

네이버는 수천 테라바이트의 데이터를 어떻게 서비스할까?

저장시스템개발팀/김태웅

목차

- I. 풀어야 할 숙제들
- II. NHN 파일스토키지 OwFS
- III. OwFS 적용으로 얻은 이점
- IV. 맺음말

풀어야 할 숙제들

네이버 서비스의 데이터

- 검색서비스를 위한 데이터
- 콘텐츠 제공자로부터 공급받은 데이터
- 개인사용자가 생성하는 메일, 업로드된 파일, UGC (User Generated Content), ...



NAVER

검색

류현진 세계신기록 | 국가장학기금 폐지 | 수암골 삼식이 요양

메일 카페 블로그 지식N 쇼핑 | 사전 뉴스 증권 부동산 지도 영화 뮤직 | N 드라이브 더보기

2 플로리스트

408

AD캐스트

네이버
굿바이! 익스플로러6

버려야 할 것은..
언제 만들었는지 기억도 가물가물한
지갑속의 카드들만이 아니다



아이디

ID저장

비밀번호

로그인

회원가입

아이디/비밀번호 찾기

보안 2단계

1 2 3

IP보안 ON

N드라이브 서비스

3백만명 이상의 이용자
이용자당 10GB 저장공간 무료제공

전체 저장공간은 최대 수십 PB이상 늘어날 수 있음

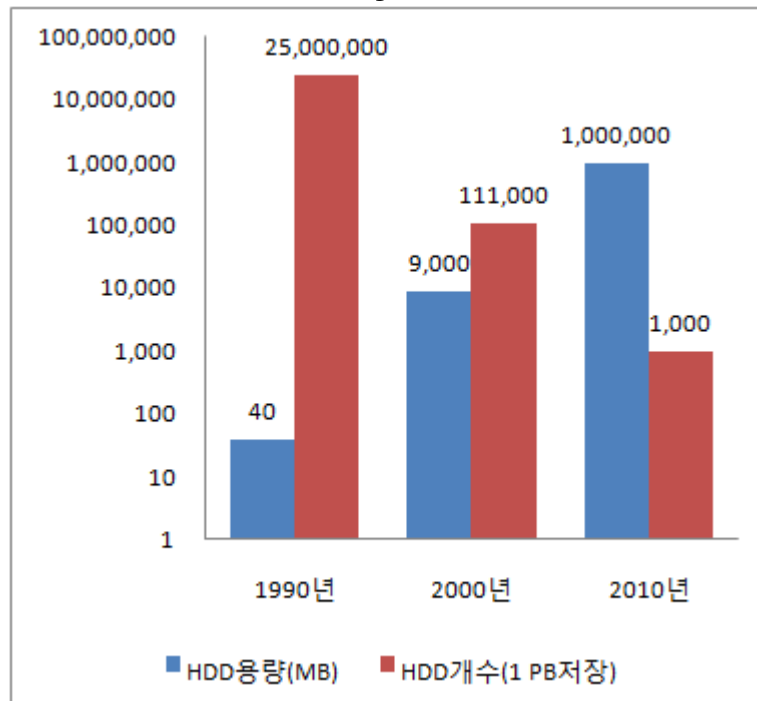


웹에서 만나는 나만의 저장 공간 네이버 **N드라이브**

이제, 온라인에서 여러분의 파일을 편리하게 관리하고 다양하게 활용해 보세요.

데이터 저장의 문제

1,000 TB (Terabyte) = 1 PB (Petabyte)를 저장하려면?



HDD의 Datasheet

MTTF (Mean Time To Failure) = 1M ~ 1.5M hours

AFR (Annual Failure Rate) < 0.88%

그러나, 실제환경에서는 4% 이상의 AFR을 나타낼 수 있음

스토리지 하부구조에 대한 요구사항

- 장애에 대한 고가용성
- 다운타임없는 유지보수

Better

- 확장성 (성능/용량)
- 높은 처리량

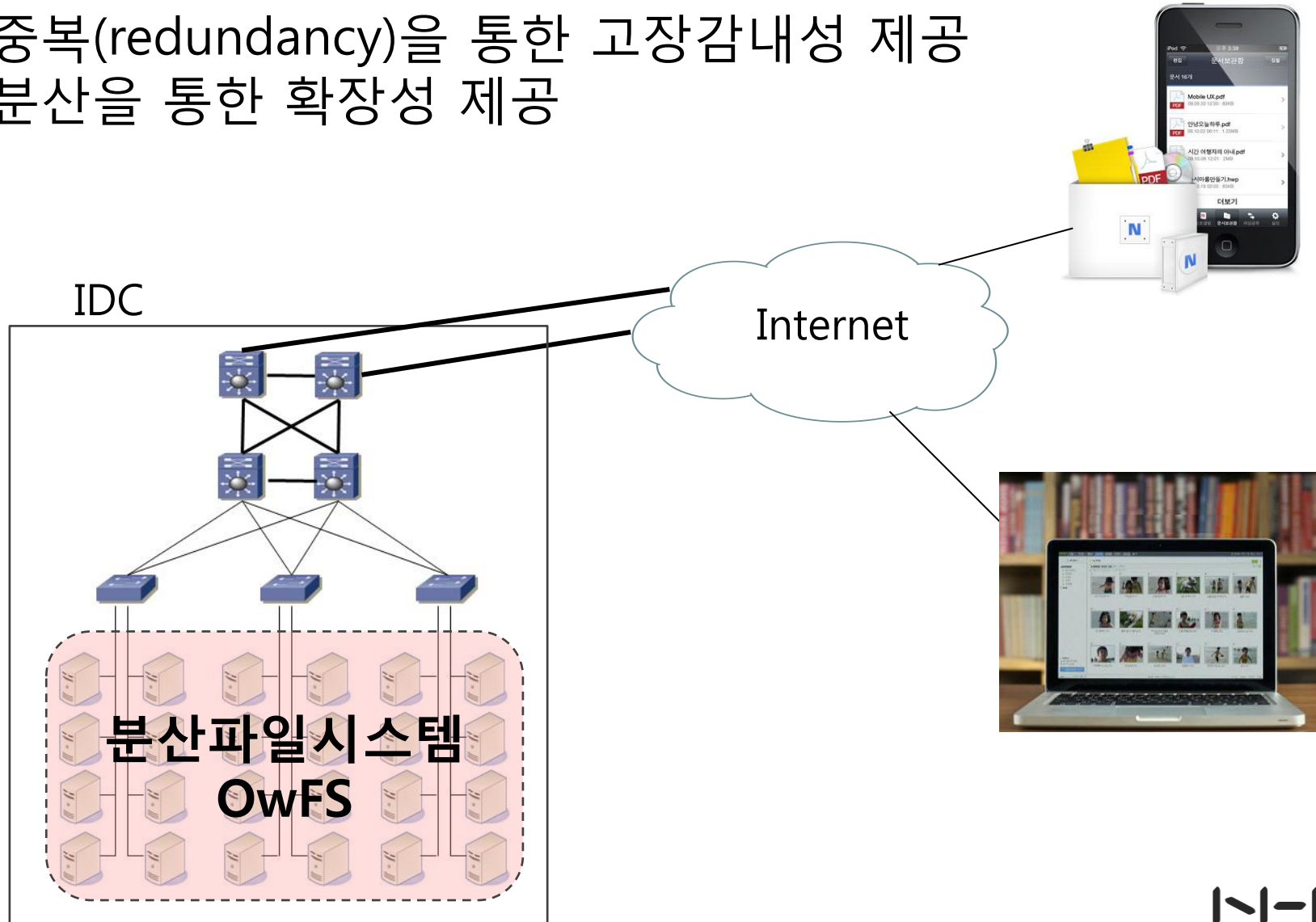
- TB당 구축 비용
- 낮은 운영 비용

Faster

Cheaper

문제해결 방안

중복(redundancy)을 통한 고장감내성 제공
분산을 통한 확장성 제공



NHN 파일스토리지 OwFS

온라인 파일서비스의 특성

● 파일의 특성

- 파일의 개수가 수십억개 이상으로 늘어남
- 개개의 파일의 크기는 작음 (수 KB ~ 수십 MB이 대부분)
- 단일 서비스의 저장공간이 수십 Petabyte 이상으로 늘어남

● 파일 접근 패턴

- WORM (Write-Once-Read-Many)
 - Create, Read, Delete
- 새롭게 생성된 파일에 대한 참조 지역성 있음

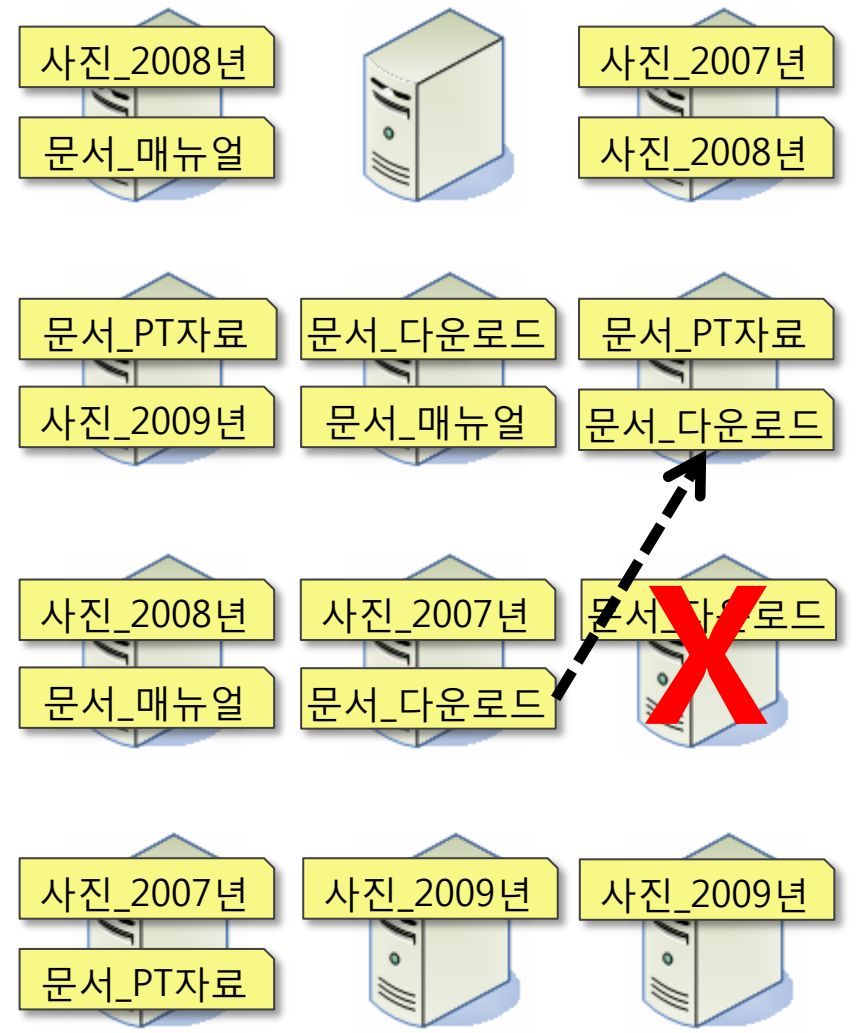
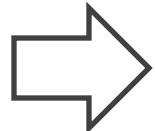
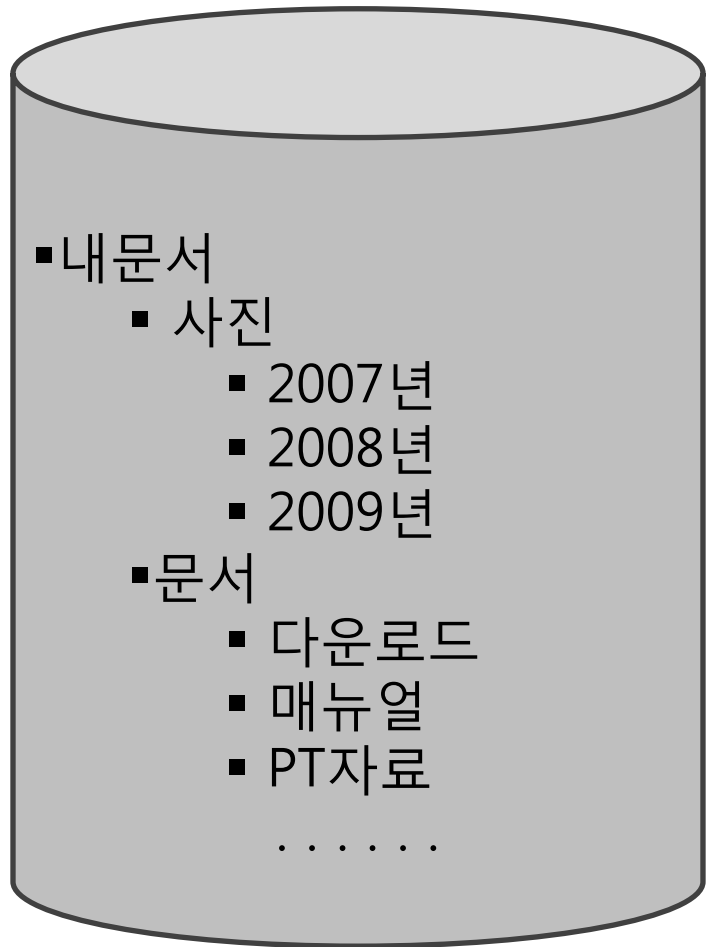
어떻게 저장하는 것이 효과적일까?

- 개별 파일의 저장
 - 파일은 분할해서 저장하지 않고 전체를 저장
- 파일은 3개의 복제본 가짐
 - HDD/서버 고장에 대응
 - 파일 쓰기 부담은 늘어나지만, 읽기 부하는 분산 가능
- 파일의 복제본 정보 관리 (파일시스템의 메타데이터)
 - 개별 파일 단위로 복제본 정보를 저장하지 말고 모아서 관리
 - 서로 관련된 파일들을 모아 놓은 것이 "Owner"

Owner는 분산과 복제의 기본 단위

**OwFS (Owner-based File System)에서 파일의 경로
(Owner이름, Pathname)**

Owner 개념

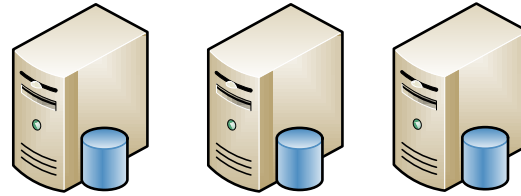


Kernel level vs. User level

- 분산 파일시스템의 구현 방법
- 고려사항
 - 성능
 - Legacy 응용과 호환성
 - 개발 및 시험 기간
 - 코드 유지보수
 - 파일시스템 버그로 인한 영향도
 - 플랫폼 업그레이드 용이성
 - 이식성
- User level 구현의 장점이 더 많음

OwFS 동작방식

메타데이터 서버 (MDS)



Owner map

Owner이름	복제본
홍길동	1,2,3
이몽룡	3,4,5
성춘향	2,5,6
변학도	1,4,6

Owner 조회

Owner의 복제본 정보

Owner map 캐쉬



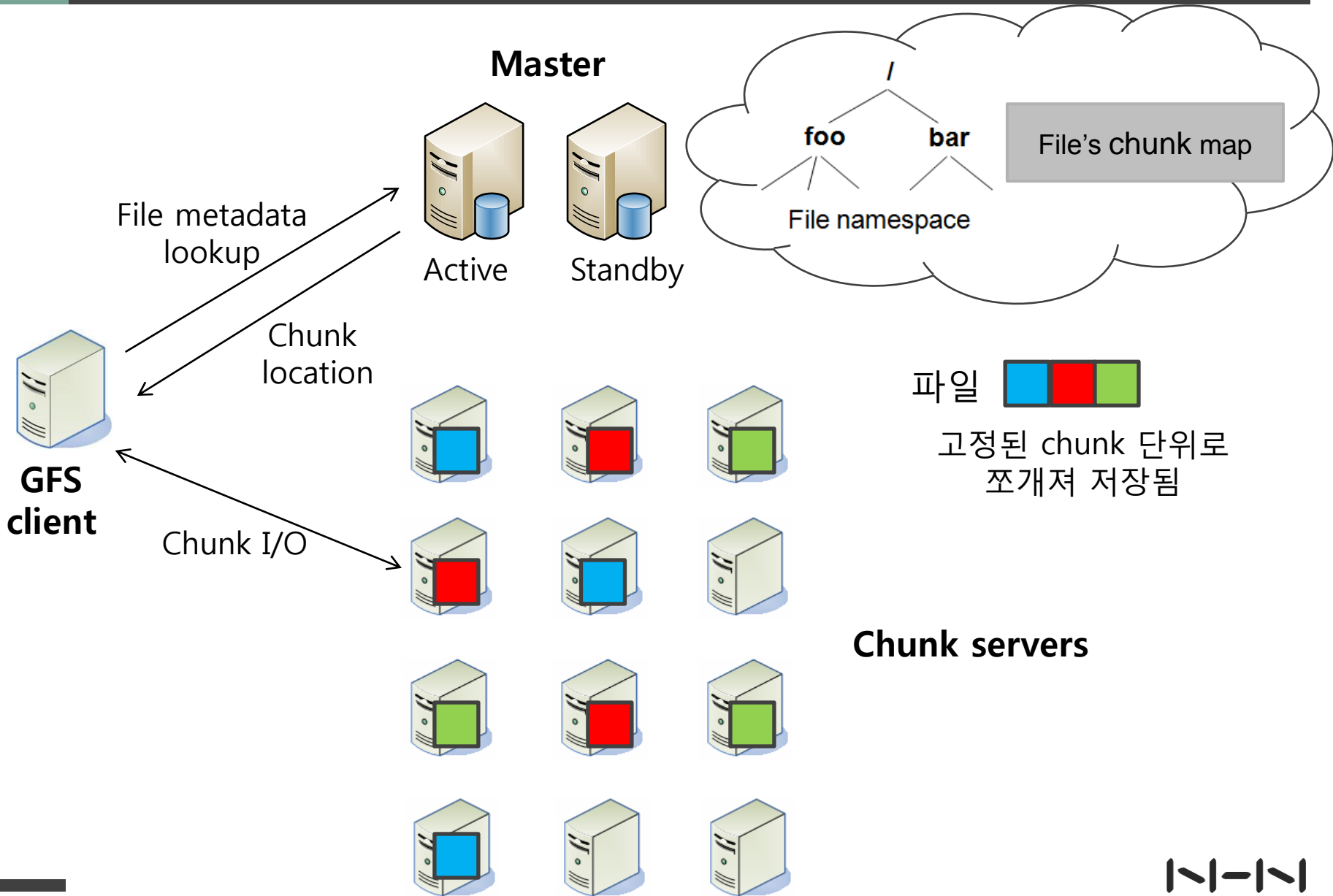
응용 서버

파일 I/O

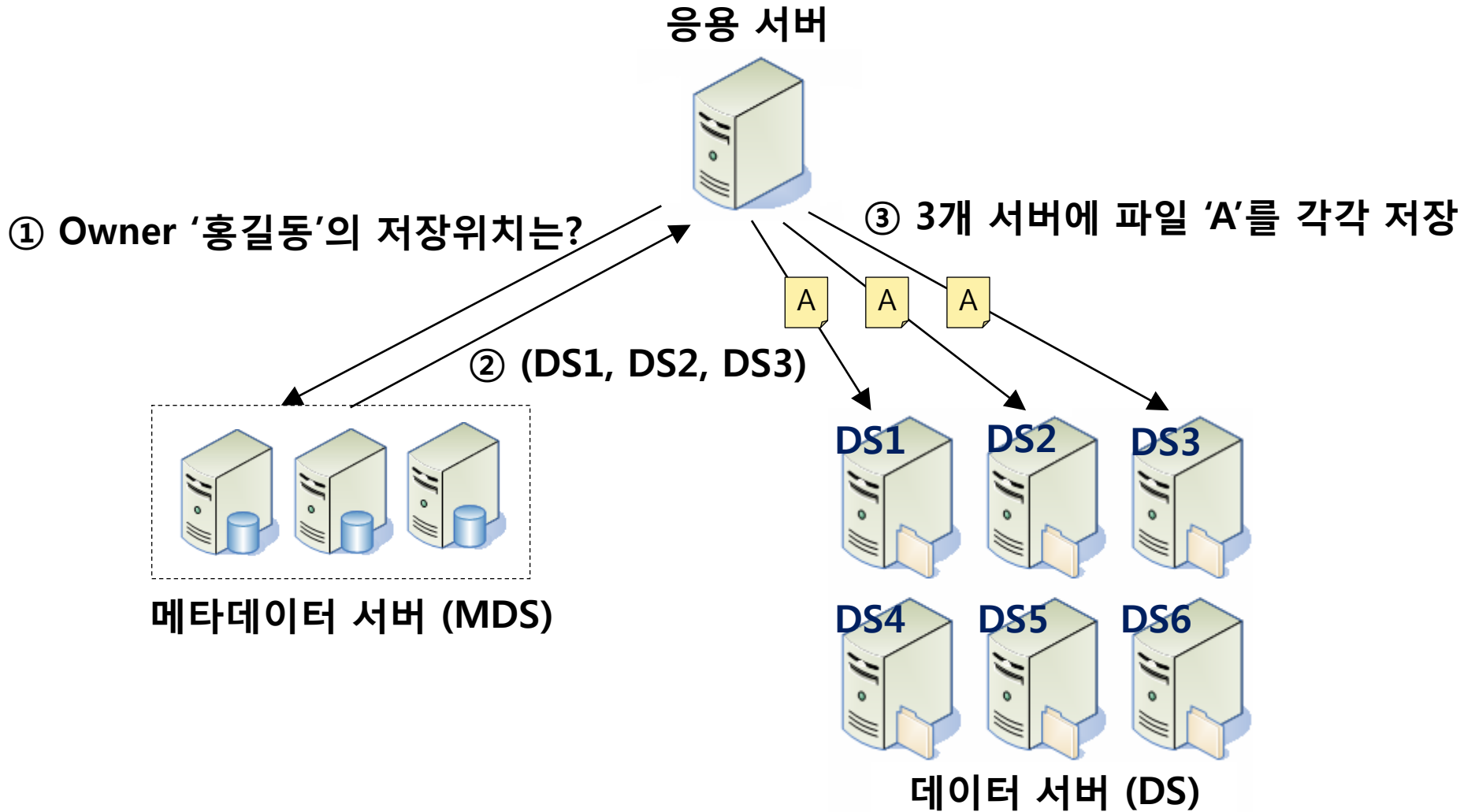


데이터 서버 (DS)

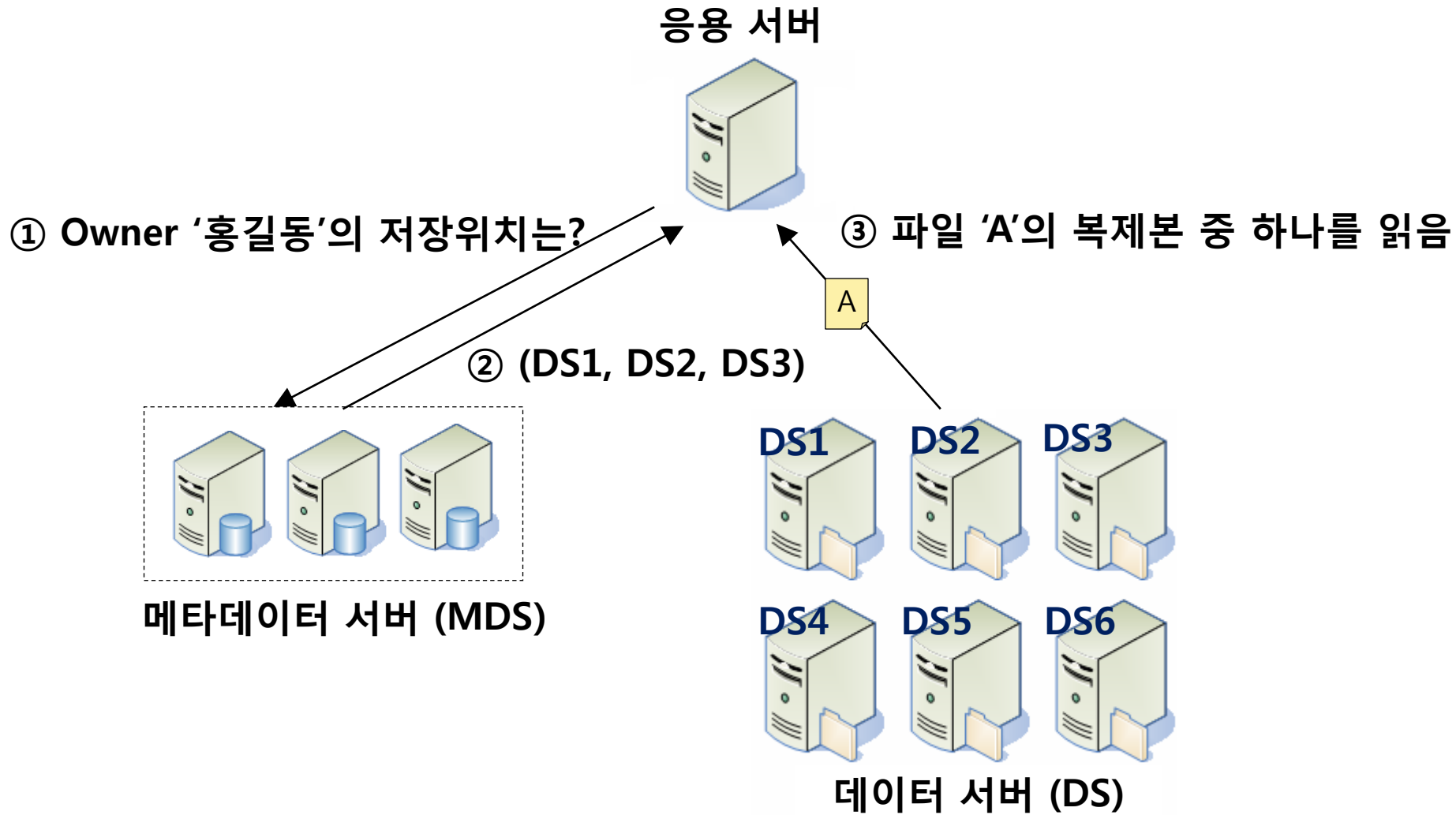
타 파일시스템과 비교 (GFS: Google File System)



파일 쓰기 동작



파일 읽기 동작



OwFS가 지원하는 API

● Owner 연산

- Owner 생성/삭제 (undelete도 가능)/이름변경/리스트조회

● 파일 연산

- 파일 생성/덮어쓰기/append
 - 파일의 중간 부분변경은 지원하지 않음
- 파일 읽기
- 파일 삭제 (undelete도 가능)
- 파일 이름 변경
- 파일 속성 읽기
- 파일 존재 여부 확인

● 디렉토리 연산

- 디렉토리 생성/삭제/이름변경/파일리스트 조회

- **OwFS의 고장모델**

- Fail-Stop

- **개별 서버 자체 모니터링**

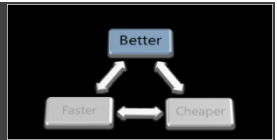
종류	내용
고장모니터링	HDD고장, 커널/파일시스템 고장
	네트워크 포트
	데몬 비정상 종료
성능모니터링	CPU/네트워크/디스크 사용률, 처리량
	메모리, 쓰레드, 네트워크 연결
	파일 연산별 누적 카운터

- **서버간 모니터링**

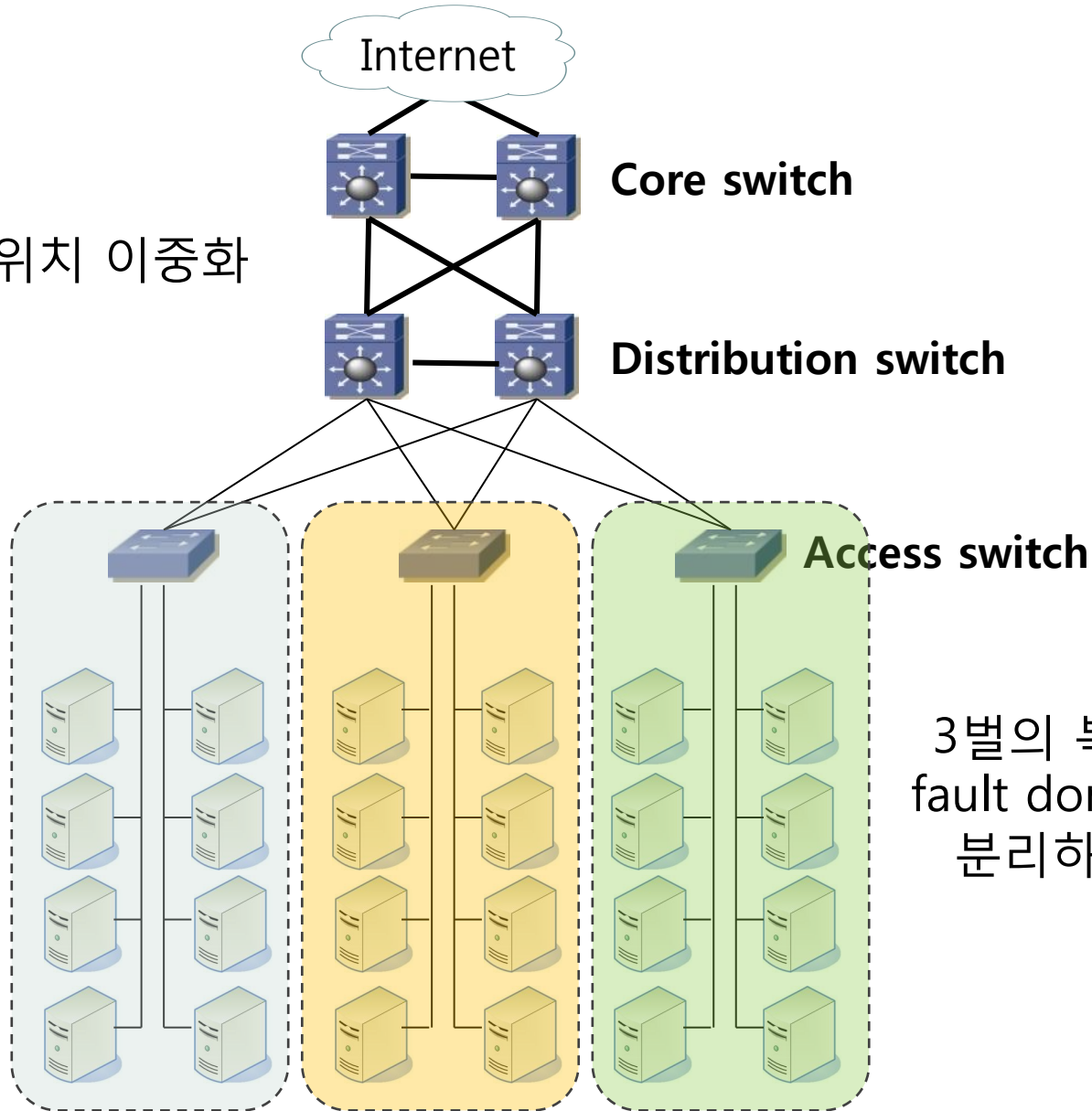
- Timeout (Heartbeat, 개별 요청처리)

OwFS 적용으로 얻은 이점

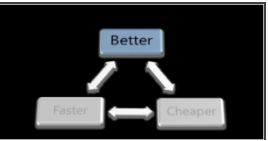
고장에 대한 대응



네트워크 스위치 이중화

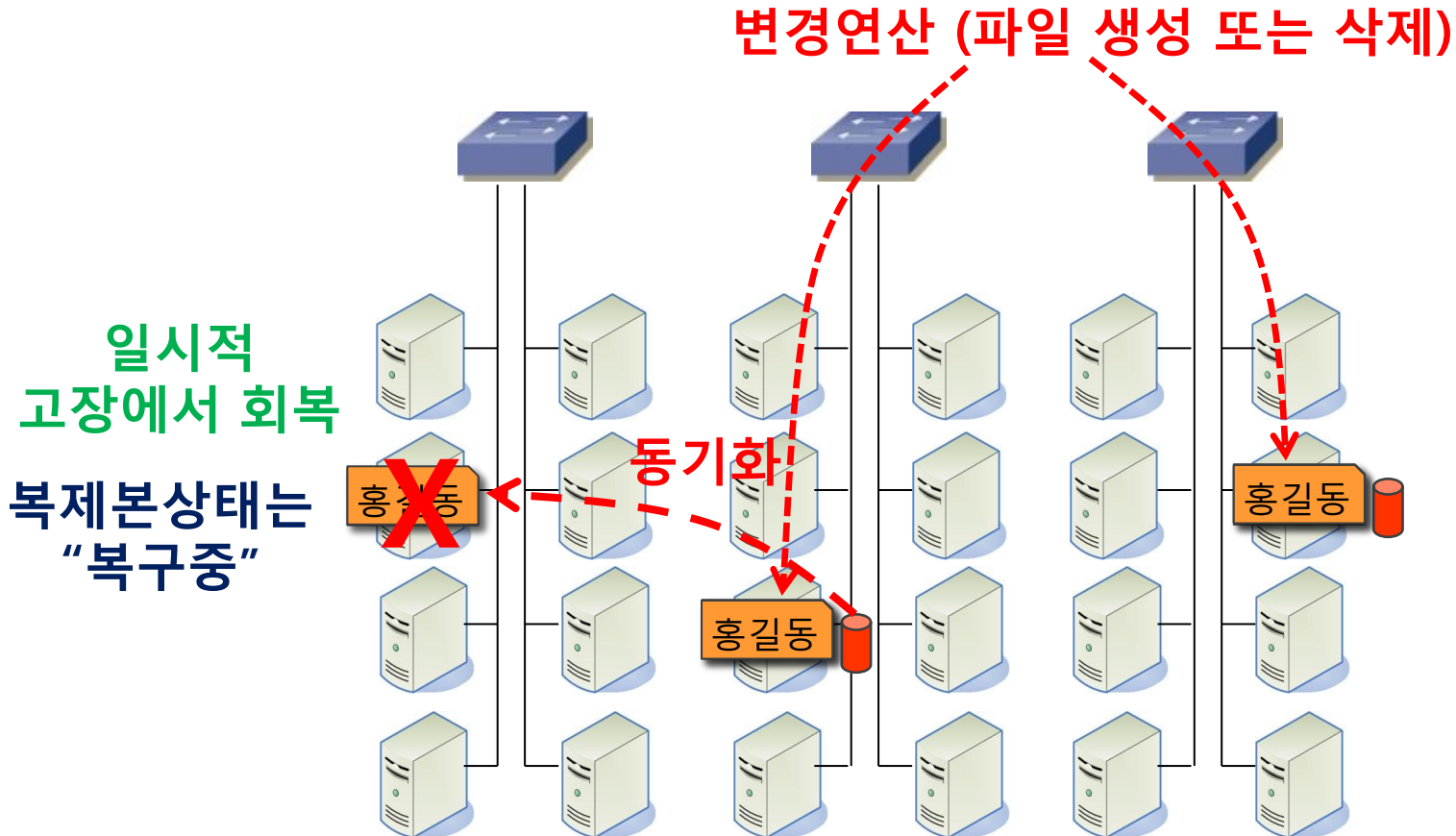


고장 유형과 복제본 동기화

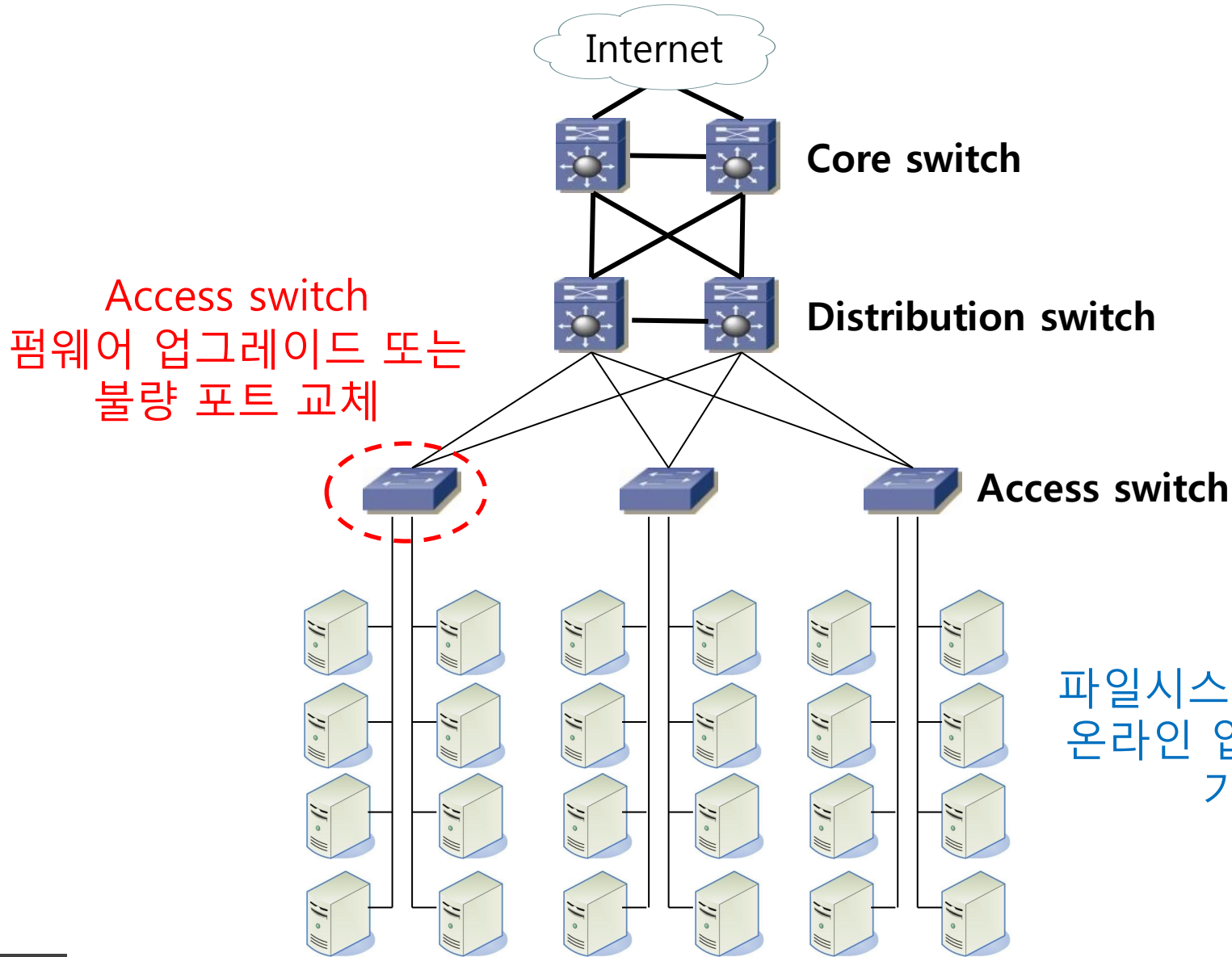
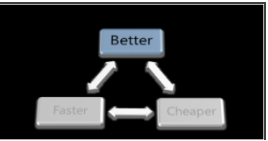


● 고장 유형

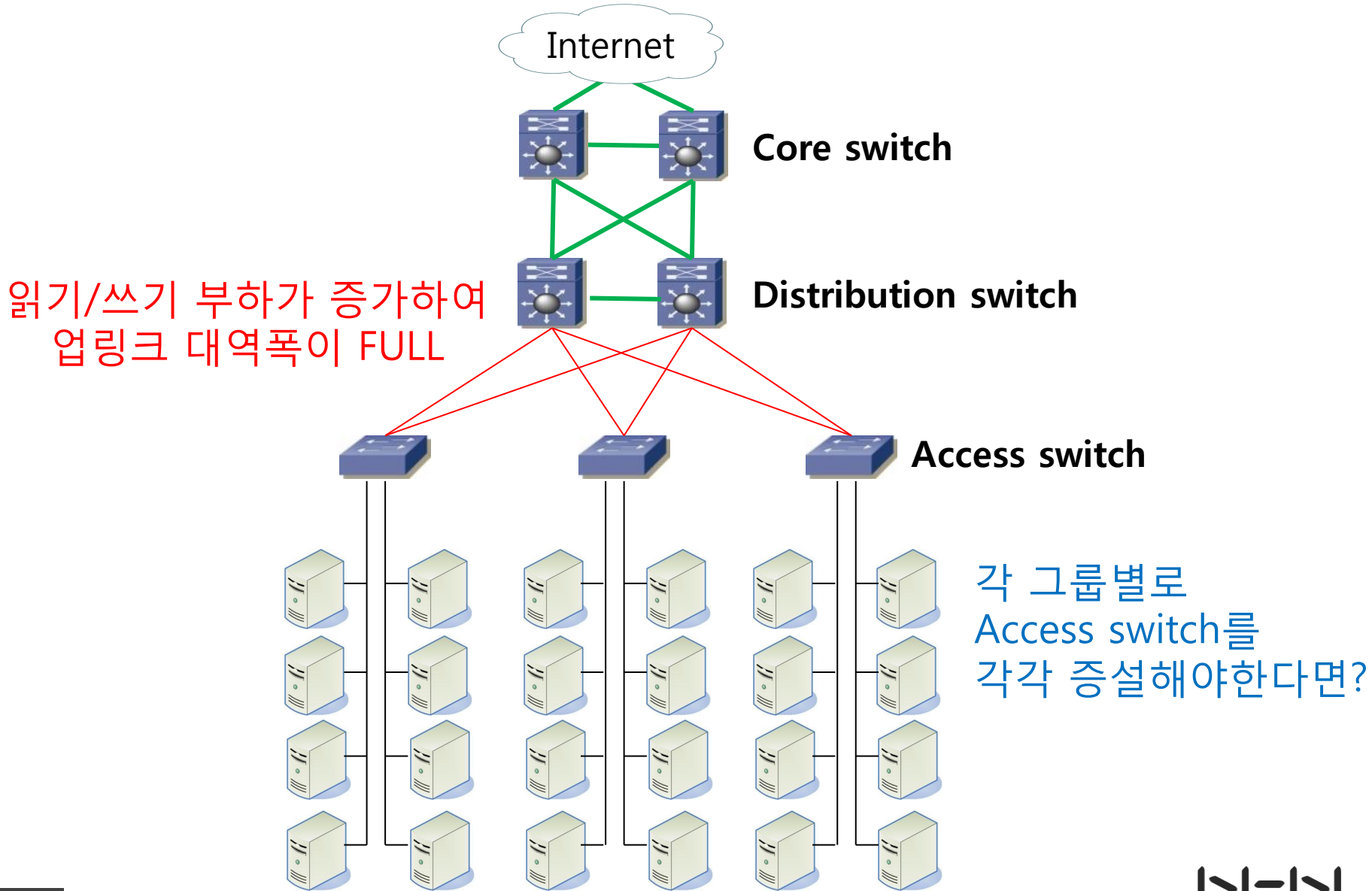
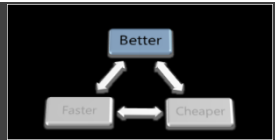
- 일시적 고장
- 영구적 고장



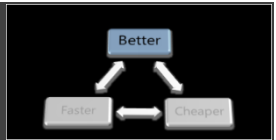
서비스 중단없이 인프라 유지보수



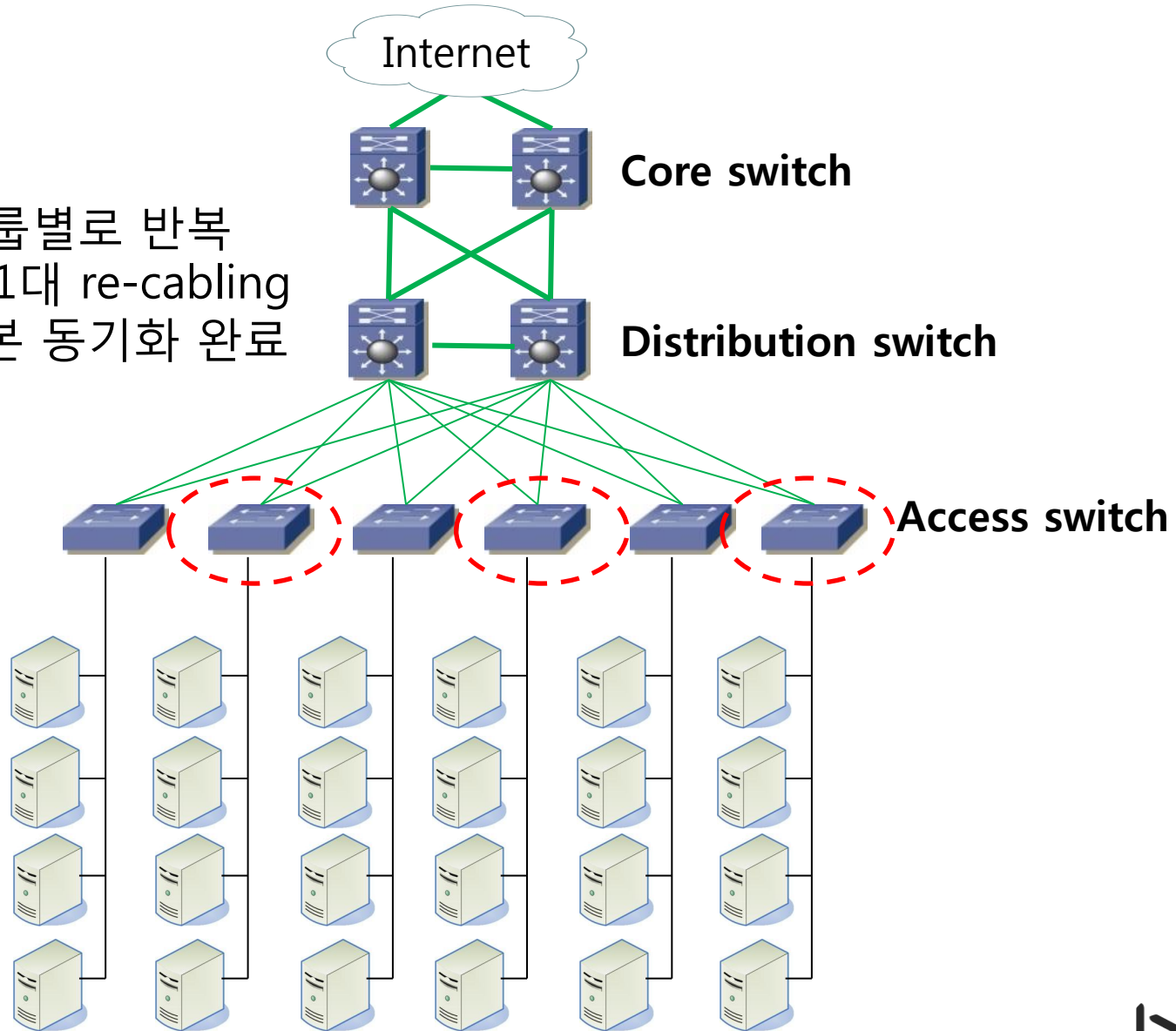
서비스 중단없이 네트워크 구성 변경 (1/2)

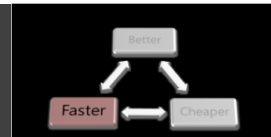


서비스 중단없이 네트워크 구성변경 (2/2)



스위치그룹별로 반복
Step1. 서버 1대 re-cabling
Step2. 복제본 동기화 완료



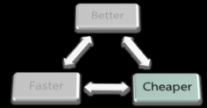


● 용량 확장성

- 저장 공간을 늘리려면 데이터 서버를 추가로 설치
- 데이터 서버가 추가되면 서버당 저장 용량 배분 작업 수행
 - 관리자에 의해 설정된 임계점에 도달하면 자동으로 용량 배분 작업이 기동됨
 - 용량 배분 작업은 데이터 서버에 추가 I/O 부담을 주기 때문에 부하 수준을 제어할 수 있는 방법 제공
- Owner에 대한 이름공간은 그대로 유지됨

● 성능 확장성

- Owner 공간의 분배가 공평하다면, 각 데이터서버는 비슷한 수준의 파일 연산 처리
- 서버를 증설하면 전체 파일연산 수와 처리량이 선형적으로 증가



- OwFS는 스토리지 운영상의 다양한 장점을 가지면서 TCO 절감도 가능
- Commodity 서버의 내장 SATA 디스크 채용
- 표준화된 서버와 네트워크 스위치 구성 관리
- 기존 네트워크 스토리지에 비해 TB당 TCO 절감

맺음말

●네이버는 안정적이고 효율적인 서비스를 위해 끊임없이 노력하고 있습니다.

●아직도 충분하지 않다!

- 디지털 데이터의 생산 속도에 보조를 맞추도록 지속적인 혁신이 필요
- 데이터 생명주기에 따른 관리
 - 롱테일 데이터
 - 보관 목적의 데이터

감사합니다.