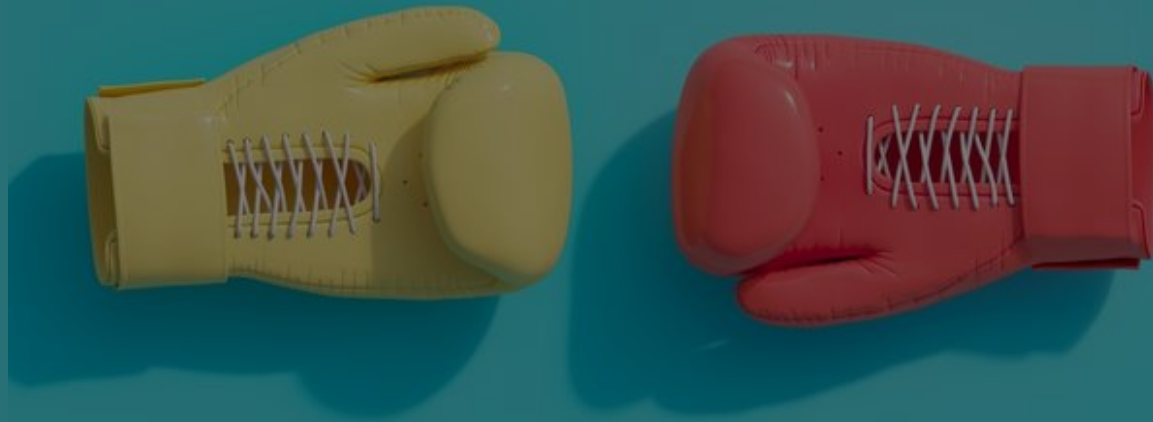


# Storage : HDFS vs S3



컴퓨터공학과 2018110646 김연수

# Contents

## 1. HDFS

- Concept / Architecture / Replication / High-Availability

## 2. S3

- concept

## 3. HDFS vs S3

- Scalability / Durability / Persistence / Price / Read Performance

# 1. HDFS

# 1. Concept

## What is Hadoop?



Hadoop is an **open source**, **Java** based framework, used for distributed storage and processing **big data**.

- Storage : HDFS
- Programming model : MapReduce

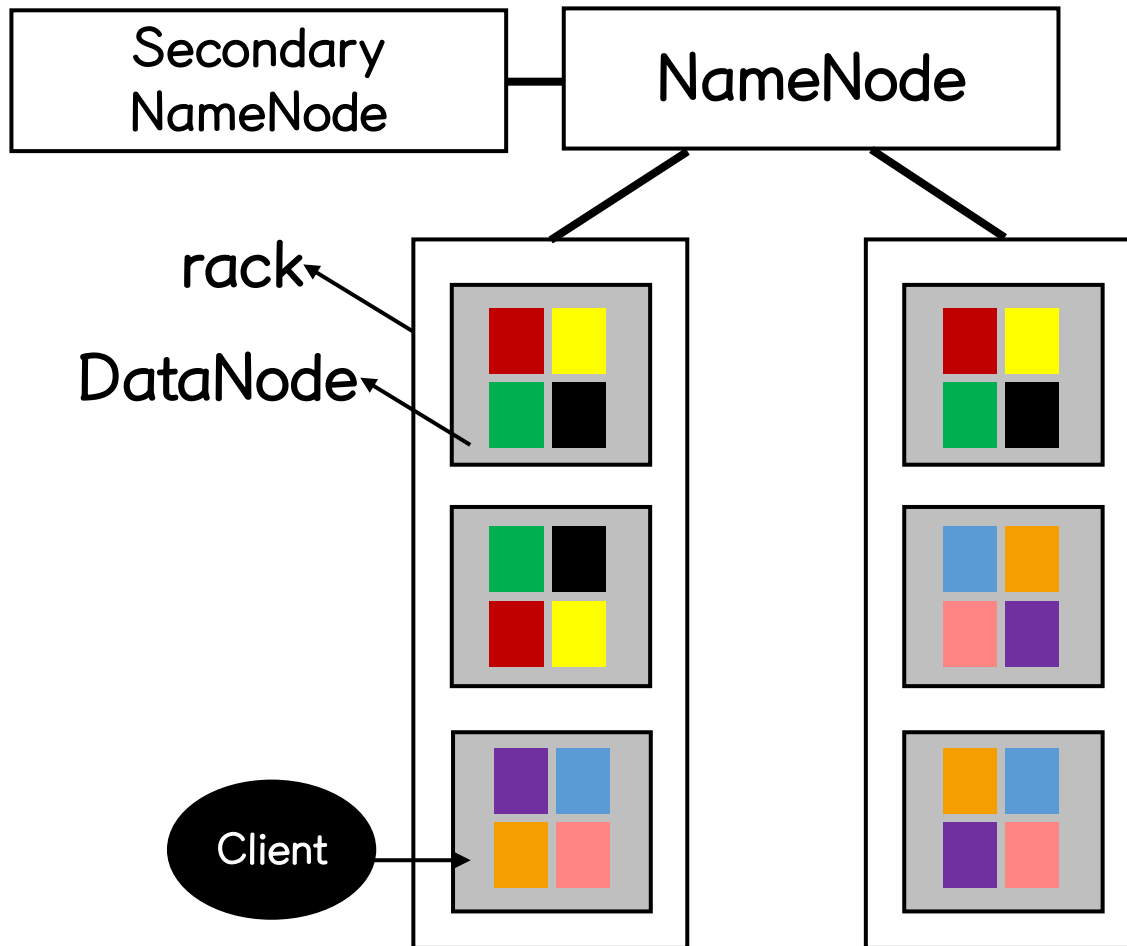
 <https://velog.io/@kimdukbae/MapReduce>

# 1. Concept

## What is HDFS?

- Store very **large** files in **commodity** hardware
- Data access patterns : **Streaming**
- Data **integrity** : write-once, read-many-times
- Failure **recovery**

## 2. Architecture



### ■ Replica Policy

- Client
- other rack
- other rack and other datanode

### ■ High availability

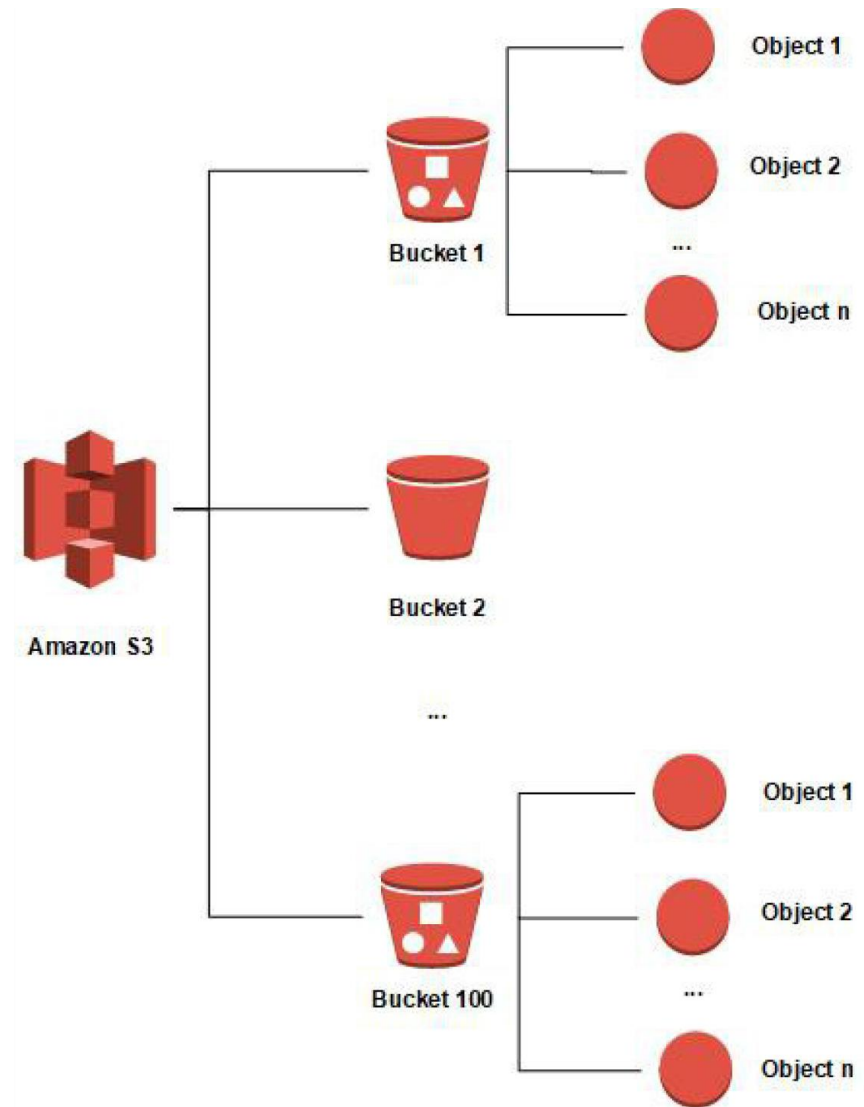
- Secondary NameNode
- Active-standby configuration

2. S3

# 1. Concept

## What is S3?

- S3 : Simple Storage Service
- Object Storage
  - Object
  - Bucket
- version control

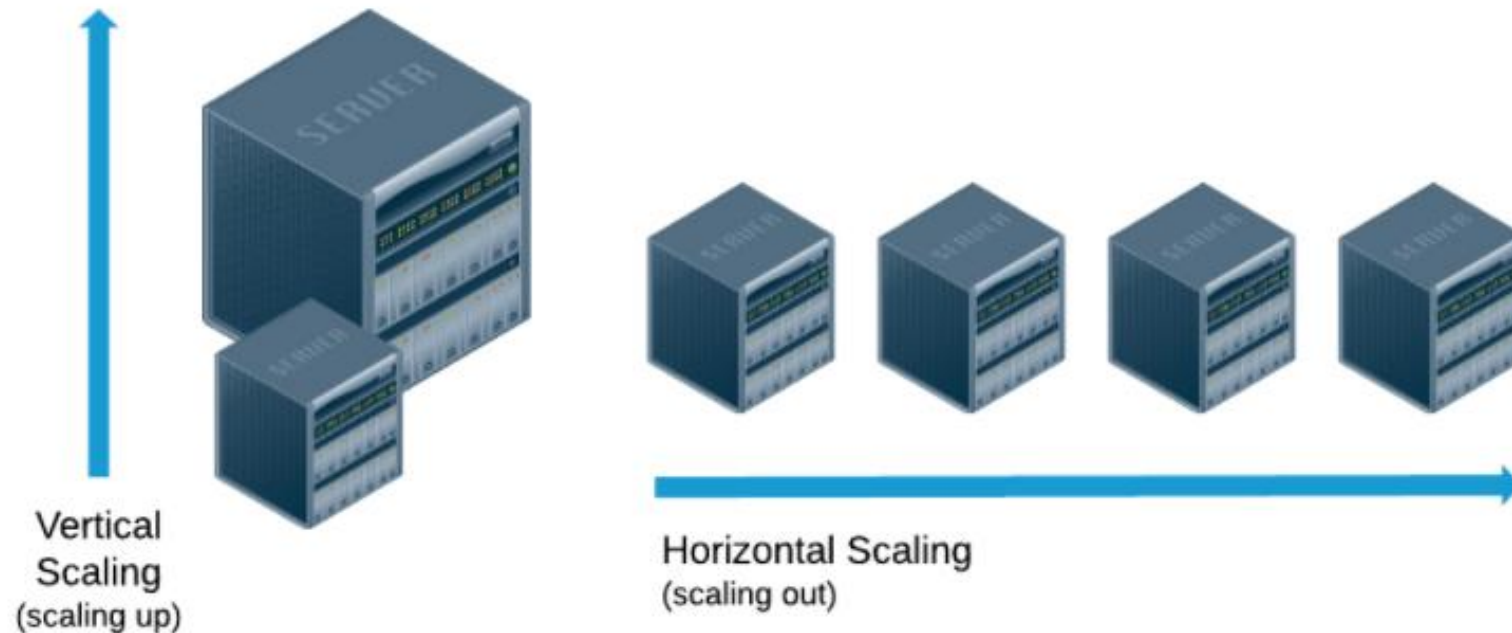




# 3. HDFS vs S3

# 1. Scalability

**Scaling** : Ability to handle a growing amount of work in a capable manner



- **Scale up** : the ability to grow by using stronger hardware
- **Scale out** : the ability to grow by adding more hardware

# 1. Scalability

trait where a software solution can handle increased loads of work.

## HDFS

- scales **horizontally**.
- This is **feasible**, but more **complicated** than S3.

## S3

- scales **vertically**
- **automatic** : without any need for action on your part.
- **infinite** amount of space available.

♥ S3 is better than HDFS in Scalability

## 2. Durability

ability to keep your information **intact long-term** in cloud data storage, without suffering bit rot or corruption.

### HDFS

- No support for **multiple writers** or for **modification**

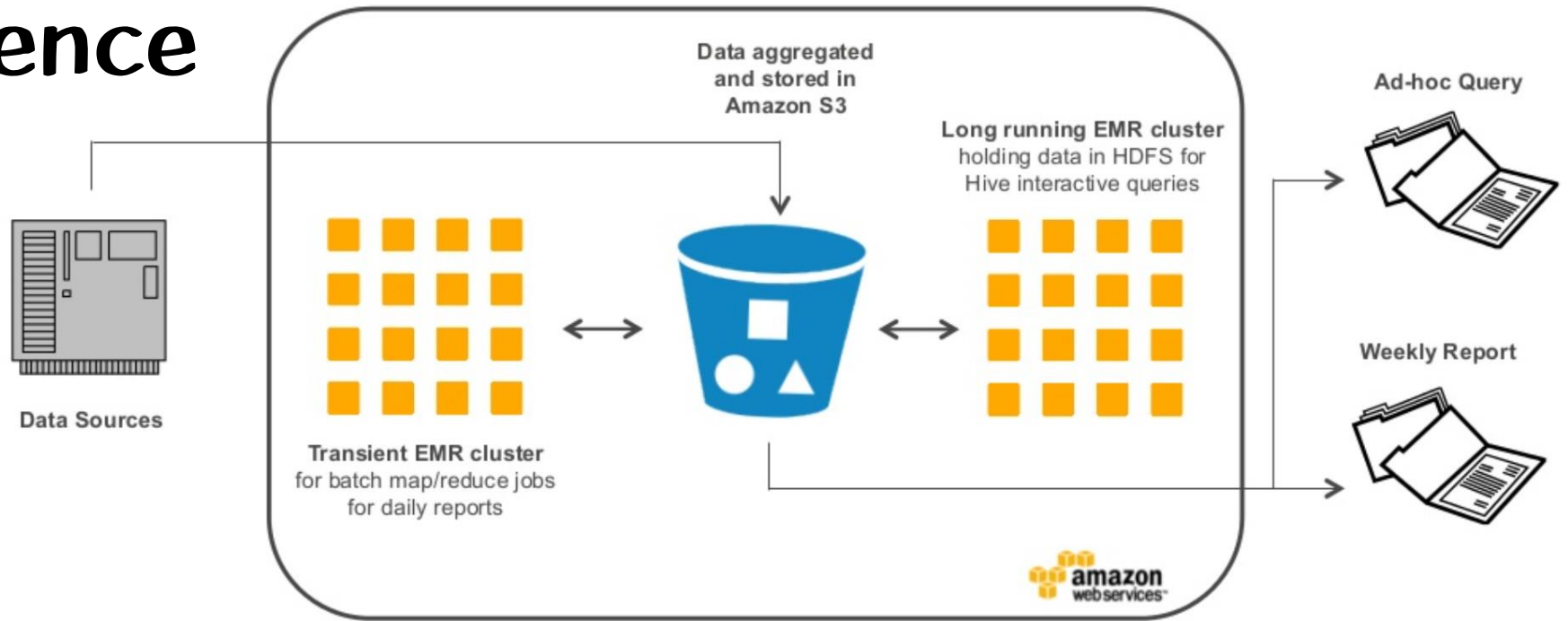
<b>Probability of losing a block on a large (4000 nodes) cluster</b>	
In the next 24 hours	$5.7 \times 10^{-7}$
In the next 365 days	$2.1 \times 10^{-4}$

### S3

- Using **checksum** and CRC
- durability of 99.999999999% of objects per year.
- This means that a single object could be lost per 10,000,000 objects once every 10,000 years

♥ S3 is more durable than HDFS.

# 3. Persistence



## HDFS

- data **doesn't persist** when stopping EC2 or EMR instances.

## S3

- data is **always persistent** in S3—simple as that.

♥ If you want to access the data anytime and from anywhere, use S3

# 4. Price

## HDFS

- costs **\$103/month** for 1TB of data.
- Why expensive? preserve data integrity, HDFS stores three copies of each block of data by default. So, HDFS requires **triple the amount of storage** space for your data—and therefore triple the cost.

## S3

- costs **\$23/month** for 1TB of data.
- Why cheap? pay for **only the storage** that you actually need.
- S3 also supports storing **compressed** files, which can help slash your storage costs.

 storage cost alone, **S3 is 5X cheaper than HDFS.**

## 4. Performance

$$T(n) = a \cdot n + b$$

- $T(n)$  : total time to access
- $n$  : number of data
- $a$  : throughput
- $b$  : latency



# 4. Performance

## 1) throughput

### HDFS

- **Block** : parallel processing
- High throughput : 3GB/node local read throughput (e.g. i2.8xl, roughly 90MB/s per core)
- on a per node basis, HDFS can yield 6X higher read throughput than S3

### S3

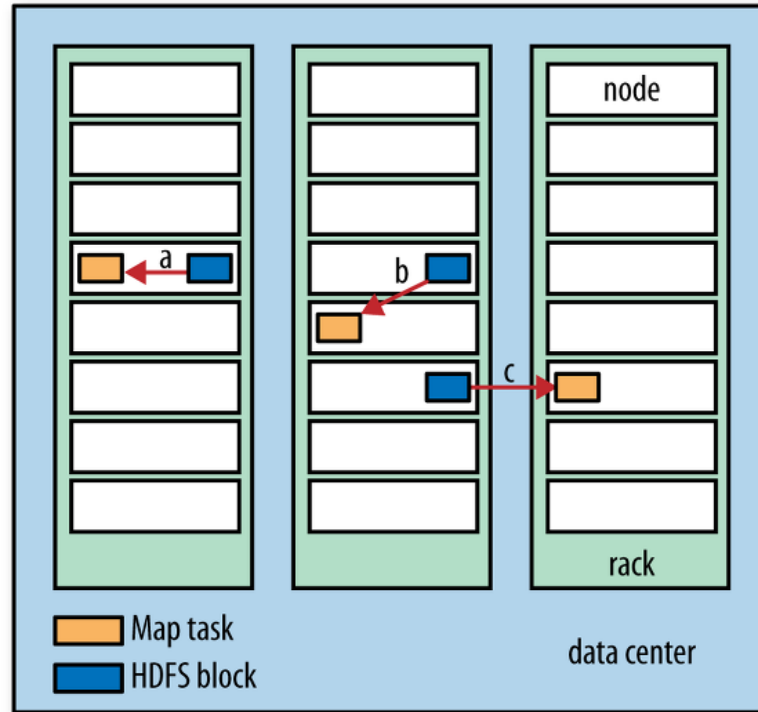
- **Object**
- Not high as HDFS : 600MB/s read throughput on i2.8xl (roughly 20MB/s per core)



# 4. Performance

## 2) Latency

### HDFS



#### < Locality >

- A : data local
- B : rack local
- C : off local

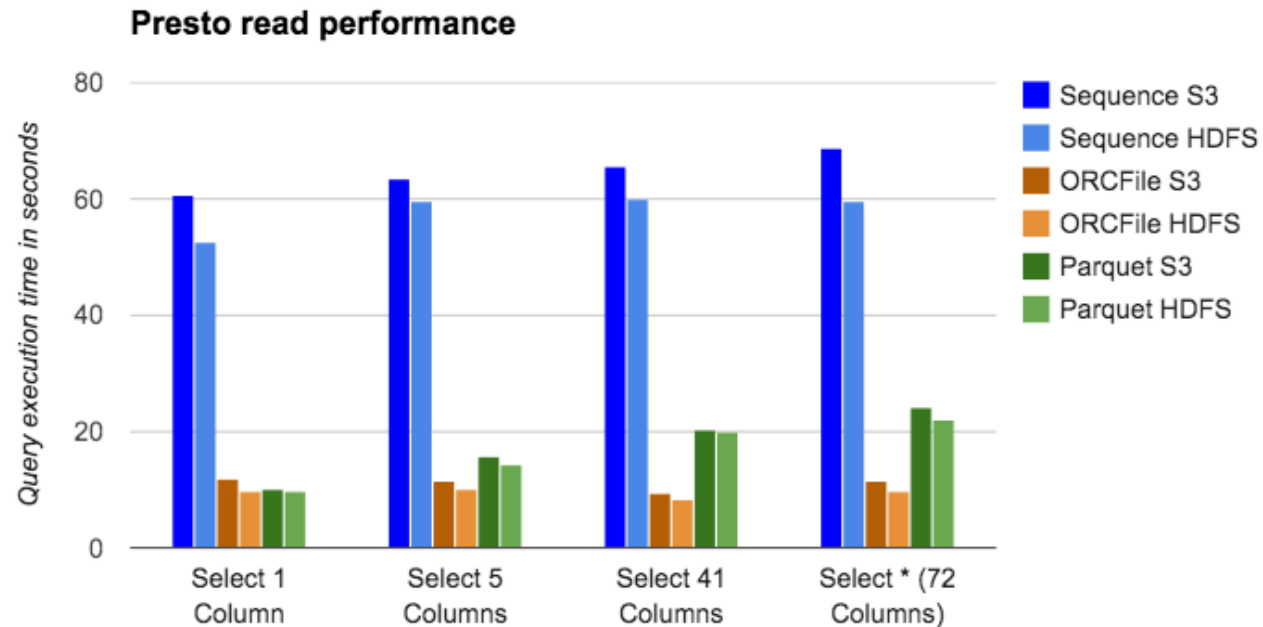
## S3

- Why latency is high? all reads need to transfer data across the network

💖 performance per dollar is better in S3. **but...!**

# 4. Performance

## 3) conclusion



S3 : cloud service

- network bandwidth is low (bottleneck)
- not suitable for handling bigdata
- solution : **use HDFS for data locality** > high bandwidth, low cost

# 4. Performance

## 3) conclusion

	HDFS on Ephemeral Storage	Amazon S3
Read	350 mbps/node	120 mbps/node
Write	200 mbps/node	100 mbps/node

## HDFS

- HDFS performance is excellent.
- Because data is stored and processed on the same machines, access and processing speed are lightning-fast.

## S3

- S3 doesn't perform as well as HDFS. The latency is obviously higher and the data throughput is lower.



HDFS is strong all-round performance.

Thank You 😊