



하둡을 이용한 빅데이터 기본과정

(2일차. 오픈소스 하둡 운영 실습)

2015. 02. 24.

CMIKorea

(주)시엠아이코리아

소속 (주)시엠아이코리아

이름 이 상 훈

목 차

1. 하둡 2.0 클러스터 구성
 - 하둡 2.0 설정
 - 하둡 2.0 실행 및 확인
 - HDFS 정보 확인

2. 하둡 2.0 관리
 - HDFS 관리
 - 네임노드 VIEW
 - 네임노드 페더레이션

3. 하둡의 데이터 처리 기술
 - 분산병렬처리
 - 맵리듀스
 - 프로그램 구조

Chapter

01 하둡 2.0 클러스터 구성



하둡 클러스터 설치

- 하둡의 모든 기능이 갖추어진 클러스터
 - Master – Namenode와 JobTracker
 - Backup – SNN (Secondary NameNode)
 - Slave – DataNode와 TaskTracker

- 사용 목적
 - 분산저장
 - 분산연산

hadoop-env.sh 파일설정

JDK경로, 클래스 패스, 데몬 실행 옵션 등 설정

```
$cd $HADOOP_HOME/etc/hadoop/
```

```
$vi hadoop-env.sh
```

➤ Export JAVA_HOME=/usr/local/default

➤자바 설치된 경로 확인 후 설정 하세요.

하둡 실행 시, 아래와 같이 경고메시지가 계속 뜰 수 있습니다.

"Warning: \$HADOOP_HOME is deprecated"

아래 내용을 hadoop-env.sh에 추가 해 주세요.

```
export HADOOP_HOME_WARN_SUPPRESS=1
```

Core-site.xml 파일 수정

```
[root@namenode root]# vi core-site.xml
```

```
<configuration>  
<property>  
<name>fs.default.name</name>  
<value>hdfs://namenode:9000/</value>  
</property>  
<property>  
<name>hadoop.tmp.dir</name>  
<value>/home/root/hadoop-2.2.0/tmp</value>  
</property>  
</configuration>
```

hdfs-site.xml 파일 수정

```
[root@namenode root]# vi hdfs-site.xml
```

```
<configuration>  
<property>  
<name>dfs.replication</name>  
<value>3</value>  
</property>  
</configuration>
```

mapred-site.xml 파일 수정

```
[root@namenode root]# vi mapred-site.xml
```

```
<configuration>  
<property>  
<name>mapreduce.framework.name</name>  
<value>yarn</value>  
</property>  
</configuration>
```


yarn-site.xml 파일 수정

```
[root@namenode root]# vi yarn-site.xml
```

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-
services.mapreduce_shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
<property>
<name>yarn.resourcemanager.hostname</name>
<value>namenode</value>
```

설정파일 배포

네임노드에서

```
$cd $HADOOP_HOME/etc/hadoop
```

```
$scp ./* root@datanode1:/home/root/hadoop-2.2.0/etc/hadoop/
```

```
$scp ./* root@datanode2:/home/root/hadoop-2.2.0/etc/hadoop/
```

네임노드 초기화

◆ 네임노드에서 한번만 실행

```
$cd $HADOOP_HOME/bin
```

```
$hdfs namenode -format
```

에러메시지가 있다면 환경설정 파일이 잘못된 것입니다. 확인하고 다시 실행시킵니다.

- 주의 : 데이터 노드에 이미 데이터 파일이 생성된 상황에서 초기화 하면 데이터 노드와 동기화가 되지 않음.
- `Java.io.IOException: Incompatilble namespaceIDs`
- 증상으로는 하둡을 설치 후 다시 포맷하는 경우 데이터 노드들이 안 붙는 에러
- 데이터 노드에서 `rm -rf $HADOOP_HOME/tmp/dfs/data/*`

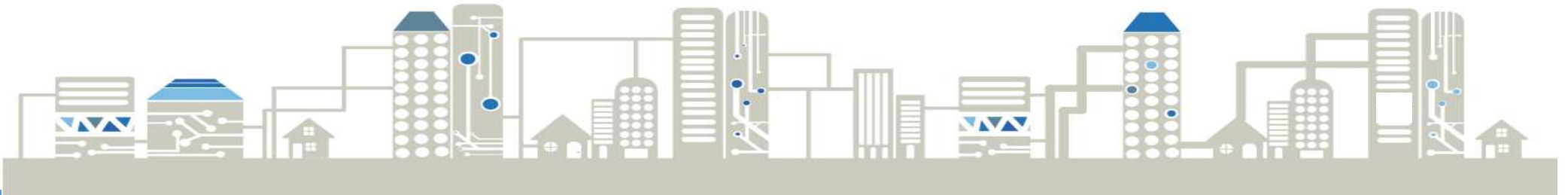
실행방법

```
$sbin/hadoop-daemon.sh start namenode  
$sbin/hadoop-daemon.sh start datanode  
$sbin/yarn-daemon.sh start resourcemanager  
$sbin/yarn-daemon.sh start nodemanager  
$sbin/mr-jobhistory-daemon.sh start historyserver
```

◆ 브라우저에서 <http://namenode:50070/dfshealth.jsp> 실행 후 파일
시스템 상태를 확인 합니다.

Chapter

02 하둡 2.0 관리



하둡 HDFS 정보

NameNode 'master:9000'

Started: Wed Mar 27 15:52:42 KST 2013
Version: 1.1.2, r1440782
Compiled: Thu Jan 31 02:03:24 UTC 2013 by hortonfo
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)

Cluster Summary

10 files and directories, 3 blocks = 13 total. Heap Size is 48.12 MB / 966.69 MB (4%)

Configured Capacity	:	151.04 GB
DFS Used	:	3.07 MB
Non DFS Used	:	54.08 GB
DFS Remaining	:	96.95 GB
DFS Used%	:	0 %
DFS Remaining%	:	64.19 %
Live Nodes	:	3
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

NameNode Storage:

Storage Directory	Type	State
/data/name	IMAGE_AND_EDITS	Active
/data/backup	IMAGE_AND_EDITS	Active

This is [Apache Hadoop](#) release 1.1.2

➤ <http://master.namenode:50070>

➤ Live Nodes 가 연결된 DataNode의 수

하둡 HDFS 데이터노드 정보

➤ # DataNode 노드 리스트 확인

➤ <http://master.namenode:50070/dfsnodelist.jsp?whatNodes=LIVE>

NameNode 'master:9000'

Started: Wed Mar 27 15:52:42 KST 2013
Version: 1.1.2, r1440782
Compiled: Thu Jan 31 02:03:24 UTC 2013 by hortonfo
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)
[Go back to DFS home](#)

Live Datanodes : 3

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
localhost	2	In Service	41.7	0	36.74	4.96	0	<input type="text"/>	11.89	3
localhost	0	In Service	98.43	0	12.64	85.79	0	<input type="text"/>	87.16	1
slave	2	In Service	10.9	0	4.7	6.2	0.01	<input type="text"/>	56.88	2

This is [Apache Hadoop](#) release 1.1.2

하둡 HDFS 디렉토리 정보

➤ # DataNode 노드 디렉토리 정보 확인

➤ <http://slave.datanode01:50075/browseDirectory.jsp?namenodeInfoPort=50070&dir=%2F>

Contents of directory /

Goto : × go

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
tmp	dir				2013-03-27 15:53	rwxr-xr-x	hadoop	supergroup
user	dir				2013-03-27 15:54	rwxr-xr-x	hadoop	supergroup

[Go back to DFS home](#)

Local logs

[Log](#) directory

This is [Apache Hadoop](#) release 1.1.2

하둡 HDFS 노드의 파일 정보 확인

File: [/user/hadoop/oracle_plsql_language_pocket_reference_fourth_edition.pdf](#)

Goto :

[Go back to dir listing](#)
[Advanced view/download options](#)

[View Next chunk](#)

```
%PDF-1.6
%
2996 0 obj <</Linearized 1/L 869419/O 3001/E 60625/N 180/T 809450/H [ 496 2245]>>
endobj

xref
2996 10
0000000016 00000 n
0000002741 00000 n
0000002963 00000 n
0000003008 00000 n
0000003140 00000 n
0000003177 00000 n
0000003374 00000 n
0000003479 00000 n
0000060360 00000 n
0000000496 00000 n
trailer
<</Size 3006/Prev 809437/Root 2997 0 R/Info 2995 0 R/ID[<2970B584BCD2F62514136FC0C37AF9CE>
startxref
0
%%EOF

3005 0 obj<</Length 2138/E 5164/Filter/FlateDecode/I 5180/O 5126/S 4991/V 5142>>stream
x
(w
+4
:G
;
z
d
r
$
j
i
}
U
Q
i
:
5
8
!
T
3
R
 &
F
N
P
o
t
:
```

[Download this file](#)
[Tail this file](#)

Chunk size to view (in bytes, up to file's DFS block size):

Total number of blocks: 1
-4234428498334613222: 192.168.1.111:50010 192.168.1.113:50010

[Go back to DFS home](#)

Local logs

[Log directory](#)

This is [Apache Hadoop](#) release 1.1.2

하둡 HDFS 노드의 파일 정보 확인

- ❖ 이와 같이 지정된 데이터노드에 복제가 된 것을 확인.
- ❖ 복제가 되는 기준은 hdfs-site.xml 설정에서 **dfs.replication** 값을 반영.

Total number of blocks: 1
-4234428498334613222: 192.168.1.111:50010 192.168.1.113:50010

Total number of blocks: 1
-1479047574059270437: 192.168.1.113:50010 192.168.1.112:50010

Total number of blocks: 1
-8675812117651072288: 192.168.1.112:50010 192.168.1.111:50010

- ❖ 파일의 사이즈가 64MB를 초과할 경우에는 64MB 단위로 나누어서 저장.
- ❖ 블록사이즈 기본 값은 64MB이고, hdfs-site.xml 설정에서 **dfs.block.size** 값을 반영.

// 3개의 블록에 2개씩의 복제
Total number of blocks: 3

-527122170720419961: 192.168.1.111:50010 192.168.1.113:50010
8606886155927734038: 192.168.1.112:50010 192.168.1.113:50010
3788523866326037174: 192.168.1.111:50010 192.168.1.113:50010

파일시스템 리스트 조회

➤ 명령어 : `dfs -ls /;`

```
hive> dfs -ls /;  
Found 3 items  
drwxr-xr-x  - root supergroup      0 2014-11-13 11:29 /readme3  
drwxr-xr-x  - root supergroup      0 2014-11-13 20:10 /tmp  
drwxr-xr-x  - root supergroup      0 2014-11-13 20:06 /user  
hive> █
```

파일시스템에 파일 올리기

➤명령어 : `dfs -put /home/root/hadoop-1.0.4/README.txt /;`

```
hive> dfs -put /home/root/hadoop-1.0.4/README.txt /;
hive> dfs -ls /;
Found 4 items
-rw-r--r--    1 root supergroup      1366 2014-11-19 12:56 /README.txt
drwxr-xr-x    - root supergroup         0 2014-11-13 11:29 /readme3
drwxr-xr-x    - root supergroup         0 2014-11-13 20:10 /tmp
drwxr-xr-x    - root supergroup         0 2014-11-13 20:06 /user
hive> █
```

로컬시스템에 있는 README.txt 파일을 분산 파일 시스템으로 옮긴다.
dfs -ls 명령어를 이용하여 분산 파일 시스템에 README.txt 파일이 생성된 것을 확인할 수 있다.

파일시스템 디렉토리 만들기

➤명령어 : `dfs -mkdir /data_dir;`

```
hive> dfs -mkdir /data_dir;
hive> dfs -ls /;
Found 5 items
-rw-r--r--    1 root supergroup      1366 2014-11-19 12:56 /README.txt
drwxr-xr-x    - root supergroup         0 2014-11-19 23:25 /data_dir
drwxr-xr-x    - root supergroup         0 2014-11-13 11:29 /readme3
drwxr-xr-x    - root supergroup         0 2014-11-19 09:40 /tmp
drwxr-xr-x    - root supergroup         0 2014-11-13 20:06 /user
```

dfs -mkdir 명령어로 /data_dir 디렉토리를 생성한 다음 dfs -ls 명령어를 이용하여 분산 파일 시스템을 조회하면 /data_dir 디렉토리가 생성된 것을 확인할 수 있다.

파일시스템 파일 복사

➤명령어 : `dfs -cp /README.txt /data_dir/;`

```
hive> dfs -cp /README.txt /data_dir/
> ;
hive> dfs -lsr /
> ;
-rw-r--r--    1 root supergroup      1366 2014-11-19 12:56 /README.txt
drwxr-xr-x    - root supergroup         0 2014-11-19 23:44 /data_dir
-rw-r--r--    1 root supergroup      1366 2014-11-19 23:44 /data_dir/README.txt
drwxr-xr-x    - root supergroup         0 2014-11-13 11:29 /readme3
drwxr-xr-x    - root supergroup         0 2014-11-19 09:40 /tmp
```

`dfs -cp` 명령어로 `/README.txt` 파일을 복사해서 `/data_dir` 디렉토리 안에 복사본을 추가 시킨다.

`dfs -lsr` 명령어를 이용하여 `/data_dir` 디렉토리 안에 `README.txt` 파일을 조회할 수 있다.

파일시스템 파일 이동

➤명령어 : `dfs -mv /README.txt /data_dir/READMEDATA.txt;`

```
hive> dfs -mv /README.txt /data_dir/READMEDATA.txt;
hive> dfs -lsr /;
drwxr-xr-x  - root supergroup          0 2014-11-19 23:54 /data_dir
-rw-r--r--  1 root supergroup      1366 2014-11-19 23:44 /data_dir/README.txt
-rw-r--r--  1 root supergroup      1366 2014-11-19 12:56 /data_dir/READMEDATA.
txt
```

dfs -mv 명령어로 /README.txt 파일을 /data_dir/로 이동 시킨다.
이때, 이동된 파일은 READMEDATA.txt로 파일명이 변경된다.
dfs -lsr 명령어를 이용하여 /data_dir 디렉토리 안에 README.txt 와 READMEDATA.txt 파일을 조회할 수 있다.

파일시스템 파일 수 확인

➤명령어 : dfs -count /;

```
hive> dfs -count /;
      17          4          4102 hdfs://localhost:9000/
```

dfs -count 명령어를 이용하여 분산 파일 시스템의 디렉토리와 파일 수를 확인할 수 있다..

내용은 / 아래의 17개의 디렉토리를 조사하였으며, 총 파일은 4개가 존재한다.

파일시스템 파일 삭제

➤명령어 : `dfs -rmr /README.txt;`

```
hive> dfs -rmr /README.txt;
Deleted hdfs://localhost:9000/README.txt
hive> dfs -ls /;
Found 5 items
drwxr-xr-x  - root supergroup      0 2014-11-19 23:54 /data_dir
drwxr-xr-x  - root supergroup      0 2014-11-13 11:29 /readme3
drwxr-xr-x  - root supergroup      0 2014-11-19 23:58 /skt
drwxr-xr-x  - root supergroup      0 2014-11-19 09:40 /tmp
drwxr-xr-x  - root supergroup      0 2014-11-13 20:06 /user
```

`dfs -rmr` 명령어로 `/README.txt` 파일을 삭제 시킨다.

`dfs -ls` 명령어를 이용하여 조회할 수 있다.

하둡 HDFS 관리

➤HDFS 관리 작업

- 데이터노드 추가
- 데이터노드 삭제
- fsck로 파일시스템 무결성 점검하기
- HDFS 블록 데이터 밸런싱
- 실패한 디스크 처리하기

데이터노드 추가

➤ 데이터노드 추가하기

- `hdfs-site.xml`에서 `dfs.hosts` 위치 파일에 데이터노드 IP 주소를 추가
- `hadoop dfsadmin -refreshNodes` 실행
- 만약 랙-인지 기능을 사용하고 있으면, 랙 정보에도 새로운 호스트를 추가
- 데이터노드 프로세스를 시작
- 네임노드 웹 UI나 `Hadoop dfsadmin -report` 명령을 실행하여 확인

데이터노드 삭제

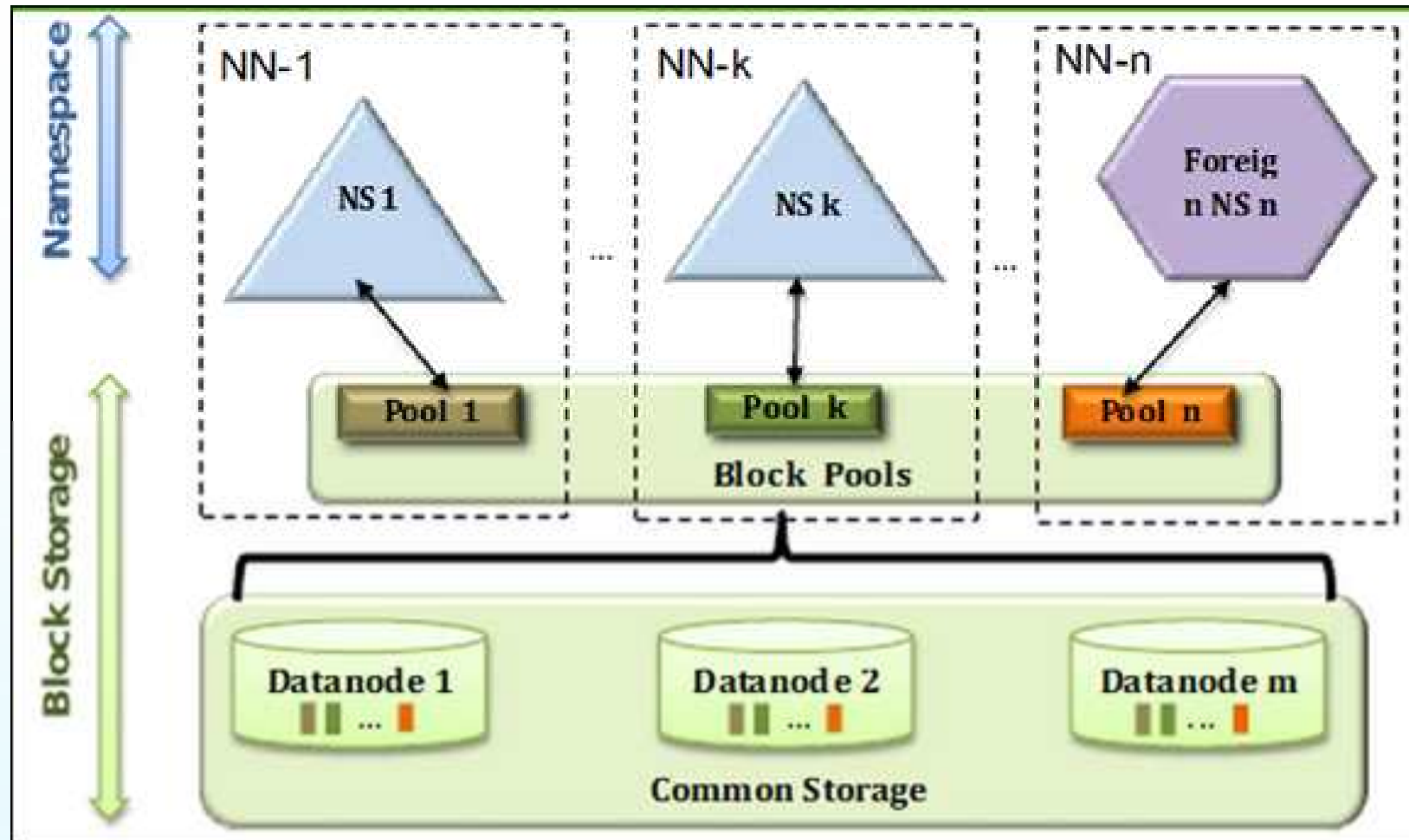
➤ 데이터노드 삭제

- `hdfs-site.xml`에서 `dfs.hosts.exclude` 위치 파일에 IP 주소를 추가
- `hadoop dfsadmin -refreshNodes` 실행
- 만약 랙-인지 기능을 사용하고 있으면, 랙 정보에 새로운 호스트를 추가
- 데이터노드 프로세스를 중단
- 네임노드 웹 UI나 `Hadoop dfsadmin -report` 명령을 실행하여 확인

네임노드 고가용성(1)

- 네임노드 페더레이션
- 하나의 cluster에 여러 개의 독립된 namenode들과 namespace volume이 존재
 - 아파치 하둡 2.0에서 처음 소개
 - 파일시스템의 전체 메타데이터 저장하는데 발생하는 한계를 극복
 - 네임노드 고가용성 : 두 네임노드로 단일 네임스페이스를 관리
 - 모든 네임노드를 포맷하면 네임노드에 개별적으로 저장된 데이터들의 블록 풀이 생성
 - 데이터노드는 다수의 블록 풀에 데이터를 저장하고 네임노드들과 각각 통신

네임노드 고가용성(2)



네임노드 뷰

- 네임노드 페더레이션-단일 네임노드 뷰
- 특정 네임노드의 네임스페이스 정보를 표시
- 네임노드 페더레이션-클러스터 뷰
- 연결된 데이터노드의 수와 총 용량을 포함한 네임노드들의 요약 뷰, 클러스터 수준의 뷰에서 페더레이션 클러스터의 개요를 볼 수 있으며, 모든 네임노드의 뷰는 동일

페더레이션

- nameservice-id 설정
- hdfs-site.xml 파일의 dfs.nameservices 속성에 정의
- 데이터노드들은 dfs.nameservices의 목록에 있는 모든 nameservice-id에 하트비트를 전송
- dfs.namenode.rpc-address.nameservice-id에 nameservice-id와 호스트명과 포트를 각각 지정
- 설정 파일을 수정하고 모든 호스트에 반영한 다음, 모든 네임노드를 다시 포맷

hdfs-site.xml 파일 설정(1)

```
<property>
<name>dfs.nameservices</name>
<value>nid1,nid2</value>
</property>
<!-- nameservices에 선언된 nid1을 datanode1에 연결한다. -->
<property>
<name>dfs.namenode.rpc-address.nid1</name>
<value>datanode1:8020</value>
</property>
```

hdfs-site.xml 파일 설정(2)

```
<!-- nameservices에 선언된 nid2을 datanode2에 연결한다. -->  
<property>  
  <name>dfs.namenode.rpc-address.nid2</name>  
  <value>datanode2:8020</value>  
</property>  
<property>  
  <name>dfs.name.dir</name>  
  <value>{fs.tmp.dir}/data/nid1,{fs.tmp.dir}/data/nid2</value>  
</property>
```

네임노드 포맷 방법

```
hdfs namenode -format -clusterId nid-group
```

<http://datanode1:50070/dfshealth.jsp>에서 네임노드의
네임스페이스 정보를 확인한다.

<http://namenode:50070/dfsclusterhealth.jsp> 네임노드의
클러스터 정보를 확인한다.

core-site.xml VIEW설정

```
<proper ty>
<name>fs.defaultFS</name>
<value>viewfs:///</value>
</proper ty>
<proper ty>
<name>fs.viewfs.mounttable.default.link./nid1</name>
<value>hdfs://namenode:8020/</value>
</proper ty>
<proper ty>
<name>fs.viewfs.mounttable.default.link./nid2</name>
<value>hdfs://datanode1:8020/</value>
</proper ty>
```

Chapter

03 하둡의 데이터 처리 기술



MapReduce

- 대용량의 데이터를 병렬로 처리하기 위한 소프트웨어 프레임워크
- 배치형 데이터 처리 시스템
- 대용량 데이터로부터 Key-Value mapping을 통해 reduction을 하는 프로그래밍 모델
- Google에서 발표한 병렬처리 모델

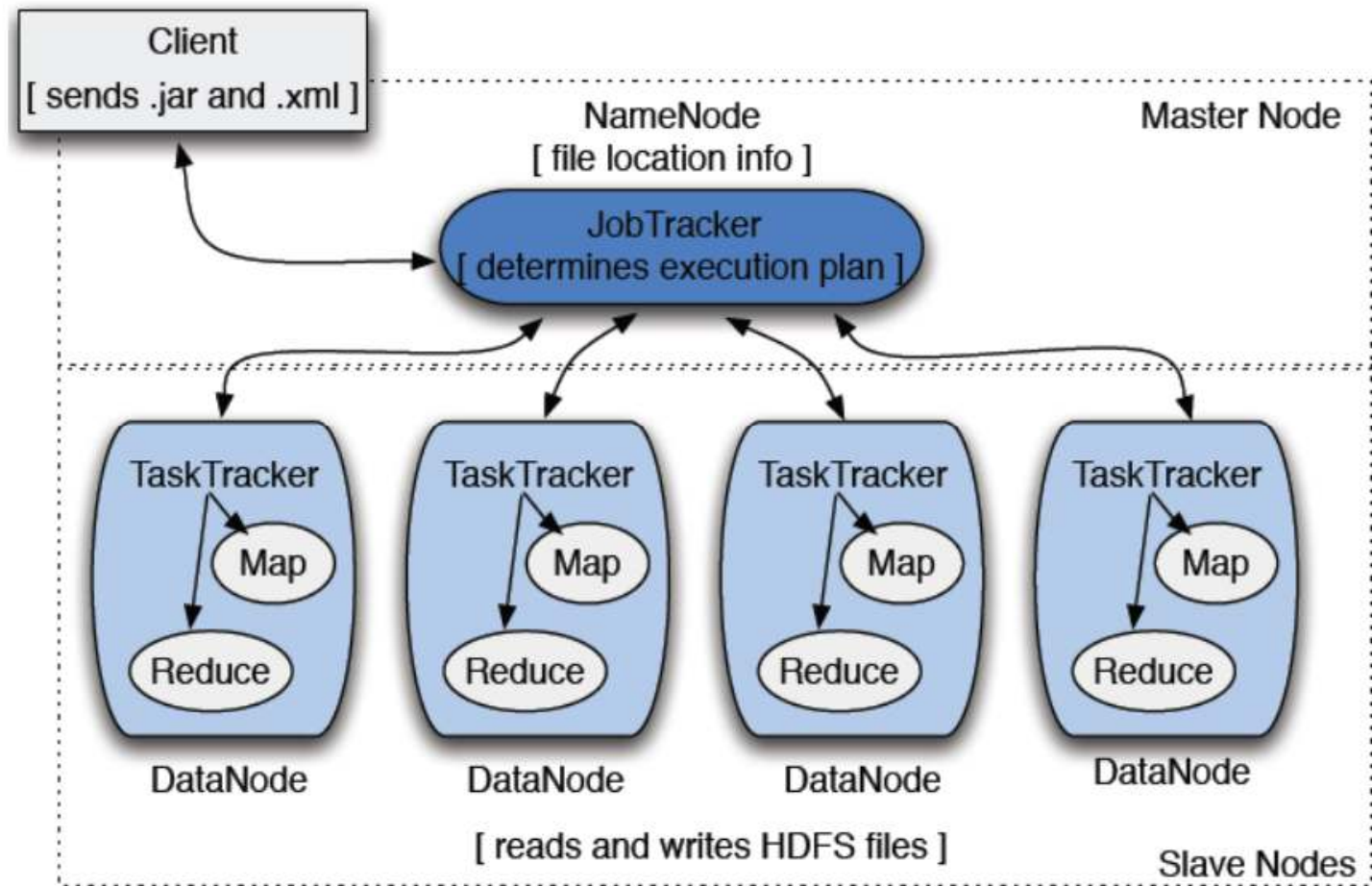
프로그램 차이점

- 종래의 데이터 처리 모델은 데이터를 어플리케이션으로 모아서 처리
- 데이터 사이즈가 증가함에 따라 프로그램 처리는 점점 어려워짐.
- MapReduce는 데이터가 있는 곳으로 프로그램을 배포하여 처리
- 분산된 물리적인 환경에서의 데이터를 나눠서 처리함.

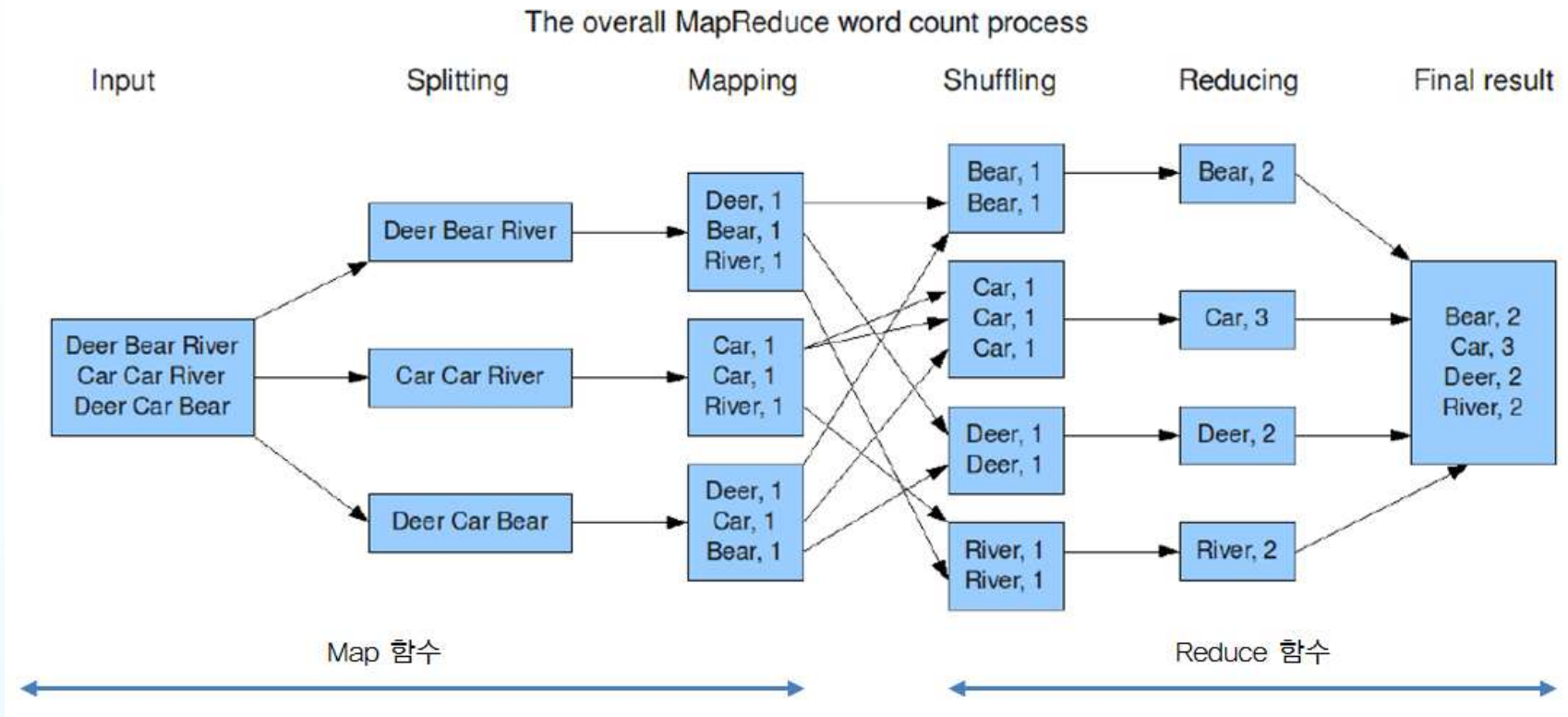
MapReduce 기본구성(1)

- Job, JobTracker
 - : 전체 프로그램 범위
 - : Job을 관리하는 프로세스(데몬)
- Task, TaskTracker
 - : 분산처리하는 작업 단위
 - : Task를 관리하는 프로세스(데몬)
- Master node / Slave node

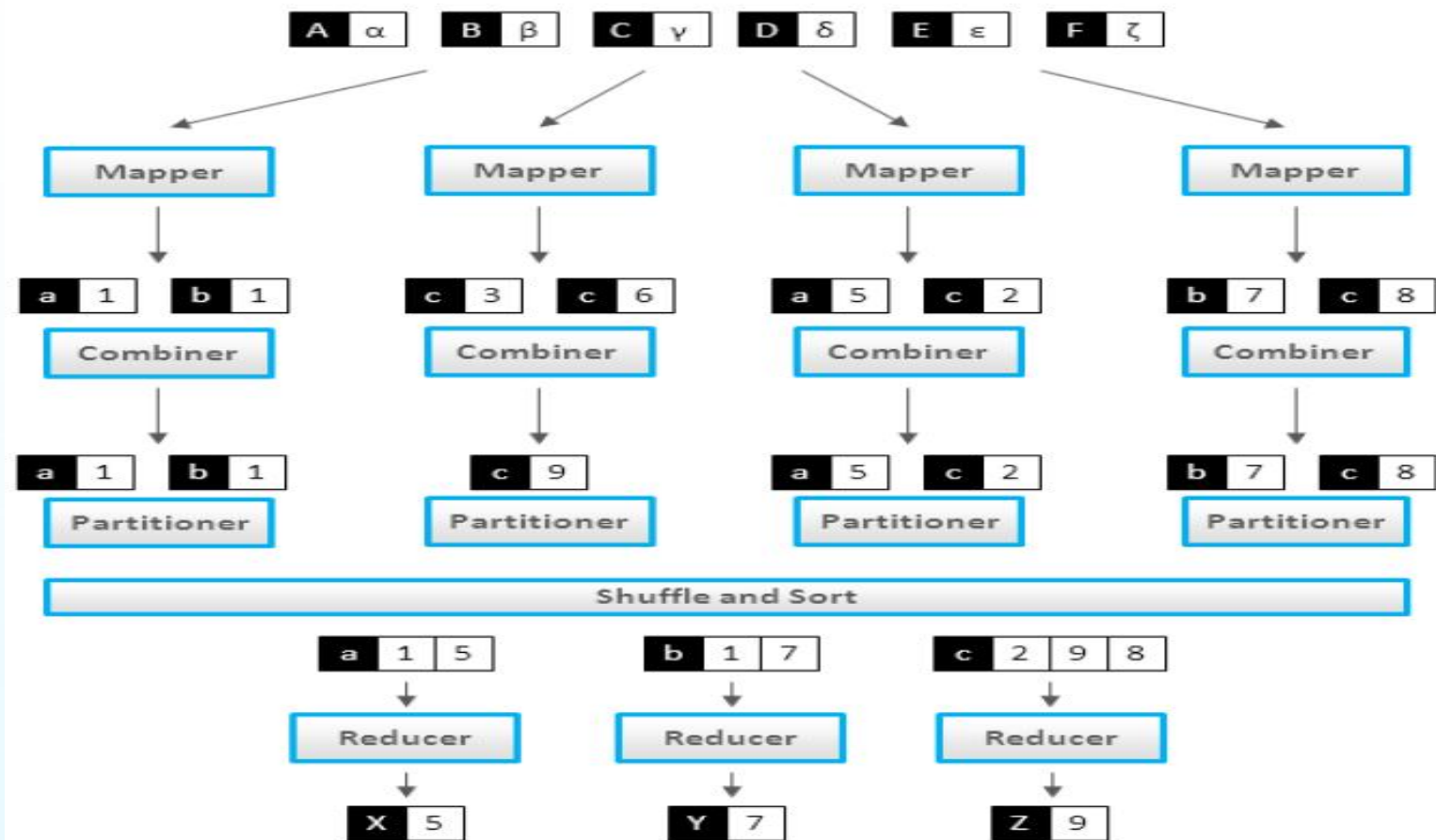
MapReduce 기본구성(2)



데이터처리(1)



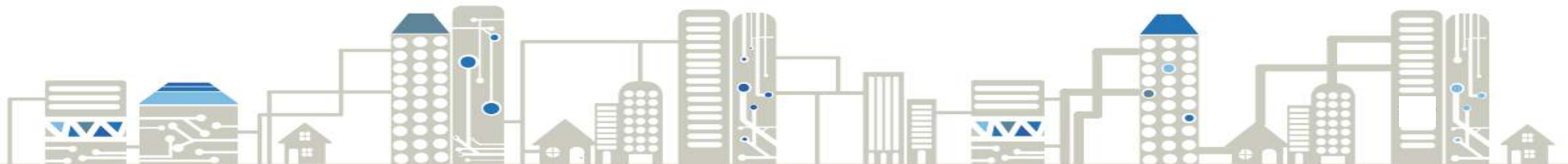
데이터처리(2)



CMIKorea

(주)시엠아이코리아

감사합니다.



(주)시엠아이코리아 이상훈

E-MAIL. PRD1210@NAVER.COM