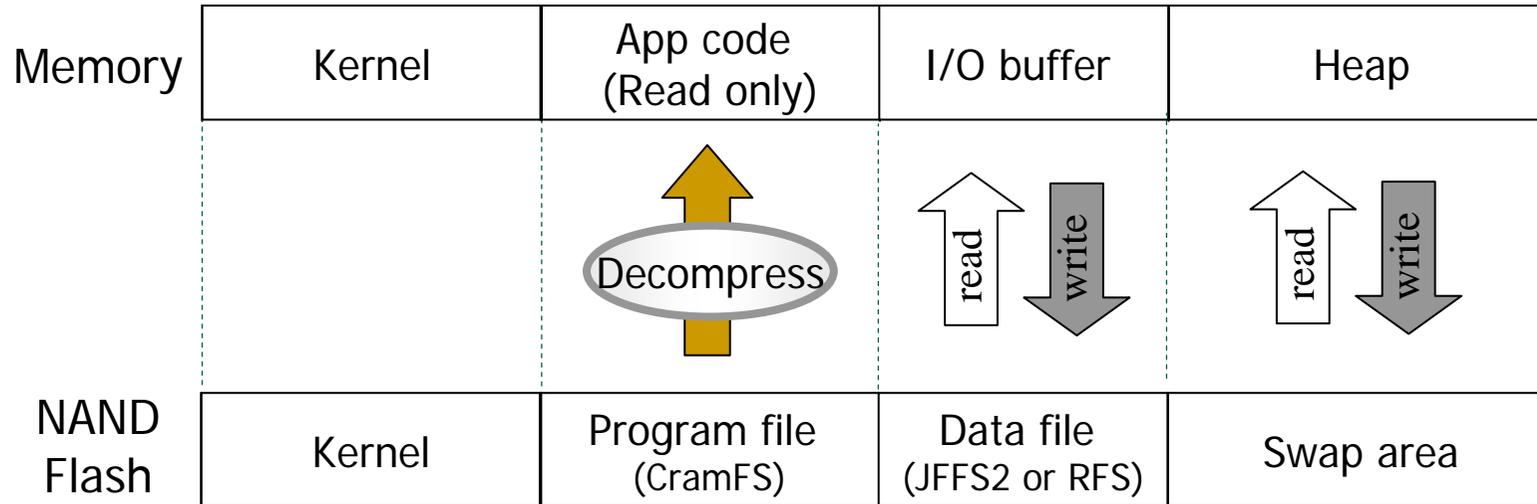
## 연구의 구성

1) 가상메모리의 쓰기 참조 특성 고려

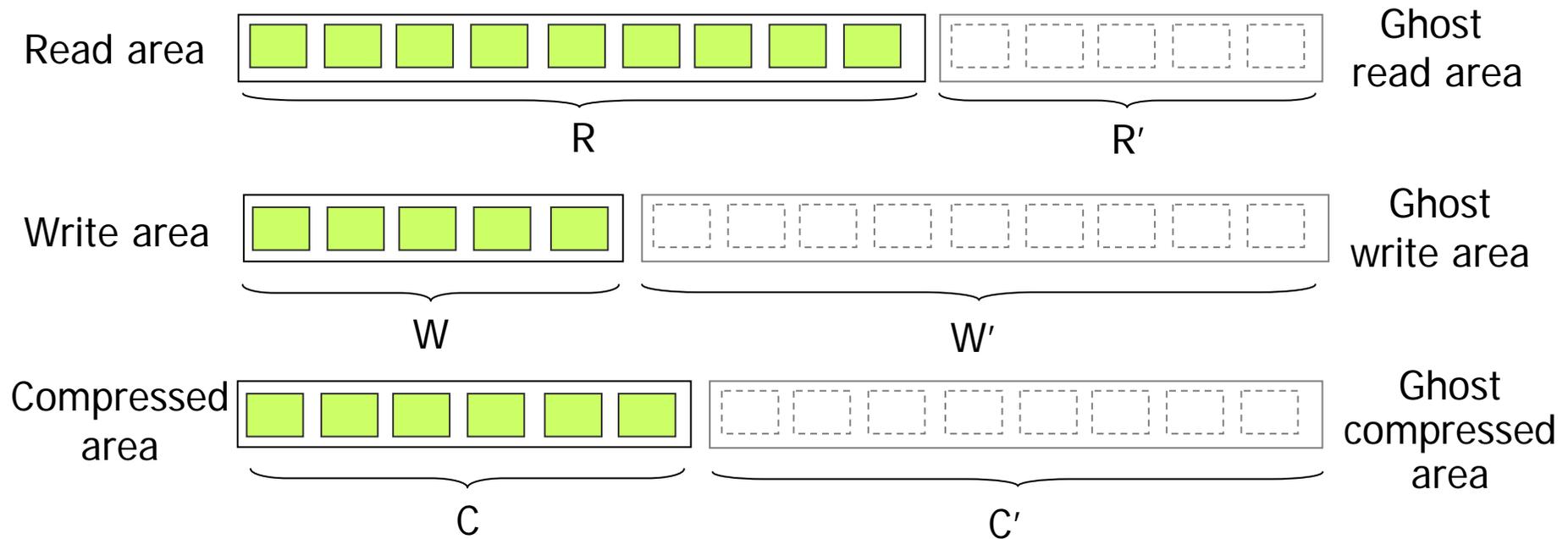
2) 압축 파일 시스템에서의 인출 비용 고려

# 모바일 기기의 파일시스템 구조

- 프로그램 파일: 읽기전용 압축파일시스템에 저장
- 데이터 파일: 일반 파일시스템에 저장



# 읽기 영역, 쓰기 영역, 압축 영역



$$|R| + |R'| = |S|, \quad |W| + |W'| = |S|, \quad |C| + |C'| = |S|$$

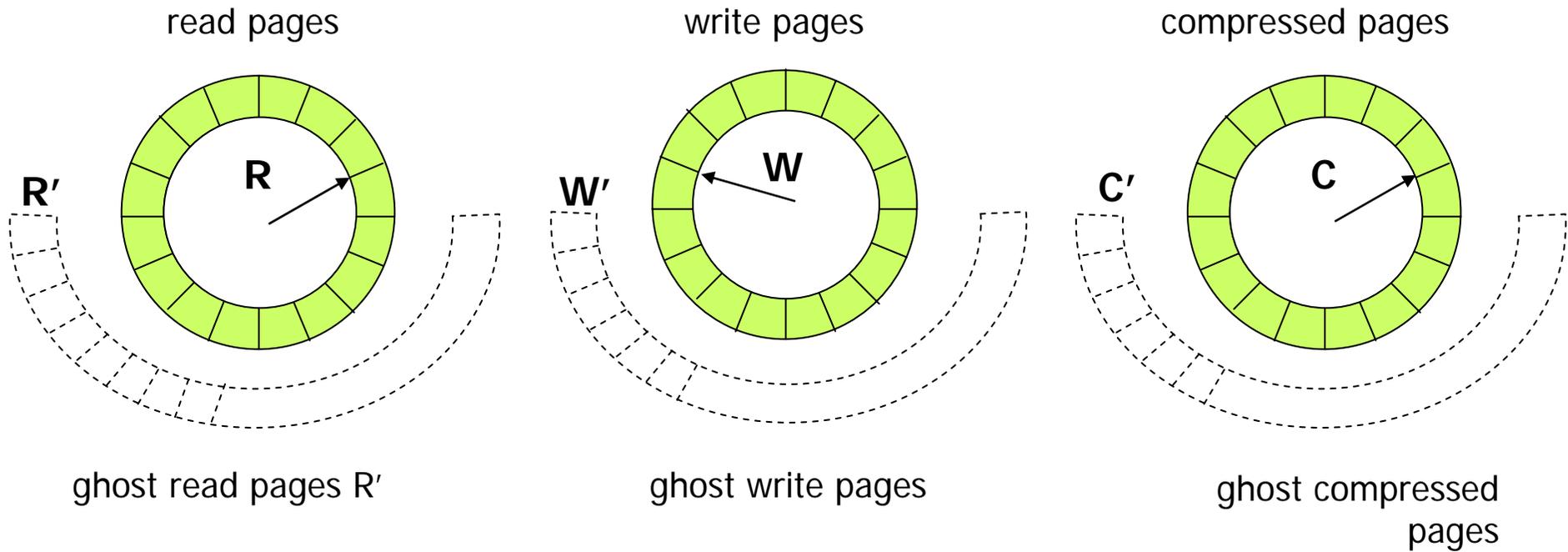
$$R \cup W \cup C = S, \quad |R'| + |W'| + |C'| \leq |2S|$$

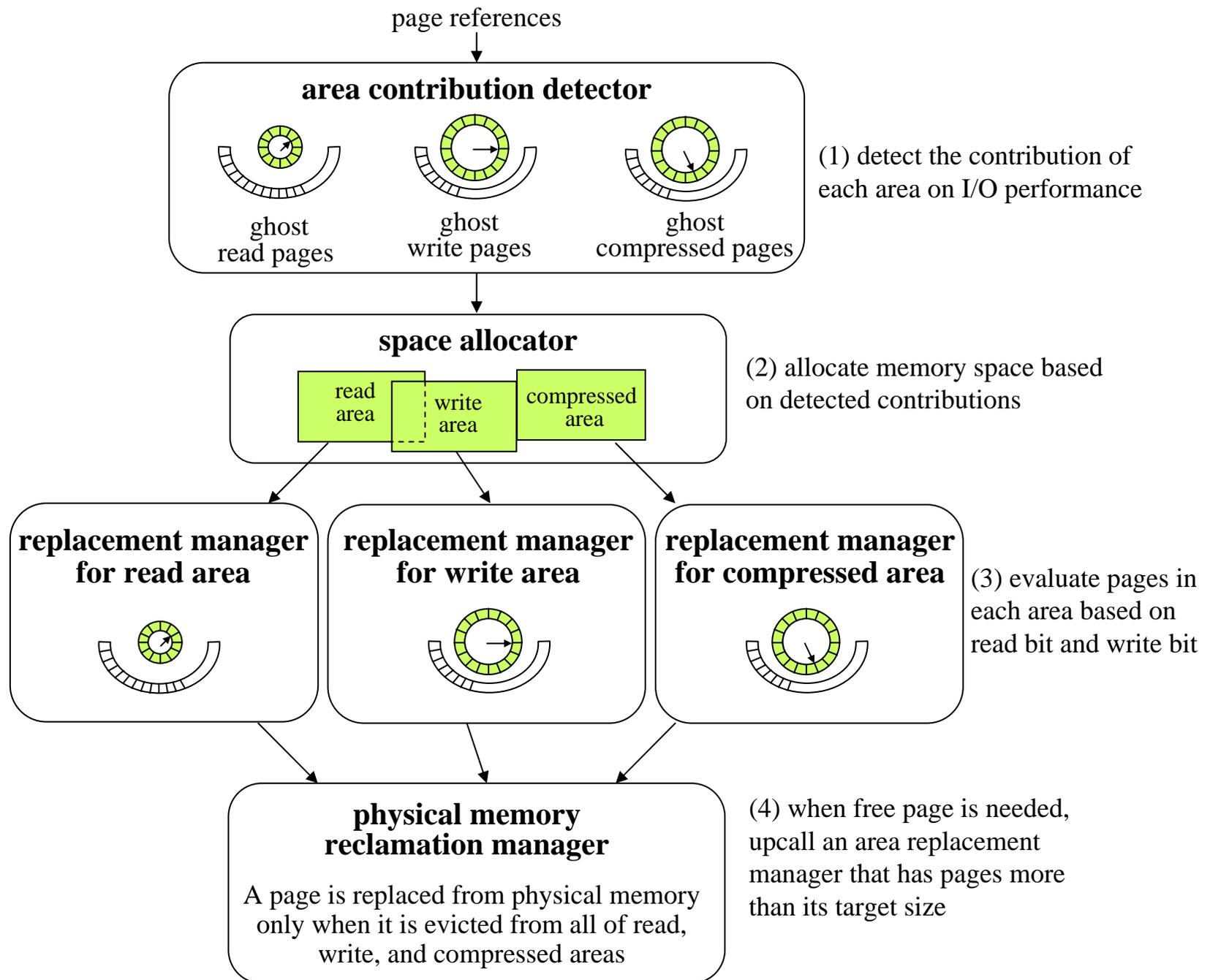
S: total page frames in memory

□ Ghost page

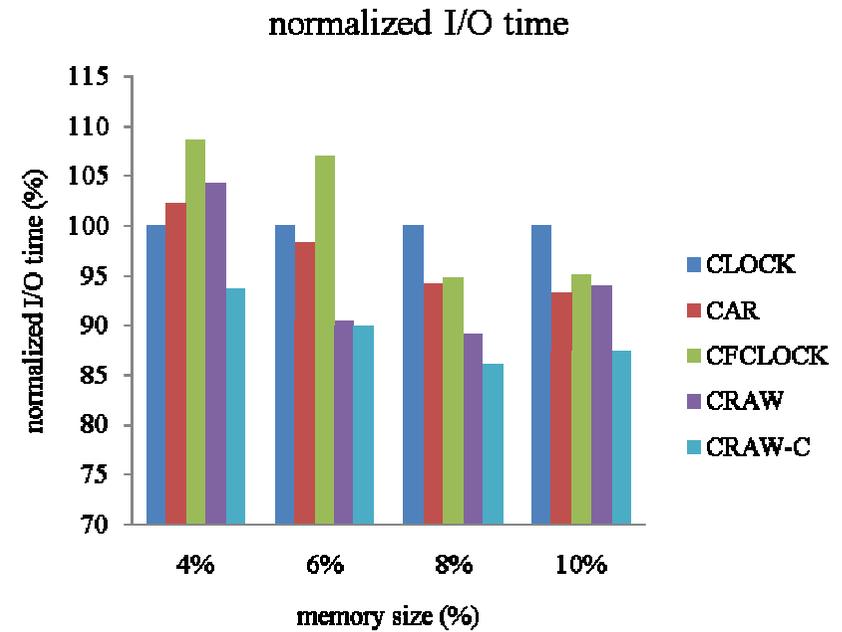
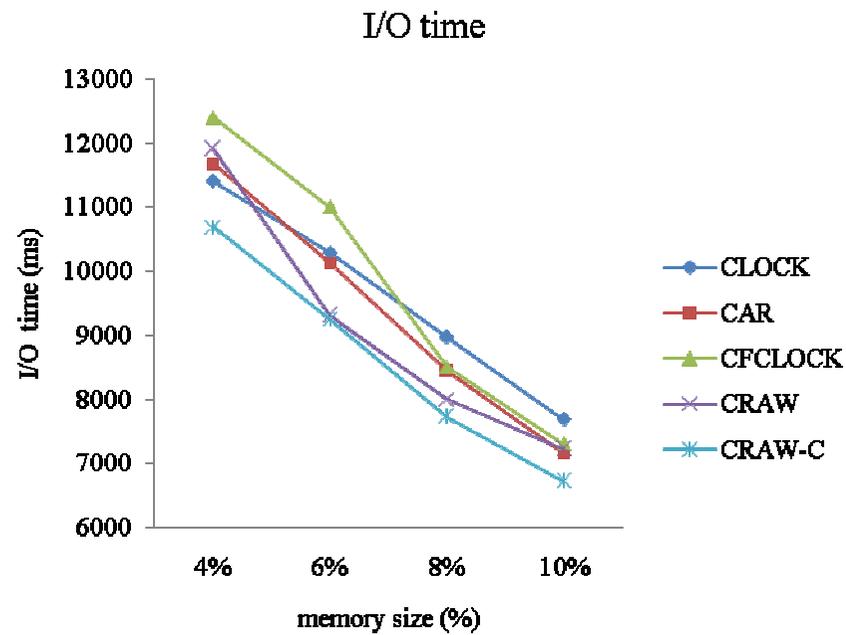
■ Real page

# 읽기 영역, 쓰기 영역, 압축 영역

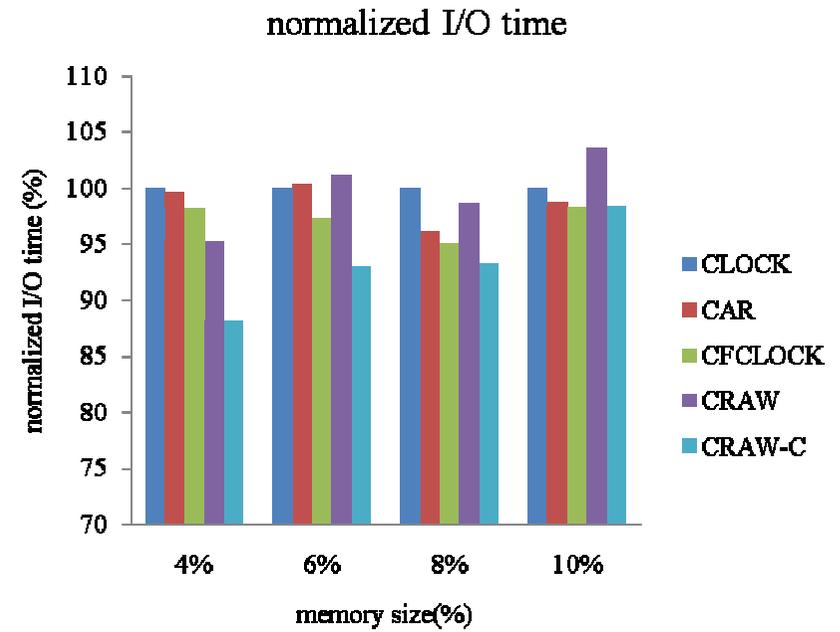
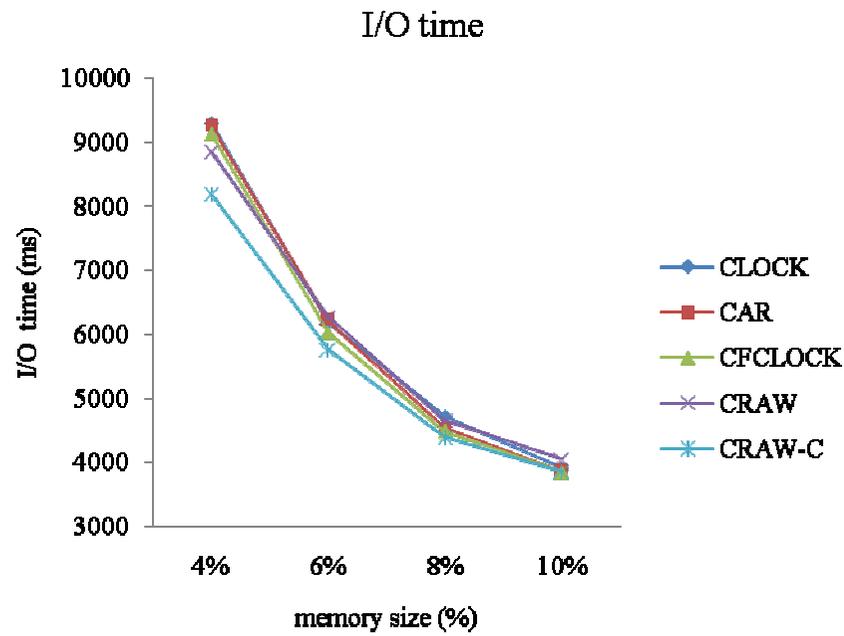




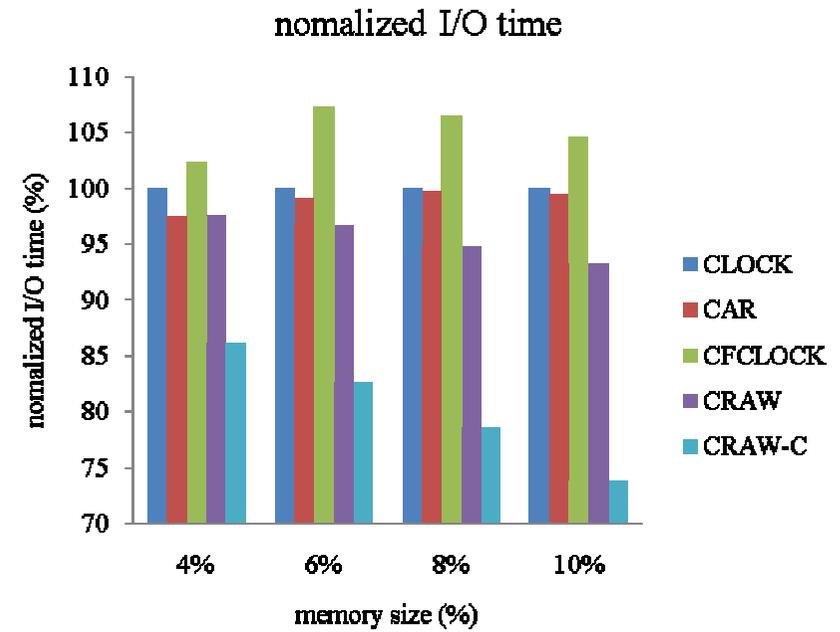
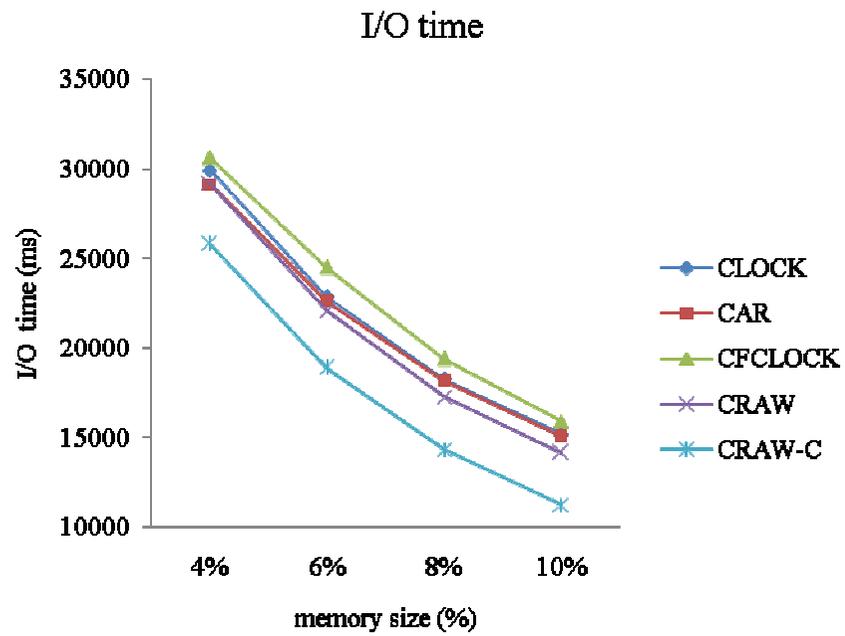
# xmms



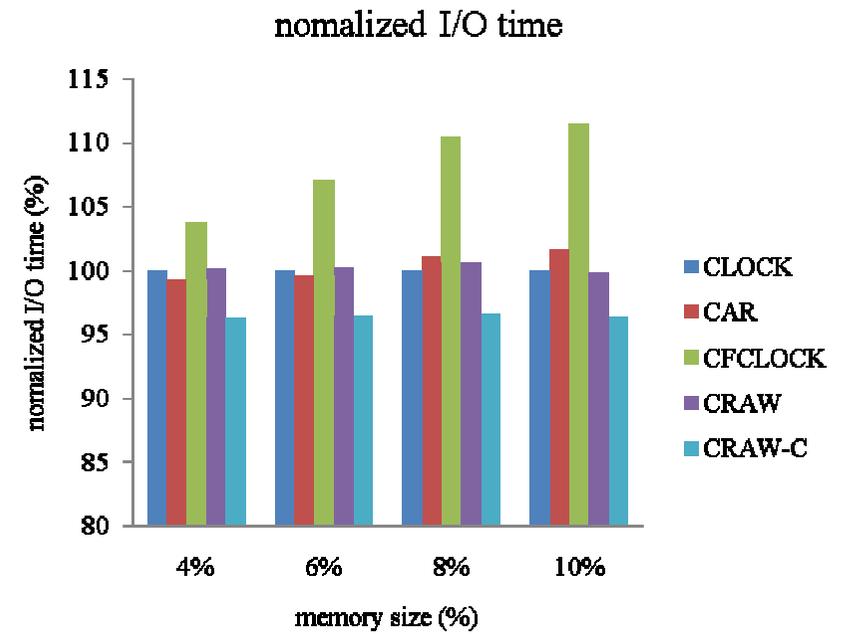
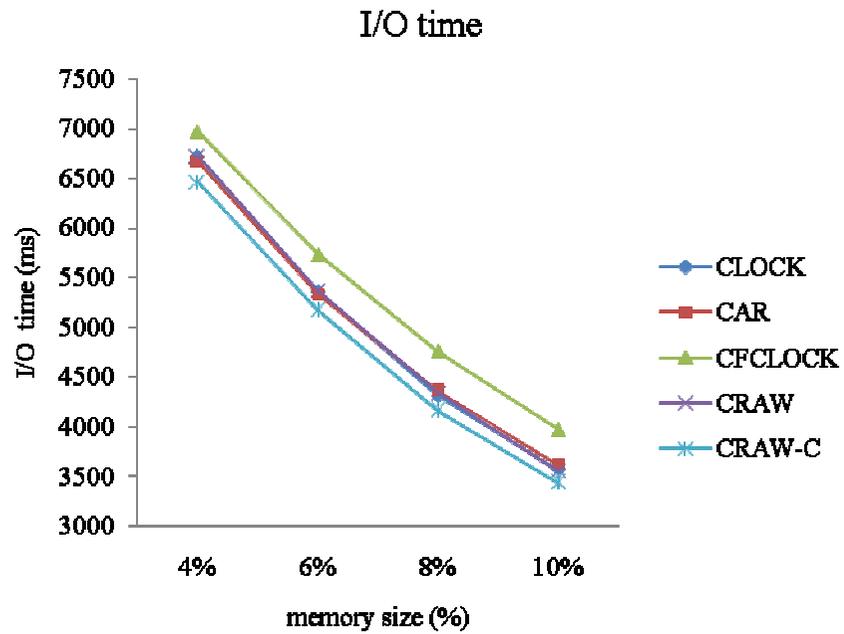
# gqview



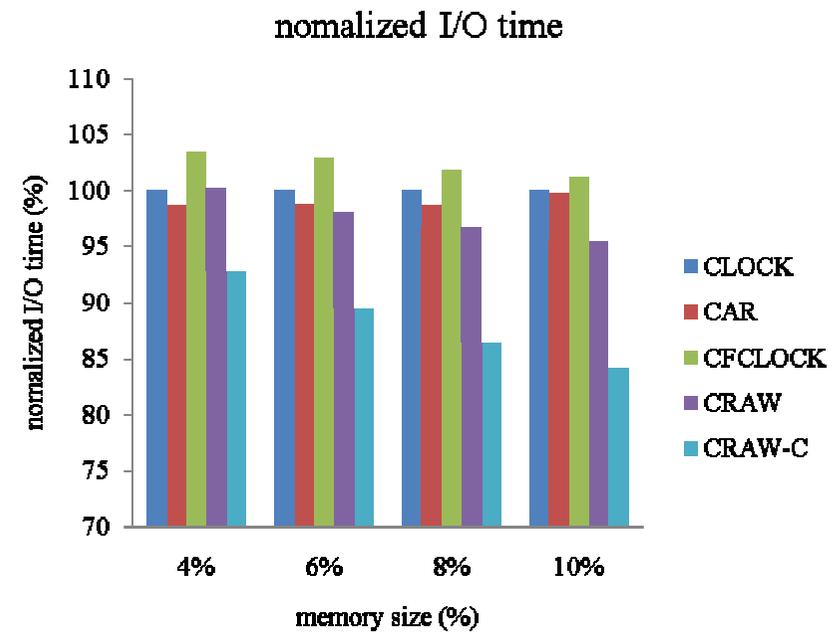
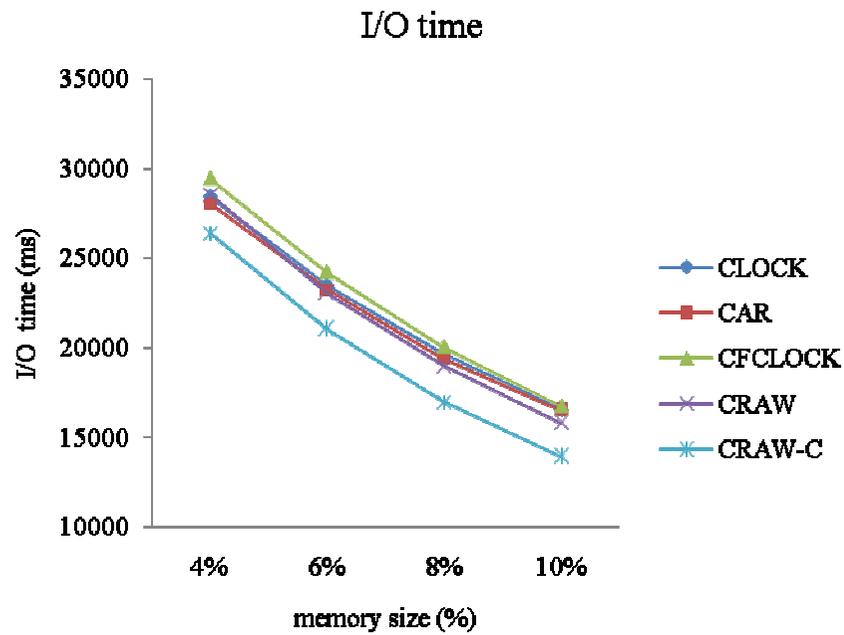
# gedit



# freecell



# kgghostview



## 연구 내용의 요약

- 페이지별 연산 비용 차이
  - 읽기 페이지(JFFS2 또는 스왑 영역)
  - 쓰기 페이지(JFFS2 또는 스왑 영역)
  - 압축 페이지(CramFS)
- 읽기, 쓰기, 압축 영역의 독립적 할당
  - 영역별 크기의 동적 자동 조절 기능
    - 가상 영역의 참조 패턴 변화
    - 읽기/쓰기/압축 페이지의 연산 비용 차이
- 가상메모리에서 사용될 수 있는 최적의 조건