Apache Hdaoop

Hadoop 구성 요소

- hadoop Map Reduce
- hadoop YARN
- hadoop HDFS

Hadoop1

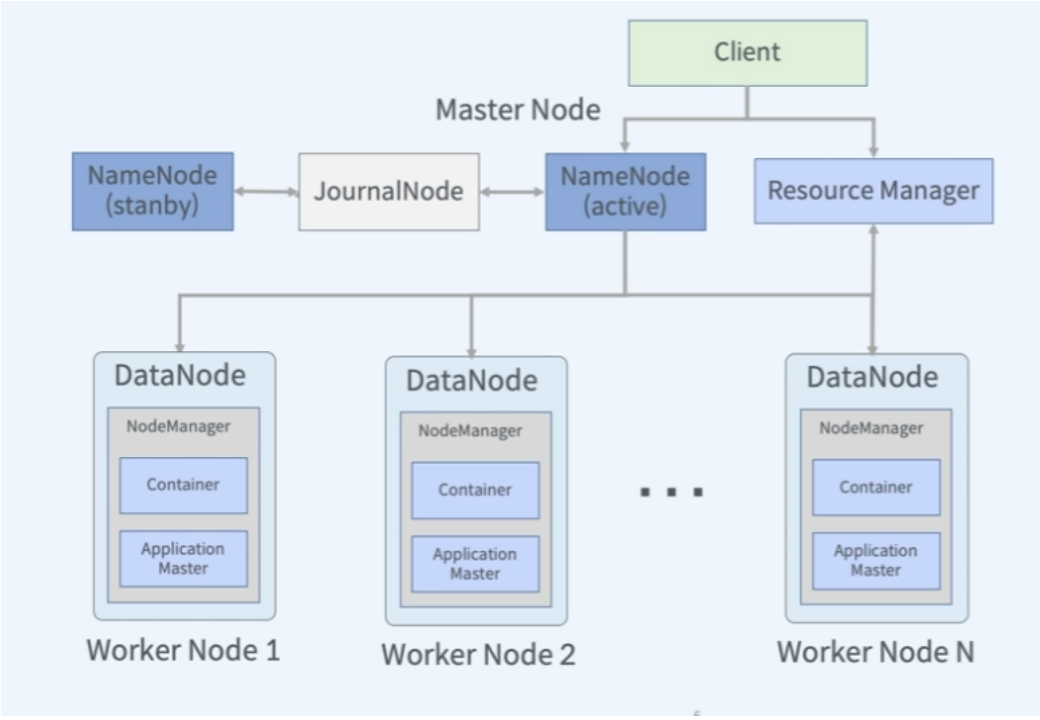
- HDFS + MapReduce
- **단점**
  - NameNode하나로 모든것을 결정
  - secondary namenode가 있지만 동기화가 느림

Hadoop2.0

- HDFS2 + YARN + MapReduce + Other
- **단점**
  - Job Tracker가 사라지고 NodeManager가 생김
  - Resource Manager는 NodeManager의 리스트를 파악하고 최적의 실행을 위한 DataNode를 찾음
  - Node Manaer의 컨테이너와 Application Manager는 다른 어플리케이션이 실행될 수 있게함

Hadoop3.0

- Erasure Coding 지원
  - 기존의 HDFS의 경우 저장 용량이 N배가 필요했다.
  - 복사본의 사이즈를 줄여서 저장 효율을 높여줌
- YARN Timeline Service v2 도입
  - 어플리케이션의 일반적인 정보 저장
  - 확장성 + 신뢰성
- Java 8
- NameNode 이중화 기능 강화



분산 파일 시스템

- 네트워크로 연결된 여러 머신의 스토리지를 관리하는 파일 시스템

HDFS의 특징

- 범용 하드웨어를 사용하여 분산 파일 시스템 구성
- 블록단위 저장
- 마스터/워커 구조
- 내고장성(Fault-tolerance)제공 (고장에 대한 내성)
- 확장성

HDFS Block

- 하나의 파일을 여러 블록으로 저장
- 하둡2에서는 기본 블록 사이즈가 128MB
- 실제 파일 크기가 블록 사이즈보다 작은 경우 파일 크기만큼만 디스크 사용

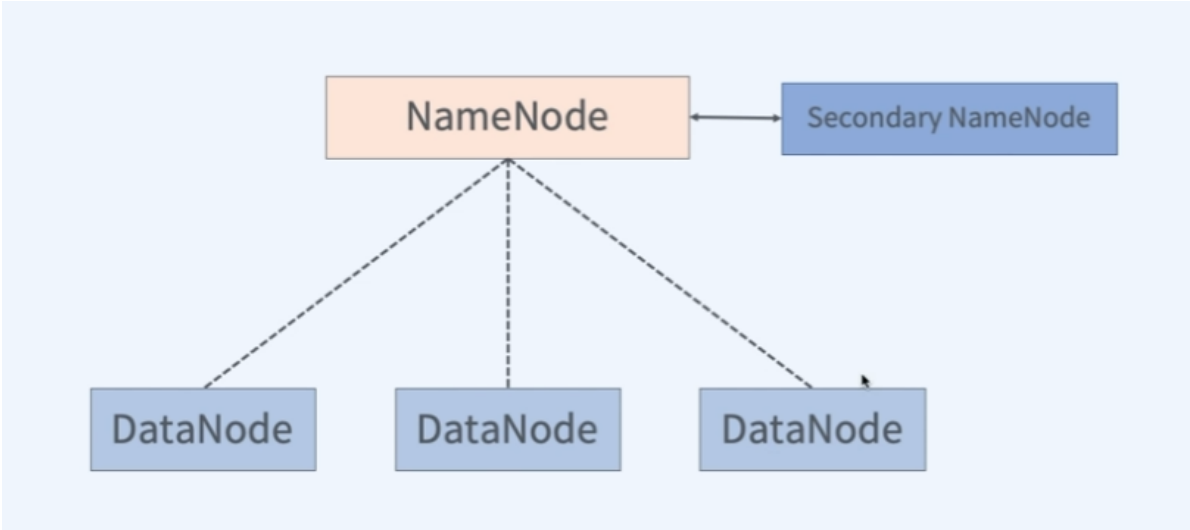
왜 HDFS Block은 클까?

- 탐색 비용 최소화
- 블록의 시작점을 탐색하는데 적게 걸림
- 메타 데이터 크기 감소

Block 단위 처리 이점

- 파일 하나의 크기가 실제 하나의 물리 디스크 사이즈보다 커질 수 있음
- 스토리지 관리 단순화
- 내고장성과 가용성을 지원하는 복제 기능 지원 적합

HDFS의 구조



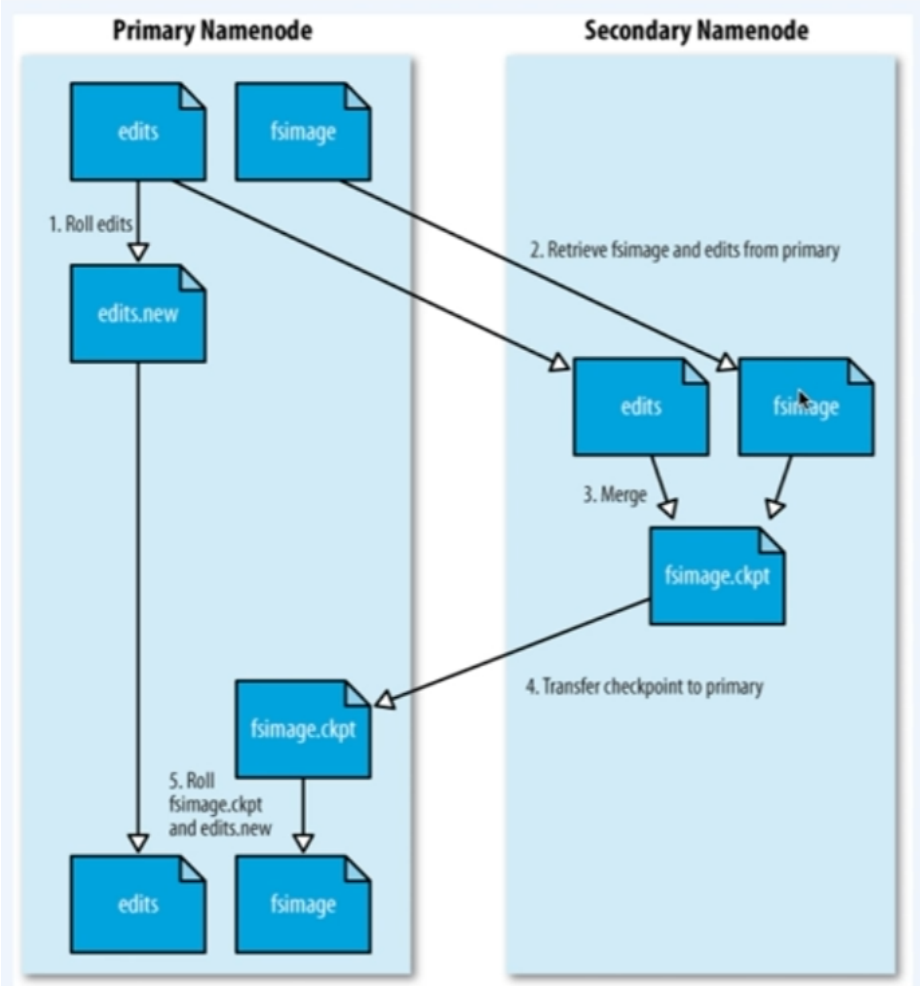
- NameNode는 파일 시스템의 메타 데이터를 가짐
- 어떤 데이터 노드에 어떤 데이터 블록들이 있는지

NameNode

- 메타 데이터 관리
  - FsImage: 네임 스페이스를 포함한 데이터의 모든 정보 (네임 스페이스 + 블록)
  - EditLog: 데이터 노드에서 발생한 데이터 변환 내역 (트랜잭션 로그)
- 데이터 노드 관리

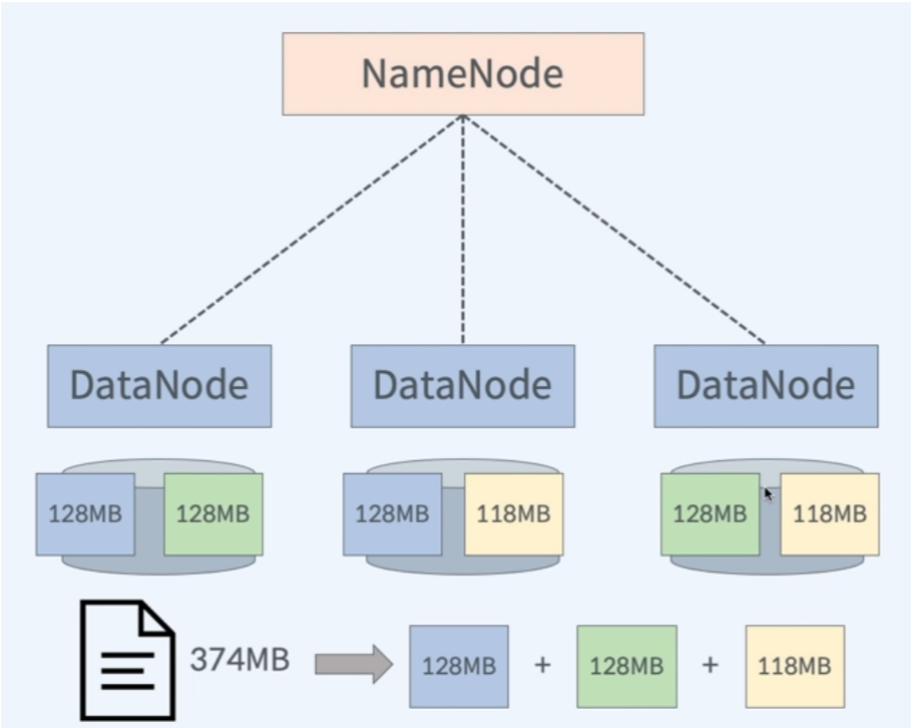
Secondary NameNode

- 체크 포인트
  - FsImage와 EditLog를 주기적으로 병합
- 주기적으로 NameNode의 FsImage를 백업

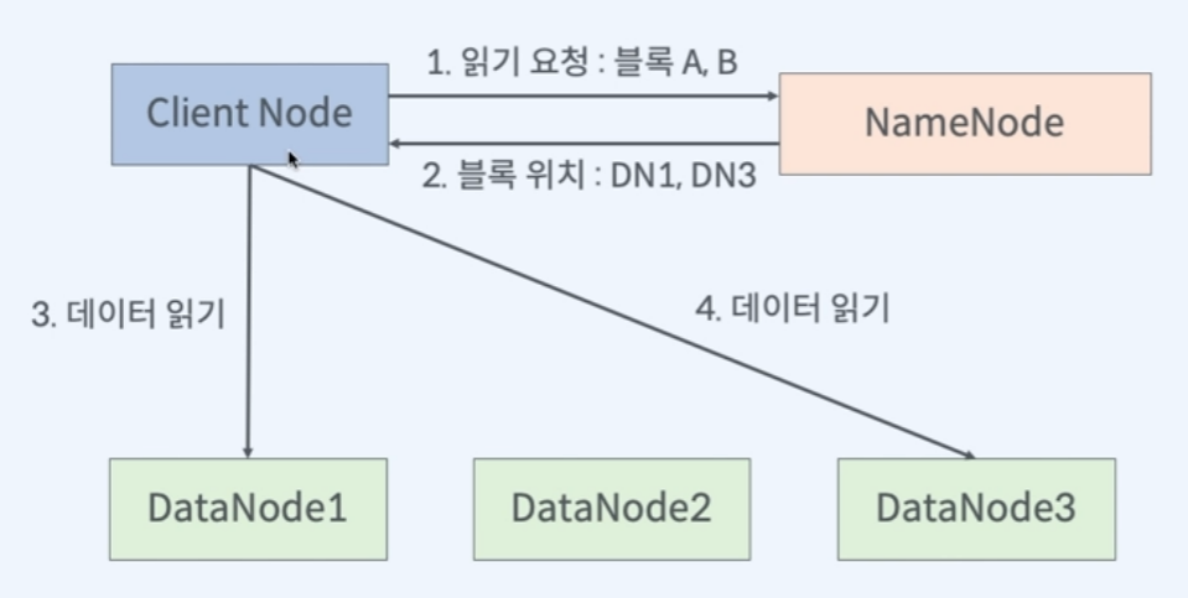


DataNode

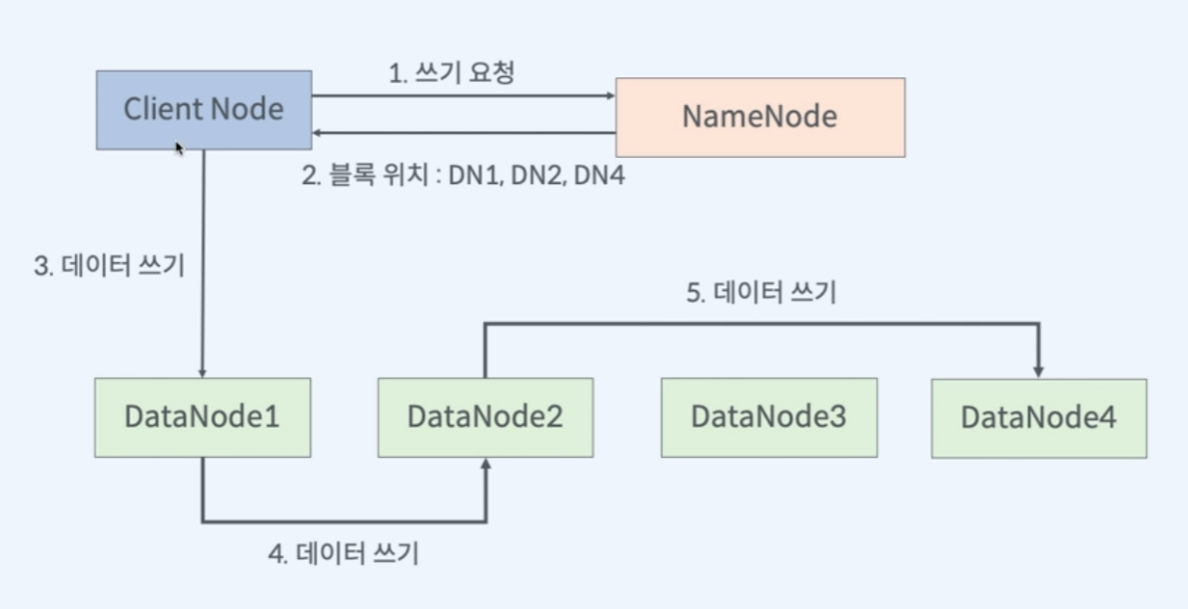
- 실제 파일을 로컬 파일 시스템에 HDFS 데이터를 저장
- 하트비트를 통한 데이터 노드 동작 여부 전달
- 저장하고 있는 블록의 목록을 주기적으로 네임ㄴ노드에 보고



HDFS 읽기 연산



HDFS 쓰기연산



HDFS의 추가 특징

- 블록 캐싱 기능 제공
  - 데이터 노드에 저장된 데이터 중 자주 읽어오는 데이터를 블록캐시로 명시적으로 지정 가능
    - 블록 단위, 파일 단위
- HDFS Federation 지원
  - 메타데이터는 메모리를 사용해 관리
  - NameSpace단위로 NameNode를 등록하여 사용하는 것
- 고가용성(HA) 지원
  - NameNode는 단일 장애지점
  - HDFS에서는 Active와 Standby NameNode 두개를 사용함으로써 이런 문제 해결