빅데이터 실무

강사 : 송재성 (zzyzigy@naver.com)

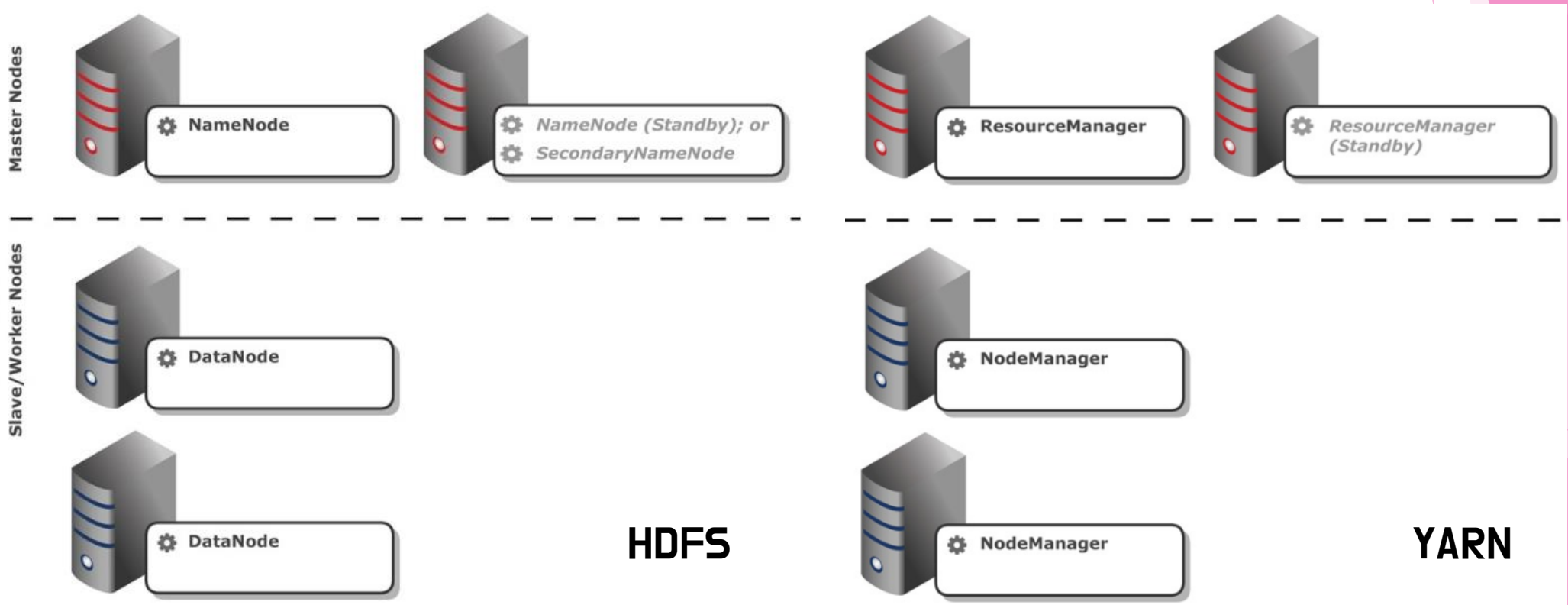
학습목표

- 하둡 소개
- HDFS 아키텍처
- MapReduce 아키텍처
- YARN 아키텍처
- 단일/의사분산/완전분산 모드 하둡 설치

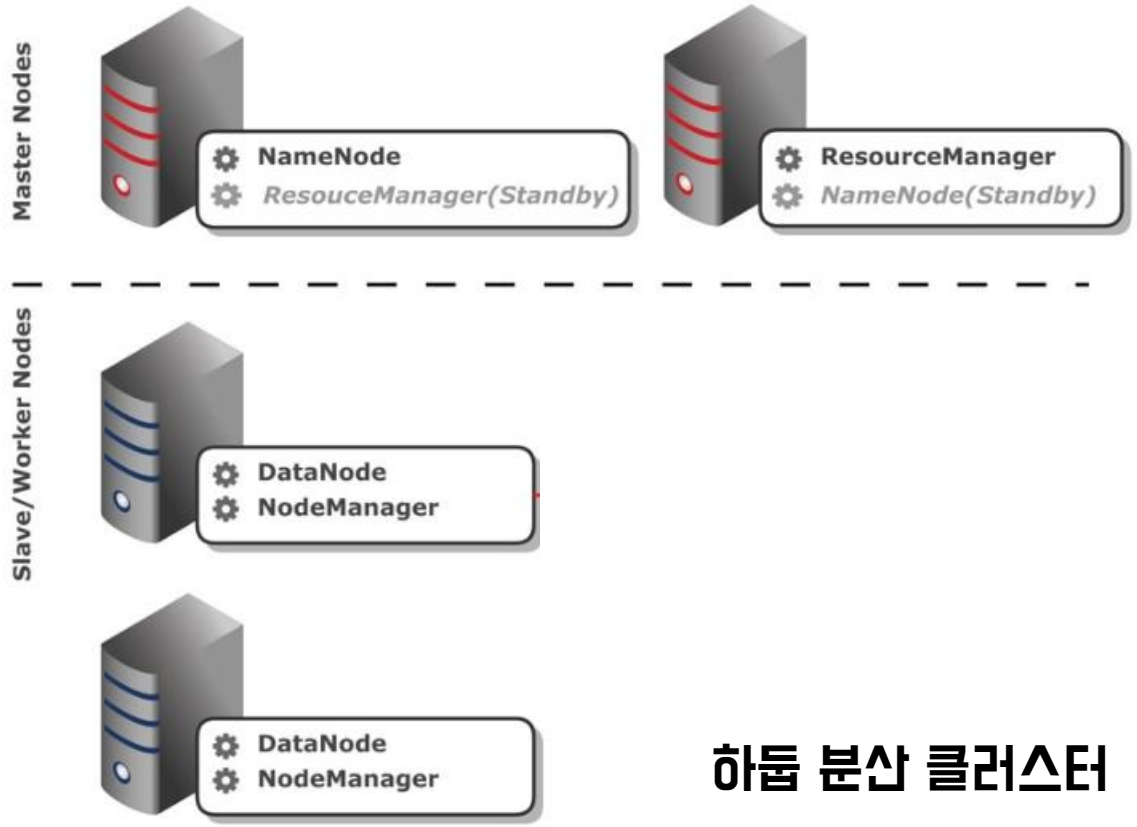
하둡

- 너무나도 잘 알려진 빅데이터 핵심기술 - 빅데이터 생태계는 대부분 하둡에 의존적
- 대용량 데이터 처리를 위해 분산 병렬처리 기술 사용
- 구글이 2003년 'Google File System', 2004년 'MapReduce'에 관한 논문 발표하며, 구글 검색엔진 기술 공개
- 이 논문을 기반으로 더그 커팅이 2005년 넛치/루씬 검색엔진의 서브 프로젝트로 진행하면서 하둡 프로젝트를 시작

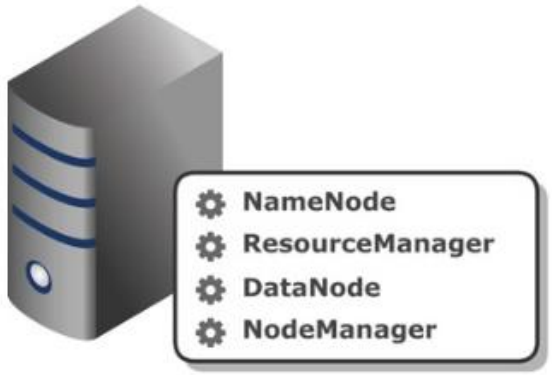
클러스터


Master Nodes
NameNode
NameNode (Standby); or
SecondaryNameNode
ResourceManager
ResourceManager
(Standby)
Slave/Worker Nodes
DataNode
DataNode
NodeManager
NodeManager
HDFS
YARN

클러스터

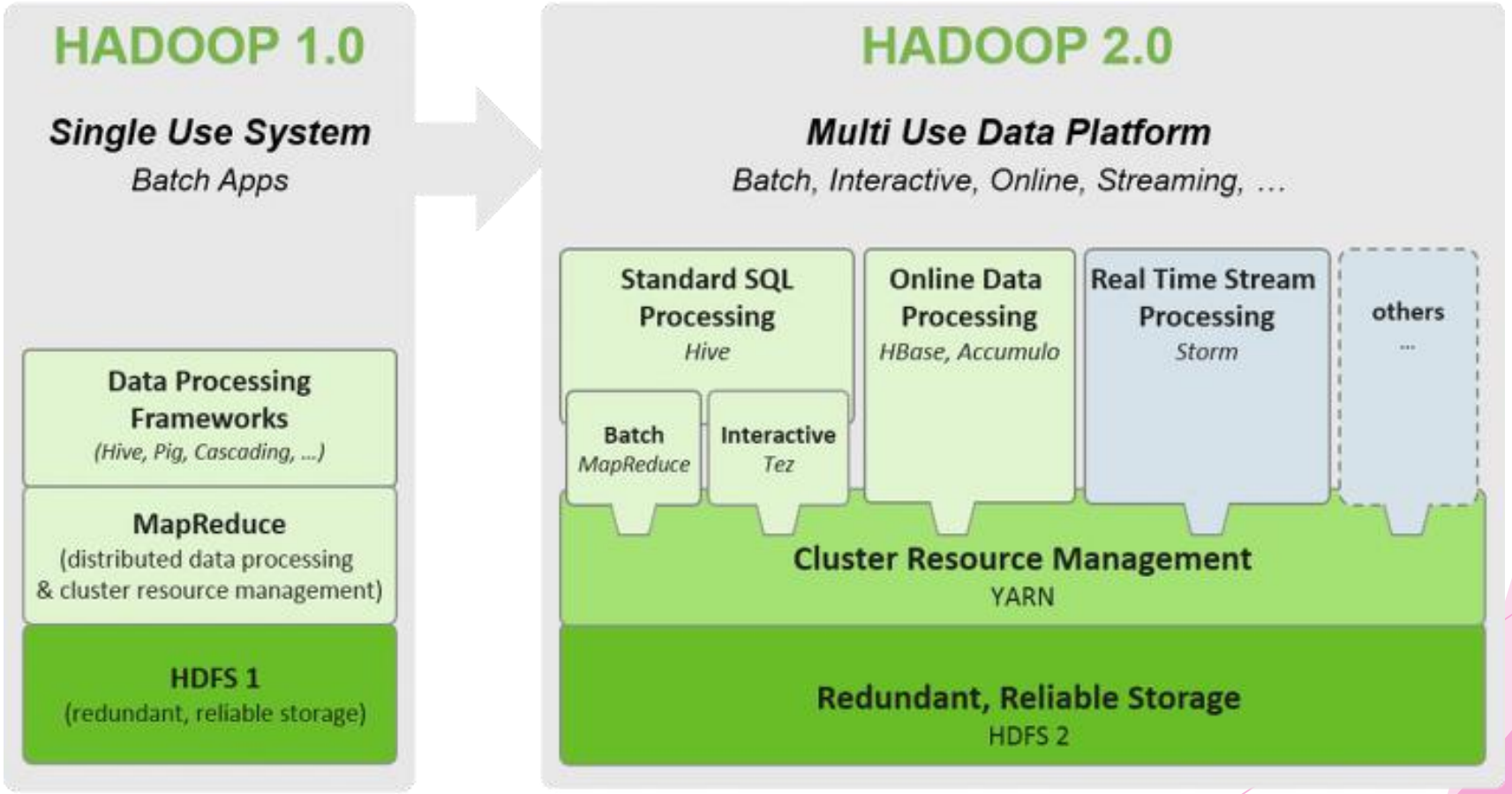


하둡 분산 클러스터

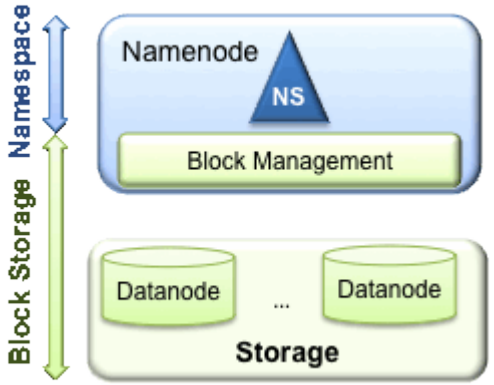


하둡 가상 분산 클러스터

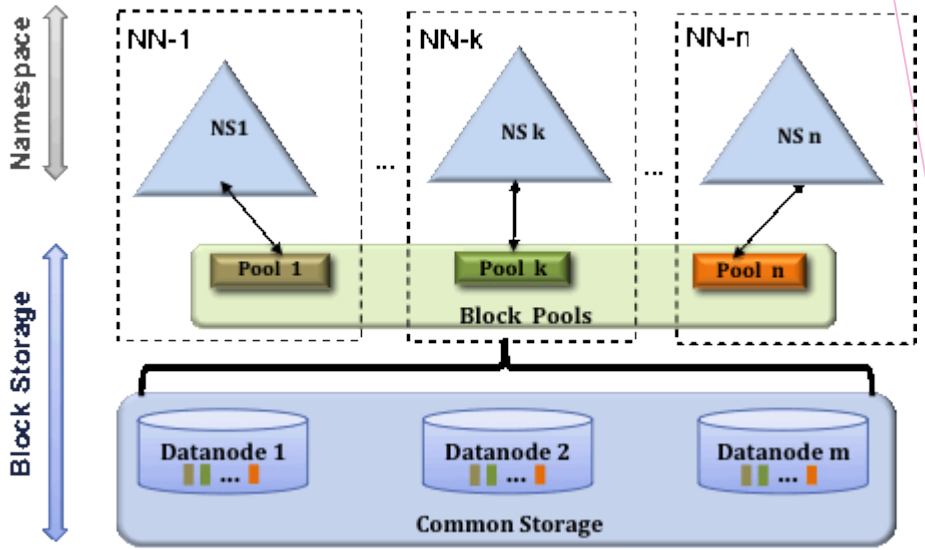
하둡 1 vs 2


HADOOP 1.0
Single Use System
Batch Apps
Data Processing Frameworks
(Hive, Pig, Cascading, ...)
MapReduce
(distributed data processing & cluster resource management)
HDFS 1
(redundant, reliable storage)
HADOOP 2.0
Multi Use Data Platform
Batch, Interactive, Online, Streaming, ...
Standard SQL Processing
Hive
Batch
MapReduce
Interactive
Tez
Online Data Processing
HBase, Accumulo
Real Time Stream Processing
Storm
others
...
Cluster Resource Management
YARN
Redundant, Reliable Storage
HDFS 2

하둡 1 vs 2

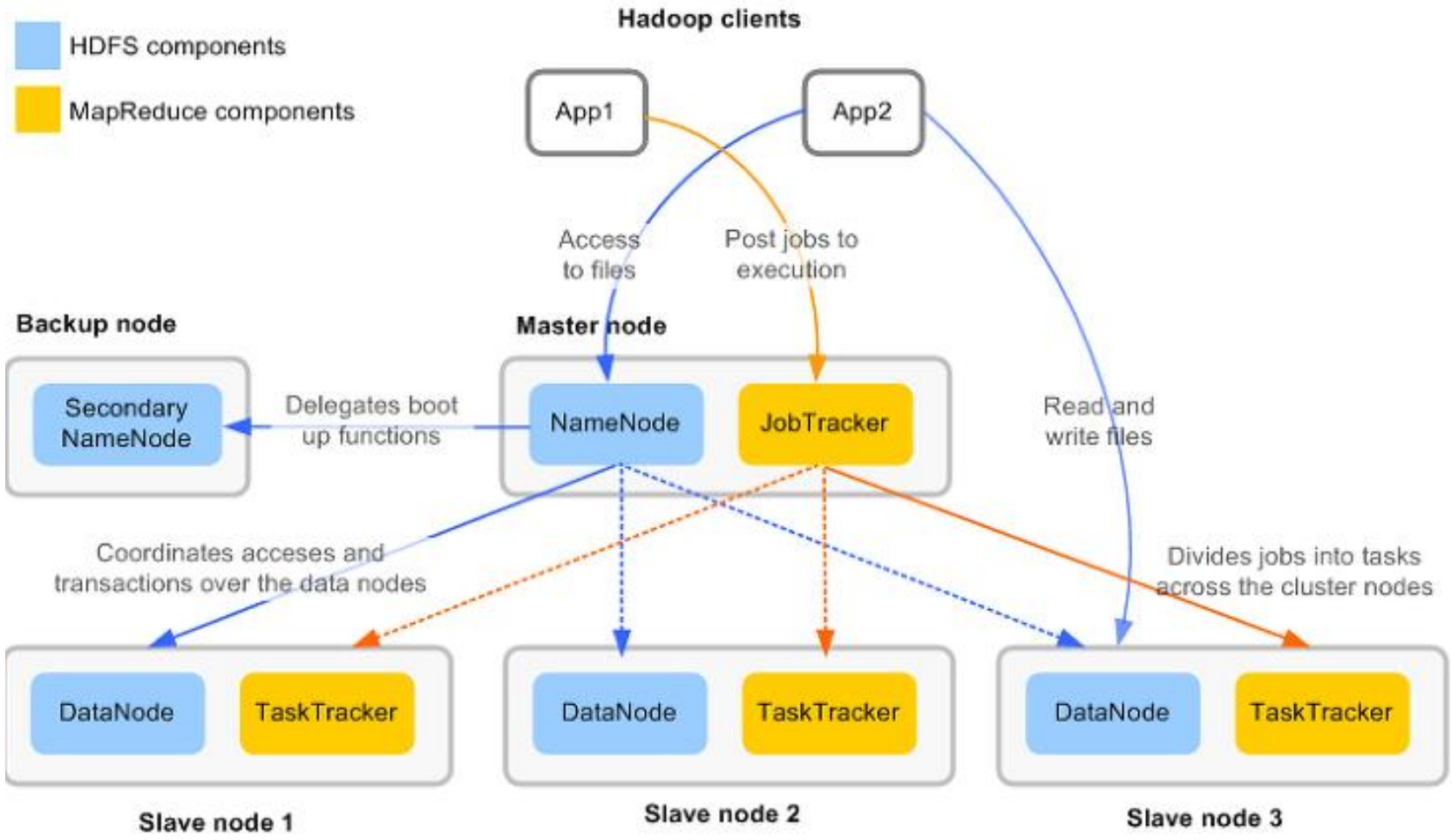


네임노드 이중화 미지원
즉, 네임노드에 문제가 생기면
하둡 클러스터 전체에 장애 발생
최대 4,000개 노드 지원



독립적인 네임노드 여러 대 운용 가능
단, 네임노드에 active/standby 상태 도입
클러스터 내 모든 네임노드에 동기화 지원
최대 10,000개 노드 지원

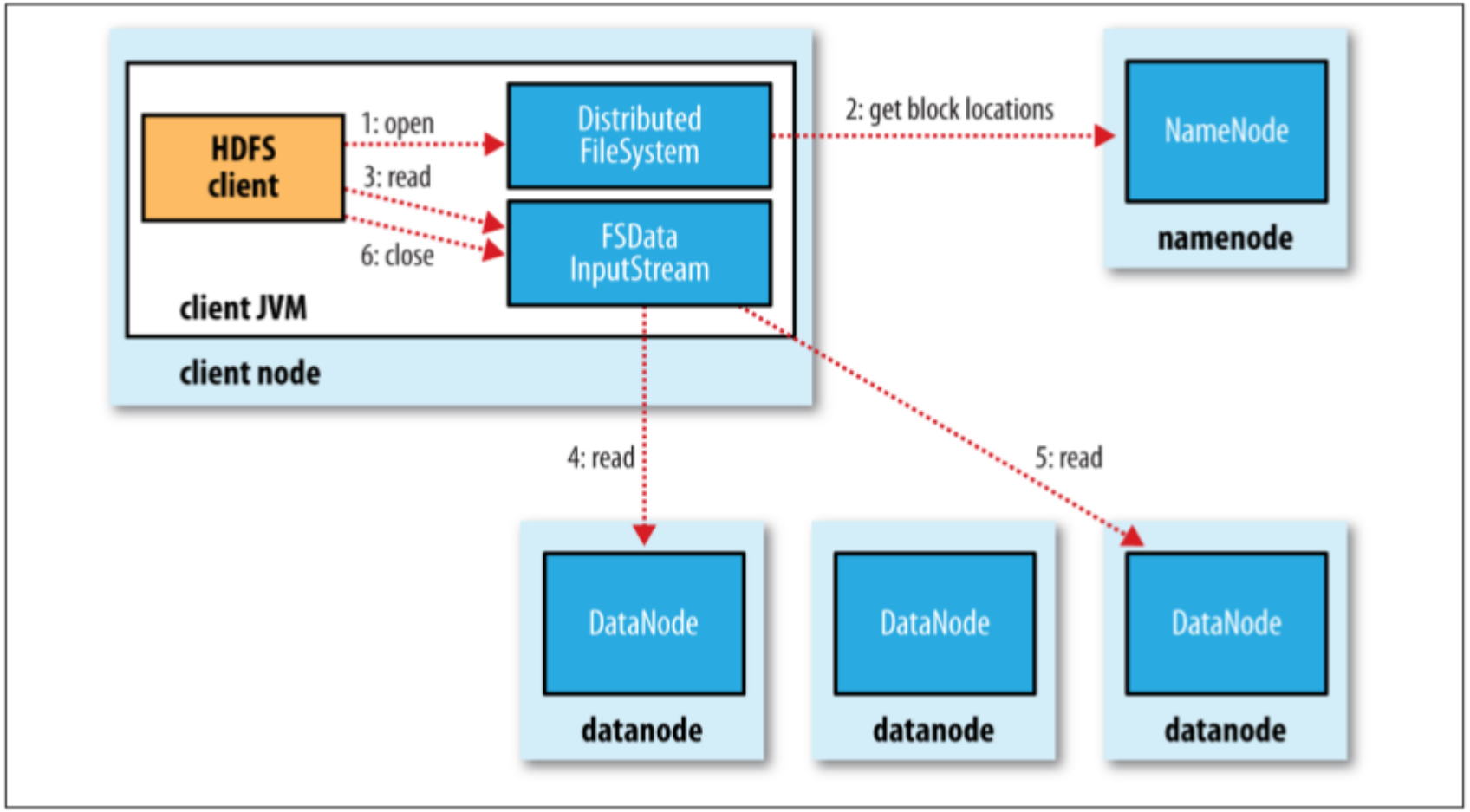
MapReduce & HDFS



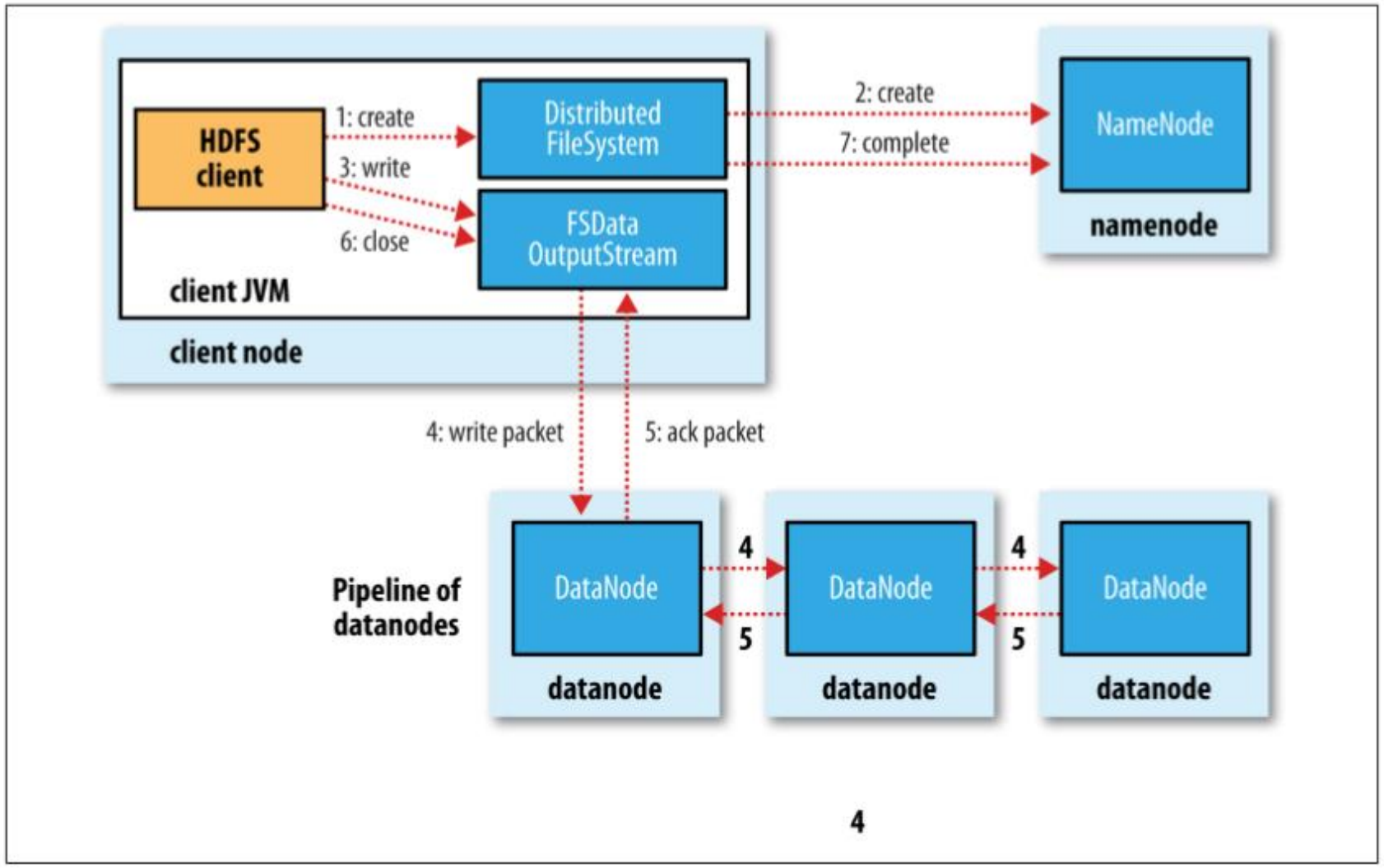
HDFS

- 하둡의 분산 저장소 컴포넌트
- 구글의 GFS 논문에 따라 모델링
- 기존 파일 시스템과 달리 큰 블록크기를 통한 최적화를 활용해 네트워크 입출력을 줄여준다
- 저가의 하드웨어로도 구현이 가능

HDFS 아키텍처


HDFS client
1: open
3: read
6: close
Distributed FileSystem
FSData InputStream
client JVM
client node
2: get block locations
NameNode
namenode
4: read
5: read
DataNode
datanode
DataNode
datanode
DataNode
datanode

HDFS 아키텍처


HDFS client
1: create
3: write
6: close
Distributed FileSystem
FSData OutputStream
client JVM
client node
2: create
7: complete
NameNode
namenode
4: write packet
5: ack packet
Pipeline of datanodes
DataNode
DataNode
DataNode
4
5
4
5
datanode
datanode
datanode
4

HDFS 구성요소 - namenode

- 데이터노드에 저장된 파일들의 메타정보를 관리
- 메타정보는 파일시스템이미지와 에디트로그로 구성
- FSImage는 HDFS의 이름영역(디렉토리명, 파일명, 상태 정보)과 파일에 대한 매핑 정보를 저장하는 파일
- Edit log는 HDFS의 메타정보에 대한 모든 명령어들을 기록하는 로그 파일
- 데이터노드에 입출력과 관련된 작업을 지시

HDFS 구성요소 - datanode

- 대용량 데이터를 블록(64/128MB) 단위로 분할해서 디스크에 분산 저장하고 관리
- 블록단위 기록을 위해 각각의 데이터노드들은 서로 통신
- 파일에 대한 각 블록은 서로 복제 (기본:3)

HDFS 구성요소 - secondary namenode

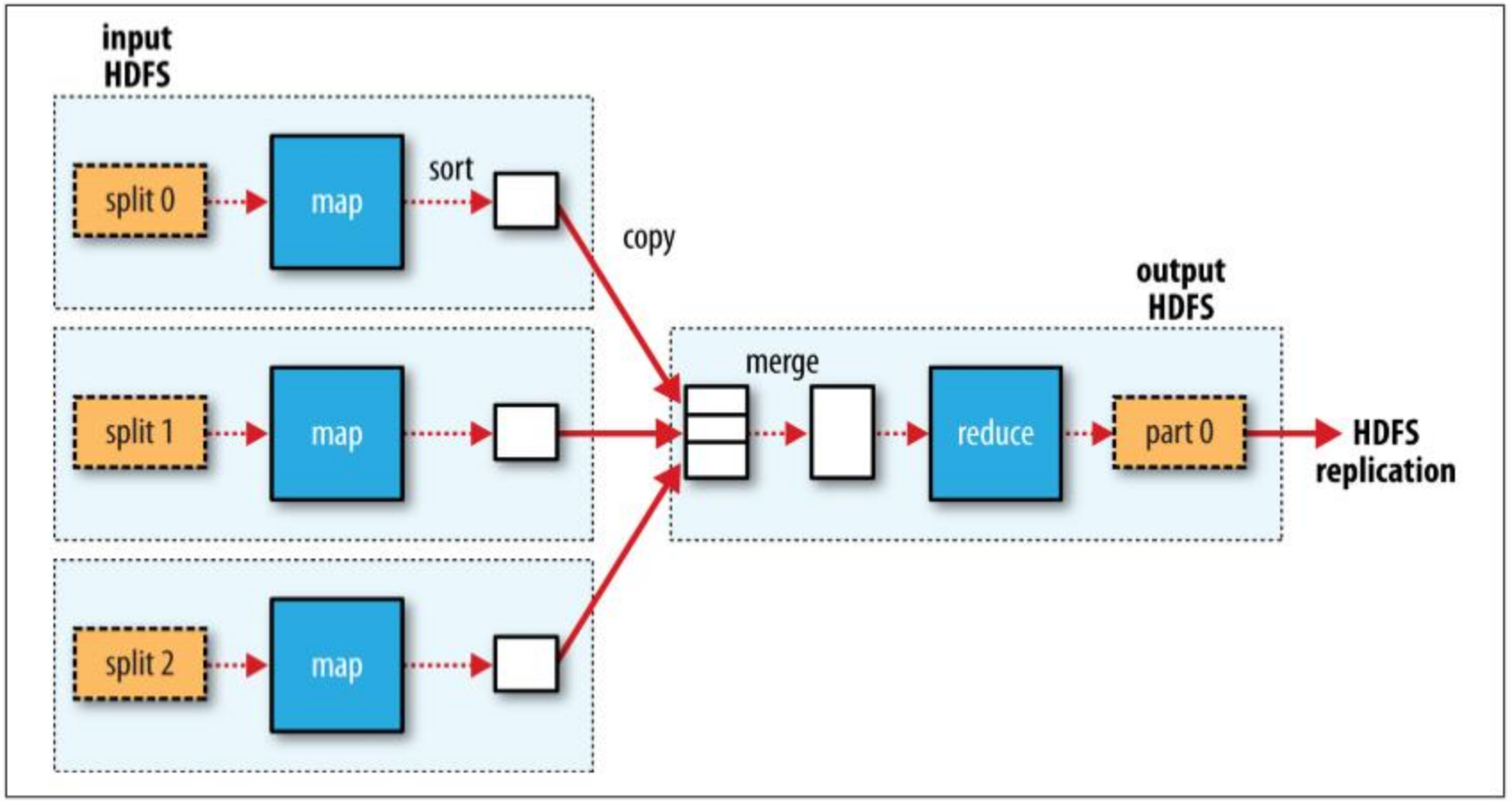
- 주 네임노드의 FSImage에 EditLog를 합쳐 주기적으로 새로운 이미지를 만드는 일을 담당
- 즉, 네임노드의 EditLog를 축소시켜 FSImage에 갱신 저장하는 역할
- 이것은 일정시간 주기로 발생하는데 이러한 작업을 Check point라 함

하둡 MapReduce 아키텍처

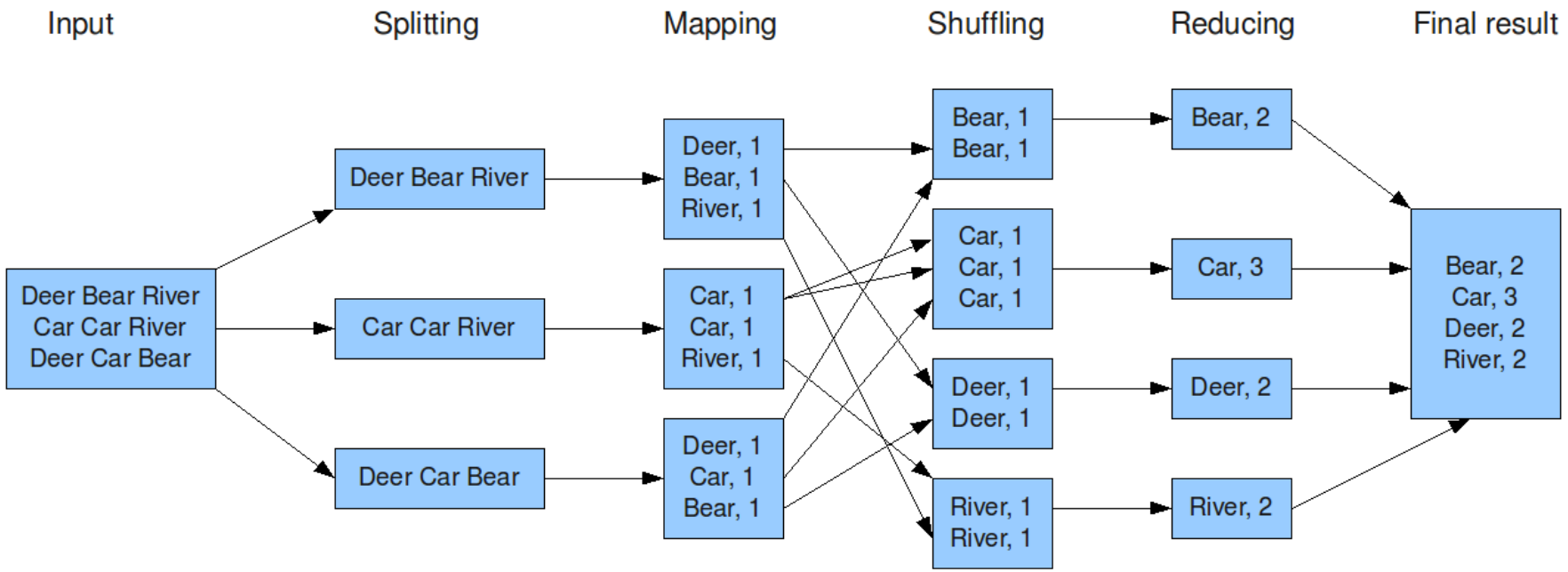
- 분산 클러스터 환경에서 <Key, Value> 구조를 이용해서 대규모 데이터를 처리하기 위한 **병렬 컴퓨팅 프레임워크**

- 즉, 데이터를 여러조각으로 **분할**한 뒤 분산 노드에서 <Key, Value> 의 구조로 데이터를 **병렬처리**하고 이를 다시 새로운 <Key, Value> 구조로 **가공**하여 출력

하둡 MapReduce 아키텍처



하둡 MapReduce 아키텍처



하둡 MapReduce 아키텍처

- 대용량 데이터를 분산처리하기 위해 분산 처리 프레임워크
- Terabyte, Petabyte이상의 대용량 데이터를 저렴한 x86 서버를 클러스터링해 분산 처리
- MapReduce에서 데이터를 처리하는 기본 단위는 Mapper와 Reduce

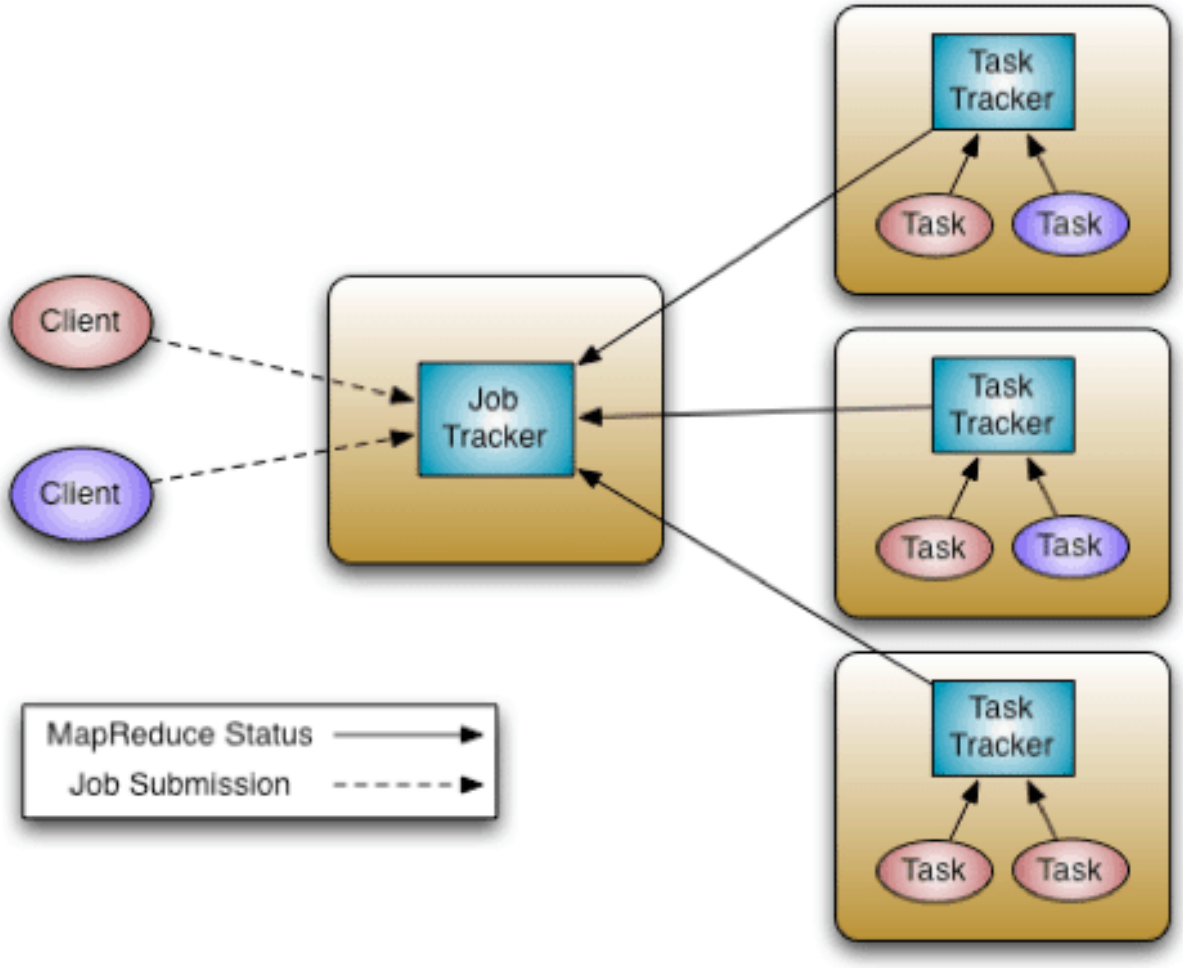
- 스트림 처리 및 실시간 처리 지원 미비

하둡 MapReduce 아키텍처

- Map은 산재된 데이터를 key, value형태로 연관성이 있는 데이터끼리 묶는 작업
- Reduce는 맵 작업 결과에서 중복 데이터를 제거한 후 원하는 데이터를 추출

- 클라이언트가 hadoop에 실행을 요청하는 MR 단위프로그램을 job 이라고 함

하둡 MapReduce 아키텍처



JobTracker

- 클라이언트가 하둡에 제출한 MR job 스케줄링
- 자원 관리 (TaskTrackers), 자원 소비 / 가용성 추적 , 작업 생명주기 관리를 담당
- 병목현상 발생가능성 높음 – 성능저하 발생

TaskTracker

- 클라이언트가 작성한 Map/Reduce 소스를 블록단위로 데이터가 저장된 데이터노드에서 실행하는 역할 담당
- 실행한 Map/Reduce 작업 결과를 JobTracker로 전달

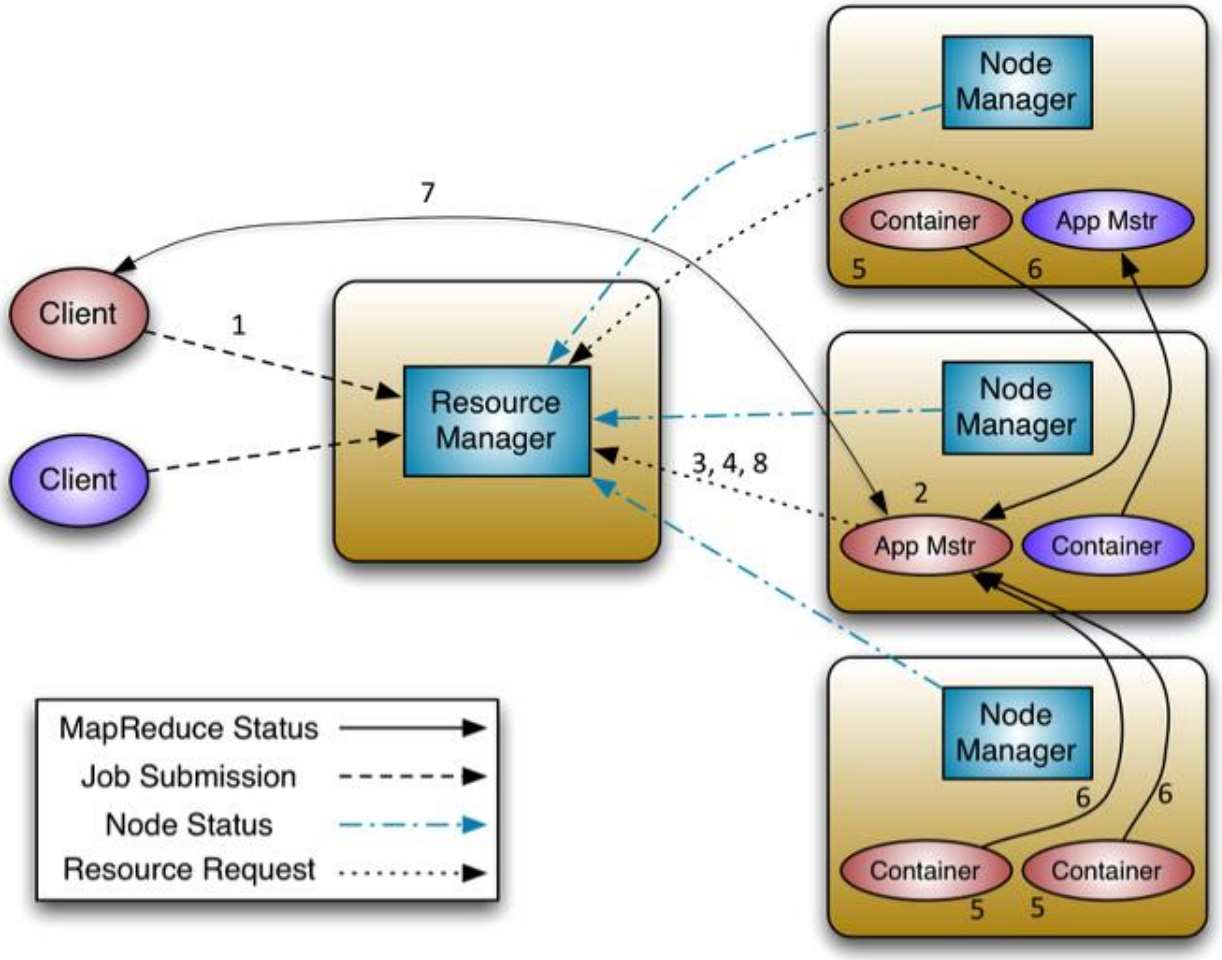
Yarn 아키텍처

- MapReduce는 기본적으로 일괄처리 지향적이지만, 실시간 혹은 준 실시간 처리도 사용자에게 중요한 방식
- 기업/기관의 데이터센터에서는 처리 성능을 향상시키기 위해 더 많은 노드를 추가하길 바라고 있음
- 수많은 노드들 간 자원을 효율적으로 관리해야 할 필요 대두

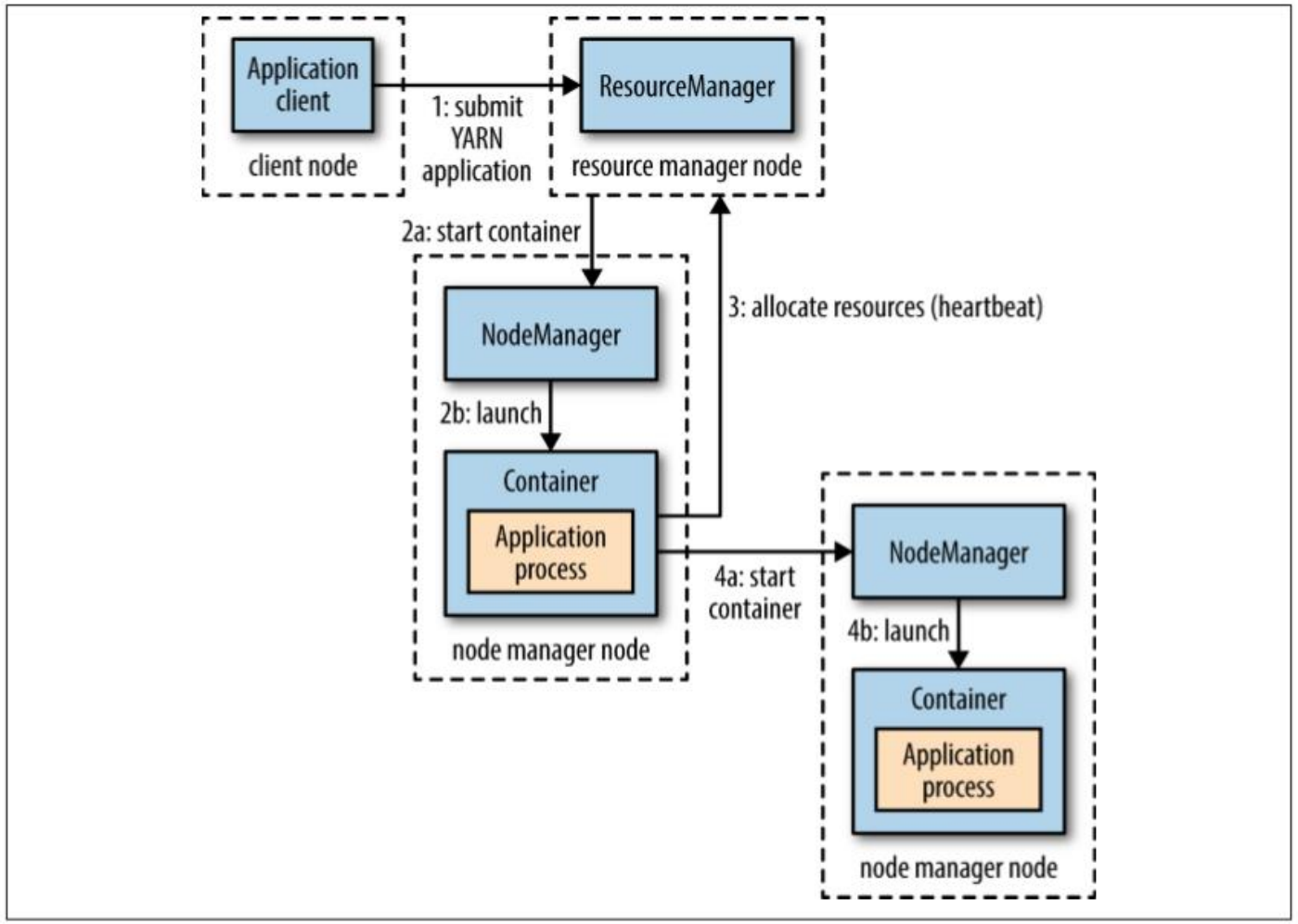
YARN 아키텍처

- 하둡 클러스터의 각 어플리케이션에 필요한 리소스 할당, 모니터링
- 따라서, 다양한 어플리케이션이 하둡 클러스터의 리소스 공유 가능
- MapReduce의 성능을 높이기 위해 도입 - MRv2
- JobTracker = ResourceManagr + ApplicationMaster

YARN 아키텍처


Node Manager
Container
App Mstr
5
6
7
Client
1
Resource Manager
Client
3, 4, 8
Node Manager
2
App Mstr
Container
Node Manager
6
6
Container
Container
5
5
MapReduce Status
Job Submission
Node Status
Resource Request

YARN 아키텍처


Application client
client node
1: submit YARN application
ResourceManager
resource manager node
2a: start container
NodeManager
2b: launch
Container
Application process
node manager node
3: allocate resources (heartbeat)
4a: start container
NodeManager
4b: launch
Container
Application process
node manager node

Resource Manager

- 클러스터마다 존재, 클러스터 전반의 자원 관리와 task들의 스케쥴링 담당
- 클라이언트로부터 애플리케이션 실행 요청을 받으면 그 애플리케이션의 실행을 책임질 Application Master를 실행
- 또한, 클러스터 내에 설치된 모든 Node Manager와 통신을 통해서 각 서버마다 할당된 자원과 사용중인 자원 상황 파악
- Application Master들과의 통신을 통해 필요한 자원 파악 및 관리

ApplicationMaster

- Resource Manager 과 협상하여 하둡 클러스터에서 자기가 담당하는 어플리케이션에 필요한 리소스를 할당받음
- Node Manager 과 협의하여 자기가 담당하는 어플리케이션을 실행하고 그 결과를 주기적으로 모니터링

Node Manager

- 노드 당 한개씩 존재
- 해당 Container의 리소스 사용량을 모니터링하고, 관련 정보를 Resource Manager에게 알리는 역할을 담당

Container

- task를 실행하기 위해 필요한 CPU, Disk, Memory 등 자원묶음을 의미
- 모든 작업은 결국 여러 개의 task로 세분화되며, 각 task는 하나의 Container 안에서 실행
- 필요한 자원의 요청은 Application Master가, 승인 여부는 Resource Manager가 담당

Hadoop 2.0 배포판

Hadoop 2.0 배포판

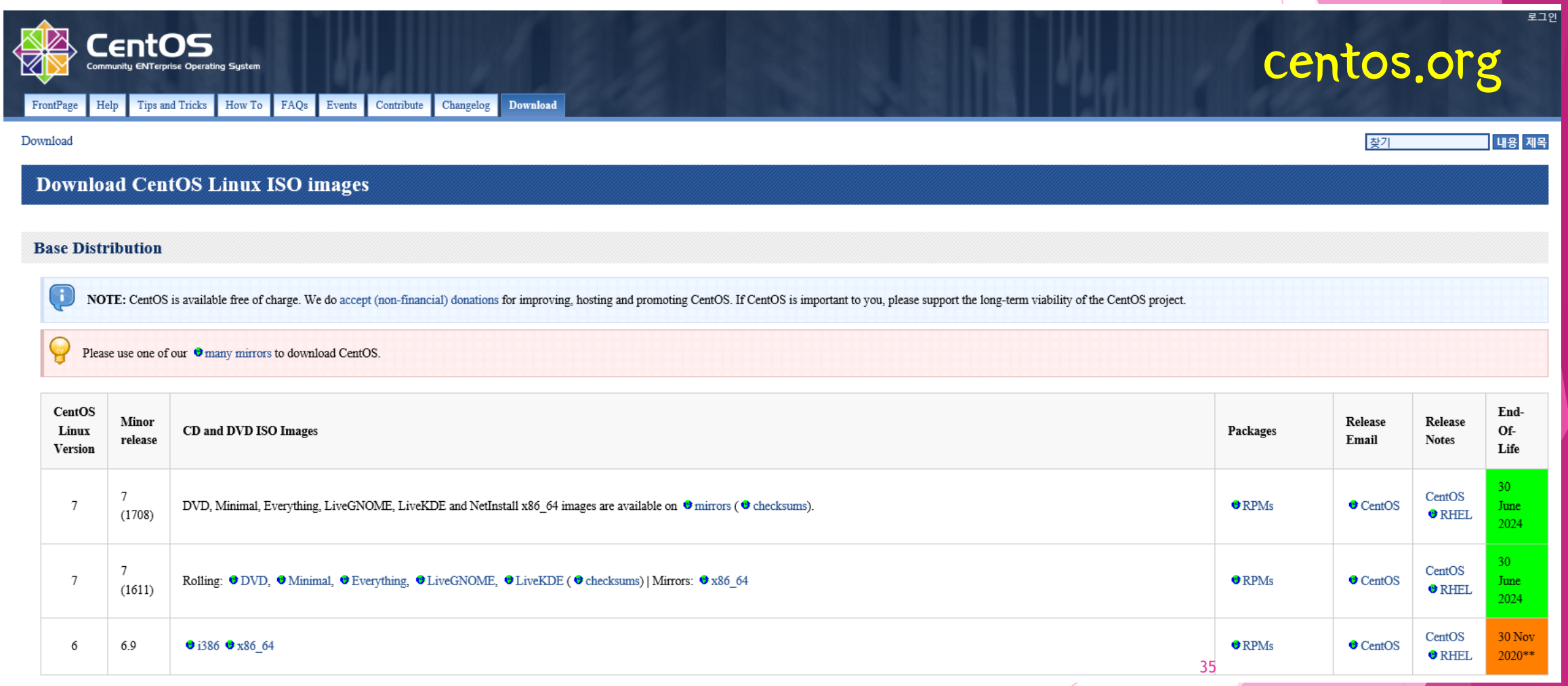
빅데이터 실무 구축환경

- 총 17개의 소프트웨어 설치 예정
- 최소 3대의 가상머신 - VirtualBox 또는 VMware Player
- 리눅스 계열 운영체제 - CentOS 6.x


빅데이터 실무 구축환경

- JDK - 1.7 또는 1.8
- 개발/배포 - Eclispe 또는 IntelliJ
- SSH 클라이언트 - putty

CentOS 6.x



CentOS Linux Version	Minor release	CD and DVD ISO Images	Packages	Release Email	Release Notes	End-Of-Life
7	7 (1708)	DVD, Minimal, Everything, LiveGNOME, LiveKDE and NetInstall x86_64 images are available on mirrors (checksums).	RPMs	CentOS	CentOS RHEL	30 June 2024
7	7 (1611)	Rolling: DVD, Minimal, Everything, LiveGNOME, LiveKDE (checksums) | Mirrors: x86_64	RPMs	CentOS	CentOS RHEL	30 June 2024
6	6.9	i386 x86_64	RPMs	CentOS	CentOS RHEL	30 Nov 2020**

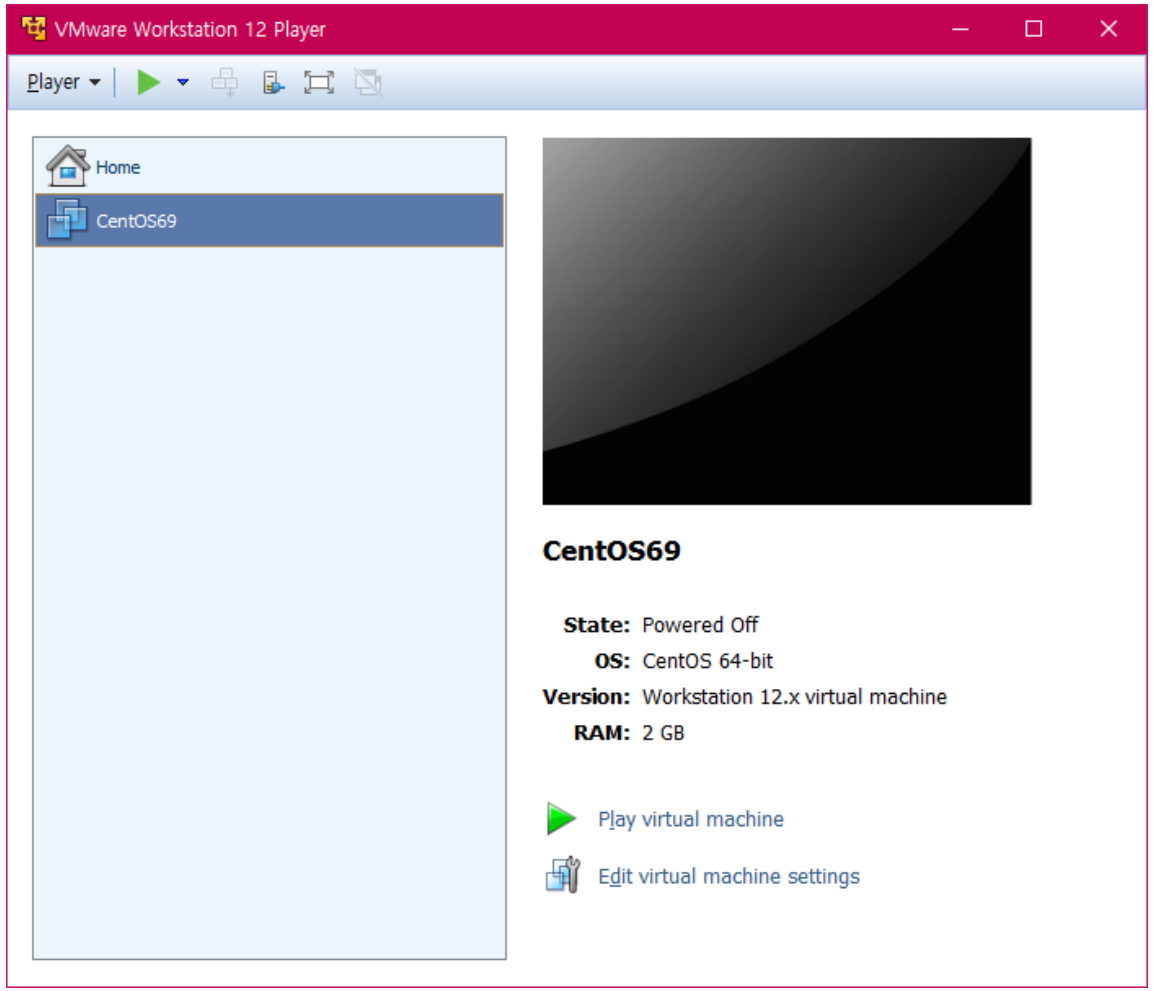
CentOS 6.x

Index of /centos/6.9/isos/x86_64

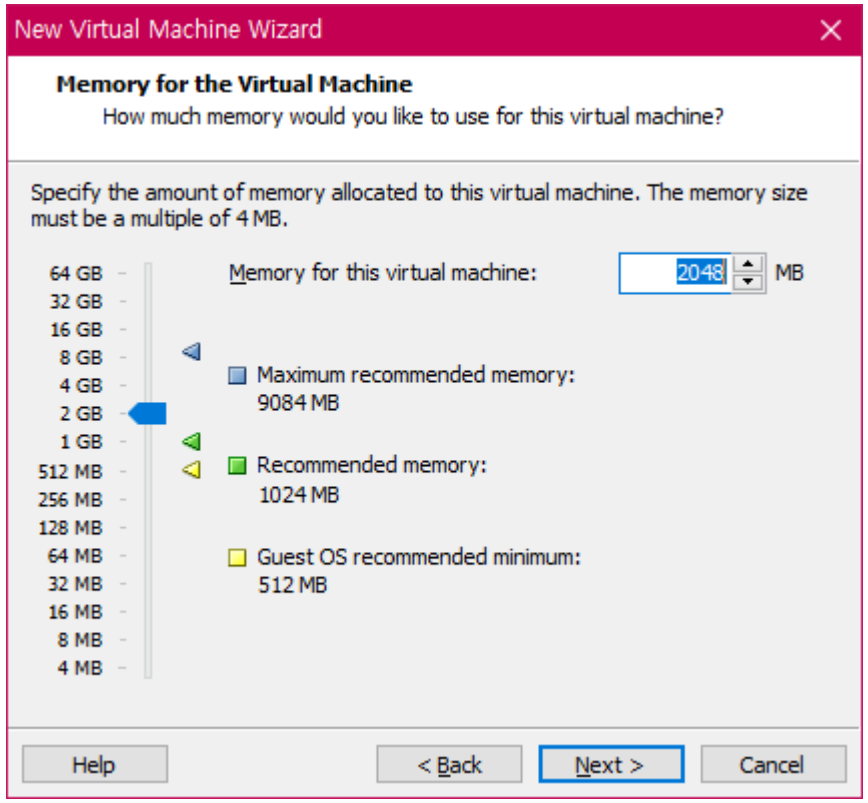
Name	Last modified	Size	Description
Parent Directory		-	
0_README.txt	2017-04-03 19:25	1.7K	
CentOS-6.9-x86_64-LiveDVD.iso	2017-04-01 03:49	1.9G	
CentOS-6.9-x86_64-LiveDVD.torrent	2017-04-05 05:04	76K	
CentOS-6.9-x86_64-bin-DVD1.iso	2017-03-29 03:26	3.7G	
CentOS-6.9-x86_64-bin-DVD1to2.torrent	2017-04-05 05:04	230K	
CentOS-6.9-x86_64-bin-DVD2.iso	2017-03-29 03:26	2.0G	
CentOS-6.9-x86_64-minimal.iso	2017-03-29 03:31	408M	
CentOS-6.9-x86_64-minimal.torrent	2017-04-05 05:04	17K	
CentOS-6.9-x86_64-netinstall.iso	2017-03-29 03:19	230M	
CentOS-6.9-x86_64-netinstall.torrent	2017-04-05 05:04	9.7K	
README.txt	2017-04-03 19:25	1.7K	
md5sum.txt	2017-04-05 04:41	325	
md5sum.txt.asc	2017-04-05 04:56	1.2K	
sha1sum.txt	2017-04-05 04:42	365	
sha1sum.txt.asc	2017-04-05 04:56	1.2K	
sha256sum.txt	2017-03-31 23:46	485	
sha256sum.txt.asc	2017-04-05 04:56	1.3K	

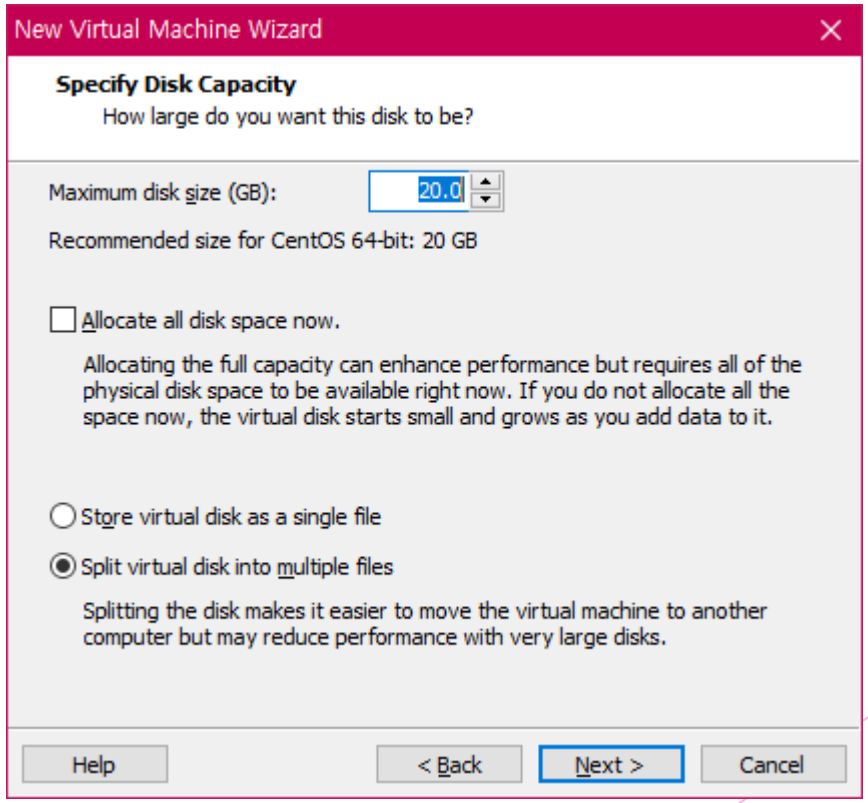
VMware Player 12.x

VMware Player 12.x



VMware Player 12.x





CentOS 6.9 설치 - mini

순서	설정항목	대응처리
1	CentOS 6.9 환영화면	'Install or upgrade an existing system' 선택
2	미디어 테스트	'Skip' 버튼 선택
3	CentOS 6.9 GUI 설치화면	'Next' 버튼 클릭
4	설치 사용 언어 선택	'한국어' 선택 후 '다음' 클릭
5	사용 키보드 선택	'U.S 영어' 선택 후 '다음' 클릭
6	설치 장치 선택 화면	'기본 저장 장치' 선택 후 '다음' 클릭
7	저장 장치 경고	'예, 모든 데이터를 삭제합니다' 선택

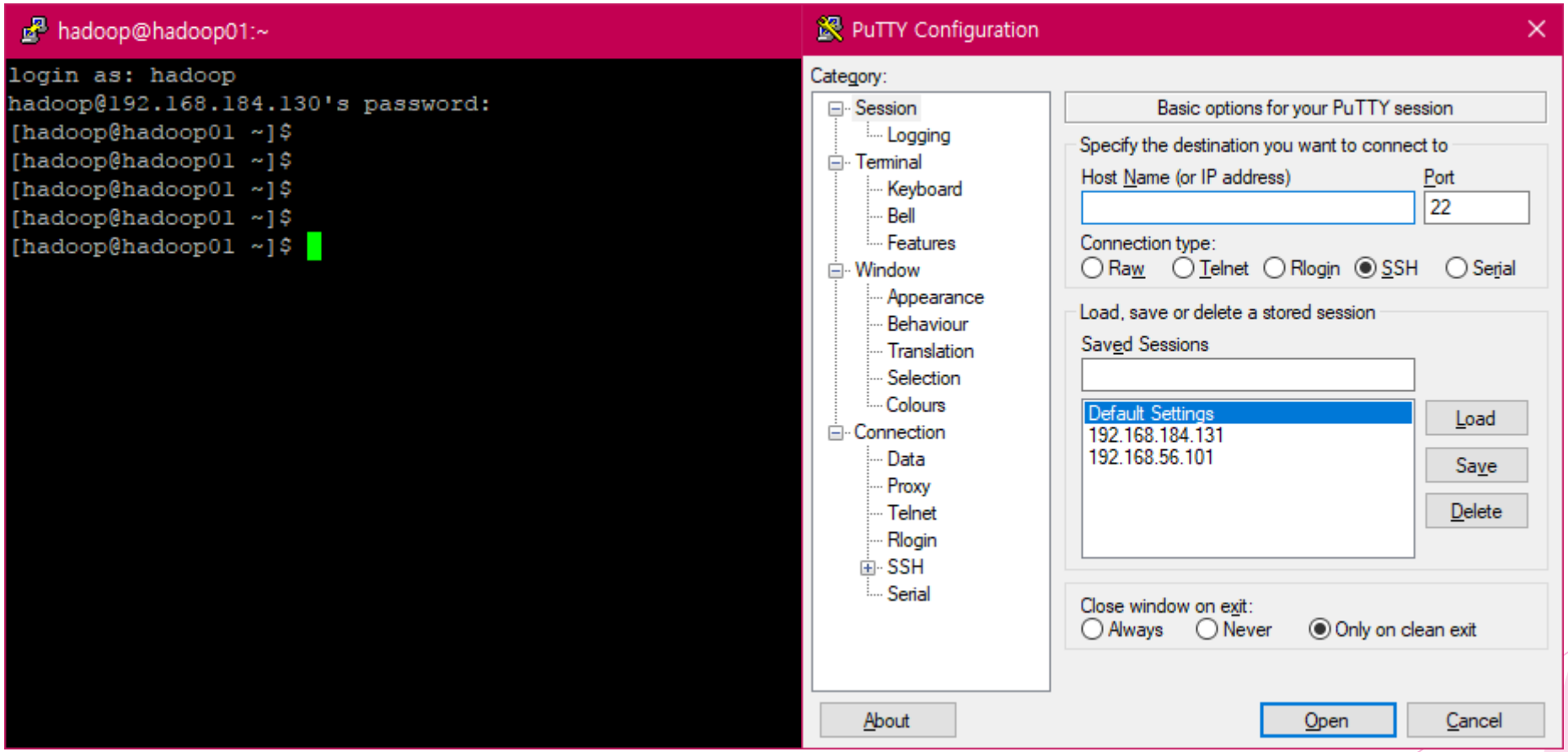
CentOS 6.9 설치 - mini

8	컴퓨터 이름 지정	호스트명에 hadoop01 로 지정
9	네트워크 설정	'System eth0' 클릭 후 '변경하기' 버튼 클릭
10	System eth0 편집	'자동으로 연결' 체크 'IPv4 설정' 탭 클릭 후 '주소만 자동(DHCP)' 선택 DNS 서버 항목에 '8.8.8.8' 입력 '적용' 버튼 클릭 후 '닫기' 버튼 클릭
11	지역 선택	'아시아/서울' 확인 후 '다음' 클릭
12	관리자 암호 설정	'hadoop' 으로 설정하고 '다음' 클릭 '추측하기 쉬운 암호'라는 경고창 뜨면 '어쨌든 사용' 클릭 후 '다음' 클릭
13	파티션 설치방법 선택	'모든 공간 사용' 선택 '파티션 레이아웃 확인 및 변경' 선택 후 '다음'

CentOS 6.9 설치 - mini

14	파티션 나누기	생성된 파티션을 확인하고 '다음' 클릭 '포맷경고' 창 뜨면 '포맷' 버튼 클릭 '디스크 저장장치 설정' 창 뜨면 '디스크에 변경사항 기록' 버튼 클릭
15	부트로더 설치	설정 내용 확인 후 '다음' 버튼 클릭
16	설치 완료	'재부팅' 버튼 클릭
17	로그인 화면	사용자 root 입력, 사용자 root 암호 입력
18	사용자 추가 및 비밀번호 설정	adduser hadoop passwd hadoop
19	hosts 설정	Ifconfig 명령으로 시스템의 IP주소 확인 vi /etc/hosts 127.0.0.1 localhost IP주소 hadoop01
20	시스템 종료	shutdown -h now

Putty 접속


hadoop@hadoop01:~
login as: hadoop
hadoop@192.168.184.130's password:
[hadoop@hadoop01 ~]$
[hadoop@hadoop01 ~]$
[hadoop@hadoop01 ~]$
[hadoop@hadoop01 ~]$
[hadoop@hadoop01 ~]$
PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
192.168.184.131
192.168.56.101
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel

CentOS 6.9 추가 설정

- 추가 패키지 설치 및 시스템 업데이트하기
  - yum groupinstall -y "development tools"
  - yum groupinstall -y "compatibility libraries"
  - yum install -y wget

CentOS 6.9 추가 설정

- ntp 서버를 이용한 시간 동기화
- ntp 패키지 설치 - yum install -y ntp

- ntp 환경설정 - vi /etc/ntp.conf
- server kr.pool.ntp.org
- server time.bora.net
- server time.kornet.net

CentOS 6.9 추가 설정

- ntp 서비스를 시작프로그램에 등록
- **chkconfig ntpd on**

- ntp 서비스 시작
- **service ntpd start**

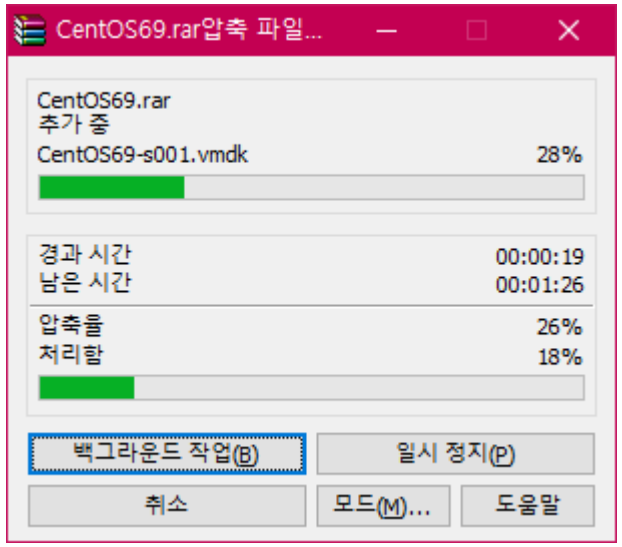
- 동기화 상태 확인 (소요시간 5~10분)
- **ntpq -p**

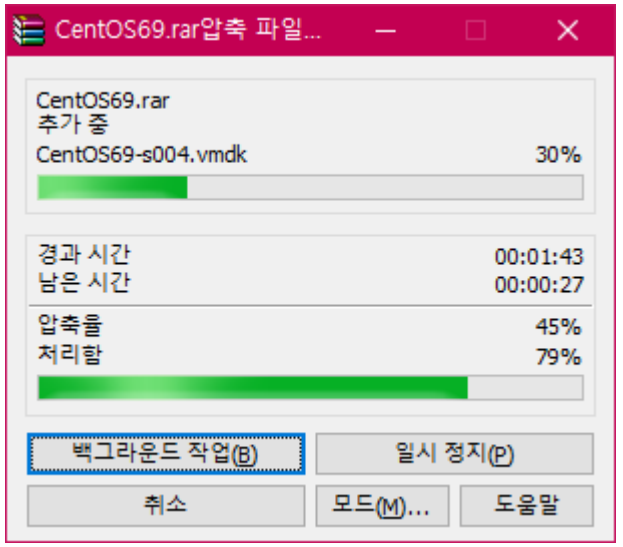
Hadoop 설치

- http://hadoop.apache.org/로 이동해서 2.7.4 버전(2017-09-12)을 다운로드 후 설치
- 2.6.5 버전은 JDK6을 지원하고 2.7.4, 2.8.1은 JDK7을 3.0.0-alpha4는 JDK8을 지원

- 단독모드 : 단일 노드에서 실행, 테스트 및 디버깅 용도로 사용
- 가상분산모드 : 단일 노드에서 클러스터를 구성 (ssh 인증)
- 완전분산모드 : 두 대 이상의 노드를 클러스터로 묶음, 실제 운영시 사용 (ssh 인증)

중간 백업


CentOS69.rar압축 파일...
CentOS69.rar
추가 중
CentOS69-s001.vmdk 28%
경과 시간 00:00:19
남은 시간 00:01:26
압축률 26%
처리함 18%
백그라운드 작업(B)
일시 정지(P)
취소
모드(M)...
도움말


CentOS69.rar압축 파일...
CentOS69.rar
추가 중
CentOS69-s004.vmdk 30%
경과 시간 00:01:43
남은 시간 00:00:27
압축률 45%
처리함 79%
백그라운드 작업(B)
일시 정지(P)
취소
모드(M)...
도움말

Hadoop 단독모드 실행

- !!! Hadoop 계정으로 작업할 것 !!!

- Hadoop 환경 설정
- cd $HADOOP_HOME
- vi etc/hadoop/hadoop-env.sh

- 25번 라인 JAVA_HOME 설정 부분 주석후
- export JAVA_HOME=/usr/local/java 추가

Hadoop 단독모드 실행

- mkdir input
- cp etc/hadoop/* input

- hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar grep input output 'dfs[a-z.]+'
- cat output/*

- hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar wordcount input output2
- cat output2/*

Hadoop 단독모드 실행

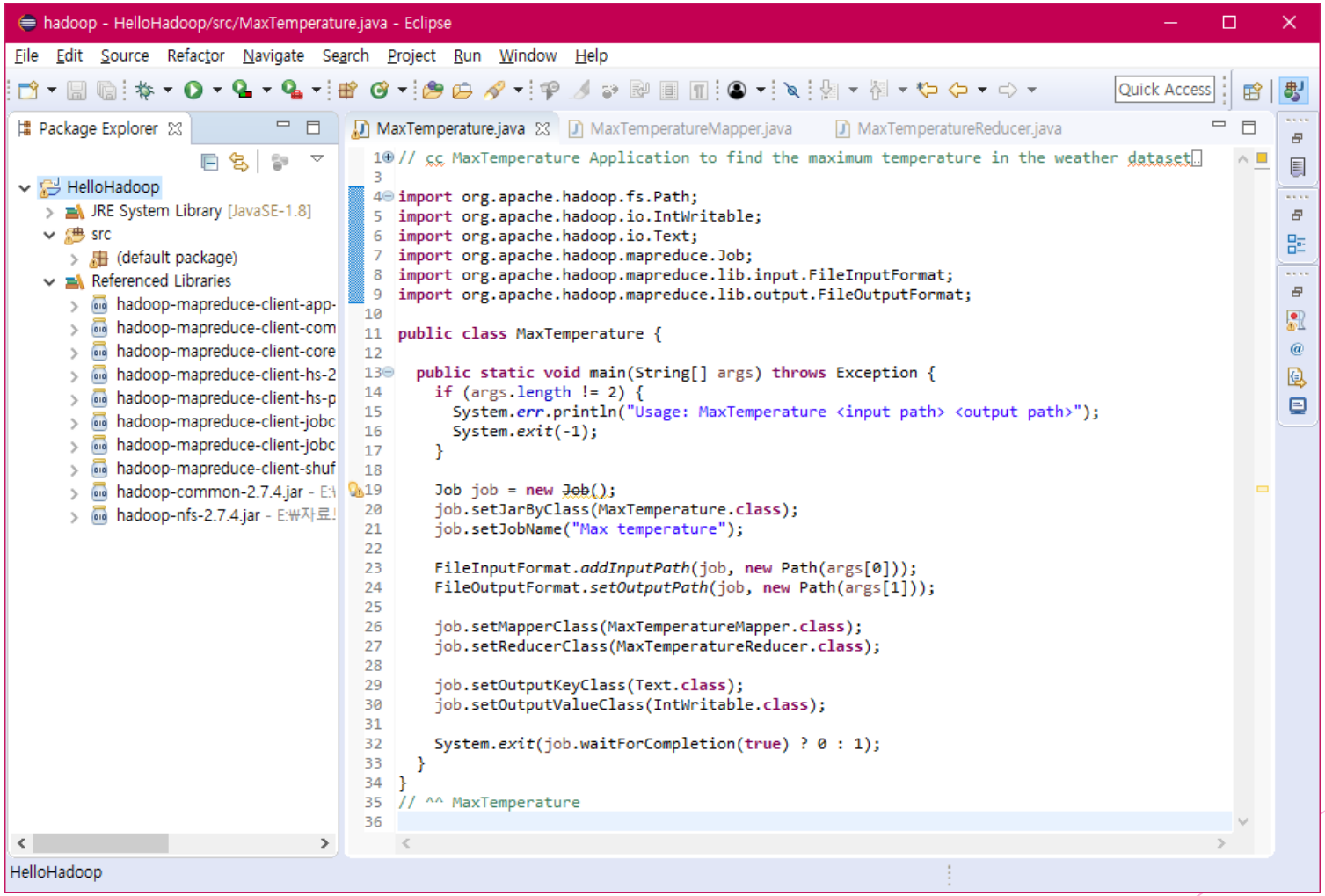
- vi input/sudoku.data
- 8 5 ? 3 9 ? ? ? ?
- ? ? 2 ? ? ? ? ? ?
- ? ? 6 ? 1 ? ? ? 2
- ? ? 4 ? ? 3 ? 5 9
- ? ? 8 9 ? 1 4 ? ?
- 3 2 ? 4 ? ? 8 ? ?
- 9 ? ? ? 8 ? 5 ? ?
- ? ? ? ? ? ? 2 ? ?
- ? ? ? ? 4 5 ? 7 8

```
hadoop jar share/hadoop/mapreduce/hadoop-
mapreduce-examples-2.7.4.jar sudoku input/sudoku.data
```

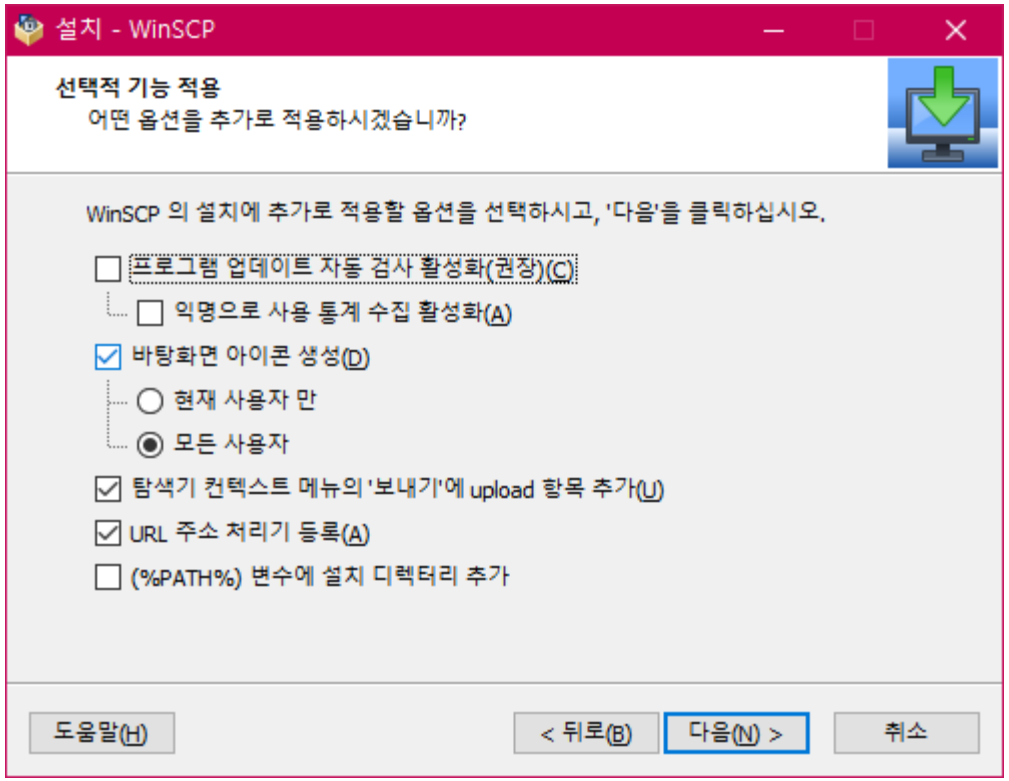
Hadoop 단독모드 실행

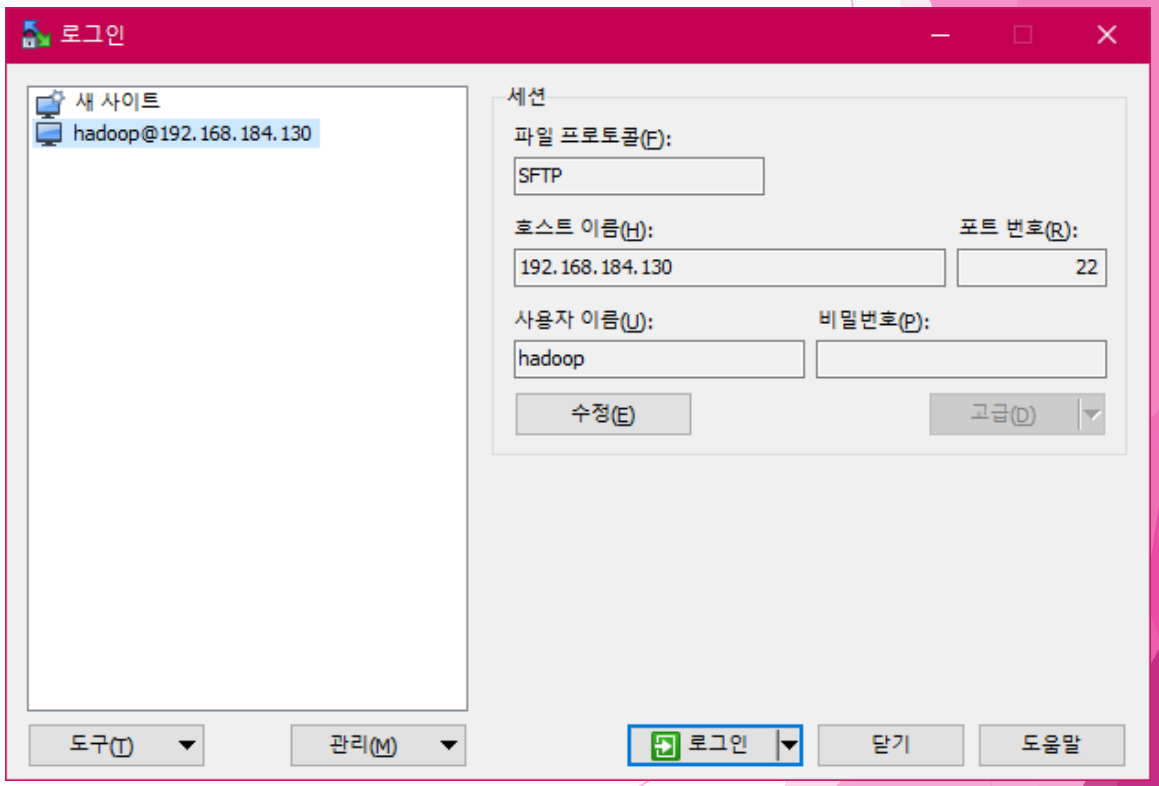
- hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar randomwriter rand
- hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar sort rand rand-sort
- teragen: Generate data for the terasort
- terasort: Run the terasort

MapReduce 예제 실습

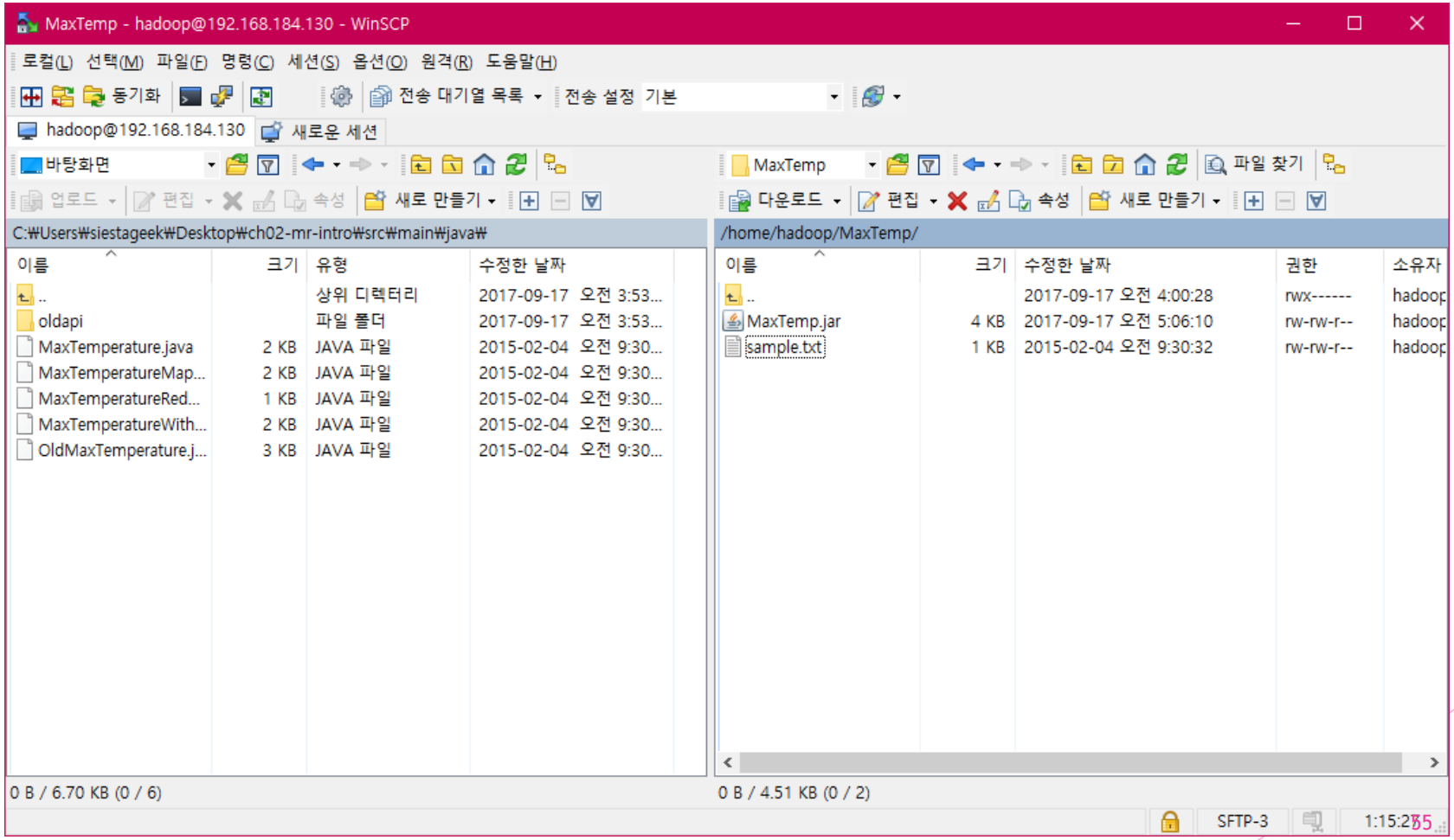


WinSCP


설치 - WinSCP
선택적 기능 적용
어떤 옵션을 추가로 적용하시겠습니까?
WinSCP 의 설치에 추가로 적용할 옵션을 선택하시고, '다음'을 클릭하십시오.
프로그램 업데이트 자동 검사 활성화(권장)(C)
익명으로 사용 통계 수집 활성화(A)
바탕화면 아이콘 생성(D)
현재 사용자 만
모든 사용자
탐색기 컨텍스트 메뉴의 '보내기'에 upload 항목 추가(U)
URL 주소 처리기 등록(A)
(%PATH%) 변수에 설치 디렉터리 추가
도움말(H)
< 뒤로(B)
다음(N) >
취소


로그인
새 사이트
hadoop@192.168.184.130
세션
파일 프로토콜(F):
SFTP
호스트 이름(H):
포트 번호(R):
192.168.184.130
22
사용자 이름(U):
비밀번호(P):
hadoop
수정(E)
고급(D)
도구(T)
관리(M)
로그인
닫기
도움말

WinSCP


MaxTemp - hadoop@192.168.184.130 - WinSCP
로컬(L) 선택(M) 파일(F) 명령(C) 세션(S) 옵션(O) 원격(R) 도움말(H)
동기화
전송 대기열 목록
전송 설정
기본
hadoop@192.168.184.130
새로운 세션
바탕화면
업로드
편집
속성
새로 만들기
C:₩Users₩siestageek₩Desktop₩ch02-mr-intro₩src₩main₩java₩
이름
크기
유형
수정한 날짜
..
상위 디렉터리
2017-09-17 오전 3:53...
oldapi
파일 폴더
2017-09-17 오전 3:53...
MaxTemperature.java
2 KB
JAVA 파일
2015-02-04 오전 9:30...
MaxTemperatureMap...
2 KB
JAVA 파일
2015-02-04 오전 9:30...
MaxTemperatureRed...
1 KB
JAVA 파일
2015-02-04 오전 9:30...
MaxTemperatureWith...
2 KB
JAVA 파일
2015-02-04 오전 9:30...
OldMaxTemperature.j...
3 KB
JAVA 파일
2015-02-04 오전 9:30...
0 B / 6.70 KB (0 / 6)
MaxTemp
파일 찾기
다운로드
편집
속성
새로 만들기
/home/hadoop/MaxTemp/
이름
크기
수정한 날짜
권한
소유자
..
2017-09-17 오전 4:00:28
rwx------
hadoop
MaxTemp.jar
4 KB
2017-09-17 오전 5:06:10
rw-rw-r--
hadoop
sample.txt
1 KB
2015-02-04 오전 9:30:32
rw-rw-r--
hadoop
0 B / 4.51 KB (0 / 2)
SFTP-3

MapReduce 예제 실습

- hadoop jar ~/MaxTemp/MaxTemp.jar MaxTemperature ~/MaxTemp/sample.txt output3

- cat output3/*

ssh 공개키 생성

- !!! 작업시 hadoop 계정을 사용 !!!

- Hadoop은 각각의 서버와 통신할 때 ssh 프로토콜을 사용
- 각 호스트들이 공개키를 가지고 있지 않으면 접속시 마다 로그인 암호를 입력해야 함
- 따라서, 미리 각 호스트의 hadoop 유저 홈에 공개키를 공유해야 함
-
- ssh-keygen -t rsa -P ""
- ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@localhost
- ssh localhost

ssh 공개키 생성

- [hadoop@hadoop01 ~]$ ssh-keygen -t rsa -P ""
- Generating public/private rsa key pair.
- Enter file in which to save the key (/home/hadoop/.ssh/id_rsa): [엔터]
- Created directory '/home/hadoop/.ssh'.
- Your identification has been saved in /home/hadoop/.ssh/id_rsa.
- Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
- The key fingerprint is:
- 04:82:a0:0e:69:5f:ee:4f:9c:f1:9f:cc:d6:05:a4:27 hadoop@hadoop01
- The key's randomart image is:
- +--[RSA 2048]----+
- ... ...
- +-----------------+

ssh 공개키 생성

- [hadoop@hadoop01 ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@localhost
- The authenticity of host 'localhost (127.0.0.1)' can't be established.
- RSA key fingerprint is 47:d4:ef:06:53:1a:6e:df:21:50:d9:56:69:8f:f9:a7.
- Are you sure you want to continue connecting (yes/no)? [yes 입력]
- Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
- hadoop@localhost's password: [hadoop 계정 암호 입력]
- Now try logging into the machine, with "ssh 'hadoop@localhost'", and check in:

- .ssh/authorized_keys

- to make sure we haven't added extra keys that you weren't expecting.

ssh 공개키 생성

- [hadoop@hadoop01 ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@hadoop01
- The authenticity of host 'hadoop01 (192.168.184.130)' can't be established.
- RSA key fingerprint is 47:d4:ef:06:53:1a:6e:df:21:50:d9:56:69:8f:f9:a7.
- Are you sure you want to continue connecting (yes/no)? [yes 입력]
- Warning: Permanently added 'hadoop01,192.168.184.130' (RSA) to the list of known hosts.
- Now try logging into the machine. with "ssh 'hadoop@hadoop01'", and check in:

- .ssh/authorized_keys

- to make sure we haven't added extra keys that you weren't expecting.

Hadoop 의사분산 모드 실행

- 리소스매니저와 노드매니저 데몬 실행 start-yarn.sh
- 노드매니저 관리 상태 확인 yarn node -list
- 리소스매니저 웹 인터페이스 확인 http://localhost:8088/
- 노드매니저 웹 인터페이스 확인 http://localhost:8042/
- 리소스매니저와 노드매니저 데몬 중지 stop-yarn.sh

Hadoop 의사분산 모드 실행

- 분산파일시스템에 입력파일 복사
- cd $HADOOP_HOME/etc/hadoop
- hdfs dfs -put * /user/hadoop
- hdfs dfs -ls /user/hadoop

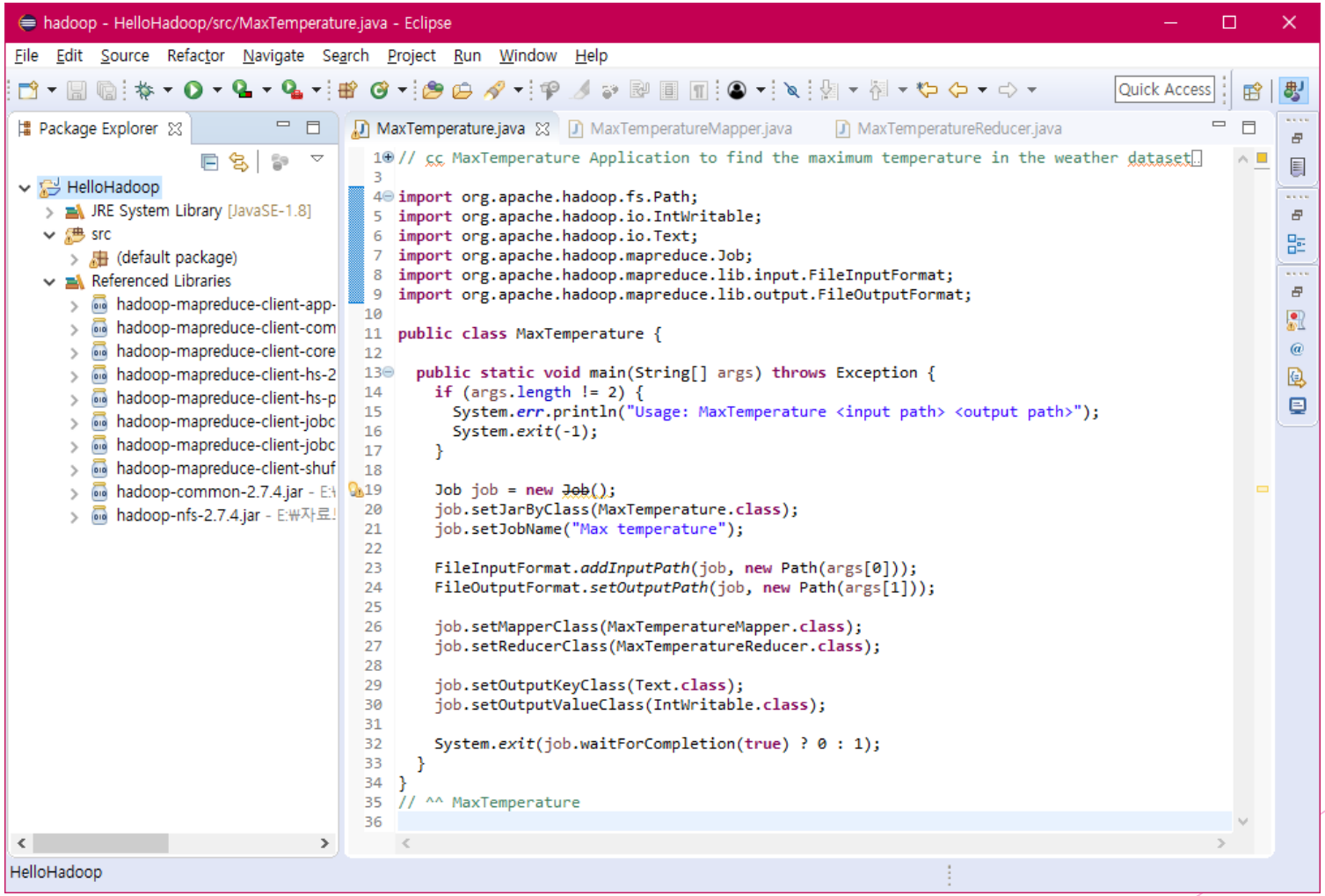
- hadoop 예제 중 grep 샘플 실행
- cd $HADOOP_HOME
- hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar grep /user/hadoop output3 'dfs[a-z.]+'

Hadoop 의사분산 모드 실행

- 분산파일시스템에서 결과파일을 복사해서 로컬파일시스템에 저장한 후 결과출력
- hdfs dfs -get output3 output3
- cat output3/*

- 또는 분산파일시스템에서 바로 결과출력
- hdfs dfs -cat output3/*

MapReduce 예제 실습



MapReduce 예제 실습

- hdfs dfs -put ~/MaxTemp/sample.txt /user/hadoop
- hadoop jar ~/MaxTemp/MaxTemp.jar MaxTemperature /user/hadoop/sample.txt output
- hdfs dfs -cat output/*

감사합니다