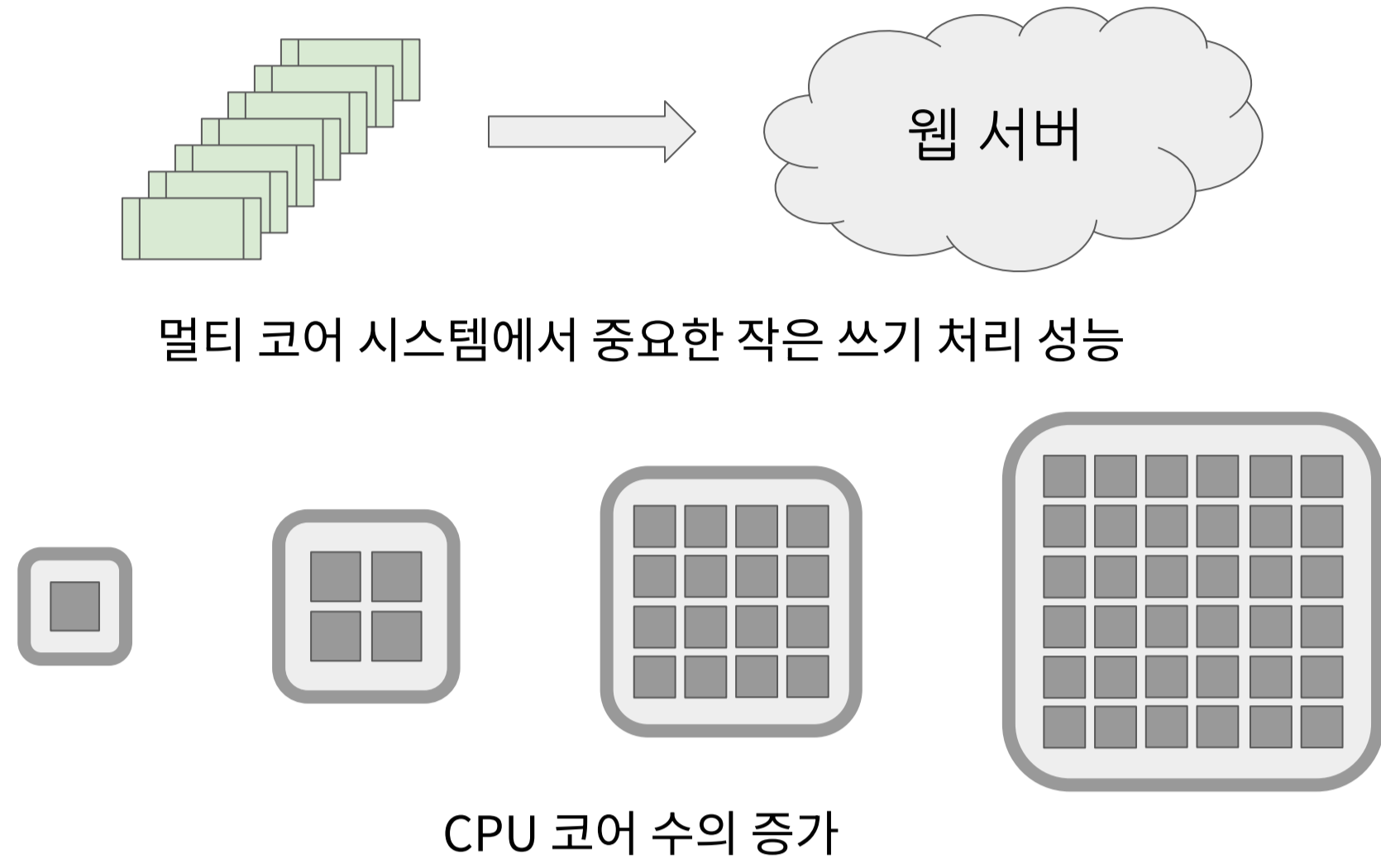


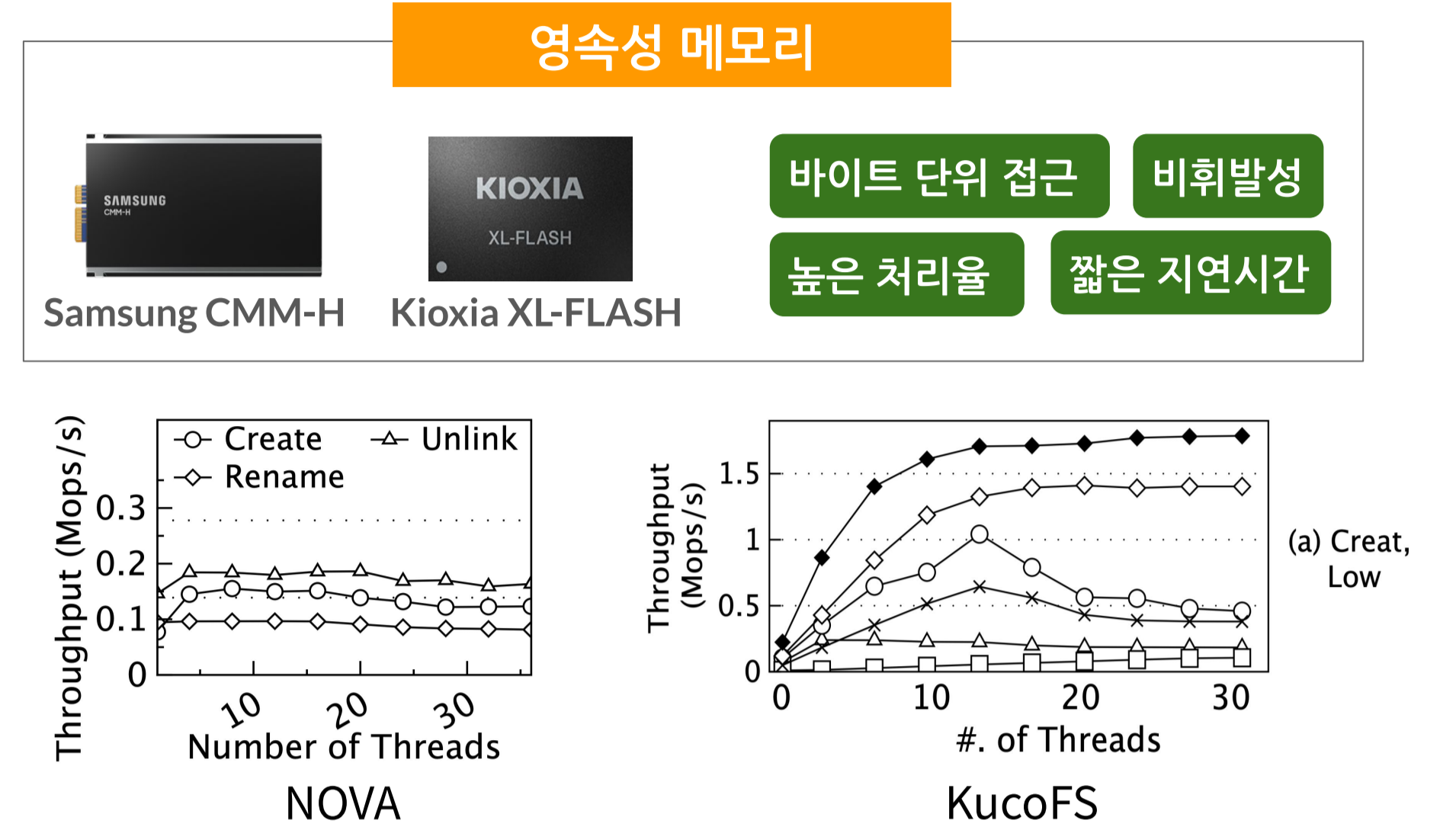
# 프로그래밍 언어 방법론으로 파일 시스템 확장성 개선하기

## 배경: 영속성 메모리 파일 시스템의 메타데이터 확장성 한계

### ● 파일 시스템 메타데이터 확장성의 중요성



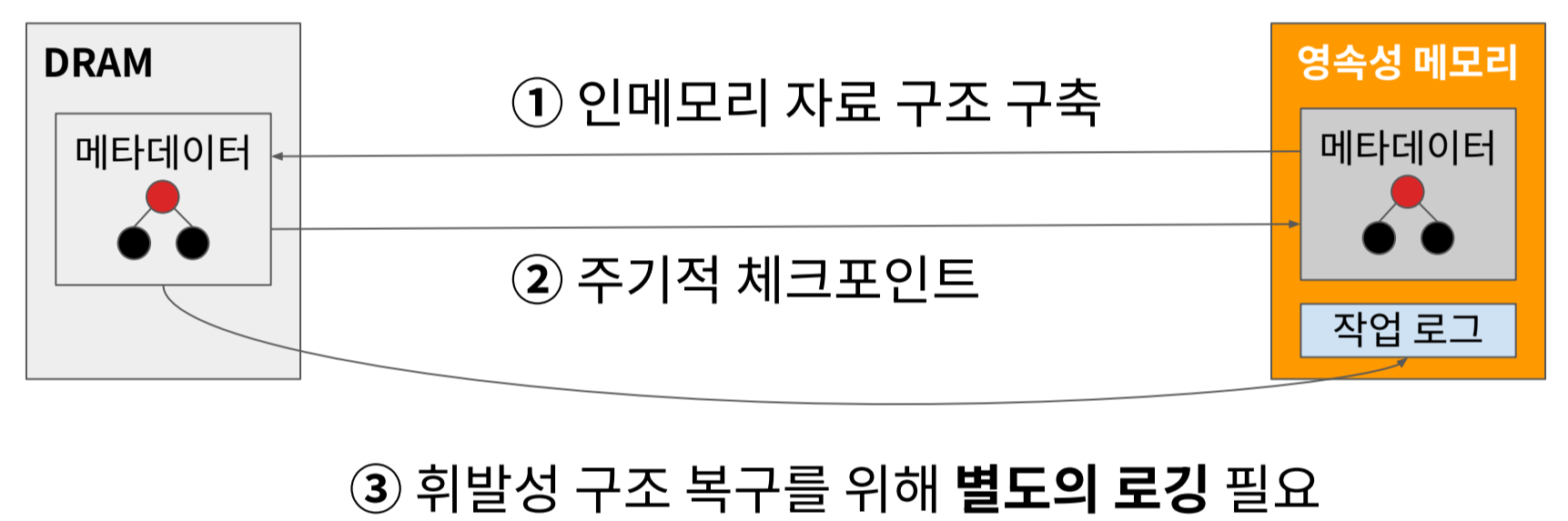
### ● 영속성 메모리 파일 시스템의 확장성 한계



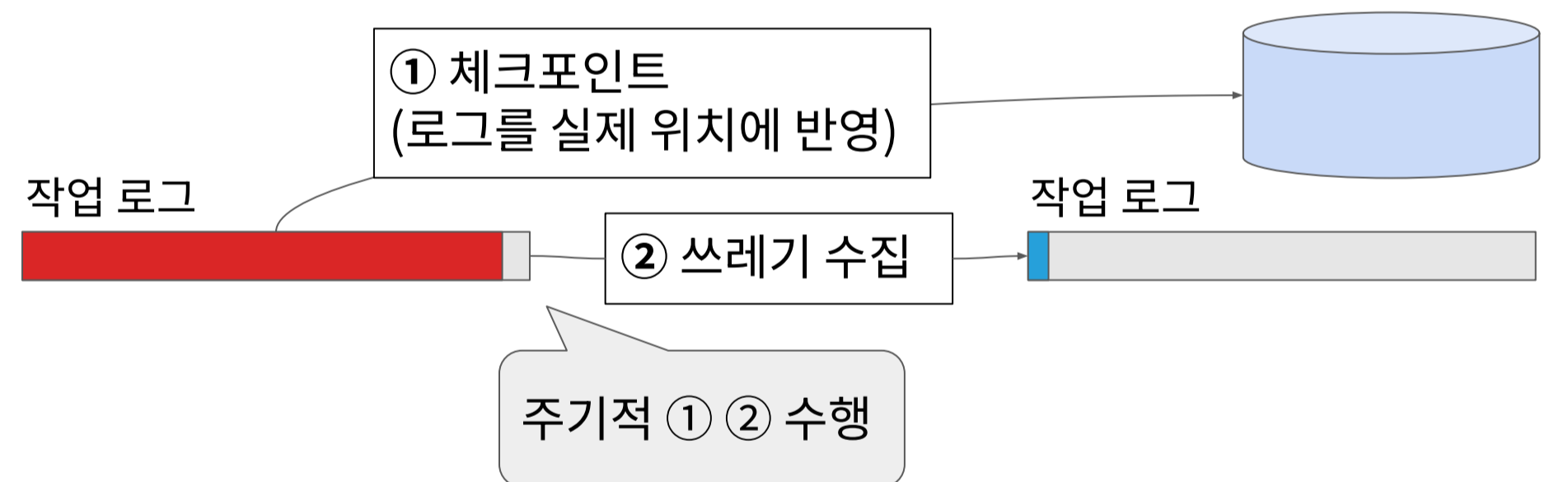
## 문제: 인메모리 및 디렉토리 자료 구조의 한계

파일 시스템	NOVA	SPMFS	KucoFS
확장성 달성 방법	Inode별 로그 관리	코어별 로그 관리	유저 영역으로 다량 전가
병렬성	일부 순차적	병렬적	일부 순차적
파일 생성 성능 (서로 다른 폴더)	△	△	△
파일 생성 성능 (같은 폴더)	👎	△	△
주기적 성능 하락	👎	👎	👎
크래시 복구 시간	👎	👎	👎
일반성	👎	👎	👎
별도 인메모리 구조	○	○	○
커널 오버헤드	👎	👍	👍

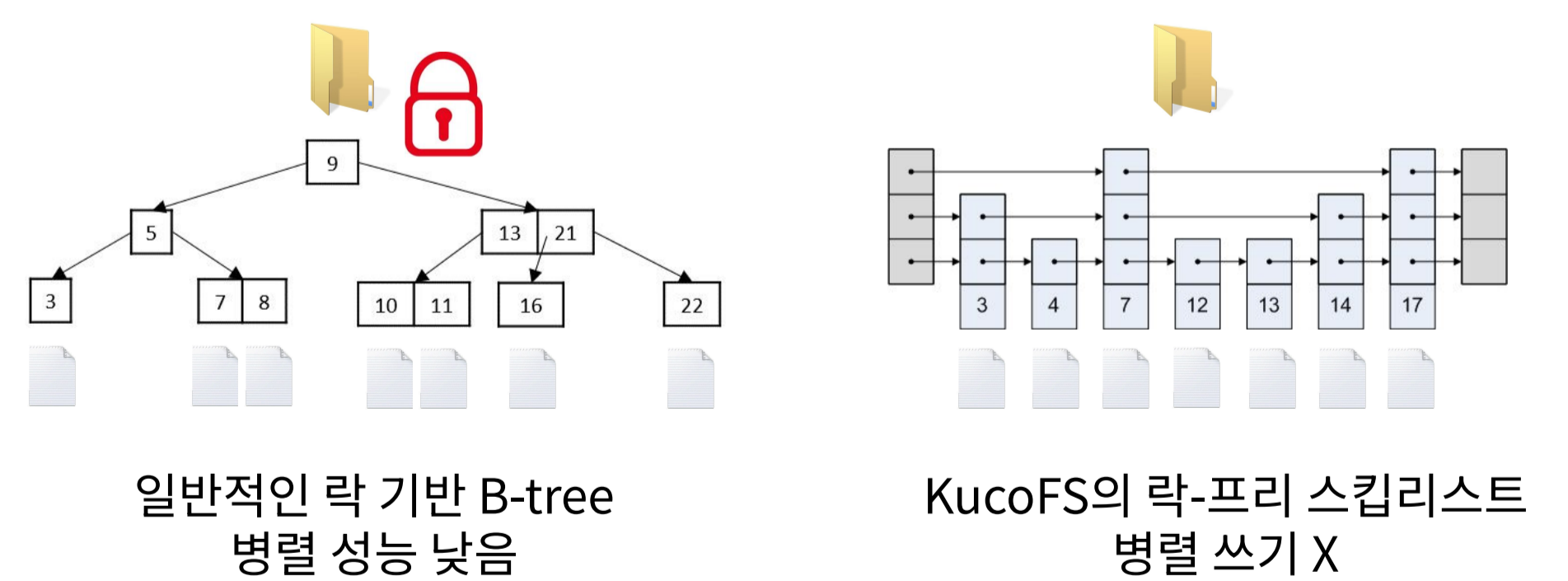
### ● 인메모리 자료 구조 관리로 인한 부하



### ● 로그 처리로 인한 주기적 성능 하락

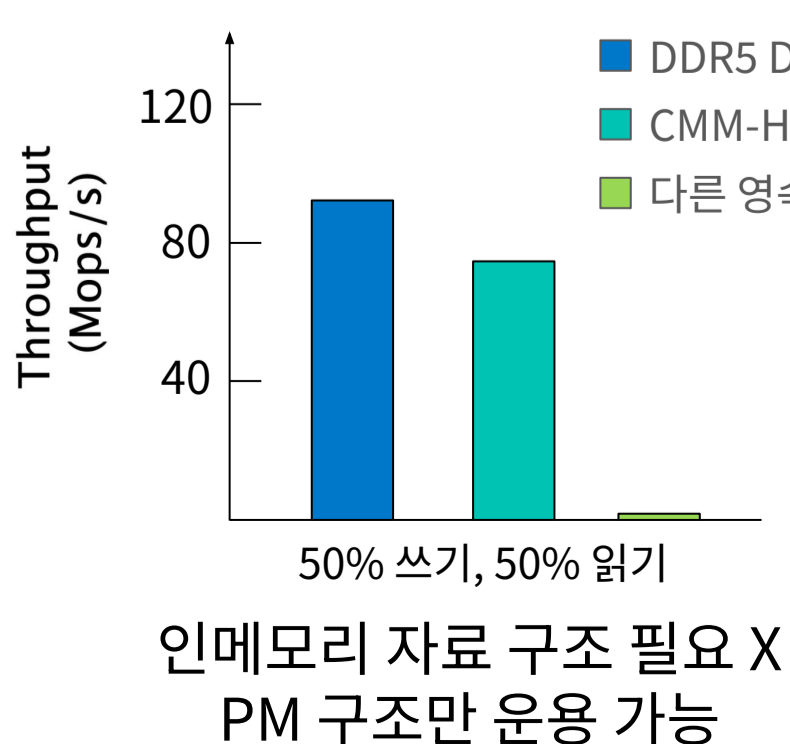


### ● 디렉토리 자료 구조로 인한 확장성 한계

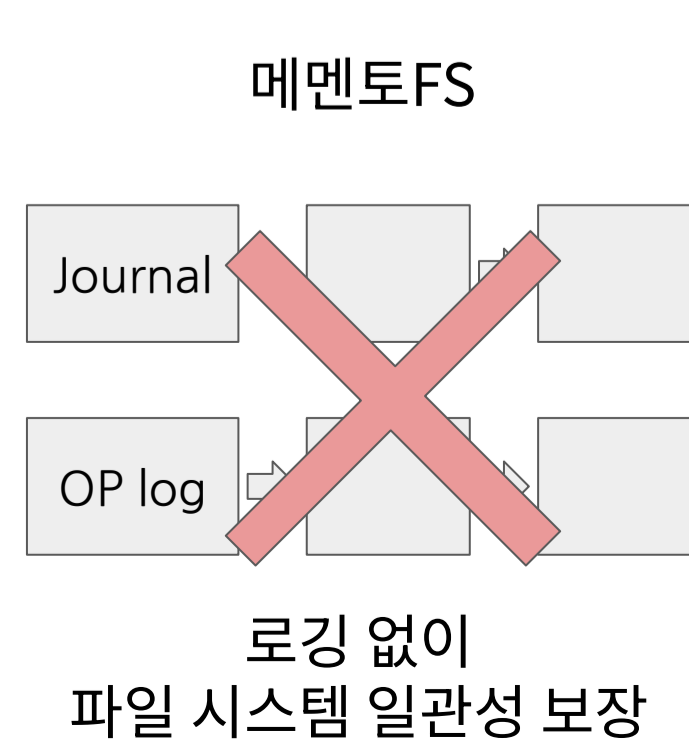


## 해결: 빠른 하드웨어와 크래시 일관성 보장 프레임워크 기반 확장성 높은 파일 시스템 개발

### ● DRAM만큼 빠른 CMM-H



### ● PM 연산 일관성 보장 프레임워크, 메멘토



### ● 메멘토로 도입하는 병렬 디렉토리 구조

